A PERSON-CENTERED GUIDE

to demystifying technology

2nd Edition

MARTIN WOLSKE

A Person-Centered Guide to Demystifying Technology, 2nd Edition

A Person-Centered Guide to Demystifying Technology, 2nd Edition

Working together to observe, question, design, prototype, and implement/reject technology in support of people's valued beings and doings

By Martin Wolske

With special contributions by
Betty Bayer, Henry Grob, Sara Rasmussen, Dinesh Rathi, Stephanie Shallcross, Vandana Singh, and Yingying Han

Windsor & Downs Press ● Urbana, IL

Published by Windsor & Downs Press, Urbana, Ill., part of the Illinois Open Publishing Network (IOPN). IOPN is a project of the University Library, University of Illinois Urbana-Champaign.

Published as part of the OPN Textbook series.

ISBN (2nd Edition, Online): 978-1-946011-36-7
ISBN (2nd Edition, PDF): 978-1-946011-23-7

Please cite this book using the DOI: https://doi.org/10.21900/wd.20.

This book was produced with Pressbooks (https://presbooks.com) and rendered with Prince.

The PDF version of this title excludes interactive content available in the online version. Links to excluded videos are provided in the text locations where they would appear. Most interactive assessments have been remediated to static text, but links to the fully interactive versions in the chapters online have been provided.

Credits

Cover photography: School of Information Sciences, University of Illinois at Urbana-Champaign. Used with permission.

Original cover design by Kaylen Dwyer and Sara Rasmussen, with updates for the second edition by Elizabeth Budd.

Contents

Blue Unit: Computational Tinkering

Rainbow Unit: Networks Big and Small

Introduction to the Book

Background Knowledge Probe

- What is something that you have that you have used in a way it wasn't meant to be used in order to do something you value?
- What is a notable time when you have come together with others to struggle through a complex problem and come out on the other side with a workable solution?

Overview

Historically, women, people of color, and others experiencing systemic oppression have played very active, even leading roles in hands-on work, including essential creative endeavors. And yet, a vast majority of these remain hidden or lost from our memories, and in key ways this is an intentional work of oppression. These are endeavors of home, craft, and trade workers, and also of information, engineering, and design professionals. Meanwhile, research reveals ways in which the K-12 and higher education systems create hurdles to participation in emerging digital technology spaces for women and people of color. We'll explore these points periodically throughout the book. This book seeks to advance a person-centered orientation to technologies in ways that enrich each of our functional diversities and capability sets, and this often starts with recognizing that we can do far more than we're led to believe. Technology has absolutely no ability to help us. None. Let me repeat that: Technology has no capability whatsoever to do anything for us. It's not a person. It's a thing. Things aren't alive; they don't breathe; they don't think. But they can, sometimes, be a useful tool in the hands of people. And the usefulness of that tool is determined directly and indirectly through the shaping of that tool by other people. People are creative, living, thinking beings. They fail. They screw up. They hurt themselves and others. But they also fail forward, learning to do better and better through application of a growth mindset. At least, people do if they are invited into communities of practice that incorporate diverse cultural wealth and knowledge through collective leadership, community inquiry, and critical pedagogy action-research cycles. Too often people are taught to be submissive, to consume things created by others, to be saved by the thing itself. And this is especially so for those not white and for those not male.

> Most girls are taught to avoid failure and risk. To smile pretty, play it safe, get all A's. Boys, on the other hand, are taught to play rough, swing high, crawl to the top of the monkey bars and then jump off head first. By the time they're adults and whether they're negotiating a raise or even asking someone out on a date, men are habituated to take risk after risk. They're rewarded for it. It's often said in Silicon Valley that no one even takes you seriously unless you've had two failed startups. In other words, **we're raising our girls to be perfect and we're raising our boys to be brave.** Some people worry about our federal deficit. But I worry about our bravery deficit. Our economy, our society, we're losing out because we're not raising our girls to be brave. The bravery deficit is the reason why women are underrepresented in STEM, in C-suites, in boardrooms, in Congress, and pretty much everywhere you look. Reshma Saujani, "Teach Girls Bravery, Not Perfection"

This book seeks to advance the fail-forward and growth mindsets of all, especially those who have been oppressed through works advancing the power of others over them. We will work together to discover ways to advance power, both power within and power with others. We will work to advance our technical skills, but also and even more, our progressive community engagement skills, our critical social + technical skills, and our cognitive, information, and social emotional skills. To do this, we will also work to advance our collective leadership through storytelling, and especially counterstorytelling, within a framing of reciprocity.

Overall Objectives

The general learning outcome objectives of this book are to help readers:

- Develop a clear hands-on working understanding of the hardware and software layers of computers and networks. As learners journey through the units of this book, they will hopefully develop a growing comfort and competency: working with the basic nuts and bolts of computers and networks; appropriately integrating components to serve as tools for computational and information processing; and performing basic troubleshooting.
- Evolve a more holistic and nuanced understanding of the **sociotechnical** artifacts we use as a daily part of our professional lives. The hardware, software, human, and social whole that is a digital artifact is greater than the sum of the parts. Beyond developing technical competencies, we need to develop an awareness of, and skillsets to influence, the emergent properties that come from specific combinations of the different social and technical building blocks of information systems.
- Develop a critical approach to **sociotechnical** artifacts. Social systems are constructs of economy, politics, matters of race, class, and gender, social institutions, and other cultural

dynamics. Design, diffusion, and implementation of technical innovations both reflect and shape these social systems. Critically examining social + technical information systems from multiple individual and societal perspectives opens up consideration of idealized expectations vs. actual positive and negative impacts within specific user communities.

- Advance community agency in appropriating technology to achieve our individual and community development goals through a reconsidered digital literacy learning and practice. Far from being just passive adopters of different digital technology artifacts used to find, evaluate, create, and communicate information, as Information Science professionals we have opportunities to initiate and lead communities of practice, leveraging the plurality of our community's social and technical insights.

Humans are analog. We hear, see, and feel in a rich range of frequencies, hues, and senses. We actively create and use these analog circuits to network with others around us. We actively live as highly functioning analog networked information systems. And we do this quite well!

On the other hand, programming code and the computers used to run that code are primarily digital and binary—zeros and ones. Digital, too, is the Internet, cell networks, Bluetooth devices, and other inter-networking systems we generally think of when we speak of networked information systems.

Makerspaces and other Maker-type environments are places to encourage people to gather together to create, invent, and learn. They are analog networked information systems first and foremost. Such spaces often serve as a counter to consumption and individualism. By their very nature, Makerspaces also embrace change. While you may certainly go to a Makerspace to attend workshops on traditional crafts, you may also find side-by-side ways to create using new digital technologies. But always the focus remains on personalization rather than assembly-line production.

As such, cognitive, socio-emotional, information, and progressive community engagement skills are as much or more important than specific technical skills within Maker-type environments. **Cognitive skills** include the ability to logically analyze and organize problems in ways that allow use of **digital** and **analog** tools to help solve them, and to generalize new processes to other problems. **Socio-emotional skills** include the ability to communicate and collaborate with others, along with personal confidence, persistence, and tolerance, in order to tackle complex, ambiguous, open-ended problems. **Information skills** include the ability to seek, evaluate, interpret, and apply relevant and trustworthy information across multiple media. **Progressive community engagement skills** advance our ability to work together in communities of practice advancing and making continued use of collective leadership and individuals' unique cultural wealth and capability sets to help each member better achieve their valued beings and doings. **Technical skills**, then, become a response to needs identified using cognitive, socio-emotional, and information skills. They are just-in-time in-fill learning.

Combined, the intent of this book is not to exclusively provide technical skills as specific learning outcomes. Rather, the primary goal is to use exercises to teach the technical logic of electronics, coding, and networking in a way that can be generalized to various problems at hand and the unique technical skills needed within that context. At the same time, it is to expand our understanding of the social systems that have applied this technical logic to various design problems in ways that have positively addressed some issues while potentially negatively addressing others. The intent is to also help build up confidence and persistence to tackle ever-more-complex technology problems, while discovering that we are shaped by people, including the creators of this book, who worked to create these **sociotechnical** artifacts. And those who created the artifacts were shaped by other people who created earlier works. And you will begin shaping others as you innovate-in-use the artifacts associated with the book in a way it wasn't necessarily meant to be used in order to do something *you* value.

Ultimately, the aim of the book is to prime the readers for a lifetime of co-exploring within Maker-type environments as information professionals. It is to recognize that at their best, these studios bring together diverse populations of people as a collective to address a key issue of shared interest in ways that maximize individual and social benefits and minimize individual and social harms. It is to identify what participation and inclusion mean within the contexts that are unique to each community of practice and each issue of shared interest. It is to begin to identify the everyday technologies that may be unseen and displaced because of an overly narrow definition of what should be considered appropriate, and the local innovators whose technologies might be championed if only they were made visible. And it is to clarify ways each person might serve in the moment as an innovator-in-use of the technologies to make it practical and appropriate for their functional diversities, that is, their ways of better achieving that which they most value being and doing within a given context.

Some Useful Working Standards and Frameworks

The best works in a Maker-type environment are works of community, in community, and for community. They combine active hands-on innovation with individual and corporate reflection — collective leadership with Paulo Freire's action-reflection cycles and the community inquiry model developed initially by John Dewey and Jane Addams and further advanced especially through the works of Bertram "Chip" Bruce and Ann Peterson-Kemp. Combined, the outcome is development of a critical conceptual understanding of innovation-in-use from a person-centered perspective. This is a book meant to be done as part of a community of practice.

To facilitate these works of collective leadership, action and reflection, and community inquiry, those working through this book are encouraged to test out a range of community of practice standards and frameworks that might apply within our professional practice moving forward. What follows is a short introduction to each. We'll revisit and expand upon these frameworks as we go through the book.

Community Inquiry in Practice is a virtuous cycle that includes asking initial questions, joining in active co-learning, reading the words and worlds of knowledges shared in a range of ways, internal reflection, group discussion, critical questions, and journaling.

Community of Practice

A core idea within the social constructivism of John Dewey and others as incorporated throughout this book is that learning is constructed by social interaction between an individual and the social environment. It cannot happen by an individual doing independent work. Sometimes this learning community comprises a range of people within a local community. At other times, it is a community of practice formed by a group of people engaging in a process of collective learning. While in both of these learning community examples the objective is knowing and learning so as to achieve a specific human endeavor, the means by which this objective can best be achieved varies considerably. As this book is written first and foremost as a textbook for use in classrooms, after-school sessions, workshops, and other Maker-type studio environments, the structure of, and activities within, the book are designed to especially facilitate communities of practice.

In "Communities of practice: a brief introduction," Etienne Wenger provides this nutshell description of the term:

> Communities of practice are groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly.[1]

1. Wenger, Etienne, "Communities of Practice: A Brief Introduction," October 20, 2011, 1. http://hdl.handle.net/1794/11736.

Members of a community of practice are people with a shared domain of interest that distinguishes them from other people as it applies within this context. This may be a multigenerational gathering of people all living within a specific local neighborhood, a gathering of people living within different neighborhoods of a diverse city who all are part of a local club, a gathering of professionals from across the nation or world meeting using online technology, and so many other communities of practice. They are people who engage in joint activities and discussions to share information and help each other, whether in a specific physical location at which they meet on a daily basis, or via an online forum in which they meet synchronously periodically with other asynchronous meetings as infill. In so doing, they commit time and effort to work as formal or informal practitioners so as to develop a shared body of resources. Your past and future interests may vary greatly, but for a time and for a specific purpose as related the contents of this book, you may find it of great value to become a community of practice within this context.

Collective Leadership

When people gather to mobilize human, cultural, and technological resources in ways that address opportunities and challenges of common interest and for the common good, then bidirectional learning, joint action, shared responsibility, and mutual accountability have an opportunity to blossom. This shift away from prioritizing individual change agents to instead emphasize collective leadership serves to facilitate the crossing of boundaries of all types, including age, race, gender, income, culture, and religion. Through a cycle of preparation through trust building, of co-constructing a purpose and strategic plan, of implementation through allyship and collective action, and of sustaining of the work as part of the community fabric, robust strategies and partnerships come together to advance long-term impact.

[A component has been omitted from the PDF edition of this book because it is not fully legible in this format. View the poster, "Collective Leadership for Collaborative Community Action: A New Framework for Library Engagement around Digital Literacy," online.]

As your community of practice work together through the units, you'll find various "*Do Something New!*" innovation-in-use/remix prompts encouraging you to make use of collective leadership to mobilize the sessions of a unit in ways that may address opportunities and challenges of common interest, even if just as a small prototype of a larger collective leadership cycle down the road as you enter into other communities of practice. There's an International Leadership Association interview of leadership educator and practitioner Kirstin Phelps and textbook author Martin Wolske further highlighting digital leadership development through technology education.[2] And the Innovation Center for Community and Youth Development and the Kellogg Leadership for Community Change provide a useful starting guide on collective leadership for community change.

2. Phelps, Kirstin and Wolske, Martin. "Pause for Pedagogy: Perspectives on Digital Leadership Development from Technology Education." International Leadership Development Interview, June 13, 2018. https://youtu.be/Q4dCUie0mUw

Community Inquiry

Inquiry-based learning is open-ended, democratic, participatory engagement connected to people's values, history, and lived experiences. It builds from John Dewey's framing of logic as the theory of inquiry. Community inquiry as framed in this post by Bertram "Chip" Bruce further emphasizes collaborative activity and knowledge creation conducted of, for, and by communities as living social organisms. This book and the readings and videos linked within transmit information to others. The various prompts and comprehension checks add a starting level of interactive communication that hopefully grows with in-person question-and-answer opportunities and workshop/course feedback for improvement. But as Bruce notes in his newest book, *Beyond the Classroom Walls*, what is conveyed through the transmission and interaction modes of communication is a partial message. The expertise and lived experiences of the sender may assure that this is solid data and information based on reliable research and practice. But it is essential that we continue to bring together the work of the mind and of the body, of the classroom and the lab of that which comes from lived real-world experiences.

> In contrast with the transmission and interaction models, the transaction model does not assume that we even have a message to communicate, or that one party could hold such a message. Utterances are not just shared, but co-constructed, and out of that, meaning is constructed. [...] It is the norm for human conversation that an utterance initiated by one person can be completed by another. Speaking overlaps so that it is unclear even what each individual has heard. The initial speaker may alter meaning by stopping mid-sentence, shifting their gaze, or changing intonations. Other participants show their assent, interest, or disagreement only partly through words, using also facial expressions, gestures, or eye glances. The content, such as it is, emerges only as the talk proceeds; it does not exist a priori. (Bruce, "Models of Communication", pg. 13.[3])

The classroom and formal learning remain important, but must be complimented by the informal learning of life in practice. Readers of this book hopefully have the opportunity to come together with other readers taking on transactional communication co-constructing information and knowledge—information that has become actionable—through community inquiry. And as you collectively work through the social and technical aspects of each session, you inquire together about—that is, ask probing questions with regard to—your opportunities and challenges in bringing this learning to action in your lives. As you enter into collaborative investigation and creation, you also will need to regularly set aside time for individual and collective reflection and discussion, leading to a new cycle of asking, investigating, creating, reflecting, and discussing. These cycles don't always happen in a set order. But the best community inquiries strategically incorporate each of these into their works, even when they sometimes seem out of place or of lesser value at a given moment. This community-driven inquiry can also lead to inquiry about the nature and meaning of the community itself, and the webs

3. Bruce, Bertram C. *Beyond the Classroom Walls: Imagining the Future of Education, from Community Schools to Communiversities.* Lanham: Rowman & Littlefield, 2022.

of privilege and oppression that shape and are shaped by the community and those impacting that community.[4]

The Community of Practice framework above is presented as a likely starting understanding of the type of learning community many users of this textbook may be joining into as they travel through the book. The Collective Leadership framework then highlights the importance of crossing boundaries of all types to better achieve collective action in a way that advances the community fabric. Community Inquiry is a framework that further emphasizes the social organism that is community. While this book is an outcome of two decades of work bringing digital technologies and literacies into diverse communities, especially focusing on those historically marginalized, key to each has been the community inquiry practitioner dialogue in community, with community, and for community. This book, then, is intended to help advance an understanding of the nuts and bolts of analog and digital hardware, software, and networks, while also questioning the underlying assumptions of specific sociotechnical artifacts so as to facilitate the deepening of the community fabric. In exploring the book using the frameworks of Community of Practice, Collective Leadership, and Community Inquiry, the intent is to help the learning community consider the ways in which the social and technical chapters within each session of the three units of this book may serve as effective infill to advance the additional, community-specific learning outcome objectives.

Action-Reflection

Paulo Freire was an adult educator and philosopher from Brazil, who in the 1960s worked to develop educational projects and went on to significantly inform and advocate critical pedagogy. He not only taught reading and writing to illiterate adults, but at the same time worked to help raise awareness of the agency people had to bring a new reality into existence. He used techniques that would be familiar to us today: show a picture, show a word, help people to pronounce the word and associate it with the picture, connect the syllables of the word with specific sounds, generalize to other words. But as Freire went through these steps, he also encouraged learners to combine syllables in unique ways to create new words. And he encouraged them to see how the old word was often associated with objects that served to oppress the learner (for instance, moving from learning the word brick to considering that the learner manufactured for elites who paid them a sub-living wage and then used the brick to build walled fortresses that kept wealth in and others out). At the same time they made connections of syllable sounds with new words, they also made connections between creating their own new words with creating new, more just realities.

While Freire developed his approach to link together small group reflection with positive action for change and development within an adult literacy context, he and many others after him have taken this action-reflection critical thinking approach into many different arenas, including the sociotech-

4. One detailed look at community inquiry in practice can be found in *Youth Community Inquiry: New Media for Community and Personal Growth*, eds. Bertram C. Bruce, Ann Peterson Bishop, and Nama R. Budhathoki (New York: Peter Lang, 2014). https://doi.org/10.3726/978-1-4539-1201-0.

nical realm. We first must move away from a "banking" model of education in which expert teachers deposit their knowledge into passive students. Instead we must enter into a problem-posing educational process that launches a dialectic approach to knowledge combined with abductive logic to facilitate a work of *conscientization*, that of perceiving in new ways the social, political, and economic contradictions within existing social reality and then taking action against the oppressive elements to bring about a new reality.[5]

Core to each unit of the book are cycles of critical reflection and hands-on action. During the reflective inquiry phase of the cycle, we first learn the words and concepts underlying technologies, and the social contexts that have shaped and are shaped by these technologies. Individual notetaking is an essential aspect of this inquiry—jot down several key points from the readings and videos, a couple insights that emerged from these, and a question or two still lingering in your mind. Individual inquiry provides a critical contribution as a session moves to a second level of reflection using community inquiry in which diverse perspectives bring forward a more holistic investigation of text within context—words within worlds. Each session incorporates hands-on activities within learning communities to help each person identify with key technical aspects of these sociotechnical artifacts. In so doing, we more fully sense the nuances of the situations being covered in the unit by sharpening existing, and introducing new, understanding of the specific sociotechnical artifact and the broader sociotechnical systems around us. In this way, the initial codification of the words and concepts under exploration that have now been contextually decodified are recoded. Our increasingly nuanced understanding these structures comprised of interdependent social and technical elements reveal relations and properties which uniquely emerge by their function as a whole. Only in such critical action and reflection as part of community inquiry using collective leadership can we advance a more liberated community fabric. Capturing these through journal reflections to conclude the inquiry cycle for a session provides a snapshot that can be taken into subsequent action and reflection inquiry cycles throughout the book and into our broader work as information professionals.

Pair/Triplet Programming

Pair programming is increasingly common in software development. Two programmers collaborate on design, coding, and testing, with qualitative evidence suggesting the subsequent design is better, resulting in simpler code that is easier to extend. Further, whether the pair programming occurs between two novice programmers, between a novice programmer and a more experienced programmer, or between two experienced programmers, people learn significantly more about the system and about software development as both participants bring in unique insights, as long as what is being done is not a repetitive task. Conversation between the programming pair can occur at many levels as the driver working at the keyboard takes charge of all changes made in the program and the navigator observes all the code that is entered, considers coding options, works to spot and address problems,

5. pg. 35, Paulo Freire, *Pedagogy of the Oppressed, 30th Anniversary Edition.* 2000, Continuum International Publishing Group

considers and recommends simplifications, helps with programming style, and designs and verifies testing.[6]

Extensive testing of the hands-on activities throughout this book have indicated the substantial value of collaborative programming in small groups of two or three people. Further, while pair programming is specifically associated with computer programming environments, small group collaborations in which one drives the specific hands-on work and the other(s) navigates has also proved very useful in doing the electronics and networking exercises as well. Throughout, it is essential that participants switch roles regularly from driver to navigator and back. This technique is not about the expert doing while the novice observes. It is about collaboration that recognizes the unique expertise each brings to bear, including bringing in new sites when this is a person's first time doing something.

The Toolkit

In working through specific hands-on activities in this book, we'll make use of a range of items, from individual components like light-emitting diodes and switches, to complex circuit boards like the Circuit Playground Express microcontroller and the Raspberry Pi microcomputer. There are many variations of these components and the supporting electrical conductors and tools that may shape how you can or should work with various parts to achieve hands-on activities. As a carpenter, general contractor, and overall tinkerer, I make regular use of both general- and special-purpose tools. However, the focus of this book is to foster development of a more holistic and nuanced understanding of the hardware, software, human, and social whole of our sociotechnical devices, an understanding that is greater than the sum of the social and technical parts. The tools we'll use for the exercises in this book, then, are general-purpose ones specifically chosen in support of a range of different activities that have proven solid in promoting this holistic and nuanced educational work advancing a broader sociotechnical mindset. These tools have also proven adaptable by a range of audiences as they work to strengthen those aspects of their technical, cognitive, socio-emotional, informational, and progressive community engagement skills which need honing to implement this mindset as part of collective action and reflection cycles for social good.

As selected portions of this book are used in my various other courses and workshops and have proved useful as standalone sources for use around the globe, the tools have been selected as ones that can also be available in a drawer or cabinet for use when a specialized need requires. Many of the exercises in the Rainbow Unit, for instance, could be done on any GNU/Linux computer, and potentially also on Mac OSX and Windows computers. Multiple parallel input and output data pins are likely not

6. For more on pair programming, see: Jeff Dalton, "Pair Programming," in *Great Big Agile* (Berkeley, CA: Apress, 2019) 199–200. https://doi.org/10.1007/978-1-4842-4206-3_42. Franz Zieris and Lutz Prechelt, "Does Pair Programming Pay Off?" In *Rethinking Productivity in Software Engineering*, ed. Caitlin Sadowski and Thomas Zimmermann (Berkeley, CA: Apress, 2019), 251–59. https://doi.org/10.1007/978-1-4842-4221-6_21. Eric Snow, Carol Tate, Daisy Rutstein, and Marie Bienkowski, "Assessment Design Patterns for Computational Thinking Practices in Exploring Computer Science," (Menlo Park, CA: SRI International) 2017. http://pact.sri.com/resources.html.

available on such computers. But the introductory and key takeaway parts of the chapter may still prove of value, as they have for my own community workshops and progressive community engagement classes, which have used computers available within the spaces. But even here, I sometimes bring along some Raspberry Pi's, as they've proven of value given the Raspberry Pi Foundation's educationally-motivated design of the computer and its huge catalog of projects whose origins also come from an educational person-centered focus.

The steps and diagrams of activities in this book are based on components purchased through Adafruit industries, a one hundred percent woman-owned manufacturing company. Founded in 2005 by Limor "Ladyada" Fried, the company's goal is to "create the best place online for learning electronics and making the best designed products for makers of all ages and skill levels." [7]

These lists of recommended items were last updated August 2022 and may be further updated on occasion to match revisions made to the technologies of the day.

Orange Unit Items

The following are used throughout the Orange Unit of the textbook. The main package is also used for exercises in session one of the Rainbow Unit.

- A package of electrical components that each participant will use to build their own stable electronic circuit prototype for ongoing use:
    - One full sized breadboard
    - 20 each of 3″ jumper wires and 6″ jumper wires
    - One USB to TTL Serial Cable
    - One diffused RGB LED with common anode
    - Three tactile switch push buttons
    - One breadboard-friendly slide switch
    - Three PNP bipolar transistors
    - Five 470-ohm resistors
    - One 10K-ohm resistor
    - One 16-GB or larger Class A1 microSD card on which the Raspberry Pi operating system is installed[8]
    - One T-Cobbler Plus to attach an individual, breadboard-based prototype platform to a Raspberry Pi
- Additional electronics that might be used temporarily within the Orange Unit include:
    - LED sequins of the same color (two for each participant)

7. https://www.adafruit.com/about
8. For example, the Kingston Canvas Select Plus – flash memory card – 32 GB – microSDHC UHS-I was selected for the fall, 2023, toolkit.

    - One NPN bipolar transistor
    - One 10K breadboard trim potentiometer
    - One 3.3V linear voltage regulator

Blue Unit Items

The following are used throughout the Blue Unit of the textbook. They are also used for exercises in session one of the Rainbow Unit.

- The Circuit Playground Express
- A micro-USB to USB cable to connect the Circuit Playground Express to a personal computer or to a Raspberry Pi
    - In some cases, a micro-USB to USB-C cable may be needed instead, as many personal computers as well as the Raspberry Pi 4 and 400 now provide USB-C ports
- Six small alligator clip to male jumper wires

The Raspberry Pi Kit

The following Raspberry Pi kit is used in the fourth sessions of both the Orange and Blue Units, and as the base of the Rainbow Unit. To keep overall costs down, one kit can serve multiple participants.

- Raspberry Pi 3, 4, or 400 microcomputer
    - If using a Raspberry Pi 3 or 4, an Adafruit Pi Protector case with GPIO labeling is recommended
- Wall outlet to 5V power supply with appropriate microUSB (Raspberry Pi 3) or USB-C (Raspberry Pi 4 and 400) adapter
- Adafruit PiOLED 128×32 display
- A keyboard, mouse, and monitor will briefly be needed, but can be shared amongst multiple Raspberry Pi's.[9]
- A 40-pin GPIO extension cable has proven helpful to provide a safe connection point when multiple different breadboard prototypes are tested using one shared Raspberry Pi. Attaching the ribbon cable from a cobbler or T-cobbler to this extension cable helps prevent wear and tear on the GPIO pins of the Raspberry Pi itself.

9. The Raspberry Pi operating system includes an installation of the Real VNC server which can be enabled through the Raspberry Pi Configuration software. However, initial setup and configuration of the operating system is generally best done using a standard keyboard, mouse, and monitor attached to a Raspberry Pi. Once complete, the Adafruit PiOLED can be used to provide the IP Address currently assigned to the Raspberry Pi, allowing a Real VNC viewer installed on a participant's computer or mobile phone to connect to the Raspberry Pi running "headless", that is without a keyboard, mouse, and monitor attached.

Toolkits and Their Contexts

I grew up, literally, in the family sawmill. I don't remember far enough back to remember being carried into the mill by my mother, or my first ramblings through it as a toddler. But I remember that at age five, I was paid fifty cents an hour by my parents to sweep up. And I remember that as an apprentice, wanting to do so much more but being instructed to just sweep and listen, I started discovering the music of the sawmill. The most alarming of sounds didn't cause a flinch in my parents, but a quiet change in pitch would send them around, rapidly turning everything off.

And I slowly learned the trade in ways that I can't, even now, tell you I know. Some are pretty straightforward, such as how to build a basic toolbox. But some, like how to fix an electronic circuit to keep a light or a power tool up and running, have just become second nature—hidden knowledge that's hard to share with others.

> What might be some of your hidden knowledge? How can you each provide support to the others to help rediscover that hidden knowledge, as so many have done in support of my rediscoveries? How, as a community of practice, can we work to bring these gems of knowledge into a transactional communication to co-create new understandings?

When I was thirteen, the logger who brought us the logs we used in the sawmill to build pallets for local businesses was severely injured. He ended up healing almost completely, but at the time he didn't think he would. As a result, he sold his logging equipment to us at a bargain. And my dad bought it because he figured I was now thirteen, and advanced enough as an apprentice to partner with him in cutting down and preparing trees for use in the sawmill. So I became a lumberjack as well. Many of the tools of the sawmilling trade matched those of the logging trade. But others were unique given the changing contexts of working with felled versus standing timber, of working indoors versus out, of working to do more precision cutting versus approximate ones. I began to learn the importance of general purpose versus dedicated purpose tools. Still today, whether in doing carpentry, general contracting, or broader tinkering on projects, I've learned the importance of choosing the right tool for the right job at the right time, given the evolving context.

> Take a minute to bring to your mind a project you've done recently. Any kind of project—big or small, personal or professional—will do, just so long as it's something you took a leadership role in doing. What tools did you use? Why those tools? What other tools might have been a better fit for the job at that time within that context?
>
> As you go through this book, keep reflecting back on your own projects as a starting context for considering the toolkit being used for the projects of this book, within the time and contexts within which you are doing them. Why these tools? What other tools might have been a better fit for the job, in this moment and within this context?

I was in junior high when we started the logging portion of the family business and I was also just learning how to write computer programs—it was the mid-seventies after all. For a class presentation, I tried adding in digital tools to measure the board feet of the logs. It proved an early discovery of how the old standby analog tools can be so much better in the field than digital tools at times. The formal learning in the classroom showed me the possibilities, while the learning in the field provided essential informal learning regarding the importance of focusing on people first. I was fortunate to have a diverse group of teachers, mentors, community, and culture in which to experience this model of interactional communication, in which the two are unified through collective leadership and community inquiry.

> As you gather within a community of inquiry, be sure to regularly pause to identify those who identify as "students" who are actually the right "teachers" at this time, in this place, and within this context. Work to lift them up as you each work to advance the collective leadership of all. Especially work to cross boundaries of race, class, gender, and other identifiers that so often leave us othering those different from us. If we are to truly advance a more holistic and nuanced person-centered understanding of networked information systems, diversity isn't a nicety but a necessity.

Learning not only included formal learning in the classroom and informal learning within the work of the trade, either. Changing communities, cultures, and contexts further enrich our learning in the wild. Our family was invited to log amongst farms seeking to preserve the woods because my dad strove to carefully and selectively fell trees in support of the ecosystem for both environmental and economic goals. And we started saving more and more of the premium lumber for use on projects around the house as gifts of the trees. The trees were more than just things to be bought and sold. They were "more-than-human people," as Robin Wall Kimmerer shares in her book *Braiding Sweetgrass.* Even after death, trees gift us so much if we take the time to discover it. And, as Kimmerer notes, I began to understand that we need to learn to "remember to remember … to pick up what was left for us: the stories, the teachings, the songs, each other, our more-than-human relatives that were scattered along that path."[10]

Sometime in my late teens or early twenties, through many physical experiences and through spoken and written words, I increasingly discovered how the tulip poplar, or whitewood, has been a central resource for the creation of North America. Even before the landing of the first white Europeans, residents of this land used whitewood in support of their valued beings and doings. And this fastest growing and easiest working of the hardwoods gives itself willingly if we wisely choose which trees to cut down, and to cut it carefully and with respect. And as I learned more, I also discovered it was a "more-than-human" tree. To this day, when I'm feeling low for some reason, I might discover that on a dime I begin feeling better. And if I look around, I'll often discover close at hand a tulip poplar, silently

10. Kimmerer, Robin Wall "Mapping a New Geography of Hope: Robin Wall Kimmerer Keynote," July 27, 2015. https://www.youtube.com/watch?v=QhQKdJHLDcw.

singing its song of support to me. While this is my personal story, over the years I've come to find similar stories told by people from different histories and cultures as well.

In 1997 I began teaching the Library and Information Science course Introduction to Networked Information Systems. In 2000, a service-learning component was added in answer to a call by residents of East St. Louis, Illinois, to provide infill assistance building and upgrading computer labs in support of the community work addressing the digital divide in local neighborhoods. Throughout, at the start of each semester, the following quote by the Rev. Dr. Martin Luther King, Jr. is referenced:

> We must rapidly begin the shift from a 'thing-oriented' society to a 'person-oriented' society. When machines and computers, profit motives and property rights are considered more important than people, the giant triplets of racism, materialism, and militarism are incapable of being conquered.
>
> Rev. Dr. Martin Luther King, Jr.[11]

I've continuously searched for ways to put into practice these words from Dr. King as part of all my teaching, research, and broader community engagement works. Over and over, I've joined into communities of practice going out to address the digital divide and to advance digital inclusion and equity, only to find that I have remained too thing-oriented. And so this book is being created as part of an ongoing search for better ways to become increasingly person-centered.

For that reason, I've built a toolbox using tulip poplar, in which I put the tools used for activities in this book as a reminder that all we have has been gifted to us by nature. While for most, you'll use containers on hand to store your toolkits, I encourage you to bring your creative side—or that of a willing friend's—to bear in personalizing the container, to keep this connection to people and to nature front and center in your minds as you work through the book. Whether in practice or in spirit, I hope that such a visualization serves you, as it has me, as inspiration for this person-centered mindset, by sharing counterstories of wisdom and knowledge the Tulip Poplar brings as one of the most noble of trees.

Layout of the Book

To help us advance our full range of skills—social and technical—needed to achieve our works as craftspeople each session in this book includes two thematically linked chapters, one more social-oriented and one more technical-oriented, devoted to a particular set of activities and associated learning outcomes.

- The social chapter includes a lesson plan with essential works to be reviewed, along with question probes to be incorporated into a professional journal reflection response.

11. pp. 157-158, King, Martin Luther. "Beyond Vietnam." In: Carson, Clayborne and Shepard, Kris, eds. *A Call to Conscience*, 139-164. Grand Central Publishing, 2001.

- The technical chapter includes a number of hands-on exercises and short quiz.
- A third element is highly recommended through your creation of Reading Notes, helping you to bring your own lived experiences and knowledge into conversation with the chapters, associated readings, and videos. As you proceed through each, jot down several important points that especially stood out to you from that text or video, a couple of insights that emerged from these points, and the major questions that still remain.

Each of these thematically linked chapters, along with the various activities, group discussions, and your Reading Notes and Professional Journal Reflections, form one session. Sessions are brought together into several larger units:

- **Orange Unit:** This includes social chapters introducing the book, the underlying perspectives and mindsets, and broader skill sets underlying the framing of a person-centered guide to demystifying technology. The technical chapters introduce the electronics, circuits, and prototyping tools we'll use through the remainder of the book and beyond.
- **Blue Unit:** The Blue Unit is comprised of four social and technical sessions. This includes social chapters exploring the histories of race, gender, and technology, and the mutual shaping of sociotechnical information systems by innovators, innovators-in-use, and those in between, including Adafruit Industries and the Raspberry Pi Foundation. The technical chapters move from individual electronic components to complex circuit boards such as the Raspberry Pi microcomputer and the Circuit Playground Express microcontroller, and explore how these can be used to bring together electronics and programming code so that individuals and communities of practice can shape digital tools in ways that better serve those things they value being and doing.
- **Rainbow Unit:** The four sociotechnical sessions within the Rainbow Unit seek to tighten our growing holistic and nuanced understanding of the tools and technologies we use as a daily part of our professional, community, and personal lives, which has been building within the Orange and Blue Units. This includes social chapters exploring the broader history and design of the Internet, the guiding forces and principles that have shaped and reshaped it, and the new movements and design principles leading to counterstories and person-centered products. The technical chapters explore and prototype ways to network together a range of circuit boards and operating systems, to create a more just networked information system as we use collective leadership community inquiry, transactional communications, and action-reflection praxis across diverse stakeholder groups to identify digital inequalities.

The grouping of thematically linked chapters within specific units builds from two decades of participatory design in and with community, and from the teaching of these concepts and practices in courses and workshops. However, the use of colors rather than numbers for the units, and the lack of chapter labels for each web page, is in recognition of the value of flexibility in how these materials are covered in a specific context and moment in time.

Lesson Plan

As you travel through the lesson plans for each session within the three units, think of these as pre-season training activities within a sports metaphor. Each is meant to strengthen a different aspect of ourselves and ourselves in relation to others. As with pre-season training, by the end of the Rainbow Unit we will have not reached the finish line, but rather the end of pre-season training. It will then be time to enter into the start of a new season of our information science lives with a clearer and stronger person-centered community inquiry mindset and muscle memory facilitating new levels of action-reflection collective leadership as professional taking on the sociotechnical challenges of the day.

Essential Resources:

- Wolske, Martin. “Ethical Electronic Consumerism.” Martin Wolske’s Blog, August 15, 2013. https://martin.wolske.site/2013/08/15/ethicalelectronicconsumerism/.
- Kimmerer, Robin. “Reclaiming the Honorable Harvest: Robin Kimmerer at TEDxSitka.” YouTube, August 18, 2012. https://www.youtube.com/watch?v=Lz1vgfZ3etE.
- Saujani, Reshma. “Teach Girls Bravery, Not Perfection.” YouTube, March 28, 2016. https://www.youtube.com/watch?v=fC9da6eqaqg.
- Dweck, Carol. “The Power of Believing That You Can Improve.” YouTube, December 17, 2014. https://www.youtube.com/watch?v=_X0mgOOSpLU.

Professional Journal Reflections:

1. What social and technical contexts do you bring with you into the reading of the written, spoken, and visual texts shared in this book? Share a remembered story or two.
2. Why might Robin Wall Kimmerer’s talk be an essential resource for this chapter? How might it relate to the other texts in the chapter, and to the remainder of the book?
3. Where might you be in the development of a fail-forward and growth mindset? Why?
4. Where might others around you be in the development of a fail-forward and growth mindset? Why?

Orange Unit: A Person-Centered Launch

At a Demystifying Technology Workshop, elementary school parents sit in front of a desktop computer and practice pair programming along with School of Information Science student.

Table of Contents

Orange Unit Overview

> We must rapidly begin the shift from a 'thing-oriented' society to a 'person-oriented' society. When machines and computers, profit motives and property rights are considered more important than people, the giant triplets of racism, materialism, and militarism are incapable of being conquered.
>
> Rev. Dr. Martin Luther King, Jr.[12]

Rev. Dr. Martin Luther King, Jr. delivered his "Beyond Vietnam" speech at the Riverside Church in New York City in April of 1967. It was given against the advice of many of King's closest advisors. They were concerned, with merit, that bringing together the civil rights, poverty, and peace movements would lose the support of some major donors, influential liberals, and many white supporters. King would not be dissuaded, as he saw all things interconnected. The structures in place were already appropriating the civil rights gains. The above quote ends a paragraph that began with Dr. King referencing a statement made by the late John F. Kennedy:

> Five years ago he said, 'Those who make peaceful revolution impossible will make violent revolution inevitable.' Increasingly, by choice or by accident, this is the role our nation has taken, the role of those who make peaceful revolution impossible by refusing to give up the privileges and the pleasures that come from the immense profits of overseas investments.
>
> Rev. Dr. Martin Luther King, Jr.[13]

12. pgs. 157-158, King, Martin Luther. "Beyond Vietnam." In: Carson, Clayborne and Shepard, Kris, eds. A Call to Conscience, 139-164. Grand Central Publishing, 2001.
13. pgs. 157, King, Martin Luther. "Beyond Vietnam." In: Carson, Clayborne and Shepard, Kris, eds. A Call to Conscience, 139-164. Grand Central Publishing, 2001.

Throughout this speech, Dr. King repeatedly emphasized our important need for "a radical revolution of values."

The primary objective of this textbook is to provide a deep, hands-on sociotechnical dive into technology including electronics, software, and networks, culminating in a holistic understanding of networked information systems. The textbook also explores the methodological landscape of networked information systems, exploring the theoretical assumptions, historical contexts, social structures, and research methods and techniques used by those known and unknown who influenced the design, making, gifting and/or marketing of technologies. Throughout, co-explorers will be introduced to, and make active use of skill sets, frameworks, and standards employed by a wide range of information professionals in selecting, co-designing, appropriating, and innovating-in-use networked information systems.

But might we be at risk of approaching digital technology, literacy training, and programming through dominant paradigms that keep invisible the various ways our digital technology and media are controlled and mediated so as to privilege a few over the many?

Is it possible, in our very efforts to "bridge the digital divide" and build "21st century digital literacy skills," that we are actually deconstructing civil society and civic engagement, and instead furthering magical thinking about technology, a belief in the supremacy of the technocrat, and the centrality of market forces?

While these are questions we'll only introduce over the course of the book, the Orange Unit is our starting point for considering how our everyday works with technologies can serve as a means to rapidly shift from a 'thing-oriented' society to a 'person-centered' society, as so importantly introduced by Dr. King in 1967.

Orange Unit Overview

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with the option to export your responses, use the online version.]

Key Learning Outcomes Self-Check

The key learning outcomes objectives for the **Orange Unit** include the following. As you view these, take a moment to ***rate 1-4*** these items based on where you think you are in your skills development on each of these items, with additional notes as helpful. Keep these at your side as you work through the Unit to facilitate guided inquiry.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough
4: Yet

Technical Skills

You can (rate 1-4):

- Identify core electronic components, such as breadboards, resistors, and diodes.
- Read diagram and schematic drawings of circuits and use them to prototype your own circuits.
- Identify parts of the circuit that are breaking the loop, thereby interrupting the functioning of the circuit.
- Identify the hardware and operating system building blocks that make up all computers.

Information Skills

You can (rate 1-4):

- Explore a range of information sources, guides, and worksheets in support of current research and practice.

Cognitive Skills

You can (rate 1-4):

- Generalize past and current problem solving to new contexts.
- Logically analyze and organize project tasks in ways that allow use of digital tools to help accomplish them.

Socio-Emotional Skills

You can (rate 1-4):

- Communicate and collaborate with others.
- Embrace failure as an essential step in project development, plan for failure, and use failure as a stepping stone (fail forward mindset).
- Analyze current skills, compare to needed skills to accomplish a project, and develop a plan of action to achieve skills development (growth mindset).

Progressive Community Engagement Skills

You can (rate 1-4):

- Use an informal, distributed process to share the responsibility of leadership among all team members of a community of practice in ways that build off each member's strengths and unique insights.
- Understand and respect broad diversities of functioning with regard to analog and digital technologies, and to see this functional diversity not as a deficit, but rather as an inherent and valued aspect of being human.
- Interact as a community of practice in ways that advance each member's capability set and positive freedoms, so as to more effectively achieve and enjoy their valued beings and doings.
- Focus on and learn from the array of often unrecognized and unacknowledged cultural knowledges, skills, abilities, and contacts possessed by a community of practice, and especially those of socially marginalized members.

Critical Sociotechnical Skills

You can (rate 1-4):

1. Analyze how economic, social, cultural, historical, and political contexts, and personal preferences and biases, are embedded into a specific implementation of technology throughout the life cycle of the technology.
2. Evaluate positive and negative impacts that different technologies and specific implementations of a technology have on individuals and social groups from different economic, social, and cultural contexts.

Your Personal Learning Outcome Objectives

Add your own personal learning outcome objectives

Rate Your Personal Learning Outcome Objectives

Take a moment to create a ***1-3 rating*** of your personal learning outcome objectives. Add additional notes as helpful to keep further specific reflections on these specific skills development.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough

Goals	Rating
Your first goal...	

1A: Information Systems

Background Knowledge Probe

- What is the first thing that comes to mind when someone brings up the term "information system" in a conversation?
- What components, if any, do you consider to be part of that information system?

What is an Information System?

> Information System: An integrated set of components for collecting, storing, and processing data and for providing information, knowledge, and digital products.
>
> Encyclopaedia Britannica

As we work through the Orange Unit, we'll explore the basic parts of our everyday information systems. Over the millennia of human existence, **sociotechnical** devices have been created for collecting, processing, storing, disseminating and using data, the symbols used to represent real-world entities. Sociotechnical strategies have also emerged to advance understanding of data in the form of information and to make this information actionable in the form of knowledge. The accumulation of knowledge and a refined understanding of this accumulated knowledge advances what some refer to as wisdom. The data, information, knowledge, and wisdom pyramid has proved very useful in the information sciences as a means of investigating how different aspects of this pyramid have influenced the shaping of individuals, communities, societies, organizations, and governing bodies. [1]

For much of our history, humans have worked with mechanical devices to create information systems. Consider for instance Johannes Gutenberg's printing press or Herman Hollerith's census calculator as but two examples of Euro-American inventions. But you can go back much further, to devices like the

1. Kate McDowell. "Storytelling wisdom: Story, information, and DIKW." *Journal of the Association for Information Science and Technology* 72, no. 10 (2021: 1223-1233. https://doi.org/10.1002/asi.24466.

abacus, and even earlier still — to find innovations of mechanical technologies from around the world that have facilitated progress along the data, information, knowledge, and wisdom pyramid. Carolyn de la Peña provides a very useful lens in her article "Slow and Low Progress":

> Defined here as the material or systemic result of human attempts to extend the limits of power over the body and its surroundings, technology has been an essential tool in our ability to feed, clothe, house, and protect our bodies. It has unified individuals across expanses of time and space in a manner that has extended pleasure and prevented pain.[2]

Rapid growth of the human population in the 20th century, combined with growing organizational and governance structures shaped via the industrial revolution and the impact of two major world wars, led to the rapid advancement of electronic computing systems. Key electronic systems were developed during World War II. After the war, new searches began for innovative uses of these systems and the components from which they were created. Business, academic, governmental, and other research institutions began a dedicated effort to consider what to do with the machineries of war, things like large tractors and poisonous gases. During World War I, there were posters proclaiming that Uncle Sam wants you to have two chickens per person. In World War II, Victory Gardens were seen as a civic duty to support the war effort. Entering the 1950s, the call was made by president after president, asking that we do our duty and buy from grocery stores in support of the modern farm that made use of those former war machineries to create innovative new agricultural systems.

Take a moment to consider again what comes to mind when someone brings up the term "information system" in conversation — in particular, in relation to these key transitions in information systems described above. What are the most beneficial outcomes that come to mind with the inventions of these information systems? What are the most problematic outcomes? Why do these come to mind? How do these come to mind?

Mechanical and electronic information systems are tightly interrelated social and technical systems. The mutual shaping of social and technical components is highly complex and has seen, unseen, and unforeseen positive and negative impacts on individuals and the things that they value being and doing. We shape them and we are shaped by them. Others shape them and are shaped by them. And yet, too often we miss seeing inside the black box that we are an active part of, even if we don't know we are.

And so we launch the Orange Unit by discovering the building blocks that make up basic electronic circuits. It's time to meet that often deeply hidden essential component of electronic systems face-to-face. We will use some instrumental approaches to apply deductive reasoning to discover relationships between individual components within an electronic circuit. We will begin to learn the key terms, con-

2. Carolyn de la Peña. "'Slow and Low Progress,' or Why American Studies Should Do Technology." *American Quarterly* 58, no. 3 (2006): 915–41. http://www.jstor.org/stable/40068398.

cepts, and practices underlying the creation and use of these components to build electronic tools and larger systems.

But at the same time, we start Session 1 also beginning to test out some practical approaches to knowledge by exploring the context and experiences of others' lifeworlds through inquiry using inductive reasoning. And, on occasion we will also find ourselves using abductive reasoning, seeking creative leaps to reveal hidden forces and structures underlying the mutual shaping of the information systems around us.

Lesson Plan

Essential Resources:

- Urban, Ivette Bayo. "We Are Our Stories: Is Technology Rewriting Our Values?" YouTube, December 23, 2016. https://www.youtube.com/watch?v=G7oN1zdS8HE.[3]
- Bate, Alex. "Electronics 101.1: Electricity Basics." Raspberry Pi, October 16, 2018. https://www.raspberrypi.com/news/electronics-101-1-electricity-basics/.
- Adafruit Industries. "Collin's Lab: Breadboards & Perfboards #Adafruit." YouTube, February 28, 2014. https://www.youtube.com/watch?v=w0c3t0fJhXU.

Key Technical Terms

- Closed and open **circuits**
- Key conductive materials such as **breadboards** and **jumper wires**
- Key electronic components, including **resistors**, **switches**, and **Light Emitting Diodes (LEDs)**, which include both **anode** and **cathode** legs
- Key terms used when working with electricity, including **current** or amps, **voltage** or volts, **watts**, and **ground**

Professional Journal Reflections:

1. Ivette Bayo Urban notes that we are our stories. She also asks, is technology rewriting our values? After reading the text and context of the above essential resources, and before diving into the "Introduction to Electronic Circuits" chapter of Session 1, take some time to reflect on one or two of your own stories, and:
    - Ways these stories have shaped your selection and use of your everyday electronics?

3. TED Talks are licensed under CC-BY-NC-ND 4.0.

- Ways you have shaped these electronics, if any?

2. After learning more about the basics of electronics, revisit and reflect again on your own stories and the mutual shaping of you by these and your/others' shaping of these.
3. What key takeaways did you discover through these essential resources and the hands-on exercises of "Introduction to Electronic Circuits"?

1B: Introduction to Electronic Circuits

Background Knowledge Probe

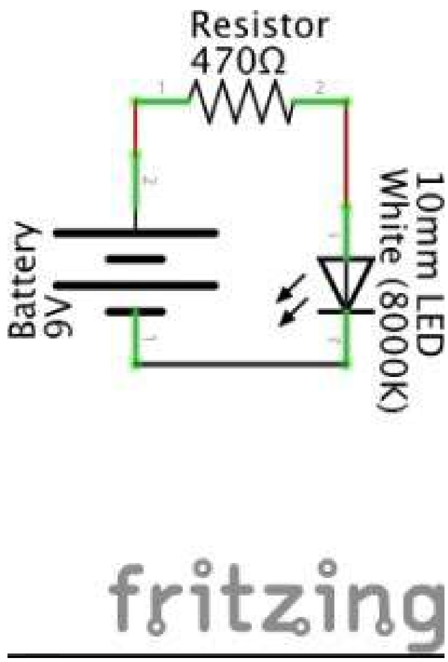
Write a list of the electronics that you make use of on a regular basis.

- Who comes to mind as the designers of these?
- What parts, inside and outside the hood, were used to build each one? To use it?
- What have you done with any items that started to work less than optimally?

Technical Introduction

We do more work with electronics in our daily lives than we might ever imagine. Many of our core tools for daily activities make use of electricity. The remote we use to change channels on the television uses **diodes**, **transistors**, capacitors, resistors, and other electronics. We replace batteries to keep these working, or we plug them in to recharge them. Or we choose alternative options, including software applications, commonly called apps, that we install on our mobile phones, thereby using shared electronic components on a single device, instead of the ones found over and over on each different handheld device. With the growth of DIY — Do it Yourself — we can follow relatively straightforward instructions to build our own replacement electronics, ofttimes expanding our own craft making ethos to this expanded toolmaker realm. This book kicks off with some basic activities in electronics. There are many good resources to get a detailed understanding of how electricity and electronics work. But for what we're doing, only the most basic descriptions of the core components used will be needed for you to jump right in. You don't even really need to remember specific names and descriptions. You can always come back here to glance at names and descriptions later. But as should always be the case, feel encouraged to read strategically, not linearly. Do a general mining of this text to help you get started in the upcoming activities, then come back as needed to dig in a little deeper on this or that. As Miriam Sweeney suggests, "instead of cruising along the narrative, you need to dive in, find the information you need, and move along." With that stated, let's move forward with definitions of key electronics used throughout the rest of this book.

Example: A Single LED Circuit



An electrical **circuit** is a path in which electrons flow from source to ground. The source is usually measured in **voltage** (the force, expressed in volts) or **current** (the flow, expressed in amps). A **resistor** (expressed in ohms) controls the flow of this source.

When I water my garden from a rain barrel, a full barrel has more pressure than one that is almost empty. That full barrel is the equivalent of a higher voltage **power source**, for instance, one that is 240 volts. A barrel two-thirds of the way full may be more like a 120 volt power source, while a nearly empty rain barrel may be closer to a 5 volt power source.

If I have a spray nozzle hooked up to my rain barrel, I can often use a lever to increase or decrease the water's flow. This is equivalent to the current of the power source. Some of my water hoses are 1/2″ thick, while some are 3/4″ thick. This is equivalent to bigger or smaller ohms of a resistor, respectively, since a thinner hose increases the resistance of the flow of water compared to the thicker hose.

As a side note, in electrical engineering, **Ohm's Law** provides a basic equation I=V/R. This states that current (I) is equal to voltage (V) divided by resistance (R). And as all Trekkies know from their days with Jean-Luc Picard, resistance is not futile, it's essential! Beyond this, don't worry if you are ready to put aside these particular details.

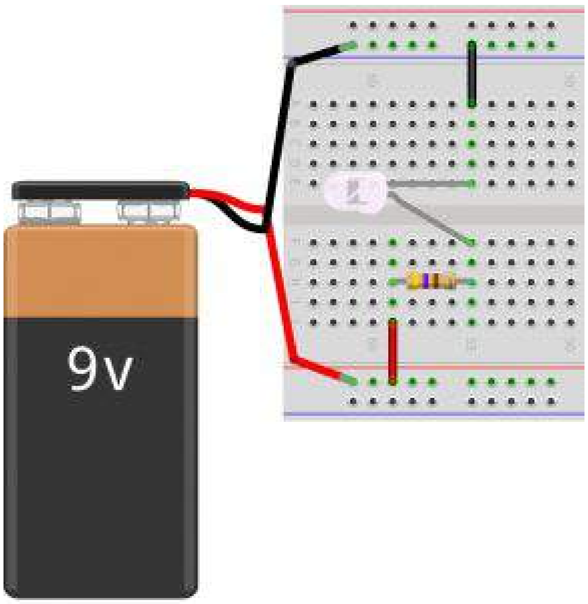
Schematics are commonly used to draw out the path used by the electronic components to complete a circuit.

As an example, the Fritzing image above is a schematic with three electronic components:

- a battery
- a resistor
- a **light emitting diode (LED)**

The lines between are not electronic components, but rather are some form of conductive material like a metal wire that is used to carry the current from component to component. In the schematic, a positive current leaves the 9 volt battery and passes through a 560 ohm resistor before connecting to the positive charge of a 10mm white LED. From there, it passes through the negatively charged leg of the LED to the ground connection of the 9 volt battery, completing the circuit. Only when this circuit is fully and properly completed and the battery has sufficient charge to dispatch a current will the LED turn on. (Schematics and **breadboard** diagrams in this book were primarily created using the open source program Fritzing.)[1]

Schematics don't tell us anything about how the circuit is actually built, and indeed the visuals for the different components needed are confusing artwork until their interpretation is provided. But with a little bit of practice, schematics become a unique conceptual information source. We can use design thinking and rapid prototyping to physically make the circuit using specific parts on hand or acquired to achieve something we value, even if it's just good enough to help more than bother us.



Often times, when you look for a plan or pattern to use electronics to perform a certain task, instead of a schematic, you'll find a physical diagram of the parts as demonstrated in the Fritzing image shown above. This diagram is based on someone's layout choices using available components to actually implement the circuit drawn in the Fritzing schematic at the start of this example. In the physical diagram, we see electrical components connected using a solderless **breadboard**, a plastic board with conductive clips under groupings of plastic holes, thereby allowing the passage of current between components.

At the left of this particular diagram, a red wire brings the positive current of the battery to the lower red rail of the breadboard. Another red wire moves it from this rail to row J, column 59. Row H, column 59 then connects the current to one leg of a 470 ohm resistor, which carries that current, now under resistance, to row H, column 55. This brings the current to the positive charge of a 10mm white LED. Row E, column 55 is connected to the negatively charged leg of the LED, which then passes to row A, column 55 where it is connected to a black wire that carries the current to the upper blue rail on the breadboard. A black wire then brings the current to the ground connection of the 9 volt battery, completing the circuit.

As you move forward, you may find yourself exploring and evaluating a variety of online guides, resources, and examples and comparing them to what is shown in the book. (Indeed, take a deep dive through the many examples found at Adafruit Learning System, from which we draw many for this book, for an extraordinarily wide range of electronic systems, along with supporting code.) Sometimes you may find schematics. But more often you'll find diagrams illustrating how someone or a group have put the circuit into practice. Know that diagrams may be rigid as they tend to focus on a specific task in a specific configuration in a specific environment that suits a specific cohort, community, or culture. Schematics, on the other hand, are found in more professional settings and as such can be

1. Fritzing breadboard graphics are licensed under CC-BY-SA 3.0.

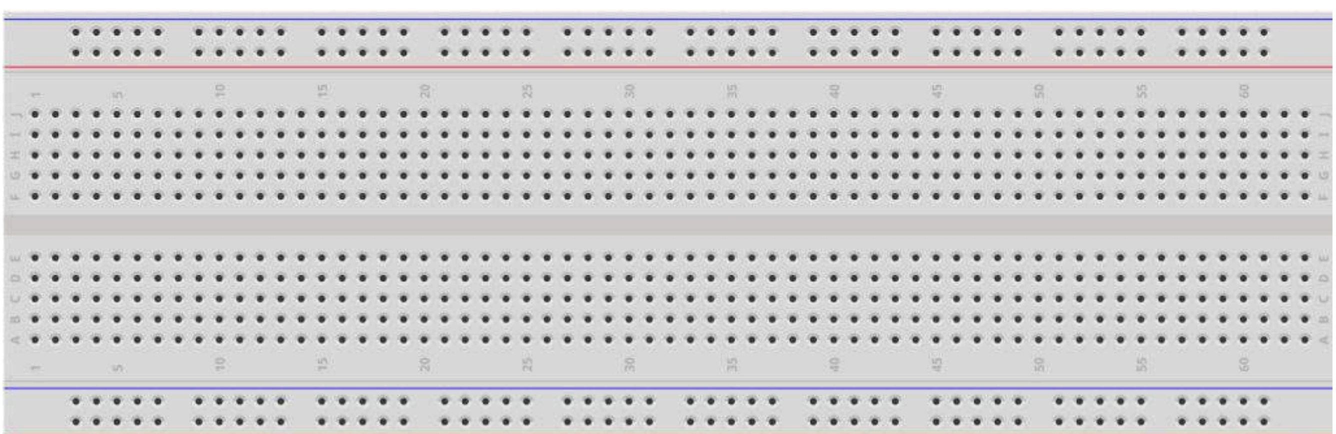
applied to a greater range of applications. These schematics, after doing research to define the components illustrated using specific visualizations, will also prove very helpful for problem solving or rapid prototyping for all innovators, regardless of experience.

It may be helpful to think of schematics as standards used for many different applications, projects, and occupations. Schematic drawings are relevant for many users from electrical engineers to K-12 students. For example, circuits and schematics are part of the Next Generation Science Standards (NGSS) Energy Unit for elementary school students in grade 4.[2]

Underneath the Hood of the Breadboard

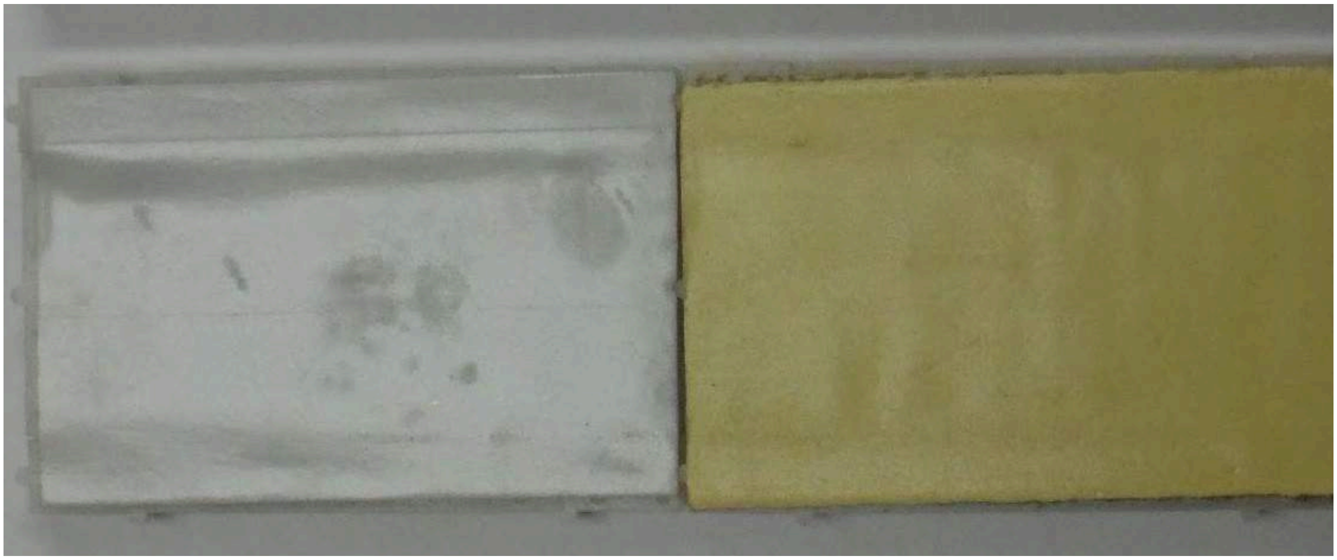
A solderless breadboard is a plastic board used to create models or prototypes of an electric circuit. They're solderless because you can easily push wires into the provided holes and later pull them out to move them into new configurations. They come in different shapes and sizes. The most commonly used "full sized" breadboard is 6 1/2″ long and 2 1/8″ wide.

Below, you'll note several different images of a full sized breadboard. In the first image, you see the typical working view of the breadboard.

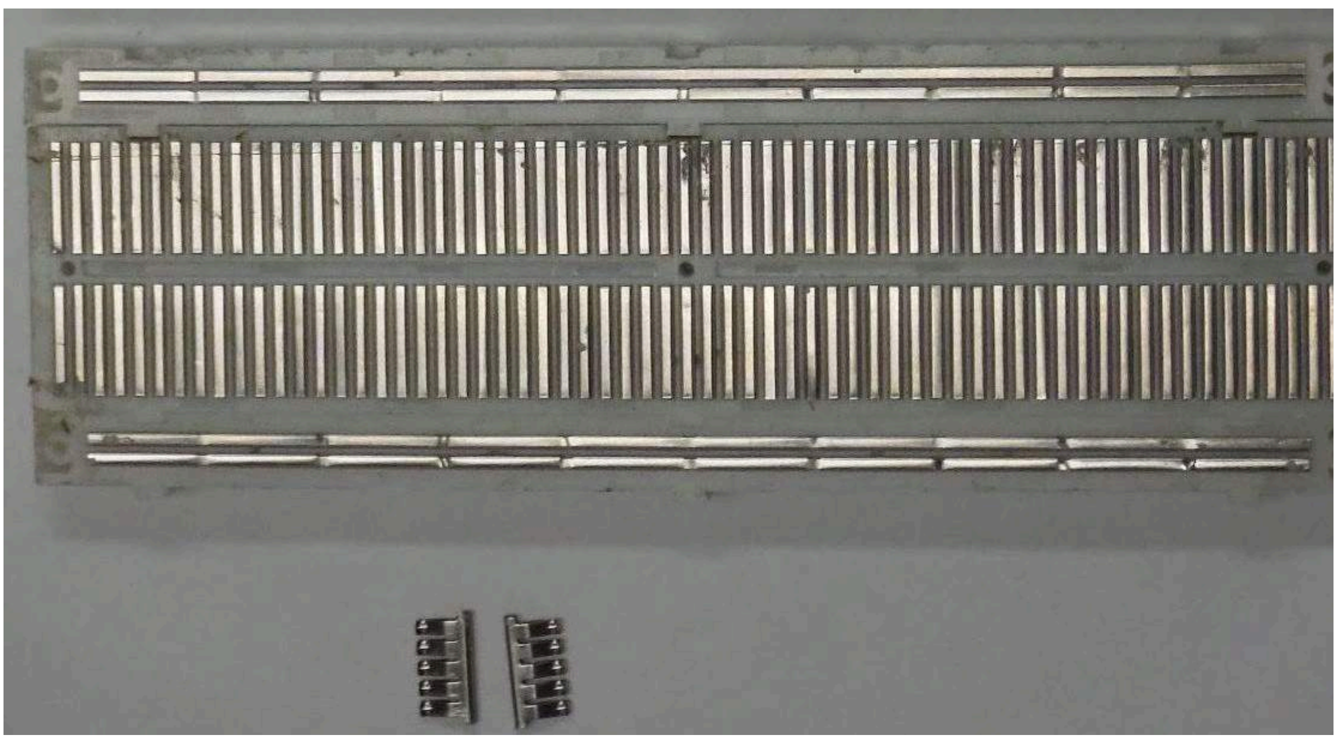


In the next image, we see the bottom side of a breadboard. The left half of the center image has the sticky covering revealed, while the right half has the protective covering installed. If the breadboard is moved around a table or desk, a protective covering remains on the underside of the breadboard (seen in the right half of the center breadboard). You can remove that covering to enable the breadboard to stick to a surface (seen in the left half of the center breadboard).

2. See Next Generation Science Standards' 4-PS3 Energy for more examples.



In the breadboard image below, we see the underside of a breadboard, but this time with the sticky tape fully removed. Breadboards typically come with rails along the edges, with conductive metal clips (two of which are shown at the very bottom of the image) installed so as to carry the flow of current along the extent of each rail.

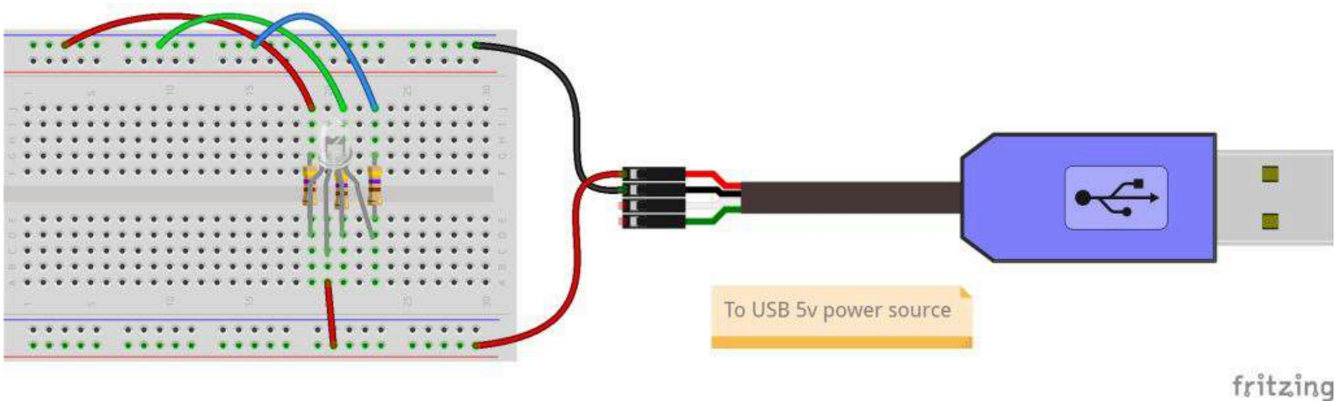


If you compare this image of the breadboard metal clips to the image of the top side of a breadboard, you'll note there's a red colored rail and a blue colored rail on the upper part, and another red rail and blue rail on the lower part of the breadboard. The red rail is typically used to carry a positive source to a circuit or series of circuits, while the blue rail is typically used to carry the circuit or series of circuits to ground. The main part of the breadboard connects together groupings of five holes using 126 conductive metal clips.

Thus, column 1, row A is connected to rows B, C, D, and E of column 1 with one metal clip. Column 1, rows F, G, H, I, and J are interconnected with a second metal clip. Column 1, rows A, B, C, D, and E do NOT connect with column 1, rows F, G, H, I, and J by default. They will only interconnect if a wire or electronic component is plugged into one of the left holes of column 1 and also to one of the right holes of column 1, and is set to pass a current from one connection to the other.

Exercise: Building and Testing an Electronics Prototyping Platform

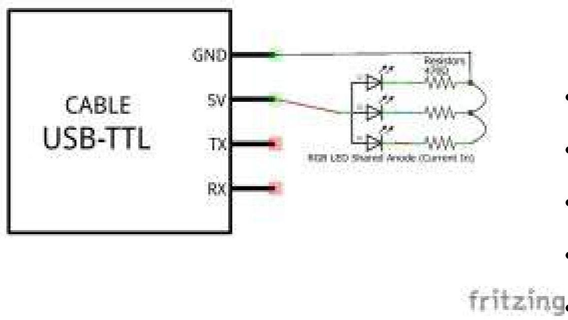
In the previous example, a circuit was created using a 9V battery, as shown in both a schematic and a diagram. Throughout the rest of this book, we will typically not use a battery to power our electronics. We'll start by using the USB to TTL cable that comes with the toolkit, allowing us to connect the USB side to a 5-volt power source. The 5-volt power source itself is likely connected to either a 120- or 240-volt power alternating current (AC) source, which it inverts to 5-volt direct current (DC) power. The TTL side of the cable will only be used for now to provide power through the red cable and ground through the black cable. Later in the book, we'll also use the white transmit cable and green receive cable to facilitate data communications between devices.



The image above provides a diagram of the basic configuration of our prototyping breadboard, in which power is provided using a USB to TTL cable from the toolkit. The basic circuit located on the breadboard illustrates how 5 volts of current pass through conductive material from the USB to TTL cable, conductive material within the breadboard, and jumper wires to an electronic component called an LED, before returning through additional resistors to ground, thereby completing the circuit.

In the above diagram, it is important to note that the only <u>electronic</u> components are the 470 Ω (ohm) resistors and the Red/Green/Blue (RGB tricolor) LED. The breadboard, jumper wires, and USB to TTL cable are not electronic components, but rather are forms of conductive material used to carry the current from component to component.

Another way to illustrate a circuit is with the schematic below, helping to visualize the single physical LED as three separate functional LED chips incorporated into a single package. Each of the three LED chips are themselves small **semiconductive** materials in which a Light Emitting Diode electronic component can be embedded.

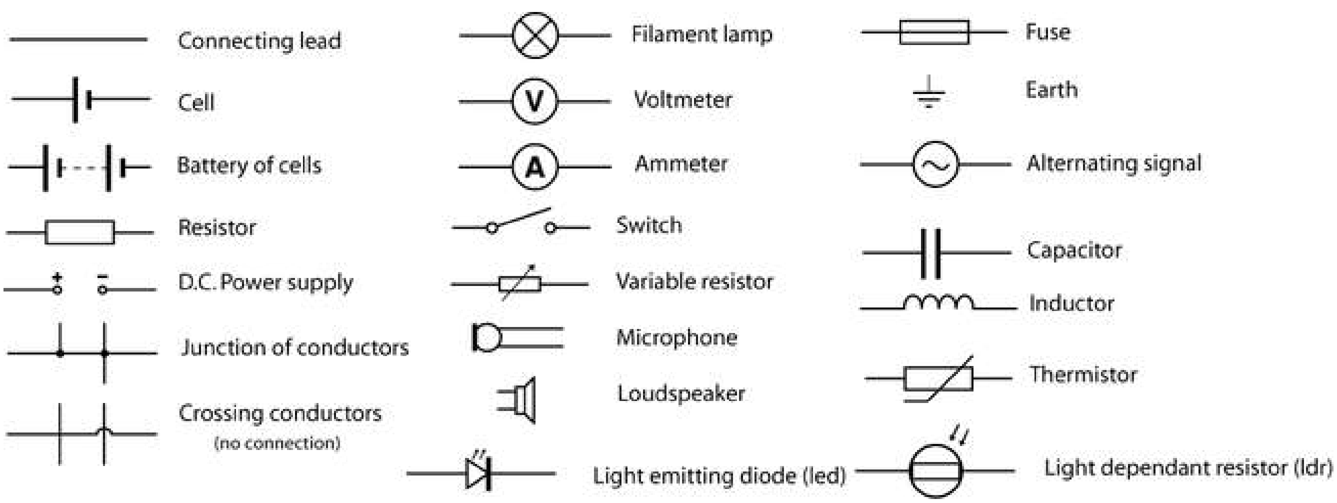


In the schematic, there are several standard symbols used:

- The United States standard symbol for a resistor;
- The symbol for a light emitting diode;
- The symbol for a connecting lead;
- The symbol for a junction of conductors; and
- The symbol for an integrated circuit.

For some, such as an **integrated circuit** (IC) schematic, IC subcategories are further developed for the specific specialty component being used, as is the case for the [USB-TTL Cable](#) from Adafruit.com. Inside the blue USB plug case is a USB↔Serial conversion chip that can work with drivers to provide **Universal Asynchronous Transmit Receive** (UART) communications which we'll start using in this exercise.

Together, these standard symbols are a helpful way to view both diagrams and schematics when starting a new electronics project.

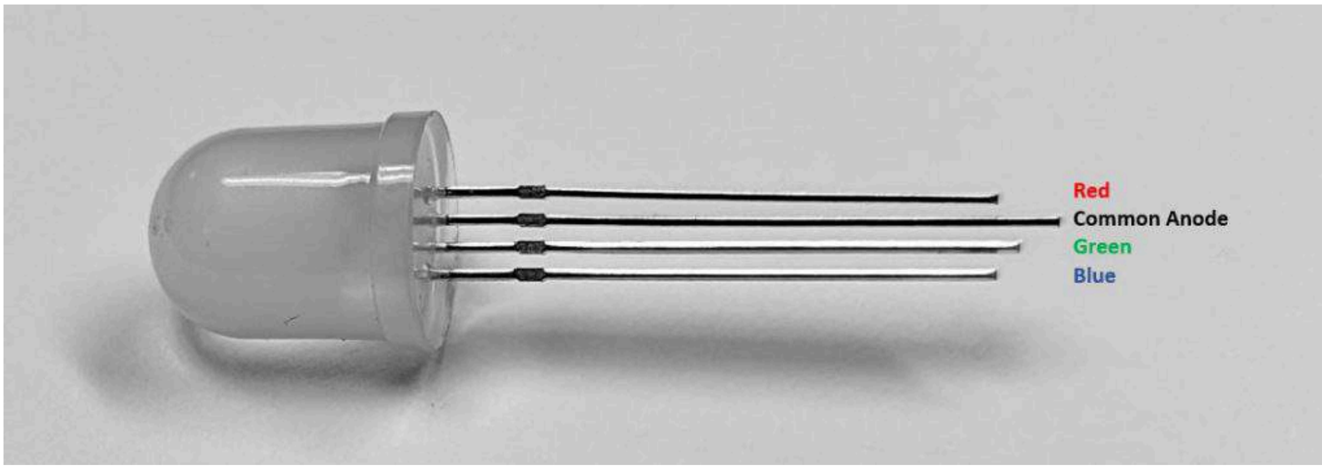


A set of example circuit elements and their associated symbols in common use Source: [Wikibooks](#) licensed under: [CC BY-SA 3.0](#)

In the above figure, you'll see a range of symbols commonly found within schematics and diagrams. Note that in this chart, the resistor schematic is a bit different from the one found within the Fritzing schematic for this exercise, as the Lumen Learning modules use the European standard. The Fritzing diagram uses the symbols common within the United States. While most are the same, a few, like resistors, do differ[3].

For the breadboard exercises in the book, we will be using a RGB (red/green/blue) LED. This three-in-one **LED** makes use of one leg that can either be the common **anode** with a shared power source for the RGB legs, or the common **cathode** leg with a shared ground source for the three LEDs.

For exercises throughout this book, we are using the diffused RGB (tri-color) 10mm LED with common anode from Adafruit. The anode (longest) leg can accept up to a 5-volt power source. The anode leg then connects to each of the red, green, and blue LED "chips" ("chip" is another term for an **integrated circuit**). Each chip connects to its own metal wire, which serves as the cathode leg for that LED. For exercises within this book, we then individually return each chip to ground in parallel through three resistors.



In the image above, you will find the shorter red cathode metal leg at the top, then the long anode metal leg, the middle-length green cathode metal leg, and finally the short blue cathode metal leg at the bottom of the image.

Steps

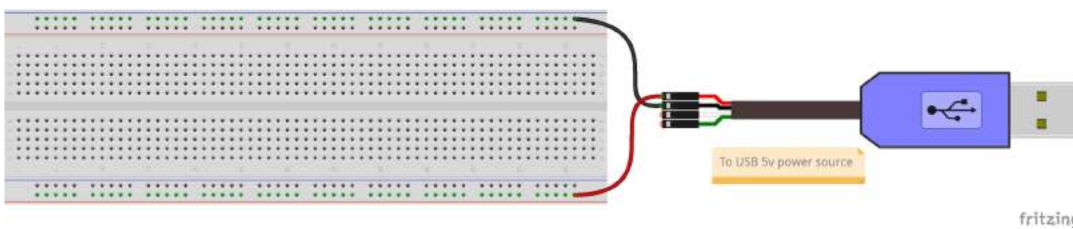
1. Take the needed parts out of the toolkit:
   - Full-size breadboard (illustrations will use either a half- or a full-size breadboard for better visualization of a given step)
   - USB to TTL cable
   - RGB LED
   - Three 470 Ω or 560 Ω resistors
   - Six male/male jumper cables. Cables, breadboards, and other conductive

3. See the Department of Energy National Training Center document on schematic symbols for further information at: https://sites.ntc.doe.gov/partners/tr/Training%20Textbooks/08-Engineering%20Symbology,%20Prints,%20and%20Drawings/4-Mod%204-Electronic%20Diagrams%20and%20Schematics.pdf

materials have one of two different end connector types:

- Male-type ends have an exposed conductive metal extending beyond the protective case.
- Female-type ends have a conductive metal hole within a plastic or other non-conductive case into which a male end is inserted.

2. Connect the TTL to USB cable on the far right of the breadboard to provide 5-volt (5v) and ground (GND) sources to the board. In the illustration below, you'll notice the Fritzing software tool used throughout this book for electronics design projects provides a green highlighting of all female holes, which are also now accessible to the male end which was inserted into one of the holes in the conductive series.

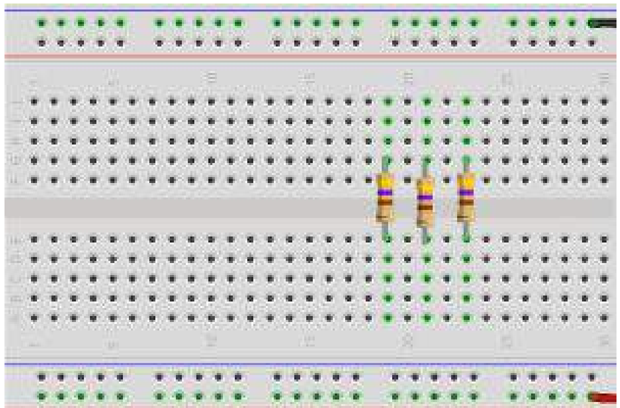


- Position the breadboard with column 1 towards your left.
- Use a male/male jumper cable to connect the female 5-volt (red) TTL connector of the USB to TTL cable to the lower red rail of the breadboard. The above illustration highlights that all holes along this red rail can now be used to connect to this 5-volt power source.
- Use a male/male jumper cable to connect the female ground (black) TTL connector of the USB to TTL cable to the upper blue rail of the breadboard. The above illustration highlights that all holes along this blue rail can now be used to connect to this ground source.
  - NOTE: The color-coding of jumper wire insulation does not impact the conductive capability of the copper wire within the insulation. However, it is often used strategically and sometimes through coding standards to indicate a purpose.
  - In the context of the breadboard and USB to TTL cable, red is commonly used to indicate a power source within a direct current (DC) circuit, while black or blue are used to indicate ground.
  - In the context of the book, it is also used to indicate the red, green, and blue legs of the RGB LED.
  - In practice, packages of jumper wires may be all of the same color or only a few different colors, and can be used as avail-

ability dictates.

- Depending on the type and level of color blindness, some individuals may need a mix of diverse individual and collaborative approaches to fail forward. This may include strategic color-coding selection, process of elimination, and labeling.

3. Install three 470 Ω resistors on the breadboard.

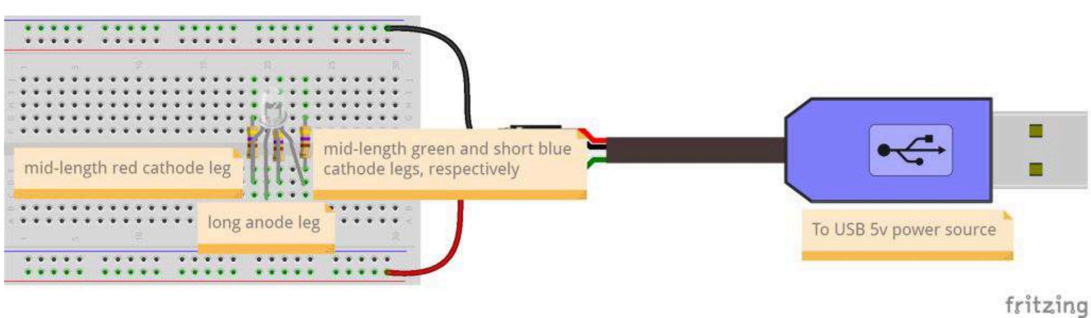


- Tightly bend down 90 degrees the two legs of each 470 Ω resistor.
    - These are resistors with yellow (associated with the number 4 in the 10s column), then violet (associated with the number 7 in the 1s column), then brown (associated with times-10 multiply column) bands and are followed by a gold tolerance band.
    - You can also use 560 Ω resistors with green (number 5 in the 10s column), then blue (number 6 in the 1s column), then brown bands. The 560 Ω resistors come with some Raspberry Pi Starter Packs and are very similar in resistance capacity to the 470 Ω resistor used in this book.

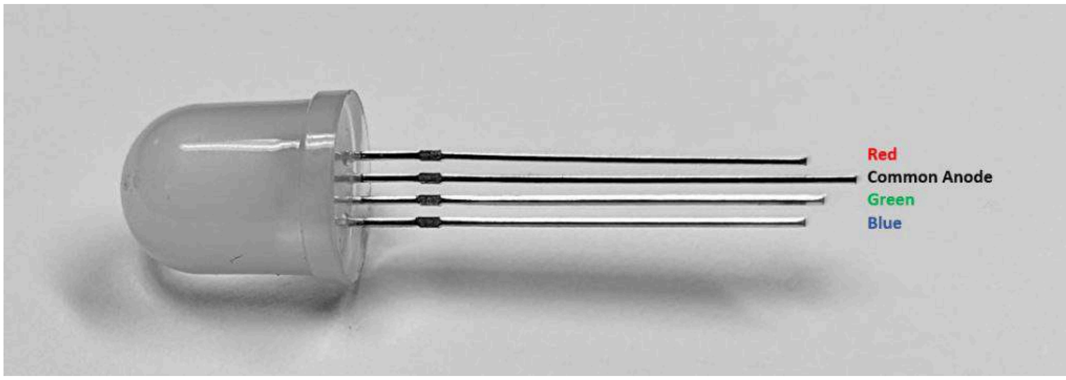
Resistors come with standard 4- or 5-band color coding. This may prove a challenge for those with a color vision impairment.

4 Band Resistor Color Chart				
Color	1st Digit	2nd Digit	Multiplier	Tolerance
Black	0	0	X1	
Brown	1	1	X10	1%
Red	2	2	X100	2%
Orange	3	3	X1000	
Yellow	4	4	X10000	
Green	5	5	X100000	
Blue	6	6	X1000000	
Violet	7	7		
Grey	8	8	Gold X0.1	Gold 5%
White	9	9	Silver X0.01	Silver 10%

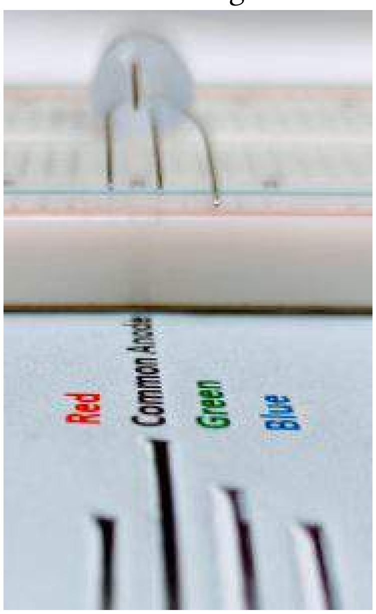
- Attach one leg of the resistor to row E, and the other to row G or thereabouts. One resistor will be in column 19, one in column 21, one in column 23.
    - Each resistor will provide a small amount of resistance to the current, leaving the red, green, or blue legs of the RGB LED on its way to ground and completion of the circuit.
    - As a reminder, rows A, B, C, D, and E have a metal conductor connecting each together within a given column. Likewise, F, G, H, I, and J have a separate metal conductor connecting each together within a given column. A-E is NOT connected to F-J within a given column by default.
    - We will use three 470 or 560 Ω resistors to connect the lower A-E part of the column, to which an anode leg of the RGB LED is connected, to the upper F-J part of the same column. Fritzing highlights such connections in green within their illustrations, although in practice, unused holes remain the same dark color whether or not other associated holes on a conductor now have a conductive wire or electronic component attached.



4. Install the RGB LED on the breadboard
    - Pick up the RGB LED and hold it in your hand matching the order seen in the image below, such that the short red cathode leg is at the top, the long anode leg next, followed by the mid-length green cathode leg, and finally the blue cathode leg at the bottom.



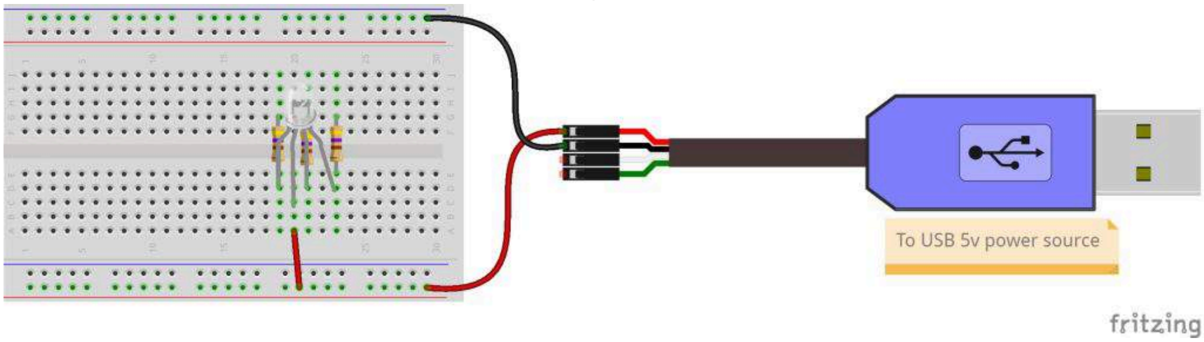
    - Bend the mid-length red cathode leg up slightly, and then return it to the right.



    - Bend the long anode leg away and then return it to the right.
    - Keep the mid-length green cathode leg as is, facing straight to the right.
    - Bend the short blue cathode leg down and then return it to the right.
    - Turn the RGB LED so that the red cathode leg faces left and the common anode leg faces up from the breadboard. Insert the four legs of the RGB LED into the breadboard such that the red cathode leg is in column 19, row D, the anode leg is in column 20, row C, the green cathode leg is in column 21, row D, and the blue anode leg is in column 23, row D.

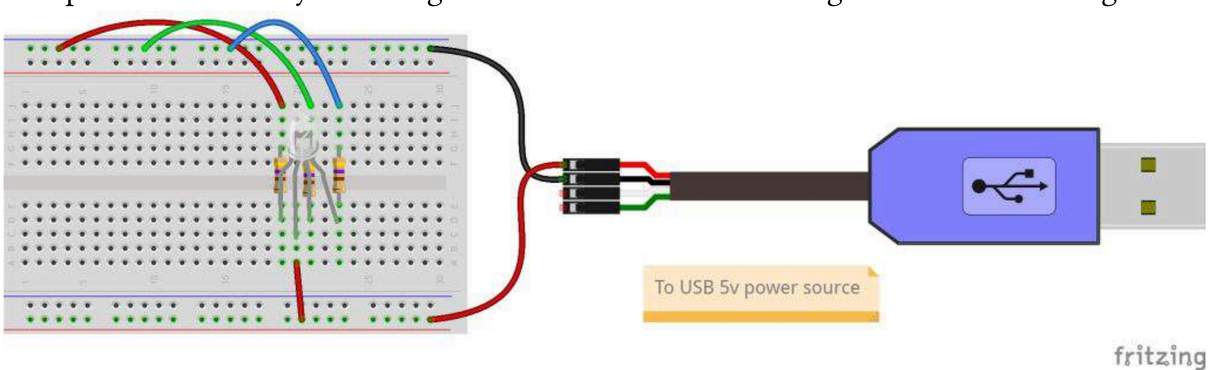
5. Connect the USB side of the USB to TTL cable to a 5-volt USB power source, such as your laptop, or a 120/240-volt power source with power adaptor for a USB connector.

6. Provide 5-volt current to the shared anode leg of the RGB LED.



   - Connect one leg of a male/male jumper wire to the bottom red rail, and the other leg of the jumper wire to column 20 (the same column as the anode leg), row A.
     - NOTE: The circuit is not yet complete, as the 5-volt power now flowing from the USB to TTL into the red rail of the breadboard and then to the anode leg of the RGB and out through the cathode legs and through the three resistors does NOT reach the blue ground rail of the breadboard, and thereby does not return back to a ground source.

7. Complete the circuit by returning current from each cathode leg of the RGB LED to ground.



   - Connect three male/male jumper wires from the blue ground rail at the top to row J of columns 19 (red cathode leg), 21 (green cathode leg), and 23 (blue cathode leg).
     - What happens when only one jumper wire is connected between the resistor and ground at a time?
     - What happens when two jumper wires are connected at a time? All three?
     - How many different LED colors can be achieved with these three base LED colors?
     - Does the brightness of the LED change depending on how many circuits are plugged into a USB hub? Into a USB 1, USB 2, or a USB 3 port on a laptop? Into another USB 5-volt power source?

Key Takeaways

In this exercise, we've entered into a first apprenticeship crafting a **rapid prototyping platforming**, using a USB to TTL cable as a source for power and ground, and using a breadboard as the physical prototyping base tool. To do this, we brought together several **conductor** wires, like the male-to-male **jumper wires** made of individual wires. To test this platform, we brought together two electronic components, a **resistor** and a RGB **LED**, to provide some literal light on the platform.

In addition to these technical skills, we sought to bring in the development of several socio-emotional skills, including the ability to communicate and collaborate with others using a "see once, do once, teach once" strategy using pair/triplet programming, and potentially began exploring how failure serves as an essential step in project development—that is, a fail-forward mindset.

It's also helpful to think about our daily use technologies as **sociotechnical** devices that use a specific form of thinking called deductive reasoning. The bigger picture is broken down into the smallest possible parts needed to best succeed with the least amount of work. At the initial design phase, the questions focus on doing the most with the least number of components. On the other side of the coin, the instructions for others to carry out their own building of the electronic device need to also be written using deductive reasoning. Personally, I've often found I don't read the instructions if I can just look at some pictures or watch a video and then work through things myself. But in the electronics realm, this can bring us to "catastrophic" failures in which the components are no longer usable and new ones need to be purchased. Sometimes packages include extra parts in expectation of this happening. But other times, such failure can have unacceptable threats. Consider pilots who have flown thousands of flights, but still go through the reading of a checklist from beginning to end.

While failure is common and serves as an essential step in project development, when might it be helpful to add "read once" as a precursor to "see once, do once, teach once"?

There are many ways to design and build electronic artifacts. This can be done as individuals working to serve personal interests. This can be done as individuals working as experts to serve others broadly considered. This can be done as individuals working as experts to serve others more strategically. It can be done in groups randomly or strategically partnered to serve inwards or to serve outwards. It can be done as a community of practice in ways that include the contributions of a range of stakeholders. And it can be done in so many other ways as well.

- How did you work to do this first hands-on exercise of the book with electronics?
- How did I and others work to develop this exercise? How did it influence your work with electronics? Who else influenced your work?
- What makes an LED and a resistor electronic components? What makes wires, the Cobbler, and the breadboard conductors parts which *support* electronic components?

Troubleshooting Essentials

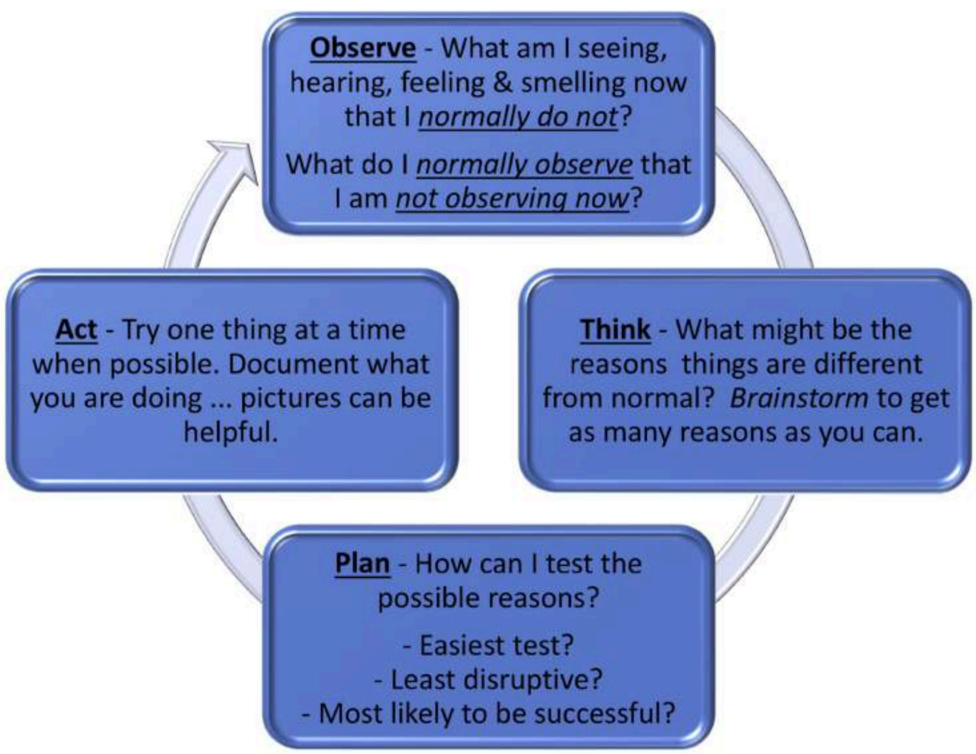
Collective leadership and community inquiry using groups of two or three people to collaboratively test out hands-on work brings together the expertise of multiple individuals.. Failure is common, and generally remains the norm, from iteration to iteration for an extended period.

To positively fail forward, our iterations are an opportunity for us to discover how our inspirations and ideations aren't aligning with our past and current iteration prototype. Reflecting on our actions individually and as part of community inquiry is central to that work of discovery. Said another way, iterations help us see how one part, much, or all of our working idea is falling short of good enough. This helps inspire better ideations and sometimes even new inspirations, ultimately leading to better iterations — at least for us within the given testing context.

Let me say this again: Trying and failing is not optional. It is the norm. The general alternative to failing forward is failing to improve.

Accepting failure as a normal, active, ongoing part of what we do leads us to failing forward in ways that facilitate growth and betterment of ourselves, and when done across functional diversities and community cultural wealth, the betterment of others as well. In so doing, we foster a high performing community of practice.

So let's assume the LED did not light up in the last exercise, or that it burned up in the last exercise, or that it won't light up in the next exercise. Why not? What happened?



Humans and other more-than-human persons around us have a rich range of analog senses, such as sight, hearing, feeling, and smelling. These are a wonderful resource in the troubleshooting process. These analog senses provide a continuously variable physical quantity, such as the visible wavelengths of light. This stands in contrast to the digits 0 and 1 that typically represent the values of a physical quality provided by digital measurements.

Bringing our observations through those analog, continuously variable, senses into a process of critical thinking helps us create a plan for testing out a range of possibilities. Enacting that plan one step at a time helps us to continue the cycle of observation, thinking, planning, and acting. This is the heart and soul of inquiry-based learning as we collaborate in pairs and small groups through dialogue.

Throughout the remainder of the exercises in this book, and as you work through your own design activities putting your learning into practice, be sure to make extensive, preferably documented, use of these troubleshooting basics. It's tempting to try it a few times and move on because troubleshooting the current exercises begins to become second nature. These experiences serve as a learning opportunity, but at the risk of having the learning become hidden knowledge that fails to fully help you as you encounter future problems.

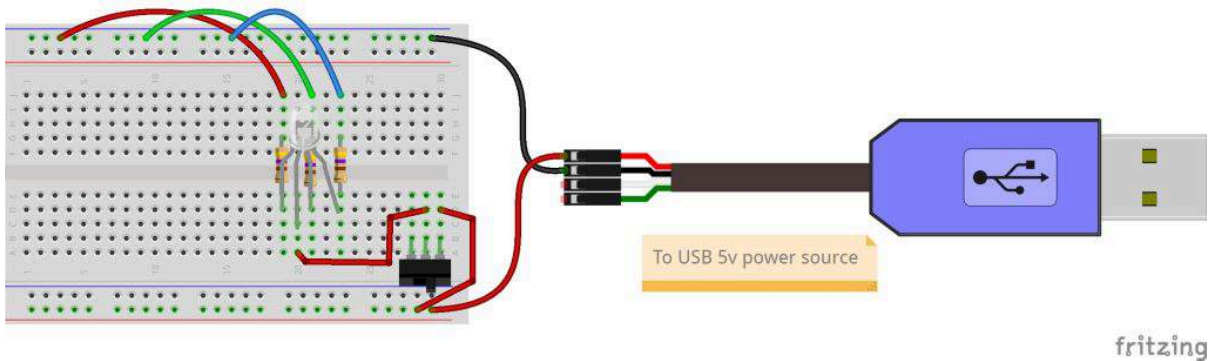
Indeed, this is a reason why this book has been written using the Creative Commons Attribution-ShareAlike license. We hope you will add in annotations, page notes, and remixes for yourself and others to use as sources for their own design thinking projects. We hope the resulting inspirations, ideations, iterations, and failures support an advancing betterment of ourselves, those in our community of practice, and those we are serving through our creative works.

Exercise: Adding a Two-position Switch to Our Circuit

In this activity, we will add a slide switch into our circuit. It is called a “switch” because it changes the flow of current depending on the state of the switch. This switch creates a change in state when a physical action happens—in this case, when a slider is moved to the left or to the right. This type of single pole, double throw (SPDT) switch is a great electronic component to turn something on and off, or to select between this or that. The center pin provides the rest of the circuit with the results of the user's choice. The left and right throw pins provide the user with one of two options. We'll be using this switch in a variety of ways moving forward. For now, we'll connect the single center pole to the RGB LED, and the right pole to the lower red rail providing a 5-volt power source. The left pole will be left open. The user then chooses between providing the RGB LED with 5 volts of power (on state) or no power (off state). When the slide is to the left, we have a ***closed circuit*** because the switch creates a state in which power completes circle over to ground is complete. When the slide is to the right, we have an ***open circuit*** because there is no starting power needed to begin the flow of current required for a completed circuit.

Steps

1. Attach a slide switch to columns 28, 29, and 30 of row B.
2. Connect one end of a male/male jumper wire to column 20, row A, and the other end to column 29, row D. This is the common leg of the slide switch.
3. Connect a second male/male jumper wire to the lower red rail. The other end of this jumper wire gets connected to column 30, row D. This is the right selection leg of the slide switch.
4. Finally, make sure the TTL to USB is plugged into a power source, and then slide the switch left and right to turn the RGB LEDs on and off.



Key Takeaways

Designing and building electronic devices needs to include consideration of the mundane—for instance, turning something on or off. In this second exercise, we've further explored how conductive materials like a breadboard and jumper wires can be combined with electronic components like a slide switch to provide the users of an electronic device with such on/off choice. But while mundane, there are still key aspects shaping the layout.

- Why might it be that I put the switch where I did?
- Does the placement work well for you when you work to move the slide left or right?
- Is there someplace else on the breadboard where you might have placed the slide switch?

If possible, jot down your answers to these reflective question prompts so that you can revisit the questions and your answers later. Doing so is an essential part of action-reflection cycles that help bring a new lens of understanding on sociotechnical artifacts such as the electronic circuits we'll be building throughout the book.

In writing the exercises of this book, I shaped the design of the device for a range of functional, socio-emotional, and cognitive reasons. As you build it, you have the agency to further shape the design as you read through and implement the steps of the current activity. Taking time to reflect on actions and to make a recording of those reflections helps to keep an ongoing account of your socio-technical ***praxis*** as you exercise that agency to innovate-in-use, or stick with the instructions as written, or discover a mistake in the instructions, or make a mistake following those instructions. Reflections like this go on to provide opportunities not only for you, but for others in a community of inquiry to go beyond finding and using information. It is in action-reflection that we move collectively from holding information to advancing a better understanding of the information and of the people and contexts shaping that information.

Voltage, Resistance, and LED Brightness

Earlier in this chapter, we covered a bit on **Ohm's law**, which provides a basic equation I=V/R stating that **current** (I) is equal to **voltage** (V) divided by **resistance** (R). Voltage is like the pressure of water in a hose. Current, measured in amperage, or amps for short, is like the flow of water through a hose. The thickness of the hose impacts the resistance of the flow of water through it, which is like the ohms of resistance in a circuit.

Let's take the water analogy one step further. Imagine the water is now used to turn a wheel, which is used to power a small mill making gristmill, which is used to turn cereal grain into flour. You can increase the power generated by the water wheel by either increasing the pressure (voltage) of the water or increasing the flow (amps) of the water. In electrical terms, this power (P) is called wattage, or **watts**.

> The formula for this is Volts x Amps = Watts.

Some may be familiar with, and perhaps make regular use of, lighting in which you can control the brightness of the light. Perhaps you have a lamp with a three-way light bulb providing three different watts of power, depending on the position of a rotary switch. Each turn of the light switch steps you up to the next wattage before returning to off. The lightbulb may state that it is a 60/120/180-watt bulb. If you are in the United States, you likely plug this lamp into a 120-volt outlet. We now are using one of three amps of current

- 60 watts / 120 volts = 0.5 amps
- 120 watts / 120 volts = 1.0 amps
- 180 watts / 120 volts = 1.5 amps

If our 120-volt outlet is listed as being a 15-amp circuit, we should keep in mind that if all these lamps were running at full wattage, we should have a maximum of 15 amps / 1.5 amps = 10 lights at 180 watts.

Let's look back at our RGB LED. It is rated for a maximum of 5 volts. Adafruit has a USB 2.0 Powered Hub with 7 ports, each with 5 volts at 2 amps. If we were using this for our RGB LED, the common anode would be receiving

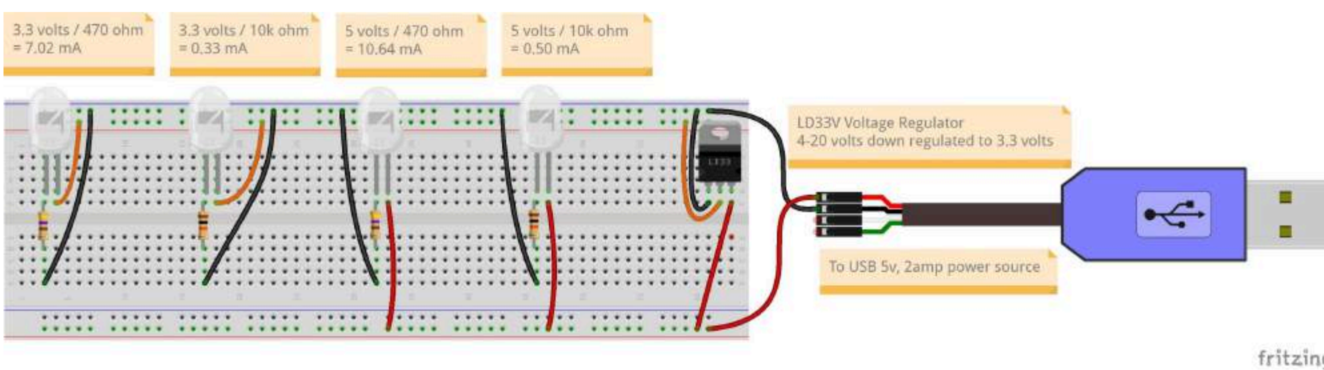
> 5 volts x 2 amps = 10 watts of power

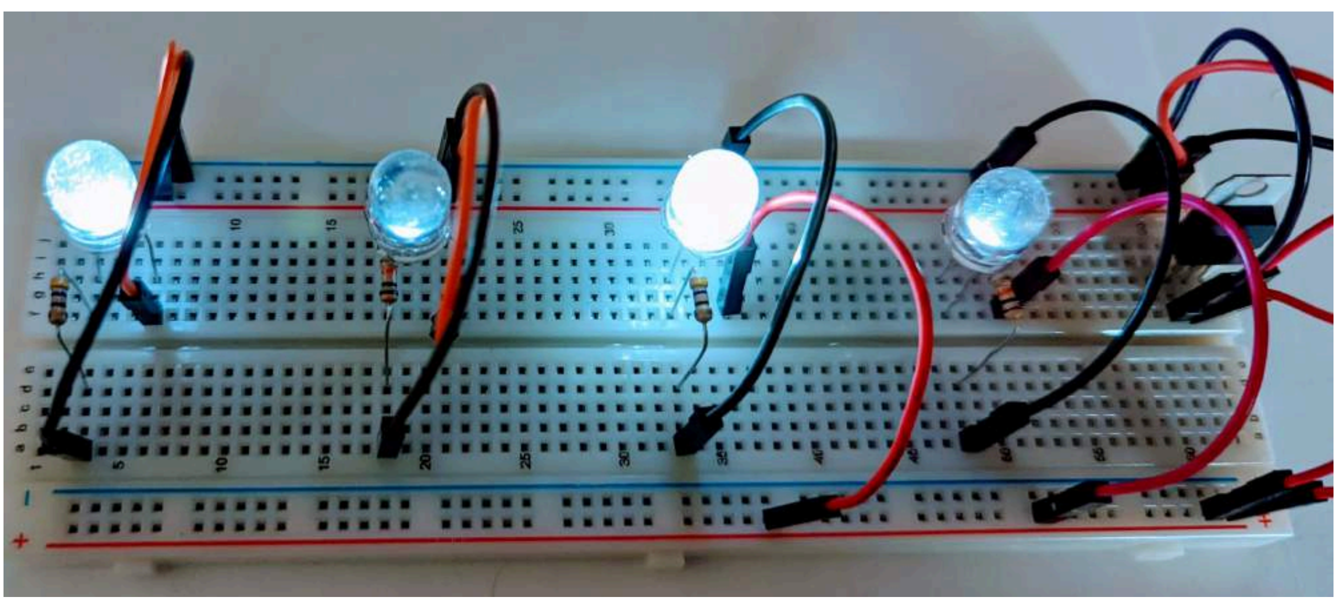
But given we have a 470 Ω resistor wired to each cathode leg to assure it doesn't get so much current that it burns out, each of the parallel circuits is reduced a bit, as we see in Ohm's law (I=V/R):

> 5 volts / 470 ohms = .01 amps
> 5 volts x .01 amps = .05 watts

To demonstrate, four parallel LED circuits have been created, each receiving different amps of power, thereby providing different watts of light. The breadboard circuits use the following components:

- A USB to TTL connected to a USB 2 powered hub providing 5 volts at 2 amps. This is attached to a full-sized breadboard as before.
- A voltage regulator, an electronic component that brings the voltage of a power supply down to a level compatible with the other electronic components. Here a voltage regulator is being used that can work with 4-20 volts of DC power as the source (in our case, 5 volts at 2 amps), which it then regulates down to a stable 3.3 volts DC power.
- Two 470 Ω and two 10,000 Ω resistors
- Four 10mm bright white LEDs





From left to right, the watts of power of each LED in the image above are as follows:

3.3 volts / 470 ohms = 0.0070 amps = 7.0 mA	3.3 volts x 0.0070 amps = 0.0231 watts = 23.1 mW
3.3 volts / 10,000 ohms = 0.0003 amps = 0.3 mA	3.3 volts x 0.0003 amps = 0.0011 watts = 1.1 mW
5 volts / 470 ohms = 0.0106 amps = 10.6 mA	5 volts x 0.0106 amps = 0.0532 watts = 53.2 mW
5 volts / 10,000 ohms = 0.0005 amps = 0.5 mA	5 volts x 0.0005 amps = 0.0025 watts = 2.5 mW

These are clearly not nearly as bright as a 60-watt light bulb in a house! But as the plastic cases of each of these bright white LEDs are not diffused like the diffused RGB LEDs we've been using, we can see how much more they brighten the view of the breadboard, especially the 1/20th of a watt from the 5V + 470 Ω LED circuit.

Do Something New!

For some, this is your first journey into electronic circuits. You're *Doing Something New* through troubleshooting electronic circuits and failing forward! You are at a "yet-enough" point to head on down to the Wrap Up and Comprehension Check at the bottom of this chapter.

For others, you may already have a lived experience with electronic circuits. This is not something new to you. But if you're at all like me, some of this work has become rote, and the underlying terms, concepts, and principles hidden knowledge. For you, *Doing Something New* may be a work of re-remembering the codes to the electronic wheelhouse. And moving beyond, maybe it's discovering some of the social influences that have shaped the non-neutral aspects of the otherwise strict laws of physics within.

For all, it's likely that at least some aspects of Collective Leadership, Community Inquiry, Action-Reflection, the Information Search Process, and Pair/Triplet Programming are new to you. *Do Something New!* by incorporating one or two of these into your work, this and every session, individually and as part of your Community of Practice work.

Below are two *Do Something New!* exercises putting the above brightness display into practice using your own starting breadboard circuit prototype, this time using the RGB LED. How much do they change the brightness of red, green, and blue diffused LEDs? For those not yet ready to take on the exercises but still wanting to dig a bit more into what's behind the hood, you can also skim one or both activities themselves and then delve more into the Key Takeaways of each to advance from understanding less to understanding more.

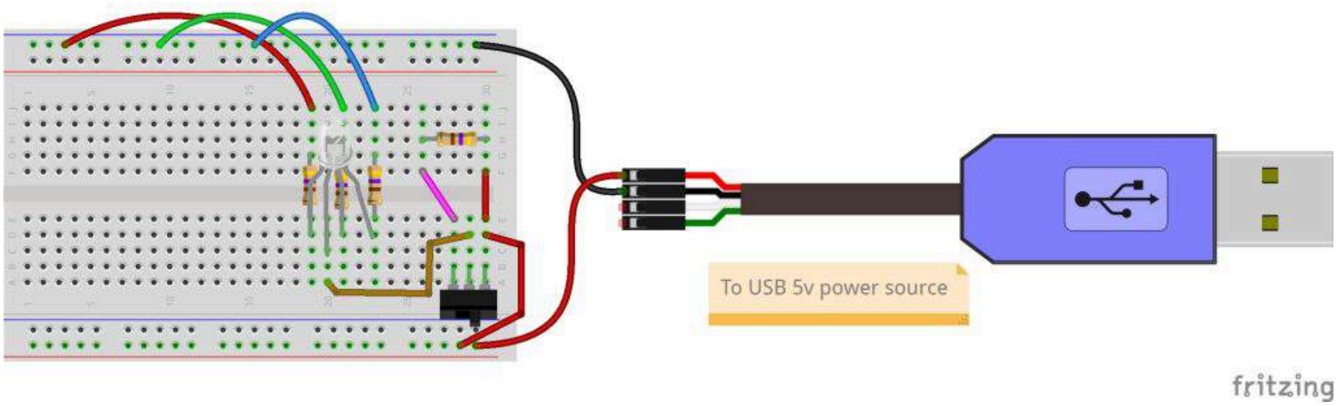
Exercise: Using a Switch to Compare Differences in Resistance of Current Flow

In setting up our parallel red, green, and blue RGB LED circuits, each cathode leg is passed through a 470-ohm resistor on the way to ground. The change in amps remains the same whether the resistor is placed before or after LED within a circuit, in this case 5V/470 Ω = 0.01064 amps (10.64 milliAmps or mA). The switch is currently used to select between "on," in which case 10.64 mA sent to the common anode leg of the RGB LED, or "off," in which case no current is sent to the RGB LED.

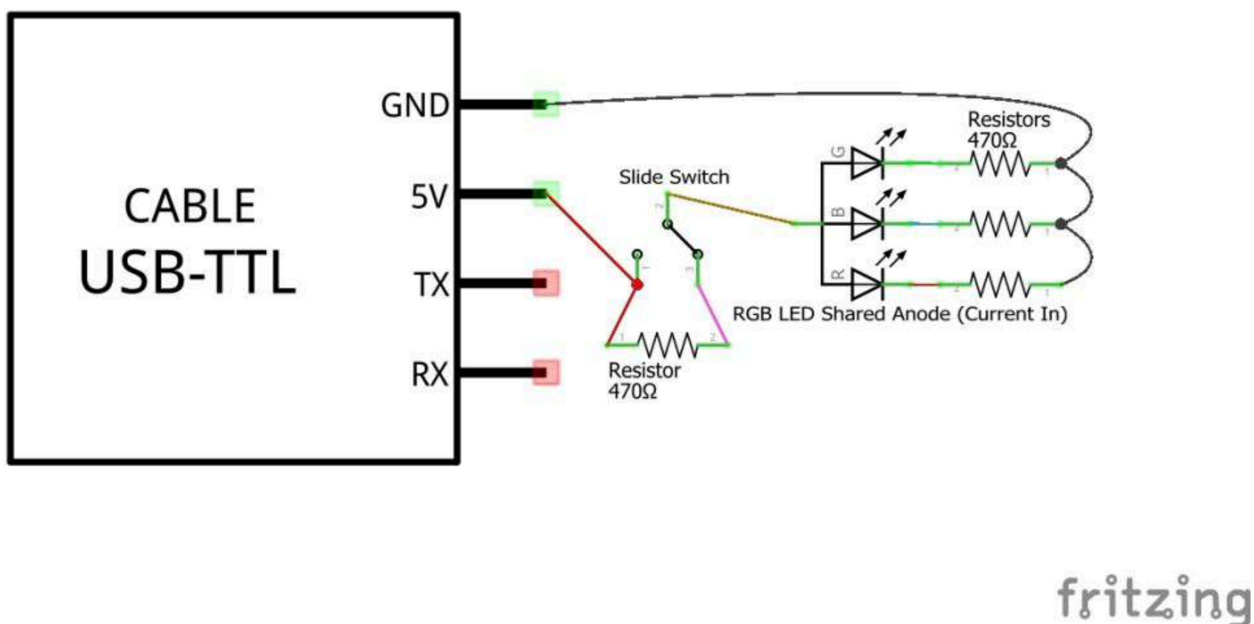
In this *Do Something New!* activity, we'll add a new 470 Ω resistor to the left throw pin of the slide switch. This resistance will be added to each of the three cathode legs so that all three LEDs now pass through 940 ohms of resistance. Using Ohm's law, 5V/940 Ω = 5.32 mA, or half the 10.64 mA sent to the common anode leg when the slide switch is to the right.

Steps

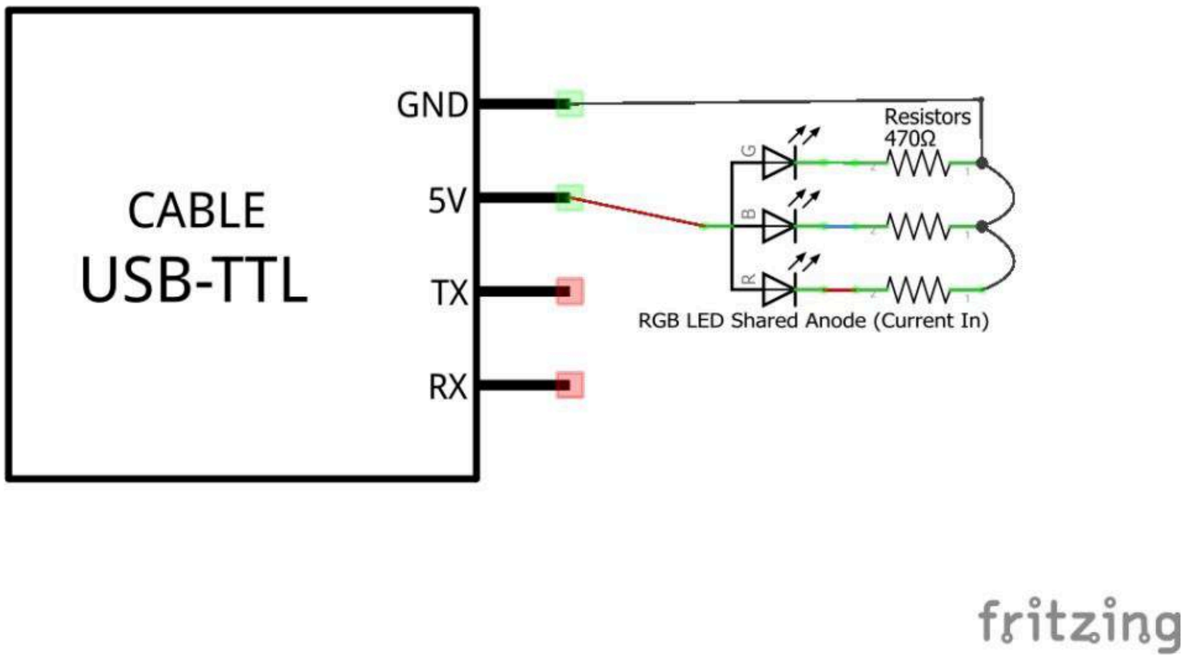
1. Tightly bend down 90 degrees the two legs of a fourth 470 Ω resistor.
2. Insert one leg of the resistor in column 30, row H, and the other in column 26, row H.
3. Connect one end of a male/male jumper wire to column 30, row E, and the other end to column 30, row F. Now current from the red power rail is passed to rows A-E of column 30, the jumper wire attached in step 1, and to rows F-J using a second jumper wire we just installed as part of step 2.
4. Connect one end of another male/male jumper wire to column 26, row F, and the other end to column 28, row E. This is the left selection leg of the slide switch.
5. Finally, make sure the TTL to USB is plugged into a power source. Then slide the switch left to examine the brightness of the RGB LEDs when the current passes through an additional 470 ohms of resistance before entering the shared anode leg of the RGB LED. Compare this to the brightness of the RGB LEDs when current enters directly to the shared anode leg of the RGB LED.
    - This change will impact the red, green, and blue LEDs differently, as each of these light-emitting diodes work within different ranges. Plus, red alone is different than red plus green, which is different from red plus green plus blue. Try different combinations of red, green, and blue circuits connected to ground to fully test the differences.
    - If you can, take pictures of the RGB LED circuit with the switch to the left and to the right so that you can more readily compare the differences between them.



Below, you can see the schematic of our new circuit, which you can compare to the Fritzing diagram above. Looking at the standard switch symbol used in the schematic drawing, we see that what passes from leg 2 over to the common anode leg of the RGB LED is determined by the line that rocks between legs 1 and 3 of the switch. In designing this circuit, I chose to connect the 5-volt current directly to leg 1, and to run the 5-volt current through a 470 Ω resistor before connecting to leg 3.



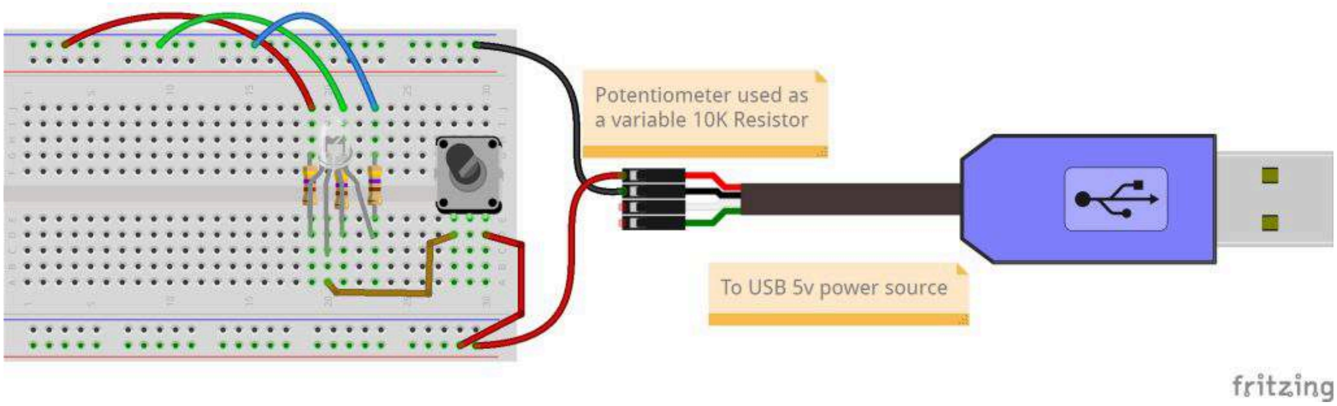
Let's further compare our switched resistance schematic above with the schematic from the first exercise, in which the 5-volt current was connected directly to the common anode leg of the RGB LED. Note that while initially 5 volts flowed directly from the USB to TTL cable to the common anode leg of the RGB LED, it now first flows through a slide switch, in which it either passes directly to the common anode leg or passes first through an additional 470 Ω resistor before reaching the common anode leg.

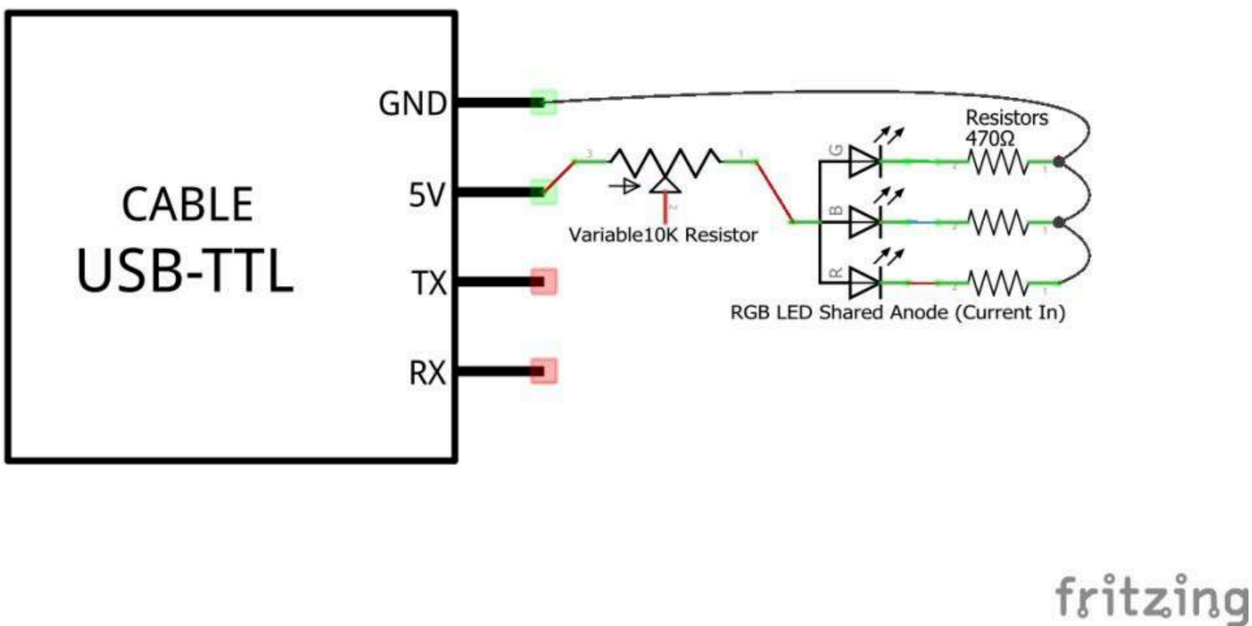


Key Takeaways

Resistance is a common way to control the brightness of lights. In our breadboard prototype, resistance changes the flow of current from source into the common anode leg of the RGB LED, and out to ground by way of the red, green, and blue cathode legs. We have already included 470-ohm resistors in our red, green, and blue circuits, to assure that the flow of current when connected to a 9-volt battery does not exceed the amount of current listed within the current rating of the RGB LED we're using. (This is a general introduction—for more on current ratings of LEDs and how to use this to select power sources and a resistor for a circuit, I highly recommend "All About LEDs" by Tyler Cooper.) While it is essential that you include at least the minimum required resistors within your circuit, resistors are often used to intentionally restrict the flow of current further than required, as a way to control the brightness of light provided.

The flow of current can be controlled using many different types of switches. You might have lights at home that can be dimmed. Your mobile phone might let you use a slide to control its brightness. Or you might be able to enable or disable "nighttime mode" for your clock. When a circuit is designed to provide a steady gradation in resistance, the type of electronic switch commonly used is called a variable resistance potentiometer. Potentiometers slowly change the level of resistance provided from a defined minimum (e.g., 470 ohm) to a defined maximum (e.g., 10K ohm). Below are Fritzing images of a 10K potentiometer that could be used for our testing.





For this exercise, we used an electronic single pole double throw (SPDT) slide switch. The single common pole in our exercise is used to connect to the common anode leg of the RGB LED. The two throw legs are then connected directly to a 5-volt power source and to a 5-volt current that is first passed through a 470-ohm resistor. Selecting one of the two throw legs of the switch chooses whether to provide current directly from the rail or via a resistor. The slide switch was selected as a simple means for comparing two different conditions—this one or that one. In the above, we used it first to choose between off and on. Then we chose between a direct 5-volt connection and a 5-volt connection with 470-ohm resistance. And next we'll use it to choose between different voltages.

- How does the brightness of the light from the RGB LED—either together or in different combinations or red, green, and blue—change when the slide switch is moved left or right? Why, and how much?
- What are some reasons you can think of that I decided to design the selection and use of the LEDs, resistors, switches, and wires in the way that I did to set up your execution of this activity?

- What are some ways we might have done it differently, and in so doing may have created a better way for you to complete this activity?

Exercise: Using a Switch to Compare Differences in Voltage

For this second *Do Something New!* activity, let's use the two-way switch to select between a 5-volt and a 3.3-volt current, to consider how a difference in the voltage of current compares to a difference in the resistance—that is, the flow of current. Today's USB standard works with 5-volt direct current (DC). Most homes and offices have wall outlets that use 120V or 240V alternating current (AC). A power converter is used to switch from alternating to direct current. 5-volt DC has proved a solid choice for many electronic circuits, but others, like fans and motors, often make use of the widely used 12-volt direct current. And some electronics have worked to reduce power usage by going the other direction, decreasing current to 3.3 volts. Unlike the AC voltage standards used within different countries, the 3.3V, 5V, and 12V DC are not standards, just widely used voltages. When selecting electronic components, it's important to identify the range of voltages which can be used for that component and to plan accordingly.

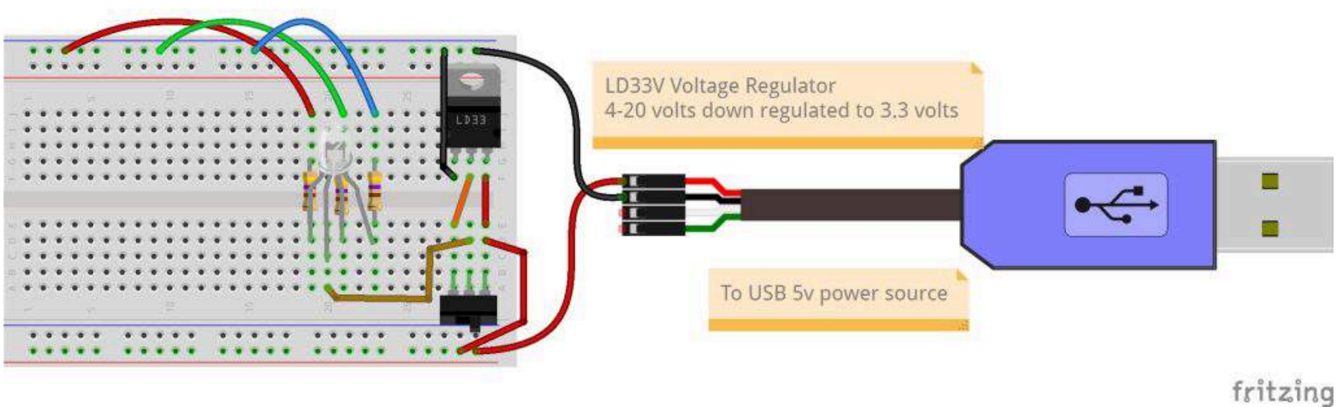
The RGB LED we're using has a maximum forward current rating of 20 milliamps (mA). Looking back at the start of this chapter, we saw a demonstration of a circuit built using a 9-volt battery and a 470 ohm resistor. Together, 9 volts at 470 ohm is 19 mA, just under the 20-mA maximum current rating of many of the LEDs we use today. Using a 5-volt power source, we could have safely gone down to a 100 ohm resistor, but stayed with the 470 ohm resistor in case a 9-volt battery was used instead of a 5-volt USB power source. Together, 5 volts at 470 ohm is 10.6 mA. If we go the other direction, down to 3.3 volts at 470 ohm, we are now at 7 mA, or 1/3rd of the amperage used with 9-volt power and 2/3rd the amperage used with the 5-volt power.

Keeping resistance constant, how does decreasing current usage by decreasing power impact the brightness of an LED? Let's find out by using a voltage regulator. A voltage regulator is useful when you have a possible range of voltages for a power source, but are needing a stable voltage for your electronics. Look at your laptop power adaptor, and it will likely say it can work within the range of 110V to 240V AC input. It will also list a stable DC output voltage or voltages to be provided regardless of the input voltage.

In our case, we'll use a voltage regulator that can work with 4-20 volts of DC power as the source, which it then regulates down to a stable 3.3 volts of DC power. To keep this simple, we'll replace the 10k ohm resistor with the voltage regulator. But this could be connected to its own switch that then feeds into the slide switch used for the resistor to do a 2×2 comparison of resistance and voltage on LED brightness. And as an important side note, the GPIO pins of a Raspberry Pi we'll be using in session four of the Orange Unit provide both pins with 5-volt power, which they receive from the micro-USB or USB-C plugin and pins with voltage regulated 3.3-volt power, allowing us to bring both to the breadboard for ongoing use. This *Do Something New!* provides a first glimpse at some possibilities for if we were connected to the Raspberry Pi GPIO instead of the USB to TTL cable for breadboard power and ground.

Steps

1. Return to the on/off switch circuit for the RGB LED
    - Remove the 470 Ω resistor and associated jumper wires.
2. Connect an LD33V voltage regulator such that the LD33V label is facing the switch.
3. Use jumper wires to provide the ground and 5V power sources to the regulator, and connect the regulator output to the left throw leg of the switch.
4. Finally, make sure the TTL to USB is plugged into a power source. Then slide the switch left to examine the brightness of the RGB LEDs when 3.3 volts of power are provided, compared to when the switch is to the right and 5 volts of power are provided.
    - If you can, take pictures of the RGB LED circuit with the switch to the left and to the right so that you can more readily compare the differences between these.



Key Takeaways

This layout of the breadboard now provides an opportunity to compare brightness depending on voltage. For this exercise, we incorporated another commonly used electronic component, known as a voltage regulator, to take the 5 volts of power provided through the USB to TTL cable and regulate it down to a stable 3.3 volts of DC power. As we've now seen in these two *Do Something New!* activities, there are two ways to adjust the brightness of an LED: by changing resistance and by changing voltage.

1. Of these different layouts, how does voltage change brightness when all else is equal? How does this compare to the brightness when the 5-volt power remains consistent and resistance is changed, as was done in the second exercise?
2. What are some key takeaways regarding the brightness of LEDs, given the combined fac-

tors of different power sources and resistance levels?

Wrap Up

This is just a first glimpse at electronic circuits and some ways we can work hands-on in design and prototyping of new circuits. We've learned a few basics of creating a completed circuit and how to troubleshoot when the circuit doesn't seem be working. We've learned out to use components to switch the circuit from complete (on) to incomplete (off), and how to switch between levels of resistance and voltage to change the brightness of an LED. Along the way, we've started discovering there are many choices to be made. For instance, both resistance and voltage can be used to control the brightness of a light. But while batteries and AC outlets **generate** power for use by electronics, resistors and LEDs make **use** as Tyler Cooper notes near the end of "All About LEDs." Use the lowest voltage possible to get a sufficiently bright LED. But beyond that, switching between voltages can become laborious if you need to regularly change brightness. That's when resistors especially come into play. On the other hand, too often we continue make use of resistors and voltage regulators to stay with the same large power generators we've always used when more environmentally friendly low power sources could be used if this is kept in mind during design.

And so there have also been some question prompts urging you to question the reasoning behind the design of the circuits themselves. Accomplishing an electronics goal can be achieved using many different components and circuit designs. Helping others to build, make use of, and further innovate electronics is yet another place allowing for extensive variations. It's important we keep front and center these extended question prompts if we are to rapidly shift from a thing- to a person-orientation as we work to demystify technology.

There's much more to come as we proceed through the Orange Unit and beyond. For now, take a few minutes to do a quick comprehension check on some of the key circuit terms and concepts we've used so far.

Comprehension Check

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with answers, see the online version.]

1. Which of the following is a proper circuit: a complete and closed path around which an electric current can flow? Choose all that apply.
    - A. red wire -> resistor -> 3mm white LED -> black wire
    - B. 5 volts -> resistor -> 3mm white LED -> ground
    - C. 3.3 volts -> 3mm white LED -> ground
    - D. resistor -> 3mm white LED

2. We created three different circuits combining different resistors and power sources, such that the associated white LED in these circuits had a different level of brightness. Sort the power sources and resistor values as they would light a 3mm white LED, listing them from dimmest to brightest.
    - 3.3 volts, 10K Ω resistor
    - 3.3 volts, 560 Ω resistor
    - 5 volts, 560 Ω resistor
    - 5 volts, 10K Ω resistor

2A: Critical Social + Technical Perspective

Background Knowledge Probe

- In what ways have you been shaped by technology? When, where, how, and why?
- In what ways have you helped to shape technology? When, where, how, and why?
- In what ways have the people closest to you been shaped, positively and negatively, by this shaping of technology? When, where, how, and why?

Digital Technology = Social + Technical

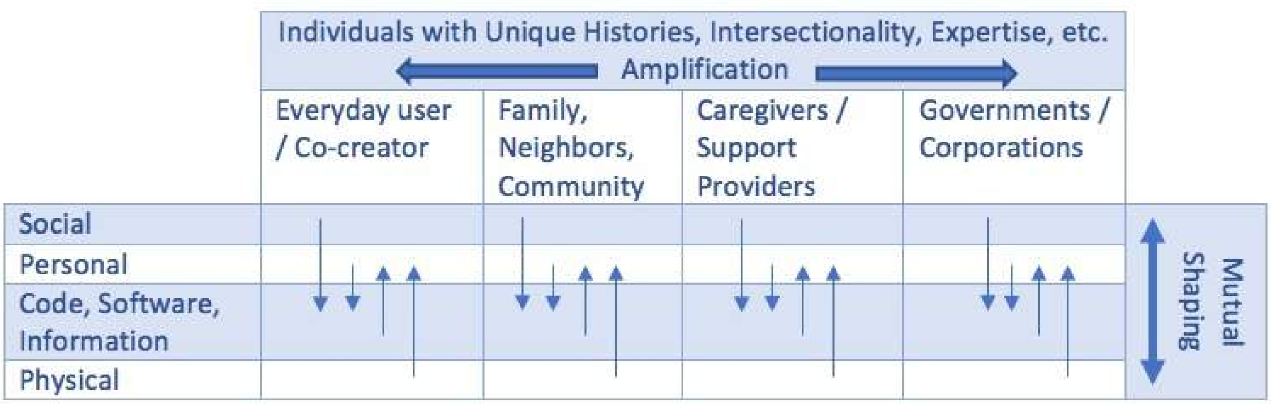
There's a growing realization that our digital technologies are actually a seamless, indivisible combination of artifacts, people, organizations, policies, economics, histories, cultures, and knowledge – that they are sociotechnical products. And over the last several decades, this has gone further to acknowledge a mutual shaping of gender and technology.[1] As a result, the reliable, anticipatable relationship between user input and system output is complicated by the continuous evolution of experience, knowledge, history, culture, economics, and policies of users and society. The social characteristics cannot be readily planned for or controlled, especially as sociotechnical products are continuously co-created by everyday users to fit the ever-changing contexts and knowledge of users and their communities.[2] As such, a reductionist and positivist approach and associated deductive reasoning common in much of basic research must be complemented by, and live in continuous tension with, an interpretive approach and associated inductive reasoning of the social scientists, and even more importantly, with the community cultural wealth, expertise, and knowledges of community members.

1. Wajcman, Judy, "Feminist Theories of Technology," *Cambridge Journal of Economics* 34, no. 1 (January 1, 2010): 143–52. https://doi.org/10.4018/978-1-60566-264-0.ch045.
2. Bruce, Bertram, Andee Rubin, and Junghyun An, "Situated Evaluation of Socio-Technical Systems." In *Handbook of Research on Socio-Technical Design and Social Networking Systems*, edited by Brian Whitworth and Aldo de Moor, Information Science Reference (Hershey, PA: IGI Global, 2009), 2: 685–98. https://doi.org/10.4018/978-1-60566-264-0.

Innovations-in-Use

> [A]n innovation is not an object that can be packed inside a box, but rather a set of practices that emerges from the social setting of its use. Thus, in a sense, the user does not accept or reject an innovation but instead creates it through action in the world.
>
> Bertram Bruce, Andee Rubin, and Junghyun An, "Situated Evaluation of Socio-Technical Systems"

In his thought-provoking 2015 book, *Geek Heresy*, Kentaro Toyama notes that technology does not itself have agency to transform us and the world around us. Rather, humans use technology to amplify our individual and group forces to transform our world. This is primarily framed from the perspective of the everyday user. However, if we bring criticality together with an interpretive sociotechnical perspective, we also need to reflect on the human forces amplified at each level of a product's life cycle.[34] We need to ask: How are the forces of engineers, computer scientists, and garage inventors; of the president, CEO, board, and shareholders of corporations; of the marketers and salespeople; of government legislators and administrators; of educators and social service agencies; of individuals and groups as co-creators; and of the many others in the product life cycle, amplified in ways that are consistent with and counter to the values and goals of those using the sociotechnical product at each stage of the artifact's life cycle?

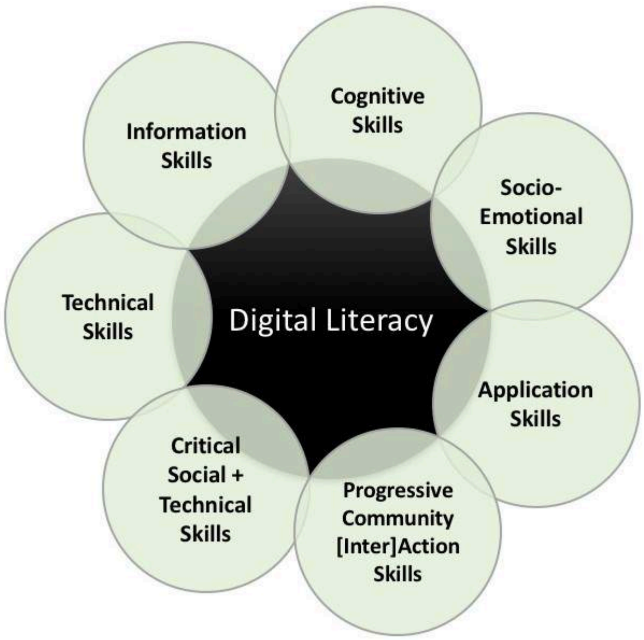


Each stakeholder, from the everyday user/co-creator to the government and corporate official, brings their unique history, intersectionality, expertise, etc., to bear, amplifying the influencing social and technical structures of the digital technologies around us. The personal and social aspects thereby shape the code, software, and information — as well as the physical electronics on which the code, soft-

3. Fischer, Gerhard, and Thomas Herrmann, "Socio-Technical Systems: A Meta-Design Perspective," *International Journal for Sociotechnology and Knowledge Development* 3, no. 1 (2011): 1–33. https://doi.org/10.4018/jskd.2011010101.
4. Rhinesmith, Colin, and Martin Wolske, "Community Informatics Studio: A Conceptual Framework." In *CIRN Community Informatics Research Network Conference:* "Challenges and Solutions, "Prato, Italy: Centre for Community and Social Informatics, Faculty of IT, Monash University, 2014. https://shareok.org/handle/11244/13631.

ware, and information reside. At the same time, the code, software, information, and physical structure shape the individuals and social structures making use of those technologies. Given this amplification and mutual shaping of the sociotechnical artifacts and the overarching systems, what literacies are required for us to move from passive consumers to active innovators-in-use of digital technologies?

Community Inquiry as the Basis for Digital Literacy Radically Reconsidered



Literacy is a set of competencies and knowledge within a certain domain. Digital literacy, then, is literacy within the realm of digital information and communication technologies. Summarizing a variety of definitions regarding digital literacy and related computational thinking, this set of competencies includes:

- **Technical skills:** The ability to appropriately select and effectively use a range of technologies.
- **Information skills:** The ability to seek, evaluate, interpret, and apply relevant and trustworthy information across multiple media.
- **Cognitive skills:** The ability to logically analyze and organize problems in ways that allow use of digital and other tools to help solve them and to generalize new processes to other problems.
- **Socio-emotional skills:** The ability to communicate and collaborate with others, along with the personal confidence, persistence, and tolerance needed to tackle complex, ambiguous, and open-ended problems.
- **Application skills:** The ability to integrate the above skills into our everyday experiences in order to advance our professional, personal, and civic interests and responsibilities.

If we are to assure active participation in civic society and meaningful, diverse and inclusive contributions to vibrant, informed, and engaged community, it is essential to add the social justice-oriented progressive community [inter]action and critical social + technical skills to these other five points.

- The **critical social + technical skills** are brought forward as part of person- and community-centered deliberative dialogue processes using critical research paradigms, pedagogies, and abductive reasoning in combination with each of the five commonly considered digital literacy skills listed above.

- **Progressive community [inter]action skills** are simultaneously used in combination with the starting five digital literacy skills by advancing our ability to work together in communities of inquiry that make continued use of collective leadership, functional diversity, members' capability sets and freedom of choice to better achieve valued beings and doings, community cultural wealth, and care work and ethics.

Whether as part of preprofessional and professional development of library and information management workers, or as part of programming offered by these professionals, a radical reconsideration of digital literacy is essential if we are to effectively use sociotechnical products to amplify human forces which advance human and community development. This social-forward approach to digital literacy training doesn't negate learning about the nuts and bolts of the hardware and software. It instead works to advance our critical social + technical skills so as to situate such learning within the individual and group goals and values.

A Person-Centered Approach to Digital Literacy

Strategies

- Meet with stakeholders to understand what creative works participants would like to do and what digital technologies might be needed to do them.
- Incorporate digital literacy training as infill into projects as needed.
- Include exercises exploring all dimensions of socio-technical artifacts.
- Intersperse discussion and critical reflection with hands-on activities to bring to the fore participants' human and social expertise to complement hardware and software skills development.

Goals

- Achieve the creative works as determined by participants during stakeholder meetings.
- Advance participants' capability set (the increased existence, sense, exercise, and achievement of choice).
- Increase participant's ability to select and appropriate strategies/technologies that align with their values and goals.
- Counter the dehumanizing impacts of digital technologies.
- Work towards the essential outcome of more resilient, inclusive, and just communities.

Values

Technology is shaped by and shapes society. From this starting point, we posit that:

- The social, cultural, historical, economic, and political values and practices of stakeholders at each point in an artifact's life cycle tend to become embedded within that artifact.
- There are exclusionary social structures, some of which we actively — even if unintentionally — reinforce through our choices and actions regarding technology creation and use.
- Digital literacy without a critical and sociotechnical perspective is at risk of fostering magical thinking and technological utopianism.
- A liberative approach to technology requires lived and academic expertise within multiple domains, including hardware, software, human, community, social, cultural, historical, political, and economic.

Key Takeaways

- Those of us with technical expertise may enter into an engagement as digital literacy instructors, but we also need to be willing learners if we are to understand the exclusionary social structures embedded within sociotechnical artifacts and thereby champion justice.
- The digitally excluded and participants from the margins of society may enter into digital literacy training as learners, but also bring essential sociotechnical expertise and teaching to communities of inquiry.
- Difference is not just a nicety, but an essential resource in building more resilient, inclusive, and just communities.

Advancing digital inclusion, equity, and literacy often presents unique information needs and potentially burdensome financial, human, and infrastructure challenges. Partnerships among stakeholders, including those from the information sciences and information management, often serve at the front lines to mitigate these challenges. But how do we best help organizations build the capacity to meet the information needs of communities? In what ways can we develop 21st century skills to foster individual, social, and economic development? How can we integrate standards and frameworks such as collective leadership, community inquiry, action-reflection cycles, information seeking, studio-based learning and design thinking, collaborative programming and computational thinking, and creativity, play, and tinkering into the partnership process?

Lesson Plan

While the focus of the technical chapter, "Electronic Components in Series," is on how we can reuse the LEDs and resistors in combination with momentary switch push buttons, it is important for us to also continually reflect on the social aspects of these sociotechnical artifacts. Our educational and professional systems are built so as to often separate these two aspects. How can we work together in this session and in each of the following sessions to identify ways in which mutual shaping and amplification of sociotechnical artifacts has happened or is happening with this moment? How are we working to further the gendering of the artifacts? How are we working to problematize these artifacts so as to advance a more person-centered approach?

Essential Resources:

- Toyama, Kentaro. "TEDxTokyo – Kentaro Toyama." YouTube, May 10, 2010. https://www.youtube.com/watch?v=cxutDM2r534.[5]
- Oxford Internet Institute. "OII Awards 2018: In Conversation with Judy Wajcman." YouTube, November 26, 2019. https://youtu.be/pf7-93ioYsE.

Additional Resources:

- Bruce, Bertram C., Andee Rubin, and Junghyun An. "Situated Evaluation of Socio-Technical Systems." In *Handbook of Research on Socio-Technical Design and Social Networking Systems*, edited by Brian Whitworth and Aldo de Moor, 2: 685–98. Information Science Reference. Hershey, PA: IGI Global, 2009. http://hdl.handle.net/2142/9710.
- Fischer, Gerhard, and Thomas Herrmann, "Socio-Technical Systems: A Meta-Design Perspective," *International Journal for Sociotechnology and Knowledge Development* 3, no. 1 (2011): 1–33. https://doi.org/10.4018/jskd.2011010101.
- Wajcman, J. "Feminist Theories of Technology." *Cambridge Journal of Economics* 34, no. 1 (January 1, 2010): 143–52. https://doi.org/10.1093/cje/ben057.

Key Technical Terms

- Closed and open **circuits** running in **series** and **parallel**
- Key conductive materials such as **breadboards** and **jumper wires**
- Key electronic components, including **resistors**, **switches**, and **Light Emitting Diodes (LEDs)**, which include both **anode** and **cathode** legs
- Educationally defined skillsets necessary for individual and collective sociotechnical innovation in the real world, including technical, information, cognitive, socio-emotional, application-based, critical, and progressive community interaction skills

5. TED Talks are licensed under CC-BY-NC-ND 4.0.

Professional Journal Reflections:

1. How would you have described the terms digital technology, digital literacy, and digital innovation prior to reading this chapter and related resources? What are some of the contexts leading you to these initial descriptions?
2. What are some of the different descriptions of the terms that come forward through the reading of this chapter and related essential resources? What are some of the contexts shaping the authors of these descriptions, and the contexts shaping your understanding of their descriptions?
3. What are some of the different ways you've learned about networks, information systems, and digital technologies as part of your lived history? How has that influenced your sharing of knowledge and best practices with others? How might we work together throughout this book to teach and learn together about systems in ways that foster sustainable development?
4. From your lived experiences and reading of text and context coming into the reading of the book and in your works so far through the book, in what ways have deductive reasoning, inductive reasoning, and/or abductive reasoning played a role in shaping the creation and marketing of electronic technology around you? Your selection, use, and innovation-in-use of that technology?

2B: Electronic Components in Series

Background Knowledge Probe

Last session, we began to build our first electronic circuits. Along the way, we used diagrams and schematics to inform and guide our work.

- How does this build on your everyday works in the communities of which you are a part? How does it contrast with your everyday works?
- How did you work with others on last session's exercises, whether in person or remotely, synchronously or asynchronously, to achieve the immediate goal of physical making and the more general learning outcomes of the exercise and chapter?
- Going into this session's exercises, what might you do similarly? What might you do differently so as to better achieve immediate goals and general learning outcomes?

Our RGB LED is part of a larger electric circuit providing power to the wall outlets, and from there to the breadboards, and finally to the anode leg of the RGB LED. Wall outlet electric circuits generally work in series with the wall outlet connected to one of the many branch circuit breakers within the building's main service panel. The wall outlet then connects to the next wall outlet, which connects to the next, which then connects to the next, creating a series of wall outlets on this branch circuit.[1] When different electronic devices are plugged into different wall outlets, each electronic device is working parallel to the other electronic devices.

- Unplugging one circuit from the breadboard—for instance, by disconnecting the voltage regulator used to switch 5-volt power to 3.3-volt power used by two for four bright white LEDs used in session one of the orange unit—does not disrupt the other parallel circuits on the breadboard.
- Unplugging something connected to a wall outlet will disrupt the flow of current to itself and everything else connected to it in series, as would be the case if we unplugged the USB

1. This is a simple example for descriptive purposes. In practice, one outlet might be the only one on a branch circuit, as is done for high-demand devices like an electrical stove and rangetop cooking combination or a large refrigerator. Or the wall outlet might branch to two other wall outlets creating two branch series circuits.

to TTL cable from a wall outlet, which then removes the flow of current to all circuits on the breadboard.

- Unplugging something connected to one receptacle of a two-receptacle wall outlet does not disrupt the electrical device plugged into the other, as they are running in parallel to each other, as would be the case if two different USB to TTL cables were plugged in two different receptacles on one wall outlet.
- Unplugging something from this one wall outlet does not disrupt the flow of current passing along the electrical wiring connecting the other wall outlets in series with this branch circuit.
- Plugging something into one wall outlet that requires more amps than provided by this one branch circuit, or plugging in multiple electronic devices that together require more amps than provided by this one branch circuit, will likely trigger the breaker for this branch circuit, stopping the flow of current to everything on this series.[2]
- Disconnecting the electrical wires between one wall outlet and the next further down in the series does not disrupt the functioning of electrical devices running on this wall outlet, or the functioning of those before this one in the series.
- Turning off the branch circuit breaker in the main service panel turns off all wall outlets running in series from this branch circuit, along with all electrical devices plugged into these wall outlets.
- Turning off the main breaker in the main service panel turns off this and all other branch circuit breakers, which turns off all wall outlets, lights, and other electronic components running in series from each of the branch circuit breakers.

Looking specifically at our current breadboard configuration, the on/off slide switch is in series, with the USB to TTL cable on the upstream side and the anode leg of the RGB LED on the other. Unplugging the USB to TTL cable breaks the flow of current for all devices on the breadboard. Sliding the switch from on to off breaks the flow of current to the anode leg of the RGB LED. But beyond the anode leg, the three LED chips of the RGB LED work in parallel. Failure of one LED chip does not necessarily mean the other two LED chips will fail, depending on cause of failure. As we discovered in the first activity of Orange Unit 1B, we could plug only one of the three chips into the blue ground rail, and it would work even as the other two did not because of their lack of a ground connection completing that circuit. If the resistor for one failed, the other two parallel circuits would continue to work.

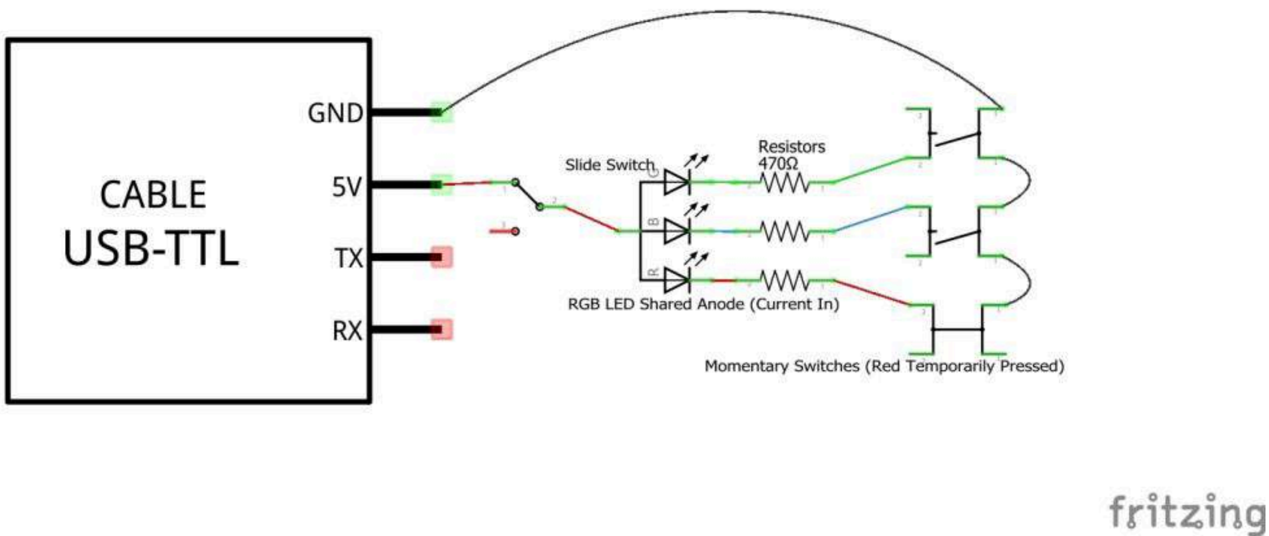
Consider strings of holiday lights, along with other lights plugged into the wall outlets of a circuit. Modern holiday lights will continue to work even if one light within the string fails—each light is its own parallel circuit on the string. But in some cases, there may be two separate series of lights, so that if one series reaches an open state, only half the lights stop working. The second, parallel series of lights remain in a closed state. Also consider that the lights plugged into one wall outlet do not impact

2. Within the United States context, general-purpose wall outlets typically provide 120 volts at 15- or 20-amps.

the lights plugged into other outlets, unless together the number of lights all plugged in require more amps than can be provided by this branch circuit of wall outlets running in series. The different lights are working in parallel to each other.

Exercise: Momentary Switch RGB LEDs in Series

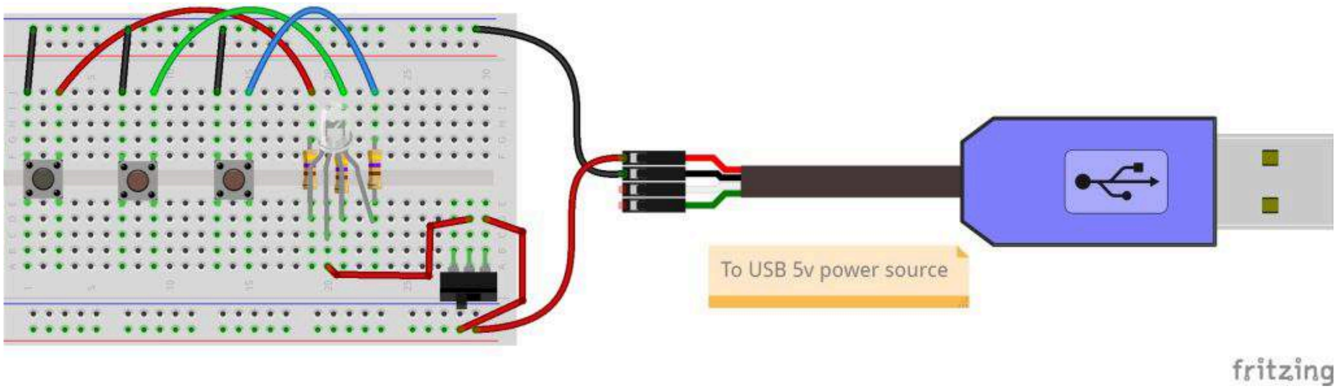
In this activity, we will add three momentary switch push button electronic components. As with the slide switch we added in the first unit, the push button is called a "switch" because it changes the flow of current depending on the state of the switch. These push buttons are called "momentary" switches because this change in state only occurs while a physical action is happening—in this case, the pressing of the button. Each switch is connected in series with one of the RGB LED cathode legs that themselves are running in parallel to each other.



In the Fritzing[3] schematic above, the momentary switch push button symbols for the green and blue cathode legs of the RGB LED are in their default state, and so they are not connected to ground and are open circuits. As the circuits are incomplete, the green and blue cathode legs are "off," just as they are when the slide switch slides to the left, or off, pin, and do not receive current at the beginning of the circuit cycle. The momentary switch push button symbol for the red cathode leg, on the other hand, is in the momentary pressed position. As a result, there is a line connecting the cathode leg and 470 ohm resistor to the rest of the cycle moving to ground. In this schematic, the slide switch symbol also indicates it is in the "on" position to provide 5V power to the common anode leg, so the red LED circuit is temporarily a closed circuit, and the red LED is lit!

This is represented in the diagram below, with two brown momentary switch push buttons for the green and blue cathode legs of the RGB LED, and a black momentary switch push button for the red cathode leg of the RGB LED.

3. Fritzing breadboard graphics are licensed under CC-BY-SA 3.0.



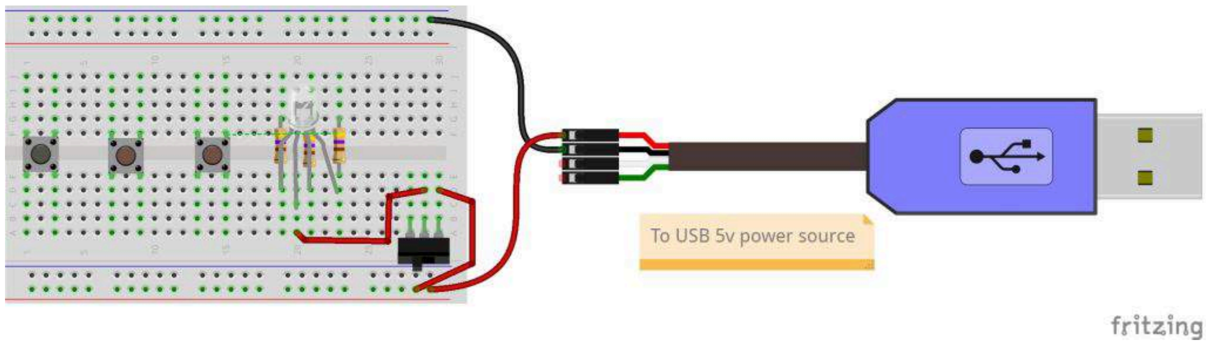
Each of the momentary switch push buttons has four legs, paired such that two on each side bend slightly out and then back in. Each pair is designed to sit above or below the center line, delineating A-E from F-J of a column. Each pair consists of a side 1 and a side 2 of the switch as labeled within the schematic, in which the sides are by default disconnected and are only momentarily connected when the button is pushed down. In so doing, the flow of current is passed through the switch.

Steps

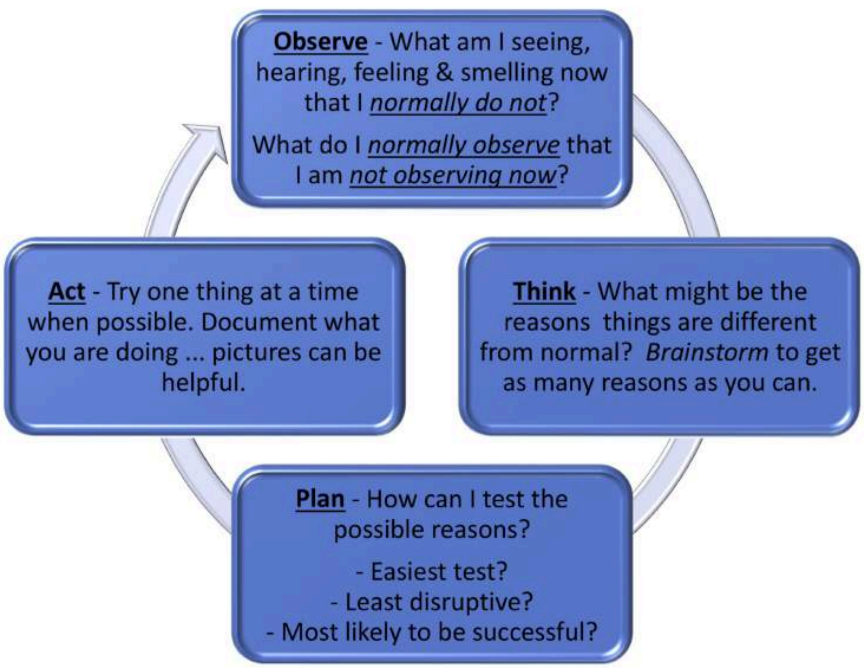
1. Disconnect the male/male **jumper wires** connecting the red, green, and blue cathode legs of the RGB LED to the upper blue rail of the breadboard.

2. Straighten the four legs of the momentary switch.
    - The design of the legs for the switches is primarily for a permaproto board, a breadboard that requires connections be joined together by putting a melted filler metal into the joints.
    - That said, the legs of the switch can also work in our breadboards where there the conductive ends of wires and electronic components are slid into the metal filler.
    - To improve these, use a flathead screwdriver or some pliers to straighten each of the four legs on each of the switches.

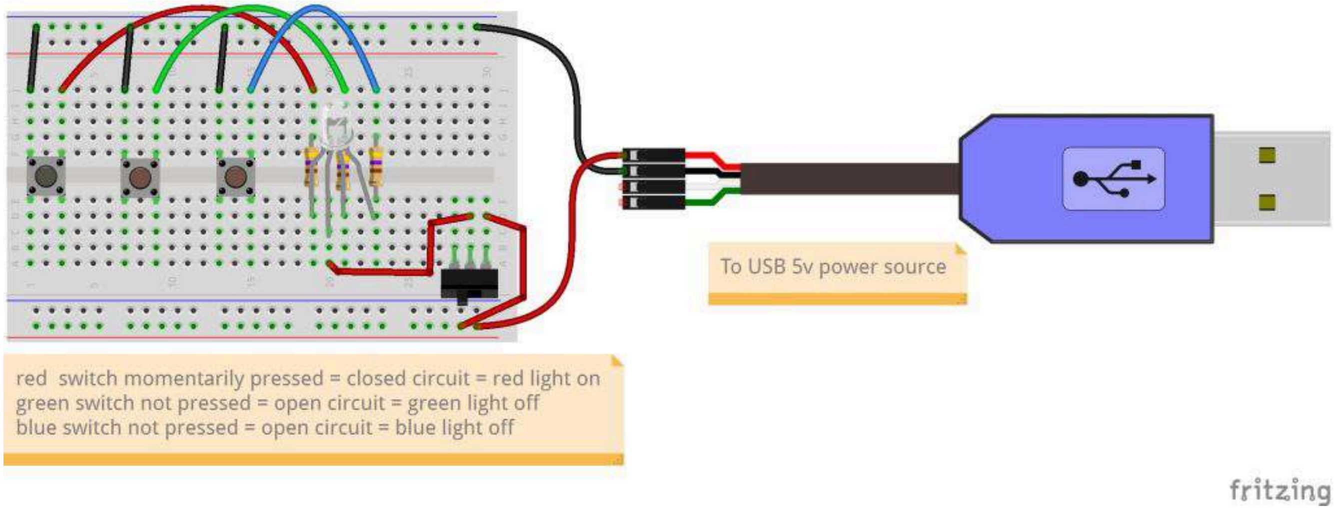
3. Connect one switch to columns 1 and 3. Connect a second switch to columns 7 and 9. Connect a third switch to columns 12 and 15.



4. Using three male/male jumper wires, connect row J of columns 3, 7, and 11 to the upper blue rail of the breadboard as shown in the diagram.
5. Using three more male/male jumper wires, connect row J of columns 5, 9, and 13 to row J of columns 26, 28, and 30. In so doing, you now have set up three switched circuits for the red, green, and blue cathode legs of the RGB LED!
6. This may prove a "not yet" moment for one or more of these circuits.



Bring together pair programming and our troubleshooting strategies to fail forward for any breadboards that are not yet working on your pod. Be sure not to have those who have reached a state of "yet" fix the problem for others, but rather provide extra eyes on the instructions and troubleshooting strategies list to work collaboratively on the problem. In so doing, all have the potential for social and technical learning beyond what either partner might do individually.



Key Takeaways

The flow of an electric current through a **circuit** occurs in a predictable and controllable fashion, and deductive reasoning serves exceptionally well in the exploration of that which is predictable and controllable. As we saw in this exercise, no matter the physical change we make with the momentary switch push button, current always passes through the resistor and through one leg of the momentary switch. The momentary switch plays a **make-or-break** role. For our momentary switch, while the button is unpressed, it is in a break state, and the circuit does not reach ground to complete the circuit. It is an **open circuit**. We also saw that when the push button is sufficiently pressed to reach a make state, current flows to the other leg of the switch. Only in returning the current to ground does it move from an open to a closed state. It is a now a **closed circuit**. In the above Fritzing image, the black dot in the center of the red momentary switch is used to indicate a pressed button, while the brown dots of the green and blue momentary switches are used to indicate buttons that are *not* pressed.

We've now seen several of the many forms in which **switches** come. In this exercise, the slide switch and momentary switch push buttons are mechanically moved between the make state and the break state. In this aspect, they are like the mechanical breaker switch of a building's main electrical panel and the multiple branch circuit breakers within the main panel which was explored to introduce this chapter. As with the slide switch, breakers can be physically moved from the make, or ON, side, to the break, or OFF, side. For building electrical, though, these breaker switches also have additional components to electrically trigger mechanical movement of these under certain circumstances, such as when electronic devices are pulling more amperage than is allowed by the breakers.

This exercise also compared the view of a circuit using **schematics** and diagrams which could be viewed simultaneously with the physical, constructed breadboard prototype. As we seek to demystify technology, we'll continue to see how multiple points of view help to fill in gaps in our understanding of what is actually happening "beneath the hood," and how we might implement it in our own works to shape its impact in our daily lives.

Exercise: Adding LED Sequins in Series and in Parallel

In the first session of the Orange Unit, we discovered that if you increase the size of the resistor from 470 Ω to 10,000 Ω, the brightness of an **LED** is reduced considerably. Indeed, the increased resistance dims the LED more than a reduction in the voltage of our electrical source from 5 volts to 3.3 volts.

When working with electrical conductors, there is a proportional relationship between three measurements influencing the passage of an electrical charge:

- **Electromotive force, listed in volts** (e.g., 5V, 3.3V)
- **Electrical resistance, listed in ohms** (e.g., 560 ohms, 10K ohms)
- **Electrical current, listed as amperes, or amps**

The influence of different voltages and different ohms on the brightness of an LED was tested in several ways to demonstrate that as we increase voltage, we also increase brightness. On the other hand, as we increase the ohms of resistance, we decrease brightness. This is because voltage increases current, while resistance decreases current.

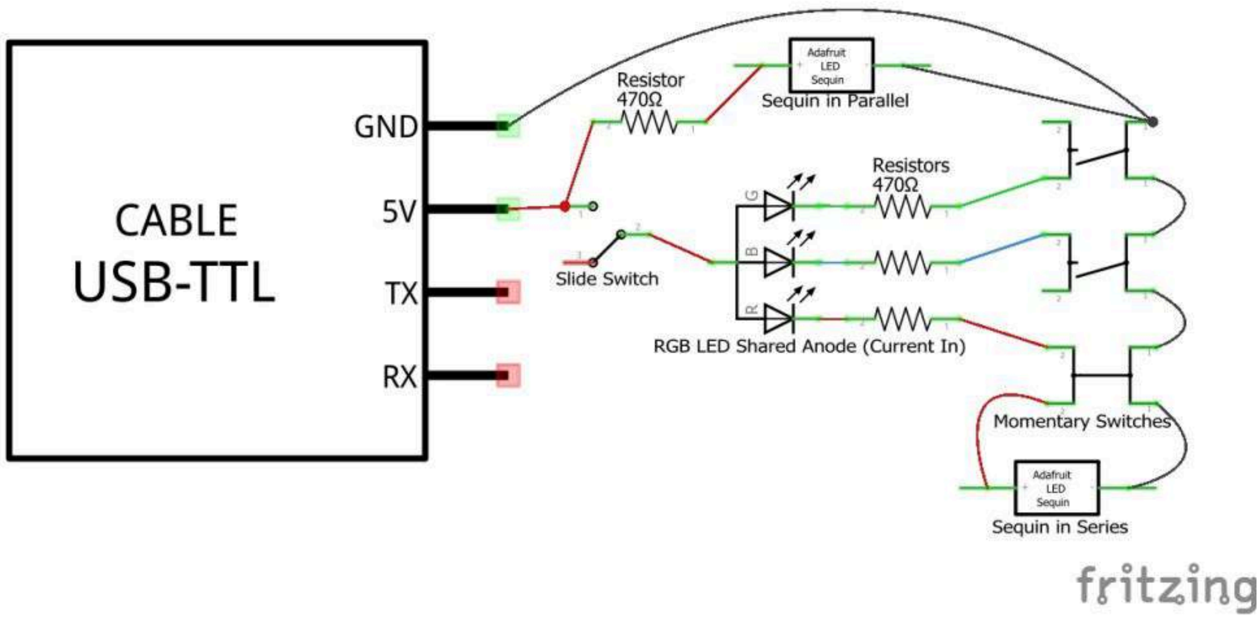
And it is current that is used by an LED to make light. More current, more light.

A fourth measurement, **watts, is a comprehensive measurement of power** as influenced by the above three measurements. You may recognize the term “watts” in relation to lighting, as in a 60-watt incandescent light bulb. An equivalent-brightness LED light bulb would use about 8.5 watts of power. Our 10-millimeter diffused RGB LED is rated at 150 milliwatts (mW) of power at maximum.

Like lights everywhere, they dissipate power in a predictable physical manner, such that at a certain point, increasing the number of LEDs within a defined circuit decreases the brightness of those LEDs.

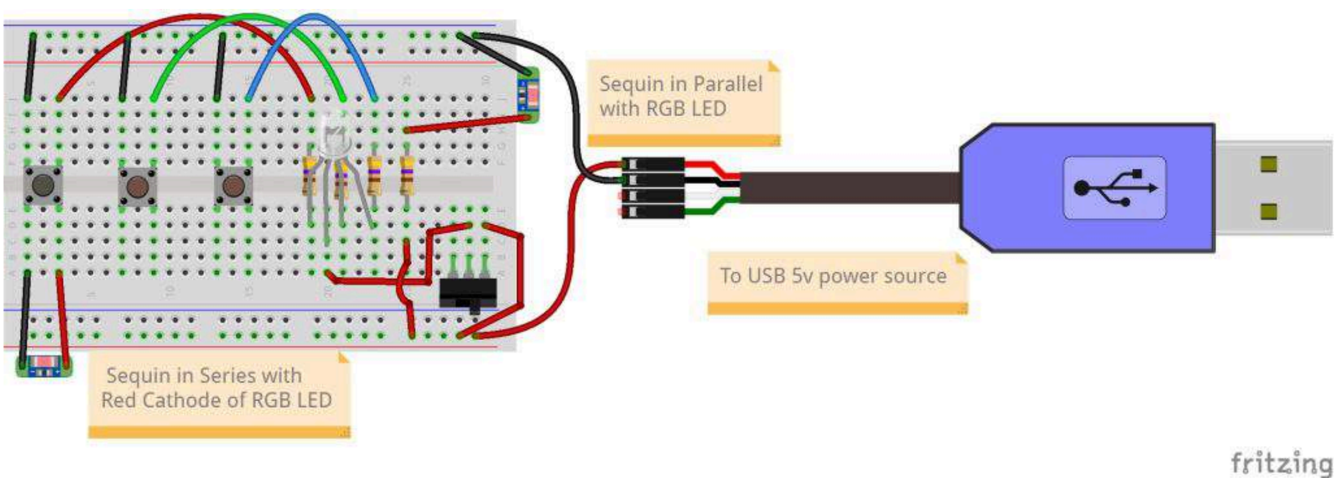
In this exercise, we will temporarily add two LED Sequins into the mix. Each Sequin is an **integrated circuit** comprised of a **resistor** and an LED. These are incorporated onto a printed circuit board along with conducting material, such that when voltage and ground sources are applied to the + and − strips, the LED is lit. If you pick one of these up, you’ll notice they look and feel like some sort of flat plastic or other material. That’s the feel of a typical printed circuit board (PCB) used as the base for each of the microcontrollers and microcomputers we make use of daily.

For this activity, we’ll add one LED Sequin integrated circuit in series with the red cathode leg of the RGB LED. Breaking the slide switch, LED, or momentary switch connections in any way breaks the circuit for all items working in the series. We’ll add a second LED Sequin integrated circuit that will work in parallel to the other LED Sequin. As it is connected directly to its own 5-volt and ground rails, it will remain a closed circuit as long as power is provided to the breadboard. Together, these will allow us to further test the relationship of electronic components within a series and in relation to those of other electronic components in other, parallel circuits.



Note within the schematic that we have four parallel LED circuits:

- A red parallel LED circuit in series with an LED Sequin, a resistor, and a momentary switch, all with a single lead to ground.
- A blue parallel LED circuit in series with its own resistor and momentary switch with a separate lead to ground.
- A green parallel LED circuit in series with its own resistor and momentary switch with a third separate lead to ground.
- A second parallel LED circuit with its own resistor and a fourth separate lead to ground.



Steps

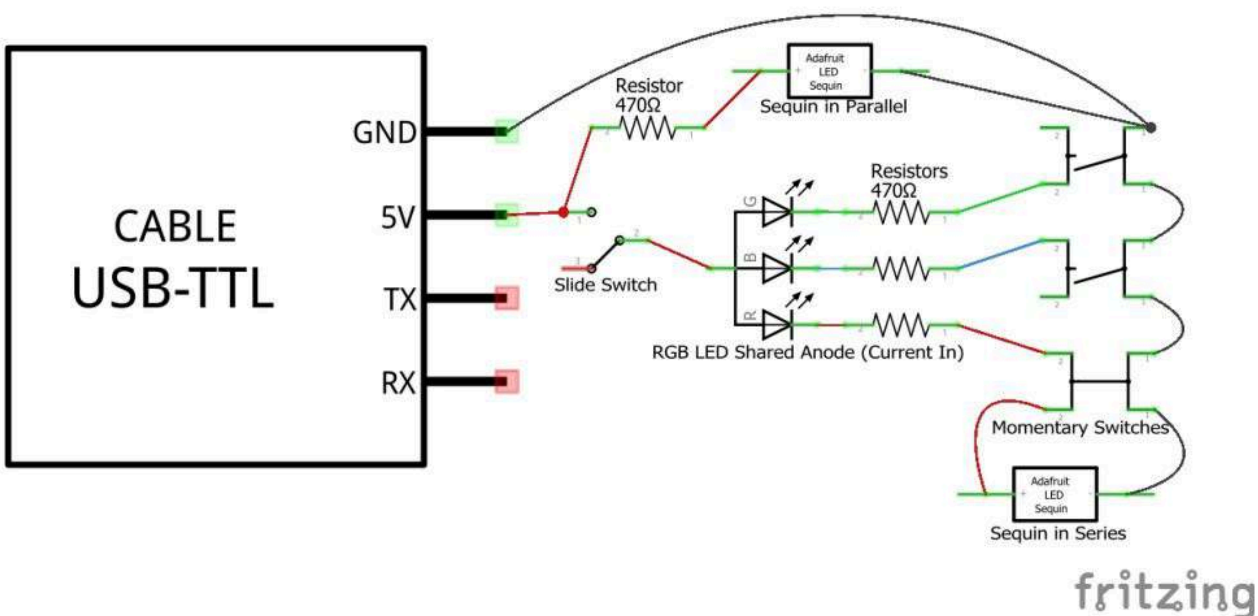
1. Attach a male/alligator clip jumper wire from row A of column 3 to the + side of the sequin.
2. Attach a male/alligator clip jumper wire from row A of column 1 to the − side of the sequin.

3. Press the "red" momentary switch to confirm that both the Red RGB LED and Sequin LED series now light up when pressed.
4. Attach a 470 Ω resistor in column 25 to bring together rows A-E and F-J.
5. Connect a male/male jumper wire from the red 5V rail at the bottom to any available hole in rows A-E of column 25.
6. Connect a male/alligator clip jumper wire in any available hole in rows F-J of column 25 and to the + side of the sequin.
7. Connect a male/alligator clip jumper wire to the top blue ground rail and to the – side of the sequin. This sequin should remain on, independent of the condition of any of the other LED circuits.

There are now four parallel LED circuits with different light colors, brightnesses, and conditions in which they move from off to on and back.

Key Takeaways

There is a key distinction between **circuits** that are running in **parallel** versus those running in **series**. If there are **LED**s in parallel, there are multiple paths **current** can flow from source to ground. If the LEDs are in series, there is only one path from source, through the LEDs, and back to **ground**. Look again at the **schematic** for this exercise:



Trace the route from 5V to GND for the Sequin in parallel. The current flows from a 5V power source through a **resistor** and into the plus (+) side of the Sequin. It then leaves the negative (-) side of the Sequin before passing to GND. At the same time, the schematic makes note that as ground is reached through the blue rail of the **breadboard**, it completes the flow of the circuit down the same path as the other circuits. But this Sequin is doing it in parallel to the circuits of the other LEDs.

Now trace the path of the Sequin in series. The current leaves the 5V and passes through the slide switch, which if "on" sends the flow of current to the common **anode** leg of the RGB LED. Here it takes the path leading it to the red **cathode** leg of the LED and passes through the resistor before entering one leg of the momentary switch. If this switch is pushed down, both the Sequin and the "red" RGB LED pass through an opposite leg of the momentary switch and return to ground. These LEDs are running in series to each other. There is only one path for both to progress from 5V to GND.

Circuits in parallel and series each have their pluses and minuses. The Sequin in parallel is not controlled by the slide switch and so has a constant brightness compared to the Sequin that passes through the slide switch and can be switched on or off. This is an advantage if you want an LED with constant brightness to give you an indication of something being plugged into a wall outlet or being active. It is a negative if you want to be able to turn off a light even though it still has 5-volt power available.

Of importance, note that we also find the amount of electricity used in the parallel LED circuit is double that used for the LEDs in series. There is a higher power consumption that would drain a battery faster or would require more power sent to the building from the electrical grid. While minute in overall terms, this is an important consideration when working to reduce overall power consumption.

When LEDs run in parallel, they each get one hundred percent of the current passing through the circuit. When LEDs run in series, they share a percentage of the current—in this example, fifty percent each. Since brightness depends on the amount of current passing through, the parallel Sequin will be brighter than the series Sequin. You'll also note the brightness of the red RGB LED dims as well. Each of the three cathode legs of the RGB LED are running in parallel as they each follow a unique path back to ground. If each of these also had their own Sequin running in series, the cost in brightness might be offset by only using the 5V power option, by decreasing the resistor from 470 ohms, or a combination of the two.

As we see with hanging lights, those connected in series mean that a break at one point breaks everything within that series as the flow of current is switched from closed to open. Lights in parallel allow for one LED to break while others continue to work.

As we see from this example, there is often no clear and consistent answer as to which is the better circuit. And it's here that strategic design joined with rapid prototyping and testing across diverse communities of users can help arrive at a better answer. Best does not exist, as we can never fully know the contexts for each user. Nor can we fully anticipate (or sometimes even make a rough guess regarding) the changes in context for a given user. As we move forward in this book, the takeaways here will hopefully help as we delve ever deeper into the sociotechnical devices and systems from which we select, and which we use and help others to use as part of our daily lives.

Do Something New!

We'll come back to the RGB LED circuitry on the breadboard in session 4 of the Orange Unit, where we'll use the right half of the breadboard to make connections with the Raspberry Pi general purpose input and output pins. This way, we can use Python programming code to read momentary switch inputs and open and close the RGB LED circuits. Then, at the start of the Rainbow Unit, we'll bring together the programmable RGB LED circuitry of the Orange Unit with the programmable Toolbox Trumpet microcontroller we'll use in the Blue Unit to create a demonstration model that could potentially be used to make your own Internet of Things device.

As we've worked through the exercises in sessions one and two of the Orange Unit, one key takeaway has been to encourage consideration of how demystifying technology requires risk-taking that brings us to moments of failure. As we observe these "not-yet" occurrences, think through possible social as well as technical factors shaping these occurrences, develop a plan for testing, and then act one step at a time upon this plan, we see in new ways what is underneath the hood of electronics. We begin unpacking how it is shaped by others. We start seeing how failure is sometimes as much a mismatch between the shaper and the user of the device as it is our failing. And we start thinking more deeply about how we can work to either align ourselves with the shapers of the technology or find ways to further shape the technology ourselves to better align with our ways of using it. To do this, we also are unpacking the importance of reflecting on our various creative actions and making written or spoken recordings of those actions and reflections. Failing forward can be a mix of changing our way of doing something and of changing how that something gets done.

In reflecting back on your actions and reflections of sessions one and two, have there been some moments when you found a mismatch between my shaping of the circuit's design and your use of that circuit? Have you had a flash of inspiration regarding a better way to do the layout of the circuit to perhaps make it more usable for you?

Here's an opportunity to *Try Something New!* using the breadboard, wires, and electronics as the rapid prototyping tool they're intended to be. Consider testing out your idea(s) by rebuilding the switched power for the common anode of the RGB LED; the red, green, and blue momentary switched circuits; and/or the sequin LED circuits using columns 1-30 of the breadboard. Perfection is not on the table; bravery is!

Wrap Up

This exploration of electronics lays the foundation for what we may want to design and create as part of our ongoing person-centered valued designs and builds. You can likely find many other electronics in tinker- and makerspaces around you, which you could use to further dig into the many components used in the electronics of which you make daily use. Demystifying technology in this way can sometimes be an essential act helping you to see some of the situations limiting you or others from being and doing what you most value. This requires us to move beyond technical skills development, though, to include additional skills that help us to continually probe the mutual shaping and amplification of sociotechnical devices that has happened and is happening within this moment. As we work to problematize these sociotechnical artifacts, we also have the potential for advancing a more person-centered approach within our personal and professional lives.

Comprehension Check

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with answers, see the online version.]

1. Fill in the following chart with these terms: **Breadboard**, **Circuit**, **LED**, **Integrated Circuit**, **Switch**, **Resistor**, **Jumper Wire**

Definition	Term
The electronic component commonly used to reduce the flow of electricity, as calculated using Ohm's law, from one component to another.	
A system bringing together electronic conductors and components in such a way as to form a complete and closed path through which an electric current can flow from source to ground.	
A semiconductor wafer with a collection of tiny resistors, capacitors, and transistors built to serve a wide range of electronic functions.	
A collection of conductive metal terminal strips placed in an ordered pattern of holes to provide rapid testing of flexible electronics systems.	
A piece of insulated copper or other conductive material commonly used in prototyping to direct the flow of electricity from one electronic component to another.	
The electronic component used to make and break the flow of electricity along a path.	
The electronic component made of two terminals, or legs, one of which is positive and the other negative. When electricity flows properly into the positive leg and out of the negative leg, a light emitter is activated.	

2. When actively working together to design, build, and test new technologies in support of individual and group goals, we need a broader mix of skills than just the technical acumen taught in support of our use of assembly-line produced technologies. Match the following

educationally defined names and definitions: **Socio-emotional Skills**, **Cognitive Skills**, **Information Skills**, **Cognitive Skills**, **Socio-emotional Skills**, **Socio-emotional Skills**.

- Logically analyzing and organizing problems in ways that allow use of digital and analog tools to help solve them, and to generalize new processes to other problems.
- Communicating and collaborating with others, along with internalized personal confidence, persistence, and tolerance, in order to tackle complex, ambiguous, open-ended problems.
- Seeking, evaluating, interpreting, and applying relevant and trustworthy information across multiple media.
- Ability to generalize past and current problem-solving to new contexts.
- Ability to embrace failure as an essential step in project development, to plan for failure, and to use failure as a stepping stone.
- Ability to analyze current skills, compare them to needed skills to accomplish a given project, and develop a plan of action to achieve skills development.

3A: The Unknown Tech Innovators

Background Knowledge Probe

1. Draw a picture of an innovation in the process of being innovated.
2. Looking at your completed drawing, consider the following:
    - Describe the innovation in your drawing. What might be its uses? What might be its impact once it is released to the target population?
    - Who is doing the innovation? Why? For what goal(s)?

Who Shapes Technology?

"A Critical Social + Technical Perspective" and associated essential resources highlighted ways technologies are a seamless, indivisible combination of social and technical components that are co-created by everyday users. But as Wajcman notes, those things white, male, and middle- and upper-class innovators primarily focus on creating are specific "male machines" rather than broader, everyday life technologies.[1] When workshop and class participants are asked to draw a picture of "an innovator innovating," many draw individual white men innovating the modern digital technology machine. This tendency towards drawing innovations and their innovators in this way is a representation of the current dominant narrative in much of U.S. culture.

We also considered how once developed and made available for broader use by people, technologies do not remain static, but rather are adapted and co-created to fit within ever-changing contexts.[2] The co-creation happens when uses of technology are done in ways that amplify our individual and group actions in the world.[3] While this mutual shaping of technologies is ongoing, we often think of the dig-

1. Judy Wajcman, "Feminist Theories of Technology," *Cambridge Journal of Economics* 34, no. 1 (January 1, 2010): 143–52. https://doi.org/10.1093/cje/ben057.
2. Bertram Bruce, Andee Rubin, and Junghyun An, "Situated Evaluation of Socio-Technical Systems," in *Handbook of Research on Socio-Technical Design and Social Networking Systems*, eds. Brian Whitworth and Aldo de Moor, Information Science Reference (Hershey, PA: IGI Global, 2009), 2: 685–98. https://doi.org/10.4018/978-1-60566-264-0.ch045.
3. Kentaro Toyama, *Geek Heresy: Rescuing Social Change from the Cult of Technology* (New York: PublicAffairs, 2015).

ital technology product itself in the more static conceptualization. As such, often in these drawings of an innovator innovating, it's a certain subgroup of initial innovators, and does not include in the drawing the intermediary and everyday co-creators of a digital technology. Nor do we include in the drawing those who are shaping the innovation and design as part of the engineering, coding, marketing and distribution, and other stages of the product's life cycle.

Take a moment to review your picture of an innovation in the process of being innovated, and reflect on your considerations of the questions above. In what ways might your readings and works so far in the book have influenced your response to the Background Knowledge Probe?

Broadening our Understanding of Creating

Today's Maker movements often emphasize a broader set of activities in the process of making something. While some of these spaces emphasize specific digital technologies, many others view this more broadly as craft-making. As Bruce Sinclair notes:

> Since the nineteenth century, engineering in this country has depended on a published literature and on advanced formal instruction that has included physics and mathematics. Craft skill depends on a different kind of knowledge, most of it unwritten and learned on the job. Apprenticeship—whether institutionalized or not—rests on emulation and repetitive practice in the interest of acquiring manual skills, and it is married to experience with the ways in which materials behave in different circumstances. Not only is this kind of knowledge complex and difficult to transfer; it gains importance when considered in the context of the history of American slavery, the formal acquisition of knowledge by slaves having been forbidden by law.
>
> Bruce Sinclair. "Integrating the Histories of Race and Technology."[4]

Throughout U.S. history and continuing today, people of color and women have often been kept from educational opportunities, particularly in areas such as engineering, and more recently, computer science. And yet, hidden are the myriad ways elite occupations have been successfully done via these very same people while mainly white males are given the credit for the accomplishments made by the broader group of innovators, designers, and creators.

Carolyn de la Peña has highlighted how finding these hidden stories regarding the design and innovation of technologies by white women have required challenging but essential meta-historical narratives through personal papers and the records of their and their husbands' activities. Finding the records of people of color are even more challenging. To proceed, de la Peña notes:

4. Sinclair, Bruce. "Integrating the Histories of Race and Technology." In Technology and the African-American Experience: Needs and Opportunities for Study, 1–17. Boston: MIT Press, 2004. https://mitpress.mit.edu/9780262195041/.

> Rather than imagining "race" as a term that describes particular individuals marked as non-white, I want to suggest we think of race as an epistemology at play in all technological production and consumption. This concept makes it possible to see the significance of the obvious: that white people have race. And they make it, sustain it, and protect it in part through technology. More importantly, this approach suggests that it is not only the problem of source that keeps us from integrating race fully into our analyses. Instead, the real difficulty occurs in tandem: difficult-to-locate sources combine with our own tendencies to fail to see all that can be found in what is available, and to creatively *engage* and *interpret* in order to draw race out of the archive.
>
> Carolyn de la Peña, "The History of Technology, the Resistance of Archives, and the Whiteness of Race"[5]

As a white, cisgender, heterosexual male scholar working at a higher education institution in the United States, it is essential that I recognize the ethical dilemmas involved in writing this book introducing networked information systems within the information sciences. In doing so, I continue to work to identify ways in which I am building from, and integrating into this work imperialist, patriarchal, and racist ideologies which are embedded within general Western and specific United States culture and practices.

In order to broaden my understanding of digital technology design and innovation, I have drawn heavily from Community Informatics (CI) practice, a field that seeks to make "effective use" of technology through a practice of community inquiry, participatory design, popular education, and asset-based development to enhance quality of life.[6] This led me to co-develop a list of critical questions for Community Informatics in practice.[7] As related specifically to the philosophy of technology, rather than bringing forward the digital technologies we think are essential for information seeking and sharing, we need to instead consider:

> What everyday technologies might be unseen and displaced because of an overly narrow definition of what should be considered an appropriate technology? Who are the local innovators whose technologies might be championed as part of a CI project?… How might the voices of technology skeptics and traditionalists inform adoption, or non-adoption, of a CI project? What important insights regarding culture, values, and history are these perspectives bringing to the engagement?

5. de la Peña, Carolyn. "The History of Technology, the Resistance of Archives, and the Whiteness of Race." Technology and Culture 51, no. 4 (October 2010): 919–37. https://muse.jhu.edu/article/403272.
6. See, for example: Gurstein, Bruce, Campbell, and Eubanks, all available at https://openjournals.uwaterloo.ca/index.php/JoCI/index.
7. Martin Wolske and Colin Rhinesmith, "Critical Questions for Community Informatics in Practice from an Ethical Perspective," *The Journal of Community Informatics* 12, no. 3 (2016): 236–42. https://doi.org/10.15353/joci.v12i3.3289.

Martin Wolske and Colin Rhinesmith, “Critical Questions for Community Informatics in Practice from an Ethical Perspective”

Lesson Plan

With this historical context in mind, in the next session we will consider the evolution of computers and their “building blocks.”

Essential Resources:

- Articles and chapters:
    - de la Peña, Carolyn. “The History of Technology, the Resistance of Archives, and the Whiteness of Race.” *Technology and Culture* 51, no. 4 (October 2010): 919–37. https://muse.jhu.edu/article/403272.
    - Wolske, Martin, and Colin Rhinesmith. “Critical Questions for Community Informatics in Practice from an Ethical Perspective.” *The Journal of Community Informatics 12*, no. 3 (2016): 236–42. https://doi.org/10.15353/joci.v12i3.3289.
- A few sources about women and people of color in technology, past and present:
    - National Public Radio’s Short Wave Podcast. “Henry Cort stole his iron innovation from Black metallurgists in Jamaica.” August 7, 2023. https://www.npr.org/2023/08/03/1191989712/henry-cort-stole-his-iron-innovation-from-black-metallurgists-in-jamaica. (11-Minute Listen.)
    - Biography.com. “Black Inventors.” Accessed August 3, 2023. https://www.biography.com/black-inventors.
    - The White House: President Barack Obama. “The Untold History of Women in Science and Technology.” Accessed February 9, 2020. https://obamawhitehouse.archives.gov/women-in-stem.
    - McDonald, Clare. “Computer Weekly Announces the Most Influential Women in UK IT 2018.” ComputerWeekly.com, September 26, 2018. https://www.computerweekly.com/news/252449081/Computer-Weekly-announces-the-Most-Influential-Women-in-UK-IT-2018. (Especially note #8, Carrie Anne Philbin, director of education at the Raspberry Pi Foundation.)
    - Adams, Genetta M. “17 Black Internet Pioneers.” The Root, February 12, 2012. https://www.theroot.com/17-black-internet-pioneers-1790868134.
    - Black Enterprise. “List of the Most Influential Blacks in Technology – Black Enterprise,” November 9, 2018. https://www.blackenterprise.com/list-of-the-most-influential-blacks-in-technology/.

- Key highlights about the Raspberry Pi Foundation, Adafruit Industries, and their respective teams:
    - ElectronicsWeekly.com. "How They Make Raspberry Pi in the UK – Electronics Weekly Visits Pencoed. Metropolis Multimedia." YouTube, September 11, 2013. https://www.youtube.com/watch?v=Tza6Hl8wSJ0.
    - Raspberry Pi Foundation. "About Us." Accessed February 9, 2020. https://www.raspberrypi.org/about/.
        - Raspberry Pi Foundation. "Raspberry Pi Foundation Annual Review 2018." Raspberry Pi Foundation, 2018. https://static.raspberrypi.org/files/about/RaspberryPiFoundationReview2018.pdf.
        - Raspberry Pi Foundation. "Raspberry Pi Foundation Strategy 2018-2020." Raspberry Pi Foundation, 2018. https://static.raspberrypi.org/files/about/RaspberryPiFoundationStrategy2018%E2%80%932020.pdf.
    - Adafruit Industries. "About Us." Adafruit Industries. Accessed February 9, 2020. https://www.adafruit.com/about.
    - NYC Mayor's Office of Media and Entertainment. "Her Big Idea, Featuring Limor Fried." YouTube, November 6, 2017. https://www.youtube.com/watch?v=cwpIx7n6UYg.

Key Technical Terms

- Core hardware components of a computer, including the **system board/motherboard**, the **central processing unit** and other **controllers**, types of short-term **memory** and longer-term **storage**, and input/output devices
- Core software components of a computer **operating system**, including the **kernel**, **window system**, **desktop environment**, and top-level **themes and skins**.
- The history and overarching principles of the three major types of operating systems used today, Linux, Mac OSX, and Microsoft.

Professional Journal Reflections:

1. How has your consideration of digital technology and digital innovation changed after this reading of the texts and of the contexts?
2. Who are some of the unknown innovators of your everyday technologies? Why were they unknown until now? How might they still be partially unknown by you, or by others?
3. In what ways have you been an unknown innovator of a technology? Why were you unknown until now? How might you still be partially unknown by yourself and by others?

3B: Computer Building Blocks

Background Knowledge Probe

In chapter one, "Introduction to Electronic Circuits," the background knowledge probe had you create a list of the electronics you make use of on a regular basis. As you did so, you considered who designed it, what parts might have been used inside and out to build and use it, and what you did with any items that started to work less than optimally.

- Now create a new list, this time documenting your daily-use computing devices such as laptops, tablet devices, and smartphones.
    - What external and internal electronic components, such as resistors, capacitors, diodes, transistors, switches, sensors, LEDs, copper wires, pins and clips, integrated circuits, and printed circuit boards, come to mind as you think about your computing devices?
    - What design choices and design objectives might have inspired the layout of the system and the various circuitry, mechanics, and artistry of its layout function, look, and feel?
    - What have you done with any components that started to work less than optimally?

Technical Introduction

In the technical chapter of session 1, we began deconstructing the components that make up our daily electronics, something we dug further into during chapter 2, "Electronic Components in Series." We've also been expanding our fail-forward troubleshooting and innovation-in-use remixing as we further open up the hood of electronic systems and the social influences shaping, and being shaped by, these. Before launching the Raspberry Pi microcomputer, let's spend a little time exploring how electronics, and often times code, are combined in a range of ways to create computers. In this chapter, we'll:

1. Explore the hardware building blocks that make up all computers.
2. Examine the operating system building blocks that provide the major software package that supports the basic functions of a computer and other computing devices.

Computer Hardware Building Blocks

Electronic components like resistors, switches, LEDs, variable resistors, capacitors, other diodes, photocells, transistors, and integrated circuits combine to create many of the everyday devices around us. These can be prototyped using **breadboards and Perma-Proto breadboards**, ultimately leading to the creation of printed circuit boards. All computers are initially made using these prototyping tools as they evolve from innovation and design, to testing, to production and marketing, and distribution.

Inside the case of every computer, we will find one or more printed circuit boards. The **system board or motherboard** is the main board connecting computer parts. It brings together the many different electronic components we have studied, as well as the connectors, sockets, and ports that provide internal and external connections to other electronics.

All computers are made up of the same basic building blocks. In addition to the motherboard, there are several key parts of a computer's hardware:

- The **central processing unit, or CPU,** the core integrated circuit controlling and executing operations.
- **Memory**, the integrated circuits that temporarily hold data until needed by the CPU or other processors.
- **Data storage**, which includes devices like hard drives, optical media such as DVDs, solid state drives, and flash drives even when power is turned off, but that cannot share data fast enough to work directly with processors.
- **Controllers**, the crossing guards that control the flow of data from inputs to processors, and from processors to outputs.
- **I/O or Input/Output Devices** that you connect to the computer using Universal Serial Bus, or **USB** port, **HDMI** port, **Ethernet** jack, audio jack, and others, such as the Raspberry Pi **GPIO** port.

Integrated Circuits

Processors: Do the thinking work

- CPU: The main thinker and boss for the computer

Memory: Hold data until needed by processors; faster than storage but more expensive

- RAM: The main system memory; it forgets when power is off

- Flash: Changes circuits in a way that remembers even without power (USB stick, solid state drive)

Controllers: Move data from inputs to processors and from processors to outputs

Storage

Mechanical changes to media to store data even when power is turned off

- Magnetic media: Uses positive and negative charges (e.g., hard drive)
- Optical media: Uses lasers to detect presence or absence of reflected light (e.g., CD and DVD)

Printed Circuit Boards

System or Motherboard: Main highway connecting components; edge connectors and sockets provide for expansion

- Memory module: Board with group of memory integrated circuits (e.g., DIMM)
- Expansion cards: Board with controller and integrated circuits to add capability to computer (e.g., video card, network card)

Case

- Protects the building blocks from damage
- Holds the power supply used to turn wall power into the power used by components (desktop)
- Provides space for air to flow around and to cool components
- Provides a surface to disperse heat away from components (laptop)

Input/Output Devices

- Input devices let users control the computer (e.g., keyboard and mouse, touch-sensitive devices)
- Output devices inform user what is happening (e.g., monitor, printer)

Computer Operating System Building Blocks

Operating systems are the software managing the various components of a computer's basic functions, from coordination between the many applications that allow us to do our daily computer-based activities, to the **kernel** used to span the boundary between programming code and the physical electronics. The building blocks of an operating system can be seen below:

Themes and Skins

- Pre-set package containing graphical appearance details. Supported in some versions of Windows and Linux. Available as third-party applications in other cases.
- Supported for many applications like web browsers.
- Changes the look and feel of many features at once (e.g., background colors, text font and size, icons, mouse cursor, etc.).

Desktop Environment

- The desktop environment is a collection of software that provides a predictable look and feel.
- Includes a Window Manager that controls the placement and appearance of windows, support icons, menus, etc.
- For some operating systems, such as Windows and macOS, the desktop environment is the branded look of the operating system and cannot be changed.

Window System

- This is the part of the graphical user interface that communicates with the kernel.
- Many operating systems allow remote interfacing with the window system, either directly or through third-party applications.

Bundled Software

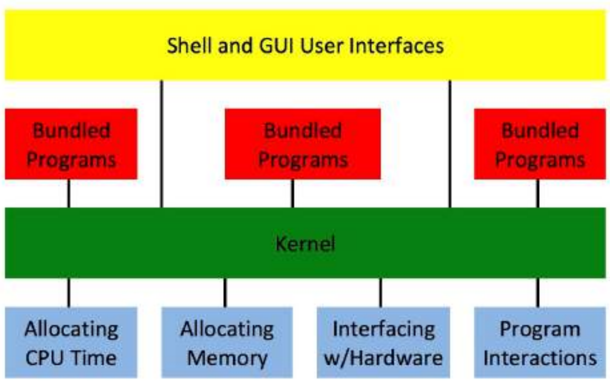
- Applications like **file managers**, user managers, software managers.
- Applications like web browsers, text editors, photo and movie editors.
- Device and print drivers determine which hardware is “plug and play” ready.

Kernel

- As the heart of the operating system, it is the go-between from applications to input/output controllers, memory, the CPU, and storage devices.

When working with a computer, names bring things together, hiding from view many of the hardware and operating system building blocks. I'm working right now on a MacBook Air. This laptop has all of the key hardware building blocks pulled together into a single system board within a laptop case. It is installed with the macOS operating system. On my desk is a Motorola moto e[5] play smartphone. This also has all the key hardware building blocks pulled together into a single system board, this time within a smartphone case. It is installed with the Android mobile operating system based on a modified version of the Linux operating system kernel and other open source software. My spouse uses a Lenovo laptop much like the MacBook Air but with a few more I/O ports and a larger laptop screen. It is installed with the Microsoft operating system.

We typically merge the hardware and software together as one device with a means for bringing in human input, using software to collect and respond to that input, and to provide output back to the human.



We primarily interface with an operating system via the shell and graphical user interfaces. These connect us to the kernel of the operating system which interfaces with installed applications as well as with the computer hardware and other computer programs.

All modern operating systems include both a graphical user interface (GUI) and a text-based shell known as a command-line interface (CLI) or terminal window. For the most part, we spend our computer time using a keyboard, monitor, and mouse/touchpad to work within these GUIs without even knowing we're in an operating system that controls each and every action we take on that computer. It's just the computer and we're just trying to do our work with a range of applications like web browsers, email clients, and word processors.

Sometimes, though, we need access from a distance, and so we use remote desktop software. In some cases, remote desktop software is operating system independent, as is the case with the Virtual Network Computing (VNC) server installed on the Raspberry Pi Operating System by default, and that can be installed on almost all different personal computers and smartphones. In other cases, it is designed for a specific operating system, as is the case for Apple Remote Desktop (Apple) and Remote Desktop Protocol (Microsoft).

For many server computers, including devices like routers and a range of open-source content management systems (CMS), these are designed to serve as a simple, responsive web interface to a computer that otherwise runs headless. Headless means the computer is run without a monitor, keyboard, and/or mouse connected to it. The content management system provides us with a remote interface device to access and use certain tasks that would otherwise be accessed using a GUI through a keyboard, mouse, and monitor or through a remote desktop application.

Another common way to work with headless devices is through a Universal Asynchronous Receive and Transmit (UART) serial connection or an Internet-based secure shell (SSH) connection. These serve to provide a simple, text-based shell, also sometimes called a console or terminal window. Indeed, the Raspberry Pi starter kit provides a default backup to connect a personal computer to a server or microcomputer using a USB (Universal Serial Bus) to TTL (Transistor Transistor Logic) cable to facilitate UART communications as a way to assure troubleshooting when all other means for user interface with a device has failed. This is how we'll start our work with the Raspberry Pi.

Where Does the OS Live?

The operating system is installed and maintained using a storage device like a hard drive or MicroSD card. But **storage devices are like our old paperwork that we store up in an attic**. We can't do anything with it beyond just storing it there. When a computer is turned on, those instruction sets in the operating system which are immediately needed are moved out of storage and into random access memory (RAM). We sometimes call this booting a computer, also called boot up or start up. This startup sequence first does a safety check of the system, then looks to one or more possible locations for the master boot record or boot manager, which then takes over control to bring forward the next modules of operating system code needed.

The reason we need to boot up is because RAM can only store information as long as it has power and ground to complete the circuits used to input, store, and output the memory. This is compared to the older long term storage devices like hard drives, and to newer storage devices like secure digital (SD) cards and other flash memory, all of which have means for permanently writing memory to spaces on them while also allowing reading, deleting, and rewriting to those spaces as needed, albeit more slowly. **RAM is like paper files within an arm's reach of a computer we're working on at the moment.**

But as is the case with many of the papers we may have sitting in close reach of the computer on our desk, they still can't be actually read and processed. To do this, a computer next moves sequences of code from dynamic RAM (DRAM), which can store larger amounts of data at a more affordable rate but that still works a bit too slowly for static RAM (SRAM), which doesn't need to be refreshed and that is faster than DRAM. SRAM serves as a system cache that works closely with the Central Processing Unit, or CPU. Consider **SRAM as those files immediately at hand**, and the **CPU as your brain, now processing the information you just read from the file** in front of you.

At any given moment, where a segment of the operating system is "living" depends on what modules, functions, variables, iterations, and sequences of code are needed. It is moving from attic to desk to hand to brain to hand to brain to desk to hand to brain to desk to attic over and over billions of times per second.

More on Boot Loaders

The computer boot loaders have gone through many stages of development. Today, the **Basic Input Output System (BIOS)** and the newer Unified Extensible Firmware Interface (UEFI) are used on personal computers, with BIOS slowly being phased out. Some still use the term BIOS, even when UEFI is now being used. Either way, the instruction sets to do self-diagnostics and to locate an operating system to which BIOS or UEFI can hand over the further booting of a computer is generally installed as an Electrically Erasable, Programmable, Read Only Memory (EEPROM) chip, sometimes also called firmware, on the main printed circuit board of the computer hardware.

Some microcomputer developers focused on mobile computing and the Internet of Things create their own unique instruction sets, which they can include on their system-on-a-chip (SoC) integrated circuit and which then look for UEFI BIOS. Others, like the Raspberry Pi SoC through version 3, do not use a conventional BIOS at all. The Raspberry Pi uses a configuration file, instead of the BIOS you would expect to find on a conventional PC. The system configuration parameters, which would traditionally be edited and stored using a BIOS, are stored instead in an optional text file named config.txt. This is read by the graphical processing unit (GPU) before the ARM CPU and Linux are initialized. It must therefore be located on the first (boot) partition of your SD card. While the Raspberry Pi 4 now includes EEPROM on the printed circuit board with UEFI BIOS to load Linux from a microSD card or a networked storage device, it also still makes use of the config.txt file to do low-level system configurations, in advance of using the CPU to load and run Linux.

The `config.txt` file is read by the early-stage boot firmware on the Raspberry Pi, so it has a very simple file format. The format is a single `property=value` statement on each line, where `value` is either an integer or a string. Comments may be added, or existing config values may be commented out and disabled, by starting a line with the # character. Raspberry Pi documentation is available on the config.txt file, where you can find an extended list of configurations settings.

The Embedded Linux Wiki, which works to "preserve and present information about the development and use of Linux in embedded systems," provides a page with instructions for editing the config.txt boot configuration file from OS X, the Raspberry Pi, and a Windows PC. After Linux has booted, you can view the current active settings using the following commands in a terminal window:

- `vcgencmd get_config <config>`: this displays a specific config value, e.g., `vcgencmd get_config arm_freq`.
- `vcgencmd get_config int`: this lists all the integer config options that are set (non-zero).
- `vcgencmd get_config str`: this lists all the string config options that are set (non-null).

You can learn more about the history of bootstrapping at Adafruit. This chapter highlights computer building blocks, but the "Bootloading Basics" guide isn't written exclusively for personal computer bootloaders, but for the general category that serves processors used on microcontrollers and microcomputers of all types. This includes the UF2 Bootloader used with the Circuit Playground Express we'll be working with in the Blue Unit.

The Kernel: The Heart of the Operating System

The operating system is the underlying program that we generally don't even know really exists. We hear about it, for instance when we buy a computer running Microsoft Windows 10, or an Apple MacBook running macOS Sierra. And once in a great while we're encouraged or required to install a new version of the operating system.

But generally we just see lots of text and graphics on a monitor, and we type on our keyboard, move around a pointer from one place to another using a mouse or touchpad or even sometimes the monitor itself. That's the user interface side. But the heart of the operating system lives in the kernel.

- The kernel controls how applications get time on the CPU(s) of the computer. There was a day when the OS only allowed one application at a time to physically run on the computer. Imagine having to save your work in your text editor, exit, then open the spreadsheet to see your data, then exit, then reopen your text editor. Today operating systems are multi-tasking, allowing multiple applications to all "run" simultaneously. They do this using a very fast time sharing strategy, in which each application is allowed to submit threads, or small segments of programming code, to the operating system, requesting execution of that thread.
- The kernel controls memory access at several levels:
    - Physical memory vs. virtual memory: Applications can easily eat up all available memory. To that extent, the OS must carefully control just how much memory is being used by a specific application. Further, when a user wants to do more than can be done with available memory, the operating system controls the movement of bits from physical memory to virtual memory or swap space, a special location on the hard drive configured to mimic, albeit very slowly, physical memory. Applications cannot directly access virtual memory, so when memory runs short, the OS spends a lot of time swapping bits between physical and virtual memory.
    - Shared vs. private address space: Some operating systems conserve physical memory allocation by only loading shared segments of code (for instance from a library file) once, then pointing all applications to this shared address space. The problem is that sometimes a badly behaving application can overwrite a critical shared code with corrupted data. In the worst cases, this can lead to a general protection fault (a type of system error). Many modern operating systems therefore decide to make greater use of private address space, always loading segments

of code again even if they already exist in memory.

- The kernel controls interactions with device controllers. Because there are many controllers manufactured for each type of input/output function (for instance, video controllers), an operating system cannot be expected to know how to interface with every one of them out of the box. Instead, each controller must come with its own device driver that can translate between the specific controller and the specific operating system. All hardware incompatibilities boil down to unavailable device drivers for the specific operating system.
- The kernel of the operating system determines which types of file systems are supported.

In all cases, it is the kernel of the operating system that does the core work of communication between electronics and programming code.

A Brief History of Operating Systems

In Microsoft and Apple operating systems, they've put together a tightly intertwined, large set of applications to cover all main aspects of the operating system, from user interface; to bundled applications to configure the computer, store, delete, and access files; to installing and updating applications; to the setup of screensavers, accounts, and passwords; to disk and monitor management, and so much more. And the operating system has the essential instruction sets needed to interact with the electronics of the computer within the kernel, the most basic level of the operating system.

Before there was Microsoft or Apple, there was Unix, a family of multitasking, multiuser computer operating systems. Unix was first developed by AT&T at Bell Labs research center in the 1970s, but then was split into a few different Unix systems. All are commercial, at least at some level, although some have been made more widely available than others. For the most part, Unix provides a distinction between the kernel of the operating system, primary bundled applications, the user interface, and any networked information systems used to pass data back and forth with devices beyond the kernel.

Very early in its development, Unix developers created the X-Window system to provide the ability to create graphical user interfaces. The developers also set up Unix to work on the Internet. In this way, even back in the 1970s and 1980s, Unix was able to have a graphical user interface to a headless computer using simple (actually called "dumb" at the time) computers, which served primarily as a text and sometimes graphical human interface device to the Unix core computer.

One company that entered into a licensing contract to use and build their own iteration of Unix was Apple, leading to the macOS.

Microsoft wrote their own text-based operating system, called MS-DOS, or Microsoft Disk Operating System. It was meant to support a personal computer (that is, only one user working only with directly attached devices), and included the kernel, basic bundled applications, and a text interface. They then added their own Windows graphical user interface extension to MS-DOS. Later still, they created a new version of the operating system called Windows NT (New Technology), which integrated the kernel, bundled applications, and graphical user interface into one, while also staying compatible with earlier versions of Windows. Ultimately, they made the break to create newer versions of the Windows operating system, and eventually also adding back a new and improved text-based terminal window. But throughout, their choices on select keyboard options, labels, and other essential communications have contrasted with those of Unix, and thereby with those of Apple as well.

Moving into the early 1990s, Unix computers were relatively expensive, and therefore were purchased by, and used in, research labs. Then, in 1991, University of Helsinki student Linus Torvalds purchased an Intel 80386-based clone of the IBM PC mini-Unix called MINIX. Later that year, he made public his first prototype of a Unix-like kernel operating system he called Linux. On March 14, 1994, he released version 1.0. From the start, he released Linux using Richard Stallman's General Public License. Stallman, a free software advocate, created the GNU Project, a software development initiative. By combining the GNU Project essential bundled applications with the Linux core, Torvalds built and released a fully functioning operating system.

Over the years, different groups have worked to create distributions of specific GNU applications, other applications, and a Linux core. The oldest surviving distribution of Linux is called Slackware, created by Patrick Volkerding in 1992. Slackware 1.0 came on twenty-four floppy disks and was built on top of Linux kernel version 0.99. I started working in the Neuronal Pattern Analysis Group of the Beckman Institute at the University of Illinois Urbana-Champaign in 1993, and soon after began using Slackware and Linux, along with the newly developing X-Mosaic graphical web browser. I used these as part of a team seeking to use the Internet to make publicly available raw electrophysiology data, which was collected as part of various research projects.

Two common distributions of Linux are used as the starting point for a rich variety of child distributions. One is called Fedora, and originated in 1995, under the name Red Hat Linux. In 2003, the Red Hat trademark was used to create two branches: Red Hat Enterprise, with a combination of free and open-source applications and commercial applications, and Fedora, a community-oriented distribution designed for "hobbyists." Commonly used distributions with Fedora at their core include CentOS and Scientific Linux.

Debian GNU/Linux was first announced in 1993 by its founder, Ian Murdock. It has been developed to every extent possible as a completely noncommercial project, through collaboration with over one thousand volunteer developers. Widely used Debian-based alternatives include the overall most widely used distribution, Ubuntu, and Raspberry Pi OS, the default Debian-based alternative for the Raspberry Pi microcomputer.

There are two widely used commercial operating systems based on the Linux kernel today as well. In September 2008, Google launched its Android operating system using a modified version of the Linux kernel and various open-source software designed for touchscreen mobile devices, and subsequently developed for Android TV, Android Auto, and Wear OS. The Android Open Source Project is used for development of its core source code, and apps for Android can therefore be developed by anyone, for virtually any purpose, and installed and run on Android devices. In May 2011, Google released the first Chromebook, running the Chrome OS. It is only available preinstalled on hardware from Google manufacturing partners. Its open-source equivalent, Chromium OS, can be freely installed.

Specific to Raspberry Pi OS, each major new release of Debian results in a major new release of Raspberry Pi OS. The code name used for each major release of Debian is taken from *Toy Story* characters. In 2013, this was Wheezy, the penguin character by that name. In 2015, this was Jessie, the cowgirl character. As of June 2017, this was Stretch, the rubber octopus from *Toy Story 3*. Edition two of the textbook was written using the April 2022 release of Raspberry Pi OS, which uses Debian 11 Bullseye, the horse character from *Toy Story*. As noted in these code name release dates, Debian works to be stable through a conservative update process. The stable release is held for about two years, although test releases become available via the new code name during the interim, for those wanting early access for use and testing of new designs.

Take a few minutes to explore a bit further the core philosophy and principles behind Debian. How might they compare with those of the operating system used for your own computer? Why might the Raspberry Pi Foundation have chosen Debian GNU/Linux instead of other versions of Linux as the starting platform for their own distribution of GNU/Linux?

- Debian: Our Philosophy: Why We Do It and How We Do It
    - Constitution for the Debian Project (v1.9)
    - Debian: What Does Free Mean?
    - GNU: What is Free Software? and the reasoning for identification as GNU/Linux
- Reasons to Use Debian

Exercise: Reviewing Your Computer

Take some time to dig into the hardware, operating system, and major installed software applications of your primary computer. While it's often more difficult, if even possible, to open up your computer to look at the system or motherboard of the computer than it used to be, if you can safely do that, do so. You may have received some default documentation with your computer, either in print or digital. There may be "About this Computer" links available on the computer or on a vendor's website. Some resources are best accessed through applications that came with your computer and that provide current notes which can account for potential updates that have since replaced the defaults that came with the computer.

Using these resources, create your own list of the physical components and operating system of your computer:

- Input/output devices, such as types of USB ports, Thunderbolt ports, audio, camera, Ethernet, Bluetooth, card readers, etc.

- Type and amount of random access memory (RAM)
- Storage devices
- Central processing unit (CPU) and, if included, graphic processing unit (GPU)
- Power source
- Operating system (OS) type and version.subversion/edition
- Windowing system, desktop environment, and themes/skins. For many computers today, these are highly integrated, so you may only see this as a singular graphical user interface system.

Take time to compare and contrast this with some of the other desktop and laptop computers that you or others around you use on a fairly regular basis.

Wrap Up

Computers, microcomputers, and microcontrollers are an everyday part of life. The core components of these devices have remained the same, even as the specifications, standards, and physical and software implementations have been shaped and reshaped repeatedly over the last fifty-plus years. Knowing at least a bit about the items within all devices is essential as we serve professional and personal roles in the selection, implementation, innovation-in-use, and support of these devices in our varying contexts. Crash Course Computer Science is a highly recommended resource that further introduces the basic hardware and logic building blocks of computing in its first ten episodes. We'll also bring a few of these, plus some of the later Crash Course episodes specifically, into sections of the Blue and Rainbow Units.

Before we move on to examining the Raspberry Pi microcomputer, take a few moments to watch this "teardown" video from electronics repair company iFixit. A product teardown is a common method for identifying the component parts of an electronic device, in order to better refurbish or repair devices of that same model. However, take caution and browse the resources available from iFixit before tearing down your own device, as the manufacturer may claim that such actions void your warranty. Moreover, for many newer devices, components are glued—not screwed—together, requiring specialized equipment to get inside the device.

In this teardown video, a 2018 MacBook Air is disassembled and evaluated. As you watch, make note of any parts you recognize. For even more detail, read the full teardown guide.

Comprehension Check

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with answers, see the online version.]

1. Fill in the following chart with the following terms: **Breadboard**, **Case**, **Central Processing Unit (CPU)**, **Controller**, **Ethernet Cable**, **Expansion Card**, **Flash Memory**, **Input/Output Devices**, **Magnetic Media**, **Motherboard**, **Optical Media**, **Power Supply**, **Random Access Memory**. *Note that terms may be used once or not at all.*

Definition	Term
Relatively fast mechanism constructed from integrated circuits that temporarily holds data needed by the processor and that requires electricity to work.	
The part of the computer that protects the main components of the computer from damage while also providing a means for cooling through airflow.	
A relatively slow mechanism for archiving data using a laser to detect the presence or absence of reflected light, allowing it to store data even when no electricity is available.	
The integrated circuit of a computer constructed to act as the main thinker and overall central controller for the computer.	
The core part of the computer with printed electrical conductors allowing it to interconnect the different computer components.	
Keyboards, mice, GPIO, monitors, projectors, scanners, and printers are all examples of this part of the computer.	
An integrated circuit used for data storage, although relatively slowly. Unlike other integrated circuit memory, it can store data even when no electricity is available.	
A board with electronic circuits that can be connected directly to the main board of the computer, allowing for additional capabilities beyond those already built into the main system board.	
An example of a relatively slow mechanism for archiving data using positive and negative charges, allowing it to store data even when no electricity is available.	
The integrated circuits of a computer used as managers of data flows between inputs and processors, processors and outputs.	

4A: Storytelling in the Information Sciences

Yingying Han and Martin Wolske

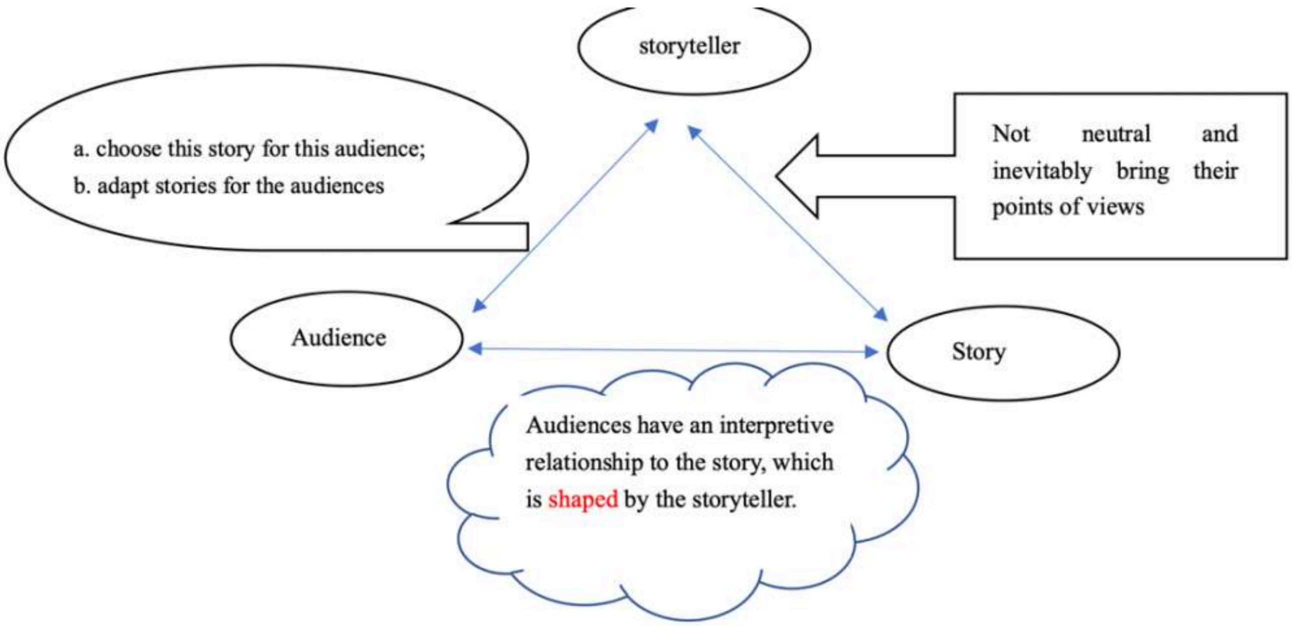
Storytelling's Long History in the Information Sciences

Many of us may have the childhood experience of sleeping while a family member tells us a bedtime story. But if we look more closely, we'll find that storytelling is a ubiquitous practice in our daily lives. The goal of storytelling is "to be able to create a story, to make it live during the moment of telling, to arouse emotions—wonder, laughter, joy, amazement"[1].

Storytelling has a rich history in the library and information sciences (LIS), dating back to the 1890s when it was established as a practice in youth services librarianship[2]. For example, the Urbana Free Library, a public library east of the University of Illinois Urbana-Champaign campus, hosts regular storytelling sessions for children, attracting many local families. The Urbana Free Library also collaborates with the East Center for East Asian and Pacific Studies at the University to host East Asian story time, providing a valuable resource for young people interested in East Asian culture.

In her paper "Storytelling wisdom: story, information, and DIKW," Kate McDowell notes that "LIS storytelling emphasizes a triangular relationship between teller, audience, and story, a dynamic process that co-creates a particular telling of a story." At its best, it is a mutual creation as the three relationships inform each other. The dynamic relationship has three main elements: "For storytelling to occur, there must first be a basic relationship of *trust between the teller and the audience*. [...] Second, the *teller has a relationship to the story*, whether as creator or reteller. [...] Third, the *audience has an interpretive relationship to the story*" (p. 1224, emphasis by author)."

1. Sawyer, Ruth. *The Way of the Storyteller*. New York: The Viking Press, 1942.
2. McDowell, Kate. "Storytelling wisdom: Story, information, and DIKW." *Journal of the Association for Information Science and Technology*, 72, no. 10 (2021): 1223–1233, https://doi.org/10.1002/asi.24466.



Chapter author Yingying Han's representation of the dynamic of storytelling described by McDowell.

As we explored in the Orange Unit 3A: The Unknown Tech Innovators, there are a range of unknown innovators of our everyday technologies. In some cases, this may be by choice of the innovator to not tell their story, as such effort isn't part of their community cultural wealth, or maybe because such effort would compete with their time dedicated to other purposes. But as Bruce Sinclair notes, the stories regarding innovators and innovations often prioritize the work of engineering over that of crafting[3]. Carolyn de la Peña further highlights how the papers and records of such innovation stories are often restricted to those of white men[4]. As we look across the dynamic triangle and consider its three main elements, it is essential for us to add this further lens that inevitably shapes the story, storyteller, and audience in a variety of ways.

Stories transform in many different ways depending on when and where they are told, whether at a fireplace or in cyberspace, to a young person or to a senior. Storytelling serves an essential role in its relationship to story and to audience. Strategically reaching target audiences in ways that encourage them to move from the role of receiver to that of collaborator within a collective meaning-making process is then a way to counter the dominant narrative that Sinclair and de la Peña describe.

> Storytelling polishes stories like editing polishes essays, with the audience serving as editor. Changes add collective understandings to the story by emphasizing particular words, characters, actions, and more. Audiences interpret stories as groups, socially constructing meanings. Retellings demonstrate how a story can be told so that others also recall and retell, and the story keeps traveling. "Stories live everywhere, but they rarely stay in one place"

3. Sinclair, Bruce. "Integrating the Histories of Race and Technology." In *Technology and the African-American Experience: Needs and Opportunities for Study*. Boston: MIT Press, 2004. https://mitpress.mit.edu/9780262195041/.
4. Peña, Carolyn de la. "The History of Technology, the Resistance of Archives, and the Whiteness of Race." *Technology and Culture* 51, no. 4 (October 2010): 919–37, https://doi.org/10.1353/tech.2010.0064.

> (Hearne, 1997).[5] Storytelling entails dynamic collective interpretation, both synchronous and across long spans of time. So what is a story?
>
> Story is here defined in two ways: structurally as narratively patterned information, and functionally as the content shared through the narrative experience of storytelling.
>
> McDowell, p. 1225[6].

The Data, Information, Knowledge, Wisdom Pyramid

At the start of the Orange Unit 1A: Information Systems, a connection was made between data, information, knowledge, and wisdom, the pyramid of which "has proved very useful in the information sciences as a means of investigating how different aspects of this pyramid have influenced the shaping of individuals, communities, societies, organizations, and governing bodies." This can also be incorporated specifically into storytelling concepts, which we'll explore in a minute.

But first, let's pause to further clarify the definitions of Data, Information, Knowledge, and Wisdom (DIKW) within the pyramid.

According to Merriam-Webster, *data* is information in digital form that can be transmitted or processed. For example, if I ask you what the following list of numbers means—01001001 01010011 00100000 00110100 00110000 00110001 00100000 01010010 01101111 01100011 01101011 01110011 00100001—you probably will be confused for a while. What if you were given more context, being told this is a binary number that can be translated into English texts and Roman numerals through an ASCII table? You can use either an ASCII table or use a Binary-Text translator (for example, Convert Binary) to find that the 13 sets of 8-bit data are translated into the text "IS 401 Rocks!", where 01001001 = the capital I, 01010011 = the capital S, 00100000 = a space character, etc.

In her article "The wisdom hierarchy: representations of the DIKW hierarchy," Rowley refers to *information* as "organized or structured data which has been processed in such a way that the information now has relevance for a specific purpose of context and is therefore meaningful, valuable, useful and relevant."[7]" People may extract different meanings from the information "IS 401 Rocks!" depending on the context.

5. Hearne, Betsy. "Story, from fireplace to cyberspace: Connecting children and narrative." In B. G. Hearne (Ed.), *Allerton Park Institute 39th* (p. 143). Urbana-Champaign, IL: University of Illinois Graduate School of Library and Information Science, 1997.
6. McDowell, Kate. "Storytelling wisdom: Story, information, and DIKW." *Journal of the Association for Information Science and Technology*, 72, no. 10 (2021): 1223–1233, https://doi.org/10.1002/asi.24466.
7. Rowley, Jennifer. "The wisdom hierarchy: Representations of the DIKW hierarchy." *Journal of Information Science*, 33, no. 2 (2007): 163–180, https://doi.org/10.1177/0165551506070706.

Starting with the first two text elements, look up IS in Merriam-Webster, and you'll find these two letters in capitals means "Information System." But this is not necessarily what the creator of the binary numbers meant. For one major target audience of this textbook, however—students at the School of Information Sciences, the iSchool at Illinois—"IS 401" is the course number for the course Introduction to Network Information Systems. Until knowing this context, it might be hard to truly move the binary data to text and then on to information.

Turning to the third element within the text, many English speakers within the United States would hopefully recognize "Rocks" in this context not as a statement regarding the relationship between this course and a solid mineral matter, but rather as a proposed statement of excitement regarding the course IS 401. For people for whom English is not their primary language, or who understand the language within different social and cultural contexts, "IS 401 Rocks!" is not meaningful information because they will not be able to interpret it.

Take a moment to consider how two different students, one a third-generation U.S. citizen and one from Asia for whom English is their second language, would interpret the answer "IS 401 Rocks!" to their question, "How was IS 401 when you took it?" If understood to be a course that is exciting to take, the information becomes "actionable information," that is, *knowledge,* that the course should be something to consider registering to take. But for others the response may be confusing or misleading, and may result in knowledge regarding a low-quality course for which it would be wrong to take. Here we can see that travel along the DIKW pyramid can take different paths. As McDowell argues, in "uncertain, conflictual, or paradoxical situations, knowledge of how to accomplish tasks is not enough. We need wisdom."[8]

Defining *wisdom* is hard. According to Rowley[9], wisdom is accumulated knowledge that allows you to understand how to apply concepts from one domain to new situations or programs. For McDowell, wisdom may be similar to being a storyteller whose action occurs in context. "[W]isdom may be socially constructed and enacted situationally. If wisdom is enacted, then it can be the province of groups as well as individuals."[10] LIS scholars have long been interested in studying the wisdom of collective information sharing, for example when using Information Communication Technologies (ICTs) to share collective information during a crisis.

So, where does the story fit in this pyramid? Below is the Story-DIKW framework proposed by Kate McDowell:

- S-Data: Ability to identify and interpret data from which information emerges that can be communicated in story.
- S-Information: Ability to inform audiences by communicating data with context

8. p. 1227, McDowell, Kate. "Storytelling wisdom: Story, information, and DIKW."
9. Rowley, Jennifer. "The wisdom hierarchy: Representations of the DIKW hierarchy."
10. p. 1229, McDowell, Kate. "Storytelling wisdom: Story, information, and DIKW."

as story, in both form and narrative experience.

- S-Knowledge: Ability to convey knowledge as complex actionable information through the construction and telling of a story, incorporating cultural and contextual cues. S-knowledge is shared frequently in innovative or experimental contexts.
- S-Wisdom: Ability to know which story to tell—including when, how, and to whom—in order to convey wisdom.

Take a few minutes to reflect back on your journey through the Orange Unit of this textbook. In what ways, if any, might it fit within the dynamic triangle of story, teller, and audience? In what ways, if any, might various aspects fit within the data, information, knowledge, and wisdom pyramid of the S-DIKW framework?

Social Justice Storytelling: Why It Matters

Memory institutions such as libraries and archives in the United States have a long history of excluding and erasing cultures of people who are marginalized.[11][12] For example, Sutherland[13] argues that by failing to create an official record of human rights abuses in the United States, American archivists have continued to perpetrate violence against Black Americans by failing to tell Black Americans' stories. This century-long refusal to engage justice for Black Americans endures in present-day through archival professions in the United States.

Digital storytelling is a technique that allows the story to be retold even while the storyteller is sleeping. This technique also allows the storyteller to use additional media to better bring home the moment. It is an innovative, community-based, participatory research technique bringing together community members and health workers to approach, research, and collaboratively address local health issues.[14] It has been incorporated into traditional quilting, along with capacitive touch sensors, to create living archives of voices in the sex work industry.[15] It has played a role in co-design initiatives

11. Caswell, Michelle. "Seeing yourself in history: Community archives and the fight against symbolic annihilation." *The Public Historian*, 36, no. 4 (2014): 26–37, https://doi.org/10.1525/tph.2014.36.4.26.
12. Caswell, Michelle., Marika Cifor, & Mario H. Ramirez. "'To suddenly discover yourself existing': Uncovering the impact of community archives." *The American Archivist*, 79, no. 1 (2016): 56–81, https://doi.org/10.17723/0360-9081.79.1.56.
13. Sutherland, Tonia. "Archival amnesty: In search of Black American transitional and restorative justice." *Journal of Critical Library and Information Studies*, 1, no. 2 (2017), https://doi.org/10.24242/jclis.v1i2.42.
14. Gubrium, Aline. "Digital Storytelling: An Emergent Method for Health Promotion Research and Practice," *Health Promotion Practice* 10, no. 2 (2009): 186–91. https://doi.org/10.1177/1524839909332600.
15. Strohmayer, Angelika and Janis Meissner. "'We Had Tough Times, but We've Sort of Sewn Our Way through It': The Partnership Quilt," *XRDS: Crossroads, The ACM Magazine for Students* 24, no. 2 (December 19, 2017): 48–51. https://doi.org/10.1145/3155128.

involving local underserved communities in data generation, and in spurring grassroots initiatives and encouraging broader local participation and engagement in community-based co-design projects.[16]

Social justice storytelling is an essential approach emerging within the Information Sciences as the profession works to give voice to people who are marginalized. Communicating in culturally competent ways about social justice as information professionals requires that training in this area be brought into the classroom as well[17]. Counterstorytelling is one of the primary tenets of critical race theory[18][19][20] and involves telling stories that "buck the status quo and challenge the long held collective stories of a hegemonic society, which tacitly maintain the narratives and normative behavior of dominant groups."[21] Counterstorytelling is thus proposed as a key approach to social justice storytelling in the information sciences generally,[22] and in our work through this textbook as a community of practice seeking to bring a more holistic and nuanced understanding of sociotechnical artifacts we use as a daily part of our professional lives specifically.

For the first edition of *A Person-Centered Guide to Demystifying Technology*, Cooke's chapter "Becoming New Storytellers: Counterstorytelling in LIS"[23] was foundational in guiding this final session of the Orange Unit. McDowell's writing on storytelling and DIKW, including her article co-authored with Cooke on social justice storytelling, were published between editions of this textbook and strongly build off of Cooke's previous foundational work. For Cooke, becoming a "new storyteller" is a call for us to take on an active information professional leadership role, facilitating discussions on issues of race, privilege, social justice, and other necessary and difficult issues. Becoming new storytellers has provided further clarity and a more solid foundation for the many different participatory action research and course offerings that have now come together to create *A Person-Centered Guide to Demystifying Technology*. The previous sessions of the Orange Unit are thus not separate from this social chapter of session four, but serve as precursors to this chapter. And this final session of the Orange Unit itself serves as not only a conclusion of this first unit, but also as the base for the Blue and

16. Lorini, Maria Rosa, Amalia Sabiescu, and Nemanja Memarovic, "Collective Digital Storytelling in Community-Based Co-Design Projects: An Emergent Approach," *The Journal of Community Informatics* 13, no. 1 (2017): 109–36, https://doi.org/10.15353/joci.v13i1.3296.
17. McDowell, Kate and Nicole A. Cooke. "Social Justice Storytelling: A Pedagogical Imperative." *The Library Quarterly: Information, Community, Policy*, 92, no. 4 (2022). https://doi.org/10.1086/721391
18. Delgado, Richard and Jean Stefancic. *Critical Race Theory: An Introduction*. New York University Press, 2023.
19. Ladson-Billings, Gloria. Critical Race Theory—What It Is Not! In *Handbook of Critical Race Theory in Education* (pp. 32–43). Routledge, 2021.
20. Solórzano, Daniel G., and Tara J. Yosso. "Critical race methodology: Counter-storytelling as an analytical framework for education research." *Qualitative Inquiry*, 8, no. 1 (2002), 23–44, https://doi.org/10.1177/107780040200800103.
21. Cooke, Nicole. "Becoming New Storytellers: Counterstorytelling in LIS." In *Information Services to Diverse Populations: Developing Culturally Competent Library Professionals*, 113–36. Santa Barbara, CA: Libraries Unlimited, 2017.
22. McDowell, Kate and Nicole A. Cooke. "Social Justice Storytelling: A Pedagogical Imperative." *The Library Quarterly: Information, Community, Policy*, 92, no. 4 (2022). https://doi.org/10.1086/721391.
23. Cooke, Nicole. "Becoming New Storytellers: Counterstorytelling in LIS." In *Information Services to Diverse Populations: Developing Culturally Competent Library Professionals*, 113–36. Santa Barbara, CA: Libraries Unlimited, 2017.

Rainbow Units, as each seeks to facilitate a collective journey to becoming new storytellers within the Information Sciences.

Digital technologies old and new are not objects that can be packed inside a box. They are a seamless, indivisible combination of people, organizations, policies, economies, histories, cultures, knowledge, and material things that are continuously shaped and reshaped. Every one of us innovates-in-use our everyday technologies; we just do not always know it. Not only are we shaped by the networked information tools in our midst; we also shape them and thereby shape others. For us to advance the individual agency of all drawing upon diverse community knowledge and cultural wealth within the fabric of communities, we need to nurture our cognitive, socio-emotional, information, and progressive community engagement skills, along with, and sometimes in advance of, our technical skills, which then serve as just-in-time in-fill learning. This is the call placed by Rev. Dr. Martin Luther King, Jr.—to rapidly shift from a "thing-oriented" society to a "person-oriented" society.[24]

One step in this, then, is to become new storytellers regarding sociotechnical artifacts and systems, including within a broadened conceptualization of networked information systems. **Stock stories** abound, bringing us into dominant narratives around the importance of technologies in changing society. It removes from the story the centrality of people, race, privilege, and power, which are used to shape the technology and its narrative. In this removal, many lose individual sociotechnical-related agency, and in addition often lose aspects of their human rights. How we buck the status quo through counterstories varies as we work to assure the right story, at the right time, in the right way, using a dynamic storytelling triangle. Forms of counterstory include: a **concealed story** that serves as a direct response to stock stories, a **resistance story** that not only bucks against stock stories but also highlights injustices, or an **emerging/transforming story** that (re)constructs knowledge built on concealed and resistance stories.[25]

Lesson Plan

We will continue to develop the full range of our technical, information, cognitive, socio-emotional, and application skills moving forward. But as has been noted throughout, to advance a person-centered and social-justice-oriented approach to demystifying technology, we also need to actively bring in progressive community [inter]action and critical social and technical skills to these other five points. Storytelling for social justice is meant to serve as an active way to bring each of these together, starting with this session. At the same time, we will continue to advance various others of our skillsets as we Meet the Microcomputer and Get Started with the Raspberry Pi, before moving on to explore

24. pg. 157, King, Martin Luther. "Beyond Vietnam." In: *A Call to Conscience*, edited by Carson, Clayborne and Kris Shepard. Grand Central Publishing, 2001.
25. Cooke, Nicole. "Becoming New Storytellers: Counterstorytelling in LIS." In *Information Services to Diverse Populations: Developing Culturally Competent Library Professionals*, 113–36. Santa Barbara, CA: Libraries Unlimited, 2017.

Coding Electronics. It is hoped that these will facilitate a solid step forward as we work to evolve a more holistic and nuanced understanding of the sociotechnical artifacts we use as a daily part of our professional lives.

Storytelling for Social Justice as Part of Demystifying Technology

(The following is a modification of the assignment as described in Kate McDowell and Nicole A. Cooke (2021). Social Justice Storytelling: A Pedagogical Imperative. The Library Quarterly: Information, Community, Policy, volume 92, number 4, October 2022.)

There are many news stories on social justice topics regarding the relationships of women, people of color, or other minorities have to and with technology. These involve the relative access to resources, opportunities, and privileges of individuals or groups within a society. This story should be designed to persuade your audience of the importance of a social justice issue, based on your knowledge of the issue (acquired through research), which can be easily recognized and referenced by your audiences, along with a healthy dose of emotional appeal.

Pick a current or historical social justice topic, from any part of the world, for which you can find some histories of people and technology. These may be stories about the hidden innovators; the ways people have used technology as means of marginalizing others; the ways people have used technology to provide agency for liberation and transformation; etc. Your story should connect with the narratives of individuals or groups whose lives have been impacted by injustice or whose work addressing an injustice has been untold or hidden.

Prepare for your storytelling by finding stories, learning the events in them so that you can tell them spontaneously, and making them your own by adapting the story you've found to your own voice and telling style. The previous units of the Orange Unit have already worked to bring aspects forward and will have been captured in part through your Reading Notes and Professional Journal Reflections. Hopefully these have also been points of discussion within our community of practice, as you work to address questions that have arisen. These issues will continue to be considered within the sessions of the Blue and Rainbow Units and should continue to be an active component of your community of inquiry as you collaboratively work through them. Consider practicing spontaneous storytelling through activities within your community of practice moving forward, to move from an early draft to later drafts and the final polishing of your stories.

The recordings of your main story (1-2 minutes), an introduction of the storyteller and the story (15 seconds), and a "for more on the story" component (around 30 seconds) will be used in the concluding exercise of Blue Unit Session 4B: Raspberry Pi Counterstory Little Free Library, where we'll bring together the Raspberry Pi microcomputer and Circuit Playground Express microcontroller. Optional but not required, you can also include a fourth "behind the scenes" story. Make sure you have open access to your three recordings so that you can transfer them to the Raspberry Pi and run them using software that comes with the Raspberry Pi OS.

Before recording, you can also work with rehearsal groups within your community of practice to exchange, critique, and improve each other's stories in advance of implementation on the Raspberry Pi. To create a compelling story, use the following guidelines. Your group should use these guidelines to ask questions when you exchange feedback:

- Have a clear beginning, middle, and ending narrative structure.
- Focus on conveying a meaningful adaptation and interpretation of the social justice issue.
- Create a narrative that demonstrates the importance of a social justice issue.

Essential Resources:

- McDowell, K. "Storytelling wisdom: Story, information, and DIKW." *Journal of the Association for Information Science and Technology, 72* (10) (2021): 1223–1233. https://doi.org/10.1002/asi.24466
- Lorini, Maria Rosa, Amalia Sabiescu, and Nemanja Memarovic. "Collective Digital Storytelling in Community-Based Co-Design Projects: An Emergent Approach." *The Journal of Community Informatics* 13, no. 1 (2017): 109–36. https://doi.org/10.15353/joci.v13i1.3296.

Additional Resources:

- Cooke, Nicole. "Becoming New Storytellers: Counterstorytelling in LIS." In *Information Services to Diverse Populations: Developing Culturally Competent Library Professionals*, 113–36. Santa Barbara, CA: Libraries Unlimited, 2017.
- StoryCorps is a nonprofit with a mission "to preserve and share humanity's stories in order to build connections between people and create a more just and compassionate world."
    - StoryCorps. "The Great Thanksgiving Listen." Accessed February 10, 2020. https://storycorps.org/discover/the-great-thanksgiving-listen/.
    - StoryCorps. "StoryCorps DIY." Accessed February 10, 2020. https://storycorps.org/participate/storycorps-diy/.
- Wolske, Martin. "Digital Storytelling Workshop." Presented at the Faculty Summer Institute, University of Illinois at Urbana-Champaign, 2011. https://uofi.box.com/s/77jgdy33acn6ezsllqh87472xnux9d56.
- Gubrium, Aline. "Digital Storytelling: An Emergent Method for Health Promotion Research and Practice." *Health Promotion Practice* 10, no. 2 (2009): 186–91. https://doi.org/10.1177/1524839909332600.
- Crane, Beverly. "Digital Storytelling Changes the Way We Write Stories." *Information Searcher* 18, no. 1 (2008): 3–9, 35.

- Drotner, Kirsten. "Boundaries and Bridges: Digital Storytelling in Education Studies and Media Studies." In *Digital Storytelling, Mediatized Stories: Self-Representations in New Media*, edited by Knut Lundby, 61–84. New York: Peter Lang, 2008.

Key Technical Terms

- Core hardware components of a computer, including the system board/motherboard, the central processing unit and other controllers, types of short-term memory and longer-term storage, and input/output devices.
    - How the above is implemented in practice within a system-on-a-chip microcomputer, using the Raspberry Pi as an example.
- Core software components of a computer operating system, including the kernel, window system, desktop environment, and top-level themes and skins.
    - How the above is implemented in practice within the Raspberry Pi OS, a distribution based on the underlying Debian GNU/Linux operating system.

Professional Journal Reflections:

Dedicate some time to using your Professional Journal Reflections forum to post two or three short, partial drafts of counterstories that you can share with others. You might draft the same story told to different audiences, different stories told to the same audience, or other mixes as you see fit. Drafts should rough out:

- How the story would build towards the key moment
- Introduction of location, characters, and questions/conflicts/challenges
- Main course of events
- Conclusion revealing what happened and its meaning

4B: Meet the Microcomputer

Background Knowledge Probe

- While looking through the list of toolkit items found in this textbook's Introduction, what was the first thing that came to mind when you saw it included a Raspberry Pi? What inspired those thoughts regarding the Pi?
- When you now compare and contrast a Raspberry Pi with your daily use computers, what are your current thoughts? Why have or haven't they changed?
- What design choices and design objectives might have inspired the Raspberry Pi's layout function, look, and feel?

Technical Introduction

So far, each of our electronic circuits has been physically constructed using electronic components and electrical conductors, along with ground and 3.3 and 5 volt power sources. Many of our everyday technologies are built in just this way. Think of the essential lights, fans, and clocks around us. Even those with controllers to dim the lights, change the speed of a fan, or selectively tick-tock once a second are often made exclusively with electronic components and conductors and no more. But others also include programming code to add in features. Consider, for instance, a light connected to a remote timer allowing you to turn it on or off from your phone or laptop. In this chapter, we're going to perform an upgrade to our Raspberry Pi. So far, it has performed as a helpful intermediate source for voltage and ground to our breadboard. Now, we'll use it as a **microcomputer**. Microcomputers are simply small computers that contain a microprocessor as its central processor. We'll then install and run an application written using the Python **programming language**, designed to work with the GPIO to turn LEDs on and off based on the current status of different momentary **switches**. As noted in the following video from the Raspberry Pi Foundation, "back in the 80s, kids had to learn how to code computers to use them. And as a result, these kids grew up with an inbuilt understanding of how computers worked."[1] The Raspberry Pi microcomputer has been built from the ground up to "reignite this spark." Not only is it being used specifically for information and communications technologies (ICT), it is serving a wide range of cross-curricular applications. And as illustrated, we'll use it to push a button to turn on a light as the final exercise in this chapter!

[Watch the "What is a Raspberry Pi?" video online.]

As an aside, I was one of the fortunate kids who, back in the mid-70s, began learning how to code computers to use them to do things I valued. In the following video, I highlight some of this history, recounting my work with a teletype machine and acoustical modem to connect to a server, a punch card interface with that server, and programming with a Commodore 64 desktop computer. Watch the video online.

[Watch the video "My History with Computers: 1976-1985" online.]

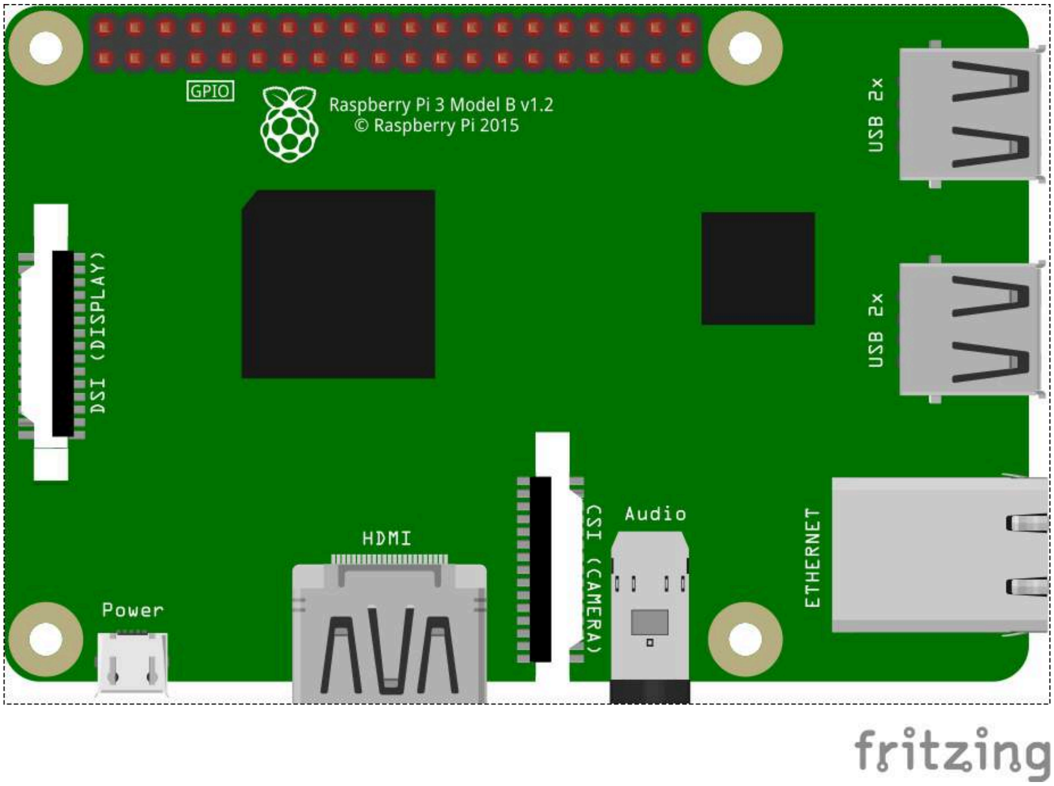
The Components of the Raspberry Pi

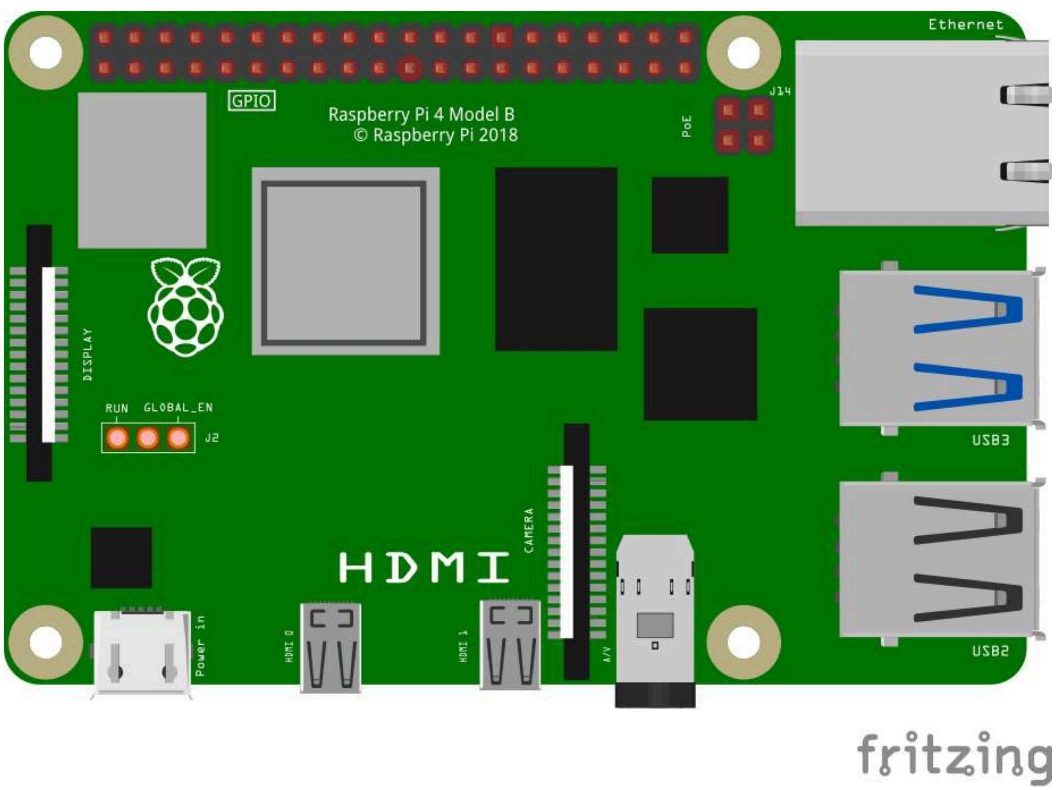
The Raspberry Pi computer used throughout this book is built using a reduced instruction set computing (RISC) architecture known as ARM. This and similar architectures have slowly taken over the computer world, allowing us to do more for less. This is because RISC-based architectures require fewer **transistors** than complex instruction set computing (CISC) architectures, such as those used in most desktop and laptop computers.

As a result, this has also opened up new computing opportunities, such as systems-on-chips (SoC) that bring together memory, input/output controllers, processors, radios, etc., into a single integrated circuit. The outcome is typically the lowering of costs, power consumption, heat dissipation, and potentially environmental impacts.

Below you can see Fritzing diagrams of both the Raspberry Pi 3 and Raspberry Pi 4 computers.

1. Raspberry Pi learning resources and documentation are licensed under CC-BY-SA 4.0.





Start by taking a few minutes to look over the hardware list for the Raspberry Pi 3 Model B, the earliest model of the third-generation Raspberry Pi.[2] This includes:

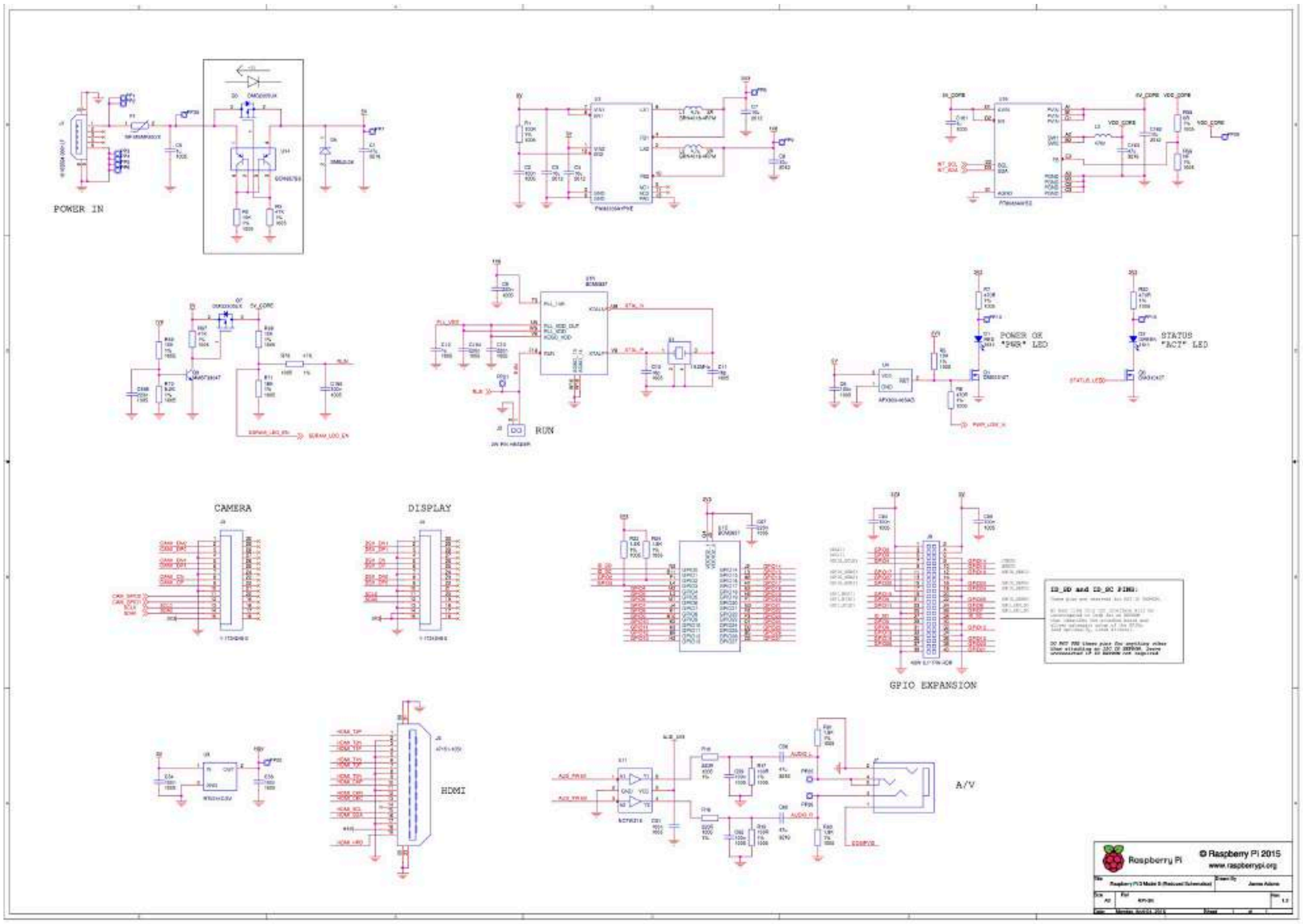
- A Broadcom BCM2837 system-on-chip (the largest of the black, square integrated circuits

2. Fritzing breadboard graphics are licensed under CC-BY-SA 3.0.

on the printed circuit board). Built into this SoC is:
    - Quad-core 1.2 GHz 64-bit ARM cortex A53 CPU
    - Full HD 1080p, H.264 Video Encode/Decode, 400MHz VideoCore IV Graphical Co-Processor Unit
    - 1 GB SDRAM
    - 512 KB cache memory
- One of the many smaller integrated circuits is a Broadcom BCM43438 chip that provides:
    - Wireless LAN
    - Bluetooth 4.1
- Next to pin 1 of the GPIO, just above the “Made In” label, is the chip antenna for wireless LAN and Bluetooth signals.
- Just off from the USB ports, you’ll find the SMSC LAN9514 chip, providing the controller for 10/100 Ethernet connectivity and four USB channels.
- A variety of I/O connectors, sockets, and ports, including:
    - 4 USB 2 ports
    - Full size HDMI port
    - Micro SD port for loading an operating system and storing data
    - 10/100 Ethernet port
    - 4 pole stereo output and composite video port
    - 40-pin extended GPIO connector
    - Camera Serial Interface Type 2 (CSI) camera port, used to connect CSI-type cameras with the SoC chip
    - Display Serial Interface (DSI) port, used with a range of LCD and similar mobile display technologies

From here, take a few more minutes at the Raspberry Pi Products website, using the “More Info >” link for one of each of the Raspberry Pi 3 Model B+, Model 3 A+, 4 Model B, 400, and Zero W products now available.

- In what ways are these the same as the Raspberry Pi 3? In what ways are these unique from the Raspberry Pi 3?
- In what contexts might one be of advantage over the others? In what contexts might one be less than ideal?



A reduced schematic of the Raspberry Pi 3 Model B highlighting the main circuits. [Raspberry Pi, 2015](). [CC BY-SA 4.0 International]()

The Raspberry Pi OS

The Raspberry Pi Project has developed its own distribution of Debian's GNU/Linux OS which it calls the Raspberry Pi OS. Debian strives to be a universal, completely free operating system, where free is not used as a monetary term but rather one of freedom. The Debian community has a written constitution that includes a social contract, diversity statement, and code of conduct, all of which are available from within the page introducing their philosophy. As a distribution based on Debian, the social philosophy of the community and technical changes to the software both eventually trickle over to the Raspberry Pi OS. Take some time to first review various social and technical aspects of Debian's community and operating system from their homepage, then make your way over to the Raspberry Pi Foundation and the Raspberry Pi Project sites to learn more about the Raspberry Pi community and operating system.

- In what ways might the vision, goals, and overarching philosophy of the Debian and Raspberry Pi communities have some parallels? What might be some ways that these could come into conflict?

- In what ways does the Raspberry Pi OS take advantage of the technical aspects of the Debian OS? What are some things the Raspberry Pi add within its own distribution to especially serve its target audience(s)?

Exercise: Reviewing the Raspberry Pi Computer

Using these resources, create your own list of the physical components and operating system of the Raspberry Pi:

- Input/output devices, such as types of USB ports, Thunderbolt ports, audio, camera, Ethernet, Bluetooth, card readers, etc.
- Type and amount of random-access memory (RAM)
- Storage devices
- Central processing unit (CPU) and, if included, graphic processing unit (GPU)
- Power source
- Operating system (OS) type and version/subversion edition
- Windowing system, desktop environment, and themes/skins. For many computers today, these are highly integrated, so you may only see this as a singular graphical user interface system.

If you can, boot up a working Raspberry Pi computer and have it running side-by-side with your own personal computer. Take a few moments to compare and contrast some of the installed applications on the Raspberry Pi (e.g., web browser, word processor, games) with similar ones on your laptop. As you do this, consider:

1. In what ways is a given software on my laptop a better application based on specific design specifications?
2. In what ways is a given software on my laptop a better application because of the hardware of my laptop?
3. In what ways is a given software on my laptop a better application because of my familiarity with its design?
4. In what ways might my answers to the above questions be shaped by the underlying vision, goals, and philosophy of the designers and distributors of a given software and of the underlying operating system on which it runs?

Repeat these three questions, this time considering in what ways the application might be better on the Raspberry Pi than on your laptop.

Wrap Up

The next task is to configure your own Raspberry Pi in preparation for the technical exercises to come. Depending on the tools at your disposal, there are numerous ways to accomplish this.

4C: Getting Started with the Raspberry Pi

Setting up a Raspberry Pi 3, 4, or 400 for the first time depends considerably on what came with the purchase of that Raspberry Pi, what you have on hand, and the ways you plan to use your Raspberry Pi. It also depends on your skill level coming in and your interest, and availability for taking on the learning of new skills just to get things running. To facilitate demystifying technology for all, the following is based on a classroom context in which exercises are done using the Raspberry Pi operating system setup with a graphical user interface (GUI), given that it is the most widely used general purpose computer interface. First, set up a new Raspberry OS which will use a Raspberry Pi with direct connection to a monitor, keyboard, and mouse. At the same time, a given classroom cannot always provide a separate keyboard, mouse, and monitor for regular use with the Raspberry Pi. And as we move on in the book, we'll be exploring ways a general-purpose computer can run software services, such as a web server used to collect, manage, and share data. These often run using a "headless" configuration in which regular work on the general-purpose computer only requires a source of power as primary management is done with either a remote command-line (CLI) or GUI interface. In introducing hardware and software server environments beyond personal computer models, ongoing use of the configured Raspberry Pi OS will allow for headless server operation and only require provision of power to the Raspberry Pi computer, and Internet connectivity of the Raspberry Pi. For most, it is also necessary to have a personal computers on the same Local Area Network as the Raspberry Pi, either physically or using a Virtual Private Network (VPN) system, such as the Cisco AnyConnect provided for the University network in which this textbook is being written. Finally, documentation of the IP address assigned to the Raspberry Pi computer will be needed, as we will not register the Raspberry Pi so that it is assigned a publicly accessible IP name. To connect to the Raspberry Pi OS in order to complete tasks, you will be able to choose between using a virtual network computer (VNC) graphical desktop system or a secure shell (SSH) command-line protocol connection. The Raspberry Pi project provides well documented instructions for getting started with a new Raspberry Pi computer and new Raspberry Pi OS.[1] This chapter and various other instructions throughout the book

1. For those interested, the Raspberry Pi Start Up instructions include a section on how to do this "headless."

make use of key parts of text and images, both as is and remixed, of which you can find the originals on the Raspberry Pi company and the Raspberry Pi foundation websites.

> The Raspberry Pi company and foundation license their material under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA) license. Regarding attribution, the Creative Commons notes that new authors have the following responsibility in using such licensed work: "You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use." Regarding share-alike: "If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original."
>
> *A Person-Centered Guide to Demystifying Technology* stands by the Creative Commons and is itself licensed using the CC BY-SA license in support of lifelong learning, with special attention to those who are in marginalized and oppressed communities around the world. Indeed, one determinant in the selection of the Raspberry Pi computer and the Raspberry Pi OS instead of competing microcomputers and Linux operating system distributions was the Raspberry Pi project's commitment to open publishing of all their educational materials under this Creative Commons license.

Choosing Your Path Forward

As noted above, the first setup of a Raspberry Pi can vary depending on what you have on hand. Here's the list of physical materials used to write the following instructions in spring 2023:

- Raspberry Pi 3 – Model B+ – 1.4GHz Cortex-A53 with 1GB RAM (also tested on a Raspberry Pi 400 Desktop available as Computer Only or as a Full Computer Kit, both of which do not need a Pi Protector and which need a USB-C power supply)
- Adafruit Pi Protector for Raspberry Pi 3
- 5V 2.5A Switching Power Supply with 20AWG MicroUSB Cable
- 16GB or larger Class A1 microSD card (optionally purchase a 16GB Card with NOOBS 3.1 for Raspberry Pi Computers including 4)
- HDMI monitor
- U.S. standard keyboard and mouse
- Internet connectivity (tested using wired and wireless Ethernet, and mobile phone hotspot)

Beyond the hardware, all exercises in this book have been tested using the Raspberry Pi OS port of Debian 11 Bullseye released by the Raspberry Pi project April 4, 2022. The focus of this release of the Raspberry Pi OS was on improved security, including a new, non-optional "Welcome to the Raspberry Pi Desktop" wizard that comes up on first boot of a new OS install. This requires the setup of a username and password prior, which is required upon login to the Raspberry Pi OS Desktop. There is a slight security risk of continuing to use the username "pi." However, there are some applications, including those written for this book, that still make use of the /home/pi directory. To balance this, it is highly recommended you use a secure password. The April 2022 release also addressed a long-standing issue in which you needed a USB mouse and/or keyboard to initially pair Bluetooth peripherals such as a Bluetooth keyboard and mouse. The wizard now provides a prompt that can be used to pair with a Bluetooth device which has been put into pairing mode. In the Raspberry Pi Imager, this release provides a new set of steps within advanced options to allow "headless" use—no keyboard, mouse, or monitor needed, with all work being done through a remote connection—of the Raspberry Pi from the start. However, this pathway for setting up headless use of the Raspberry Pi does not work with the Virtual Network Computing (VNC) package we'll be using, so this new feature will not be used.

The remainder of this chapter is broken down into three main processes:

1. Exercise: Installing a Protective Case
    - Done once with a new Raspberry Pi 3 or 4 computer
    - Not needed for Raspberry Pi 400 computer
2. Exercise: Installing the Raspberry Pi OS
    - Done for each new Raspberry Pi OS installation
    - Not needed if your microSD card came preinstalled with the Raspberry Pi OS
    - This can be done: on a new blank microSD card; to reinstall the same operating system for Raspberry Pi use; or to install a different operating system available for the Raspberry Pi
3. Exercise: Setting Up a New Raspberry Pi OS
    - General configuration of the OS with things such as a username and password, Internet access, and updates
    - Raspberry Pi configurations for more specific use such as display and interface systems
4. Exercise: Adding an Adafruit PiOLED Text Display
    - Done following each new Raspberry Pi OS configuration
    - If IP address assigned to the Raspberry Pi remains constant, the PiOLED may not be needed

Exercise: Installing a Protective Case

Look at your laptops, phones, TV's, routers, and most other computer systems, and you won't actually see the computer, but rather a protective case in which the computer is located. This is also the case for the new Raspberry Pi 400 Desktop. For those using a Raspberry Pi 3 or 4, a minimal protective acrylic case gives us the opportunity to see the Raspberry Pi computer directly and get the labeling of the General Purpose Input Output (GPIO) pins while also keeping the Pi secure.

Looking at the image at right, unseen is an acrylic pad on which the Raspberry Pi sits. It is held in place by two short and two long black screws that come with the package. On top of the Raspberry Pi, you'll see a slightly shorter acrylic pad with GPIO labels facing the associated GPIO pins. It is held in place with the two taller black screws and provides protection to the top of the printed circuit board, integrated circuits, and connector slots.

Steps

You are ready to attach the Raspberry Pi to its new case.

1. The bottom of the case has a dip on one of the shorter sides to provide room to insert a microSD card into the Raspberry Pi moving forward. Holding this at the top, insert the two longer black screws into the left holes, and the two shorter black screws into the right holes.
2. Attach these screws to the acrylic using four black nuts. These only need to be hand tightened.
3. Insert the four screws into these pins so that the microSD slot aligns with the dip in the acrylic.
4. Attach nuts to the two shorter black screws above and below the GPIO pins.
5. Place a white plastic tube to each of the two longer black screws.
6. Lower the top acrylic cover to the tall back screws so that it sits on top of the white plastic tubes.
7. Attach nuts to the two taller back screws to hold down the top of the acrylic case. These only need to be hand-tightened.

Key Takeaways

A case is used to protect electronics from wear and tear. They come in many shapes and forms. The design of the case impacts effective use in significant ways. To demystify technology, I've chosen to use an acrylic case with label identifying the GPIO pins as commonly used. But my choice of this particular Adafruit Pi Protector for Raspberry Pi was inspired through Limor Fried, whose goal in starting Adafruit in 2005 “was to create the best place online for learning electronics and making the best designed products for makers of all ages and skill levels.”[2] And it is further aligned with the Raspberry Pi project’s motivation of “Democratising technology—providing access to tools“—which in working to lower the barriers to entry by providing both a lower cost computer and extensive educational materials, is finding “Raspberry Pi computers being used everywhere from interactive museum exhibits and schools to national postal sorting offices and government call centres. Kitchen table businesses all over the world have been able to scale and find success in a way that just wasn’t possible in a world where integrating technology meant spending large sums on laptops and PCs,”[3]

A case is a social as much and more as a technical tool. It is a sociotechnical artifact that can provide you with insights into the designers, producers, and distributors of that product. Take a minute to look at the many sociotechnical cases you use in your daily life. What might be some of the motivations for each of their designs? What in these designs would you have done differently to help them work better for you? What in these designs has helped you work better than those of other devices you considered purchasing?

Exercise: Raspberry Pi OS Installation on a microSD Card

Sometimes you can find microSD cards with the Raspberry Pi OS with Desktop already installed, in which case you can jump straight to Exercise: First Boot of the Raspberry Pi OS.

But for most, an affordable option has been to purchase a fresh 16 GB or larger microSD. It’s recommended you select a microSD card whose speed categorization is the new A (for “application”) class. You’ll see an A1 logo next to the overall speed class, a number found within the C symbol. While the overall speed found with the C symbol determines the minimum write speed of the microSD, the application class goes on to determine how many read and write operations need to be completed within one second. For example, an A1 category, class 10 speed microSD card, will have a minimum write speed of 10MB/s (the class 10 speed requirement) that will also support at least 1500 read operations and 500 write operations per second (the class A1 speed requirement). All official Raspberry Pi microSD cards meet this specification.

The Raspberry Pi Foundation has developed a graphical SD card writing tool called Raspberry Pi Imager, which they suggest as the easiest option for most users, since it will download the image automatically and install it to the SD card a computer with an SD card reader. It works on Mac OS, Windows, and Ubuntu 18.04 and above. There are many USB-based SD card readers available, which you can then use if your computer does not come with its own SD card reader.

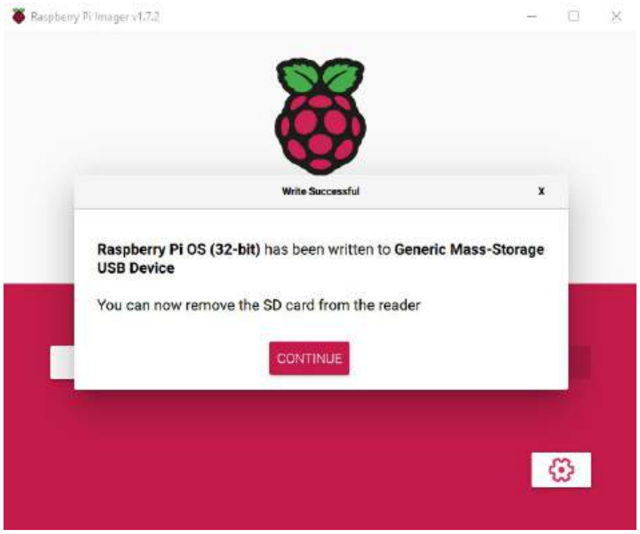
To set up and use Raspberry Pi Imager on your personal computer:

2. https://www.adafruit.com/about
3. https://www.raspberrypi.com/about/

1. Download the latest version of Raspberry Pi Imager, and install it to your personal computer.
2. Insert a microSD card to your computer in the microSD slot, or connect a USB-based SD card reader with the microSD card attached into an available USB port.
3. Open Raspberry Pi Imager.
    - For "Choose Operating System," select the Raspberry Pi OS (32-bit) from the list presented, as this is the recommended operating system for most users.
    - For "Choose Storage," select the USB mass-storage device you just added.
        - Make sure it matches the size of storage you expected. In my case, it lists 15.5 GB for my 16-gigabyte microSD card.
        - In case you're not certain, you can remove the microSD card you plan to use, and it should disappear from the display.
4. Review your selections and click on the Write button to begin writing data to the SD Card. NOTE: If using Raspberry Pi Imager on Windows 10 with controlled folder access enabled, you will need to explicitly allow Raspberry Pi Imager permission to write the SD card. If this is not done, the imaging process will fail with a "failed to write" error.

5. After the Raspberry Pi completes the writing of the image to the microSD, it will do a verification that everything was written successfully before providing a "Write Successful" popup. Click "Continue," and then, if it wasn't done automatically by the Imager, eject the new "USB Drive"[4] before removing the microSD.

4. In the file viewer for your operating system, click on the eject icon, or right-click on the filename of the microSD and select eject.



For more, [check out their brief introduction of the Raspberry Pi Imager in their 45-second video, "How to Use Raspberry Pi Imager."]

Exercise: First Boot of the Raspberry Pi OS

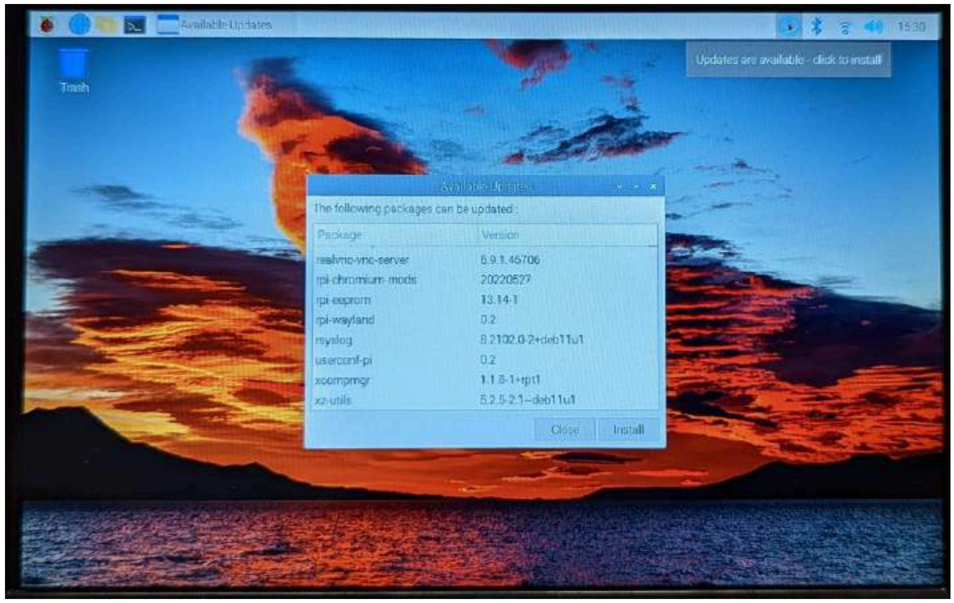
Once you have a microSD card with the Raspberry Pi OS installed, the next step is to do an initial configuration of the operating system. The following instructions were tested using Raspbian GNU/Linux 11 (bullseye).

1. Connect your Raspberry Pi to a keyboard, mouse, and monitor. Check on the specifications of your Raspberry Pi hardware model, as port types and included components vary.
2. Install the microSD card into the microSD port on the Raspberry Pi.
3. Insert power to the Raspberry Pi.
    - For every newly installed Raspberry Pi OS, the first boot will go through a couple of passes to do some initial configurations:
        - By default, the first boot will be used to resize the microSD to make use of all available space.
        - A reboot will then take a few moments to do some further initial automated configurations before arriving at a "Welcome to the Raspberry Pi Desktop" message.Insert power to the Raspberry Pi.
4. First set up:
    - Set Country—The instructions in this book were primarily tested within the United States using American English, but two-thirds of the readers of the first edition have come from the Asian and African continents. Ideally you can configure the Set Country components to match yours.
    - Create User—While it is extremely important to carefully create a highly secure

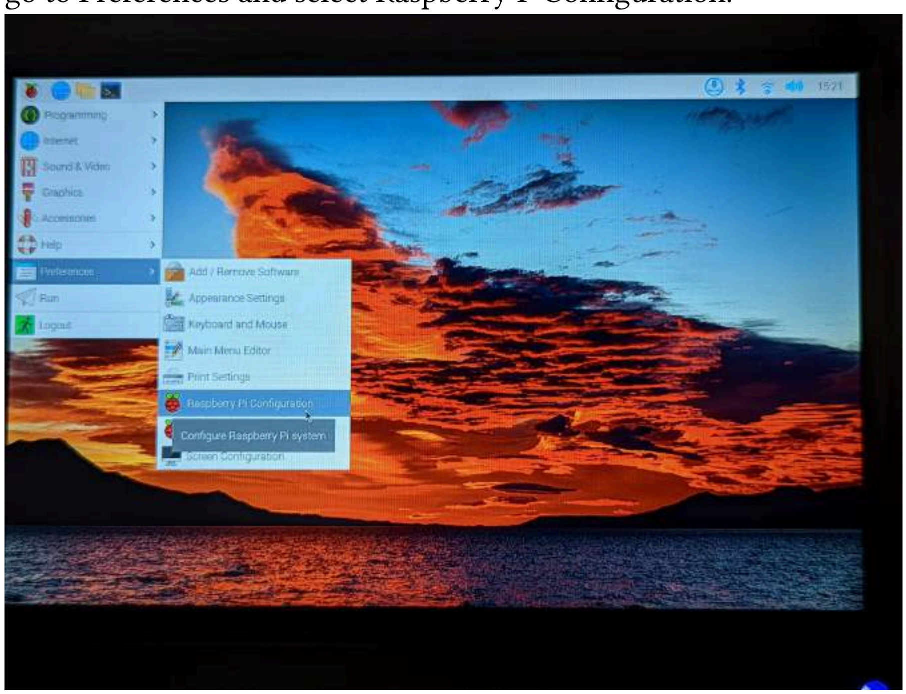
password for your personal and professional devices, when working in a collaborative environment using a prototyping and educational system such as the Raspberry Pi, you may find it helpful to have a shared password in this specific context. *But if this is done, be very careful to never put any data or information on the Raspberry Pi that you aren't willing to have go public across the Internet!*

- Set Up Screen—Enable and again disable the "Reduce the size of the desktop on this screen" option to see if parts of the image are being missed because they don't match the monitor display you are using.
- Select WiFi Network—We plan to make ongoing use of a remote computing application like Virtual Network Computing (VNC) or secure shell (SSH). In almost all cases, for security by default this requires that the Raspberry Pi be on the same local area network (LAN) as the remote computer. If possible, select the WiFi network from the list that matches that being used by your personal computer.
    - NOTE: In some educational and work contexts, security may prevent an unrecognized computer from connecting to that network. Consult with your support team at your location on pathways forward if you find this to be the case.
    - As an example, at the University of Illinois Urbana-Champaign, there is an option to register devices other than personal laptops and mobile phones using a Wireless Guest and Device Management Portal. Whether a smart TV, video game console, or a Raspberry Pi, a student can register their personal devices using this Portal by identifying the media access control (MAC) address associated with the wireless Ethernet network interface card (NIC). Additionally, departmental accounts can register Raspberry Pi's, Arduino's, and Internet of Things devices being used for their studio courses. Either way, a personal computer on the IllinoisNet Wi-Fi network is able to remotely access these other devices on the IllinoisNet_Guest Wi-Fi network.
- Update Software—If you can connect the Raspberry Pi to a WiFi network, it will now do any needed updates in the operating system and installed applications. If possible, click Next to do this now, as this will be needed for future exercises.
    - Selecting Skip may be necessary for those with limited time or Internet access now, in which case updates will need to be done later.
    - If you get a popup stating the process reached a timeout, this also indicates an Internet access problem, in which case updates will need to be done later.
- With setup complete, you'll be prompted to press Restart.
    - After restarting, you may see an update prompt to the left of the Bluetooth and Wi-Fi icons. Make sure you're connected to Wi-Fi, and then

click on the icon and select to either start the install or to first view the available updates before starting the install.



5. Raspberry Pi Configuration—While the Raspberry Pi OS took you through some general configurations upon first boot of a newly installed microSD, there are a range of other configurations that will be needed at some point. Let's do some of these next:
   - Click on the Raspberry icon in the upper left, and then, in the drop-down menu, go to Preferences and select Raspberry P Configuration.

   - The Raspberry Pi Configuration has several tabs in which you can configure different core parts of the operating system. At a minimum, make the following configuration changes:

- System
    - Password—If you didn't set one previously, now's the time to do that.
    - Disable the Splash Screen. Moving forward during the boot process, you'll see a line-by-line list of actions being taken, along with indications of success or failure. A Splash Screen is now widely used by different operating systems to hide many of these to facilitate a more inviting initial experience upon starting a computer.
- Display
    - Headless Resolution: I recommend setting this to 1280×720 to assure a reasonably usable display of the Raspberry Pi desktop, but one that is still smaller than that of the computer you will be using to remotely access your Raspberry Pi moving forward.
- Interfaces. Make sure each of the following are enabled:
    - SSH—the secure shell server for remote command line access to your Raspberry Pi. Enabling SSH means that the server is restarted each time you boot the Raspberry Pi.
    - VNC—the Real VNC server for remote Virtual Network Computing access. Enabling VNC means that the server is restarted each time you boot the Raspberry Pi.
    - SPI—the Serial Peripheral Interface on the GPIO header. Enabling here also automatically loads the SPI kernel module needed for various products using this serial interface.
    - I2C—the Inter-Integrated-Circuit bus on the GPIO header. Enabling here also automatically loads the I2C kernel module. Unlike UART and SPI, which provide send and receive communications to one or two attached devices, respectively, I2C is a bus and can work with device addressing to do serial send and receive communications with many more devices.
    - Serial Port—the GPIO transmit (GPIO #14) and receive (GPIO #15) pins on the Raspberry Pi can provide Universal Asynchronous Receive Transmit (UART) communication with a remote device, using something like a TTL to USB serial cable between GPIO #14 and GPIO #15 and the USB port on a computer.
    - Serial Console—provides a serial console display to a remote

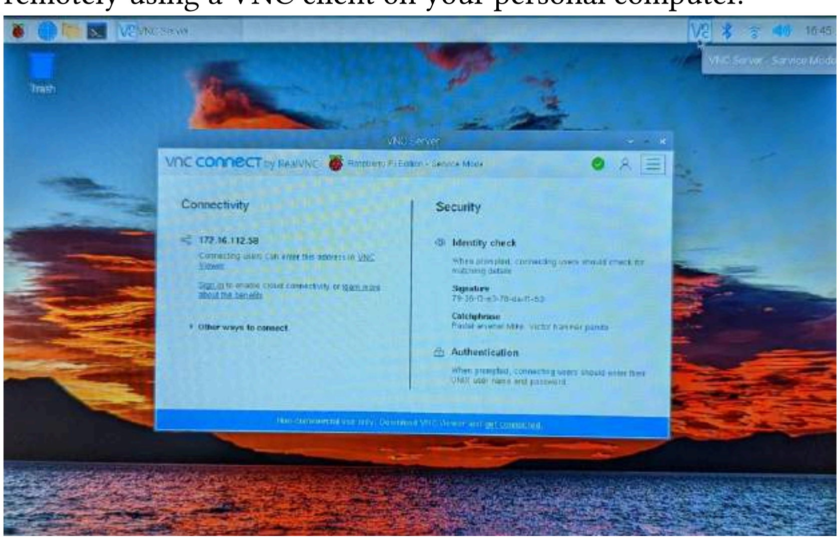
device using serial communication.

- Localisation—Here's where you can set up Locale, Timezone, Keyboard, and WiFi Country if you didn't do this during the "Welcome" setup above, or if you need to make changes at any point.

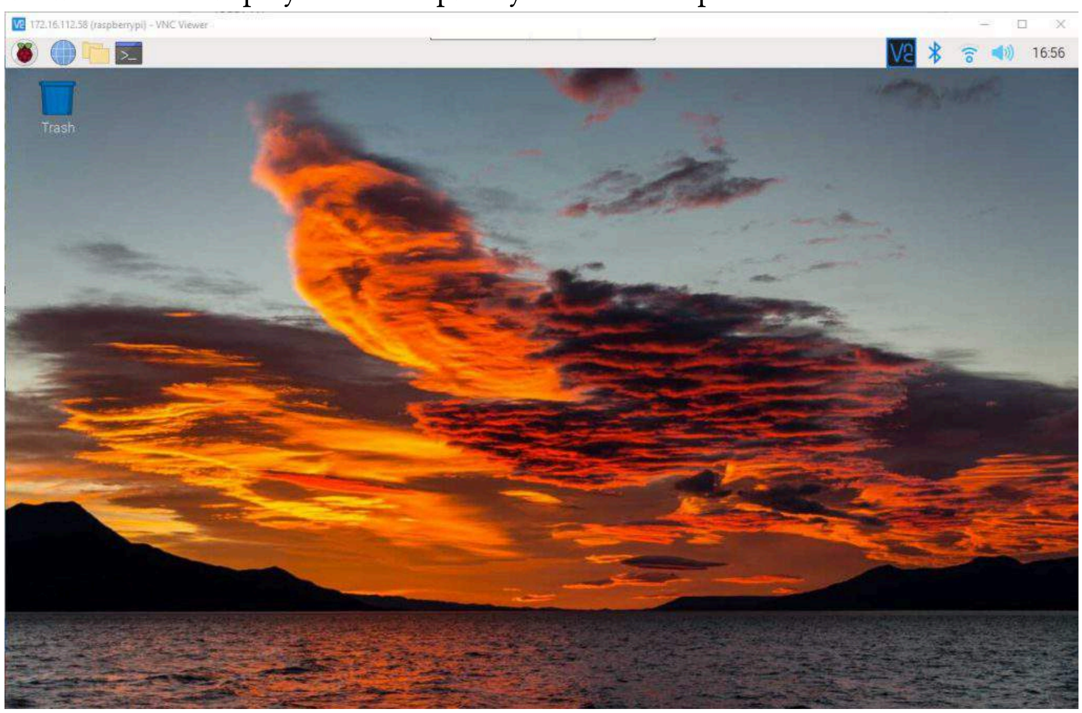
◦ Once you've completed configuration setup, click OK. You'll be prompted to reboot for the changes to take effect. You may see a few reboots occur for everything to fully take effect.

6. Once complete, you'll return to the Raspberry Pi OS desktop, where you'll now see an additional Real VNC server icon at the top:

◦ Click on the VNC tab to identify the IP address(es) assigned to your Raspberry Pi under "Connectivity." You'll use this IP address to access your Raspberry Pi remotely using a VNC client on your personal computer.



- On your computer, open a new tab in your web browser. In the address bar, type in https://www.realvnc.com/en/connect/download/viewer. Download and install RealVNC Viewer on your computer.
- Open RealVNC Viewer on your computer. As with a web browser, you'll find an address bar where it says, "Enter a VNC Server address." Type in the IP address of the Raspberry Pi you just identified.
    - If prompted regarding VNC Server Identity Check Failed, click "Continue" as you can confirm you are connecting to the expected Raspberry Pi VNC Server.
    - When prompted, use the username and password you just configured for your Raspberry Pi OS. You can save this in VNC Viewer for automatic logins in the future.
- The VNC icon in the Raspberry Pi OS menu will change from blue on white to blue on black, as viewed from both the Raspberry Pi monitor and your VNC Viewer remote display of the Raspberry Pi OS desktop.



    - Moving the Raspberry Pi mouse will be seen in the VNC viewer display, and clicking on the VNC Viewer display and then moving your computer mouse will be seen on the Raspberry Pi monitor!
    - As with past exercises, this may prove to "not-yet" be working, requiring strategic "fail-forward" troubleshooting.
- From here, as long as you know the IP address of a Raspberry Pi, you will no longer need a keyboard, mouse, and monitor for your Raspberry Pi, and will be able to instead manage things remotely from your computer. The Raspberry Pi

can now be run “headless.”

When you connect to a new Raspberry Pi, you may need a mouse, keyboard, and monitor to identify the IP address of the new Raspberry Pi before returning to headless operation. If your Raspberry Pi is provided a dynamic IP address, the address assigned may change at any moment, in which case you may need the keyboard, mouse, and monitor temporarily. Therefore, in the next exercise, we’ll setup an organic LED (OLED), on which the IP address can be listed for continuous headless operation.

NOTE: You may be able to connect from a desktop or laptop web browser to the gateway router for your local area network (look around the case of the router to find its default IP address, and enter that into the URL address bar of the browser). In the router’s website, there is often a listing of connected devices and the IP addresses they have been assigned.

NOTE: On occasion, I have found that the Headless Resolution setup above will somehow conflict with the RealVNC Viewer program on your personal computer. It has not occurred frequently enough for me to fully diagnose a cause. And in most cases, rebooting the Raspberry Pi has been sufficient to overcome this problem. Only once did I need to reconnect the Raspberry Pi to a keyboard, mouse, and monitor to again run Raspberry Pi Configuration to adjust the Headless Resolution under Display.

Try Something New!

Instead of working remotely using the VNC GUI, it’s possible to open a terminal window on your laptop to use Secure Shell to login remotely. On your computer’s terminal, type:

```
ssh pi@x.x.x.x
```

where x.x.x.x is the IP Address of your Raspberry Pi.

After logging in using SSH, you can run any terminal-based applications. Many that have a GUI version, such as Raspberry Pi Configuration, also have a CLI version. For configuration, you would instead type the following in the SSH terminal window:

```
sudo raspi-config
```

Key Takeaways

In this exercise, we launched the Raspberry Pi microcomputer with the Raspberry Pi OS for the first time. For anyone who has used a brand-new computing device (e.g., computer, tablet, mobile phone) for the first time, you’ll have likely found a range of similarities and differences. These similarities and differences, though, aren't just between the Raspberry Pi and that of all the other computers you've ever used. Rather, brands and generations of computers each have been shaped by a wide range of social, cultural, and economic factors that have influenced the look, feel, and functionality of a given device within a given context. To this extent, Raspberry Pi is just like other computers in use today.

- Compare and contrast your experiences with the different computers you have used. There are three primary types of operating systems used today:
    - Microsoft Windows-based operating systems developed by Microsoft in the early 1980s.
    - Apple Mac-based operating systems, a Unix-based system released at the end of the century to replace the "classic" Mac OS introduced in 1984.
    - Linux-based operating systems, including:
        - Android, a mobile operating system developed by Google, first commercially used in 2007.
        - Chrome OS, an operating system that uses the Google Chrome web browser as its principal user interface, first used commercially in 2011.
        - Ubuntu (first released in 2004) and other popular Linux distributions used on laptop, desktop, and server computers around the world.
        - Raspberry Pi OS with Desktop, which we just set up for the first time.
    - If possible, compare and contrast any brand-new computers you've turned on for the very first time.
    - As you compare operating systems, consider how the following components vary:
        - The command line interface
        - The setup and configuration files
        - The included everyday applications

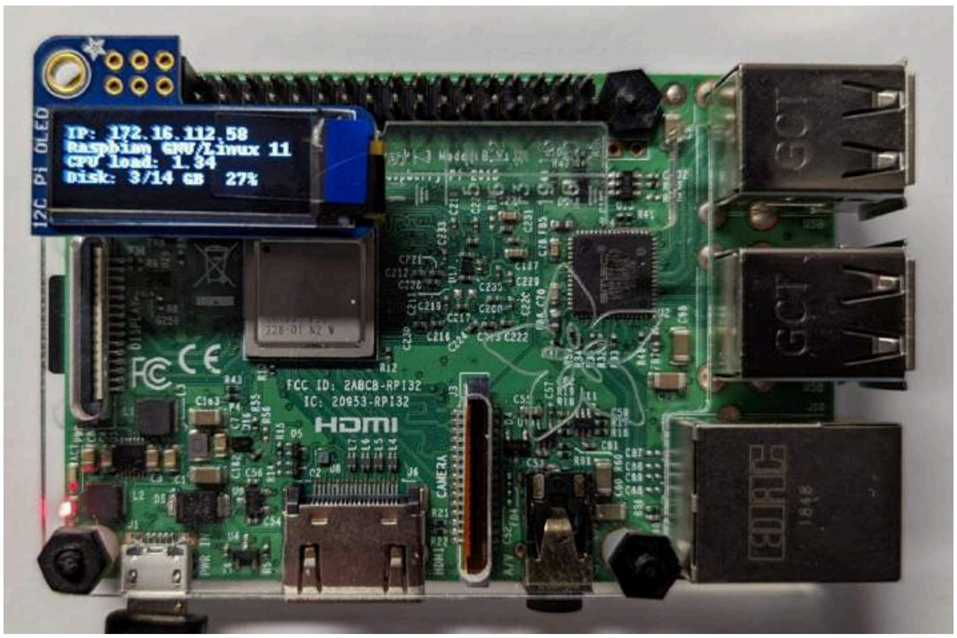
Exercise: Adding an Adafruit PiOLED Text Display

An organic light-emitting diode (OLED) is one or more LEDs in which an organic compound is added on top to emit light in response to an electric current. The OLED in the image below provides 128 individual white LEDs on the x-axis and 32 individual white LEDs on the y-axis, for a total of 4,096 white LEDs. In addition to these white LEDs, the printed circuit board of this display also has a single-chip driver and controller, designation SSD1306. It is through the SSD1306 chip that data and commands are exchanged between the OLED dot-matrix graphic display system and the device to which it is connected using the I2C serial communication bus—in our case, pins 2 and 3 on the Raspberry Pi GPIO.

Adafruit includes software drivers for use on the device in their Python library, which will allow us to keep track of the IP address, operating system, CPU load, and disk usage of the Raspberry Pi. The following steps will need to be done once for each newly installed operating system—specialty drivers generally are not installed within the base of a general-purpose operating system such as the Raspberry OS, Microsoft Windows, Mac OS, etc.

If the GPIO pins on your Raspberry Pi are readily accessible, you can connect the PiOLED to the following GPIO pins (use the table to match the image below with the associated Raspberry Pi GPIO, or General Purpose Input Output, pins being used):

5V	5V	GND
3.3V	#2	#3

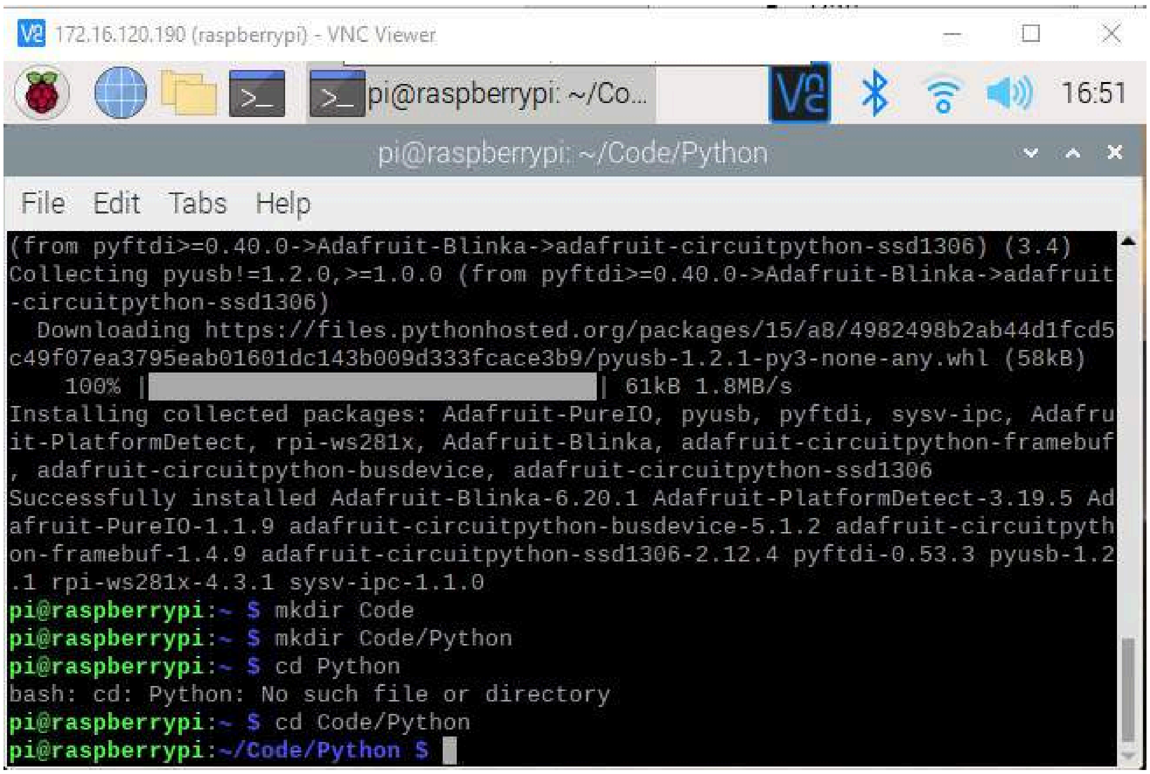


Steps

1. Open a terminal window on the Raspberry Pi. Type in the following to install the needed Python 3 libraries and drivers:
   `sudo pip3 install adafruit-circuitpython-ssd1306`
   - If you get an error that pip is not installed, you will need to rerun the above after first typing in the following:
     `sudo apt install python3-pip`
     `sudo apt install python3-pil`
   - If this still doesn't work, you may need to do a full update of the Raspberry Pi OS and installed applications by typing in the following before rerunning the above:
     `sudo apt update`
     `sudo apt full-upgrade`
2. Next, type the following in the terminal window to create new directories in your home

directory.

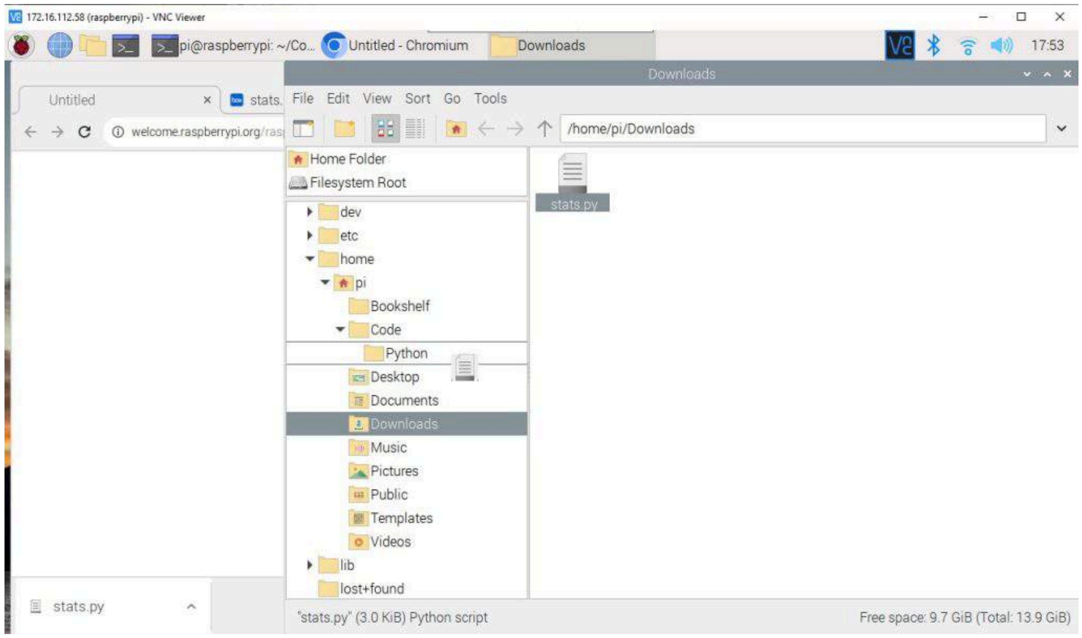
```
mkdir Code
mkdir Code/Python
cd Code/Python
```



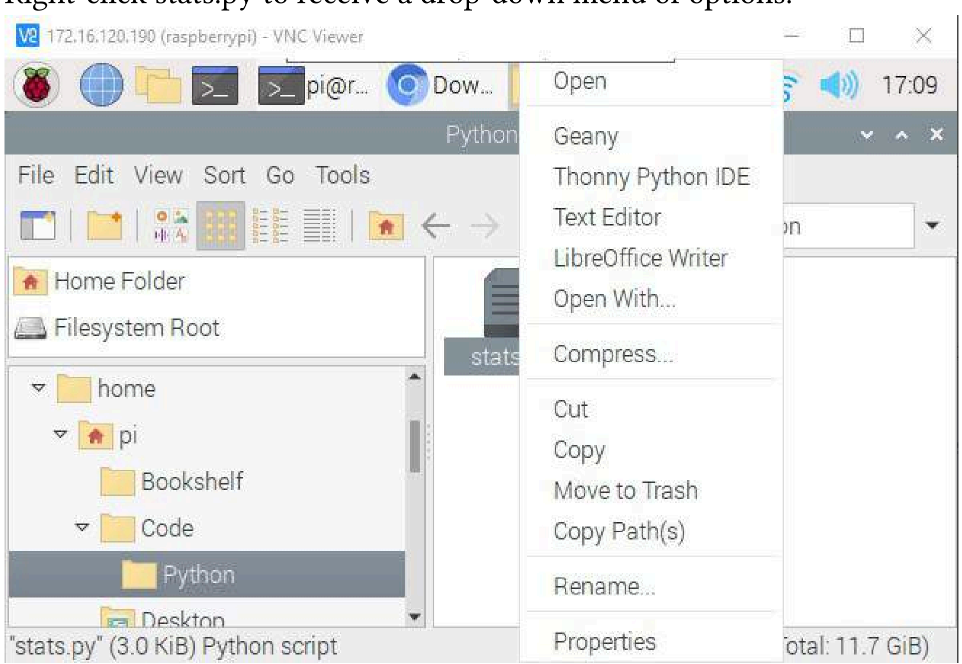
3. On the Raspberry Pi, click the blue circle icon associated with the Chromium web browser. Then enter the following URL:
   https://go.illinois.edu/PiOLED_stats
4. Skim through the Python code to get a rough sense of what it's doing, and then download the code to your Raspberry Pi.
   - It may state that "Your Download is In Progress" while also asking if you want to "Sign Up" or "Log In." This popup can just be closed—no account or log in is required.
   - You'll likely receive a prompt stating, "This type of file can harm your computer. Do you want to keep stats.py anyway?"
     - As Python files can include malicious code, this is a safety mechanism to ensure that you trust this program.
     - Select Keep here.
   - Click on the Carat (ˆ) symbol where it lists the downloaded stats.py file. From the popup, select "Show in folder" to open the Folders Application.
5. In the left section of the Folders window, click on the right-angled triangle in front of the Code directory to expand the subdirectories within Code. This will let you view the Code/

Python subdirectory.

- Drag the stats.py code from /home/pi/Downloads into the /home/pi/Code/Python subdirectory.



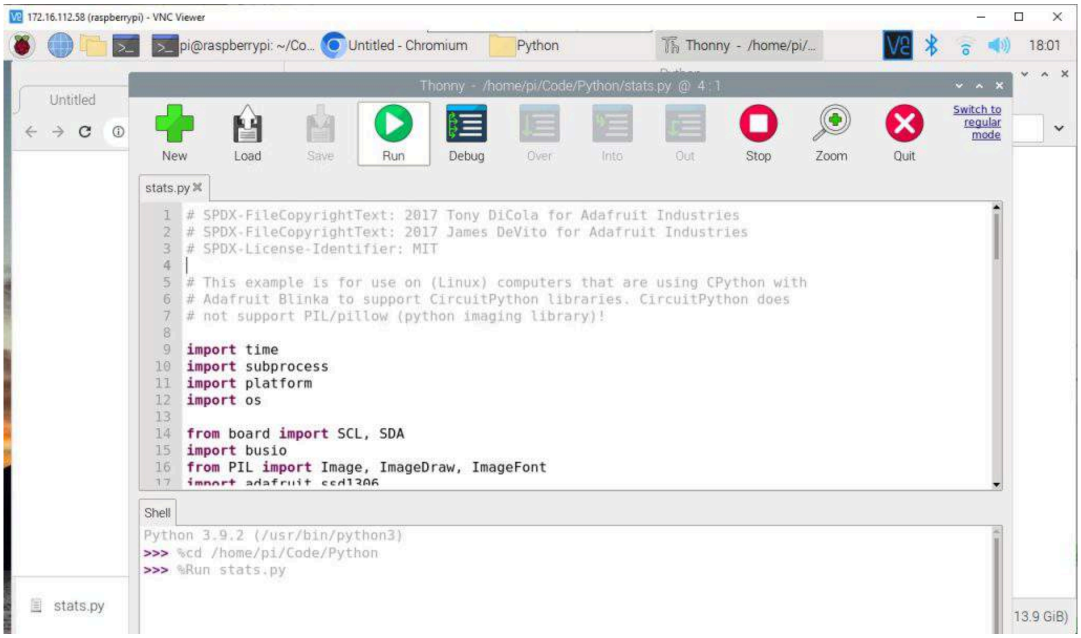
- Click on the Python subdirectory to again view stats.py in the right section of Folders.
- Right-click stats.py to receive a drop-down menu of options.



6. From here, select Thonny Python IDE, a relatively beginner-friendly Python Integrated Development Environment (IDE) that comes with Raspberry OS Desktop.
7. In Thonny, click on the Run icon.
    - If everything goes well, you should see an indication at the bottom of Thonny

that everything is OK.

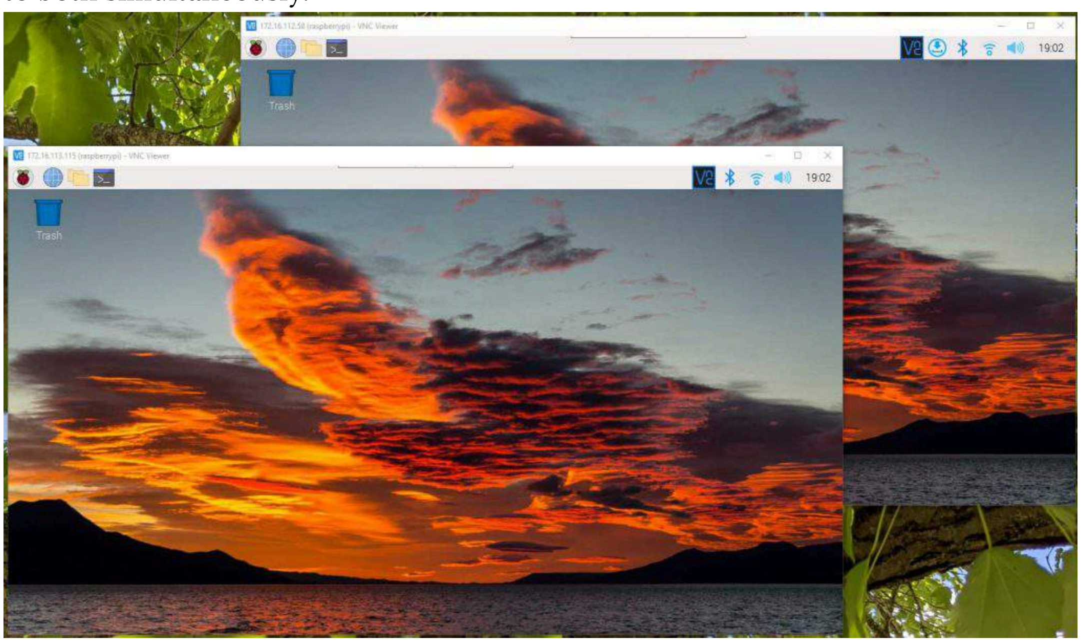
- And on the PiOLED, you should now see the IP Address of your Raspberry Pi.
- In the following images, note that the VNC Viewer and the stats.py program being displayed using the PiOLED both indicate the same Raspberry Pi IP address, 172.16.112.58. After reviewing these images and associated text, proceed to the next step to have stats.py run automatically each time the Raspberry Pi is booted.



Below is an image of a Raspberry Pi 3 side by side with a Raspberry Pi 400, on which the PiOLED is also set up. By default, the GPIO pins of the Raspberry Pi

400 are on the backside of the keyboard. In the image, an Adafruit CYBERDECK HAT is used to provide the same GPIO pin layout in a forward-facing location. And an Adafruit GPIO Reference Card for Raspberry Pi has been placed on top of the CYBERDECK to provide labeling similar to that provided by the Adafruit Pi Protector for Raspberry Pi installed during the first exercise of this chapter.

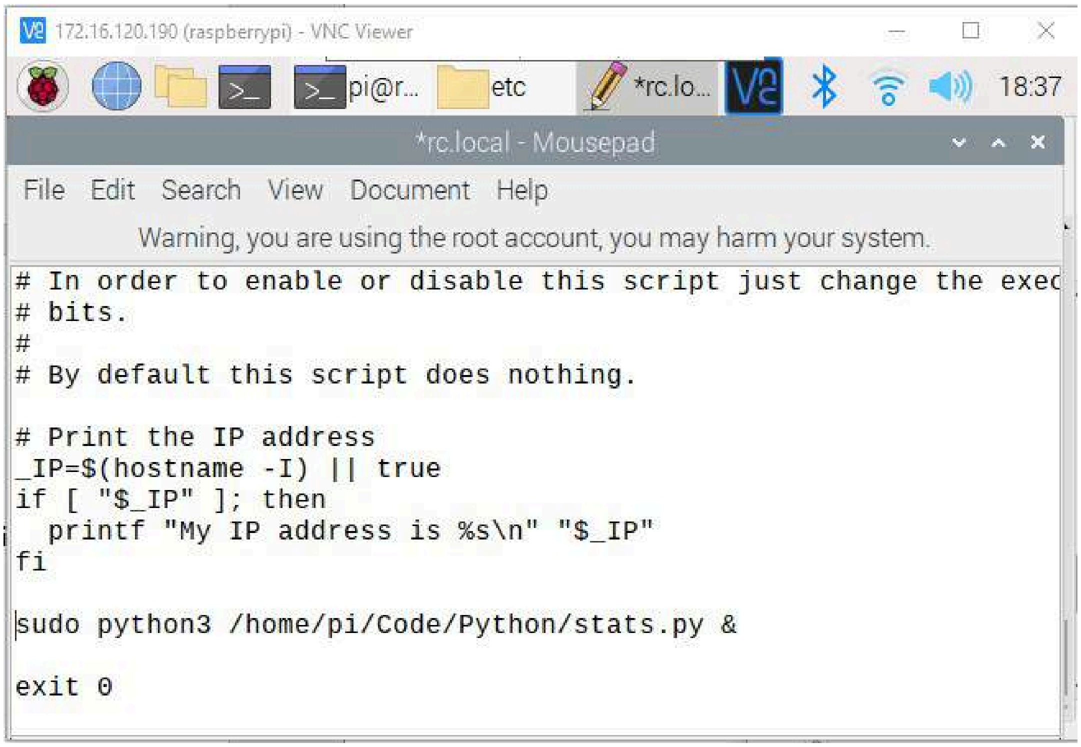
Note that both the Raspberry Pi 3 and Raspberry Pi 400 are on the same 172.16 local area network, and a remote computer's VNC Viewer can make connections to both simultaneously.



8. With the stats.py program tested, let's set it up to automatically run every time the Raspberry Pi boots up.
    - In the Raspberry Pi terminal window, type:
      `sudo mousepad /etc/rc.local`
    - Scroll to the bottom of the rc.local file you now have opened in Mousepad. In a

new line just above the command "exit," type:

```
sudo python3 /home/pi/Code/Python/stats.py &
```



- Click on the File option in Mousepad to save the edited file, and then close Mousepad.
- In the terminal window, type:

  ```
  sudo reboot now
  ```

Watch the newly installed PiOLED and the newly configured RealVNC viewer. Together, you should see that the Raspberry Pi goes through a full reboot process, and that the PiOLED automatically restarts and provides a continuous update of baseline information regarding the Pi. At this point, you should be provided with any new IP Address the Raspberry Pi is assigned if it is used in different locations in which it is set up to connect on the Local Area Network (LAN).

Wrap Up

In these exercises, we've worked to: 1) do a one-time installation of a case on the Raspberry Pi computer; 2) install the Raspberry Pi OS Desktop to a microSD card and do an initial configuration of this operating system for ongoing remote management; and 3) add drivers and Python code to the Raspberry Pi OS so that when attached, a 123×32 PiOLED mini-display can be used to determine the current IP address, operating system, CPU load, and disk usage of the Raspberry Pi.

As was noted in Orange Unit 3B: Computer Building Blocks, the operating system is installed and maintained on a storage device like a hard drive or microSD card. But even the newest versions of

SSD and microSD storage devices are like our old paperwork that we store up in an attic. We can't do anything with it beyond just storing it there. When a computer is turned on, those instruction sets in the operating system which are immediately needed are moved out of storage and into random access memory (RAM). We sometimes call this "booting" a computer, also called "boot up" or "start up." We disabled the splash screen to watch the line-by-line work done to boot a computer.

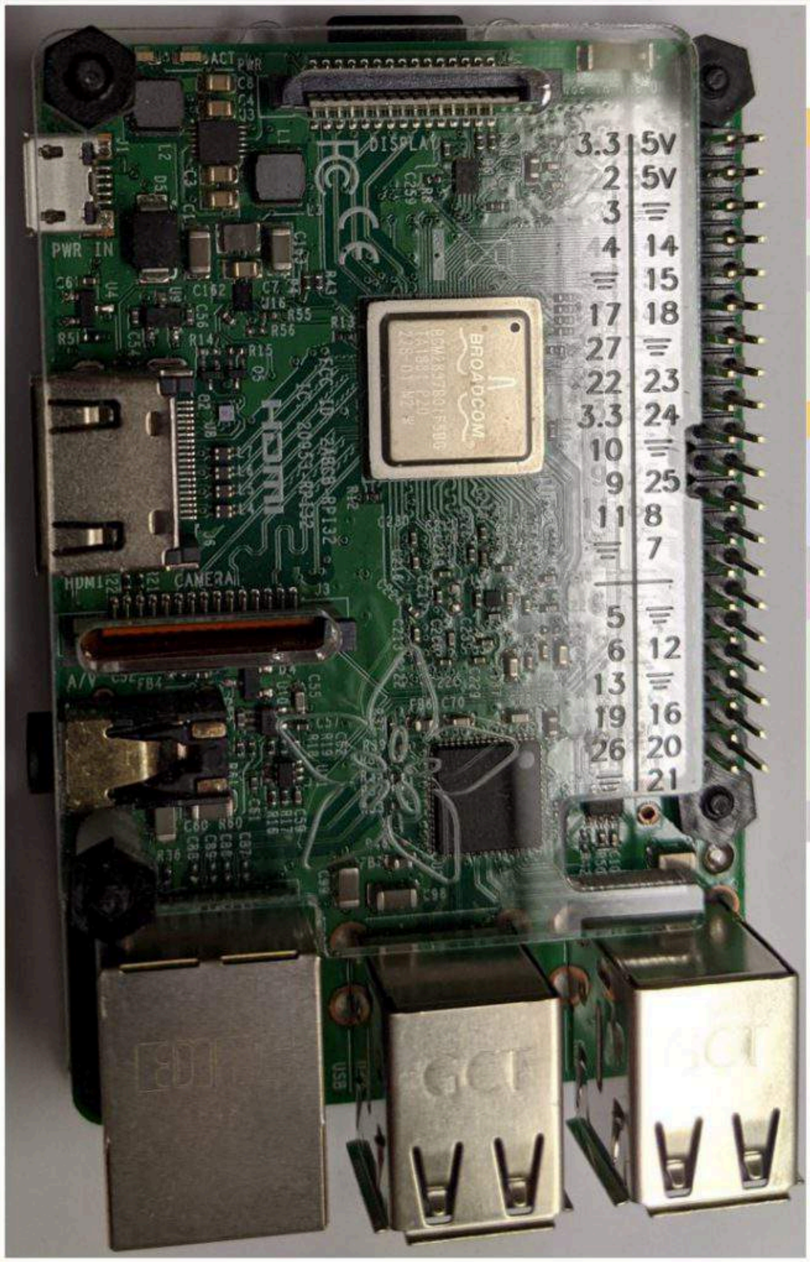
4D: Coding Electronics

As a final exploration for this unit, we'll bring the hardware together with software, implementing Python code to turn the LED lights on or off. So far, your LEDs and mechanical switches have functioned together directly as part of isolated circuits coming together as a larger circuit on the breadboard. Physical motion of the switches is used to communicate choices to other parts of the circuit. Now, let's rewire them to add digital communication using an application to build a programmable circuit, adding still another layer of complexity into the system.
NOTE: there are many microcontrollers, such as the Circuit Playground Express we'll use in the Blue Unit, that have sufficient digital IO pins and can run the Python code we'll use in this concluding chapter. However, we're using the Raspberry Pi, as it builds from the previous exercises of this session of the Orange Unit. Further, the use of the Raspberry Pi here sets the stage for use of Python code with this RGB LED breadboard prototype in the Rainbow Unit to provide digital switching using a web browser!

Exercise: Connecting a Breadboard to a Raspberry Pi

In the previous two chapters of this Orange Unit session, 4B: Meet the Microcomputer and 4C: Getting Started with the Raspberry Pi, we began familiarizing ourselves with the application of computer building blocks specifically to the Raspberry Pi microcomputer. After meeting the hardware, we went on to set up a Raspberry Pi-based Linux operating system, an organic (that is, self-luminous) light-emitting diode (OLED), and a bootup script to display some essential current data on the functioning of the Raspberry Pi. The result looked something like the photo at left.

But in doing so, we've also reserved access to six of the General Purpose Input Output (GPIO) pins, including the only two 5-volt power pins.



3V3 Power	1	2	5V Power
SDA1 I2C	3	4	5V Power
SCL1 I2C	5	6	Ground
GPIO #4	7	8	UART0 TXD
Ground	9	10	UART0 RXD
GPIO #17	11	12	GPIO #18
GPIO #27	13	14	Ground
GPIO #22	15	16	GPIO #23
3V3 Power	17	18	GPIO #24
SPI0_MOSI	19	20	Ground
SPI0_MISO	21	22	GPIO #25
SPI0_SCLK	23	24	SPI0_CE0
Ground	25	26	SPI0_CE1
I2C ID_SD	27	28	I2C ID_SC
GPIO #5	29	30	Ground
GPIO #6	31	32	GPIO #12
GPIO #13	33	34	Ground
GPIO #19	35	36	GPIO #16
GPIO #26	37	38	GPIO #20
Ground	39	40	GPIO #21

Above, you can see two different displays of the pin layout formats commonly used with the Raspberry Pi GPIO[1].

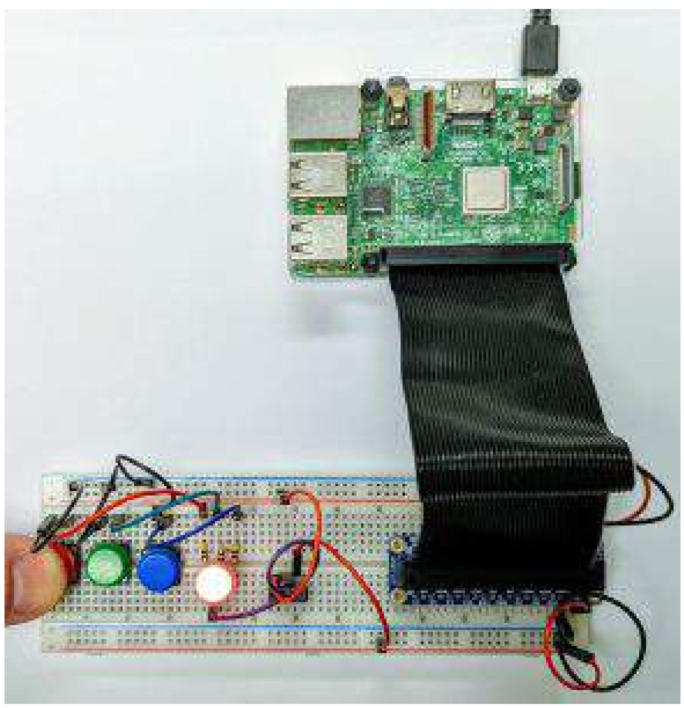
Look through these and note that some, like #10, #9, and #11 just below the middle 3.3V pin, can be used for multiple purposes, such as digital inputs and outputs, or as a serial peripheral interface (SPI) connector with Master Out/Slave In, Master In/Slave Out, and Master Clock pins.

1. The website https://pinout.xyz/ provides a useful guide on the Raspberry Pi GPIO pins. They also provide a simple program you can run from the terminal window on your Raspberry Pi where you would type: pinout

> The use of the terms "master" and "slave" is rightly discomforting for many. As noted in the tween zine *Beyond Dark Matter*, "The master/slave relationship has been used for centuries in technology, often to explain situations where one master process or component controls a slave process or component" (p. 31, *Beyond Dark Matter*).
>
> Efforts are made periodically to move away from this embedded framing of the master/slave relationship within core aspects of all our electronics. This blockquote seeks to unpack further the hidden sociopolitical and economic realities that remain encoded within our sociotechnical technologies and systems.

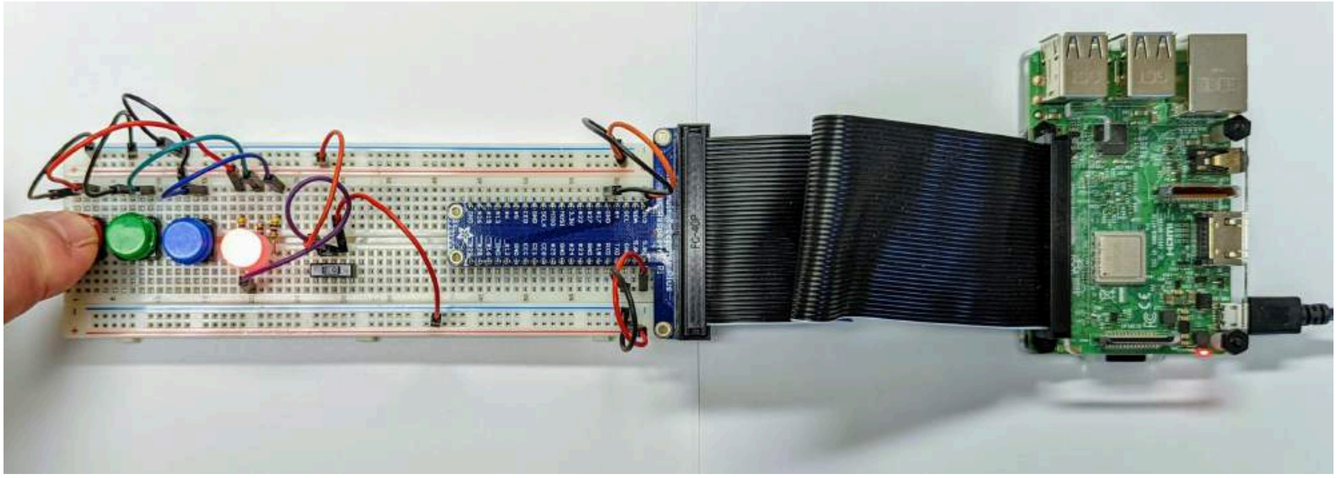
While there are many other pins for us to use than the ones covered by the PiOLED attached directly to the GPIO, this layout has proved less than optimal for later exercises, in which the Raspberry Pi is used with the breadboard electronics. This is particularly the case when a shared Raspberry Pi is connected and disconnected with different breadboards as each person tests out their own prototype[2].

For the next exercise, we'll remove the PiOLED from the Raspberry Pi GPIO, and instead connect a 40-pin ribbon cable to the 2×20 GPIO of the Raspberry Pi. This can then be attached to a Cobbler or T-Cobbler on a full-sized breadboard. As we've used columns 1-30 for past exercises in the Orange Unit, we'll now attach the T-Cobbler to columns 44-63 of the breadboard.



Breadboard connected to Raspberry Pi via Adafruit Cobbler and ribbon cable

2. During beta-testing 2021-2022, the PiOLED was left connected and a half-sized breadboard was used. Male-to-female jumper wires were used to connect the appropriate pins on the Raspberry Pi GPIO with the associated holes on the breadboard. This remains an option for those with ongoing access of the same Raspberry Pi and for whom consistent access of data from the PiOLED and/or the smaller half-sized breadboard has proved of importance.

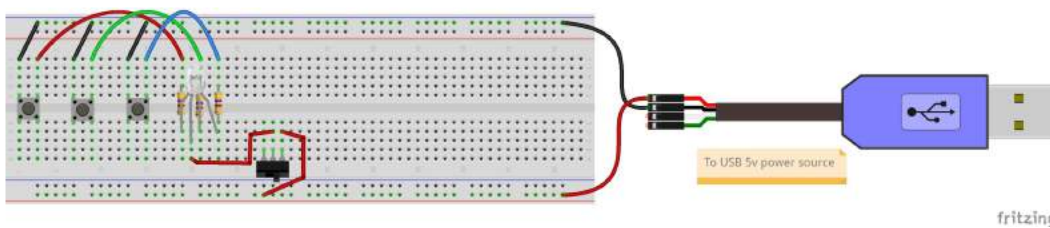


Breadboard connected to Raspberry Pi via Adafruit T-Cobbler and ribbon cable

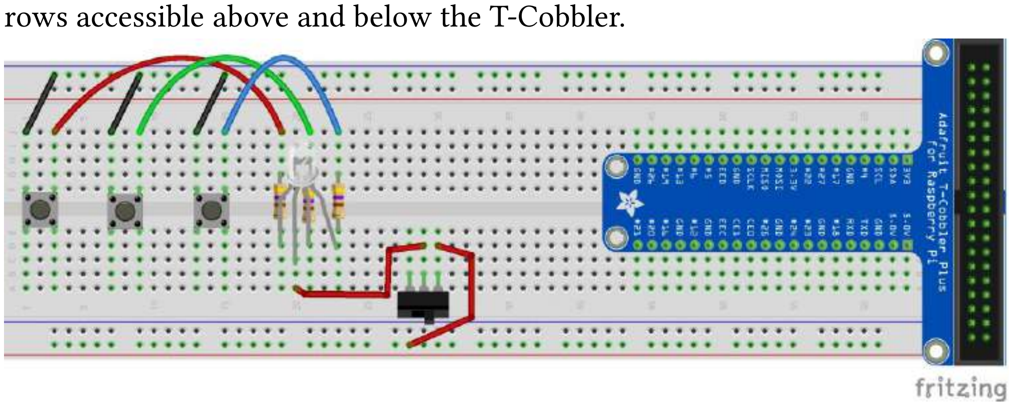
If your IP address changes moving forward, you may need to disconnect the ribbon cable from the GPIO pins of the Raspberry Pi and reconnect the PiOLED temporarily. Alternatively, you can reconnect the Raspberry Pi to a keyboard, mouse, and monitor.

Connecting the Raspberry Pi and Breadboard

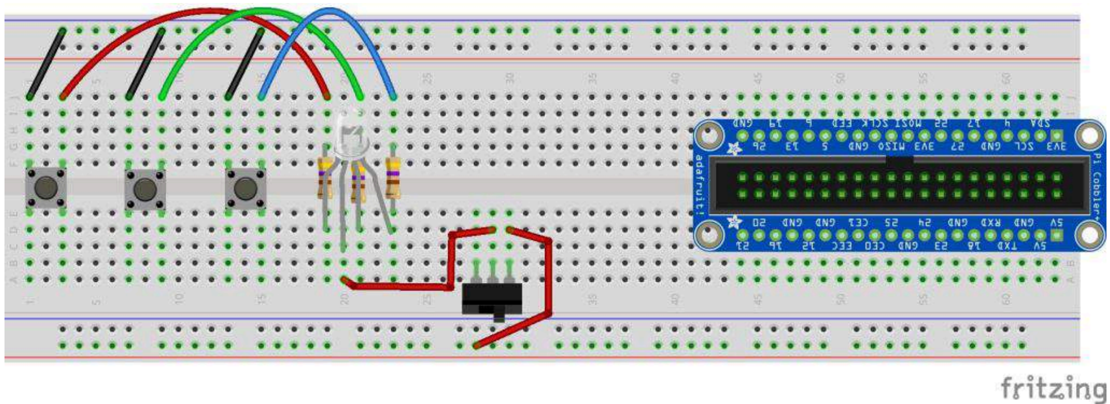
While the Fritzing diagrams in the first half of the Orange Unit were illustrated using a half-sized breadboard, these could be (and hopefully were) done using a full-sized breadboard, which would have been demonstrated with a Fritzing image as below. The unused space at the right of the breadboard is just what is needed for this part of the exercise.



1. Disconnect the USB-to-TTL cable and associated jumper wires from the breadboard.
2. Insert an Adafruit T-Cobbler Plus into columns 44-63 of the breadboard. Leave at least two rows accessible above and below the T-Cobbler.



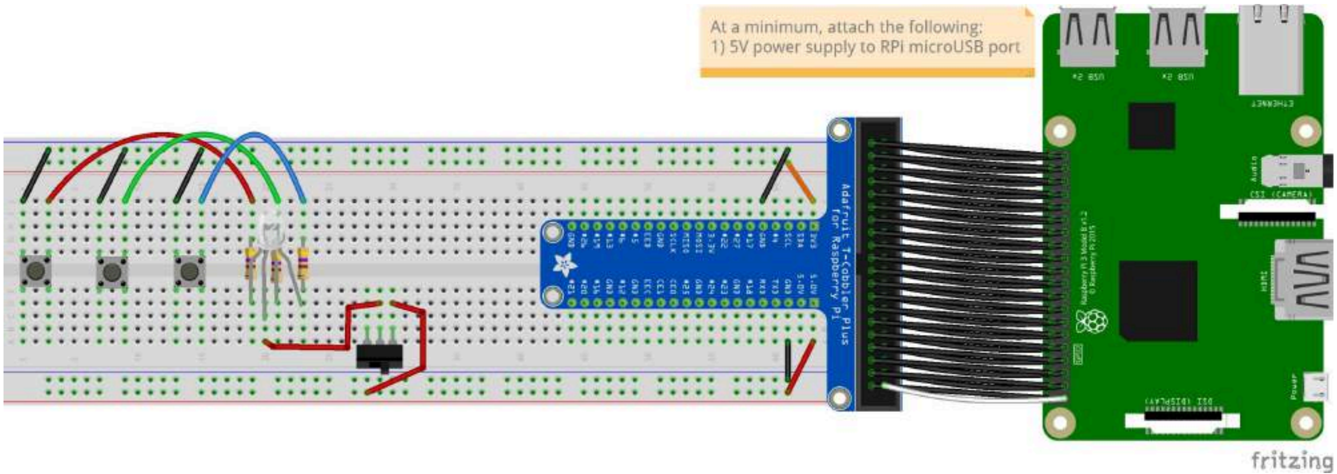
The T-Cobbler was chosen to ensure that the ribbon cable does not cover the breadboard holes and the GPIO pin labels associated with the top and bottom halves of each column. However, this results in an extended layout of full-sized breadboard, ribbon cable, and breadboard. For some, the Cobbler Plus may be the preferred alternative, keeping the full-sized length and instead adding the ribbon cable and Raspberry Pi to the height of the layout.



3. Use male-to-male jumper wires to provide 3.3 volts to the upper red rail, ground to the upper blue rail, 5 volts to the lower red rail, and ground to the lower blue rail.
    - The 3.3-volt GPIO pin immediately above the 5-volt GPIO pin below was chosen instead of the 3.3-volt GPIO pin in the middle of the T-Cobbler to keep the jumper wires to the right when working on later wiring. That said, both 3.3-volt pins and any of the ground pins can be used for this step.
    - Rapid prototyping begins with quick testing of circuits, with jumper wire placement of secondary concern. But wires can sometimes get in the way, and so in your layout of the base system for future exercises, spending a bit of time now to strategically create a platform for further work may be of value.

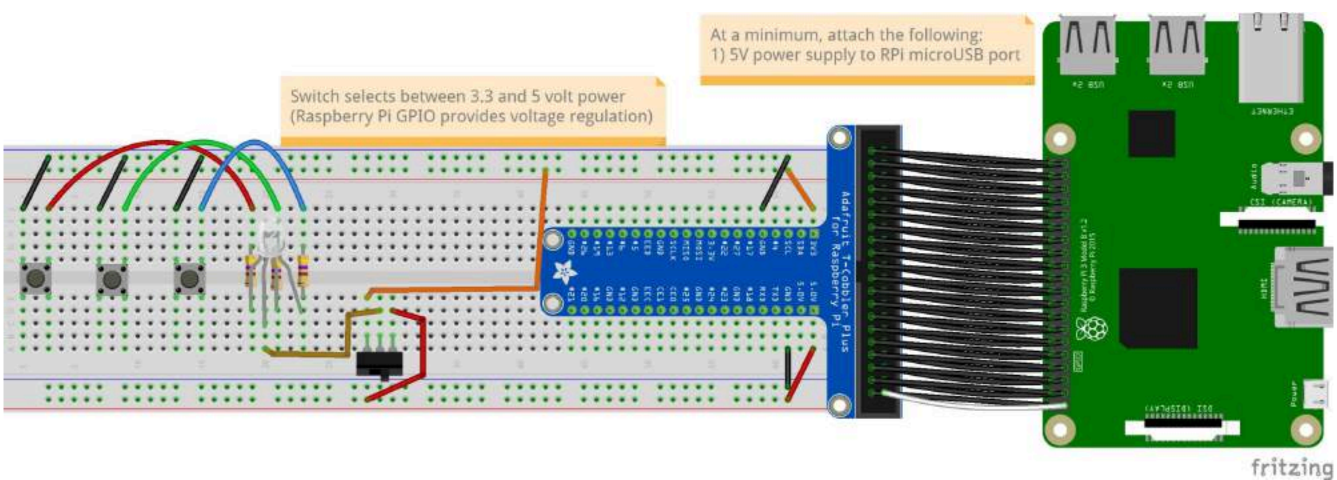
4. From here, you can insert a 40-pin ribbon cable between the breadboard and the Raspberry Pi.
    - A common practice for ribbon cables is to use a white non-conductive shell over the conductive metal wire that connects to pin one and black non-conductive shells for the remainder of the wires. For the Raspberry Pi GPIO, pin 1 is the 3.3V pin, which is now the upper half of column 63. A bump-out on the end-frame of the ribbon cable is also often used, and an associated gap is provided on the T-Cobbler to further help assure pin 1 is appropriately aligned.
    - Removing the ribbon cable does take some force. Holding it by the plastic frame or as close to it as possible while steadily pulling it from the T-Cobbler or up from the Raspberry Pi will help maintain the connections of the 40 wires with their metal connectors inside this frame.
    - The GPIO pins tend to bend a bit if the ribbon cable frame is leaned slightly while being removed from the Raspberry Pi. When possible, consider leaving the ribbon

cable attached to the Raspberry Pi and instead removing it from the T-Cobbler as needed.

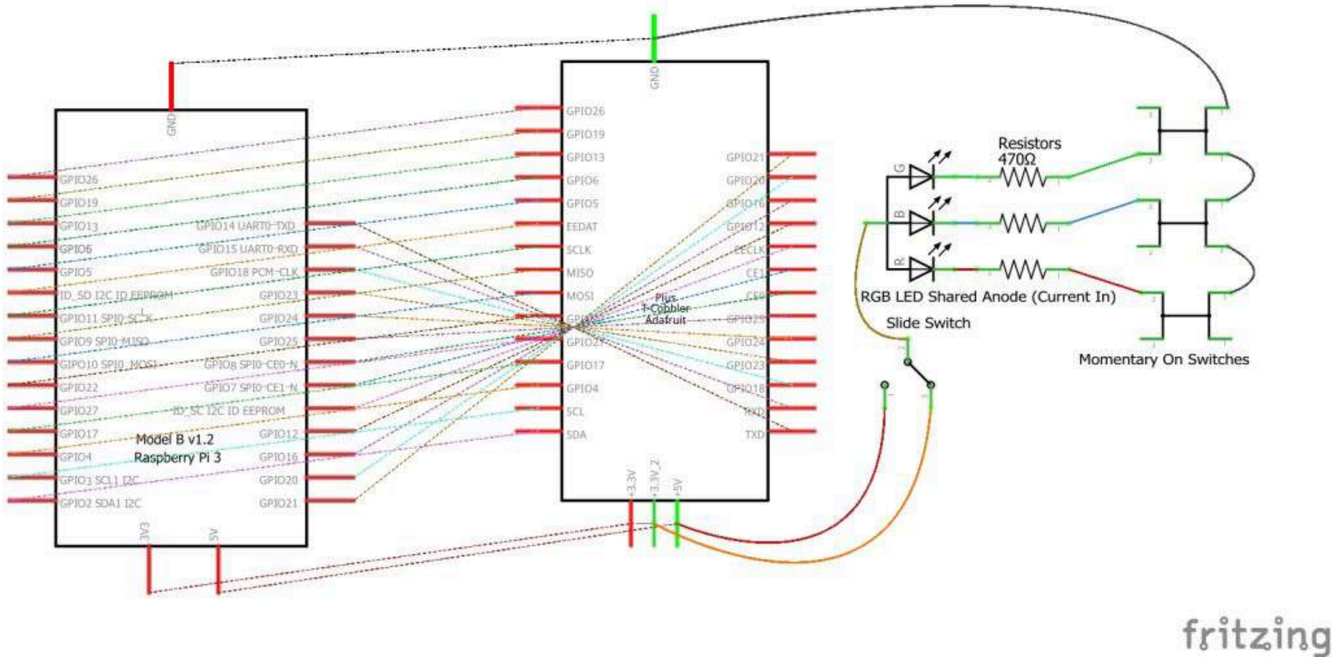


Try Something New!

The slide switch can continue to be used as a make-or-break switch to turn on and off 5 volts of current to the shared anode leg of the RGB LED. This would then provide a mechanical override of any selections made by code moving forward. However, given that the Raspberry Pi provides both 5-volt and downregulated 3.3-volt power options, the slide switch can also be used to mechanically select between voltages, helping to control the brightness of the lights by running a jumper wire from the 3.3-volt rail to the left throw leg of the switch, as shown below.



Schematically, the RGB LED, resistor, and switches within the breadboard circuitry remain the same. However, now they receive 5V and 3.3V power and ground through the Raspberry Pi by way of the T-Cobbler.



From here, consider using resistors to further control the lower level of brightness for that left side of the switch to match your preference for a dim setting.

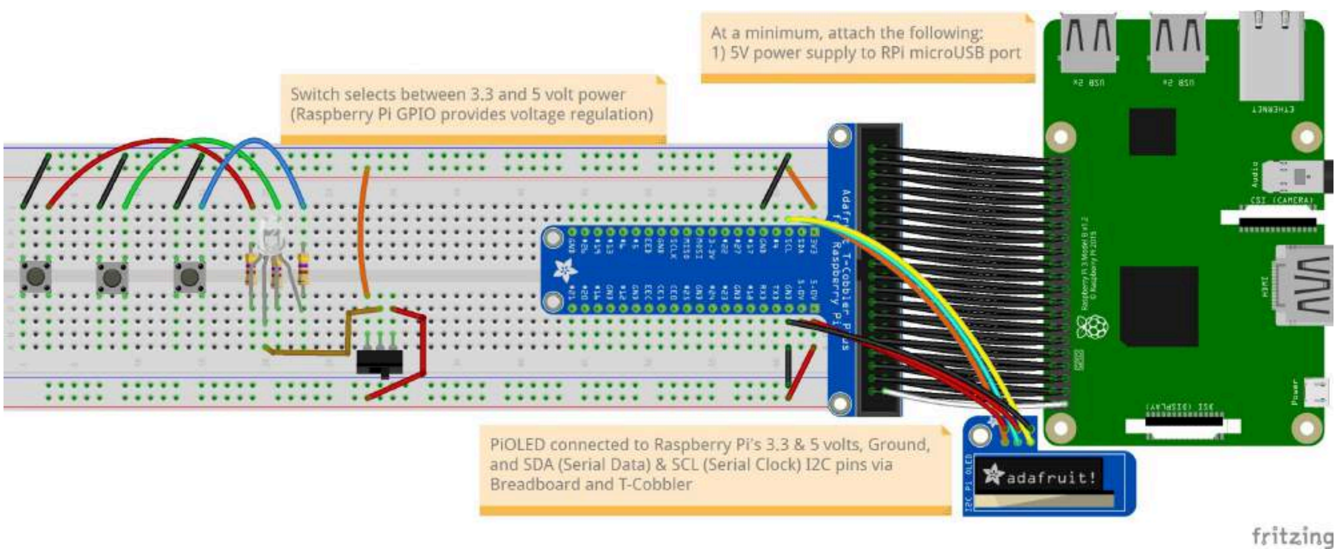
Try Something New!

There still may be those times when it is helpful to use the 128×32 pixels of the PiOLED to provide important data through code running on the Raspberry Pi, whether the stats are given through stats.py or some other information you'd like sent as part of upcoming exercises.

NOTE: Testing over two semesters and extended conversations with others who have used the PiOLED across the country have indicated that there is significant risk of improperly wiring a breadboard to the PiOLED, thereby resulting in short-circuiting of the PiOLED. This burned-out PiOLED circuit cannot be repaired. Therefore, this has been removed as a normal exercise and instead has been moved to a *Try Something New* activity. Use caution in proceeding in this selected optional task.

- To connect the PiOLED to the Raspberry Pi GPIO pins, you now need to run 6 male-to-male jumper wires from the breadboard to the PiOLED.
    - There is an Adafruit flower icon on the upper left side of the PiOLED. This points to the hole associated with 5-volt power.

    - The other two holes in the top row are a second 5-volt hole for power and a hole for ground, in that order.
    - The hole just below the 5-volt power hole is the 3.3-volt power hole.
    - The other two holes in the bottom row are the holes for SDA (serial data) and SCL (serial clock) **I2C** communications, respectively.
- Pin 1 of the PiOLED is the lower left hole in the drawing below. It is connected to a 3.3-volt source on the breadboard.
- Here it is connected to Raspberry Pi GPIO pin 1, which is the upper rows of column 63 of the breadboard. This design choice was made to help align the remainder of the pins of the PiOLED with the appropriate connections on the breadboard.
- Using an orange-colored jumper wire for the 3.3-volt connection, red jumper wires for the 5-volt connections, and a black jumper wire for the ground connection may help identify the connection points in the future.
- Once you've reached a successful point of "yes," consider using some electrical tape at the plastic frames of the jumper wires to keep these wires in their assigned layout at the point where they connect to the breadboard, and again at the PiOLED if you will only occasionally be using the PiOLED attachment.



Key Takeaways

The breadboard is a common way to do rapid prototyping. Many different electronic circuits may be tested in rapid succession. It's also possible to keep multiple breadboards with different versions of an electronic circuit on hand to take with you in testing by a diverse range of communities. While it's sometimes possible to also bring multiple Raspberry Pi's, this layout is meant to provide a means by which different breadboards can be attached and detached from a given Raspberry Pi in support of testing of multiple breadboard prototypes.

Digital Switches

So far, we've been working with mechanical switches. We've used the slide switch to move the flow of current to the anode leg of the RGB LED both between on and off states and between 5-volt and 3.3-volt power sources. We've also used the momentary "on" push button switch to temporarily complete the flow of current to ground, thereby turning on the associated red, green, or blue cathode legs of the RGB LED. Like light switches in our house, these mechanical switches can serve a valuable function in our daily lives.

But sometimes the use of a mechanical switch may prove inadequate for a task. Perhaps you still make use of the mechanical switch for regular use of a light, but need a safety switch to override a mechanical on/off switch to quickly turn on the light when a fire alarm sounds. Or maybe access to the mechanical switch is difficult and you'd like to move to a digital control through a phone app.

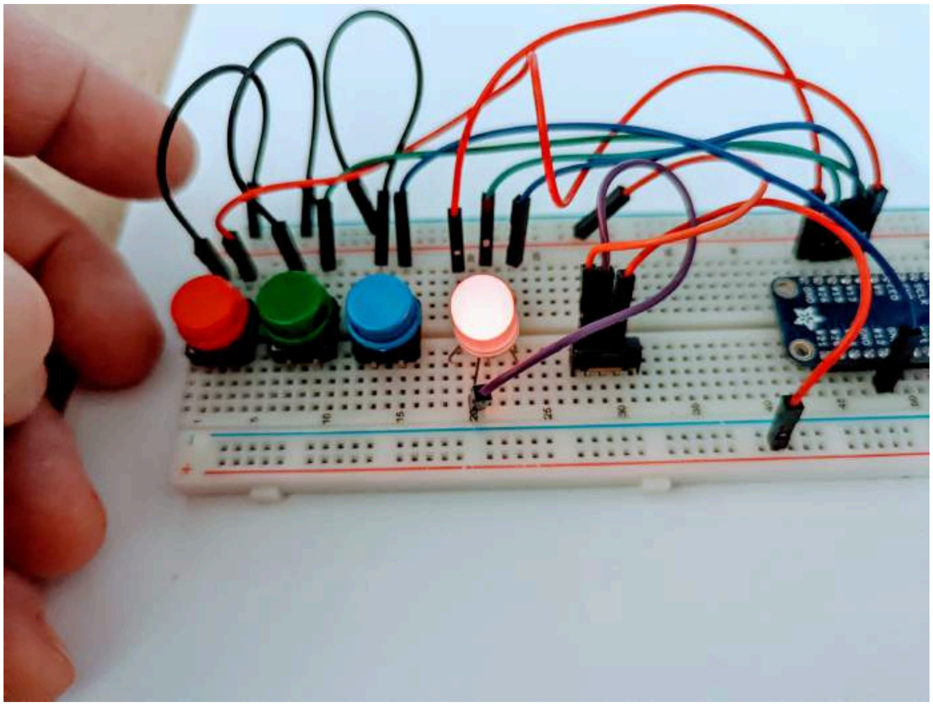
Digital Input/Output Pins vs. Power and Ground Pins

Before moving on, it's important that we look a bit closer at the labels on the GPIO pins:

3V3 Power	1	2	5V Power
SDA1 I2C	3	4	5V Power
SCL1 I2C	5	6	Ground
GPIO #4	7	8	UART0 TXD
Ground	9	10	UART0 RXD
GPIO #17	11	12	GPIO #18
GPIO #27	13	14	Ground
GPIO #22	15	16	GPIO #23
3V3 Power	17	18	GPIO #24
SPI0_MOSI	19	20	Ground
SPI0_MISO	21	22	GPIO #25
SPI0_SCLK	23	24	SPI0_CE0
Ground	25	26	SPI0_CE1
I2C ID_SD	27	28	I2C ID_SC
GPIO #5	29	30	Ground
GPIO #6	31	32	GPIO #12
GPIO #13	33	34	Ground
GPIO #19	35	36	GPIO #16
GPIO #26	37	38	GPIO #20
Ground	39	40	GPIO #21

In particular, compare those whose labels start with "GPIO #" to those labeled as "Ground." We've specifically used labeled "Ground" for the blue grounding rails on the breadboard. Those labeled "GPIO #," on the other hand, or not designed as a ground source, but are instead general-purpose **digital** input/output pins. Digital **inputs and outputs** (IO) are used on microcontroller and microcomputer GPIO pins for binary on or off logic. In code, this is sometimes written as binary High and Low or as digital True or False. The current used for digital IO is relatively small and can include negative as well as positive voltage. This can be important when pushing a speaker cone out with a DC signal that varies between 0 and some positive voltage before pulling the speaker cone back with a DC signal that varies between 0 and some negative voltage. The general-purpose digital IO pins provide a perfect match to the momentary and transistor switches.

As the GPIO digital IO pins do move from positive voltage to relative zero voltage, we could just send our digital outputs straight to the 470 Ω resistors connected to each of the three cathode legs. But as this is not true ground but a relative zero voltage, it is possible there could be a slight glow of color on an LED, even though the digitally switched circuit is an open circuit. The relative difference between power and ground is insufficient to assure the LED is off in keeping with the decision tree within the code.

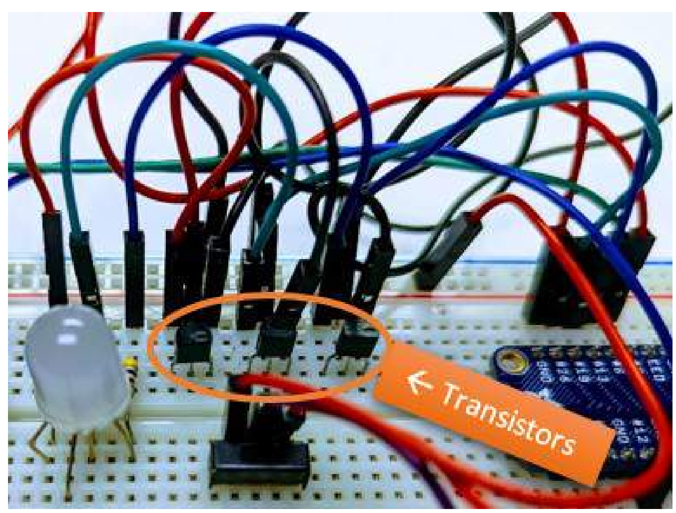


Digitally switched RGB LED with red, green, and blue legs of the RGB connected directly to their assigned GPIO ports.

Digital switches come in multiple classes and types. To address the misuse of the digital IO ports as a ground source instead of its intended digital High or Low role, we will use as digital switches transistors—solid-state electronic devices used as part of much of our daily life electronics, which are designed specifically for tasks like this. We'll place these between each of the three cathode legs of the RGB LED and their assigned GPIO ports.

More on Transistors

Starting in the 1960s, a new term was heard in relation to radios, calculators, and televisions. I remember the commercial that inspired our family to get its first "transistor radio." I also remember going to the five-and-dime or drug store to purchase a vacuum tube until we finally got that transistor television. Transistors replace those tubes by providing a semiconductor device having at least three legs: a source current terminal; a drain current terminal; and a gate terminal, which controls the output current.



At its most basic level, the **transistor serves as a switch**. While our momentary switch circuits above required a physical act, the pushing of a button or the sliding of a switch, a transistor switch uses an electrical input to determine the flow of current. Further, transistors are designed such that the switching voltage is less than or equal to the voltage of the current flowing through the transistor. The 3.3- or 5-volt output from the Raspberry Pi GPIO pins can be used to turn on and off not only a 5-volt LED, but also a 40-volt DC motor! In this way the **transistor can also serve as an amplifier**. "In fact the initial tests of the first transistor involved hooking a speaking to its output and hearing that it was louder than the input. Every modern piece of audio circuitry has transistors at its heart amplifying signals." (Dave Astels, "Transistors 101": https://learn.adafruit.com/transistors-101/basic-operation)

Types of Transistors

Transistors are built using two different types of semiconductor material. Unlike a jumper wire that is always conductive, a **semiconductor** is a substance that can conduct electricity under some conditions, but not under others. An **n-type** semiconductor is mostly made of negative electrons. A **p-type** semiconductor is mostly made of positive charge carriers.

When a transistor uses both n-type and p-type semiconductor material, and only those two types of semiconductor material, it is called a **bipolar transistor**. The bipolar transistors are three-terminal devices with the standard Emitter, Base, Collector (EBC) pinout. The emitter and collector on either end of the transistor are of the same type of semiconductor, while the center base is of the other type of semiconductor. In this way, the base provides a **junction** between the collector and emitter. When a biasing voltage is applied to the base terminal, the junction breaks down, and current can flow from the collector to the emitter.

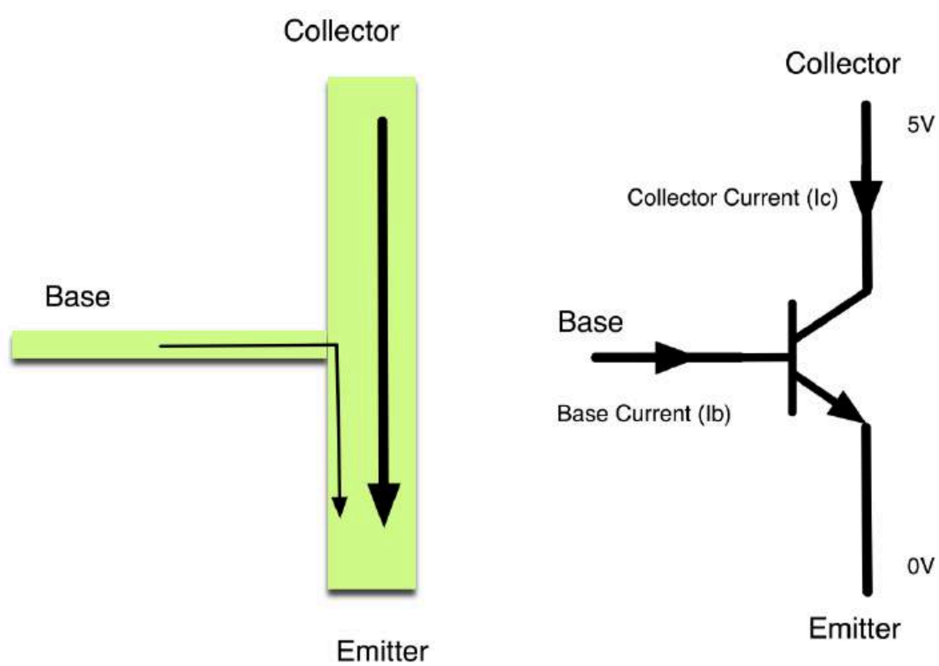
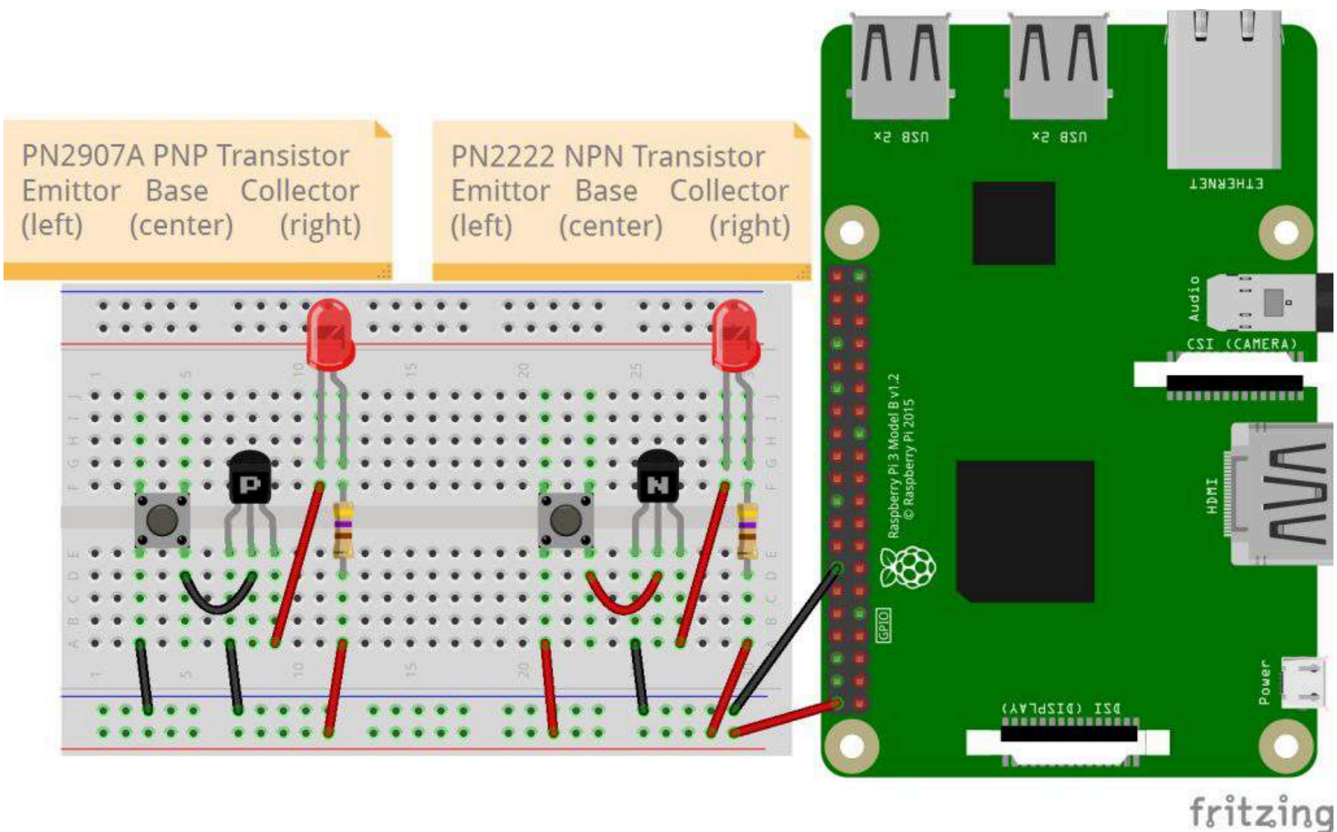


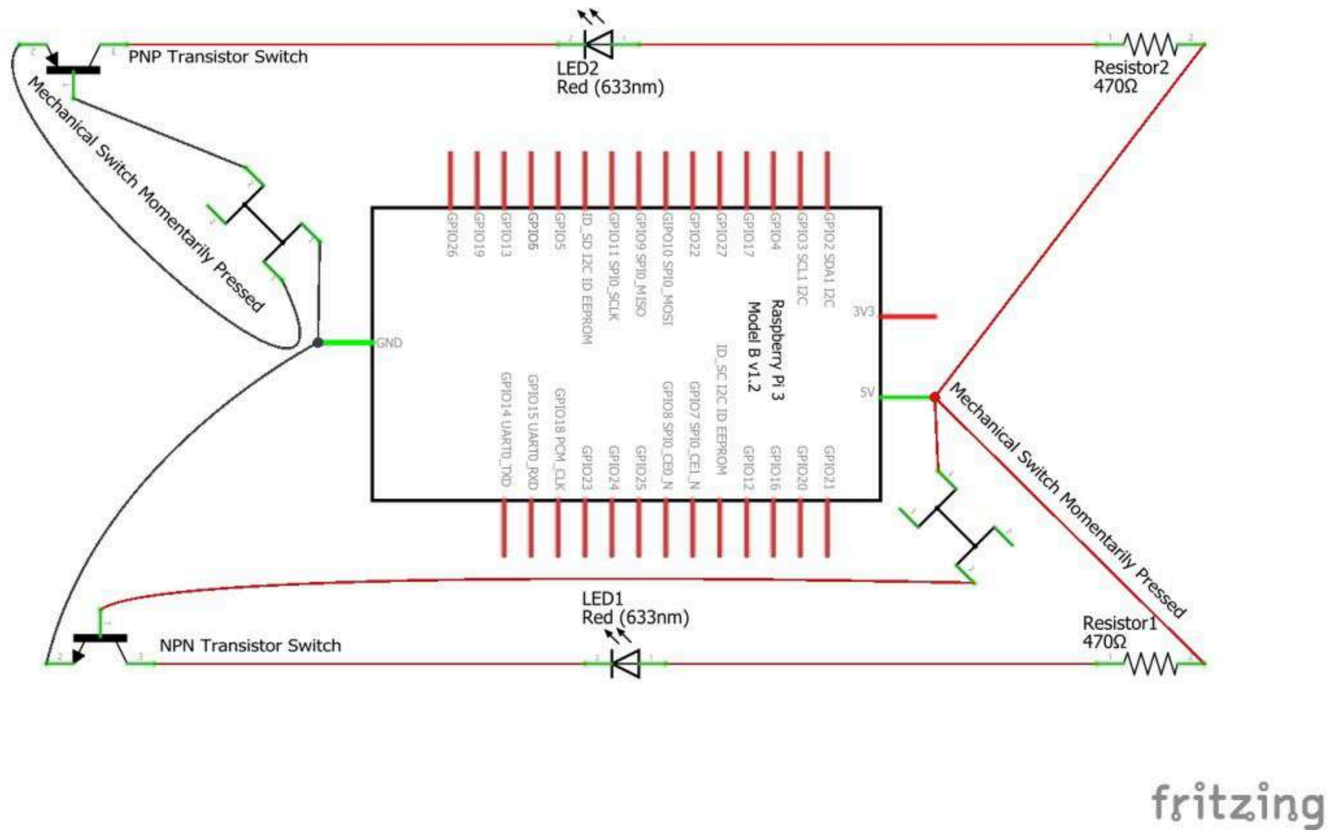
Image by Simon Monk, [Arduino Lesson 13. DC Motors: Transistors,]Adafruit. Licensed under [CC BY-SA 3.0].

As we've been doing with the momentary switches of our parallel red, green, and blue RGB LED circuits, we can use three bipolar transistors as switches to determine the state of each circuit. To do this, we need to decide between the two different configurations of a basic bipolar transistor.

- The **NPN** bipolar transistor has a collector and emitter made of N-type negative electrons and a center base made of P-type positive electrons. For our LED circuits to have current flow from the cathode leg of an LED to the collector and emitter of the transistor and thereby to ground, voltage is sent to the base leg of this type of transistor to turn on its gate.
- The **PNP** bipolar transistor has a collector and emitter made of P-type positive electrons and a center base made of N-type negative electrons. For our LED circuits to have current flow from the cathode leg of an LED to the collector and emitter of the transistor and thereby to ground, ground is sent to the base leg of this type of transistor to turn on its gate.

Below is a Fritzing drawing and associated schematic in which a momentary switch is used to provide ground to the PNP transistor base and 5 volts to the NPN transistor base. Let's now use this information as our starting baseline for designing and prototyping our programmable RGB LED switched circuitry.





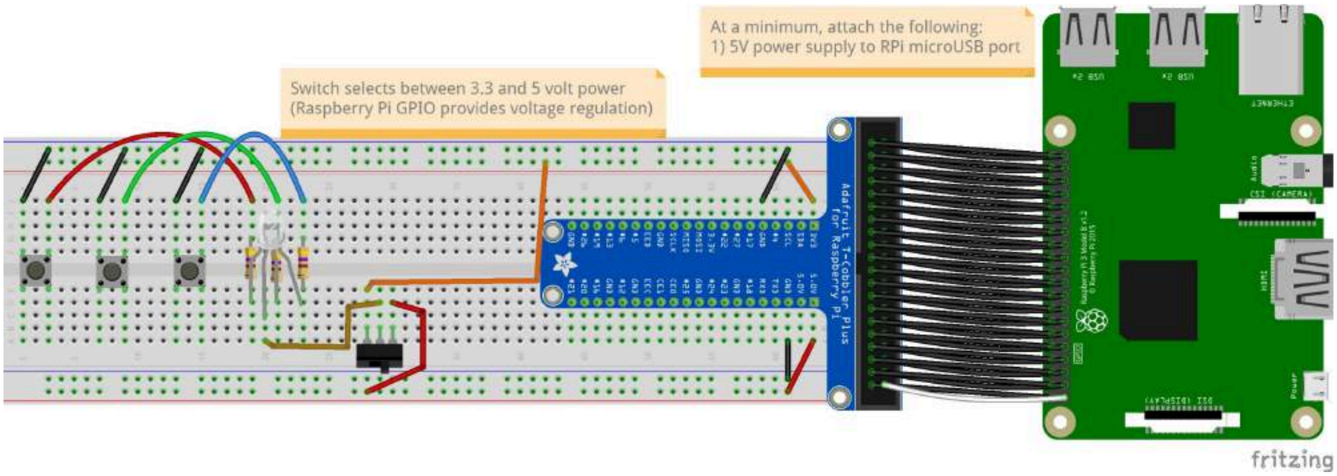
Exercise: Control LEDs with Python Code

With the GPIO pins on a Raspberry Pi readily accessible via a Cobbler or T-Cobbler on the breadboard and with a rudimentary understanding of PNP and NPN bipolar transistors, we're ready to rewire the electronics to use Python code to input data from the momentary switches and to output data to transistor switches connected to the three legs of the RGB LED.

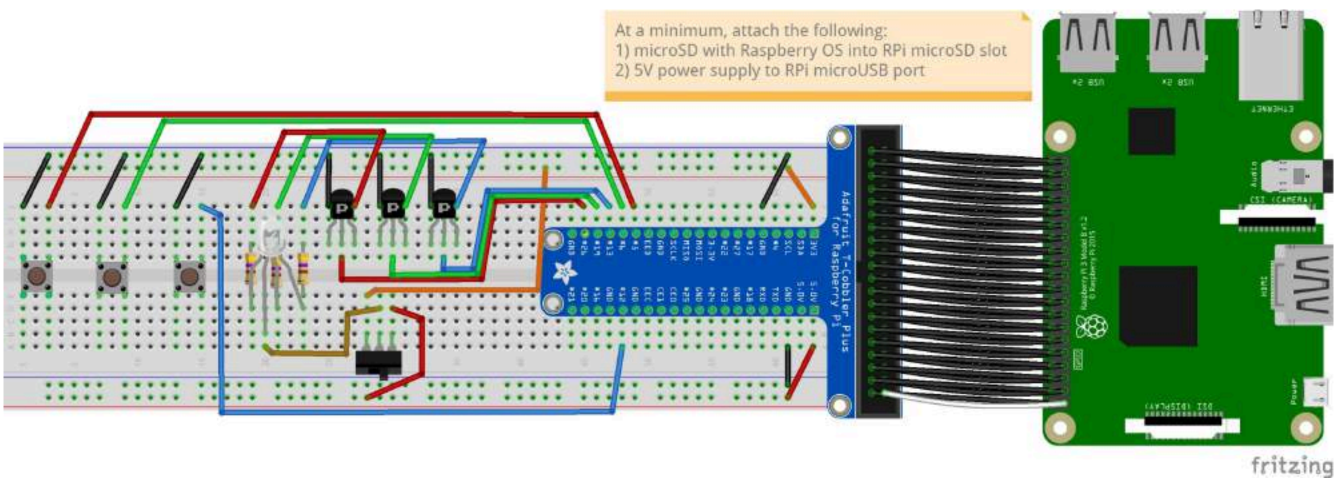
For this exercise, we'll use GPIO #5, #6, and #12 for the momentary switches, and GPIO #26, #19, and #13 for the base of the transistor switches to be integrated into programming code. The grouping of these pins was selected to roughly match electronics with GPIO pin groupings. But other digital IO pins with a "GPIO #" hash label at their start could have also been used. What works for one may not be ideal for others, which is where alpha and beta testing, along with open channels for feedback after implementation, come in. Feedback from past students doing these exercises has proved essential in improving functionality for diverse audiences.

Part 1: Steps to Rewire the Circuits

Here's the configuration of our current circuits (without sequin LEDs) as a reminder of where we're starting:

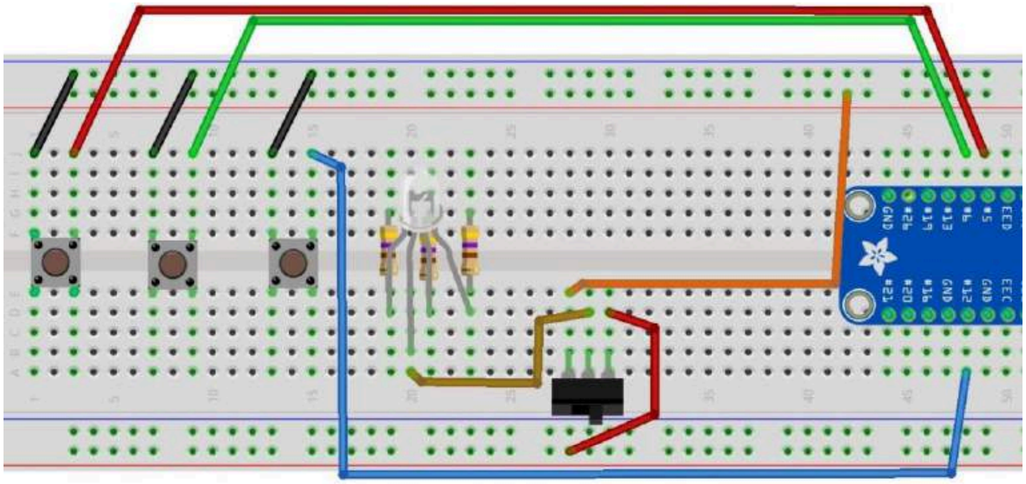


And here's the Fritzing image we're working towards:



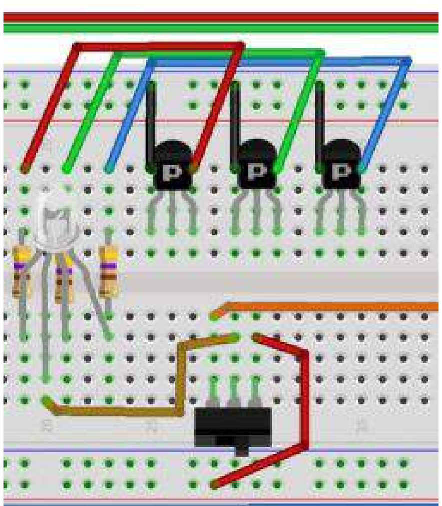
Instead of using the pressing of momentary switches to directly complete the flow of current from power to ground by way of an associated leg of the RGB LED, a Python program will now continuously collect digital inputs from each momentary switch to determine if they are pressed or released. The code will also continuously send output to the base of each transistor to inform them of the state of the associated momentary switch, thereby electronically switching the red, green, and blue LEDs between on and off.

1. Repurpose the three male-to-male jumper wires connecting the three momentary switches to the associated red, green, and blue cathode legs of the RGB LED. Connect the right legs of each momentary switch with its associated GPIO digital input pin.



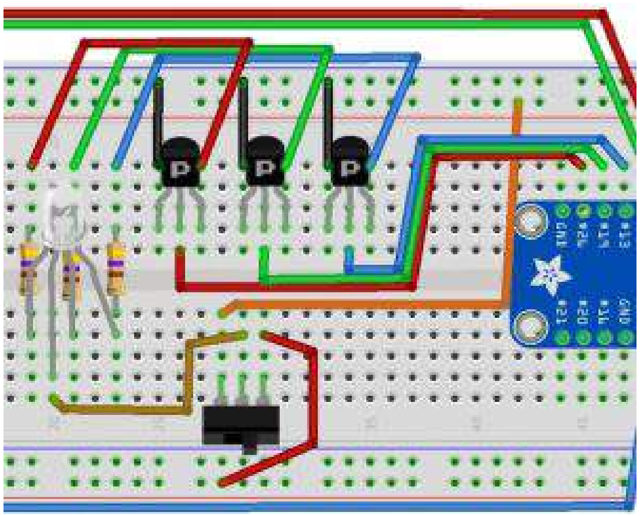
- Move the jumper wire connected to column 19 to instead connect to the upper rows of column 49, which is associated with GPIO #5. Leave the wire connected to column 3, bringing together the "red" push button with GPIO #5.
- Move the jumper wire connected to column 21 to instead connect to the upper rows of column 48 which is associated with GPIO #6. Leave the wire connected to column 9, bringing together the "green" push button with GPIO #6.
- Move the jumper wire connected to column 23 to instead connect to the lower rows of column 48, which is associated with GPIO #12. Leave the wire connected to column 15, bringing together the "blue" push button with GPIO #12.
- Be sure to leave the male/male jumper wires connecting the left legs of the momentary switches with the blue ground rail on the breadboard.

2. Add three PN2907A PNP bipolar transistors to the breadboard, connecting the collector legs to the associated red, green, and blue cathode legs of the RGB LED and the emitter legs to ground. These should be placed in the upper rows of the breadboard, with the flat face of each transistor facing the lower rows. In so doing, the emitter leg will be on the left side of the transistors and the collector leg on the right side of the transistors.



- Place one transistor in the upper rows of columns 25, 26, and 27. Connect column 25 to the blue ground rail. Connect column 27 to the upper rows of column 19.
- Place a second transistor in the upper rows of columns 29, 30, and 31. Connect column 29 to the blue ground rail. Connect column 31 to the upper rows of column 21.
- Place a third transistor in the upper rows of columns 33, 34, and 35. Connect column 33 to the blue ground rail. Connect column 35 to the upper rows of column 23.

3. Finally, connect the base of each transistor with its associated GPIO digital output pin.



- Connect the transistor base found in column 26 with the upper rows of column 45, which is GPIO #26.
- Connect the transistor base found in column 30 with the upper rows of column 46, which is GPIO #19.
- Connect the transistor base found in column 34 with the upper rows of column 47, which is GPIO #13.

In practice, when doing any of these breadboard exercises, your implementation may look a bit different than the Fritzing image to adjust for the actual components at hand. The key is to achieve the designated associations that we'll next make use of within a Python program.

Part 2: Steps to Run the Code

For Part 2, we'll now be working in the Raspberry Pi OS. While this can all be done from a terminal window (the standard Linux command-line Interface, or CLI), for most this is best done through a graphical user interface (GUI) using a monitor, keyboard, and mouse. These can either be directly connected to a Raspberry Pi, or as demonstrated in Orange Unit 4C, by using a RealVNC Viewer on your personal computer.

1. In the web browser on the Raspberry Pi, enter:
   https://go.illinois.edu/switchLED
2. Download the file “switchLED.py” to the Raspberry Pi. You do NOT need to log in or sign up to successfully open and download the Python code.
3. Click on the “File Manager” icon in the Raspberry Pi Desktop Menu. Move switchLED.py from the Downloads folder to the /home/pi/Code/Python folder.
4. Double click on the “switchLED.py” file. By default, files that end with .py will be opened with Thonny, a popular Python integrated development environment (IDE) that comes pre-installed on the Raspberry OS Desktop.
   - Click on the Run icon in Thonny. This is a common method within integrated development environments to test out new code. As such, you will see in the “Shell” window at the bottom the same display that would be shown if this code were being executed in a terminal window in practice moving forward.
   - Click on the Stop icon in Thonny, and then close the Thonny window.
5. Click on the “Terminal” icon in the Raspberry Pi Desktop Menu. In terminal, type:
   ```
   cd /home/pi/Code/Python
   python switchLED.py
   ```
   You’ll now fully see the text provided by the switchLED.py code, with instructions for using the programmable switch. It also provides instructions to hold down the <CTRL> key and press the “c” key to exit this program.

Key Takeaways

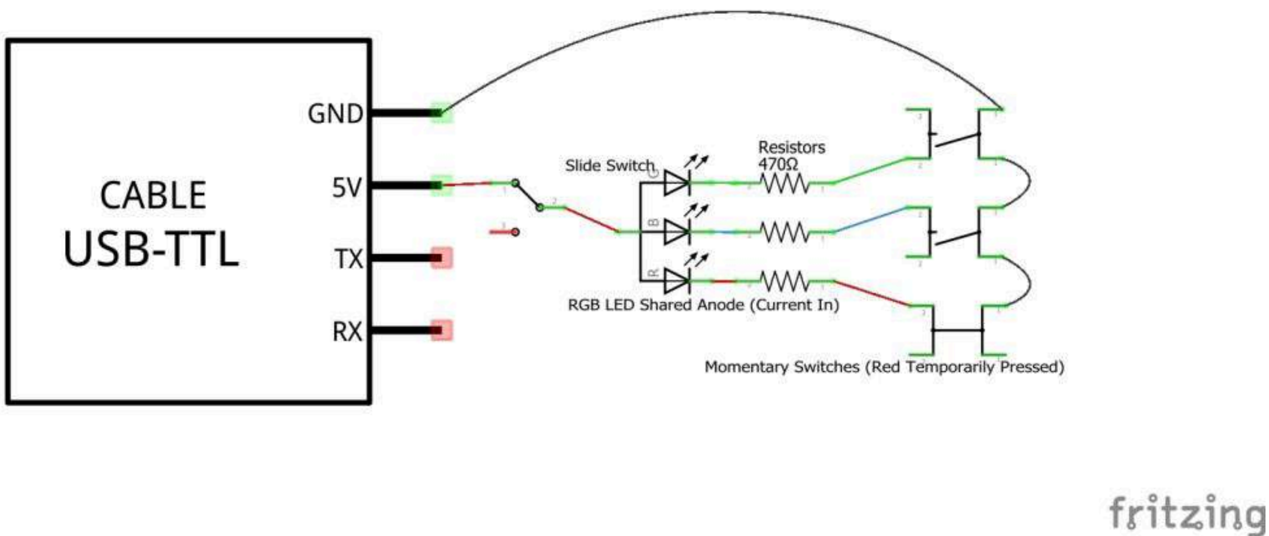
Take a few minutes to go back to Thonny and skim through the code, especially noting:

1. The three groupings of components associated with the red, green, and blue parallel circuits.
2. The outputs are designated by the associated LED that will be lit. The inputs are designated by the associated momentary switch that is mechanically pressed and released.
3. Your Raspberry Pi GPIO # should match the listed board.D number, being assigned within the digitalio.DigitalInOut library object definitions. For instance, the right leg of the “red” momentary switch should be connected to GPIO #5 to match board.D5 being assigned to switchR.
4. At the bottom of the code, you’ll find the line “while True:” which starts a Python forever loop. The next three lines will continuously be run to update ledR.value to match the current switchR.value.
   - When a momentary switch is pressed, the digital input has a flow to ground. This sets the current to ground, which is identified as a binary zero. And as setup

within the digitalio library, this is identified as a digital False.

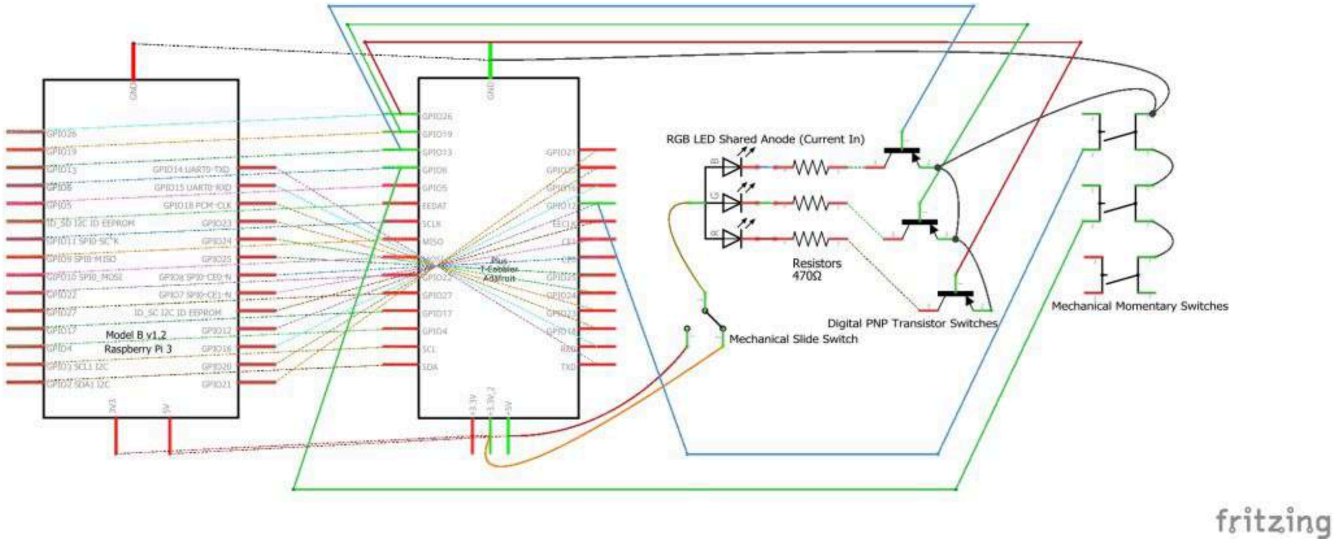
- When the ledR.value is next updated with the switchR.value, it will now be set to digital False/binary zero as well.
- When the digital False/binary zero output reaches the base of the red transistor, the red rail's 5-volt current, which flowed through the shared anode leg to the red cathode leg and through the 470 Ω resistor to the collector leg of the transistor, can now flow to the emitter leg of the transistor and then to the blue ground rail. The red circuit is closed, and the light comes on!

In previous sessions of the Orange Unit, circuits were constructed using electronic components without code.



Looking at the new programmable code schematically, we can see three parallel momentary switches and three parallel LEDs as before. But each momentary switch is no longer directly, physically in series with its associated LED. Circuits are instead dynamically constructed and completed through code. This schematic shows us only the electronics of our new system. Code is not shown in schematics, as it can be readily changed to do different acts with the same physical structures. We are now truly opening the literal and conceptual hood of technology in a way that hopefully begins to demystify not just this example, but also more of your electronic components used as part of your daily life.

As mentioned at the start of this activity, we chose to use the Raspberry Pi GPIO #5, #6, and #12 for the momentary switches, and GPIO #26, #19, and #13 for the LEDs to be integrated into programming code. **In this case, the Raspberry Pi's GPIO and associated electronics and embedded code together create one, more complex circuit**. Code, in this case using the Python programming language, is used to determine the activities based on the state of the momentary switches and the design decisions controlling the state of the LEDs.



In what ways do you find the following true? False? As we move from circuits made exclusively of physical electronic parts to ones that also include programming code, greater opportunities for innovation-in-use become available.

Opening the Hidden Box of Today's Keyboards

We've been steadily exploring different parts of basic electronic circuits, leading to the most recent exercises using a microcomputer to read inputs from a switch when it is pressed to complete a circuit. The microcomputer then provides a designated response, sending a ground signal to the associated cathode leg of an RGB LED to complete that circuit, thereby turning on that color of light.

The Raspberry Pi is a general-purpose microcomputer, allowing us to make use of a range of unassigned GPIO pins to create the electronic circuits and to edit and run Python code to work with these circuits. Take a moment to consider how many other applications are also running right now on the Raspberry Pi.

While general-purpose computers provide considerable flexibility, there are times when single-use computers and microprocessors are instead used. As one example, consider that which is used within a typical keyboard. Each key on the keyboard is physically a momentary switch push button and performs the same base function as that on our own coded circuit. But imagine what this might look like if there were 100+ parallel circuits being used, one for each key of your keyboard. That would be a lot of individual pins going into that microcontroller I/O port!

It's for this reason that the keyboard uses a key matrix. Conceptually visualize your keyboard's key matrix like a spreadsheet with rows and columns. Maybe you have a spreadsheet with 6 rows and 18 columns, providing you with 108 cells. By creating a cell map, when you see that one cell has just changed color, you just need to see which row and which column are associated with that cell, and then look at the cell map to determine what's up.

As I look at the keyboard I'm using right now to type this text, I can see that my left pinky finger is hovering over row four, column four, starting from the upper left of the keyboard. That is, it is hovering over the letter "a" of my keyboard. If I move my right pinky finger to row 5, column 15 of my keyboard, I would hold down the shift key before pressing row four, column four, to get the letter "A" of my keyboard.

> Instead of needing 108 IO pins for the keyboard microcontroller, I now only need 6 + 18 = **24** IO pins, plus an associated software character map.

Keyboards are a great case study in how basic circuit building, combined with strategic electronic layout design and data mapping, opens a wide range of human-centered design possibilities. For the Raspberry OS and the Raspbian Configuration program, this means we can quickly change the character map for any attached keyboard to make it work appropriately for the language of the user. Even if the keyboard microcontroller has a different character map with which it associates points on the key matrix of the keyboard, operating system software can often override some or all of that character map. Of course, this doesn't change the physical characters written on the keyboard itself. But that doesn't hold back someone who is a touch typist looking at the screen to see which momentary switch push button was just pressed.

One other note of import regarding our review of the keyboard as an introductory case study. The microcontroller inside the keyboard case may have 24 IO pins (probably more, as some are used as outputs for LEDs, and still others to communicate with the keyboard). But a wired keyboard cable is typically only providing four wires: the ground, power, transmit, and receive associated with the USB, or Universal Serial Bus, connection. This serves as another great example of how the electronics around us are often made with a mix of parallel and serial circuits joining and breaking away from other parallel and serial circuits. The keyboard, then, is a helpful case study in how various simple serial and parallel circuits artifacts come together to create a somewhat more complex set of serial and parallel circuit artifacts. Together, these become the system of a keyboard!

Wrap Up

We've gone from basic LED circuits in parallel using different levels of resistance and different voltages, to LEDs in series controlled by momentary switches, to integrated circuits, to code-based circuits. We have also taken our first dips into the Raspberry Pi microcomputer pond to see how these electronic parts, along with a range of other electronics, come together to create large, complex systems like computers.

A wide range of individual electronic components can be brought together using power and ground sources and conductors to build individual circuits, like the red, green, and blue LED circuits we created using the RGB LED with common anode. Using a printed circuit board, electronic components can be put together to create an integrated circuit, like the Sequin LED integrated circuit. Integrated circuits and additional electronic components can be further put together with code to create limited purpose microcontrollers like the Circuit Playground Express we'll use in the Blue Unit, and general-purpose microcomputers like the Raspberry Pi.

We've begun to discover that software can open opportunities for further innovation-in-use, as is the case with different character maps for one keyboard key matrix. And in the character map example, we've also seen how software can at the same time restrict innovation-in-use, as is the case when an operating system or other software overrides a selected character map with its own standards for keyboard character associations. The configurations of our sociotechnical devices are multi-tiered hardware, software, and social constructs that are far from neutral. These constructs can come into conflict as data and information regarding past design choices are lost or hidden from viewing by current innovators. They shape current work, even as the current work seeks to further shape these sociotechnical devices.

As mentioned in a video segment at the beginning of this session, my history with computers goes back to the mid-1970s in junior high, and my work with general electronics even longer, as I grew up in my family's sawmill. In this next video, I share my memories of electronics and programming during the late 1980s and early 1990s, as I worked on my research as a PhD student at Rutgers. You may find some similarities to the work we've just done! [Watch the video "My History with Computers: Rutgers, 1987-1993" online.]

Comprehension Check

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with answers, see the online version.]

While we do an increasing amount of online work using mobile phones and tablet computers, laptop and desktop computers have remained a regular part of our everyday lives, whether at work, school, or home. For many, this is the first time you've worked with the physical printed circuit board and electronics hands-on, as many of these everyday devices are locked into their cases. Still, our everyday laptops and desktop computers have much in common with the Raspberry Pi.

Which of the following features, if any, are common to the Raspberry Pi and other computers you've used? Select as many as apply.

NOTE: Of below, only external storage devices are less frequently seen with desktop and laptop computers today, although you may have used these for school and work to transfer files.

- Typing tool, such as a keyboard or touchscreen
- Selection tool, such as a mouse of touchpad
- Display screen
- Desktop (Graphical User Interface)

- External storage device
- Case
- Operating System
- Tools of updating, installing, and running general software
- File manager
- Capability of Internet connectivity
- System board
- Electronic components, integrated circuits, and microprocessors
- Input/output ports
- None of the above

Orange Unit Review

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with the option to export your responses, use the online version.]

Key Learning Outcomes Self-Check

Listed below are the key learning outcome objectives for the various skill sets we sought to advance in the **Orange Unit** of the book. Using the scale below, take a moment to create a second ***1-4 rating*** of these items based on where you think you are in your skills development now that you've completed a pass through the Orange Unit cycle. Add additional notes as helpful to keep further specific reflections on your development of these specific skills.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough
4: Yet

Technical Skills

You can (rate 1-4):

- Identify core electronic components, such as breadboards, resistors, and diodes.
- Read diagram and schematic drawings of circuits and use them to prototype your own circuits.
- Identify parts of the circuit that are breaking the loop, thereby interrupting the functioning of the circuit.
- Identify the hardware and operating system building blocks that make up all computers.

Information Skills

You can (rate 1-4):

- Explore a range of information sources, guides, and worksheets in support of current research and practice.

Cognitive Skills

You can (rate 1-4):

- Generalize past and current problem solving to new contexts.
- Logically analyze and organize project tasks in ways that allow use of digital tools to help accomplish them.

Socio-Emotional Skills

You can (rate 1-4):

- Communicate and collaborate with others.
- Embrace failure as an essential step in project development, plan for failure, and use failure as a stepping stone (fail forward mindset).
- Analyze current skills, compare to needed skills to accomplish a project, and develop a plan of action to achieve skills development (growth mindset).

Progressive Community Engagement Skills

You can (rate 1-4):

- Use an informal, distributed process to share the responsibility of leadership among all team members of a community of practice in ways that build off each member's strengths and unique insights.
- Understand and respect broad diversities of functioning with regard to analog and digital technologies, and to see this functional diversity not as a deficit, but rather as an inherent and valued aspect of being human.
- Interact as a community of practice in ways that advance each member's capability set and positive freedoms, so as to more effectively achieve and enjoy their valued beings and doings.
- Focus on and learn from the array of often unrecognized and unacknowledged cultural knowledges, skills, abilities, and contacts possessed by a community of practice, and especially those of socially marginalized members.

Critical Sociotechnical Skills

You can (rate 1-4):

1. Analyze how economic, social, cultural, historical, and political contexts, and personal preferences and biases, are embedded into a specific implementation of technology throughout the life cycle of the technology.
2. Evaluate positive and negative impacts that different technologies and specific implementations of a technology have on individuals and social groups from different economic, social, and cultural contexts.

Revisit Your Personal Learning Outcome Objectives

Add the personal learning outcome objectives you created at the beginning of this Unit.

Rate Your Personal Learning Outcome Objectives

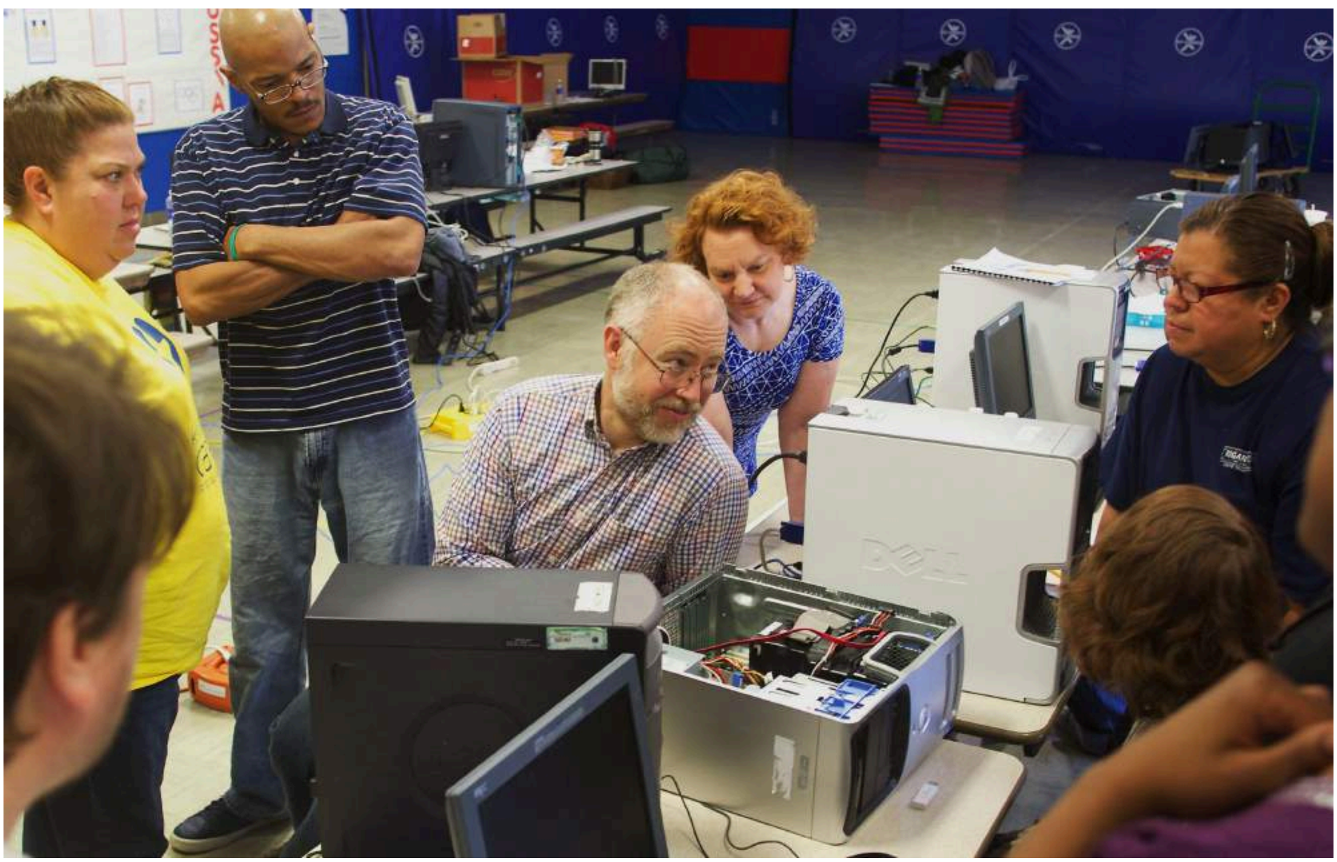
Take a moment to create a second ***1-3 rating*** of these items based on where you think you are in your skills development now that you've completed a pass through the Orange Unit cycle. Add additional notes as helpful to keep further specific reflections on these specific skills development.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough

Goals	Rating
Your first goal...	

Blue Unit: Computational Tinkering



At a Demystifying Technology Workshop, elementary school parents gather around a desktop computer which has its cover removed. Author Martin Wolske sits to the left of the computer.

Table of Contents

Blue Unit Overview

Humans are analog, while computers are digital and binary – zeros and ones. As we will begin exploring in the Blue Unit, part of our work with the following coding activities is to perform translations between analog and digital. However, it's important to realize that sometimes the analog materials cannot be replaced by digital. And sometimes, even if the digital could replace the analog, this translated version may not serve as well as the analog version. There are also times, though, when it proves best to either use the digital translation or to use information that is collected digitally. In this section, we will begin using electronics and human analog tools as well as digital-only materials to begin doing computational thinking in new, creative, inclusive ways by using computational tinkering.

Tinkering, the making of relatively small changes in order to improve or repair, is both a part of everyday life and also a historic way for sustainable living. While dictionaries recognize that it is often associated with a person living in an itinerant community, from my personal experience, this also includes members of non-traveling communities as well, as some of my close relatives and I identify as tinkers.

Makerspaces are places to both respect traditional crafts and to also explore, create, invent, and learn using digitally-based crafts that, too, can reflect a tinker mindset. Creative tinkering studios are starting to emerge around the country, such as the Tinkering Studio at the Exploratorium in San Francisco, California, meant to allow museum visitors to slow down and deeply engage in the investigation of scientific phenomena in ways that, at the same time, open up creations representing their ideas and aesthetics. Project Computational Tinkering at the MIT Media Lab has worked with the Exploratorium's Tinkering Studio to explore computational thinking concepts and practices through both analog and digital materials to teach and explore the important skills of communication, collaboration, creativity, and critical thinking as well as the design thinking skills of inspiration, ideation, and iteration. They define the term computational tinkering as "bringing together decomposition, pattern recognition, abstraction, and algorithms of computational thinking with tinkering's idea generating, designing, personalizing, expressing, remixing, collaborating, questioning, reflecting, and iterating."

The International Society for Technology in Education (ISTE), the Computer Science Teachers Association (CSTA), and a range of computer science teachers, researchers, and practitioners collaborated to create an operational definition and series of standards, defining computational thinking as a problem-solving process that brings together computer and other tools with critical and algorithmic thinking, confidence in dealing with complexity, tolerance for ambiguity, and the ability to communicate and work with others to achieve a common goal or solution.[1] Moreover, initiatives in K-12 education, such as Computer Science for All, focus on educating students in these computational thinking skills, as they are now considered the basic skills necessary for economic opportunity and social mobility. These skills are not only valuable in the classroom, but also in informal learning environments, corporate teams, and the boardroom.

But as Bertram Bruce notes, there is a "commonly-held view that computer technology is a tool that will in and of itself improve education, and ultimately ameliorate social ills."[2] This itself builds from an underlying assumption that technology and social relations are independent from each other, that technologies are neutral tools with fixed meanings, and "that people are agents independent of their technologies."[3]

To this end, it is important we further consider the deeply intertwined advancements of society, hardware, and software; that we continue to expand our methodological landscape to incorporate the deductive and inductive methods along with critical methodologies; that we search out knowledges and wealths beyond those in the dominant narratives of society; and that we work to execute computational tinkering in ways that advance inclusive computer science experiences.

Side by side, we'll also tinker with electronics and code so as to create digital storytelling devices so that by the end of this unit, you can use them to share your own counterstory prototypes as developed in the Orange Unit.

As we consider our future professions, it may be helpful to reflect on the following questions adapted from the Exploratorium's Tinkering Studio.

- How can we support people in bringing their different contexts and perspectives into computational tinkering activities in our future professional endeavors?
- What could be the role of equity and narrative in computational tinkering activities?
- How do we create activities that don't result in people just copying the examples at hand but having opportunities to get creative in drop-in learning spaces?

1. International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA). "Operational Definition of Computational Thinking for K-12 Education," 2011. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf. See the ISTE Standards website for more information as this continues to evolve.
2. Bertram C. Bruce, "Technology as Social Practice," *Educational Foundations* 10, no. 4 (1996): 1. http://hdl.handle.net/2142/13369.
3. Bruce, "Technology as Social Practice," 1.

- How can we support learners and teachers alike to be comfortable not knowing the end result or the answer with computational tinkering activities?

As noted by Linda Liukas, computer code is becoming the next universal language. It is urgent for us to find ways to help young people move forward with greater optimism and bravery regarding technologies. To achieve this, we need to join together and imagine a world in which technology tools can be tinkered with in ways that make them more wonderful, whimsical, and a tiny bit weird. Above all, the Blue Unit seeks to provide a studio for communities of practice to begin just this sort of journey. [Watch the video of Linda Liukas' TED Talk, "A Delightful Way to Teach Kids About Computers," online.]

Blue Unit Overview

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with the option to export your responses, use the online version.]

Key Learning Outcomes Self-Check

The key learning outcomes objectives for the **Blue Unit** include the following. As you view these, take a moment to ***rate 1-4*** these items based on where you think you are in your skills development on each of these items, with additional notes as helpful. Keep these at your side as you work through the Unit to facilitate guided inquiry.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough
4: Yet

Technical Skills

You can (rate 1-4):

- Understand computer science programming concepts, including: Conditional and Boolean logic, iteration loops, variables, and modularization.
- Tinker with the programming languages MakeCode and Python.
- Use programming code to interact with a circuit to achieve an application goal.

Information Skills

You can (rate 1-4):

- Explore a range of information sources, guides, and worksheets in support of current research and practice.
- Apply reference interview skills to design context so as to better understand the strengths, opportunities, and aspirations that are motivating project initiation by an individual or social group.

Cognitive Skills

You can (rate 1-4):

- Generalize past and current problem solving to new contexts.
- Consider the ethical implications of creating or adopting specific technologies and implementations within different social, economic, and cultural contexts.
- Logically analyze the steps for effective adoption and application by different individuals of a specific technology implementation to support individual and social group aspirations and goals.
- Turn personal and community strengths, opportunities, and aspirations into project ideas.

Socio-Emotional Skills

You can (rate 1-4):

- Communicate and collaborate effectively with others.
- Nurture personal confidence, persistence, and tolerance to tackle complex, ambiguous, open-ended problems.
- Empathize with individuals and social groups across a wide spectrum of social and cultural contexts.

Progressive Community Engagement Skills

You can (rate 1-4):

- Use an informal, distributed process to share the responsibility of leadership among all team members of a community of practice in ways that build off each member's strengths and unique insights.
- Understand and respect broad diversities of functioning with regard to analog and digital technologies, and to see this functional diversity not as a deficit, but rather as an inherent and valued aspect of being human.

- Interact as a community of practice in ways that advance each member's capability set and positive freedoms, so as to more effectively achieve and enjoy their valued beings and doings.
- Focus on and learn from the array of often unrecognized and unacknowledged cultural knowledges, skills, abilities, and contacts possessed by a community of practice, and especially those of socially marginalized members.

Critical Sociotechnical Skills

You can (rate 1-4):

- Analyze how economic, social, cultural, historical, and political contexts, and personal preferences and biases, are embedded into a specific implementation of technology throughout the life cycle of the technology.
- Evaluate positive and negative impacts that different technologies and specific implementations of a technology have on individuals and social groups from different economic, social, and cultural contexts.
- Consider the social and individual ramifications of choosing open versus closed software and hardware products within different contexts and for different social, economic, and cultural contexts.

Your Personal Learning Outcome Objectives

Add your own personal learning outcome objectives

Rate Your Personal Learning Outcome Objectives

Take a moment to create a ***1-3 rating*** of your personal learning outcome objectives. Add additional notes as helpful to keep further specific reflections on these specific skills development.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough

Goals	Rating
Your first goal...	

1A: The Logic of Hardware and Programming

Background Knowledge Probe

Bring to mind an everyday computing tool at your regular disposal. On initial reflection:

- What design motivations may have inspired the original creators of this tool?
    - What **may have been happening** that they perhaps wanted to **happen more** through creation of this new/improved/remixed tool?
    - What **may have not been happening** that they perhaps wanted to **happen for the first time** through the creation of this new/improved/remixed tool?
    - What **may have been happening** that they perhaps wanted to **reduce or stop from happening** through creation of this new/improved/remixed tool?
- Who may have been involved in the design, development, and creation of this tool?
- What may have been some of the motivations for those involved?

An Evolution of Computing Tools

From abacus to tabulating machine, early computing tools were designed to achieve inspirational goals. Who was inspired to guide the design and creation of the tools, their strategies for failing forward in that design and creation process, the motivating factors for their involvement, and the ultimate implementation of these early computing tools varied considerably. Further, the impact of these early tools on more recent computing devices, including the digital computers of today, are sometimes embedded within the design inspirations and creation process of past computing devices, even if they aren't specifically interrelated.

Keep these knowledge probes in mind as you take 11 minutes to watch the video on "Early Computing" from Carrie Anne Philbin's Crash Course Computer Science series hosted by PBS Digital Studios:

Knowledge Probe #2

As we gather information on the journeys of hardware and programming logic, let's build up a picture of real situations and real people as they relate to an everyday computing tool. On further reflection:

- What design motivations may have inspired the original creators of this tool?
  - What **may have been happening** that they perhaps wanted to **happen more** through creation of this new/improved/remixed tool?
  - What **may have not been happening** that they perhaps wanted to **happen for the first time** through the creation of this new/improved/remixed tool?
  - What **may have been happening** that they perhaps wanted to **reduce or stop from happening** through creation of this new/improved/remixed tool?
- Who may have been involved in the design, development, and creation of this tool?
- What may have been some of the motivations for those involved?

Many factors have influenced the growth of human civilization. The ways in which this growth was shaped by the design and creation of computing machines, and the ways in which this growth inspired and shaped the design and creation of these machines varies considerably. What is certain is that the demand for more sophisticated and powerful devices was an outcome of the growth of human civilization, whether it was a designed and planned objective or not.

Next, take 11 minutes to watch the video on "Electronic Computing" from Carrie Anne Philbin's Crash Course Computer Science series hosted by PBS Digital Studios:

Knowledge Probe #3

As we gather more specific information regarding the hardware and programming logic of electronic computing, let's work to further build up a picture around real situations and real people as they relate to an everyday computing tool at your regular disposal. On further reflection:

- What design motivations may have inspired the original creators of this tool?
  - What **may have been happening** that they perhaps wanted to **happen more** through creation of this new/improved/remixed tool?
  - What **may have not been happening** that they perhaps wanted to **happen for the first time** through the creation of this new/improved/remixed tool?
  - What **may have been happening** that they perhaps wanted to **reduce or stop from happening** through creation of this new/improved/remixed tool?
- Who may have been involved in the design, development, and creation of this tool?
- What may have been some of the motivations for those involved?

Reading the World, Reading the Word, and Tinkering

As we look through the journeys of hardware and programming logic, and as we reflect on our experiences through reflection probes, we begin a process of codification: bringing together the context and the content of real situations and real people. This work of systematizing, or reducing to a code, a broader range of ideas, practices, policies, and standards into a more ordered plan or scheme, has been an ongoing work of sciencific and professional practices throughout human history. Writing computer programming code is not a radical change to this historic work of codification, but a new technique for doing so using emerging digital technologies in addition to, or instead of, regular analog tools.

Learning these codifications, and then problematizing, or decodifying, these to ask questions and critique more deeply the issues at hand as related to the codifications of these real situations and real problems is at the root of Paulo Freire's critical pedagogy. It is an educational process that brings action and reflection into praxis through dialogue. While there may be individuals in a specific learning community who serve in lead instructional roles within that learning community, their knowledge is less than complete, thus requiring them to periodically shift to the role of student. And students, while they may have no or limited knowledge and expertise with the specific content and context under consideration, bring other knowledge and expertise to bear, requiring periodic shifting of roles to that of instructor.

This approach stands in opposition to the dominant banking concept of education, in which knowledge is bestowed as a gift by experts to the empty vessels who have come to be filled by this learned teacher. For in critical pedagogy, the dialogue of all within a learning community presupposes equality amongst participants. Each must trust the others as people with expertise in their own rights. An instructor-student comes into the learning community with a significant level of knowledge or skill in an area of study, and so has certain clear responsibilities with regards to learning outcomes achieved using expository techniques such as synchronous and asynchronous audio and video lectures and authoritative choices on readings and experiential learning activities.[1]
At the same time, the instructor-student must also have mutual respect and love for the student-instructor as each one questions what they know. And together, instructor-student and student-instructor must realize that through the dialogic action-reflection cycles of problem-posing, education leads to potentially reconsidering knowledge in ways that further humanize all participants.

A central tenet of Freirean literacy campaigns, which strongly fit within the digital literacies of today, is the essential recognition that in reading the words, in this case of technology, we are also learning to read the world. The instructor-student strives to incorporate strategically selected words, words that generate exploration and can be combined to create generative themes, into co-explorations that iden-

1. Drew W. Chambers, "Is Freire Incoherent? Reconciling Directiveness and Dialogue in Freirean Pedagogy," *Journal of Philosophy of Education* 53, no. 1 (February 2019): 26-28, https://doi.org/10.1111/1467-9752.12340.

tifiy limit-situations built through domination. And for Freire, domination implies its opposite, limit-acts leading to the objective of liberation.[2]

As we enter into this exploration of person-centered computational tinkering, it is essential that we join with a diverse group of others. This is the liberatory literacy work of Jane Addams and the Hull House, of Miles Horton and the Highlander Folk School, and it is the work of community inquiry and community networking today.[3] And it is the work that inspires the construction of the generative themes within this book. This requires that we work across difference, bringing together a breadth of functional diversities, cultural wealths, and knowledges into dialogue facilitating our critical exploration of the sociotechnical artifacts in our daily lives. As we do this, we must work towards decodifying the real situations and real people that are shaping, and being shaped by, these sociotechnical artifacts. Through this, we can begin to identify with aspects of the situation, feeling the ways we are in the situation, reflecting upon and discussing these various aspects, and thereby bringing the picture into better focus through our new recodifications of the concepts and terms at hand.

As Bertram Bruce notes in "Technology as Social Practice," technology and social practices are not separate. Social relations are encoded in technologies. Conversely, technologies are encoded in social relations. Social relations and technologies are mutually constituted. But as we become enmeshed in *these* social + technical networks, we are left seeing only the black box. And so we cannot work to codify the social separate from working to codify the technical. Just as the invention of writing could be considered the first educational technology, so too the moveable-type printing press further altered educational practice. But these were mutually shaped, not just mutually shaping. As Bruce further notes: "A given technology brought about educational change, but at the same time, requirements of society operating through the educational system generated changes in the associated technology." Technologies and social practices cannot be separated. Bruce concludes: "A technology is a system of people, texts, artifacts, activities, ideology, and cultural meanings. It doesn't so much determine, as become social practices. Our task then must be consider critically what those social practices are now and what they can become in the future."

Lesson Plan

Let's begin by considering how all forms of the creative process, including computational tinkering, are social and technical works. Analyzing this can only be done together with others from different backgrounds, experiences, knowledges, wealths, and valued ways of being and doing. We'll start by working to advance our reflections, more fully bringing together the social and the technical into a new picture of real situations and real people. And we'll work to use community of practice group dis-

2. Paulo Freire, *Pedagogy of the Oppressed, 50th Anniversary Edition* (New York: Bloomsbury Academic, 2018), 99.
3. Bertram C. Bruce and Ann P. Bishop, "New Literacies and Community Inquiry" in *The Handbook of Research in New Literacies*, edited by J. Coiro, C. Knobel, C. Lankshear, and D. Leu (New York: Routledge, 2008), 699–742, http://hdl.handle.net/2142/15133.

cussions and individual reflections, along with the hands-on activities, to bring this into sharper focus in the context of coding concepts and practices.

Our objective is to bring together a complex blend of ideas, concepts, and hands-on practice with components of our everyday sociotechnical artifacts. Along the way, look for innovation-in-use/remix prompts encouraging you to make creative adjustments and expansions of the sociotechnical artifacts to do more of what you and your community of practice value being and doing. What hopes, doubts, values and challenges might be brought into dialectical interaction with their opposites? How might this exchange inform our use of existing analog information systems and our design, creation, and use of complementary digital tools?

Essential Resources:

- Bruce, Bertram C., and Ann P. Bishop. "New Literacies and Community Inquiry." In *The Handbook of Research in New Literacies*, edited by J. Coiro, C. Knobel, C. Lankshear, and D. Leu, 699–742. New York: Routledge, 2008. http://hdl.handle.net/2142/15133.
- Bruce, Bertram C. "Technology as Social Practice." *Educational Foundations* 10, no. 4 (1996): 51–58. http://hdl.handle.net/2142/13369.

Additional Resources:

- Freire, Paulo. "First Letter: Reading the World/Reading the Word." In *Teachers as Cultural Workers: Letters to Those Who Dare Teach*, 31–47. Boulder, CO: Westview Press, 2005.

Key Technical Terms

- The distinctions between **coding, programming**, and popular **programming languages**.
- Programming **functions**, **statements**, **variables**, **arrays**, **Forever**, **For**, and **While** loops, **conditionals**, and **Combinational** and **Sequential** logic.

Professional Journal Reflections:

In this week's viewing of videos, reading of text and context, in small and large group discussion, and your hands-on actions with **sociotechnical** artifacts, consider:

- What ideas have you built or codified regarding hardware and programming logic, its influences, and its influencers?
- What ideas are coming into clearer focus through community reflection and action, regarding the sociotechnical artifacts explored in this book? With other sociotechnical artifacts in your life?

1B: Essential Coding Concepts

Background Knowledge Probe

Take a few minutes to bring to mind an electronic device from home or work that you use daily. Then think through the following questions as they relate to this device:

- What do you see as the primary purposes for having this electronic device?
- What inputs and outputs do you think are used to ensure this electronic device performs as expected?
- What calculations do you think this device makes with the inputs and outputs to ensure it performs as expected?

The Shaping of Electronics

In the Orange Unit, we worked to demystify the basic electronics and hardware found in the digital technologies of our daily lives. To do so, we primarily worked with the physical electrical components and supporting parts to build a complete, closed path for current to flow from source to ground. That is, we created **circuits**.

Consider that someone chose to design and provide instructions for others to create a circuit in which a momentary switch push button is placed on a breadboard next to a Raspberry Pi computer in a way that one diffused white LED stays lit as long as that circuit passes current through a resistor to the passthrough legs of that push button on to the anode (positively charged) leg of the LED, then through the cathode (ground) leg of the LED and to a grounded connection. Two other LEDs are placed in a circuit such that they only light up when the momentary switch push button is depressed.

- Why use diffused LEDs and not ultra bright LEDs?
- Why white and blue LEDs and not different colors of LEDs?

- Why is only one LED lit unless a switch is pressed while two other LEDs are only lit when the switch is pressed?
- Why connect the wires and electronics in the way selected on the breadboard and then through a 40-pin **ribbon cable** to the Raspberry Pi, and not in one of the many other ways they could have been connected?
- Were these all good enough, less than you would have preferred, exceptionally well chosen, or is this question even of importance?

Asking these kinds of questions expands our understanding of the ways in which the shaping of electronic hardware shapes what data is collected, how it is collected, and ultimately its influence on knowledge/power.

How are you being shaped by the design choices in this book? Who is doing the shaping? Why?

In the last unit, our hope was to join together, author and reader, as a learning community to shape the actions of the electronic components through the building of circuits only using those electronic components, thereby establishing complete and closed paths in which an electric circuit can flow. We learned how current flows from a source, through electrical components like resistors, LEDs, and momentary switch push buttons, to reach ground.

We saw how **integrated circuits** use **printed circuit boards** to bring all these pieces together into a single whole. In this way, all that is apparent is the source entering into the + side of the integrated circuit and then leaving through the − side to reach ground. In some ways, then, the integrated circuit begins to place these components into a "black box." But the individual electronics are still there, all the same.

We also saw it is possible to control the reading and execution of circuits using programming code rather than through rewiring of physical electronics. Many of our everyday electronic tools, and not just our computers and smartphones, are built using both physical electronics and programming code.

Indeed, integrated circuits, printed circuit boards, and other components of microcontrollers, microcomputers, computers, smartphones, and our many other digital technologies, are simply these electrical components integrated into tight containers that are sometimes difficult to see from the outside.

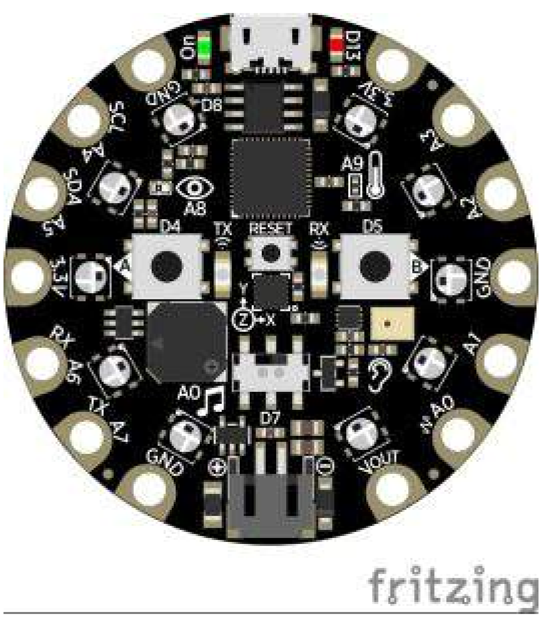
Essential Coding Concepts

There was a long journey for people to move from the mechanical computing done with an abacus to the electromechanical computing done for the 1900 census to today's general purpose programmable computers that use transistors to perform calculations.

Many argue that the first person to create a set of programmed instructions to be used with a mechanical calculating engine so that it could achieve its full potential was Augusta Ada King-Noel, Countess of Lovelace, a mathematician and writer. Ada Lovelace, as she is generally known, worked closely in the early 1800s with Charles Babbage, who focused on the mechanical general-purpose computer, the Analytical Engine. Lovelace's papers and presentations were deeply influential within the British research circles of the day in seeing the full potential of computing machines, today known as computers.

We'll be working with the Raspberry Pi microcomputer, Circuit Playground Express microcontroller, and electronics purchased from Adafruit, a woman-owned business named in honor of Lady Ada. Our work with coding is done, also, in honor of this original creator of programmed instructions.

Let's explore how we can create our own programming code to read inputs, control outputs, and perform calculations. To do this, rather than working with a general purpose programmable computer like the Raspberry Pi microcomputer, let's use the limited purpose programmable Circuit Playground Express microcontroller.



Microcontrollers are designed to meet specific input/output **specifications**. As the purpose of the Circuit Playground Express microcontroller isn't high performance, per se, but rather education, it is a bit more flexible than other microcontrollers. To meet various educational specifications, designers, engineers, and others came together to move from prototyping, likely at times using basic electronic components, wires, and breadboards, to ultimately create a diversity of integrated circuits put together on one printed circuit board to result in the Circuit Playground Express microcontroller. In shaping this tinker microcontroller, the intent was to provide us with flexibility to use the device in ways that can serve a range of open-ended learning and design specifications. As we'll learn throughout this unit, it is a participatory design process that intentionally underdesigns the sociotechnical artifact to assure greater innovation-in-use by others.

Similar to a diagram showing the LED Sequin Integrated Circuit, this diagram of the Circuit Playground Express doesn't provide specific descriptions of the electronic components actually built into the printed circuit board, nor how they are brought together with conductive materials. But all the same, they are miniature versions of the resistors, LEDs, and push buttons, as well as sensors, switches, speakers and other electronics, just in miniaturized form. They are soldered to a printed circuit board and connected together using sturdy conductive materials.

So before moving on, take a few minutes to learn about the Circuit Playground Express. The best way is to visit the the Adafruit Learning Center:

```
https://learn.adafruit.com/adafruit-circuit-playground-express
```

Especially check out the "Overview," "Guided Tour," and "Pinout" sections to see the otherwise hidden details of the electronic components put together on this printed circuit board.

Guiding Questions

As you read, keep in mind these social guiding questions:

- Who shaped the designers and builders of the Circuit Playground Express to create it in this way?
- Was that shaping intentional, accidental, or a little of both?
- How does Adafruit's goal of teaching electronics and learning to code fit into all of this?
- As a result of the design and implementation of the Circuit Playground Express, how are you being shaped?
- And how might that shaping ultimately result in the shaping of still other people?

Coding the Circuit Playground Express

We're pretty familiar with our everyday general purpose computers that play an important role in running a wide range of applications continuously, or at selected times, to accomplish a diverse range of tasks. To access this book via a web browser, to communicate via email, to talk and text using a phone, to participate in social media, and so many other activities, you have become rather familiar with laptop and/or desktop computers, and smartphones, which are microcomputers. (Microcomputers are simply small computers that contain a microprocessor as its central processor.)

The instructions created to control or evaluate electronic components are commonly termed **code**. As coded instructions are brought together to control the complex operations of computers and other machines, they are typically termed **programs**. These days, doing this work is referred to either as **coding or programming**. The specific words, characters, symbols and structuring of code to achieve the desired works are typically termed a **programming language**.

As a general rule, microcontrollers are designed to continuously run a single application using a specific programming language.

The Circuit Playground Express has been designed especially for educational use and so works with a range of programming languages, including Microsoft's education-friendly MakeCode block coding language, Adafruit's CircuitPython extension to the Python programming language, and Arduino's Arduino integrated development environment (IDE), written using the Java programming language.[1]

For the next few sessions, we'll work with MakeCode using Adafruit's website:

```
https://makecode.adafruit.com
```

It was designed to work specifically with their Circuit Playground Express. It has many similarities to Scratch, a programming language developed by the Lifelong Kindergarten Group at the MIT Media Lab.[2]

- Both use visualized blocks of code which can be combined to create error-free code which achieves a certain event.
- Both are freely accessible to anyone, and creators are encouraged to work collaboratively in online communities to share creations.
- Both are being used by a wide range of explorers, from those in the early grade levels to older adults, and all in between.

A key difference, though, is that Scratch focuses on creation of interactive stories, games, and animations visualized using sprites wearing costumes and performing on stage. Adafruit's MakeCode is created online in a Circuit Playground Express prototyping space and tested in a visual simulator before code is then downloaded and moved to your physical Circuit Playground Express for live testing.

Exercise: Introducing MakeCode and Circuit Playground Express

Adafruit has worked extensively to come up with a range of introductory tutorials, courses, and web pages, along with a range of exciting projects for you to use the Circuit Playground Express in your everyday lives. Links to these can be found on the MakeCode homepage.

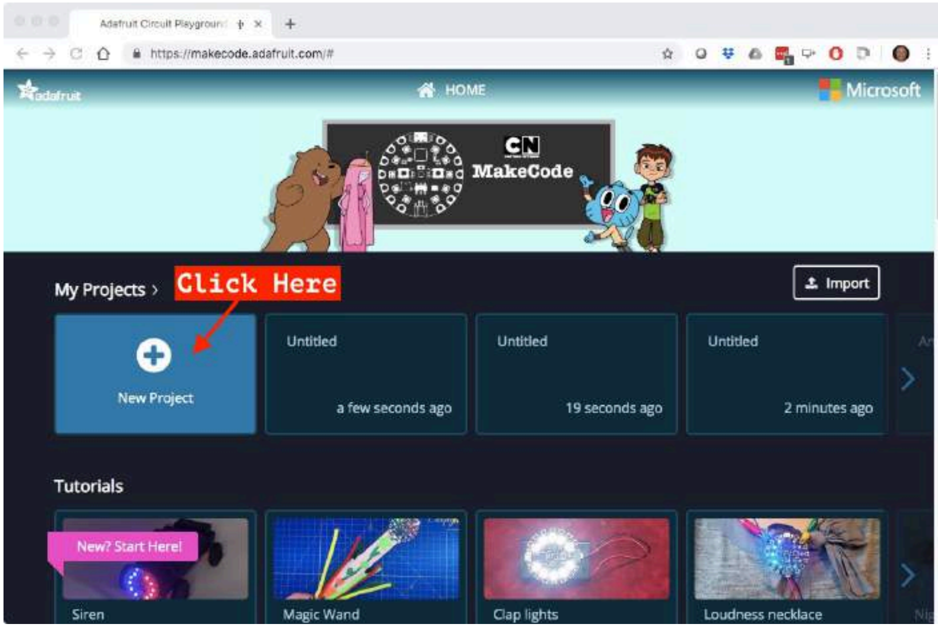
This next exercise is a remix of different projects and guides found through Adafruit and other sources. We'll work to familiarize ourselves with coding concepts while also repeating certain LED and switch activities from the Orange Unit.

But first, let's test the 10 general-use LEDs built into the Circuit Playground Express ring. Take four minutes to view the video showing the steps to write the MakeCode Animated Blinky FOREVER function, to save this to your laptop Downloads folder, and to flash this to the CPLAYBOOT device after connecting the Circuit Playground Express to the USB port on your laptop.

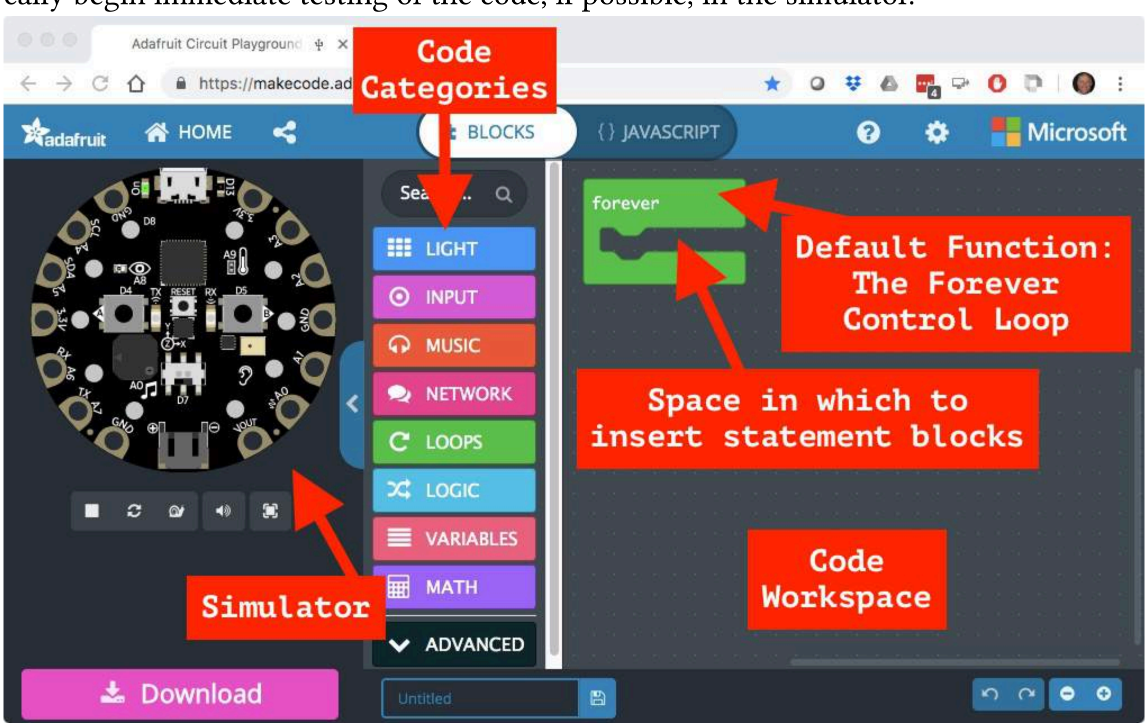
1. MakeCode © 2020 Microsoft and Adafruit. Screenshots used with permission.
2. Those who are familiar with Python or the Arduino IDE are welcome to create remixes of the following exercises in those languages, if desired.

Steps

1. In your laptop's web browser, click on "New Project" from [the MakeCode homepage](#).

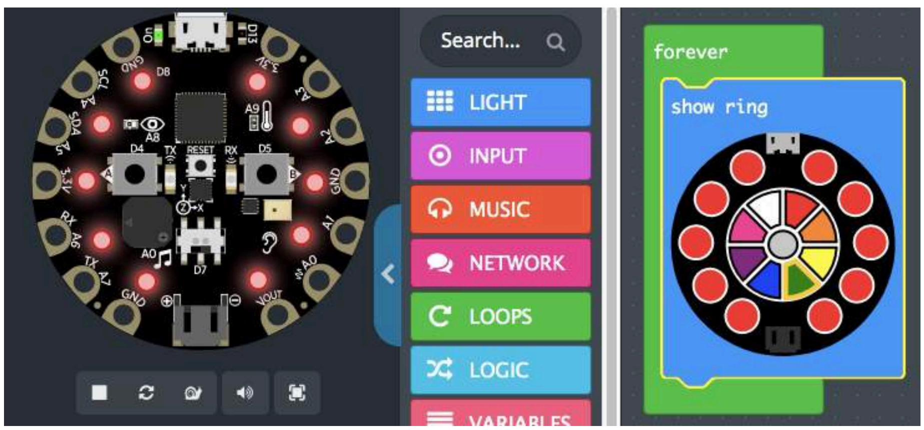


When programming in MakeCode, we select blocks from within one of the "Category" menus and drag those into the workspace. These are either dropped into an open workspace to create a new **function**, like the default FOREVER **control flow statement** loop, or within an existing function as a **statement**. All blocks in a category are color coded to match the color of that category as seen in the parent window. So we can tell that the **green** FOREVER block came from the **green** LOOPS category. MakeCode will automatically begin immediate testing of the code, if possible, in the simulator.

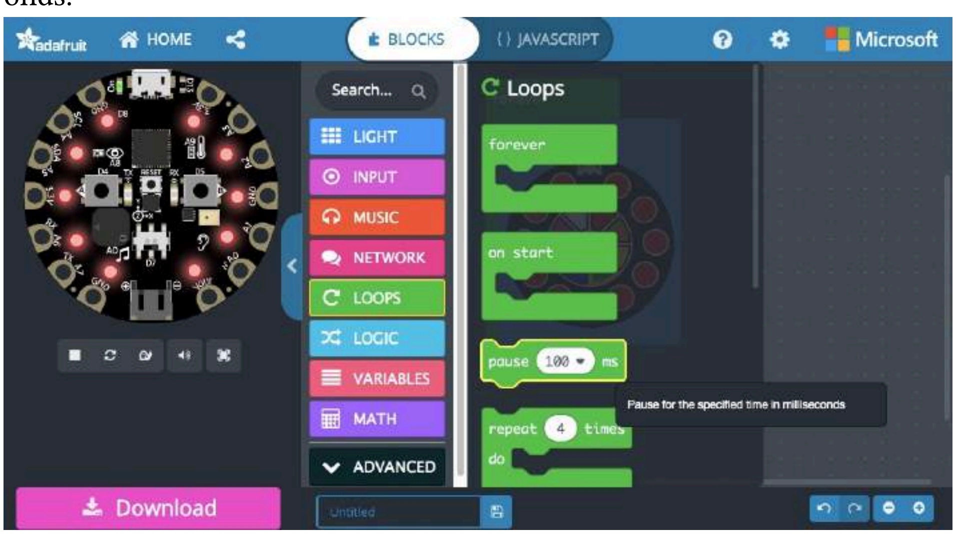


2. In the LIGHT category, drag the "show ring" block and drop it into the **FOREVER loop**.

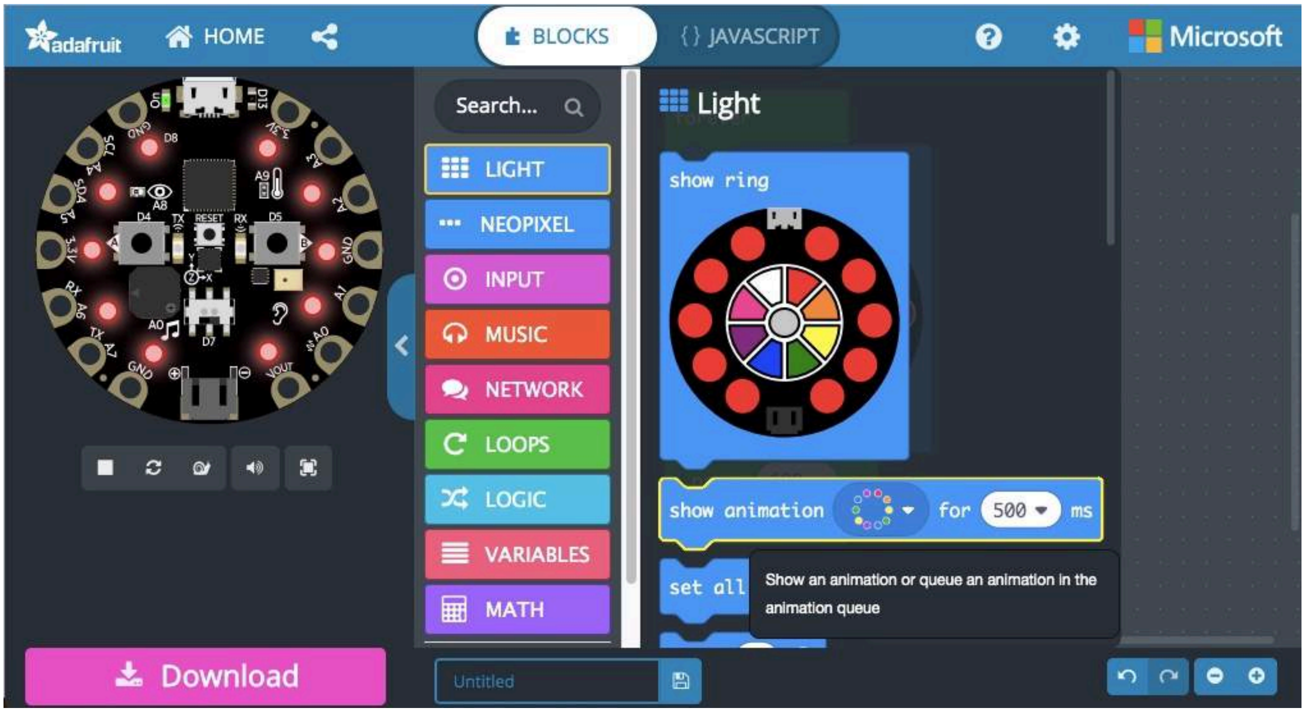
By default, all 10 LEDs are set to red, and after a few seconds the LEDs in the simulator should also turn red.



3. From the LOOPS category, drag the "pause" block and drop it below the "show ring" block, providing a statement to pause the loop for a set amount of time – by default 100 milliseconds.

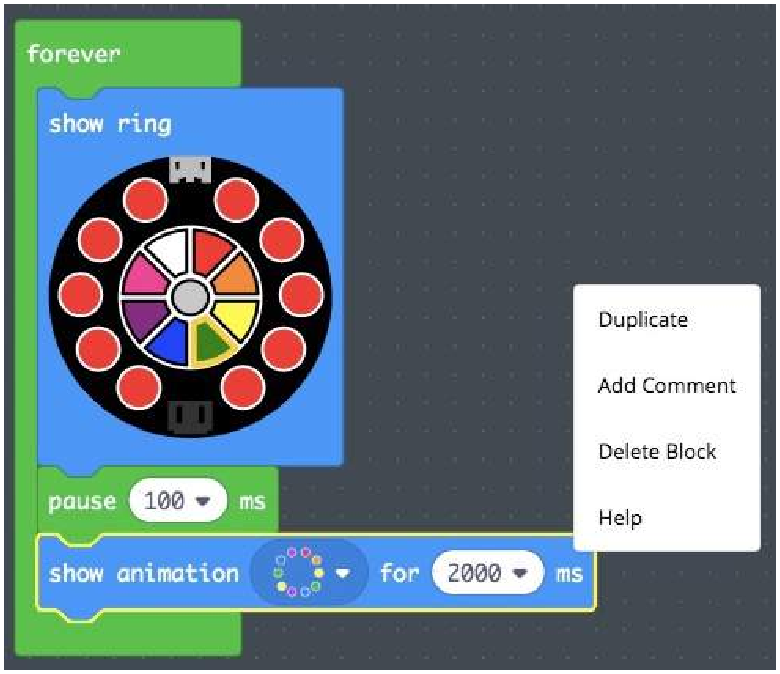


4. From the LIGHT category, drag the “show animation” block and drop it below the “pause” block.

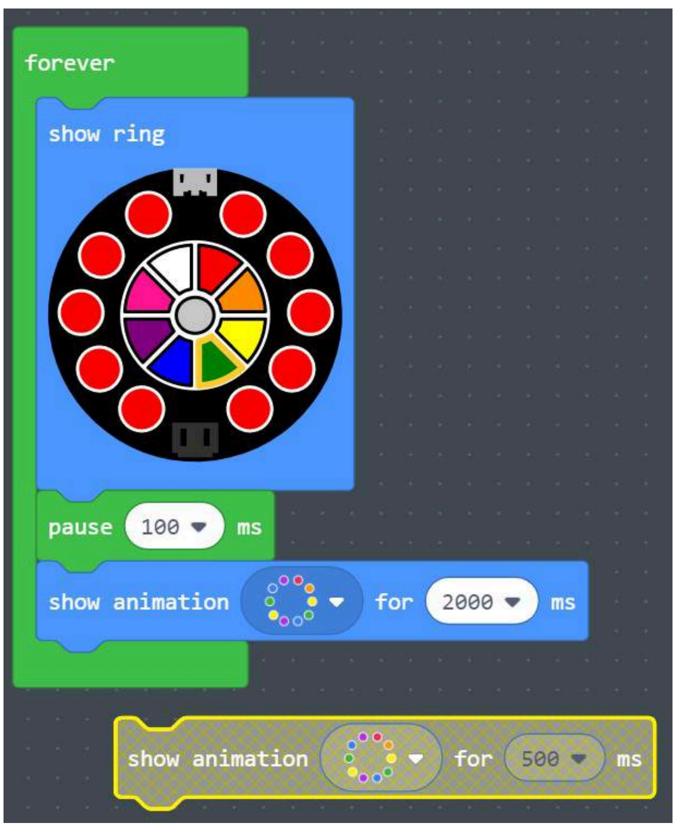


5. Increase the “show animation” time to 2000 milliseconds, that is, to 2 seconds, by clicking on the default “500 ∇” setting and choosing 2 seconds.

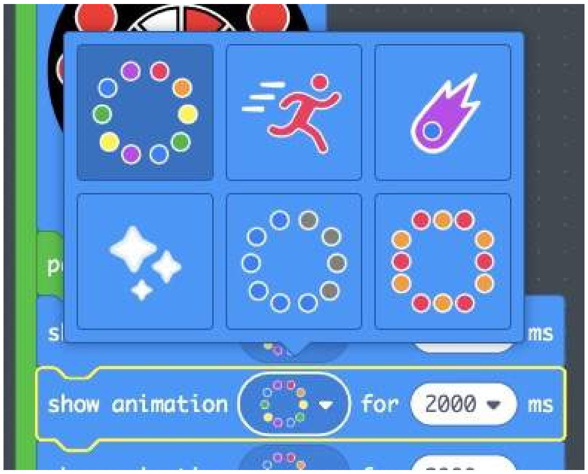
6. Right click on the existing "show animation" block and select "Duplicate" from the provided window that pops up.



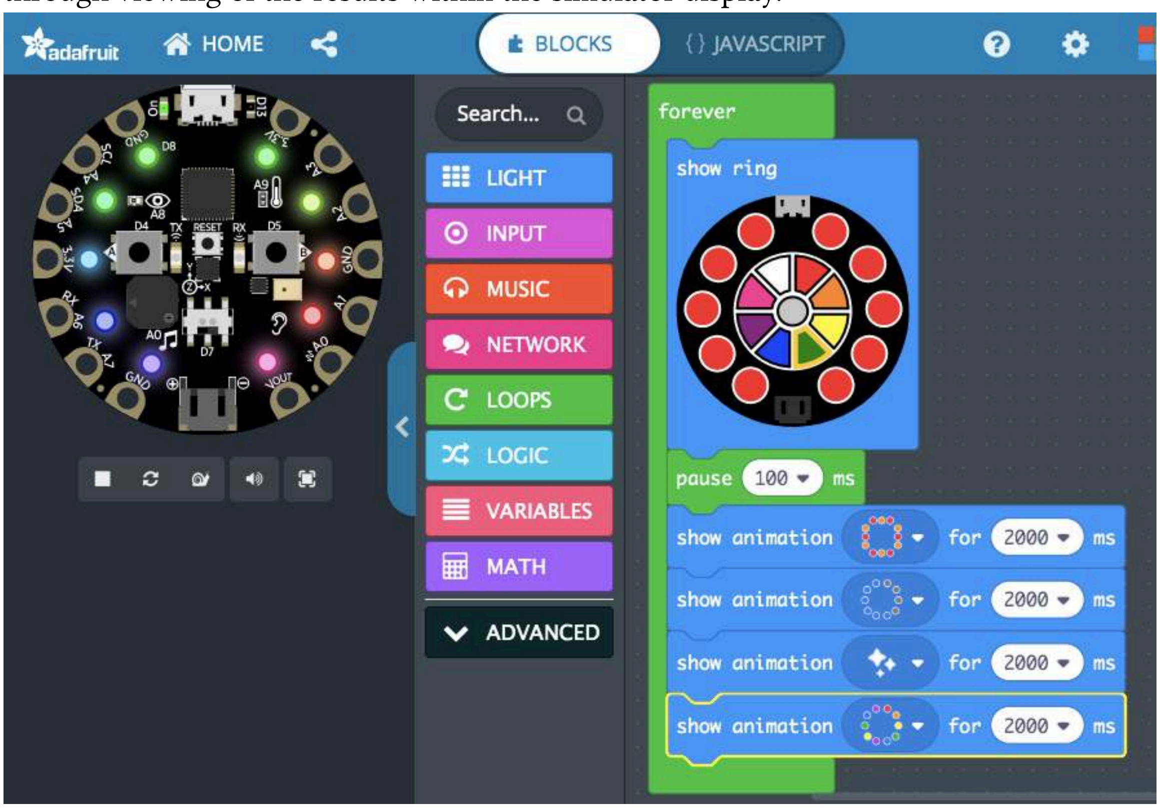
7. Drag the second "show animation" block from within the workspace and drop it below the first "show animation" block.



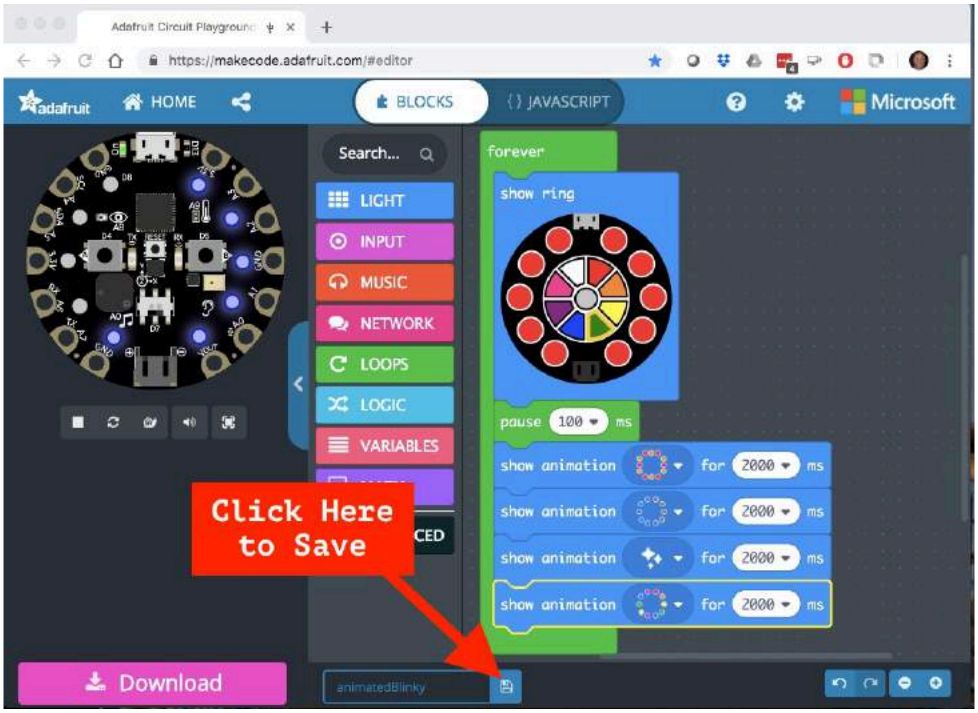
8. Right click again on the first "show animation" block and select "Duplicate" from the provided window that pops up. This time it will duplicate its own block, and all blocks below it. In this case, that's only the second "show animation" block. Place these below the first two.
9. As of now, the simulator should "show ring," pause for 100 milliseconds, then "show animation" default for 4 x 2000 ms, or 8 seconds. Click on one each of the "show animation" block animation ∇ visuals and select from the six options provided.

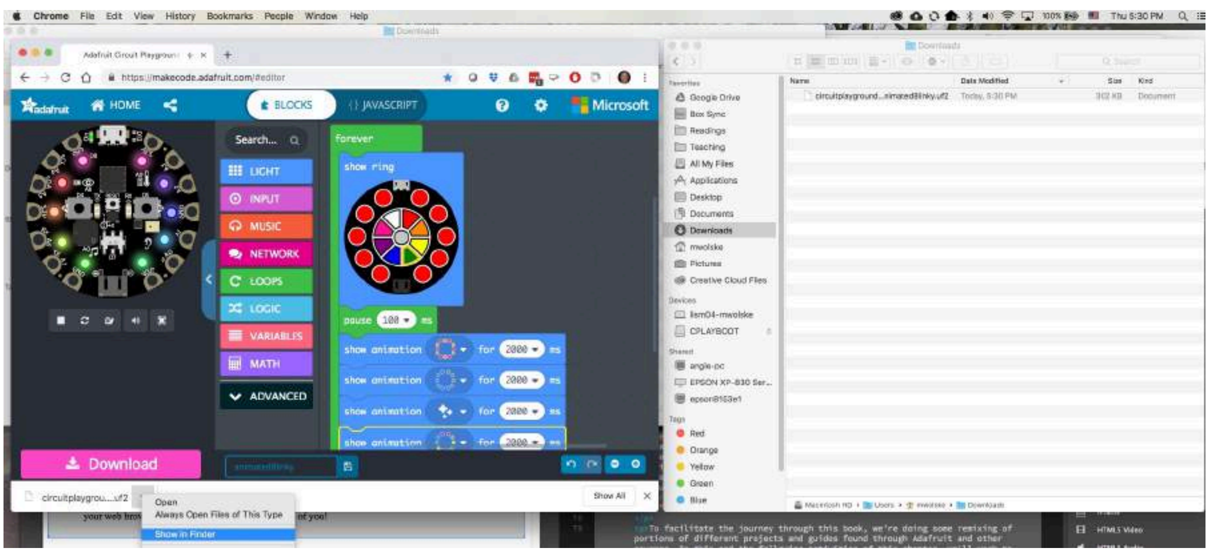


10. This is rapid **prototyping**, so only take a few moments to create an acceptable testing prototype through modifications of the six lines of code now within the FOREVER loop, and through viewing of the results within the simulator display.



11. At the bottom of the MakeCode window, replace the word "Untitled" with "animated-Blinky" and click the save icon to the right of the filename.





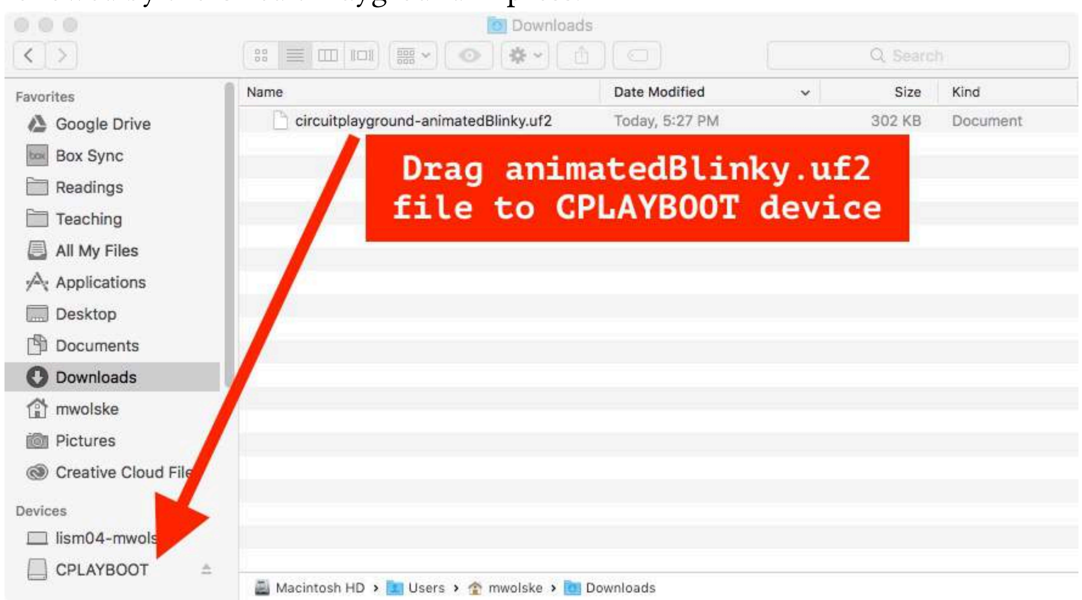
12. You now need to flash the code to your Circuit Playground Express. You can [review the video "Connecting the Circuit Playground Express to Your Laptop," which is a section of the previous video to see the specifics on these steps again]().
    a. Connect your personal laptop to the Circuit Playground Express via a USB to micro USB cable.
        - The lights on the Circuit Playground Express should start flashing different colors if this is the first time you've plugged it in.

    b. Press the center "reset" push button on the Circuit Playground Express.

- You may need to push it longer or shorter to move from flashing LEDs to a solid green set of LEDs.
- This indicates it's ready to be physically reset with new code.

c. In your Windows Explorer or Mac Finder window, you should now see the device named "CPLAYBOOT" listed. [Watch the video "Saving and Flashing Code to the Circuit Playground Express" to see a short section of the earlier video demonstrating these last steps.]

d. Drag the file `circuitplayground-blinky.uf2` from your downloads folder and into the CPLAYBOOT device. This saves the MakeCode you created onto the Circuit Playground Express in away that allows it to be understood and followed by the Circuit Playground Express.



Key Takeaways

- A **block code programming language** and Integrated Development Environment (IDE) like **MakeCode** allows you to drag and drop error-free sections of code from the CATEGORY list into a working programming screen.
- **Microcontrollers such as the Circuit Playground Express** are typically directed by one or more small programming **modules** or functions with instructions which are continuously run unless the microcontroller is disconnected from power or otherwise turned off.
  - To achieve this, the MakeCode environment launches a **FOREVER block** by default when you open a new project. This creates a continuous **iteration**, a **control flow statement** used in computer programming to loop back around.
- We dragged and dropped six blocks that served as individual lines of code to be executed in

sequence each time the FOREVER loop was restarted at the top.

- A **live-screen simulator** automatically restarts and runs the updated code each time you drag and drop a block, letting you test your program in a browser.
- The Circuit Playground Express is designed by default to allow you to then **flash** the code you created onto it, at which time the physical microcontroller will restart and run the code, just like the simulator.

Choose Your Own Adventure

A Person-Centered Guide to Demystifying Technology is an introductory textbook using a variety of hands-on exercises to evolve a more holistic and nuanced understanding of the sociotechnical artifacts we use as a daily part of our professional lives. It seeks to develop a critical approach to sociotechnical artifacts and to advance community agency in appropriating technology to achieve our individual and community development goals. As we've probed deeper, we've begun to expand our understanding of innovation beyond the package inside a box, but to also include the set of practices and remixes that emerge from the social setting of its use. We've also explored the hidden and forgotten innovators of technologies, including technologies beyond those defined as innovative within male-dominated professions and monopolies.

Working through this space is messy even at the best of times as we work to unpack the social and technical bits and their underlying, non-neutral shapers. And so the exercises seek to find a means through which a broader Community of Practice collaboration is possible. For the Blue Unit, this has been done through the selection of the education-focused Circuit Playground Express microcontroller board, and related exercises that explore core programming concepts through the use of the block-programming language **MakeCode**.

Alternative Programming Adventures

Whether or not you enter into this book with a background in programming, you may want to **choose an alternative programming adventure** to MakeCode to perform these exercises. Or you may want to do the exercises following the written MakeCode instructions, and then do your own **innovation-in-use remixes** using an alternative programming language. To that end, the Circuit Playground Express is compatible with three major text-based programming languages, including:

- **JavaScript**: a high-level programming language originally focused on livening up web pages through the use of its plain-text scripts. While its naming includes the term Java, a general-purpose programming language, it is independent from it and includes important features that make it a "safe" programming language that can be used with web pages and on web browsers in ways that Java cannot. Of special note, the design choices incorporated into JavaScript make it an ideal complement to the block-based MakeCode programming language.

  - It's easy to switch between MakeCode and JavaScript with a click of the code selection window options at the top, switching between adventure types. In this way JavaScript might prove an ideal text-based complement to the block-based MakeCode throughout the exercises of the Blue Unit to *Do Something New*!

- **Python**: a popular Open Source Software programming language that has proven very helpful within educational contexts as well as within the broader programming community. Its support of **modules** and packages means it can easily be reused in new projects. And as is the case here, it can be expanded to serve special use contexts, such as with microcontroller boards like the Circuit Playground Express. **CircuitPython** is based on Python and provides the essential hardware support needed to work with microcontrollers while including the Python coding base.
  - The Circuit Playground Express is CircuitPython-ready, but does not have CircuitPython installed by default. This is because you can only work with one programming language at a time and MakeCode was chosen as the perfect introductory pathway. However, it's easy to install CircuitPython. When installed, a new CIRCUITPY drive will appear in addition to CPLAYBOOT. NOTE: be sure to copy your **.py** program files to CIRCUITPY as that's the only place from which they will run. The Adafruit Explore & Learn site has more to get you started with CircuitPython on the Circuit Playground Express.
  - To go back to MakeCode, all you need to do is flash your selected MakeCode **.uf2** file to CPLAYBOOT. NOTE: in so doing, you are also uninstalling CircuitPython; that said, you can switch between these regularly and often. Adafruit provides a helpful guide on Moving Circuit Playground Express to MakeCode.
- **Arduino**: a programming language adding a library of hardare interfaces to the C/C++ programming language. The Circuit Playground Express is Arduino-ready and you can find out more on this from Adafruit. While Arduino outperforms MakeCode/JavaScript and CircuitPython when run on microcontroller boards, it can take more time within development uploading/compiling and memory management. As such, the other two are recommended for the exercises of the Blue Unit.

The adventure is yours for the choosing. While it's important to exercise your choice in ways that help you reach a sufficient state of "Yet" with regard to the key learning outcome objectives within the unit, it's likely you have as many and more of your own valued beings and doings that you are working towards. It is our intent to assure you have choices and the means to exercise and achieve these choices in support of those values.

Exercise: Using Momentary Switch Circuits With Conditional Code

The animated blinky exercise gave us a chance to test out the MakeCode web browser. It also introduced us to the downloading and flashing of code to our physical Circuit Playground Express.

From here, you could further tweak the light display, unplug the Circuit Playground Express, connect it to a portable battery source, and voilà! Now you can carry this around after dark as you walk or bike home!! [Watch a video of the finished Circuit Playground Express Animated Blinky display online.]

Now let's try using inputs, like the left and right momentary switches of the Circuit Playground Express, to determine the state of the LEDs. So far the sequence of our FOREVER control flow statement loop is run precisely the same over and over with no intentional changes. Using switches to turn on an off LEDs in our breadboard circuits is an example of **conditionals**.

Let's work to do the same thing using the Circuit Playground Express LEDs and switches.

For this activity, we'll repeat our Pi-Controlled LEDs activity from the Orange Unit. In that exercise, one white LED was constantly on. Two more white LEDs only came on when the left switch was pressed down. The blue LED, on the other hand, was on unless the right switch was pressed down. Let's use MakeCode to do something similar on the Circuit Playground Express. This is an example of taking the insights gained from the many prototypes done with a breadboard and electromechanical components, and using them to extend our prototyping at a higher level of **abstraction** using a microcontroller.

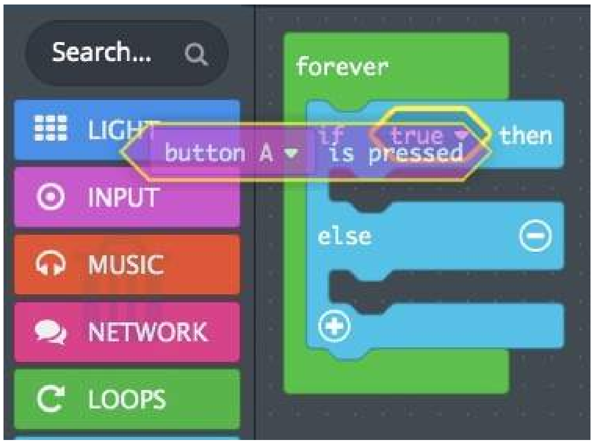
One of the most common conditional statements is the **"If...Then" statement**, also referred to as an "If...Then...Else" statement. In our case, we'll use this statement such that if the button A is pressed then turn a white LED on, else turn/leave it off. On the other hand, if button B is pressed, then turn a blue LED off, else turn/leave it on.

An important first step in coding is to **initialize** items before moving on to the regular work of the code. In MakeCode, this is done with the "ON START" initialization control flow statement block. Any lines of code in this block are done in the order listed, and are done only once when power is first provided to the Circuit Playground Express. Hence the term initialization. We'll use this to set a brightness for the LEDs and turn on and leave on another white LED as we've done in the Orange Unit exercises.

Let’s get started.

Steps

1. On the MakeCode homepage, click to start a new project.
2. From the LOGIC category, drag an “If…Then…Else” block and drop it into the FOREVER loop.
3. From the INPUT category, drag a “button A is pressed” block and drop it within the <true ∇> window, like so:

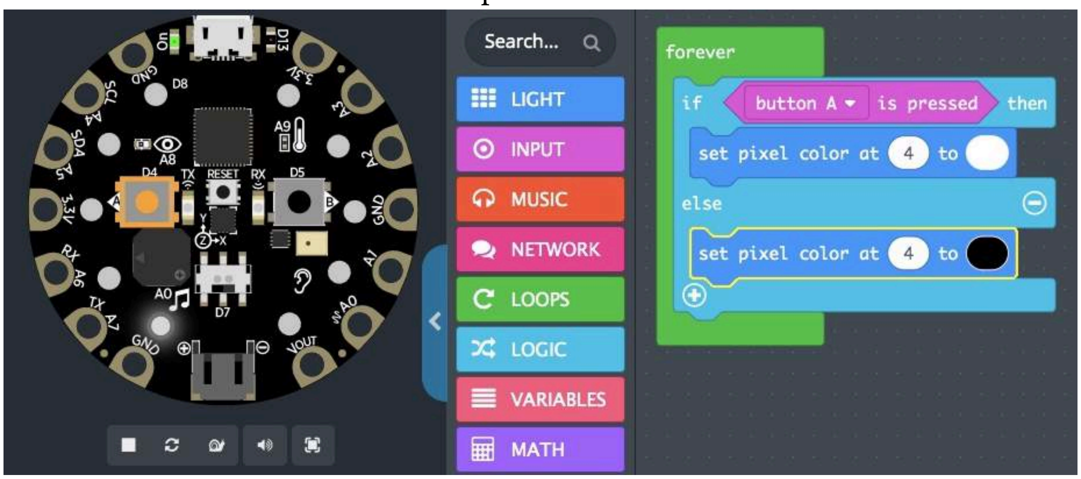


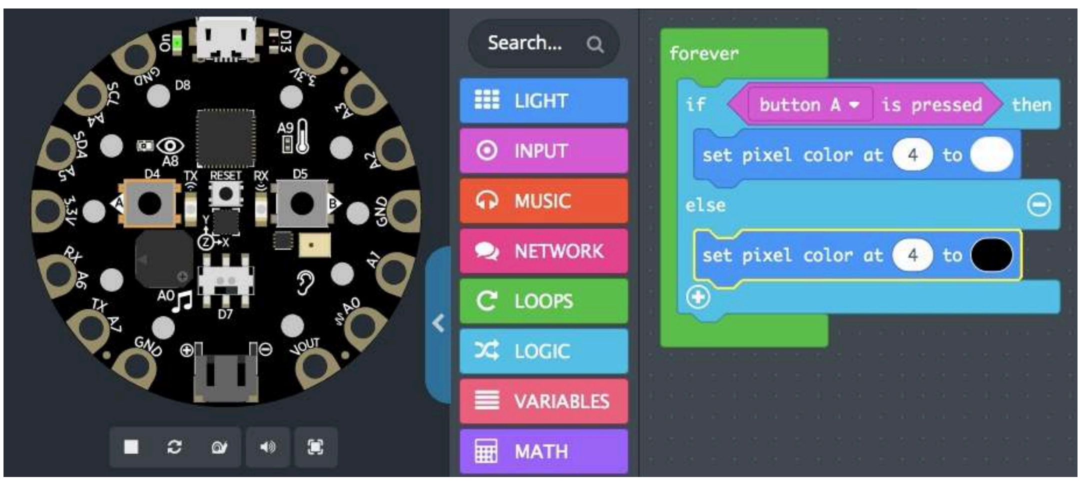
Notice in the image that the <true ∇> and <button A ∇ is pressed> windows both have a yellow border around them. This indicates that, if you release the mouse or touchpad button now, the upper Button A window will be dropped in the the lower True window. It’ll take a little practice to get good at this, and I still find I sometimes drop one window into the wrong lower window. Fortunately, by dragging and dropping, it can be moved to the intended window from wherever it was accidentally placed.

4. Time for a design choice. Which button will work with the normally off white LED, and which the normally on blue LED? For now, let’s go with button A = white LED off unless pressed.
5. From the LIGHT category, within the More section, drag the “set pixel color at” block and drop it into the then section.
6. Change the at subwindow to 4, and change the color from red to white.
7. In the simulator, press the “A” button on the left. Notice the LED below the musical note symbol turns white and stays so.
    - Could it be that the location of the buttons in the simulator looks similar to the location of the buttons on the breadboard sitting next to my computer workstation? Design influences come from many physical as well as social and mental influences!

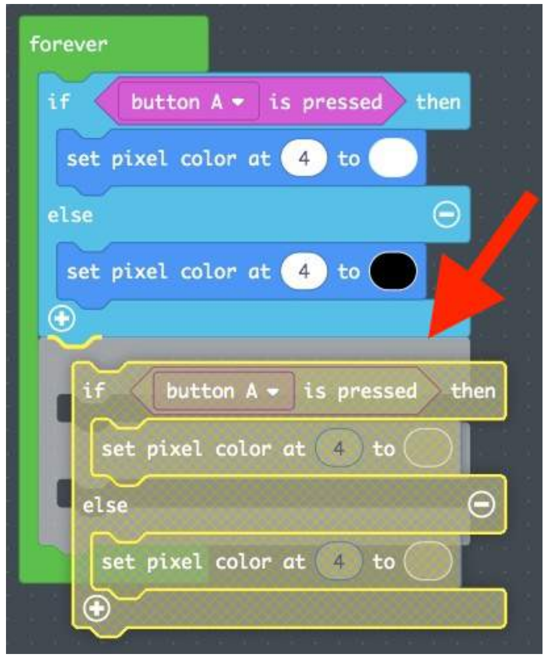
8. The "else" part of an **"If…Then…Else" logic statement** is just the place we need next so that we can turn/leave the white LED off if the "button A is pressed" condition above is NOT true.

    - Take another "set pixel color at" block and drop it into the else section, changing the at to 4, and the color to black.NOTE: You can do this either by duplicating the "set pixel color at" block from the then section and then dragging the duplicate into the else section, or you can enter into the INPUT category and drag a new "set pixel color at" block from there to drop into the else section.Our white LED is now off on the simulator **unless** we press button A!





9. Right click in the area of the "then" statement text, and choose duplicate from the pop-up window.
10. Drag the duplicate "If…Then…Else" logic block and drop it **below** the + symbol at the bottom of the current "If…Then…Else" statement block.



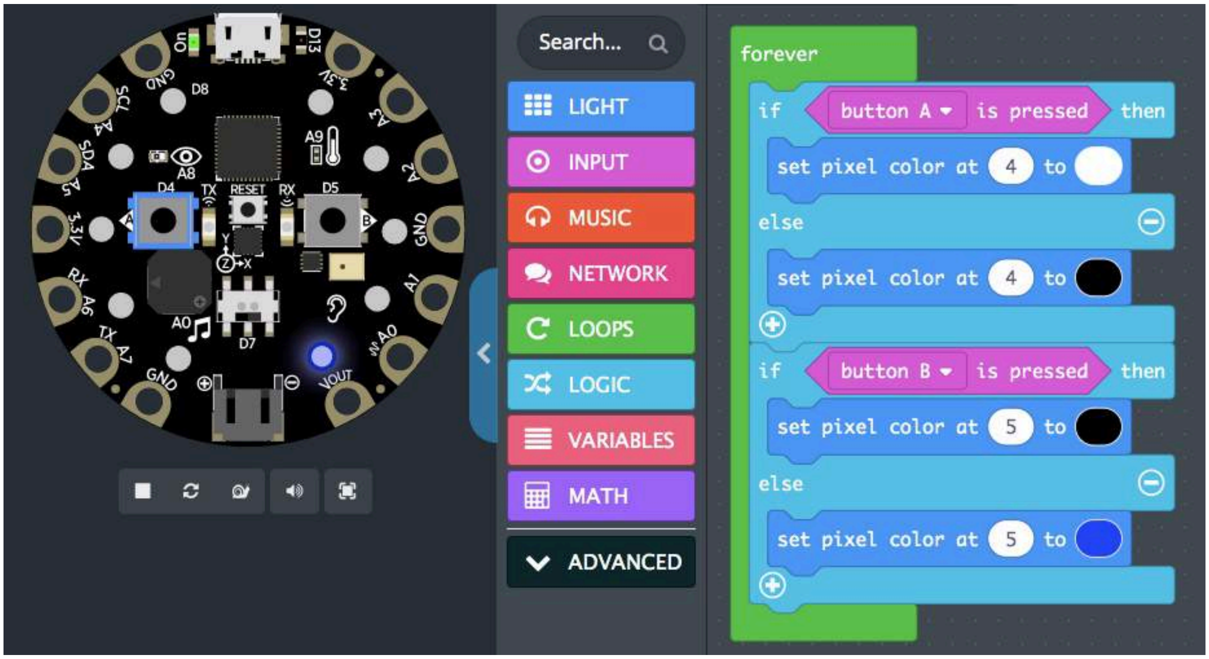
Second "If…Then…Else" logic block properly placed below first "If…Then…Else" logic block.

Second "If…Then…Else" logic block incorrectly placed within "Else" of first "If…Then…Else" logic block.

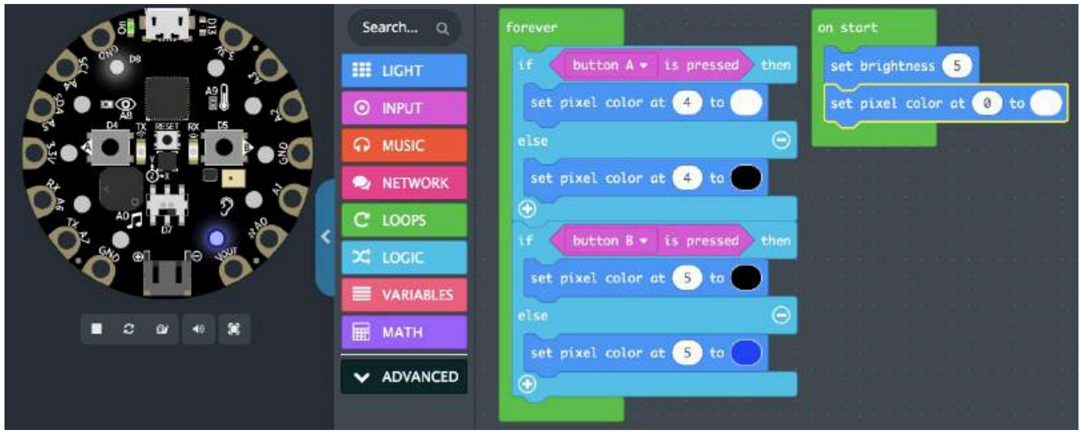
It is important to drop this into the right location. In the images above, dropping it below the current "If…Then…Else" statement block specifies that, during each pass through the sequence of statements within the FOREVER loop, we will first do the top "If…Then…Else" conditional logic block, then the second "If…Then…Else" conditional logic block.The image on the right, crossed out in red, demonstrates how we could have otherwise done a single

"If…Then…Else" conditional logic block, and one if it is NOT true. We would turn/leave the white LED #4 off and then check the conditional as stated in the "If…Then…Else" **subroutine within** the else statement. More on this later.

11. In the second "If…Then…Else" conditional logic block, change button A to button B and the at statement from 4 to 5.
12. To reverse the conditions determining when the LED turned on, in the second "If…Then…Else" conditional logic block, change the white color of the pixel within the "then" section to black. Change the black color of the pixel within the "else" section to blue.



13. Experience has shown these LEDs – especially the white LEDs – can be very bright, sometimes alarmingly so for those with certain visual functional diversities. Before flashing this over to the physical Circuit Playground Express, let's add in an "ON START" initialization block to determine the brightness of the LEDs, and to also turn on one of the white LEDs as a working symbol, as we've used in the Orange Unit prototypes.
    a. From the LOOPS category, drag an "on start" initialization block and drop it into an open area of the workspace.
    b. From the LIGHT category, drag a "set brightness" block and drop it into the ON START initialization block.
    c. From the LIGHT category, drag a "set pixel color at" block and drop it into the On START initialization block underneath the "set brightness" block.
    d. To finish, change the brightness from 20 to 5, and the set pixel color at 0 from red to white.



14. Do further testing in the simulator until you think it is working as it should.
15. In MakeCode, change the file name from "Untitled" to "pushButton," click on the save icon, and Download.
16. On the physical Circuit Playground Express, push down the "reset" button, and drag the new `circuitplayground-pushButton.uf2` file to the CPLAYBOOT device.
17. Do initial testing on the physical device, confirming it is working as it should, or identifying fixes needed within the code. This can include adjustments in the overall design, not just in the code as created through the above steps.
18. Make any changes as identified on the physical device within MakeCode, test further in the simulator, save a new copy and flash it to the physical Circuit Playground Express, and do additional testing there. Repeat this process until it works well for you, for your co-explorers, and perhaps even for your intended audience!

Key Takeaways

In this exercise, we extended our electronics prototyping skillset using the Circuit Playground Express microcontroller.

- We coded within the FOREVER function loop, extending the number of statements, or **lines of code**, that occur within each sequence through this loop.
- We studied the "If…Then…Else" **conditional** control flow statement, which uses a logic question such as "If button A is pressed," and provides a platform to then indicate that action when true. In our case, that meant turning on a white LED. The "Else" section lets us specify what to do if the question is false. In our case, that meant turning off, or leaving off, that same white LED.
- We found how it is often important to **initialize items**, for instance, by setting the brightness to a certain chosen level before those LEDs are turned on.
- We worked in **multiple layers of abstraction**, while balancing our prior and current knowledge and skills and our considerations regarding investment of time and capital to

move through this activity.

- There's quite a bit of electronics now "under the hood," and much of the engineering specifics are no longer known by us, although as used it performs much the same as did our later breadboard prototyping in the Orange Unit.
- How our blocks turn into text before being compiled and turned into machine-readable code is now hidden from us, but this trade-off gives us a relatively easy way to dip our toes into coding.
- The programming code that takes our FOREVER function loop and ON START initialization control flow statement and uses them to change to state of the electronics as instructed is also largely underneath the hood and unknown to us. In its design, though, is a pathway for us to do some extensive innovation-in-use of the software and associated electronics to meet our own chosen valued design choices.
- Abstractions, when designed and implemented well, provide us with new opportunities to go places we had otherwise only dreamed. But, there's always a risk that underneath those hoods are choices that ultimately may take us places we consider problematic or don't value. No matter the stage of design and implementation we find ourselves at, it is valuable to pause regularly throughout and reflect on the seen, the unseen, and the unforeseen in these **sociotechnical** artifacts.

Exercise: Testing Out Innovation-in-Use to Build a Five Digit Counter

Our Circuit Playground Express now performs as a tool fairly similar to the one we prototyped on our breadboard previously. But with the Circuit Playground Express, we have more LEDs available. This might be a good way to further expand our budding design, prototyping, and coding skills.

With five LEDs in the area of button A, what if we used them as a five digit counter?

Through trial and error, I found that the programming code behind MakeCode preset the 10 LEDs with numbers 0 to 9, in which LED 0 is nearest the green "On" LED of the Circuit Playground Express, and LED 9 is just below D13 and just above the A9 thermometer symbol of the Circuit Playground Express.

This brings to mind that there are five digits on the same half of the Circuit Playground Express as button A, numbered 0 to 4. There are five additional digits on the other half of the Circuit Playground Express, those numbered 5 to 9, on the same side as button B.

This design matches well the normative design of the human body with two hands (analogous to two buttons) with five fingers on each (analogous to five LEDs in proximity to each of the two buttons). This prompted new ideas for using the Circuit Playground Express to further advance our coding logic flows.

For this exercise, let's equate one push of each button with raising the fingers within that analogical hand. Once all five LED digits are raised within a hand, one with each new press of the associated button, the next press of the button will turn off all five of its LEDs.

Knowledge Probe

First, let's think through the logic of this new design a bit more, starting with the five LEDs of button A, LED pixels 0 to 4.

- In the last activity, for each pass through the FOREVER sequence:
    - If button A is pressed down, pixel at 4 is set to color white, whether it was already set that way or not.
    - If button A is not pressed down, pixel at 4 is set to color black, whether it was already set that way or not.

- This is an example of **combinational logic**. In combinational logic, the result depends only on the current states of the inputs. What we do in each sequence is determined just based on the current state of the system. It is not based on any past states of the system. Sometimes combinational logic is the absolute best choice, like what I do if I see a squirrel trying to eat fruit from my garden. It doesn't matter whether this is the first time or the 50th time. Combining species (squirrel), location (garden), conditions (fruits ripe or nearly so), and knowledge (squirrels eat this type of fruit), I'll chase it away. This is combinational logic. Sometimes combinational logic is the choice we go with just because it's at hand. I'll just keep chasing off the squirrel from the fruit tree because it's right next to my easy chair where I'm reading a book. But other times, we may use past logic to influence future logic. Perhaps squirrels haven't been here before, so chasing it off five times might make good sense. But on the sixth attempt to invade our fruit tree, we might logically consider it's time to do something more drastic than just chasing it off. Then there are times when combinational logic just doesn't make sense. Consider the ceiling lights in a kitchen. What if combinational logic was only available to turn them on? Combining intent (cooking dinner), conditions (after dark), and knowledge (I've previously used my vision to select and use ingredients) suggests we turn on the light. If this meant we needed to walk over and push and hold down a button to keep the light on, this combinational logic would keep us from doing the rest of the activities.

- In this activity, we not only want the press of the button to turn on an LED; we want it to move down the line, lighting a sequence of up to five LEDs. What action using Button A could change the pixel number being lit next?

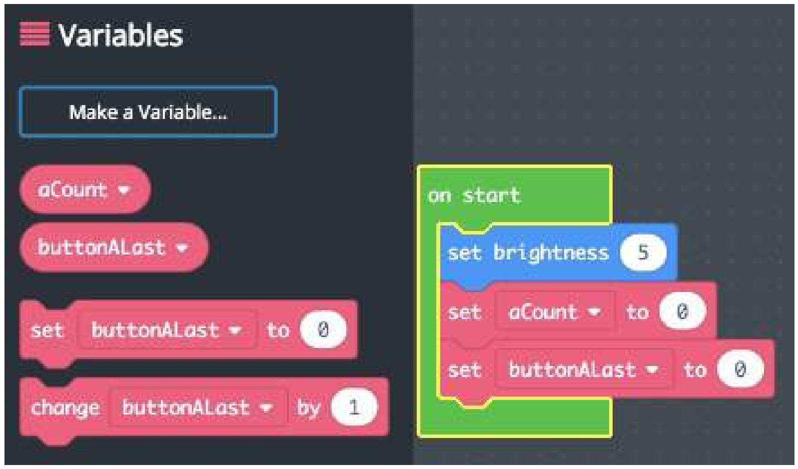
    a. Each new pass through the FOREVER sequence while button A is pressed, move to the next pixel?

b. Every 100 passes through the FOREVER sequence while button A is pressed, move to the next pixel?
c. At initial change of button A from not pressed to pressed, move to the next pixel?
d. Something yet different from that?

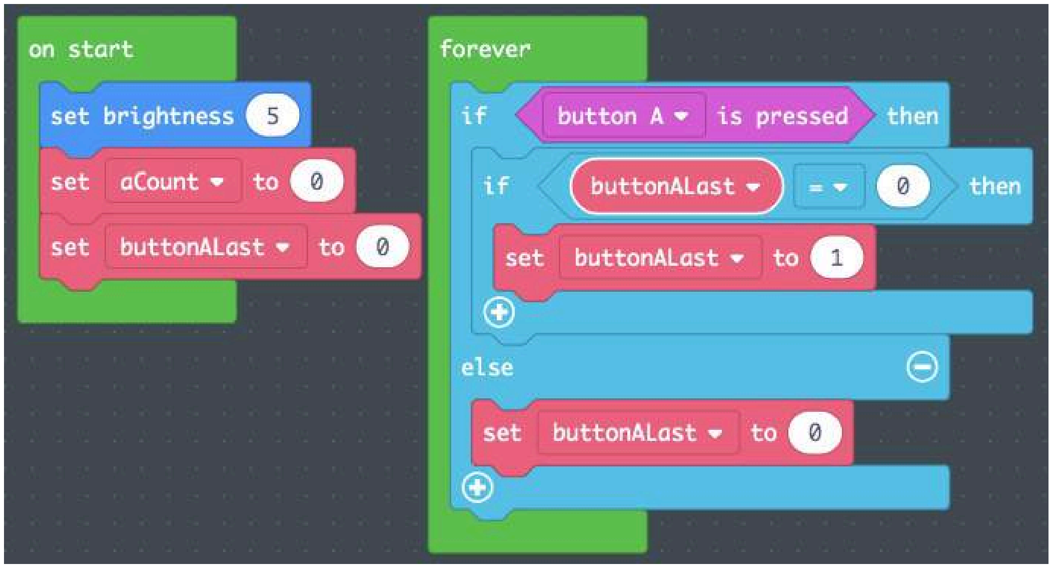
- Back when I used to use a click counter to keep track of an event, a clicker either: 1) started a counter that ran until I did a second click, which matches well a and b above, or 2) it increased the count by one every time I hit that clicker, which matches well with option c above. For now, let's go with option c, then.

Steps

1. Start a new project in MakeCode.
2. From the LOOPS category, drag over an "on start" initialization block, and drop it somewhere in the workspace.
3. From the LIGHT category, drag over a "set brightness" to the ON START initialization block and set it to 5 (or your chosen value) to keep the LEDs from being too bright.
4. For this exercise, we'll need some **variables** to keep check on specific things. We'll start by creating these variables within the VARIABLES category. Create each by clicking on "Make a Variable":
    - Name one "aCount". We'll use this to determine how many times the button has been pressed recently.
    - Name one "buttonALast". We'll use this to remember if it was pressed down during the last sequence with the number 1, or if it was not pressed down during the last sequence with the number 0.
    - Using variables to remember recent past states and to determine current events is the basis for **sequential logic**. What to do at this point in a sequence is determined not just on current state, but also on past states.
5. From the VARIABLES category, drag a "set _____ to 0" to the ON START initialization block. This will be labeled the same as the last variable created.
    - Copy two of these over to the ON START initialization block.
    - Use the ∇ label to drop down options, and set one to aCount and the other to buttonALast.
    - Leave each set to 0 within the ON START initialization block.



6. As in the last exercise, from the LOGIC category, drag over an "If…Then…Else" block and drop it into the FOREVER function.
7. As before, from the INPUT category, drag over a "button A is pressed" block and drop it into the <True ∇> window.
8. This time, rather than turning a specific pixel white or blue, we'll instead use a new "If…Then" block (no "…Else" needed here) within the "Then" section. We'll use this to determine whether this is a new press of the button, or an ongoing press of that button.
9. From the LOGIC category, from within the Comparison section, drag over a <0 = 0> block and use it to replace the <True ∇> of this second "If…Then" logic statement.
10. From the VARIABLES category, drag a "buttonALast" variable block and replace the left "0" of the comparison we just added. Leave the other "0" as is.
    - This "If…Then" conditional is now checking to see if button A was NOT pressed previously as a first step within the "If…Then" conditional in which "button A is pressed" was True.
    - To complete the sequential conditional logic being setup with the buttonALast variable, we need to:
        - Drag the "set _____ to 0" variable block within the VARIABLES category and drop it into the else section of the parent "If button A is pressed" logic statement. Update it to read "set buttonALast to 0," as this "else" section is reached only when button A is NOT pressed. We want to remember that in the past iteration of this FOREVER function sequence, button A was NOT pressed.
        - Drag a second "set _____ to 0" variable block within the VARIABLES category and drop it into the "Then" section of the "If buttonALast = 0" logic statement. Update it to read "set buttonALast to 1" as this "Then" section is reached when button A is first pressed, and we want to remember it is already pressed moving forward into future iterations of this FOREVER function.

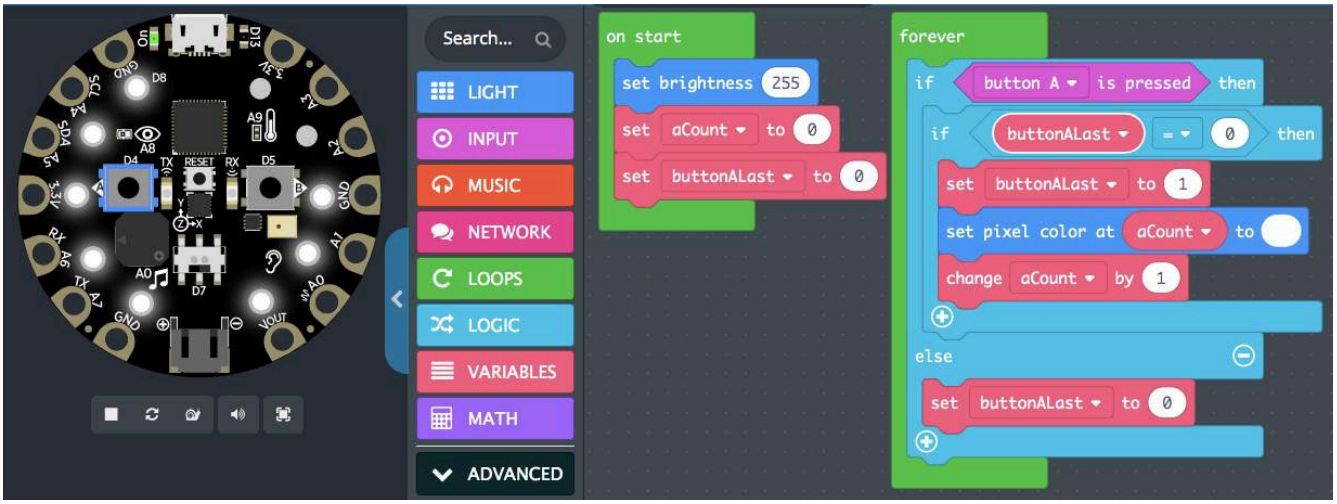


Looking at our code, we see that within the FOREVER loop, we now have a parent "If…Then…Else" block that uses a "button A is pressed" block conditional as its starting point.

- If "False," that is, button A is NOT pressed, we move to the "Else" section and set the buttonALast Variable to 0, or False.
- If "True," that is, button A IS pressed, we move to the "Then" section. But rather than immediately taking an action, this time we first do another logic check, this time, whether button A was NOT pressed last time using an "If buttonALast = 0" condition. That's because we already know we set the variable buttonALast to 0 whenever the "button A is pressed" condition is "False."
- Now, only when "button A is pressed" is "True" AND "buttonALast = 0" do we change the block Variable to "set buttonALast to 1."

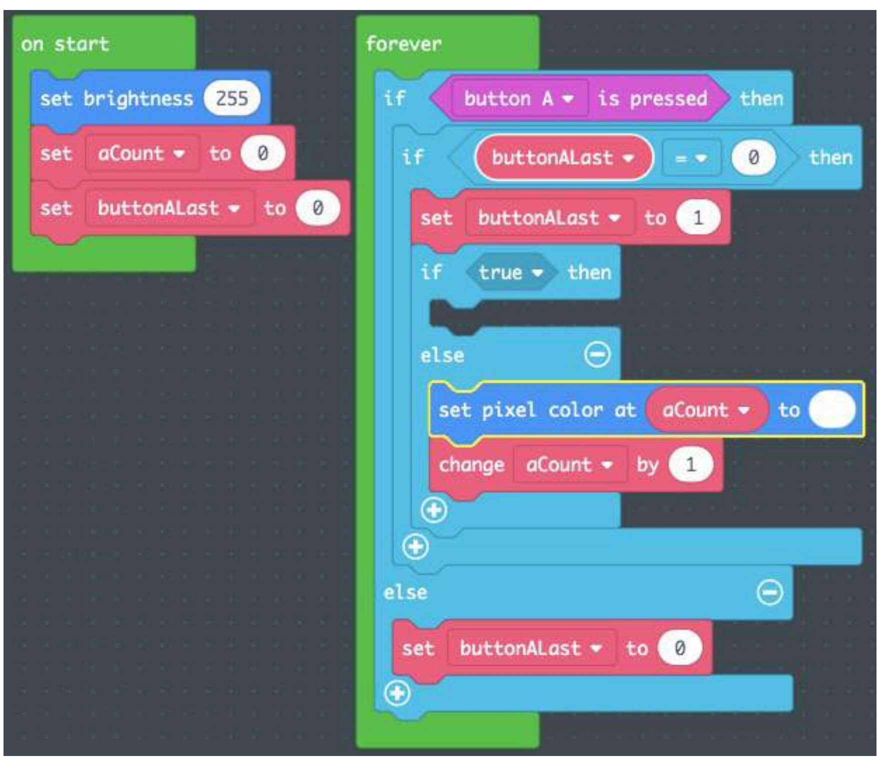
11. Within the "If buttonALast = 0 Then" statement, add a "set pixel color at" block from the LIGHT category.
    a. From the VARIABLES category, drag over an "aCount" block and drop it into the 0 window of the new "set pixel color at" block, so that it now reads "set pixel color at aCount."
    b. Change the red window to white.
12. From the VARIABLES category, drag a "change _____ by 1" block and drop it under the "set pixel color at" block. Use the ∇ key to set this block up to read "change aCount by 1."

If you press button A in the simulator now, what happens?

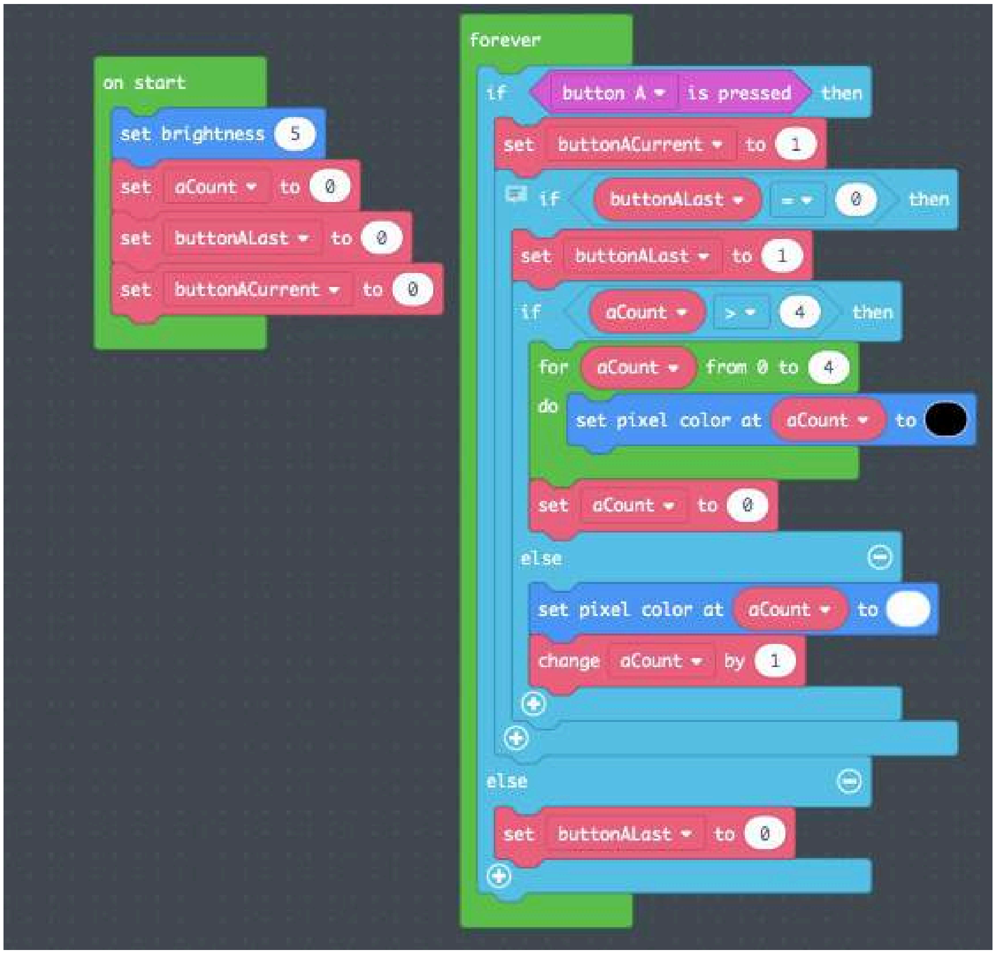


Here, we see that after 8 individual presses of button A in the simulator, eight pixels have been turned to white.

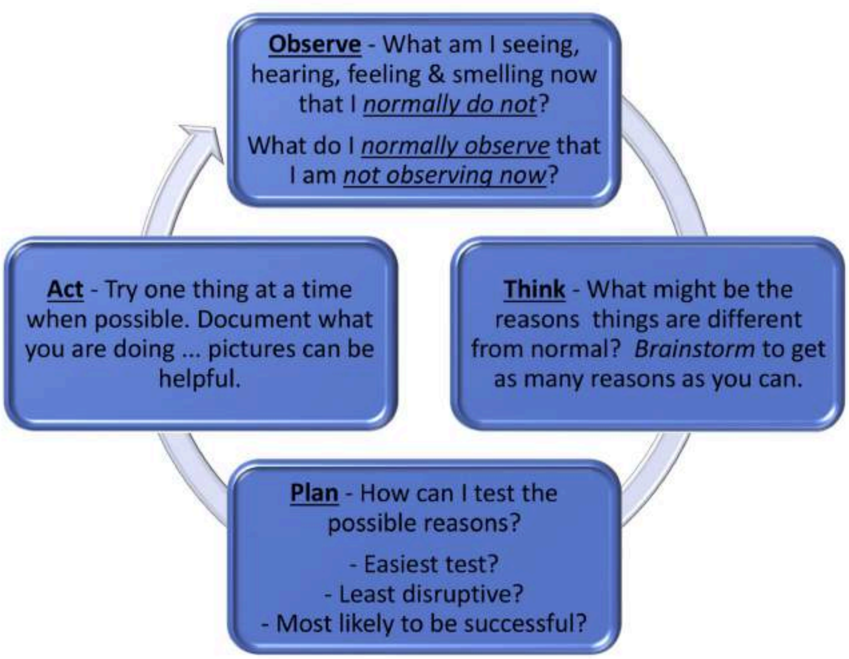
13. To add a conditional limiting the number of pixels turned to white using button A, we need yet a third embedded "If…Then" logic statement, this one as with the the top parent including an "…Else."
    - From the LOGIC category, drag over an "If…Then…Else" block and drop it under the "set buttonALast to 1" block within the "if buttonALast = 0 Then" statement.
14. Drag the blocks "set pixel color at aCount to white" and "change aCount by 1" into the "Else" section of this new "If…Then…Else" statement. It should look as follows:



15. From the "Comparison" section of the LOGIC category, drag over a "<0 = 0>" block and replace the "true" window of the new "If…Then" statement.
    a. From the VARIABLES category, drag an "aCount" block and drop it in the left 0 window.
    b. Change the "= ∇" window to ">".
    c. Change the right 0 to 4.
    d. Now, "If aCount is greater than 4, Then" we can turn all of the white pixels on the button A side to black. Else, we'll turn the next aCount to white.
16. To turn the five button A pixels from white to black, we're going to use a new control flow statement called the **FOR loop.**
    - From the LOOPS category, drag a "for index from 0 to 4" block and drop it within the "If aCount > 4 Then" statement.
17. From the VARIABLES category, drag an "aCount" block and drop it where the "index" block is now located within the new FOR loop.
18. From the LIGHT category, drag a "set pixel color at" block and drop it inside the FOR loop.
    a. From the VARIABLES block, drag a "aCount" and drop it into the new "set pixel color at" block, replacing the 0.
    b. Change the red window to black in this "set pixel color at" block.
    c. This FOR loop resets the aCount back to 0, goes through the "Do" section, then increases the aCount by 1 and repeats the "Do" section, continuing until it reaches the maximum number. In our case, this means it will go through aCount from 0 to 4, turning the pixels black.
19. Finally, add in a "set _____ to 0" from the VARIABLES category and drop it underneath the new FOR loop, using the ∇ to set this to a "set aCount to 0" block.



20. Use the simulator to test this prototype. Every sixth press of button A should now reset our counter to zero pixels lit. If this is **Not Yet** the case, **Fail Forward** using the troubleshooting process to reach **Yet**.



21. Once all looks good in the simulator, save this code as pushButtonCounter, flash it to the physical Circuit Playground Express, and give it a field test.

Ta-da!!! Now, let's create a Button B counter in blue to stay with the theme of that button.

Button B Counter Design Probe

We have more design decisions to make. We can either do the new "If button B is pressed Then" logic control flow statement **in sequence with** the "If button A is pressed Then" logic control flow statement within the **same FOREVER function**, **or** we can place it in a **new FOREVER function**.

Should one follow the other, or should each run independent of the other?

We also need to decide the order in which button B lights the pixels.

Should it flow from top to bottom on the right, just as it does for button A on the left? This means we need to start with pixel 9 and move down to pixel 5 in order. Or should we go in pixel order from pixel 5 to pixel 9?

For this exercise, we will conduct these tests independent of each other, starting from the top of the pixels as seen in the simulator. How might this be shaping your learning? How might this shape your use of this microcontroller push button counter moving forward?

Steps

Rather than taking the time to go through the copying/pasting and remixing of the first "button A" FOREVER function to create a second "button B" FOREVER function step by step, let's commit the time to noticing the key changes needed to have this fit within the design profile listed above.

1. In your laptop web browser, download this UF2 code:
    - `circuitplayground-twoPushButtonCounters.uf2`
    - If it does not do so automatically, select to download this file.
2. Flash the downloaded `circuitplayground-twoPushButtonCounters.uf2` file to the physical Circuit Playground Express and perform some initial testing of this prototype.
    - Does it fit the design profile as stated above?
    - Does it work well for you and your valued doings?

3. Go to the MakeCode homepage and drag the downloaded file `circuitplayground-twoPushButtonCounters.uf2` onto the MakeCode screen. Take a minute to test this prototype in the simulator. How does it compare to the physical testing process?
4. Both FOREVER functions have 19 "lines of text," so to speak, within the blocks.
   - How many are the same? Why?
   - How many are different? How might they have changed?

Side Note on the For Loops

Within the Button A FOREVER function, the FOR loop is written as:

```
for aCount from 0 to 4, set pixel color at aCount to black
```

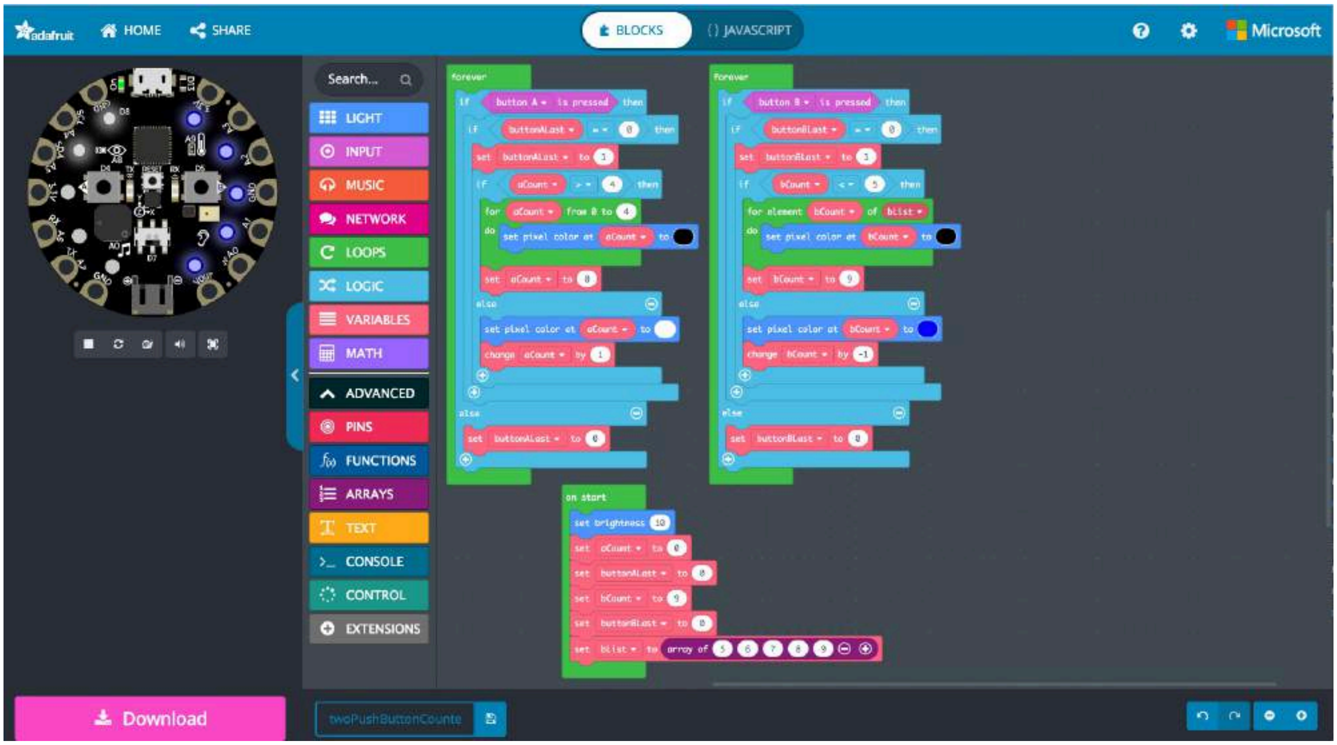
In this "For" control flow statement, aCount starts at 0 and the pixel at 0 is set to black. aCount is increased by one, and if it's still less than or equal to 4, that pixel is set to black. In this way, it sets pixels 0, 1, 2, 3, and 4 to black.

Within the Button B FOREVER function, the FOR loop is written as:

```
for element bCount of bList, set pixel color at bCount to black
```

Since Button B does not work with pixels 0, 1, 2, 3, and 4, but instead works with pixels 5, 6, 7, 8, and 9, this basic "For" control flow statement wouldn't work. So in the ON START initialization control flow statement, a new bList array was created, with the numbers 5, 6, 7, 8, and 9 put into that **list array**. In this "For" control flow statement, every bCount of bList is used to do a "set pixel at" action.

An aList could have been created with 0, 1, 2, 3, and 4 within this array, replacing the "for aCount from 0 to 4, do" statement with "for element aCount of aList, do" statement. In setting up the button A FOREVER function, you stepped through the simpler version as MakeCode wrote the block code with this simpler FOR loop block. But both forms of the FOR loop function the same, and using the list arrays would give greater flexibility moving forward. This demonstrates how there are often multiple ways to execute code, some of which depend as much on programming preference as on performance.



Key Takeaways

Sequential logic is used when past conditions (such as the previous state of a switch or the last LED that was lit) should impact what is to be done in the current sequence of a control flow statement. In our case, this was used to determine actions to be done or not be done within a sequence of "If…Then" control flow statements.

Multiple embedded "If…Then" and/or "If…Then…Else" control flow statements can be used to move **from more general** True/False, Yes/No logic questions **to more specific** logic questions.

Variables are data items whose values can be changed. In this case, the variables aCount and bCount were used to determine which of five pixels associated with button A and button B, respectively, should be lit. The variables buttonALast and buttonBLast were used to only change the condition of an LED pixel when the button's state is first changed.

Arrays provided a list specific to button B that was used to give greater flexibility regarding which LED pixels should be turned on or off. This is an example of how, often in computing, an indexed set of related elements can prove extraordinarily useful to expand the possibilities regarding data sets held within variables.

Choices shape our pathways forward. Sometimes the way forward is to use written guides and hints, or even to download and use already written code. While useful, it can make it more difficult to know what has shaped these resources as compared to those times when you participate in a community of practice to design and prototype something. Even then, it is almost always the case that you are working at a relatively high level of abstraction, in which case there are still hidden choices underneath the hood.

As an example of this last takeaway, you may find it helpful to glance at the JavaScript text to which the blocks are converted in the MakeCode twoPushButtonCounters code. The “for aCount from 0 to 4, do” as used in the button A forever function could have been changed to “for bCount from 5 to 9, do” in the button B forever function. But a MakeCode block for this type of for loop was not created, and so could only be done by making this change to the code in JavaScript, potentially a more difficult process for those shifting from block to JavaScript programming languages.

Wrap Up

To build further on the key takeaways from the above three exercises, take 11 minutes to watch the video on “Programming Basics: Statements & Functions” from Carrie Anne Philbin’s Crash Course Computer Science series hosted by PBS Digital Studios:

Coders shape how things work even as they are shaped by how past coders shaped the code. We started with the blinky code. But behind that code was more complex code, hidden from view.

- Who set up the blocks we used to work the way they did?
- How did it work well for us? How could it have worked better?
- Did we always understand how one block worked compared to another? Why or why not?

By the end, you followed instructions to create your own functions with loops and variables. Each function included other, hidden functions and variables incorporated into sequences of code operating sometimes under only certain conditions.

There’s a trend to more strongly shape programs and the distributed apps based on those programs. In doing so, control is taken away from us. In this chapter, we’ve worked to discover how we can get control, but only if the Integrated Development Environments like MakeCode and programmable electronic systems like the Circuit Playground Express have been designed and implemented in ways that facilitate innovation-in-use by those at every level of the social technical artifact.

- In what ways have you been shaped by the Circuit Playground Express, selected electronics, MakeCode, and these exercises?
- In what ways have you helped to shape the Circuit Playground Express, selected electronics, MakeCode, and these exercises?
- If you were return to the top, in what ways might you work to shape the Circuit Playground Express, selected electronics, MakeCode, and these exercises?
- Who might be impacted by these ***things*** because of the various mutual shaping and ampli-

fication by the many ***people*** involved: individuals and communities of practice with unique histories, intersectionality, expertise, valued doings, and power, privileges, and oppressions?

Comprehension Check

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. An online version of this comprehension check is also available.]

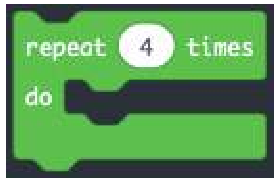
We've spent time learning how to write basic programming code to create control flow statements such as FOREVER and FOR loops, ON START initialization, and "If…Then" conditional statements. We've used variables and arrays to facilitate use of data within the code to facilitate coding through abstraction. And we've used FOREVER and ON START functions to break down code into distinct units.

Let's check our learning of these core coding concepts by bringing the following blocks together to create a new ON SHAKE function that plays a short siren and blinks between two different LED pixels rings before turning all LED pixels off. This should be added to the Button A and Button B Five Digit Counters.

The design criteria includes:

- Play the siren sound on shake.
- The ring of LED pixels should change between two different LED colors four times.
- At the end, all sound and all LED pixels should be turned off.
- This sequence of events should only happen when the slide switch is moved to the right. If the slide switch is moved to the left, no siren or LED pixel rings will occur.
- Regardless of the state of the slide switch, the five digit counters should continue to work whenever the Circuit Playground Express is not being shaken.

Each of the following should be used within the ON SHAKE function:


repeat
4
times
do


play sound
ba ding


set all pixels to


show ring

2A: The Methodological Landscape

Background Knowledge Probe

Reflect back on a way that you've taken something you have and used it in a way it wasn't meant to be used, in order to get something done that you couldn't do otherwise. Then, consider the landscape of methods you used to accomplish this innovation-in-use.

- What are some ways you've used this and other methods and practices over time in similar contexts? How they have evolved over time? What are some ways these methods and practices may have originated as a part of your everyday or occasional praxis?
- How might these methods and practices have shaped your work so far through this book, both positively and negatively?
- What are some ways these methods and practices may be further developing as you work through this book?

Tinkering as Research

My work as a tinker began very early as my family encouraged me to tear apart any and all no-longer-used tools around me. Of course, while the goal was to discover what was inside and to make them work again, those earliest days were almost always "Not Yet" fail forward moments. At around 5, I was offered the opportunity to earn 50 cents an hour to help with basic sweeping, cleaning, and moving of sawdust and wood in the family sawmill. I didn't know it at the time, but it was my first apprenticeship role. It seemed boring, but what was really happening was a first discovery of the music and performance of the sawmill as sociotechnical system. It is a combination of senses, knowledge, culture, community, and technology artifacts working in concert. As with most, if not all, such lived experiences and practices in our lives, it was not something that could be learned through a banking concept of education. Rather, it could only be done through community inquiry as part of a community of practice, which itself is rooted within the apprenticeship model of learning in which students study with master craftspeople.

I am first generation American, first generation English-as-first-language speaker, and first generation college graduate, let alone PhD. And yet, as I advanced through university and worked professionally the last two and a half decades within the university, I have found some aspects of the methodological landscape to be something I knew of, and practiced in part, even during my early years of work in that family community. While different perspectives may be incorporated, and while they may use different terms and practices, the research process is incorporated into many different disciplines beyond the academic realm. Still, it is in the academic disciplines that special emphasis has been given to identifying and practicing the perspectives and assumptions underlying our research paradigms. It has proven an extraordinarily valuable resource in addition to my apprenticeship learning, opening up new avenues for exploration, advancement, fine-tuning, and new discoveries of practice and knowledge.

Table 1. Meta-theoretical assumptions behind the three research paradigms.[1]

Paradigm	**Positivist**	**Interpretive**	**Critical**
Reason for research	To discover regularities and causal laws so that people can explain, predict and control events and processes.	To describe and understand phenomena in the social world and their meanings in context.	To empower people to change their conditions by unmasking and exposing hidden forms of oppression, false beliefs and commonly held myths.
Ontology – the nature and existence of social reality	Assumes an ordered and stable reality exists out there waiting to be discovered, irrespective of an observer.	Assumes reality is socially constructed, fluid and fragile, and exists as people experience it and assign meaning to it.	Transcends objective-subjective poles and assumes reality is socially constructed but nevertheless perceived as objectively existing.
Epistemology – the nature and the ways of knowing	Takes an instrumental approach to knowledge: knowledge enables people to master and control events. Knowledge represents reality, is stable and additive; statements about reality are true only if they are repeatedly not empirically falsified.	Takes a practical approach to knowledge; aims to include as much evidence about the subject, the research process and context as possible to enable understanding of others' lifeworlds and experiences, and how the researchers came to understand them.	Takes a dialectical approach to knowledge. Knowledge enables people to see hidden forms of control, domination and oppression, which empowers them to seek change and reform existing conditions and social order.

1. Cecez-Kecmanovic, Dubravka, and Mary Anne Kennan. "Table 1. Meta-theoretical assumptions behind the three research paradigms." From "The Methodological Landscape." In *Research Methods: Information, Systems, and Contexts*, edited by Kirsty Williamson and Graeme Johanson, 127–55. Elsevier, 2018. https://doi.org/10.1016/B978-0-08-102220-7.00005-4.

The logic of scientific explanation	The dominant logic of inquiry is hypothetic-deductive: hypothesized relations among variables (logically derived from causal laws or theories) are empirically tested in a way that can be repeated by others.	The dominant logic of inquiry is inductive and develops idiographic descriptions and explanations based on studies of people and their actions in context; explanations need to make sense to those being studied as well as to the researchers and their community.	The logic of inquiry can be deductive and inductive but also abductive, seeking creative leaps and revealing hidden forces or structures that help people understand their ways of changing them.
Ethics and claims about values and normative reasoning concerning with what 'ought' to be	Assumes both natural and social sciences are objective and value-free, operating separately from social and power structures; ideally positivist researchers are detached from the topic studied and collect value-free facts.	Questions the possibility of value-neutral science and a value-free research; values are seen as embedded in all human actions (including researchers) and hence are inevitably a part of everything we study, without the judging of one set of values as better than another.	Any research is a moral-political and value-based activity; critical researchers explicitly declare and reflect on their value position(s) and provide arguments for their normative reasoning.

Key Takeaways: Purposes of Research Paradigms

Positivist Paradigms

- Behind all of our electronics and programming code, and much of traditional scientific research, stands **deductive reasoning**, also known as deductive logic. The goal in this paradigm is to produce and continuously adapt predictable, reliable, anticipatable, and controllable events and processes.
    - **Boolean logic** brings together one or more propositions into simple Yes/No, True/False, or 1/0 statements.
    - Boolean logic moves from more general to very specific in a top-down logic, in which a conclusion is reached deductively, narrowing the range under consideration until only the conclusion is left.
    - The ordering of our Boolean logic tests matters!

Interpretive Paradigms

- Within a sociotechnical framing of technologies, the continuous changes of our technological tools are events and processes of contingency, which are not fully predictable. Case study analysis and situated evaluation use **inductive reasoning** to develop interpretive understandings of complex social phenomena.

- While inductive reasoning and interpretive paradigms generally do not serve in the building of the physical technical system, they are essential to **meta-design**, in which users become co-designers during the formal design process, and throughout ongoing use of a product.
- Evaluation of "it" moves from the product generally, or even the *idealization* of that product in design thinking terms, to the **innovation-in-use**, a situation-specific use of the product by individuals and cohorts.

Critical Paradigms

- Within a critical sociotechnical framing of technologies, deductive reasoning in support of the physical construction of a product and inductive reasoning in support of meta-design and innovation-in-use are both essential, but not sufficient. It is necessary to also incorporate moral and ethical questioning to unmask and expose hidden forms of oppression, false beliefs, and commonly held myths. To do so, **abductive reasoning** is also used, bringing together the best resources at hand to attempt creative leaps in understanding.
    - Critical research requires work in community, with community, and for community using methods such as participatory action research and participant observation to apply a **dialectic** approach to knowledge.
    - As the intersectionality of individuals brings together oppressed and oppressors in cycles of **action-reflection**, the resulting new understanding of reality serves to bring forward new starting points for deductive, inductive, and abductive reasoning in support of meta-design and innovation-in-use of sociotechnical artifacts.
    - Knowledge is not static or individualistic. At its best, **culturally responsive, sustaining pedagogy** using a critical sociotechnical lens works to foster and incorporate a dynamic, synergistic relationship between the cultures of home, community, schools/universities, library/museum spaces, for-profit/not-for-profit institutions, and many other co-exploration spaces.

Sources for further investigation:

- Bruce, Bertram C., Andee Rubin, and Junghyun An. "Situated Evaluation of Socio-Technical Systems." In *Handbook of Research on Socio-Technical Design and Social Networking Systems*, edited by Brian Whitworth and Aldo de Moor, 2: 685–98. Information Science Reference. Hershey, PA: IGI Global, 2009. http://hdl.handle.net/2142/9710.

- Cecez-Kecmanovic, Dubravka, and Mary Anne Kennan. "The Methodological Landscape." In *Research Methods: Information, Systems, and Contexts*, edited by Kirsty Williamson and Graeme Johanson, Second Edition., 127–55, 2018. https://doi.org/10.1016/B978-0-08-102220-7.00005-4.
- Fischer, Gerhard, and Thomas Herrmann. "Socio-Technical Systems: A Meta-Design Perspective." *International Journal for Sociotechnology and Knowledge Development* 3, no. 1 (2011): 1–33. https://doi.org/10.4018/jskd.2011010101. Also available at https://l3d.colorado.edu/wp-content/uploads/2021/01/Published-JOURNAL-version.pdf.
- Ladson-Billings, Gloria. "Toward a Theory of Culturally Relevant Pedagogy." *American Educational Research Journal* 32, no. 3 (September 1995): 465–91. https://doi.org/10.3102/00028312032003465.
- Milner, H. Richard. "Race, Culture, and Researcher Positionality: Working Through Dangers Seen, Unseen, and Unforeseen." *Educational Researcher* 36, no. 7 (October 2007): 388–400. https://doi.org/10.3102/0013189X07309471.
- Rodgers, Carol R. "Defining Reflection: Another Look at John Dewey and Reflective Thinking." *Teachers College Record* 104, no. 4 (June 2002): 842–66. http://www.canr.msu.edu/bsp/uploads/files/Reading_Resources/Defining_Reflection.pdf.

The Meta-Theoretical Landscape in Practice

Since its release in 2015, Robin Wall-Kimmerer's *Braiding Sweetgrass: Indigenous Wisdom, Scientific Knowledge, and the Teachings of Plants* has been an ongoing source for expanding my own meta-theoretical landscape. As I recommend this book to new generations of readers and listeners, their response is universally the same. In the context of this text, her chapters "Asters and Goldenrod" and "Learning the Grammar of Animacy" especially highlight ways in which Indigenous wisdom, scientific knowledge, and the teachings of the more-than-human persons around us complement each other. Together, this diversity of knowledge and experience enriches and greatly expands our reading of the world, and of the word as part of action and critical reflection praxis.

"Asters and Goldenrod" brings forward the ways in which our cultural knowledge and wealth is taken away from us, to be replaced by dominant paradigms and ways of being and doing. We lose a core part of who we are and what we can be in order to be that which we are told to be. It is a process of removing parts of our humanity and pushing us to be a disposable thing.

"Learning the Grammar of Animacy" highlights how the subjects around us—the living persons typically beyond that which we can conceive to exist—can and need to serve as wise elders. Listening to them can lead to creative leaps of understanding. And so it was for Kimmerer, as she discovered that the walls keeping her from learning her cultural language, Potawatomi, were ones of language, and specifically the English language. In Potawatomi, seventy percent of the words are verbs, indicating the animacy of those around us. Plants and streams alike are animate, thinking, feeling entities. This compares to English, which is primarily noun-based, and which objectifies much around us, turning it into a thing.

For us to "rapidly shift from a 'thing-oriented' society to a 'person-oriented' society," as we are called to do by Rev. Dr. Martin Luther King, Jr. in his 1967 "Beyond Vietnam" speech, we need something beyond what our existing language and landscapes can often provide. It does not mean we must put aside who we are, how we live, what we value, and so much more. But it does mean we need to enter into new communities of practice, ones that help us address the shortcomings in our own meta-theoretical landscapes, even as we bring forward our own expertise and knowledge that has built from these landscapes. Often, it is not an "either-or" work, but a "both-and" proposition that truly helps us reach ever greater individual and collective fulfillment. Lastly, it's essential that this is done in ways that push for greater justice for all, and especially for those most pushed to the margins and those who are oppressed.

To finish this chapter, let's explore one further example of the meta-theoretical landscape in practice today, as helpfully summarized in René Ostberg *Encyclopedia Britannica* page on "transhumanism."[2] Since being introduced by Julian Huxley in his 1957 essay "Transhumanism," there has been a growing number of organizations, schools, and business entrepreneurs exploring ways in which scientific advances can facilitate the extension of human life and overcome human limitations through enhanced human cognition and augmented capabilities. Institutes, associations, and individuals have worked to develop statements, books, and voices building on the vision, core concepts, and principles as different branches of thought have emerged within transhumanism. A landmark example can be found in Ray Kurzweil's *The Age of Spiritual Machines* (1999), in which the distinctions between man and machine are blurred as computers develop operational capabilities equivalent to those of the human brain. Silicon Valley venture capitalists such as Larry Page, Jeff Bezos, and Elon Musk have gone on to launch a range of research and development companies dedicated to various aspects of transhumanism-centered technology development and diffusion, through a range of biotechnologies, brain chips, animal cloning, exoskeletons, and even space travel and colonization.

While there are many ways in which we can move from an "either-or" to a "both-and" proposition, there are also those times when we need to explore the deeper logic and ethics underlying scientific explanation, and the central claims about values and normative reasoning which together shape research and development. Data mining, analytics, visualization, and curation can be done in many

2. Ostberg, René. "transhumanism." *Encyclopedia Britannica*, 3 Nov. 2022, https://www.britannica.com/topic/transhumanism. Accessed 13 March 2023.

different ways using various forms of logic and ethical assumptions. Information in the form of data goes on to underpin vast initiatives that shape information access, management, and use, which further shapes analysis, design, maintenance, and evolution of information infrastructures. As we move further along the data, information, knowledge, wisdom pyramid, the logic and ethics underlying scientific explanation and their central claims about values and normative reasoning further shape the creation of knowledge. They do this either in ways that allow a few to exercise power over the many, or in ways that allow the advancement of power within each and between all as part of active works of social justice. In his 2004 *Foreign Policy* "Transhumanism" article, Francis Fukuyama asks a couple of provocative questions: "If we start transforming ourselves into something superior, what rights will these enhanced creatures claim, and what rights will they possess when compared to those left behind? If some move ahead, can anyone afford not to follow?" He goes on to note, "Transhumanism's advocates think they understand what constitutes a good human being, and they are happy to leave behind the limited, mortal, natural beings they see around them in favor of something better. But do they really comprehend ultimate human goods?"

In WNYC Studio's *On the Media* "Salvation Through Technology" from January 18, 2023, Brooke Gladstone brings back a 2021 segment in which she interviewed Meghan O'Gieblyn, author of *God, Human, Animal, Machine: Technology, Metaphor, and the Search for Meaning,* taking us through her journey from her fundamentalist Evangelical church roots and Bible college training to her extensive reading about technology. It led her to an obsession with transhumanism in which most identify as atheists with a worldview based on materialism. Over time, O'Gieblyn began to see in transhumanism a theoretically plausible offering which included "basically everything that a religious worldview had once offered me, but it was doing so through science and technology." Bringing these two parts of her history together has helped her explore ways in which science and technology often recapitulate old spiritual narratives in ideas such as transhumanism, exposing the shared assumptions of Christian creationists and transhumanist tech leaders. But this isn't limited just to the transhumanist space, as more generally, the growing ubiquity of artificial intelligence is "almost like we're making the physical material world conscious, again, much like this old animus cosmology where we believe that spirits lived in rocks and trees and that the world was alive and that we could have social relationships with physical objects."

This shared assumption of Christian creationists and transhumanist tech leaders is distinct from Kimmerer's call for us to bring into consideration a grammar of animacy as a complement to our Western scientific explorations. It takes time and effort to unpack the ontology, epistemology, ethics, and claims about values and normative reasoning concerning what "ought" to be that also inform the research paradigms informing our works; these shape the deductive, inductive, and abductive logic within the various scientific explanations and their applications within our ongoing professional research and development activities. As we develop programming code to address important issues of the day, are we doing so in a person-oriented way, or in ways that seek to make the technology become person?

As critical researchers, we need to explicitly declare and reflect on our value position(s) and provide arguments for our normative reasoning. This can only be done through conversation with those different from us as part of both action and reflection phases of our person-centered community inquiry. In so doing, we move from an exclusive focus on the question of inquiry within the moment, to also open opportunities to unpack further the unseen and unforeseen normative reasoning which is shaping our actions and our reflections on those actions.

Lesson Plan

By demystifying these methodologies and introducing new ideas, we seek to further advance our ability to consider and tell counterstories.

In the technical session, we will see how the positivist paradigm of research is fundamental to both electronics and modern programming.

This session helps us better explore the meta-theoretical assumptions and research paradigms that shape our sociotechnical environment and the social and technical products that arise from our hands-on activities that put these assumptions and paradigms into practice. What within your Professional Journal Reflections from last session has helped you move into this session? What has proved problematic, requiring further codification and decodification in dialogue with others so as to more closely align to the valued beings and doings of all?

Essential Resources

- Kimmerer, Robin. "Asters and Goldenrod" and "The Grammar of Animacy." In *Braiding Sweetgrass: Indigenous Wisdom, Scientific Knowledge, and the Teachings of Plants*, 39–47. Minneapolis, MN: Milkweed Editions, 2013. (May be available through EBSCOhost eBook Collection)
- NPR, "Main Character of the Day," March 14, 2023. "Neurotech could connect our brains to computers. What could go wrong, right?" https://www.npr.org/2023/03/14/1163494707/neurotechnology-privacy-data-tracking-nita-farahany-battle-for-brain-book (Listen to the interview of Nita Farahany, professor of law and philosophy at Duke Law School, conducted by NPR's Manuela López Restrepo. Link also provides some associated summary text.)
- Wolske, Martin, Colin Rhinesmith, and Beth Kumar. "Community Informatics Studio: Designing Experiential Learning to Support Teaching, Research, and Practice." *Journal of Education for Library and Information Science* 55, no. 2 (April 2014): 166–77. http://hdl.handle.net/2142/48952.

Additional Resources:

- Cecez-Kecmanovic, Dubravka, and Mary Anne Kennan. "The Methodological Landscape." In *Research Methods: Information, Systems, and Contexts*, edited by Kirsty Williamson and Graeme Johanson, Second Edition, 127–55, Elsevier, 2018. https://doi.org/10.1016/B978-0-08-102220-7.00005-4.

Key Technical Terms

- The importance of **Boolean logic** when working with **conditionals**, **functions**, **variables**, and **control flow statements**
- **Computers**, **microcontrollers**, and other **digital** signals and their relationship to **binary notation**
- The relationship between **analog** and **digital** signals

Professional Journal Reflections:

1. Take a few minutes to review your Professional Journal Reflection(s) from last session. Reflect on ways this is a new reading of these reflective words and worlds one session later. How have these reflections shaped you? How have you shaped your reading of the world and word within these reflections?
2. In what ways have deductive ontological and epistemological reasoning and Boolean scientific logic contributed to your action reflection praxis, individually and within your community of practice, since your last reflection(s)?
3. In what ways have inductive reasoning, meta-design, and innovation-in-use, along with associated ontological and epistemological reasoning, contributed to your action reflection praxis, individually and within your community of practice, since your last reflection(s)?
4. In what ways have abductive reasoning and moral and ethical questioning individually and within your community of practice helped to unmask and expose hidden forms of oppression, false beliefs, and commonly held myths, thereby contributing to your action reflection praxis since your last reflection(s)?
5. Bringing these together, how has this influenced your hands-on coding to build a Toolbox Trumpet? How has the design and implementation of the sociotechnical artifacts used in these hands-on works shaped your own hands-on work?

2B: Make Music with Code

Background Knowledge Probe

Take a few minutes to bring up a mental picture of each musical instrument you've played yourself, or which you've watched others play.

- How are they played? What similarities and differences are there across instruments?
- How might they be constructed? What materials are used? How does material choice influence use of, and sound creation by, the instrument?
- Of the instruments considered, which ones might make use of digital technologies? Of programming languages?

So far, we've used the Circuit Playground Express microcontroller and MakeCode programming language to learn essential coding concepts.[1] In particular, we've explored:

- Conditional logic
- Iteration loops
- Variables
- Modularization

Technical Introduction

We've also started exploring how we can write programming code using these concepts to interact with analog and digital electronic circuits in order to develop working electronic tools.Now, let's turn our attention to the tools that might facilitate our telling of counterstories. We've seen how different folks are creating collective digital stories in community-based co-design projects using emergent approaches.[23] A Little Free Library is one of many examples in which analog information sharing

1. MakeCode © 2020 Microsoft and Adafruit. Screenshots used with permission.
2. Maria Rosa Lorini, Amalia Sabiescu, and Nemanja Memarovic, "Collective Digital Storytelling in Community-Based Co-Design Projects: An Emergent Approach," *The Journal of Community Informatics* 13, no. 1 (2017): 109–36. https://doi.org/10.15353/joci.v13i1.3296.

takes place to advance community building. Let's consider if the Circuit Playground Express microcontroller and Raspberry Pi microcomputer can work together to serve as a digital extension to an existing analog Little Free Library in a community. In what ways can digital networked information systems serve as tools to help advance the analog? In what ways might it disrupt the analog? In what ways is the disruption good? In what ways is it harmful? Rather than considering how the digital replaces the technical, perhaps a digital little free library can be a valuable tool — a digital networked information system extension — for the already existing analog networked information systems sharing information and knowledge in communities, with communities, and for communities. We will create a Little Free Library tool using the Circuit Playground Express and the Raspberry Pi for archiving and selective playback of stories. Down the line we will hopefully be able to use touch sensors to indicate which story is to be shared.But first, let's put the system through some early testing by turning the components into a Toolbox Trumpet!

Piano Keys and Trumpet Valves

Let's compare a piano to a trumpet. With a piano, you have a key for every note you want to play. For instance, if we want to play the basic C major **scale** in ascending order, we'd need seven keys as follows:

| C | D | E | F | G | A | B |

In music, an **octave** is a series of eight notes that take you from the first note back to that note, but in which that note is now twice or half the frequency of vibration that it was before.

To go down one octave, we'd need another set of seven keys to the left that cut the frequency of vibration in half. If we wanted to go up an octave, we'd need seven keys to the right that double the frequency of vibration. So to play three octaves of the C scale, we'd have something like:

| **C** | **D** | **E** | **F** | **G** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **A** | **B** |

Compare this to a trumpet. The standard trumpet has three valves. Each valve is a cylinder that, when pressed down fully, moves the airflow from one set of tubing to a different set of tubing. The airflow now takes a different path from the mouthpiece where it enters the trumpet to the bell where it exits the trumpet. In so doing, it changes the frequency of vibration slightly, changing the note being played.

To play a scale, you need to press one, two, or three of those valves, or none of the valves, depending on the note you are wanting to play. To make this work, you need to add in embouchure. A person playing the trumpet uses their lips and facial muscles to adjust the frequency of vibration up and

3. Angelika Strohmayer and Janis Meissner, "'We Had Tough Times, but We've Sort of Sewn Our Way through It': The Partnership Quilt," *XRDS: Crossroads, The ACM Magazine for Students* 24, no. 2 (December 19, 2017): 48–51. https://doi.org/10.1145/3155128.

down, thereby slowly moving up or down scales and octaves in combination with the movement of the valves.

In this case, we'd play a middle-C scale like so:

Note	Valve 1	Valve 2	Valve 3
C	open	open	open
D	pressed	open	pressed
E	pressed	pressed	open
F	pressed	open	open
G	open	open	open
A	pressed	pressed	open
B	open	pressed	open

Exercise: First Three Notes

Choose Your Own Adventure

In the introduction to the book, I shared about the tulip poplar, a wood which inspires me, and which I have used to craft a toolbox for my Raspberry Pi equipment. As visible in the video below, I've attached nylon conductive tape to the lid of the toolbox. The benefit of completing the exercise in this way is that it extends the digital capacitive touch sensors into an analog parallel. If you'd like to follow along in a similar manner, two resources are available to you:

- Purchase nylon conductive tape from Adafruit.
- Download and print the toolbox template.

But this is not the only way forward. The following exercises can be completed exclusively with the Circuit Playground Express. To pursue this remix, jump ahead to Option 2: Circuit Playground Express Trumpet.

Option 1: Toolbox Trumpet

Let's do some experimenting to find out how we might play notes on our Toolbox Trumpet by using three nylon electrically conductive tape strips on the cover of our toolboxes, and then using alligator clips to connect each to a pin on the Circuit Playground Express.

Before going through this and the following exercises, take 4 1/2 minutes to watch a video demonstration of the Toolbox Trumpet.

Connect three alligator clip wires between the nylon conductive tape pieces and the Circuit Playground Express:

1. Connect one end of an alligator clip wire to the nylon conductive tape by the **single | column**. Connect the other end of that alligator clip wire to pin **A1 on the Circuit Playground Express**.
2. Connect one end of a second alligator clip wire to the nylon conductive tape by the **double | column**. Connect the other end of this second alligator clip wire to **pin A2 on the Circuit Playground Express**.
3. Connect one end of a third alligator clip wire to the nylon conductive tape by the **quad | column**. Connect the other end of this third alligator clip wire to **pin A4 on the Circuit Playground Express**.

Next, we'll follow the steps to develop the code for playing a Middle C.

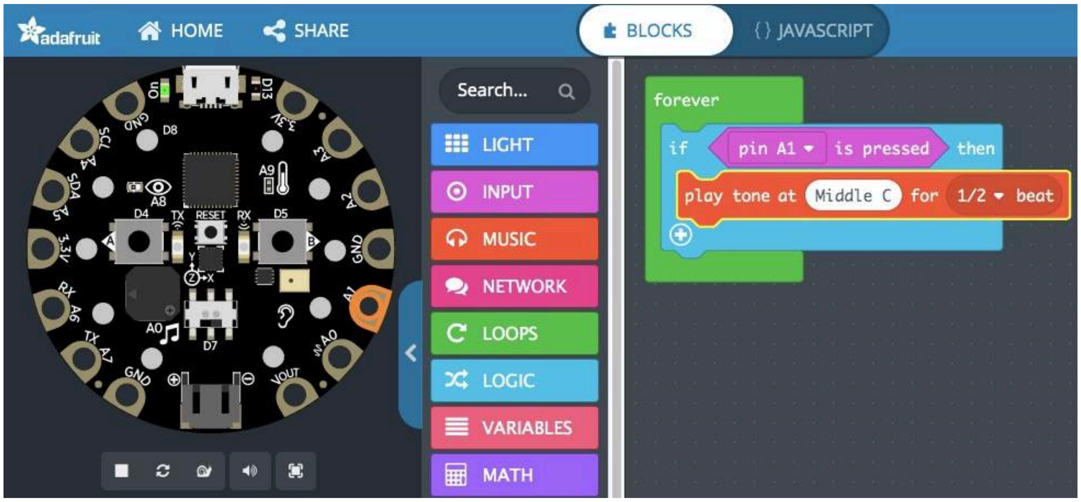
Option 2: Circuit Playground Express Trumpet

We can also make a trumpet with the capacitive touch sensors on the Circuit Playground Express. On the microcontroller, identify the sensors labeled A1, A2, and A4.

Steps

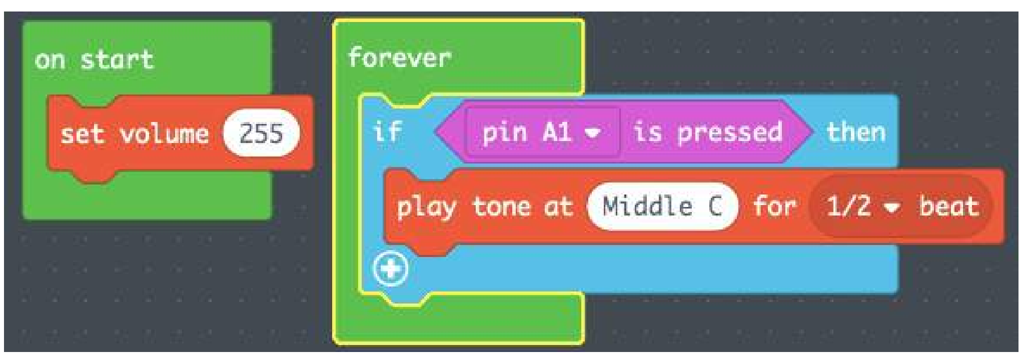
- First, let's develop the code to **play a Middle C**.
    1. Start a new project in MakeCode
    2. To a **FOREVER loop**, add in an **"If…Then" conditional** block from the "Conditionals" section of the LOGIC category.
    3. From the INPUT category, take a "button A is pressed" block statement, and drop it into the "If…Then" block statement where it currently says <True>.
    4. Hit the "button A ∇" window and change that to "pin A1." It should now say "if pin A1 is pressed, then."
    5. From the MUSIC category, take the "play tone at Middle C for 1/2 beat," and put it below the "if pin A1 is pressed, then" statement.
    6. In the simulator, move the mouse pointer over to the A1 pin and press down to play a Middle C for as long as you press it.
    We're playing our first note!
    7. Let's test it out on the real Circuit Playground Express:
        a. Save your project as `MiddleCTest1`.
        b. Flash this to your physical Circuit Playground Express.

c. Press A1 on the Circuit Playground Express.



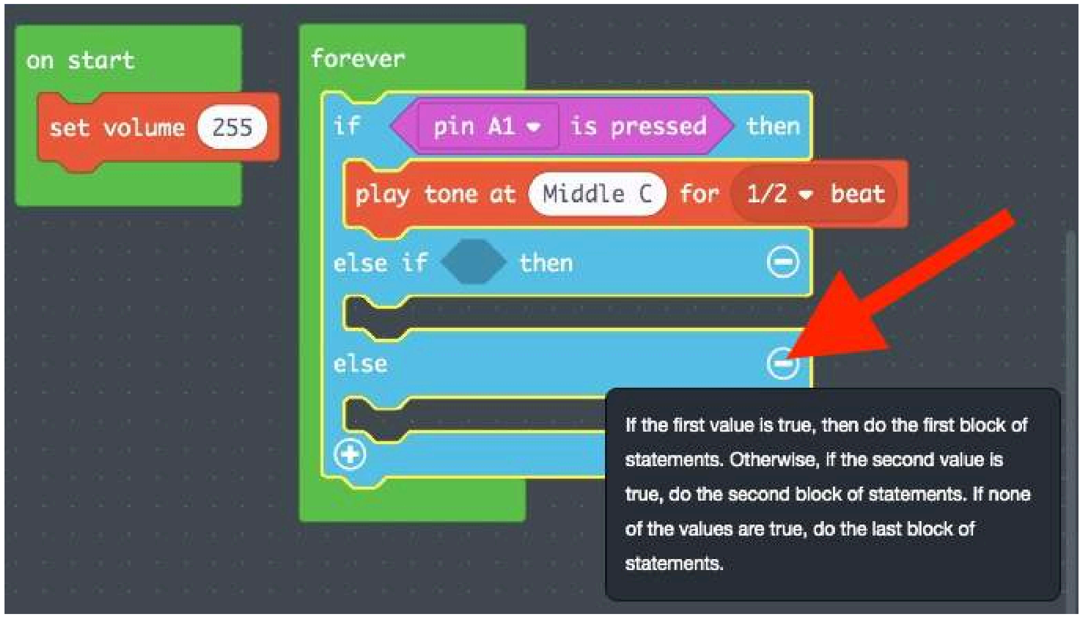
8. In this first test, the sound was a little soft. Let's crank up the volume a bit and test again.
    a. From the LOOPS category, pull out an "on start" loop.
    b. Add to this new loop a "set volume" from the MUSIC category.
    c. For testing, let's set that to 255, the largest number listed.
    d. Save this as `MiddleCTest2`.
    e. Flash this to your physical Circuit Playground Express.
    f. Press A1 on the Circuit Playground Express.

Feel encouraged to continuing adjusting this to meet your preferences.

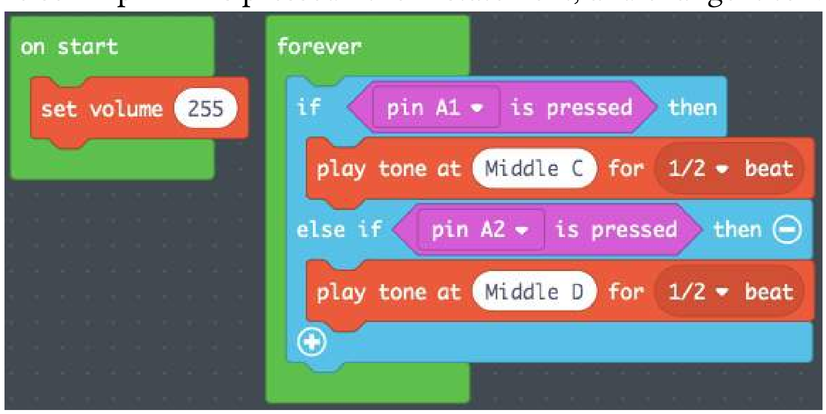


- Next, let's add the ability to **play a Middle D**. We'll do this in an else "If…Then" block statement right below the Middle C "If…Then" block statement.

1. Click on the + symbol at the lower left of the current "If…Then" block statement. This creates an else statement.
2. Click on the + symbol again to add an else "If…Then" block.
3. Hit the – symbol to the right of the else statement to remove it.



4. Duplicate the "pin A1 is pressed" block input statement and place that in the new else "If…Then" block statement. Change it to pin A2.
5. Duplicate the "play tone at Middle C for 1/2 beat" statement, place it under the "else if <pin A2 is pressed> then" statement, and change it to Middle D.



6. In the simulator, move the mouse pointer over to the A2 pin and press down to play a Middle D for as long as you press it.
We're ready to start our way up the C scale!
7. You might want to do a quick test on the physical Circuit Playground Express by saving the file and flashing it over. We're in a process of rapid design and prototyping, so for now, if it works, we can move on. But do be sure to take notes of what is working subpar based on your as self-assessment, and your understanding of preferences if this were to be made available to other users.

- With these two notes playing adequately, now what? We've only got three "valves" and we want to play seven different notes. Remember that on a real trumpet we could press 0, 1, 2, or 3 valves. We could also use embouchure to create more possible combinations. To mimic the trumpet, we are using only three buttons or “valves.” Is it possible that we can press these in combination to play a third note? Let’s test this out by pressing both pin A1 and pin A2.

1. Hit the + symbol at the bottom of the “If…Then…Else If…Then” conditional statement twice to add in a third “Else If…Then” block statement below the “pin A2 is pressed” conditional.
2. Delete the “Else” at the bottom of this extended “If…Then” control flow statement block of conditionals.
3. Go into the LOGIC category and scroll down to the “Boolean” section. From there, grab a “< > and < >” statement and add this to the third “If…Then” block statement.
4. Copy the “pin A1 is pressed,” and place that into the left “< >” space.
5. Copy the “pin A2 is pressed,” and place that into the right “< >” space.
6. Copy the “play tone at Middle D for 1/2 beat,” and drop it below the “else if <<pin A1 is pressed> and <pin A2 is pressed>> then” statement, and change it to Middle E. You should see something like this in the MakeCode programming screen.

7. Unless you have a touch screen, you cannot test this in the simulator. So let’s continue testing on our physical Circuit Playground Express.
    1. Let’s name this `ToolboxTrumpet3Notes` and click on the save icon.
    2. Flash this new downloaded file to your physical Circuit Playground Express.

Now we are ready to test our Toolbox Trumpet prototype.

Option 1: Toolbox Trumpet	Option 2: Circuit Playground Express Trumpet	Result
Press the **single | column** nylon conductive tape.	Press the **A1** CPE sensor.	You should now hear a Middle C.
Press the **double | column** nylon conductive tape.	Press the **A2** CPE sensor.	You should hear a Middle D.
Press both the **single | and double | column** nylon conductive tapes simultaneously.	Press the **A1 and A2** CPE sensors simultaneously.	You should hear that Middle E!

We've reached three notes! …Or did we?

Boolean Logic

In the code we just wrote, we created three "If…Then" block logic statements inside the forever **iteration** loop. Logical true and false, yes and no, 1 and 0 propositions are the fundamental process used in both electronics and modern programming languages, as well as in statistics and positivist paradigms of research.

Boolean logic, or **Boolean algebra**, uses two possible values, "true" and "false," to test logic statements. This concept was introduced by George Boole in his books *The Mathematical Analysis of Logic* (1847) and *An Investigation of the Laws of Thought* (1854). The Boolean logic written in our first code was as follows:

Is pin A1 pressed?	True →	Play Middle C for 1/2 beat →	End
False ↓			
Is pin A2 pressed?	True →	Play Middle D for 1/2 beat →	End
False ↓			
Is pin A1 and A2 pressed?	True →	Play Middle E for 1/2 beat →	End
False ↓			
End			

Note that the order of True/False statements in Boolean logic is essential. If pin A1 is pressed, then play Middle C. Else, if pin A2 is pressed, then play Middle D. Else, if both pin A1 and pin A2 are pressed, play Middle E.

This makes perfect sense if our ultimate goal is to move up the C scale, going from the Middle C on to the Middle D and then the Middle E.

The problem is, code uses Boolean logic, which has already determined that neither pin A1 nor pin A2 was pressed, so logically we already know both are also not pressed. That means we need to reverse our test in a way that fits within formal Boolean logic. In so doing, we should be in better shape.

Is pin A1 and A2 pressed?	True ⟶	Play Middle E for 1/2 beat ⟶	End
False ↓			
Is pin A2 pressed?	True ⟶	Play Middle D for 1/2 beat ⟶	End
False ↓			
Is pin A1 pressed?	True ⟶	Play Middle C for 1/2 beat ⟶	End
False ↓			
End			

For more on Boolean logic, take 10 minutes to watch the video "Boolean Logic & Logic Gates" from the Crash Course Computer Science series featuring Carrie Anne Philbin, hosted by PBS Digital Studios.

Exercise: Toolbox Trumpet, First Three Notes, Failing Forward

Open the MakeCode project named ToolboxTrumpet3Notes, and swap the "If…Then" block statements around using our Boolean logic diagram above. Your code should now look like this:

- Save and Download this edited version of the MakeCode, calling it `ToolboxTrumpet3Notes-FailForward`.
- Hit "reset" on the Circuit Playground Express microcontroller, and then drag and drop the `ToolboxTrumpet3Notes-FailForward.uf2` file into the CPLAYBOOT device.
- Press the toolbox's single | or CPE's A1 "valve" and you should now hear a Middle C.
- Press the toolbox's double | or CPE's A2 "valve" and you should hear a Middle D.
- Press the toolbox's single and double | or CPE's A1 and A2 "valves" together. You should hear that Middle E!

This time it should play the first three notes of a Middle C scale!

Key Takeaways

Failing forward to move from **Not Yet** to **Yet** is an unavoidable part of the creative process.

The human experience incorporates multiple ways of knowing and being. But for many of us, **Boolean logic** using true/false statements is not a part of our everyday practice.

While in the last session we worked to expand our essential coding concepts like **conditionals**, **functions**, **variables**, and **control flow statements**, it is only with the incorporation of Boolean logic that we gain some of the central deductive reasoning skills needed to effectively put our designs into executable code.

Boolean Logic and Algebraic Notation

Reflecting back on our Toolbox Trumpet code, let's step aside for a minute to consider ways we could get three "valves" to play seven different notes.

We could have set up that third else "If…Then" block statement to play Middle E with the third valve alone. What might have been the design choice to instead play it using the first two valves combined?

What would be the next valve or valve combinations of to play the next note in the C scale, a Middle F?

Base Ten Decimal Notation

Count with me from zero to nine using decimal notation:

Zero			0
One			1
Two			2
Three			3
Four			4
Five			5
Six			6
Seven			7
Eight			8
Nine			9

Excellent!

Now, what would we do if we wanted to move to the number ten?

If you kept a close eye, you'll notice the table above had a couple of blank columns to the left of the column listing numbers 0 to 9. Let's use that as a means to count from zero to twenty.

Zero		0	0
One		0	1
Two		0	2
Three		0	3
Four		0	4
Five		0	5
Six		0	6
Seven		0	7
Eight		0	8
Nine		0	9
Ten		1	0
Eleven		1	1
Twelve		1	2
Thirteen		1	3
Fourteen		1	4
Fifteen		1	5
Sixteen		1	6

Seventeen		1	7
Eighteen		1	8
Nineteen		1	9
Twenty		2	0

Note that now we've had a "tens" column all along. The value was just at zero when we started. Once we moved over to ten, we reset the "ones" column to zero, and moved the "tens" column to one.

The "tens" column remained at 1 while we moved the "ones" column from 1 to 9. To take the next step, we moved the "ones" column back to 0, and the "tens" column from 1 to 2, giving us twenty.

Using just these two columns, we could make our way up to ninety-nine. Now, let's move from ninety-nine to one hundred. To do that, we need to use the third column, that is, the "one hundreds" column:

Zero	0	0	0
…			
Nine	0	0	9
Ten	0	1	0
…			
Ninety-nine	0	9	9
One Hundred	1	0	0
One Hundred and One	1	0	1
…			
One Hundred and Ninety-nine	1	9	9
Two Hundred	2	0	0
Two Hundred and One	2	0	1
…			
Two Hundred and Ninety-nine	2	9	9
Three Hundred	3	0	0
…			
Nine Hundred and Ninety-nine	9	9	9

To go any further in base ten, we'd need a fourth column, that is, the "one thousands" column, which we don't have in the table we've been using for this. We're at our limit.

When Base Ten Decimal Notation Isn't the Best Option

So if we somehow could do a lot of real quick tapping on those valves, we could be moving quite lively through a whole bunch of different notes. Our voices can move steadily through notes by adjusting the frequency with which they vibrate. The problem is, when we use the valve of a trumpet or our prototyped equivalent, it serves as a means to accomplish Boolean logic.

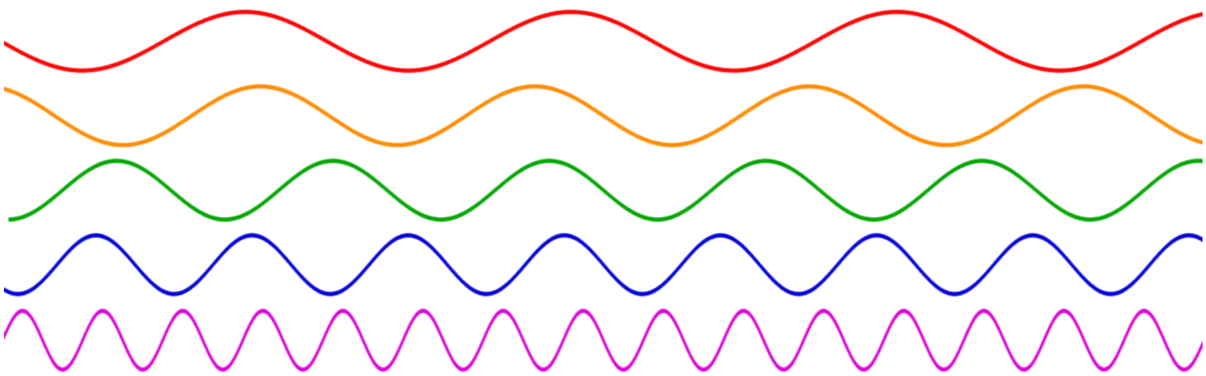
- If the trumpet valve is pressed, move the airflow through this tube. Else, pass it through that tube.
- If our prototyped valve is pressed, that is, if pin A2 is pressed, play a Middle D.

As a matter of fact, in MakeCode, when you click to change the note you were wanting to play if pin A2 is pressed, clicking on the Middle D of the visual piano also shows the number 294, or 294 Hertz. **Hertz** is the frequency, or number of cycles per second that something should cycle up and down. The slower it cycles, the lower the pitch. The faster it cycles, the higher the pitch.

Middle C is actually specifying that the little speaker on the Circuit Playground Express cycle 262 times per second, or 262 Hz, while Middle D cycles at 294 times per second, or 294 Hz. By cycling a little faster, the pitch is a little higher.

With a standard speaker, that means a membrane vibrates up and down 262 times per second or 294 times per second, depending on if we're playing a Middle C, or a Middle D, respectively. This is what makes an analog sound. And it's the same thing that human vocal cords do.

> **Digital signals** collect and distribute data using the digits 0 and 1 to serve as samples of the analog signals or information. Pins A1, A2, A3, A4, A5, A6, and A7 on the Circuit Playground Express can serve multiple roles, one of which is as **capacitive touch sensors**. When performing these readings of skin or other materials that have a slight positive charge, the input level is interpreted as a binary value of 1 and a logic value of True. When no positively charged touch is sensed, the input level is low, and a binary value of 0 and logic value of False are used.
>
> In MakeCode, the word "pressed" in the statement "if pin A1 is pressed, then" is equivalent to the binary value 1 and the logic value True.
>
> **Analog signals** are made up of continuously variable physical quantities. The speaker built into the Circuit Playground Express delivers an analog signal. The frequency of oscillation from peak to peak in a sine curve as shown below determines the musical pitch we hear. The note heard in a speaker vibrating two full oscillations, as shown in red in the image below, is 1/7th the pitch of the note played in the speaker when there are 14 full oscillations as shown in purple in the image below.

> If the sine wave below showing 14 full oscillations represents 262 Hertz, that is, a frequency of 262 oscillations per second, then the one above, showing 7 full oscillations in the same time period represents 131 Hertz. As seen in the MakeCode "Play tone at" line, 262 Hertz is a Middle C, while 131 Hertz is a Low C. The human ear can hear sounds as low as 20 Hertz, and in the image below 2 full oscillations in this time period is equivalent to about 37 Hertz given the other two frequency calculations, although it may be that the Circuit Playground Express will not play a frequency that low.
>
> 
>
> *Image by LucasVB, Wikimedia Commons, dedicated to the public domain.*
>
> For more on sound and music projects with the Circuit Playground Express, check out: Adafruit's "Circuit Playground Sound and Music" tutorial.

We could work to press our "valve" 262 times a second to play a Middle C over the next second in time. Or we could give ourselves some liberty and adjust the clocked **sequential logic** such that if we tapped it 26 times in 10 seconds, we would play a Middle C scale over the next second of time.

Probably a more practical way forward is to use the conductive touch sensor pins on the Circuit Playground Express. We can take the digital inputs on these sensors and use basic Boolean logic to identify which of these sensors is touched during the current clock cycle. Then we can calculate a mathematical conversion to a certain frequency, say 262 Hz, to provide an analog Middle C output via the Circuit Playground Express speaker.

Building from here, we could design and construct a toolbox piano made from tulip poplar, much like in the first video of this session. But that would require 21 nylon conductive tape pads, placed close enough together to let us easily switch between them, but far enough apart that we don't sense a current flow between them, confusing the code. Current can conduct over wood, especially when moist from rain or from just the wear of our moist human hands on the wood and nylon. Doable? Yes. Practical? Perhaps not.

Is there another design option?

Base Two Binary Notation

It's exactly this conundrum that moved early developers of electronics and the early computers from base ten decimal notation to base two **binary notation**. Binary works in zeros and ones only. That is, they are a standard means of implementing Boolean logic!

Is pin A1 and A2 pressed?	True →	Play Middle E for 1/2 beat
False ↓		
Is pin A2 pressed?	True →	Play Middle D for 1/2 beat
False ↓		
Is pin A1 pressed?	True →	Play Middle C for 1/2 beat

Or let's revisit the electronics studied earlier in the book: The push button itself is a Boolean logic processor.

Is the momentary switch push button pressed?	True →	Pass current through; LEDs are lit
False ↓		
Do not pass current through; LEDs are not lit		

As mentioned earlier, Boolean logic is fundamental to electronics, not just computer science programming languages. Indeed, the world around us has many ways of moving from true/false, yes/no, one/zero. This is why Boolean logic quickly shifted the design of electronics and programming over to using base two binary notation.

When we count in binary, we move from columns representing ones, tens, and hundreds of base ten decimal notation, to columns representing ones, twos, and fours. Each column is known as a **bit**. Let's see where that takes us, counting-wise, when each bit can only be a 0 or a 1:

Number	**Fours bit**	**Twos bit**	**Ones bit**	**Decimal Equivalent**
Zero	0	0	0	= 000
One	0	0	1	= 001
Two	0	1	0	= 002
Three	0	1	1	= 003
Four	1	0	0	= 004
Five	1	0	1	= 005
Six	1	1	0	= 006
Seven	1	1	1	= 007

Note, then, that in binary we now have a "ones" column. It can only move from zero to one. So to get to the number two, we need a "twos" column. Moving it from 0 to 1 and moving the ones column from 1 to 0 gives us the number two. To move to three, we then keep the twos column at 1, and move the ones column back to 1.

But we're at the end of what those two columns can reach. So we now need a "fours" column. Set the ones and twos columns back to 0, and move the fours column to 1. Ta-da! We've reached four. By switching the ones and twos column on and off, we can move the count all the way up to seven!

This demonstrates that a 3-bit counter can move between 8 different decimal numbers: 0, 1, 2, 3, 4, 5, 6, and 7. It also demonstrates that we haven't even completed filling in the ones column in decimal, let alone moving to the tens or one hundreds columns of decimal notation.

For more on binary, take 10 minutes to watch the video "Representing Numbers and Letters with Binary" from Carrie Anne Philbin's *Crash Course Computer Science* series hosted by PBS Digital Studios.

Exercise: Creating Code to Play a Full Middle C Scale

Let's put this new Boolean logic and deductive reasoning into practice to play the full Middle C scale using the three "valves" of our Toolbox Trumpet or Circuit Playground Express. As this is a tool being designed first and foremost for educational purposes, let's add in pin A4 along with pins A1 and A2.

Remember that order matters in Boolean logic:

Is pin A1 and A2 and A4 pressed?	True ⟶	Play Middle B for 1/2 beat ⟶	End
False ↓			
Is pin A2 and pin A4 pressed?	True ⟶	Play Middle A for 1/2 beat ⟶	End
False ↓			
Is pin A1 and pin A4 pressed?	True ⟶	Play Middle G for 1/2 beat ⟶	End
False ↓			
Is pin A1 and pin A2 pressed?	True ⟶	Play Middle E for 1/2 beat ⟶	End
False ↓			
Is pin A4 pressed?	True ⟶	Play Middle F for 1/2 beat ⟶	End
False ↓			
Is pin A2 pressed?	True ⟶	Play Middle D for 1/2 beat ⟶	End
False ↓			
Is pin A1 pressed?	True ⟶	Play Middle C for 1/2 beat ⟶	End
False ↓			
End			

1. Add in five more else “If…Then” block statements below the three we already have. Proceed to shuffle these so that the order goes as follows:

2. Save this project as `TrumpetToolboxCScale`, flash it to the Circuit Playground Express, and give it a test run.
3. Can you play all seven notes of a Middle C scale? Keep practicing until you can do this at a relatively good pace.

Getting in Your Groove

Feel encouraged to go into the MakeCode project and adjust the volume level that we configured in the "on start" loop. You may also want to consider adjusting the tempo at which the 1/2 beat is played. You can do this by pulling out a "set tempo to 120 (bpm)" from the "Tempo" section of the MUSIC category and adding it to the "on start" loop. Play the 1/2 beat slower by lowering the beats per minute to double or single digits. Or speed it up by raising the tempo to setting it somewhere over 200 bpm. Experiment with changing the beat from 1/2 in each of the if logic statements to something else of your choice, say down to 1/16th or up to 4. Test it out in the simulator with the Middle C, Middle D, and Middle F notes. Or test it out on the physical Circuit Playground Express by saving, flashing, and testing to hit your preferred Toolbox Trumpet performance metrics!

Key Takeaways

Computing devices like computers and microcontrollers use **binary notation** to represent numbers and letters. The fundamental Boolean logic used in electronics is founded on base two binary notation. **Digital** signals like the capacitive touch sensors of the Circuit Playground Express also use binary.

The continuously variable physical quantities of the world, including the human senses, are **analog**.

Our Toolbox Trumpet now uses digital sensors on the Circuit Playground Express as conditional inputs to control the Hertz of the analog audio output of the Circuit Playground Express speaker.

The prevalence of the digital realm, and the notion that more recent generations are born **digitally native**, does not change the core of our physical being. But it does add additional tools in our toolbox for use in our everyday lives, some of which are truly valued in helping us reach our individual and community goals. We're still **physically analog**. Or are we?

Has the creation and use of this digital trumpet shaped us in significant ways? How might we have been changed differently if this session had been conducted only using analog materials, for instance, using different lengths of wood to play different notes?

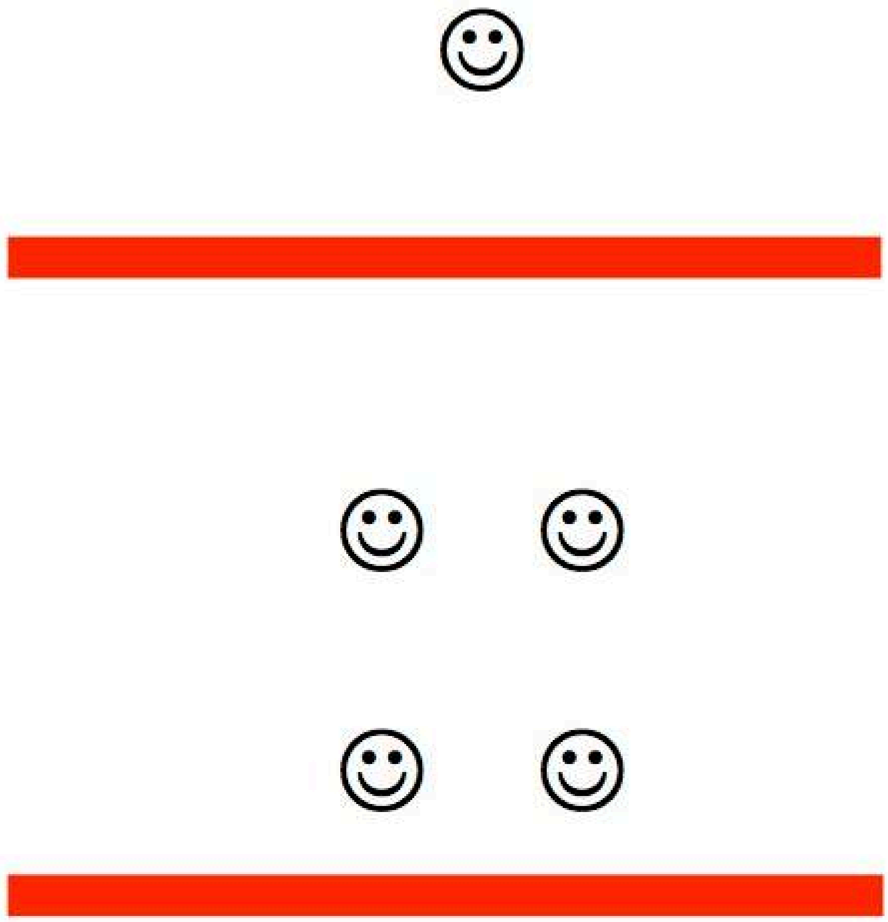
Wrap Up

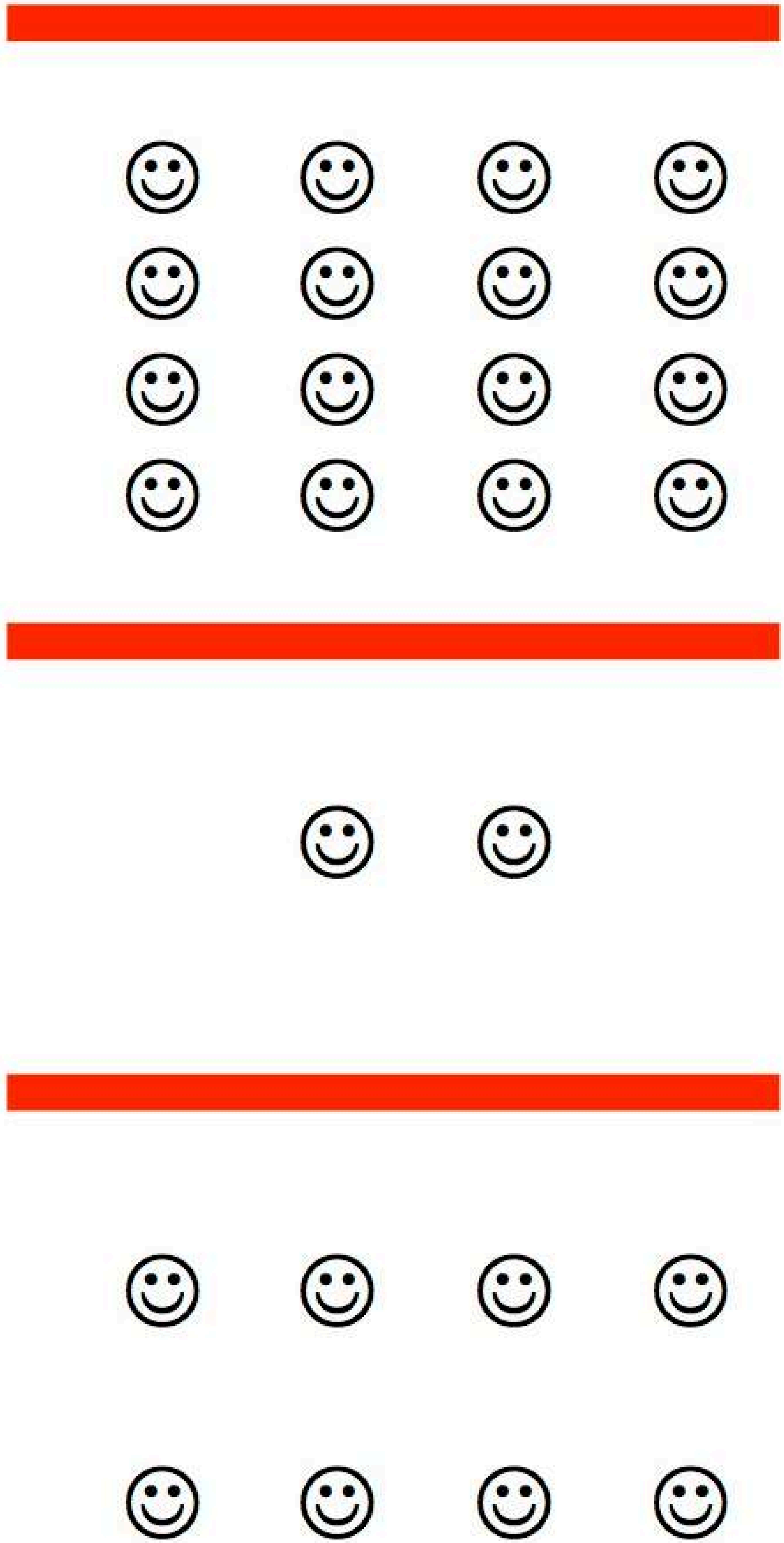
The focus of this session was two-fold: First, to help us expand our understanding of the methodological landscape. This includes positivist paradigms, as related to programming code and Boolean logic, with interpretive paradigms, as related to meta-design and innovation-in-use, as well as critical paradigms that advance moral and ethical questioning to unmask and expose hidden forms of oppression. Second, to further our more holistic journey of recognition regarding the deep interconnectedness of the social and the technical, one that cannot be fully teased apart. At the same time, it is hoped this journey is now better facilitating our use of deductive reasoning to more effectively create programming code even as we work to use inductive and abductive reasoning to reflect on our sociotechnical artifacts as we continue to move from a thing-orientation to a person-orientation.

Comprehension Check

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with answers, see the online version.]

Binary Notation





Print out the above binary cards and use scissors to cut these at the red lines. Then use these as a binary counter, with the smiley faces up equaling a binary notation 1, and smiley faces down equaling a binary notation 0. Order them from right to left as shown below:

☺☺☺☺ ☺☺☺☺ ☺☺☺☺ ☺☺☺☺	☺☺☺☺ ☺☺☺☺	☺☺ ☺☺	☺☺	☺	**Decimal Notation**
16s Bit	8s Bit	4s Bit	2s Bit	1s Bit	=
?	?	?	?	?	000
0	0	0	0	1	001
0	0	0	1	0	002
0	0	0	1	1	003
0	0	1	0	0	???
0	1	0	0	0	???
1	0	0	0	0	???
1	0	1	0	0	???
0	1	1	1	1	???
1	0	1	1	0	???
?	?	?	?	?	023
?	?	?	?	?	024
?	?	?	?	?	025
?	?	?	?	?	026
?	?	?	?	?	027
?	?	?	?	?	028
?	?	?	?	?	029
?	?	?	?	?	030
?	?	?	?	?	031
?	?	?	?	?	032
?	?	?	?	?	127
?	?	?	?	?	255

The Instructor Binary Cards may be helpful at the end of the above. By using these, you could count in binary notation from decimal zero to decimal 255, inclusive.

[An interactive element has been excluded from this location for the PDF edition of the text. Complete the activity “Core Coding Concepts” online.]

Use the book chapters and linked audio and video sources, and Blue Unit End Notes if you’re not quite sure. This is a great opportunity to come together in small groups to work on this learning check.

3A: Valued, Inclusive Information and Computing Technology Experiences

Background Knowledge Probe

What do we especially value? How do we support others, and how are we supported by others, in achieving that which is valued?

Resource	Functioning (something people value being or doing)	Capability (the freedom to achieve the functioning)	Utility (that sought to be maximized)
bicycle	mobility	ability to move around	pleasure

- Looking at the above table, how do "bicycle," "mobility," "ability to move around," and "pleasure" relate to you personally?
- How do these terms or statements relate to other people you know and who come to mind at the moment?
- Why might these terms or statements be listed within the specified columns?

Examining "Development"

The information, communication, and computing technologies we use in our everyday lives — from home to play to profession — arise from complex histories, methodologies, cultures, knowledge, and philosophies of a just society. They strongly shape our daily experiences and well-being. Strong arguments are made regarding the societal benefits of information, communication, and computer sciences: from meeting individual needs by empowering people, to advancing essential social services such as healthcare and education, to facilitating urgent data collection in advance of or during natural disasters and human-induced climate change, and beyond. Yet the sciences, including computer science, shape and are shaped by sociotechnical artifacts.

It is important to consider an oft-used word embedded within many conversations regarding the societal benefits of information systems: **development**. This term is used to indicate material prosperity, liberation from oppression, a holistic project for personal progress, development of a child, of new software, of new countries, of natural resources, and still more. It is a term used both directly, and as a hidden assumption. But however it is used, it is not a neutral term; "development" is loaded with value judgments, trade-offs, and complexities.

Some scholars within the Community Informatics community argue for the adoption of Nobel Laureate Amartya Sen's capability approach to facilitate "human, rather than technical or sociotechnical, problem solving."[1] Let's explore community informatics and computational thinking as a means to advance agency for valued, inclusive information, communication, and computing technology experiences.

The Capability Approach

The capability approach is an alternative to normative approaches to international development policies and practices that focus exclusively on resource expansionism such as economic growth, or positive/predictive approaches for strategies and practices that emphasize empirical studies, data analysis, and hypothesis testing. The capability approach has been highly influential in guiding the United Nations to a more holistic human development paradigm with its four essential pillars: equality, sustainability, productivity, and empowerment.

Central to the capability approach are those things individuals value being and doing, their **functionings**. While it is important we do all we can to identify a person's functionings, the focus for development **stakeholders** is on expanding the set of **capabilities**, that is, the freedoms for individuals to achieve their functionings. Capabilities are the real and actual possibilities from which individuals can choose so as to achieve those things they value being and value doing. Functionings are the aspirations that make up people's well-being.

Martha Nussbaum has created a list of 10 central human capabilities as follows: Life; Bodily health; Bodily integrity; Senses, imagination, and thought; Emotions; Practical reason; Affiliation; Other species; Play; and Control over one's environment.[2] However, Sen suggests each individual should be free to determine if any, all, more, or different capabilities are central to their personal functionings. While valuable food for thought as stakeholders gather to co-explore development paths, the essential determinant is to consider the possible range of capabilities needed to provide a sufficient capability set advancing individual freedom to choose, execute, and achieve their valued goals.

1. Amartya Sen, *Development as Freedom (*New York: Alfred A. Knopf, 1999). See also: Larry Stillman and Tom Denison, "The Capability Approach Community Informatics," *The Information Society* 30, no. 3 (May 27, 2014): 209. https://doi.org/10.1080/01972243.2014.896687.
2. Martha Nussbaum, *Women and Human Development: The Capabilities Approach* (Cambridge: Cambridge University Press, 2000), 78-80.

To expand the capability set, we also need to assure individuals have **agency** to pursue the goals they value. Agency is a combination of many individual resources from within a resource portfolio that includes: Material resources; Financial resources; Natural resources; Geographical resources; Human resources; Psychological resources; Information; Cultural resources; and Social resources.[3] Sen identifies **choice**, then, as the aim and the means for human development, recognizing that individual choice ultimately depends on their personal valued beings and doings. In the article "The capability approach and the 'medium of choice': steps towards conceptualizing information and communication technologies for development," Dorothea Kleine notes that choice has four key components: existence of choice, sense of choice, exercise of choice, and achievement of choice. In many cases, Kleine has found the existence of choice in her works to advance an individual's ability to exercise and achieve choice, but that this existence is hidden from the individual for a range of reasons. Advancing a person's capability set, then, is more than that of agency and skills development for the exercising and achieving of choice, but also a broader work to advance the sense of choice.

Resource	Functioning (something people value being or doing)	Capability (the freedom to achieve the functioning)	Utility (that sought to be maximized)
bicycle	mobility	ability to move around	pleasure

If we review the table, we can see that the person aspires to be mobile. A bicycle can potentially then serve to provide agency to the person as a component within their capability set. However, as some of us may have considered up front, sometimes a bicycle is not the best resource, such as for those who do not have the technical abilities to ride a bicycle. This may be based on road or environmental conditions, because of balance problems, because of conflicts with the structure of the body, or still more reasons. Or it may be simply because it is not a capability a person values doing, even though mobility is a valued being.

It is from this framing of the capability approach that Mario Toboso challenges us to rethink disability.[4] **Disability** is the accepted term used when considering treatment of those in society with physical or mental limitations outside the considered norms of a human. From within the "medical model," it is understood that disabilities are medical in nature, that disability is a disease, and that a person can only contribute to society to the extent they can be rehabilitated or normalized. The "social model," on the other hand, identifies disabilities as social in nature. Each individual is just as capable as anyone else of making contributions to society if an environment is designed taking specific requirements into account. More recently, a "diversity model" has been brought forward to overthrow old concepts of ability and instead looks for personal identities that are not perceived as negative.

3. Dorothea Kleine, "The Capability Approach and the 'Medium of Choice': Steps towards Conceptualising Information and Communication Technologies for Development," *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 119–30. https://doi.org/10.1007/s10676-010-9251-5.
4. Mario Toboso, "Rethinking Disability in Amartya Sen's Approach: ICT and Equality of Opportunity," *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 107–18. https://doi.org/10.1007/s10676-010-9254-2.

It is within this new diversity model that Toboso articulates "functional diversity" in association with Sen's concept of functionings, the valued beings and doings a person aspires to achieve to bring about well-being. In expanding the diversity model to incorporate "diversity of functionings," we can consider mobility with a new lens. Agency to move around will require many different modes in acknowledgment of the diversity of people, and the ends they are seeking to achieve through the mobility. Participating in an information systems project, moving locations under challenging environmental conditions, or playing with others in dance or sports each would require different means of executing and achieving choice regarding ability to move around, given their agency to do so.

As community informatics seeks to advance the well-being of individuals and communities through effective use of information and communications technologies, the capability approach serves as a framework for participatory action and research within a community of inquiry to assure a respected person-orientation to achieve real freedoms of agency to achieve that which individuals and collectives value being and doing. It is a valuable complement to critical methodological paradigms, and especially those within information and communications technology, that provide situated agency as a means to provide a critical account of individual agency, that move beyond a simplistic notion of technology as goods and resources often seen within capability approach research and practice, a work that advances reflexivity and a critical lens so as to identify and address reification and hegemonic potential of technology science and research methods and practices.[5] This facilitates better design and development of digital technologies through an understanding and respect of the unique and rich community cultural wealth. It is to actively work to counter oppressive components of sociotechnical artifacts, something only achievable through decodification and recodification of the text and context of the technologies, so as to see the ways in which they are removing choice, the sense of choice, and the possibilities to execute and achieve choice.

Inclusive Computational Thinking

Computational thinking brings together 21st century skills of communication, collaboration, creativity, and critical thinking with digital technologies to advance collective problem-solving. Drawing on community informatics, the capability approach, functional diversity, and inclusive computational thinking, it is essential to develop inclusive computer science experiences for learners with disabilities. Doing so advances work within the sciences, technology, engineering, arts, mathematics, computer science itself, and beyond.[6]

[The Creative Technology Research Lab at the University of Florida] has worked extensively to explore advancement of K-12 education based on frameworks and strategies valued within a broad range of information and communication technology spaces. Universal Design for Learning (UDL) is one

5. Yingqin Zheng and Bernd Carsten Stahl, "Technology, Capabilities and Critical Perspectives: What Can Critical Theory Contribute to Sen's Capability Approach?" *Ethics and Information Technology*, 77.
6. For more information, see: Creative Technology Research Lab, "Project TACTIC," accessed August 3, 2023, https://ctrl.education.ufl.edu/projects/tactic/.

framework for innovative teaching and learning that combines explicit instruction with open inquiry and collaborative problem-solving to foster co-construction of expertise with peers. When combined with the capability approach, functional diversity, and community informatics, the Universal Design for Learning framework helps build bridges across the full diversity of stakeholders to better align design of information and communication technologies in ways that further facilitate innovation-in-use to achieve functionings. The three principles of UDL are **engagement** that stimulates interest and motivation for affective learning networks, **representation** that presents information and content in different ways building recognition learning networks, and **action and expression** that differentiate the ways co-explorers can express knowledge to build strategic learning networks.[7] These principles serve as a solid base for effective, inclusive community inquiry.

> You know learning, it's not a spectator sport.
>
> Ava Wolf, eLearning Professional, Center for Innovation in Teaching and Learning

Advancing our well-being through use of digital and non-digital technologies to realize our functionings requires active co-explorations within communities of inquiry. Explorations may occur in a second grade classroom, a school library, a university studio class, a public or academic library, a salon or barbershop, a community center, a community garden, a civic center, a religious institution, a safe space for people who are homeless, or the many other gathering spaces of communities. Inquiry happens when co-exploration is driven by community problems, in which questions are raised not only of the immediate possible technology under consideration, but also the nature of the community, membership within it, competing values, and beyond, so as to challenge and advance the nature of community itself. This, as much as anything else, is what it means to take a person-centered approach. It is the transactional model of communication central to community inquiry as members of a community and those working in partnership's with and for the community which can be "integrated into our social realities in such a way that it helps us not only understand them but also create and change them[8]. It is the "radical revolution of values" the Rev. Dr. Martin Luther King, Jr. calls us to[9]. It is the problem-posing as opposed to banking system of education Paulo Freire calls us to as oppressed and oppressor come together in action-reflection praxis[10].

To achieve this, inquiry must be facilitated by the community itself. Innovation-in-use, as with all co-exploration learning processes, is a doing accomplished in community, with community, and for community. Inclusive computational thinking, then, becomes inclusive community tinkering with an

7. CAST, "About Universal Design for Learning," accessed August 3, 2023, https://www.cast.org/impact/universal-design-for-learning-udl.
8. pg. 9, Bruce, Bertram C. *Beyond the Classroom Walls: Imagining the Future of Education, from Community Schools to Communiversities.* Rowman & Littlefield, 2022
9. pg. 157, King, Martin Luther. "Beyond Vietnam." In: Carson, Clayborne and Shepard, Kris, eds. A Call to Conscience, 139-164. Grand Central Publishing, 2001.
10. see especially chapter 2 in Freire, Paulo. Pedagogy of the Oppressed. 50th anniversary edition. New York: Bloomsbury Academic, 2018.

emphasis on persons first and last, and in which technical teaching, learning, hypothesizing, and testing occur as infill. And even more importantly, as we noted in the Blue Unit chapter 1A: The Logic of Hardware and Programming this work is a process of reading of word and world so as to understand the codifications of the central tenets of the technology, to decodify these words, to discover the mutual shaping of those technologies, and a recodification of these to strategically advance individual and community valued beings and doings.

Through its integration of social and technical components, generative themes, and shared strategies, frameworks, and paradigms, in what ways might this book be advancing a life-long pattern for community inquiry and community informatics practice? In what ways is it shaped to hinder such advancement?

Lesson Plan

Together, we've worked to use a person- and social-first focus to reconsider our everyday technologies as both positive and oppressive, disruptive technologies. We have worked to advance our skill sets to serve as facilitators and co-explorers in person-centered demystification and innovation-in-use of our technologies. How can we assure a fully inclusive approach to these works?

As we now move to "Build Functions for Remixable Code," we add functions into the Toolbox Trumpet to create a four octave scale before concluding with an LED pixel counter challenge. What ways, if any, could this be used as a test space for assuring fully inclusive approaches?

Essential Resources:

- Toboso, Mario. "Rethinking Disability in Amartya Sen's Approach: ICT and Equality of Opportunity." *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 107–18. https://doi.org/10.1007/s10676-010-9254-2.
- Zheng, Yingqin, and Bernd Carsten Stahl. "Technology, Capabilities and Critical Perspectives: What Can Critical Theory Contribute to Sen's Capability Approach?" *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 69–80. https://doi.org/10.1007/s10676-011-9264-8.
- Kleine, Dorothea. "The Capability Approach and the 'Medium of Choice': Steps towards Conceptualising Information and Communication Technologies for Development." *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 119–30. https://doi.org/10.1007/s10676-010-9251-5.
- Creative Technology Research Lab | College of Education. "Project TACTIC." Accessed February 16, 2020. https://web.archive.org/web/20200808035227/https://ctrl.education.illinois.edu/.

Additional Resources:

- Robeyns, Ingrid. "The Capability Approach." In The Stanford Encyclopedia of Philosophy, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2016. https://plato.stanford.edu/archives/win2016/entries/capability-approach/.
- CAST. "About Universal Design for Learning." Accessed February 16, 2020. https://www.cast.org/impact/universal-design-for-learning-udl.

Key Technical Terms

- The value of moving certain **control flow statements** and other code into their own **function**
- The movement of data between functions

Professional Journal Reflections:

Use this reflection to bring our activities to date into conversation with the capability approach and inclusive computational thinking frameworks brought forward in this session.

1. In what ways might the toolbox trumpet and Raspberry Pi counterstories serve as a resource advancing agency? A capability?
2. In what ways might the toolbox trumpet and Raspberry Pi counterstories serve as an information system problematically disrupting valued beings and doings of individuals and communities?
3. In what ways might the toolbox trumpet and Raspberry Pi counterstories be designed for designers, that is, for the innovators-in-use who adapt this to their own functionings? In what ways might it have been designed differently leading to this point?

3B: Build Functions for Remixable Code

Background Knowledge Probe

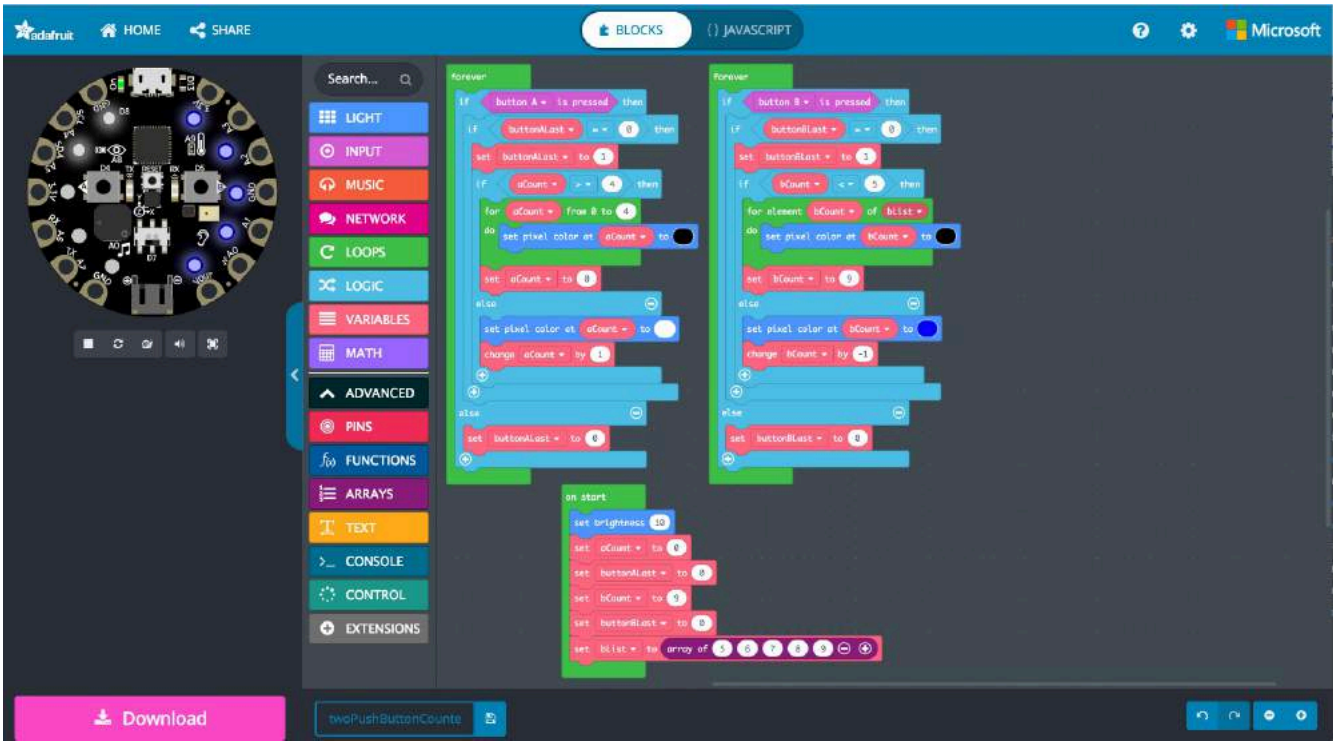
Reusing/remixing code is a central tenet of the free and open source software movement. And indeed, it's a central tenet of the Internet, as we'll discover in the Rainbow Unit. But as we've discovered throughout the book and especially highlighted in The Unknown Tech Innovators, there's also an active work to give the credit for innovations to certain people, and particularly to white males.

Take a few minutes now to reflect back on the innovative designers and builders of the code we're now using for our own Community of Practice works, and especially those hidden and forgotten that we've worked to bring forward as part of our explorations.

- How has their work helped shape your own use and reuse/remixing works?
- In what ways might your remixes be used by others in their works?

Technical Review

Let's take a minute to reflect back on the final exercises in sessions 1 and 2. Note that in the "Five Digit Counters" exercise, we used a range of **variables** and **arrays** to expand our opportunities for collecting and sharing data using several **functions**.



Compare this to the "Middle C Scale Toolbox Trumpet" exercise. Here, we committed our attention to deductive reasoning and **Boolean logic** to guide us in the creation of an appropriate, functional sequence of **conditionals**, thereby playing the notes in order from Middle C to Middle B.

Exercise: Variables & Functions Remix of the Toolbox Trumpet

Let's begin by considering how we could remix the Toolbox Trumpet using variables and functions to expand the capabilities of this tool.

Steps: Creating a binary to decimal calculator

To begin, we will create a new variable and use conditionals to collect the **binary notation** values, 000 to 111, provided using the three "valves." We will use this to calculate the decimal notation equivalent, 0, 1, 2, 3, 4, 5, 6, or 7.[1]

1. In the VARIABLES category, create a new variable called `CapTouchCounter`. This will be used to determine which capacitive touch sensor "valve" or "valves" are pressed in that sequence.
2. Drag a "set CapTouchCounter to 0" block from the VARIABLES category and place it in the default FOREVER function.
3. Add three "If…Then" blocks from the LOGIC category to the FOREVER function, and configure them using a "button A is pressed" block from the INPUT category to have conditionals for pin A1, pin A2, and pin A3.
4. In each of the "If…Then" conditionals, add a "change CapTouchCounter by" block. Set the pin A1 conditional so that CapTouchCounter is changed by 1. Set the pin A2 conditional so that CapTouchCounter is changed by 2. Set the pin A3 conditional so that CapTouch-Counter is changed by 4.
5. Under advanced, go down to the CONSOLE category. From there, drag a "console log" block and add it as the last line of code in the new FOREVER function.
6. Finally, from the VARIABLES category, drag a "CapTouchCounter" block and drop it into the right oval window of the "console log" block.

[Watch the video "2x Playback of Capacitive Touch Sensor 3 bit Counter" for a recording of these steps to compare with your own process.]

Feel encouraged to change 1X to 2X to play it back at high speed. Note that the console provides a means to enter a new level of testing within the simulator. Remember that we cannot test the pressing

1. MakeCode © 2020 Microsoft and Adafruit. Screenshots used with permission.

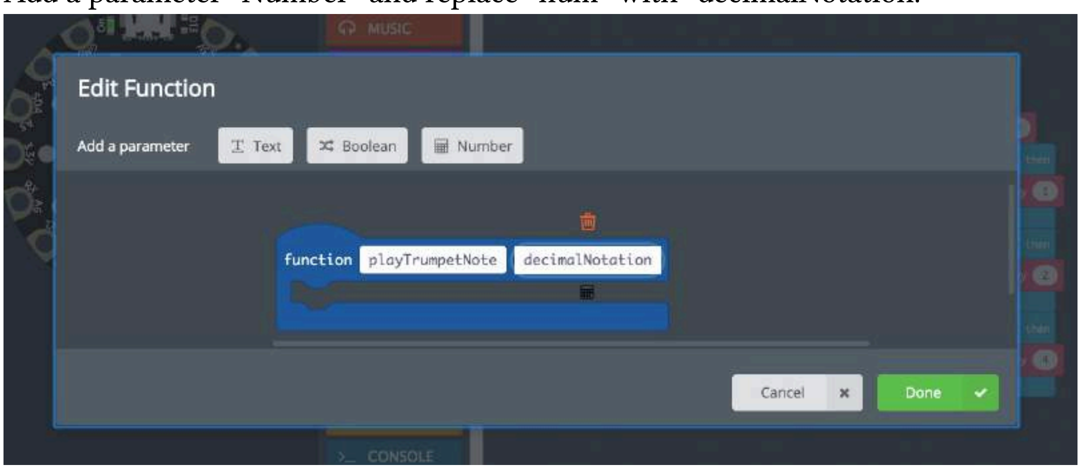
of multiple capacitive touch sensors simultaneously. At this point, we can only assume the “change” statement will add in the value, rather than replacing the previous value.

Steps: Playing the Middle C Scale

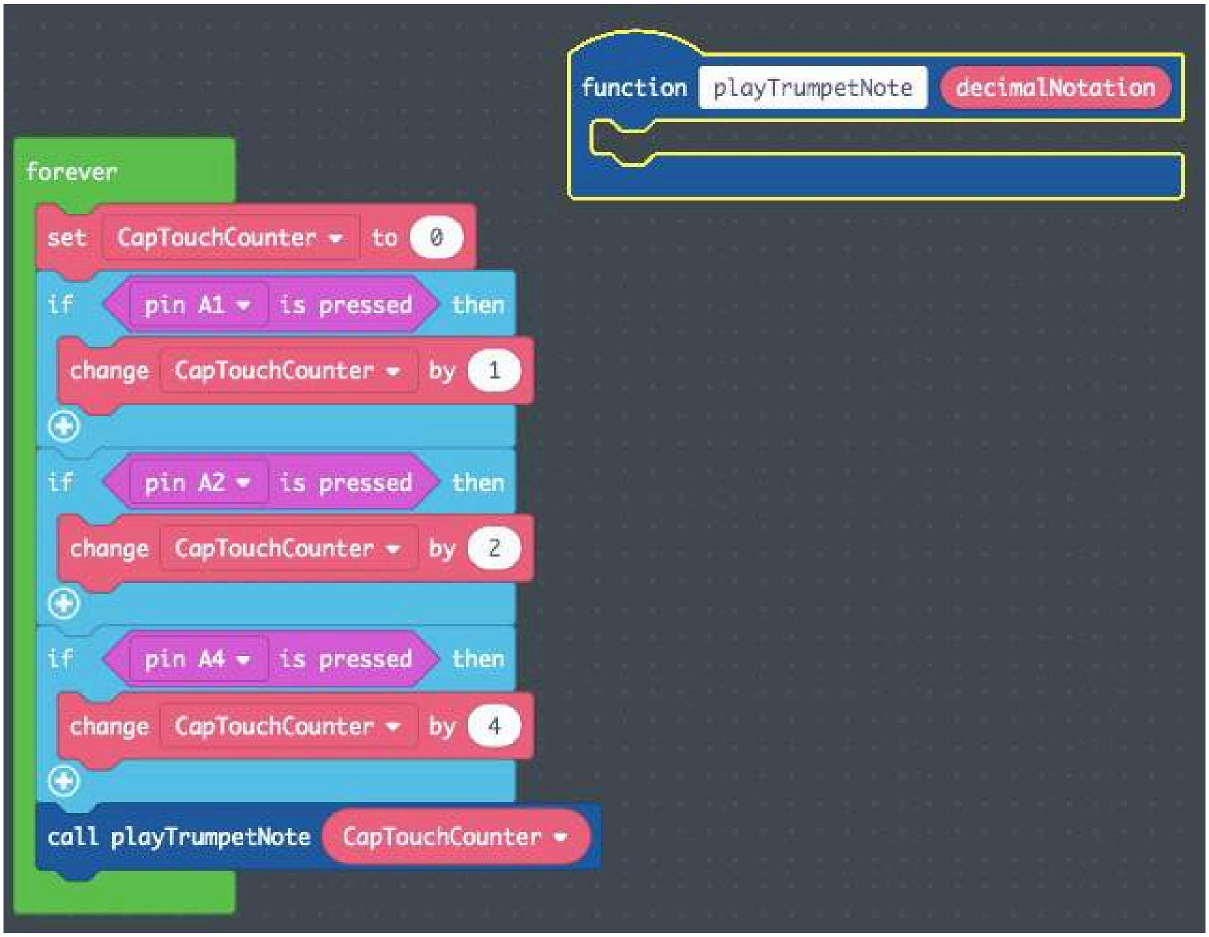
As we move forward, much of the time we use our Circuit Playground Express, we may want to use these to select a digital counterstory to play through speakers rather than playing the notes of a C scale. Sometimes we may want to do both. Sometimes we’ll want to use the calculations for yet other purposes.

For this reason, we will now create a new function called `playTrumpetNote`. We’ll pass the CapTouchCounter value to the function and set it up to determine the Hertz the note should be played at.

1. From within the ADVANCED category, open the FUNCTIONS category. Within FUNCTIONS, click on “Make a Function.”
    - Replace “doSomething” with the name `playTrumpetNote`.
    - Add a parameter “Number” and replace “num” with “decimalNotation.”

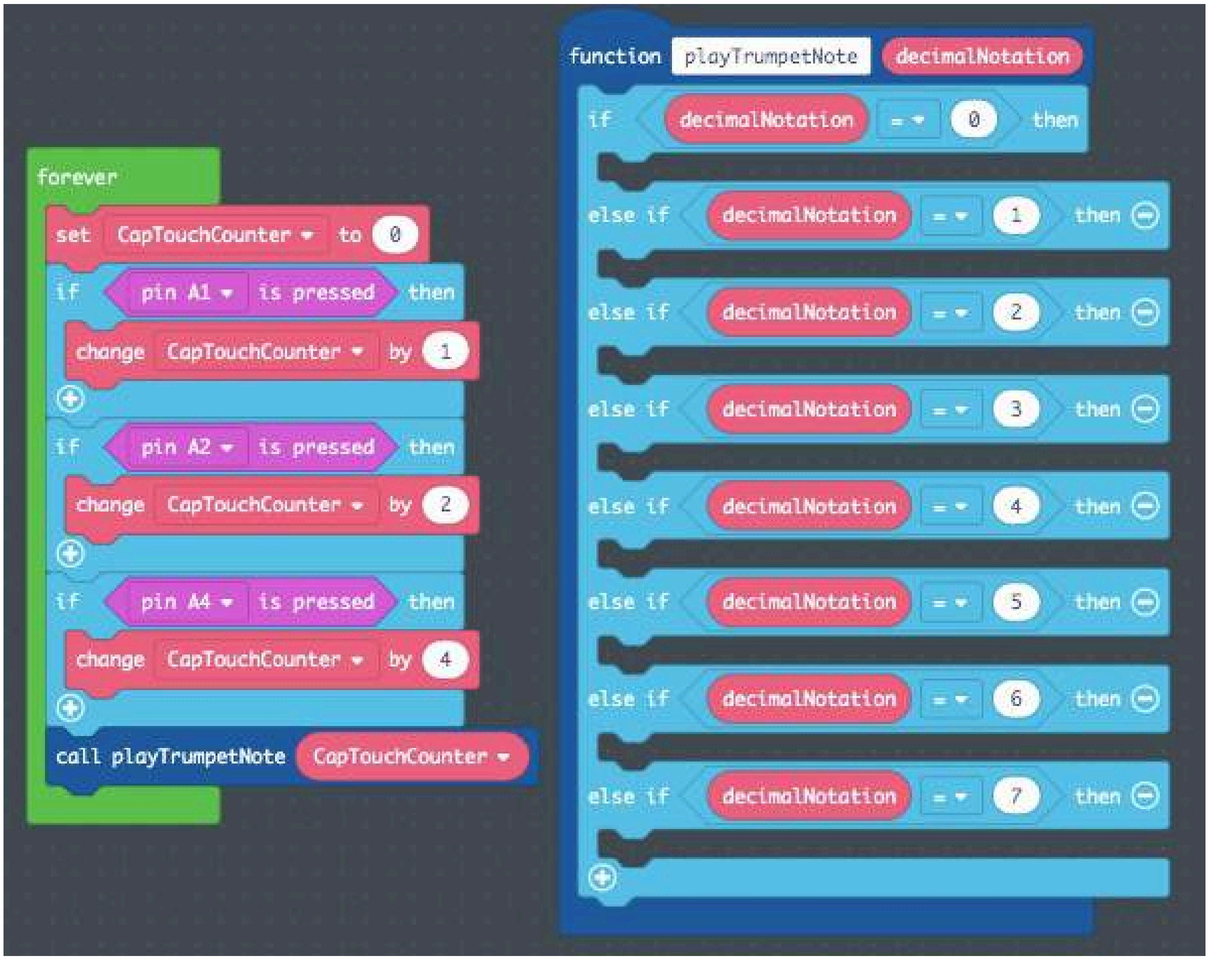
2. From the FUNCTIONS category, drag the “call playTrumpetNote 1” and drop it at the bottom of the current FOREVER function, beneath and NOT WITHIN the end of the “If pin A4 is pressed Then” block.



3. Drag a "CapTouchCounter" block from the VARIABLES category and drop it into the "call playTrumpetNote 1" function, replacing the number 1.

    - A number from 0 to 7 will now be sent to the function playTrumpetNote every time we call it.

4. We'll now use eight "If...Then...ElseIf" logic tests to work through the possible different decimal notation values we may receive given the 0 to 7 decimal scale we might receive via the function call.



5. In the VARIABLES category, create a new variable called `Hz`.
6. Drag the "set Hz to 0" block into the "If decimalNotation = 0" logic statement.
    - When the variable decimalNotation is equal to 0, no capacitive touch sensors are currently pressed, so no note should be played.
7. Copy and paste this "set Hz to 0" 7 times, putting one in each of the remaining "If decimalNotation =" logic statements. Set the Hz to the following numbers:

decimalNotation	Note	Hz
0	Silent	0
1	Middle-C	262
2	Middle-D	294
3	Middle-E	330
4	Middle-F	349
5	Middle-G	392
6	Middle-A	440
7	Middle-B	494

8. From the LOGIC category, drag over one additional “If…Then” logic statement and add beneath the last “else If decimalNotation = 7 Then” conditional.
9. To this new “If…Then” logic statement, replace “True” with a “<0 = 0>” comparison statement from the LOGIC category, within the Comparison section.
    - Replace the left 0 with a “decimalNotation” variable block.
    - Change the “=” comparison to a “>” comparison.

- This will assure that a tone will only be played if one or more of the “valves” is pressed.

10. Finally, from the MUSIC category, drag a “play tone at Middle C for 1/2 beat” and place it within the new “If decimalNotation > 0 Then” conditional statement.

- Drag a “Hz” block and drop it into the “play tone at Middle C for 1/2 beat” block, replacing the “Middle C” window.

11. Take a few minutes to test this out in the simulator. Then save this as `ToolboxTrumpet-MiddleC-variablesFunction`, and flash it to your physical Circuit Playground Express for further testing.

Key Takeaways

Functions serve to run certain control flow statements and other code when called upon by other parts of the program. Indeed, MakeCode itself is written using many thousands of lines of code, most of which we do not see or even know about. But within that code, they make a call to the FOREVER function we create. This is the MakeCode implementation of the Clocked Sequence Programming I did as a graduate student back in 1990 to collect data from a controller I designed to do my own research. Then it was in the C programming language, but the lines of code themselves have a lot of similarities.

By moving the playing of a note into its own function, we can now use the more general capTouch-Counter value for a wider range of activities beyond just playing a note in a C scale, and potentially even instead of that.

Indeed, take a moment to drag an "If…Then" logic block from the LOGIC category and drop it around the "call playTrumpetNote CapTouchCounter" block. Then, at the very bottom of the INPUT category, drag out a "switch right" block and replace the "True" statement within this "If…Then" logic block. Save and flash this to your Circuit Playground Express. Now, depending on the position of the slide switch located at D7 on the Circuit Playground Express, the notes of your scale will or will not be played when pressing the "valves." However, the CapTouchCounter values will still be calculated, as seen if you use a CONSOLE category "console log" block to record the "CapTouchCounter" value.

Exercise: 4-Octave Toolbox Trumpet

In "Essential Coding Concepts," we set up the momentary switch push buttons so that the blue LED was on unless button A was pressed, and the white LED was off unless button B was pressed. Essentially, this was equivalent to:

- No buttons = 1 LED
- Left button = 0 LEDs
- Right button = 2 LEDs

This design could be seen as equivalent to the Middle C scale with no buttons pressed, Low C scale with button A pressed, and High C scale with button B pressed. Let's do a remix of our code to see how an electronic device can do the same thing, sometimes with added functionality, by remixing the lines of code. To do this, we'll add these calculations into the FOREVER function, and then pass them on to the playTrumpetNote function.

Steps and images are provided. Or, you can download the completed file:

[circuitplayground-ToolboxTrumpet4Octave-variablesFunc.uf2]

Open and review it in [MakeCode], and flash it to your Circuit Playground Express for further prototyping.

The key additions:

- Add a new variable called `buttonCount` to conduct three "If…Then" conditionals, setting buttonCount to 0 if only Button A is pressed, to 2 if Button B is pressed, and to 3 if both Button A and Button B are pressed. Otherwise, buttonCount remains at its default of 1.
- Pass buttonCount as a second number to the function `playTrumpetNote`.
- Divide the variable Hz by 2, taking it down one octave, if buttonCount is 0. Else, the variable Hz is multiplied by buttonCount. If neither button was pressed, it equals 1 and the Hz stays in the middle octave. If button B is pressed, the Hz is multiplied by 2 and it goes up one octave. If both buttons are pressed, it is multiplied by 3, and the Hz goes up two octaves.

Steps

1. Create a new variable called `buttonCount`. At the top of the FOREVER function, add a "set buttonCount to 1" block so that in each iteration, this value is reset to the value if neither button is pressed.
2. Add a new set of "If…Then" logic statements, such that:
    - If button A is pressed and button B is pressed, then set buttonCount to 3.
    - Else if button B is pressed, then set buttonCount to 2.
    - Else if button A is pressed, then set buttonCount to 0.
3. Right click on the "call playTrumpetNote" and select "Edit Function." Add a new number parameter and call it "buttonCount."
4. In the "call playTrumpetNote" function, add a "buttonCount" block as the second number in the function call.
5. Between the "decimalNotation" "If…Then…ElseIf" logic statements and the "If decimalNotation > 0 Then" conditional statement, add in a new "If…Then…Else" logic block from the LOGIC category. Configure this sequence of lines of code such that:
    - If buttonCount = 0, then set Hz to Hz ÷ 2.
    - Else, set Hz to Hz × buttonCount.

Wrap Up

We should now have two variables, CapTouchCounter and buttonCount, that we can pass to functions. As a part of the "Comprehension Check" for this session, let's next look at how these variables can be remixed in different functions.

Comprehension Check

Do Something New!

We've used the LED Pixels as indicator lights and even 5 digit counters in a few exercises now. As a "Comprehension Check," create a new LED Pixel Counter function to do the same with our Toolbox Trumpet. Indeed, this can prove helpful if we've forgotten we have the sliding switch to the left, keeping the Toolbox Trumpet notes from being played by the Circuit Playground Express speaker. Record a video image of this being run from your physical Circuit Playground Express, as well as a copy of a screenshot of this new function as a way of demonstrating you've learned the core steps to create a new function.

Design Criteria

1. Open up the "ToolboxTrumpet4Octave" project in MakeCode. Rename this as "Toolbox-Trumpet4Octave-pixelCounter" and click on save.
2. Create a new function called "pixelCount" and add in the number called "digitalNotation" and the number called "buttonCount."
3. Create blocks of code such that pixel 9 is one of four colors, depending on the value of the variable "buttonCount."
4. Create blocks of code such that pixels 0 to 6 are turned on to match "digitalNotation – 1" and off if no values are pressed.

[Watch the video "Toolbox Trumpet 4 Octave with Pixel Counter" for a short demonstration of a simulator test of the prototype I created.]

4A: Sharing Our Counterstories

Background Knowledge Probe

- What is the first thing that comes to mind when you read or hear the word **community**?
- The word **culture** and/or **ethnicity**?
- The word **wealth**?
- What is the first thing that you say when someone asks you about yourself in relation to one or more of these terms?

A Self-Reflection on Sociotechnical Systems

My mother and my father's parents came to the United States in their 20s. Each of their journeys began in Eastern Europe, where they were a part of Volhynian German communities. My family's roots in the Volhynia region go back to the 1850s. Of those family members who lived through destruction of lands, forced starvation, and deportation to Siberia, some of my relatives have remained there, while others of my close family moved to Germany, the United States, and Canada. Some moved before World War I, including a great grandfather, some moved between the wars, including my father's parents, and some only moved at the end of World War II, including my mother. Some participated as soldiers in the German military, some the Russian military, some the American military; some were soldiers, some interpreters, some medics. Some were early Eastern European participants and leaders of a Midwestern American Christian religion with a commitment to pacifism. Others claimed to be 13 generations "pure" — a racist concept of purity that does not have a scientific basis — and so were allowed by government officials to become members of the Nazi party. All made great sacrifices to protect other family members directly and indirectly. But always we sang, played instruments, and celebrated life to its fullest.

I was born in Benton Harbor, Michigan, in the same hospital as my father. My grandfather, grandmother, and all my uncles and aunts on my father's side lived within a one-mile radius. Much of my grandmother's family, including my uncle, lived a few miles away in another nearby community. Later in life, I learned that my father faced challenges in elementary school because of his health issues, immediate connection to German immigrants, and the associated name-calling based on these. The region where I grew up was one of the areas where the Volhynian Germans found the needed support to immigrate. And they remained together to provide essential community cultural support, just as they had in the Volhynian region before they fled violence, forced recruitment, and civil wars. I am a child and grandchild of immigrants, and I am ethnically of this culture.

My middle and high school days, and my junior college days as well, brought me from the farming community into spaces primarily attended by students of color. And I thrived, participating in a range of in-class and extracurricular music and acting events. I am from a family of tinkers, although recent generations are not the formally-defined tinkers: people within itinerant communities who travel from place to place to mend metal utensils and other home and farm implements. It is from these many different lenses that I reflect on my past and present cultural and ethnic identities and values and explore my future possibilities.

I am one of the first in my families' direct line to receive advanced degrees, even though my speaking, writing, and explorations sometimes haven't fit within expected norms and practices. I also reflect on my identity as a white, cisgender, heterosexual male scholar working at a higher education institution in the United States. Through this, I have great privilege that arises in significant ways through the arrival of the first European settlers on the shores of this continent centuries ago and the introduction of a culture in which the relative worth of whites has been placed above all others.[1]

In this self-reflection, I come to see myself as a person born into and thereby inheriting many different socio-cultural-historical components. I am a first-generation German American; I am ethnically a German Christian socialist pacifist who also has disturbing, close links to Nazism and who inherits the centuries-long embeddedness of American white supremacy. But in addition, I also see myself as a person further shaped by my ongoing interactions with family, neighbors, schoolmates, the broader community, and more as my sphere of influence continued to evolve. Importantly, while much of this shaping comes from those whose socio-cultural-historical components are similar to mine, at times, whether through chance or through choice, abductive leaps in my development have occurred through associations with groups different from me.

In "Whose Culture has Capital?" Tara Yosso brings forward the essential importance of focusing on and learning from "the array of cultural knowledge, skills, abilities and contacts possessed by socially marginalized groups that often go unrecognized and unacknowledged" brought forward by communities of color. In this context, she states that:

1. Truth, Racial Healing and Transformation Implementation Guide, December 2016, W.K. Kellogg Foundation http://www.racialequityresourceguide.org, 7.

> Culture refers to behaviors and values that are learned, shared, and exhibited by a group of people. Culture is also evidenced in material and nonmaterial productions of people.[2]
>
> Community cultural wealth is an array of knowledge, skills, abilities and contacts possessed and utilized by Communities of Color to survive and resist macro and micro-forms of oppression.[3]
>
> Tara J. Yosso, "Whose Culture Has Capital? A Critical Race Theory Discussion of Community Cultural Wealth"

My own cultural influences have come in conflict with the dominant systems within which I have been shaped over my personal lifespan. But it is also true that I sometimes, and perhaps often, fit into these dominant systems specifically because of my white, cisgender, heterosexual male identity. And sometimes I fit in because I have taken on as part of my practice the dominant systems. And sometimes, perhaps even often, I inculcate the practices of those dominant systems into my students and into my socio-technical artifacts. Rather than focusing on and learning from socially marginalized groups, I emphasize dominant narratives, structures, artifacts, and systems.

Yet, I also continue to consider the ways my culture and ethnic identity matches that of my family — who often de-emphasized income and wealth accumulation and instead emphasized the broader community cultural wealth within Yosso's model, which includes aspirational, familial, social, navigational, resistant, linguistic, and cultural capital.

In what ways do the pedagogy of this textbook and the hands-on exercises therein serve the goal of education "to 'fit' students constructed as 'other' by virtue of their race/ethnicity, language, or social class into a hierarchical structure that is defined as a *meritocracy*," as so effectively highlighted by Gloria Ladson-Billings in her seminal article "Toward a Theory of Culturally Relevant Pedagogy"?[4]

The Blue Unit has sought to introduce the essential skills needed in support of the data-to-information pipeline fostering computational tinkering. We've explored the journeys of hardware and programming logic that have brought us to this point within the socio-technical artifacts of our daily lives, and the methodological landscape we can use to decodify the dominant understandings of these artifacts. Now the question turns to ways we can "systematically include student culture in the classroom as authorized or official knowledge," leading to recodification of the essential defining terms and skills of our everyday technologies.[5] To do so, we need to continuously research the self, research the self in relation to others, join in engaged reflection and representation, and shift from self to system. These

2. Tara J. Yosso, "Whose Culture Has Capital? A Critical Race Theory Discussion of Community Cultural Wealth," *Race Ethnicity and Education* 8, no. 1 (March 2005): 75. https://doi.org/10.1080/1361332052000341006.
3. Yosso, "Whose Culture Has Capital?", 77.
4. Gloria Ladson-Billings, "Toward a Theory of Culturally Relevant Pedagogy," *American Educational Research Journal* 32, no. 3 (September 1995): 467. https://doi.org/10.3102/00028312032003465.
5. Ladson-Billings, "Toward a Theory of Culturally Relevant Pedagogy," 483.

four key points constitute the framework brought forward by H. Richard Milner IV in the article "Race, Culture, and Researcher Positionality."

And directly as related to this session of the book, these four key points hopefully provide a means by which we can question anew our own counterstories.

The essential resources listed below have been an active part of my more recent works to structure my own courses and structure the layout of this textbook. My personal reflections at the start of this chapter are not placed in this section to defend or justify the works of the Blue Unit, or of the textbook more generally. Rather, I bring them forward as a hopefully useful example of an individual's self-reflection and location within a unique, interdependent system of cultures and ethnicities. This system influences my participation in, and attempts to counter, the webs of oppression which discriminate against and disadvantage certain races, classes, and genders over others. And it seeks to prompt a guiding question for the remainder of this unit: how can we, as a community of practice, work to further design and prototype our work in ways that allow us as professionals to counter the normative webs of oppression around us as we move forward?

Lesson Plan

As you work in our hands-on exercises, consider the influencing factors shaping the information system you create. MakeCode, the capacitive touch sensors and the momentary switch buttons of the Circuit Playground Express come together to collect inputs and determine audio outputs. These inputs can even be stored to track requests. Why is this information system the primary technical exercise for the Blue Unit? What cultural and pedagogical factors might have shaped the design of this unit? For the advancement of what capital? In what ways is the design of these coding exercises helping or hindering the advancement of the core learning outcome objectives for this unit? Why?

Essential Resources:

- Yosso, Tara J. "Whose Culture Has Capital? A Critical Race Theory Discussion of Community Cultural Wealth." *Race Ethnicity and Education* 8, no. 1 (March 2005): 69–91. https://doi.org/10.1080/1361332052000341006.
- Ladson-Billings, Gloria. "Toward a Theory of Culturally Relevant Pedagogy." *American Educational Research Journal* 32, no. 3 (September 1995): 465–91. https://doi.org/10.3102/00028312032003465.
- Milner, H. Richard. "Race, Culture, and Researcher Positionality: Working Through Dangers Seen, Unseen, and Unforeseen." *Educational Researcher* 36, no. 7 (October 2007): 388–400. https://doi.org/10.3102/0013189X07309471.

Key Technical Terms

- Data communications between master and slave, parent and worker, and other **integrated circuits**, **controllers**, and **computing** devices
- Protocols for communication between devices, including **I2C**, SPI, and **UART**

Professional Journal Reflections:

1. Take time this week to pause and silently reflect on the reading of the words within this chapter and within the essential resources. Silently reflect on the different contexts from which these come. And silently reflect on the coding activities of this session. Reflect but don't write. Not yet.
2. Now silently reflect on your self, and your self in relation to others in this community of inquiry. Reflect but don't write. Not yet.
3. Now silently reflect on your ongoing explorations of what it means to be a professional within the information sciences, within informatics, and/or within a profession that facilitates the selection, adaptation, training, and use of information systems. Reflect but don't write. Not yet.
4. Yet: Bring these silent reflections together into one or several written, drawn, or spoken Professional Journal Reflections.

4B: Raspberry Pi Counterstory Little Free Library

Background Knowledge Probe

- Which of your daily-use mechanical tools, if any, have attached digital electronic components? Think about this broadly, including your car or kitchen appliances, for example.
- Which of these digital electronics are microcomputers? Microcontrollers? A combination of the two?
- In what ways is the design and use of the digital component prioritized over the mechanical? The mechanical over the digital?
- In what ways are your daily-use tools, mechanical and digital, prioritized over the person? In what ways are your daily-use tools first and foremost person-centered?

Technical Overview

We've slowly been working towards creating a digital Little Free Library extension to the wonderful analog Little Free Libraries we find in many neighborhoods. Microcontrollers like the Circuit Playground Express provide a range of sensors and indicators that can be very useful to communicate back and forth with Little Free Library patrons. Microcomputers like the Raspberry Pi, on the other hand, serve as an e-library stacks, library catalog, chat room, and much more. Together, these can be hidden away in a toolbox and run with an extension cord or solar panel and battery pack. For now, our goal is to bring the Circuit Playground Express and Raspberry Pi together. We will set up the Circuit Playground Express to communicate through serial connection with the Raspberry Pi, and the Raspberry Pi to play MP3 audio recordings through its audio jack. In this exercise, we will begin to explore the possible relationships between a single-purpose computer, such as the Circuit Playground Express, and a multi-purpose computer, such as the Raspberry Pi.

Data Communications Between Devices

The Raspberry Pi, like many microcomputers and microcontrollers, provides pins to which you can attach transmit and receive wires to connect with personal computers. In this way, even if there is no network or keyboard, mouse, and monitor to provide access to one of these devices, a simple USB to serial cable can be used to configure, diagnose, or perform regular tasks on it. Many operating systems, such as the Raspberry Pi OS, also provide a control window to output and input text from and to the device through this serial interface.

A note on the technical terms **Master** and **Slave**:

- Historically, the technologists who developed these data communication tools used the term **Master** to indicate the controlling device and **Slave** to indicate devices being controlled by that device. I follow the lead of Python developers who, moving forward, are using the terms **Parent** and **Worker**, respectively. These changes within Python coding were made in November 2018, after heated discussion launched by a change request submitted in September 2018 by Victor Stinner, a developer with the software company Red Hat.[1]
- Other programming languages have implemented similar changes. Drupal uses the term replica to replace slave and primary to replace master. As another example, Django has moved to follower and leader.
- The terms still largely used within engineering and computer science remain slave and master. In discussions regarding these change requests, many argued that these common terms are used to refer to the operation of technology independent of, or at the direction of, another technology with no reference to human slavery. They argue that because there is no common replacement that could easily be made given the existing labels on electronics and in code back many decades, it is important to stay with the stock terminology.
- As you read through this document and look at abbreviations listed on your electronics and in code, note that sometimes the listed abbreviations — for instance MISO, below — do not align with written descriptions in this text — Parent In/Worker Out.

Communications between devices takes many shapes and forms using a range of protocols. Three of the most commonly used are:

- **I2C (Inter-Integrated Circuit and pronounced I-squared-C)** was developed in the early 1980s and is widely used to connect lower-speed peripheral **integrated circuits** to microprocessors and microcomputers using the bus topology (think of a physical bus you ride to travel with others between locations). It is designed to work over relatively short distances, and allows multiple "worker" integrated circuits to communicate with one or

1. Daniel Oberhaus, "'Master/Slave' Terminology Was Removed from Python Programming Language," Vice, September 13, 2018. https://www.vice.com/en_us/article/8x7akv/masterslave-terminology-was-removed-from-python-programming-language.

more "parent" integrated circuits. I2C only requires two wires and can support up to 1008 worker devices. It can also support multiple parent devices, allowing for more than one parent to communicate with all devices on the bus.

- **SPI (Serial Peripheral Interface)** is another widely used bus topology protocol used to send data between peripherals and microcontrollers/microcomputers. What is unique about SPI is that it uses a serial clock in addition to a MOSI (parent out, worker in) data line and a MISO (parent in, worker out) data line. The serial clock's oscillating signal allows for data communications to occur synchronously. As with I2C, multiple workers can connect to a single parent. To accomplish this, a fourth line, SS or Worker Select, alerts a particular worker that it should wake up and send and/or receive data, as well as to detect that multiple workers are present.
- **UART (Universal Asynchronous Receive Transmit) interfaces**: UART is actually a microprocessor with a package of integrated circuits on a **printed circuit board**, along with program logic, and is used to attach serial devices to a computer. This allows the computer and attached devices to "talk" and exchange data. There is a range of USB (Universal Serial Bus) TTL (Transistor Transistor Logic) serial cables used to provide connections between USB and serial UART devices. For our purposes, UART is the simplest and a very effective strategy for basic work in the console window of the Raspberry Pi using a personal computer.

Exercise: The Circuit Playground Express Meets the Raspberry Pi

In the first three technical session chapters, we've explored MakeCode and configured our Circuit Playground Express microcontroller, eventually working to create our own functions. In this session, we'll bring together the microcontroller and the Raspberry Pi microcomputer to allow us to use the "trumpet valves" to play selected counterstories saved on the Raspberry Pi. To do this, we'll use the UART serial communications pins found on the Circuit Playground Express and the UART Python extension on the Raspberry Pi to transmit valve press sensor data from the Circuit Playground Express to the Raspberry Pi, and to then assess this data to choose which of the available MP3 counterstories to play using the audio output of the Raspberry Pi.

We'll accomplish this in three steps:

STEP 1: Set up the Raspberry Pi to store and play counterstories using the VLC free and open-source, cross-platform multimedia player. This may have already been done previously, as it is an exercise within the Orange Unit.

STEP 2: Set up your Circuit Playground Express. The Toolbox Trumpet will now be expanded to use components of the MakeCode Serial UART Communications Extension to transfer capacitive touch and momentary switch data to the Raspberry Pi to play counterstories based on sensor inputs.

STEP 3: Use the Toolbox Trumpet to play short counterstories (no momentary switches pressed), introductions to the counterstories (left switch pressed), and extended counterstories (right switch pressed). A simple Python-based program has been included to test out the Circuit Playground Express, the USB to TTL serial cable, and the Raspberry Pi as a media center.

To work through these steps, it is now time to bring together the technical activities with the social activities. Work through these in pairs. Or perhaps before doing these activities in pairs, join in a community of practice to consider ways to approach this within a diversity of community cultural wealths that are present. This is to be a work not of doing alone, but of being present within the space, with the people in this room. The book *Emergent Strategy* highlights several principles relevant in considering technical activities within a community of practice, and a community of practice as something that is completed through being present rather than as driven-doing acts. These include:

> There is always enough time for the right work.
>
> There is a conversation in the room that only these people at this moment can have. Find it.
> Never a failure, always a lesson.
>
> brown, adrienne maree. Emergent Strategy (p. 44). AK Press. Kindle Edition.

How can the following be accomplished in such a way that the community cultural wealth of all present are advanced as part of taking on these activities collaboratively?

How can this be part of a larger task of advancing a counterstory regarding sociotechnical innovation-in-use?

Step 1: Transfer MP3 Files to the Raspberry Pi

If you haven't been using the Raspberry Pi since the end of the Orange Unit, it might be helpful to review Orange Unit 4B: Meet the Microcomputer, including videos found within the chapter, and Exercises found in Orange Unit 4C: Getting Started with the Raspberry Pi.

- If you need help using RealVNC (**V**irtual **N**etwork **C**omputing) on your laptop to serve as the keyboard, mouse, and monitor interface with the Raspberry Pi, review the later steps in the Exercise: First Boot of the Raspberry Pi OS.
- If you are now connected to a different Local Area Network (LAN), you'll need to connect the Raspberry Pi to a keyboard, mouse, and monitor temporarily to get it to connect to that network. If you have problems getting the monitor to serve as a screen for the Raspberry Pi in this new location, you'll find FAIL FORWARD Tips in the Exercise: First Boot of the Raspberry Pi OS.
- If you need help using the PiOLED to identify the IP address so that you can reconnect the VNC View on your laptop to the VNC server on your Raspberry Pi, review Exercise:

Adding an Adafruit PiOLED Text Display.

With the Raspberry Pi up and running, the first step is to set up a folder called "cstories" in the home directory of your Raspberry Pi, containing your counterstory MP3 files. You may have completed this step as part of the Orange Unit. If not, jump to Coding Electronics. In the steps below, we'll continue working with this folder.

```
pi@raspberrypi:~ $ mkdir cstories
pi@raspberrypi:~ $ cd cstories
pi@raspberrypi:~/cstories $ ls
  cStory1.mp3  cStory1-extended.mp3   cStory1-intro.mp3
pi@raspberrypi:~/cstories $
```

Transferring files from your laptop or other source will depend on a number of factors. For many, the File Transfer toolbar button in the VNC Viewer dropdown window has worked exceptionally well when VNC Viewer on your laptop is connected to the VNC server on the Raspberry Pi.[2] Others have saved files in a central course Learning Management System (LMS) or "Cloud" storage space such as Box. These files can then be downloaded to the Raspberry Pi using the Chromium web browser preinstalled within the Raspberry Pi OS. For those familiar with the Linux command line, another way is to make use of the SCP command.

However you do this, make sure to transfer your .mp3 counterstories to the /home/pi/cstories subdirectory.

NOTE regarding the naming of your counterstory files:

- The prefix of your counterstory files should be "cStory", where **lower and upper cases do matter, as Unix and Linux-based operating systems are case-sensitive**. The applications and Python code we run later will use a matching case-sensitive naming structure.
- Next, add a number between 1 and 7, with 1 being the first story told when the A1 is pressed on the Circuit Playground Express, and 2 being the second story told when the A2 is pressed, etc.
- The postfix of the filename is .mp3. For the two- to three-minute main part of the story, for the first counterstory to be told, you would now see the name:

  ```
  cStory1.mp3
  ```

- If it is the short introduction to the story, add a -intro between the story number and the postfix. The name of the introduction to the first story would thus be:

2. The instructions for enabling the VNC Server Interface on the Raspberry Pi and for setting up the VNC Viewer on your laptop can be found starting at STEP 5 of the Exercise: First Boot of the Raspberry Pi OS in Orange Unit 4C: Getting Started with the Raspberry Pi

```
cStory1-intro.mp3
```

- If it is the part providing the listener with further information, add a -extended between the story number and the postfix. The name of the additional information on the first story would thus be:

```
cStory1-extended.mp3
```

A quick way to test this out is by playing your MP3 file using the VLC media player, which runs either as a graphical display (VLC) or as a command-line program (CVLC), which is what we'll run from a terminal window:

```
pi@raspberrypi:~ $ cd cstories
pi@raspberrypi:~/cstories $ cvlc cStory1.mp3 vlc://quit
```

FAIL FORWARD TIP: You may need to right-click on the Volume control icon at the upper right of the Raspberry Pi menu. Doing so will show you the available audio ports, by default "AV Jack" and "HDMI." Plugging in a USB Headset or some speakers will add further options, from which you can hear the story currently being played by CVLC. Be sure to choose the right option for the media player you are wanting to use.

Step 2: Set Up Your Circuit Playground Express

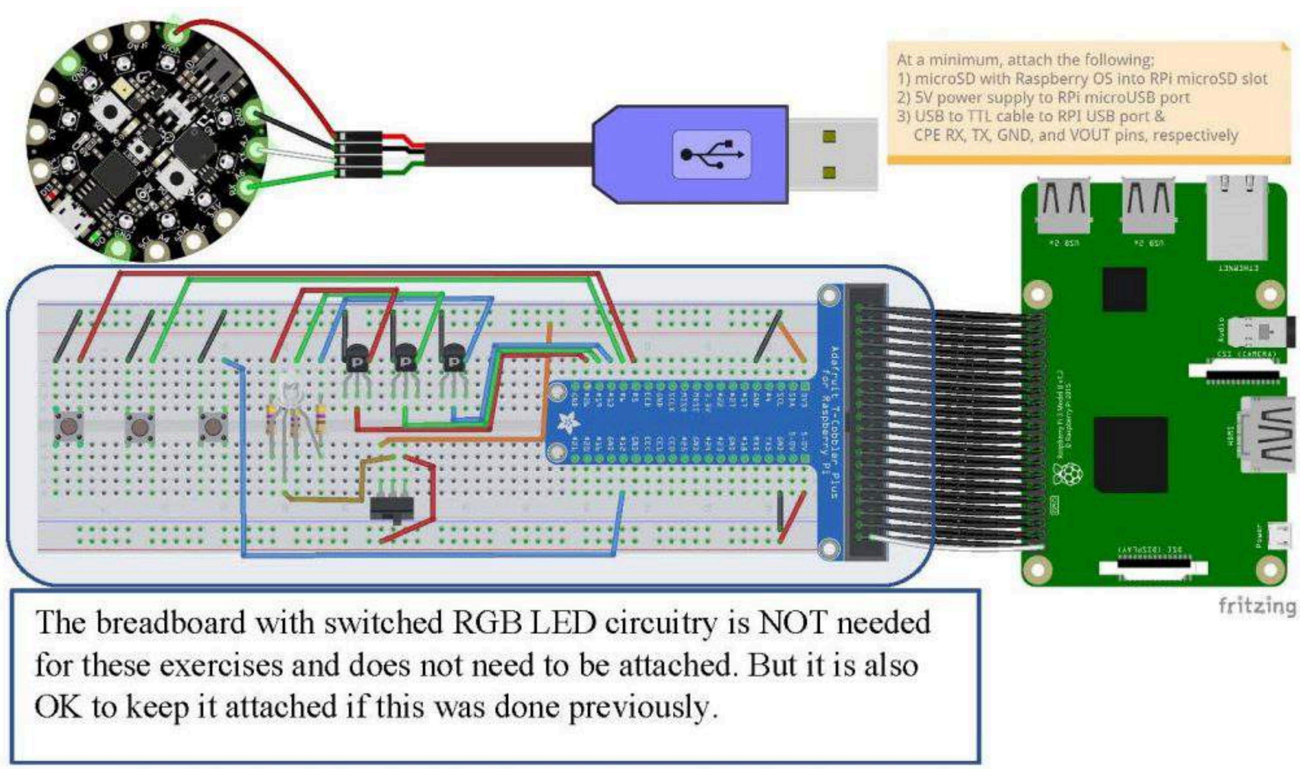
On your laptop, make the following update to the Circuit Playground Express so that it can work with the Raspberry Pi to play counterstories:

2. Download the 4-Octave Trumpet Counterstory UART Prototype for the Circuit Playground Express: https://go.illinois.edu/CPE_4OctavePlusUART
3. Review the new conditional that has been added to the MakeCode, building from the previous Toolbox Trumpet 4 Octave exercise. In this case, we now do a serial write of data whenever the capacitive touch sensor count is greater than zero. This is possible because within the Advanced section, we've taken a serial redirect Configuration block and added that to the On Start function. (NOTE: The first time it is used, you need to go into +EXTENSIONS to find Serial. From then on, it will be located directly within Advanced.)
4. Connect the Circuit Playground Express to your laptop via the microUSB to USB cable and flash the new UF2 to your Circuit Playground Express.
5. Note that you now need to move the off/on switch on the Circuit Playground Express to on to hear the audio playback of the trumpet. Turn this on and double-check that you can still play the notes.
6. Disconnect the microUSB to USB cable.

Step 3: Use the Toolbox Trumpet to Play Your MP3 Files

Now let's connect up your Raspberry Pi to your Circuit Playground Express using a USB to TTL cable and some male to alligator clips. You will now provide power to the Circuit Playground Express directly from the Raspberry Pi via the USB port, by connecting the red TTL wire to the VOUT on the Circuit Playground Express using a fourth male to alligator clip wire. In this way, one power source directly attached to the Raspberry Pi is all that is needed to run the whole digital Little Free Library!

NOTE: Do not connect the red wire to VOUT power if your Circuit Playground Express is getting power from another source, such as your laptop.[3]



1. Use three or four male to alligator clip wires to connect the TTL side of a TTL/USB Serial Cable to the Circuit Playground Express microcontroller:
    - Black wire to ground (GND)
    - Red wire to power (VOUT) (do NOT connect if Circuit Playground Express is getting power from another source, such as your laptop)
    - White wire to transmit (TX)
    - Green wire to receive (RX)

3. Fritzing breadboard graphics are licensed under CC-BY-SA 3.0.

2. Plug the USB side of the second serial cable to one of the four USB ports on the Raspberry Pi. Any one of these should work OK. You should see the Circuit Playground Express green power light go on. If not, a few tests can include:
    - Confirm that there is power on the Raspberry Pi and that it is not in the boot-up phase.
    - Check to see if the cable USB is fully pressed into the USB port of the Raspberry Pi.
    - Try using a different USB port on the Raspberry Pi.
    - Confirm that the GND and VOUT pins are properly connected to the red and black wires on the TTL/USB serial cable.

3. From a web browser, enter: https://go.illinois.edu/cstories
    - Take a glance at the code, but most importantly, right-click in the window to download the program as `cstories.py` (*not* as cstories.txt as automatically listed).
        - Make sure to move cstories.py to the /home/pi/Code/Python directory,
    - You can download the Python code in several ways:
        - Directly from the Raspberry Pi's Graphical User Interface, using a web browser on the Raspberry Pi.
        - By downloading it to your laptop's cstories folder, and then transferring the file using the USB to TTL serial console connection.
        - By downloading it to your laptop's cstories folder, and then transferring the file using SSH/SCP (secure shell and secure copy) in a terminal (Mac) or PowerShell (Windows 10) window. For this approach, first identify the IP address of your Raspberry Pi. Jump to Network Troubleshooting for instructions. Visit Introducing the Unix Command Line for details on using SSH/SCP.
    - The cstories.py code does an import of two libraries with supplemental code: one library for serial communications, which we'll use to communicate with the tty to UART cable, and the second for the VLC library of commands. The serial library comes pre-installed, but we now need to install the VLC library using pip3:

      ```
      pi@raspberrypi:~ $pip3 install python-vlc
      ```

    - If you have more than one cstory with intro, main, and extended files, open cstories.py with Thonny or the graphical editor mousepad to edit the Python code. Change the variable "numStories" from 1 to the number of cstories now on the Raspberry Pi.

```
pi@raspberrypi:~/cstories $ mousepad /home/pi/Code/Python/cstori
```

- Navigate to the row that starts "numStories." Navigate to the number 1, and change it to the number of stories you have transferred to /home/pi/cstories.
- To save in mousepad, click on the Save button and then close the window.
- To save in Thonny, click on the Save button.

4. Test it all out in either Thonny or the Terminal:
   - In Thonny, you can click on the Run icon. Look in the Shell window in the bottom half of Thonny to see the output that would normally go to a terminal window.
   - In the terminal window, type the following, and then touch the different Toolbox Trumpet valves to see if you can hear a story being read to you:

```
pi@raspberrypi:~/cstories $ python3 ~/Code/Python/cstories.py
```

- To exit the Python command output, provide a keyboard interrupt by holding down the Control (that is, CTRL or ^) key, and then hitting the "c" key.

FAIL FORWARD TIPS:

- Sometimes, the connection of the USB/TTL Serial Cable from the Circuit Playground Express to the Raspberry Pi may have changed. In the terminal window on the Raspberry Pi, type:

```
pi@raspberrypi:~ $ ls /dev/ttyUSB*
```

Note the result. If nothing is listed, then the USB/TTL Serial Cable may not be connected. Otherwise, compare the returned value(s) with the one listed in the "uart" variable within the `cstories.py` code. The best way to do this is by using nano. If the value returned from the "ls" command is different from that listed in "uart = serial.Serial," then change the number accordingly in the Python code, save, and retest.

- The cstories.py code does not have a way to cut short the playing of a selected story. The VLC player also can handle the playing of multiple stories by default. Stop the cstories.py prototype to end the telling of one story early, and restart the Python program again to start a new story with the A1, A2, and A4 key combinations, along with Button A or Button B as needed.

From Here

If you'll be jumping in and out of your Raspberry Pi periodically, feel encouraged to leave it powered on, just as you would any server. Exit VNC Viewer or close out of your ssh connection to the Raspberry Pi by hitting ^D (that is, holding down the Control key and hitting the 'd' key), or by typing `logout`. Either way, you should see the following window display at the end, making it possible for you to reconnect via VNC or SSH back in via Terminal or PuTTY whenever you need to.

Wrap Up

The primary objective of the Blue Unit is to bring together our microcontroller and microcomputer systems as a means of sharing data between devices, thereby creating a basic physical network. That is, we've worked to create a digital networked information system. Together, we can advance a range of technical skills in addition to a number of cognitive and socio-emotional skills to help your community of practice use a growth mindset to build these multi-layered systems.

Do Something New!

Entering into this final stage putting together a Raspberry Pi Counterstory Little Free Library, there have been a range of expertise and community cultural wealth brought into the activities as a team. For some, strengths may especially be in the computer programming front. For others, you may bring with you a background within issues of social justice. Consider also the diverse lived experiences, cultures, and other influences that have shaped your selection of a story that you have turned into one you would tell through this sociotechnical system.

- How might you now work together to create a series of seven stories selected to especially speak to a special audience?
- Which would play with A1, that is cStory1?
    - A2, that is cStory2?
    - A1 + A2, that is cStory3?
    - A4, that is cStory4??
    - A1 + A4, that is cStory5?
    - A2 + A4, that is cStory6?
    - A1 + A2 + A4, that is cStory7?
- What would play if someone simultaneously pressed both Button A & B? How might this be used as a secret "Easter Egg" part of the story (Easter Egg is a gaming metaphor for a hidden aspect).

The exercises have stepped you through the sharing of this story through integration of the Circuit Playground Express UF2 code and the Raspberry Pi Python code. The Python code itself provides opportunities to share between 1 and 7 different stories headless, that is without a monitor. But as we learned in Orange Unit 4C: Getting Started with the Raspberry Pi, Python also has a library file to work with a the PiOLED we've been using to identify the IP address assigned to the Raspberry Pi.

- How might you reuse the PiOLED to instead indicate a list of stories and to share which story is currently being played?
- Are there ways we could use a small, low-powered monitor attached to the Raspberry as part of a portable Little Free Library to indicate a list of stories and to share which story is currently being played?

As you explore this *Do Something New!* be sure to use the various strengths of your team, and seek to bring in other expertise as possible, to investigate and prototype these possibilities of a sociotechnical system to share seven strategically selected critical social justice stories within a person-oriented social perspective.

Comprehension Check

Now would be a great time to revisit the lesson plan listed in "Valued, Inclusive Information and Computing Technology Experiences." Spend some extended time with your Professional Journal Reflections this week, especially responding to the probes at the end of that session. As you reflect, bring our activities to date into conversation with the capability approach and inclusive computational thinking frameworks brought forward in this session.

1. In what ways might the toolbox trumpet and Raspberry Pi counterstories serve as a resource advancing agency? A capability?
2. In what ways might the toolbox trumpet and Raspberry Pi counterstories serve as an information system problematically disrupting valued beings and doings of individuals and communities?
3. In what ways might the toolbox trumpet and Raspberry Pi counterstories be designed for designers, that is, for the innovators-in-use who adapt this to their own functionings? In what ways might it have been designed differently leading to this point?

Blue Unit Review

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with the option to export your responses, use the online version.]

Key Learning Outcomes Self-Check

Listed below are the key learning outcome objectives for the various skill sets we sought to advance in the **Blue Unit** of the book. Using the scale below, take a moment to create a second ***1-4 rating*** of these items based on where you think you are in your skills development now that you've completed a pass through the Blue Unit cycle. Add additional notes as helpful to keep further specific reflections on your development of these specific skills.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough
4: Yet

Technical Skills

You can (rate 1-4):

- Understand computer science programming concepts, including: Conditional and Boolean logic, iteration loops, variables, and modularization.
- Tinker with the programming languages MakeCode and Python.
- Use programming code to interact with a circuit to achieve an application goal.

Information Skills

You can (rate 1-4):

- Explore a range of information sources, guides, and worksheets in support of current research and practice.
- Apply reference interview skills to design context so as to better understand the strengths, opportunities, and aspirations that are motivating project initiation by an individual or social group.

Cognitive Skills

You can (rate 1-4):

- Generalize past and current problem solving to new contexts.
- Consider the ethical implications of creating or adopting specific technologies and implementations within different social, economic, and cultural contexts.
- Logically analyze the steps for effective adoption and application by different individuals of a specific technology implementation to support individual and social group aspirations and goals.
- Turn personal and community strengths, opportunities, and aspirations into project ideas.

Socio-Emotional Skills

You can (rate 1-4):

- Communicate and collaborate effectively with others.
- Nurture personal confidence, persistence, and tolerance to tackle complex, ambiguous, open-ended problems.
- Empathize with individuals and social groups across a wide spectrum of social and cultural contexts.

Progressive Community Engagement Skills

You can (rate 1-4):

- Use an informal, distributed process to share the responsibility of leadership among all team members of a community of practice in ways that build off each member's strengths and unique insights.
- Understand and respect broad diversities of functioning with regard to analog and digital technologies, and to see this functional diversity not as a deficit, but rather as an inherent and valued aspect of being human.

- Interact as a community of practice in ways that advance each member's capability set and positive freedoms, so as to more effectively achieve and enjoy their valued beings and doings.
- Focus on and learn from the array of often unrecognized and unacknowledged cultural knowledges, skills, abilities, and contacts possessed by a community of practice, and especially those of socially marginalized members.

Critical Sociotechnical Skills

You can (rate 1-4):

- Analyze how economic, social, cultural, historical, and political contexts, and personal preferences and biases, are embedded into a specific implementation of technology throughout the life cycle of the technology.
- Evaluate positive and negative impacts that different technologies and specific implementations of a technology have on individuals and social groups from different economic, social, and cultural contexts.
- Consider the social and individual ramifications of choosing open versus closed software and hardware products within different contexts and for different social, economic, and cultural contexts.

Your Personal Learning Outcome Objectives

Add the personal learning outcome objectives you created at the beginning of this Unit.

Rate Your Personal Learning Outcome Objectives

Take a moment to create a second ***1-3 rating*** of these items based on where you think you are in your skills development now that you've completed a pass through the Blue Unit cycle. Add additional notes as helpful to keep further specific reflections on these specific skills development.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough

Goals	Rating
Your first goal...	

REMIX: Ideating and Iterating Code: Scratch Example

Betty Bayer and Stephanie Shallcross

This "real life" hands-on project by four information science graduate students provides a step-by-step example of ideation and iteration of code to support counterstories.

The Process: Inspiration and Ideation

As a group of four current and future educators, we knew we wanted to focus on one person, whose story is not well known in the history of technology, and we wanted the game to be fun and accessible for late elementary students. We were particularly interested in inspiring young girls because women, as a whole, are a group that is underrepresented in technological fields. Often when children are taught the history of technology, women who have contributed to the field are left out. We hoped that by creating a game that was fun and accessible for late elementary students young girls who were interested in technology would be inspired to learn, create and change the way women are represented in technology fields.

From this initial spark of inspiration, we quickly moved into the ideation phase. Fueled by the recent release of the picture book *Grace Hopper: Queen of Computer Code* by Laurie Wallmark, the group chose to focus on Grace Hopper. Much of the initial debate focused on whether or not Grace Hopper's story was truly a counter-story. However, we decided that since none of us—four educated women, who work with children and technology—had heard of her, her story did indeed meet the requirements of a counter-story.

Then, the group member who had by far the most experience coding with Scratch, suggested using a maze with scrambled letters to decode.[1] This seemed both feasible technically for us and fun and accessible for our target audience. We decided we wanted to have multiple levels of the game, each telling a different story about Hopper's life. We were especially enthused about telling the story of Hopper

1. Scratch is a project of the Scratch Foundation, in collaboration with the Lifelong Kindergarten Group at the MIT Media Lab.

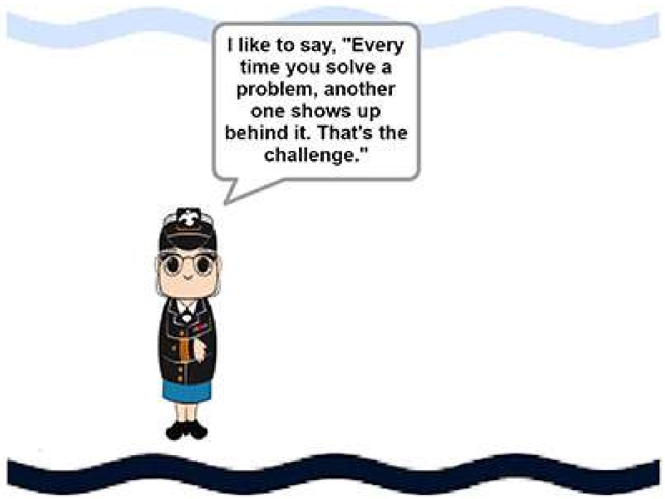


Image 1: We incorporated one of Hopper's well-known quotes that inspired the development of the game into the introduction.

finding the moth—the literal "bug"—in the Mark 2 computer because (A) it's fun and will engage students, but also (B) the term "bug" is mainstream technology terminology, but few people know it was coined by a woman. Knowing this story—and Grace Hopper's in general—can allow girls to see themselves in the mainstream history of technology, and in the terms and code we use today. We also discussed possible stories for the other levels, and some of Hopper's more pithy and applicable quotes we could work into the game [See Image 1].

The Process: Iteration

In an early iteration, we planned to include three mazes. However since one of the reasons we were taking the course was to improve our technical skills, we decided it would be ideal if all four members of the group gained experience in Scratch. So, we decided early on that the best course of action would be to have each group member individually code a level of the game.

Image 2: The backdrop and sprites in place for beginning the Level 2 Maze.

Early in the design process, one team member's letter blocks didn't work when other team member's played the game. Through some pair programming over email and on Blackboard Collaborate, we realized the problem was she used separate sprites as blocks and letters, where as the rest of us used different costumes. We were able to fix this "bug" much more quickly working as a team than one of us could have as an individual.

In another iteration, we started adding more content to each level in order to give children as much information about Grace Hopper as we could. We wanted to make sure our game was person-oriented rather than thing-oriented. After testing this prototype, a couple of the members of our group that work with children in a primary setting stated that the extra information caused a lot of visual clutter on the screen that would make it less accessible to some students–for example, those with visual deficits or those who are easily distracted. So in the interest of keeping our counter-story as tailored to our audience as possible, we decided to get rid of the extra information on each individual level and instead streamline the information into one backdrop at the end of the game. The slide also provided links to more information. This approach would provide the best experience for children of all abilities, and those students who were inspired by Grace's story could easily continue to learn more about her.

Image 3: Set of costumes for the decoy balloons.

One of the members of our team worked in an elementary school library that serves Kindergarten through fifth grade students. In order to get feedback from our target audience, she took our prototype in and asked two fifth grade students to test it out. The students identified two areas of concern. In one level, after all the letters are revealed and the Grace sprite gets to the end of the maze, the question mistakenly covered all of the letters. Based on this feedback, the team was easily able to change the location of the letters, to allow for them to be seen during the question phase.

The second issue the students found was a bigger one: if the mouse pointer touched Grace in levels 2-4, she would return to level 1. This glitch made it impossible to finish the game. Through paired programming, we determined that the issue was in one of the initial sequencing codes, specifically, the code that starts the game. Basically, the code was telling the computer every time Grace is clicked to restart at level 1, which is not what we wanted. It took some time to figure out how to fix this, but eventually, we discovered that by simply adding a stop script control to that particular sequence, we were able to "debug" our game. Without rapid prototyping our game and asking students to play it, we would not have found these errors nearly as quickly. The students' feedback was critical to the next iteration of our game.

Image 4: End screen with link to further information.

Image 5: First iteration of code. This caused the Grace sprite to return to Level 1 whenever she was clicked with the mouse.

Image 6: Second iteration of this line of code. This allows for the sprite to only complete this set of instructions once.

The Process: Mutual Shaping and Teamwork

Our group had a lot of shared social factors since we are all current and aspiring educators. We were inspired by our backgrounds working with children in a school setting, and we wanted to create a game we could use with our own students. Our backgrounds influenced us to style the game as a series of mazes to be more fun and appealing to children. Our backgrounds also determined the level of the content; we wanted to teach players about Grace Hopper in a way that would be understandable and also make them realize they too have the capacity to learn and create technology.

Despite the similarity of our backgrounds, we had different strengths and weaknesses. We were able to leverage each team member's strengths by using pair and triplet programming, thorough communication, and taking on specific team roles. As discussed previously, pair and triplet programming helped us to "debug" our game much more quickly than we could have individually. Initially, each team member worked alone on their level, but almost right away we encountered issues with the letter sprites. If one member of the team tried to program a letter sprite, it moved in other level. Through paired programming, we realized that we needed different sprites for each level and that we had to rename those sprites right away with the specific levels name to avoid confusion. Each sprite needed to be coded separately and each sprite needed to have its own costume, i.e. balloon.

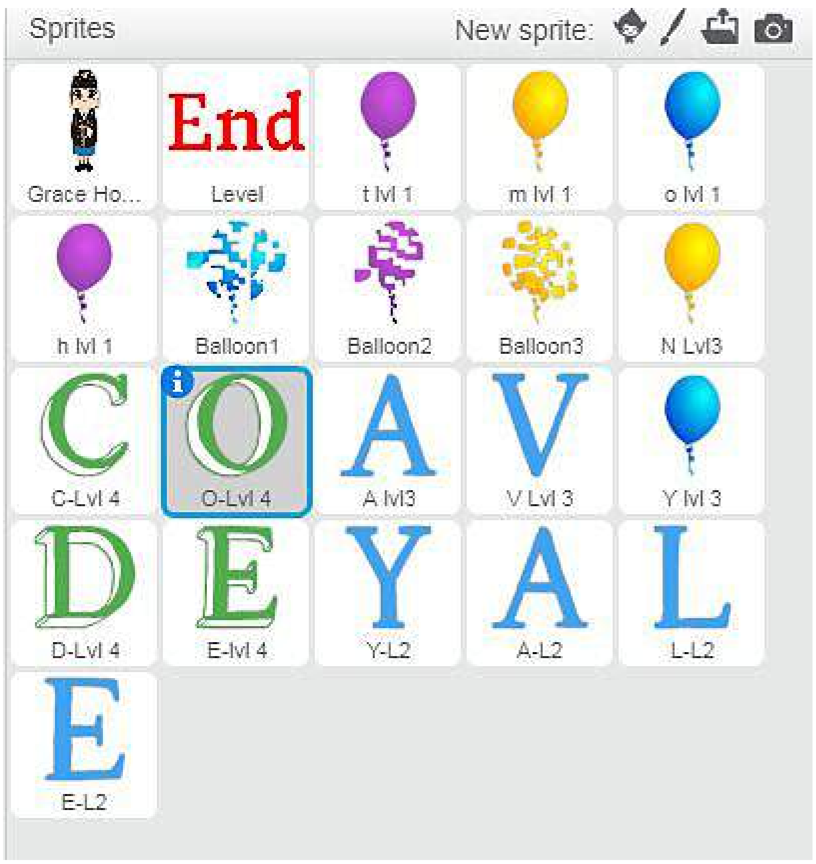


Image 7. This game required the use of 21 sprites. Each with individual coding.

Another aspect of our teamwork that made us successful was the unofficial roles that each member of our team took on. One member of the group took on a project manager role, taking meeting notes, scheduling weekly meetings, and sending out action items for each group member. Another person took on the task of managing the written portion of the project and recorded the group's progress throughout the design thinking process. The person with the most Scratch experience researched ways to code the maze backgrounds. The final person's job was to be the final editor. She carefully walked through all of the programming and located any errors in the programming that required attention, and put the finishing touches on the introduction and conclusion screens of the game.

Although we naturally fell into these roles that fit our own personal strengths and weaknesses, we also were flexible working as a team. If the Scratch expert hit an obstacle with Scratch, we would all work on the problem together. If the project manager couldn't make it to one of the meetings, another group member took notes and scheduled the next meeting.

After the desk critique with Martin, Martin was concerned that we were just "telling" a story. He questioned our motivation in choosing Grace Hooper as our counter-story and challenged us to think deeper. After much discussion, the group really focused in on using Grace's story to inspire and challenge young girls into STEM fields. As with any teaching tool, it is critical to provide context to the game. We felt strongly that this could be a launching off point for students, especially girls, to create their own games or do their own research into concealed stories in technology. Futhermore, just the fact of seeing that a woman is the reason we are coding in English instead of 1's and 0's is pretty incredible. Our hope was that young girls could see that what we are doing with computers today wasn't just because of the Bill Gates and the Steve Jobs of the world, but because of the Grace Hoppers of the world, as well.

At the first in-class peer critique, each member of the group had to present the team' idea. One of the big feedback items we all received was to simplify the word choices for our unscramble. We had initially chosen very specific words from Grace's life, losing site of the fact that our primary audience was elementary students. For example, initially one of the unscrambles was "Waves", which was the women's branch of the Navy that Grace joined. Elementary students would likely not be familiar with Waves, so we opted to change the unscramble answer to "Navy, making the game more accessible for young students.

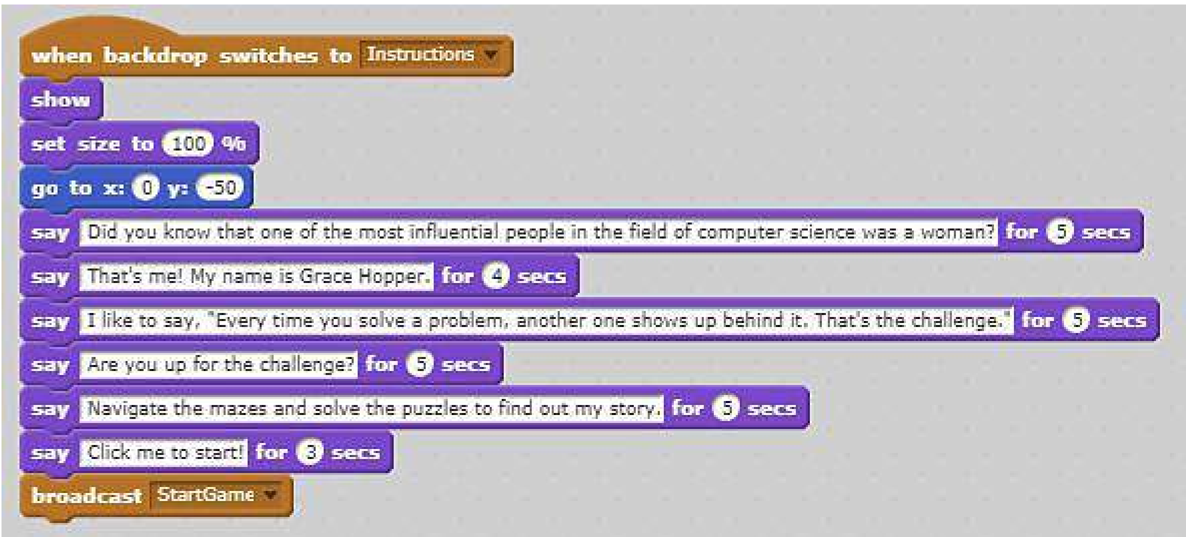
After the second class peer critique, we changed the coding to slow down the text throughout the game, giving players more time to read the words. Again, we felt this was a positive change to our prototype making it more accessible overall.

The Process: Computational Thinking

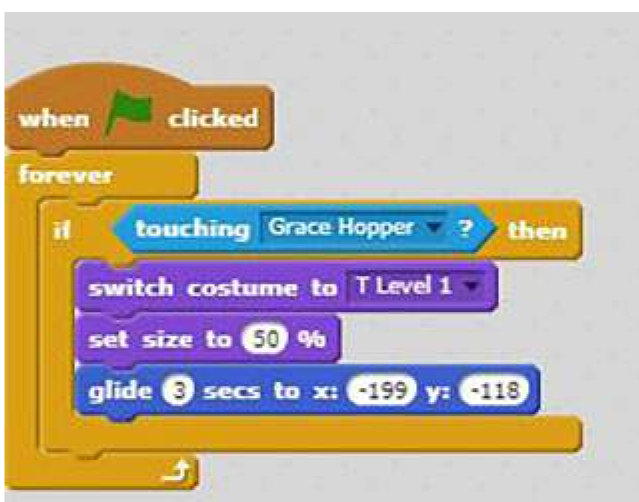
We decided to use programming that we learned in class to not over-complicate the assignment. Using the push buttons that we set up in Chapter 1 of the guidebook allowed us to focus on finding an innovative and fun way to tell Grace's story and engage our young audience. Early on, we decided not to use the touch and motion sensors because they did not add to the story we were trying to tell. We would just be including them for the sake of adding more "cool" tech stuff.

By the time we created our final iteration of the game, we had incorporated all five of the computational thinking terms into the programming. Below are examples of the how we incorporated the five elements into our final product.

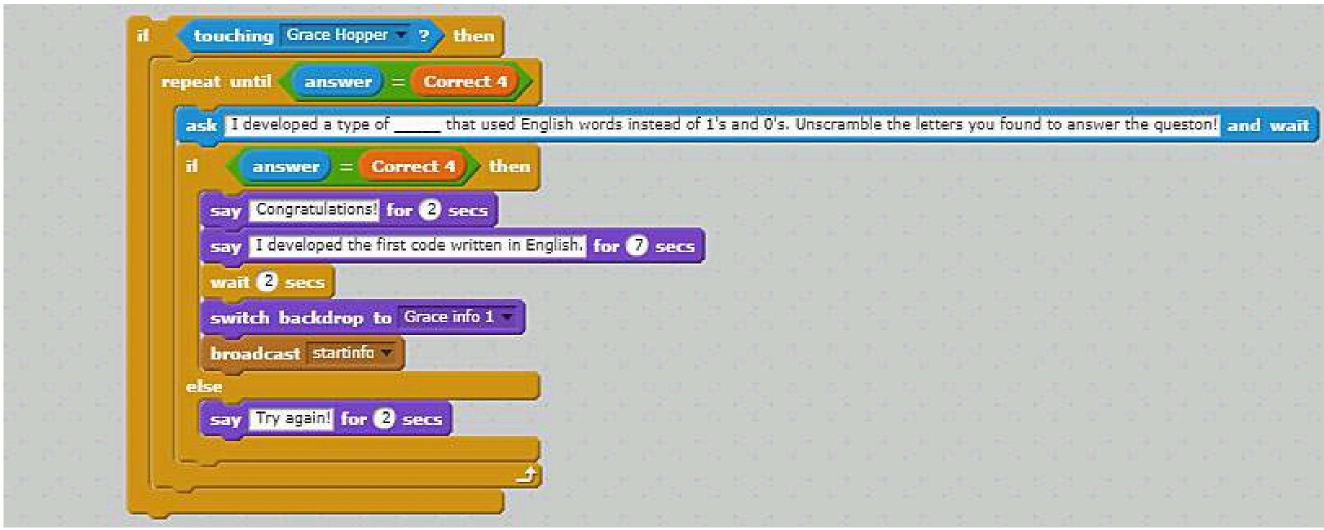
Sequence: In order to make our game user-friendly, we wanted our game to be easy for students to understand how to play. We wanted even the youngest of players to be able to play the game without a teacher or adult having to explain it. The game, quite simply, had to explain itself. We accomplished this goal by using sequencing to give players a set of "instructions" when they start the game. When the backdrop switches to "Instructions," the Grace sprite appears then goes to a specific location and says a series of statements, including basic directions, that start the game.



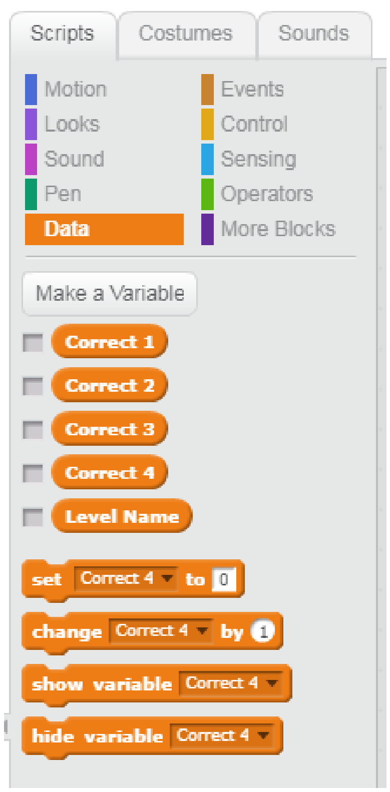
Iteration: We incorporated iteration into our programming in order to make the balloons react to Grace touching them, transform to a clue letter and fly up to the corner of the screen. Specifically, we used the forever block so that these actions would continue again and again.



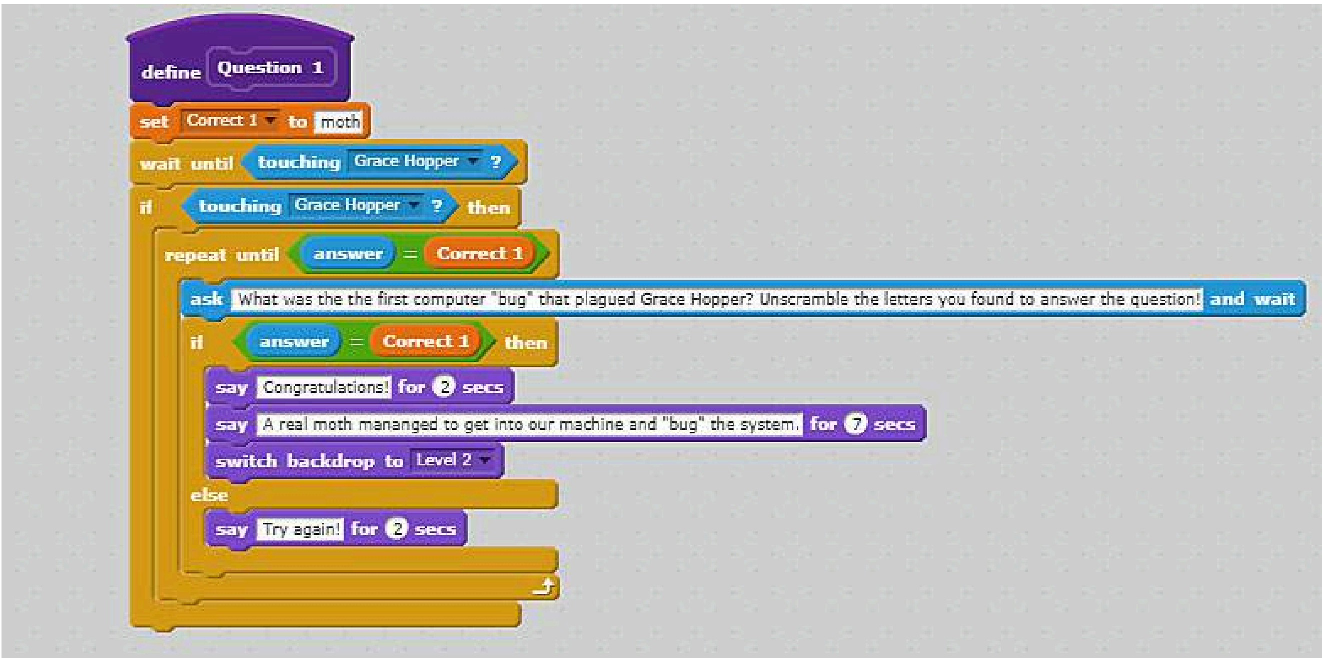
Conditionals: We wanted players to be able to progress through the different levels of the game. Creating conditionals allowed us to tell the computer that if the player gets the answer right, then open up the next level. If the player gets the answer wrong, then ask them to try again. We accomplished this leveling up function by using the if/then blocks in Scratch.



Variables: Using data blocks, we created correct answer variables. This allowed players to move forward through the game based on those variables. Variables were essential in creating a multi-level game platform.



Modularization: We knew that we wanted players to have to answer questions about Grace, so that they would gain knowledge about her while playing the game. By using the more blocks function in scratch, we were able to define questions for each game level.



Conclusion: Continued Mutual Shaping

The work we did in this class continues to shape and influence each member of our team as well as our instructor and future users of this guidebook. One member of our team has plans to use this project as the foundation of her work as an elementary school librarian. One of her goals as a practitioner is to inspire elementary students to design, create, and challenge their thinking by having them work in groups to research a lesser known individual from history and tell their story through a unique game. Furthermore, this project, which is based on an earlier version of this guidebook, shaped the version of the guidebook that you are reading. This iteration of the guidebook will continue to shape future information scientists, and, in turn, they will continue to shape and refine the contents of this guidebook.

Rainbow Unit: Networks Big and Small

At the Mary Brown Center in East St. Louis, Illinois, community members and Lessie Bates Davis Neighborhood House staff, along with School of Information Science students, setup a new public computing center created using participatory and evidence-based design principles.

Table of Contents

Rainbow Unit Overview

As we went through the Orange and Blue Units, we dove into essential electronics technology and computer hardware and software concepts that make up our **sociotechnical** artifacts and systems. We've also networked together separate sociotechnical microcontroller and microcomputer artifacts to create our first networked information system. As we move into the Rainbow Unit, it's time to bring together the technical, cognitive, socio-emotional, and critical sociotechnical skill sets, frameworks, and standards we've been advancing within our community of practice into a more holistic understanding of networked information systems.

For over a century, professionals in a range of information sciences have sought to use the information and communications technologies of the day to provide liberating directions for social justice outcomes for individuals, communities, and society. As the twentieth century came to an end and we entered the twenty-first, new digital networked information systems emerged, and the information sciences added to their repertoire digital resource services for libraries, digital humanities, social informatics, community networking, community informatics, digital literacies, citizen science, classification on the web, information infrastructures for science, human-centered data science, human-AI interaction, crisis informatics, trauma and memory studies, digital health, health informatics, critical data studies, critical archival studies, learning analytics, data analytics, biomedical informatics, everyday information behavior, data science storytelling, linked data, knowledge organization, management of data, information, and knowledge, data curation, data visualization, data and text mining, coupling cognitive systems, natural language processing, mathematical optimization, cultural theory, computational semantics, game studies, collective leadership, business information, innovation culture leadership, accessible computing, and information law, policy, and inequality. This is but a partial list as the information sciences continues to explore the data, information, knowledge, and wisdom (DIKW) realm in full as a broadly interdisciplinary space that also uses multidisciplinary research with many other sciences, from engineering to political science to journalism to education to African American and African studies to women and gender studies and beyond.

There has been a significant tension, though, between user-centered practices that too easily result in users being identified as entities completing a task, and a more human-centered, design justice focus that recognizes the ever-present relationship between design and power, a relationship that is often used to benefit some over others through the social shaping of information technology artifacts and systems. And so, here too, it is critical to bring in the voices of the marginalized and oppressed, whose community cultural wealth and indigenous ways of knowing have been excluded, hidden, and socially and physically destroyed. Social justice storytelling has been an important start, which this unit works to expand to ensure that we continue to provide liberating directions for social justice outcomes for individuals, communities, and society.

In the Rainbow Unit, each session will more closely bring together the technical and the social within the two chapters of the session to introduce the sociotechnical Internet of Things and smart devices (session one); the global and community Internet networks (session two); the expansion of historic, and introduction of new, divides through digitization (session three); and the hope of recovering community through design justice (session four). The first three sessions will continue to explore core concepts and skills through a more structured set of hands-on technical activities. Networking of data and information can happen over a range of different networks, as we'll explore. We'll integrate data and information from the breadboard electronics and Circuit Playground Express with the Raspberry Pi in new and creative ways. Many of these could instead be integrated with your laptops, or with local or remote server farms. But in using the Raspberry Pi to demystify digital networked information systems, we will hopefully also demystify the sociotechnical networked information systems in ways that open the potential for more just pathways in your future professional endeavors as an information scientist. This will especially be brought forward in session four, as we explore issues of design for justice in the information sciences.

Today, networked information systems bring together the rich, lived, analog realm that has evolved over millennia with the digital technologies that have developed over the last decade. These continue to include the still-relevant Universal Asynchronous Receive Transmit (UART) serial communications protocol we've used in previous exercises of the textbook, and related protocols like Inter-Integrated Circuit (I^2C) and Serial Peripheral Interface (SPI). In addition, as dialup bulletin-board systems and commercial online services entered the scene in the 1980s, central servers began to provide information and communication services to larger communities. In some cases, these systems came together to form larger internetworks using more open models. One serial communication protocol that emerged at this time was the peer-to-peer protocol. Peer-to-peer (P2P) applications are not just used to facilitate communication between two peers passing data back and forth, such as our Circuit Playground Express microcontroller and our Raspberry Pi microcomputer. Some peer-to-peer applications are set up for home file and multimedia sharing, while others make use of the Internet to extend peer-to-peer connections globally, often using a "store-and-forward" principle in which data is first stored at a peer hop in its journey before being forwarded to the next peer. BitTorrent is a widely used peer-to-peer file-sharing software, while Bitcoin is a well-known peer-to-peer cryptocurrency, with many emerging mobile peer-to-peer payment services coming to market using Bitcoin's underlying blockchain technology.

Another space in which peer-to-peer protocols have emerged is within the "Internet of Things" space, and this is where we'll start in session one of the Rainbow Unit as we also gain a first view of the underlying Internet Protocol (IP). But as we'll also explore within this session, too often, Internet of Things devices are actually simplified programmed devices that use a centralized command and control. This shaping of the sociotechnical "IoT" devices reduces or removes design and the advancement of power that should be centered within the individual and with the community of which the individual is part. The constant tension regarding whose design has been used, and whose power has influenced these decisions, have moved forward within the digital realm on a path that centers around extractive economic and societal dominance, something we'll especially explore in session two regarding the Transmission Control Protocol (TCP)/Internet Protocol.

The majority of Internet services used today, including many "smart" sensors, controllers, and communications devices, still function using the client-server architecture. While in peer-to-peer architecture, each device on an IP network is equally privileged so that tasks and workloads can be distributed among peers, within a client-server architecture, specific nodes play a centralized role as servers. The server provides services and resources upon request from individual client nodes, often through front-end applications that interact with the human users of the application. The open standards and protocols of the Internet are systems agnostic, and thus peer-to-peer and client-server technologies can work well together using a range of different internetworking platforms. However, today, we are shaped in a wide range of ways to consider the server as the essential information and communication thing floating up there in the cloud, from which we are the fortunate recipient and through which we can be transformed. But servers are always local devices, and the cloud is often just a bank of servers within local networks owned by corporations. It remains that the local networks on which servers are housed can be anywhere. And while there is considerable value in locating our core email, video streaming, and social media services within a Microsoft or Amazon or Google cloud located within regional server banks, there remains significant value in more localized community networks and decentralized Internet of Things devices, a concept we'll explore throughout sessions one and two as we configure our Raspberry Pi as a local "web server" using two different strategies. Together, sessions one and two are works seeking to guide us beyond the dominant, neoliberal "American dream" of digital technological determinism, radical individualism, and supply-side, free-market capitalism that has especially emerged since the 1990s.

As we move into session three of the Rainbow Unit, we'll explore how the technological determinism that embeds our digital "thing-orientation" works both intentionally and unintentionally to extend the historic racial and cultural divides into the digital realm. A name that has stuck in this regard is the "digital divide," which is especially framed as "a national crisis of competitiveness, defined as a human capital deficit and resolved through public-private partnerships for access extensions."[1] The breadth of socially driven divides is instead turned exclusively into a technical one. The challenge is to shift focus towards both digital inclusion and also digital equity, in recognition that today's digital technology "is a necessary tool for civic and cultural participation, employment, lifelong learning, and access to essential services."[2]

Our digital information and communication technologies arise through a range of design principles and processes that merge social and technical considerations. Within digital inclusion, affordable, robust Internet service provision is an essential first element, but it is something that has especially been influenced through monopoly power within the United States, something we'll especially highlight as we move further into session three. But internetworking of information systems does not need to be this way, something we'll (re)consider throughout sessions three and four of the Rainbow Unit, as each learner is encouraged to reflect back and plan forward ways that they can solidify their introductory understanding of the core learning objectives through a person-centered, networked information systems adventure. We'll also work to explore the ways in which a design justice framework and its underlying emergent strategy principles can be used to advance new pathways for social justice practice within the information sciences. The open protocols and standards of the Internet Protocol continue to give us considerable freedom of choice. As we conclude the Rainbow Unit, the adventures are meant to provide a means to test execution of choice, especially as exercised at the local premises, neighborhood, and community levels. We'll learn to track down network performance issues and rapidly move up the support tier levels to ensure that our Internet service providers give us what we pay for with regard to Internet provision. And as we'll see in the Rainbow Unit, not only can we build our own local networks and even community networks, we can also build our own web servers, and from there could expand to build our own database and communications servers, if we so choose. But there are communities and even regions that are stepping up their works to build campus and community networks at larger scales, pushing Internet service providers to play more by community rules, and not just by corporate rules.

This is a long journey, but hopefully as we work through the Rainbow Unit, we'll begin to demystify the clouds that keep networked information systems within a mist, thereby opening up a visioning of the revolution of values that are person-oriented, rather than thing-oriented, as King called us to do over forty years ago.

1. Daniel Greene, "Discovering the Divide: Technology and Poverty in the New Economy," *International Journal of Communication* 10 (2016): 1226, https://ijoc.org/index.php/ijoc/article/view/3969.
2. National Digital Inclusion Alliance, "Definitions," accessed July 6, 2020. https://www.digitalinclusion.org/definitions/.

Rainbow Unit Overview

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with the option to export your responses, use the online version.]

Key Learning Outcomes Self-Check

The key learning outcomes objectives for the **Rainbow Unit** include the following. As you view these, take a moment to ***rate 1-4*** these items based on where you think you are in your skills development on each of these items, with additional notes as helpful. Keep these at your side as you work through the Unit to facilitate guided inquiry.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough
4: Yet

Technical Skills

You can (rate 1-4):

- Understand the seven layers of the Open System Interconnection (OSI) model and their practical uses in networking.
- Understand Local Area Network components, and how to design, troubleshoot, and maintain those components.
- Understand Wide Area Networks like the Internet, how to select between Internet Service Providers, and how to join the Internet by connecting to an Internet Service Provider.
- Understand the Internet Protocol, IP addresses, IP names, and how they apply to nodes on your LAN.
- Understand the peer-to-peer and client/server architecture and open protocols at a basic level.
- Understand Infrastructure-, Platform-, and Software-as-a-Service as an operational definition of cloud computing.
- Distinguish between locally controlled Internet devices using open hardware, software, data, and/or protocols, and centrally controlled Internet devices using closed hardware,

software, data, and/or protocols.

- Use multiple methods to setup a Hypertext Transport Protocol (HTTP) server.
- Use serial communication to collect, analyze, store, and share data from microcontrollers and microcomputers, and make them accessible across a LAN and from the Internet.

Cognitive Skills

You can (rate 1-4):

- Generate a wide variety of ways to accomplish project and logically analyze divergent pathways to assess potential impacts.
- Logically analyze and organize project tasks in ways that allow use of digital tools to help accomplish them.
- Apply different critique evaluation methods periodically throughout the design process to advance effective human-centered design.
- Turn personal and community strengths, opportunities, and aspirations into project ideas.

Socio-Emotional Skills

You can (rate 1-4):

- Empathize with individuals and social groups across a wide spectrum of social and cultural contexts.
- Progress towards implementation of stages four, five, and six within the Information Search Process as applied to a design.

Critical Sociotechnical Skills

You can (rate 1-4):

- Recognize the historical development of the Internet within the United States and internationally, its further socio-cultural, -economic, and -political development through commissions, associations, think tanks, non- and for-profit groups, and the influences this has had on the design of its sociotechnical artifacts and systems.
- Integrate a basic technical understanding of Internet technologies with a critical sociotechnical perspective to advocate for more just Internet policies and implementations.
- Critically, strategically, and communally use a design thinking mindset to unpack ill-defined problems using critical abductive reasoning within a broader design justice approach.

Your Personal Learning Outcome Objectives

Add your own personal learning outcome objectives

Rate Your Personal Learning Outcome Objectives

Take a moment to create a ***1-3 rating*** of your personal learning outcome objectives. Add additional notes as helpful to keep further specific reflections on these specific skills development.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough

Goals	Rating
Your first goal...	

1A: Programmable Electronics, Smart Technology, and the Internet of Things

Background Knowledge Probe

1. Create a spreadsheet with two columns and at least 20 rows.
2. Take a few minutes to walk around your home, listing in the right column all the electronic devices you use on a regular basis.
3. Review your list of devices and note in the left column which category you think each device belongs:
    A. Non-programmed circuit
        - Example from Orange Unit: 5 Volt resistor -> momentary switch -> light -> ground

    B. Programmable circuit(s), no connectivity to other devices
        - Example from Orange Unit: Using a momentary switch and Python on Raspberry Pi to turn LEDs on or off
        - Example from Blue Unit: Using a momentary switch and MakeCode on Circuit Playground Express to turn LEDs on or off

    C. Programmable circuit(s), data connectivity with one or more local devices only
        - Example from Blue Unit: Circuit Playground Express capacitive touch sensors informing which counterstory to play on the Raspberry Pi

    D. Programmable circuit(s), Internet connectivity providing remote information and/or control
        - Examples from Rainbow Unit to follow

- Which of the items on your list are analog electronics? Which are digital electronics?

- How many more devices would be added if the list were to include non-electronic, mechanical devices?

The Mechanical World

I wake up a little before the sun rises. I stand up and stretch, then walk out of the sleeping quarters and over to the outhouse with composting toilet. It's light enough out now so my kerosene lantern isn't needed, although it's still a little dark in the outhouse. I start my campfire and 30 minutes later have some water boiling eggs and making some tea. The morning is first spent feeding and providing clean water to the chickens and washing clothes in the nearby stream before putting them out to dry. Then I take some time weeding in the garden while nibbling on the few first strawberries and snow peas of the season and picking the cold-tolerant spinach and kale greens of the spring garden. Getting out of the warming midday sun, I head into the woods to forage for mushrooms, ramps, dandelions, and nettles. Check on the clothes drying, catch a fish from the stream, make dinner over the fire or inside over the woodburning stove, turn on the lantern to read for a bit, and back to bed I go.

The romance of time in a mechanical world!

While I have and continue to live with a wide range of daily-use electronic devices for lighting, heating and air conditioning, cooking and cleaning, and the other staples of modern life, I come from a long family line who also keep a hold on the mechanical side of life in a range of ways. And I have a family who continues this tradition in our own ways. While we don't live off-grid in a tiny house producing and preserving all our own food and resources, we continually explore and periodically test ideas from local rural and urban farmers and foragers.

At the same time, we're a family who uses the Internet to explore resources such as Mother Earth News and Bryce Langston and Rasa Pescud's Living Big in a Tiny House YouTube Channel to expand our understanding of the balance between mechanical and electronic, off-grid and on-grid, physical community and the larger online communities of practice. I'm someone who writes code, designs and builds electronics, has facilitated setup of a number of local networks and helped explore a couple community networks, and consulted on Internet choices, while also using a wide range of electronic and mechanical woodworking tools.

I live in a family and in communities that flex between the both/and of the mechanical and electronic, of the analog and digital, of the in-person and online.

From Mechanical to Programmable to "Smart"

For purposes of this discussion, we'll consider mechanical devices as those involving pure physical processing. A common style of toilet within the United States flushes because we push down on a physical component that adjusts other physical components, thereby allowing water to leave a tank and flow through the toilet bowl. The water flows in an effective manner to remove liquid and solid waste through the physical porcelain S-shaped siphon, designed to remove wastes while physically protecting sewer gas from venting back into the room after the flush cycle has completed. Together, this represents a classic **mechanical technology**.

While we continue to make use of a range of mechanical devices as part of our daily lives, we have increasingly made use of electronic devices as the latter half of the 19th century experienced the advancement of electrical power generation and the eventual emergence of electrical engineering. We've moved from the building of wax candles or gas-soaked wicks in lanterns as a source of lighting to the incandescent light bulb providing an electrical device in which the flow of electricity can pass through a wire filament embedded within glass. More recently, we've seen the transition from the incandescent light bulb to light-emitting diode lamps, or LEDs, that use significantly less energy to achieve equal brightness and with greater longevity. Each of these represent **simple non-programmed circuits**.

But it many cases it is possible to expand the circuit to include additional components. The simple LED circuit can be constructed so as to include the necessary resistors, capacitors, transistors, and photocells such that it automatically dims or brightens depending on the amount of light currently within the area. This can be designed as a non-programmed circuit, but often is incorporated into a microcontroller or microprocessor to control the brightness of the LED based on certain levels of light as sensed by the photocell. It is now a **programmable circuit**.

The programmable circuits can become increasingly complex, as when an LED only comes on at the appropriate brightness based on current light levels when movement is detected via an integrated motion sensor. It may include built-in switches or touch sensors that can be used to determine whether to make use of the dimming and/or motion sensing functions of the program. Depending on the technology being used, it's possible that as we observe our own or others' uses of the technology, our behaviors are shaped by the choices being made in that use. And it may be that we begin shaping our choices to further shape behavior. The electronic devices continue on as **sociotechnical artifacts**. Over time, the programs running some of our electronic devices were further expanded to incorporate code that altered the actions of other segments of the code based on previous data collection by that software. To begin, **machine learning** code can be combined with a mathematical model using initial "training data" to create predictive decisions. Over time, machine learning **algorithms** are set up such that the software learns from new data in ways that help set up more precise predictions and corresponding actions.

The predictive analytics of machine learning has emerged from the broader exploration known as **artificial intelligence (AI)**. Programming languages such as LISP started emerging in the late 1950s, helping to advance the field of AI that was also emerging at the time. As with other programming languages, AI makes active use of "if-then" Boolean logic statements, or increasingly newer Fuzzy logic with a continuum between 0 and 1, to create a knowledge base that is then used by an inference engine to draw deductions. Autonomous devices with software facilitating perception of its environment and execution of actions that maximize chances of achieving target goals are formally referred to as **intelligent agents** within AI, and popularly as **smart devices or technologies** today.

> It might be helpful to more quickly review the Blue Unit chapter 2A: The Methodological Landscape to more fully consider the Positivist Paradigm within which the **deductive logic** of AI. While essential, work within the Positivist Paradigms, including machine learning and AI, are insufficient to investigate the full spectrum of data, information, knowledge, and wisdom within the information sciences. It is important to also use **inductive reasoning** to develop interpretive understandings of complex social phenomena, and a **dialectical approach** to knowledge that enables people to see hidden forms of control, domination and oppression. Only in this way is it possible to enable people, and especially those in the margins, to seek change and reform existing conditions and social order.

In their 1997 article, "An Overview of Smart Technology," Goddard, Kemp, and Lane noted that the term "smart" for a diverse range of materials, structures, systems and technologies originated in the early 1980s, mainly via researchers working in the United States and funded through defense budgets. It was the reports of "smart" bombs and other "smart" munitions of the Gulf war in newspapers and popular science journals that brought about public awareness. From this emerged the fashionable use of "smart" technologies as part of industrial applications well beyond the original aerospace and defense uses that began in the early 1980s. Thus, while "intelligent agents" were already being developed at about the same time as part of various dedicated research groups, these academics "had been working on 'smart' technologies for several years without realising it!"[1]

Intelligent or smart systems include: 1) sensors used to collect associated information on environment, condition, or operating history; 2) a trainable control algorithm to read sensor data, draw inferences on that data, and alter system characteristics based on these inferences; 3) control hardware interconnecting arrays of sensors, a microprocessor running the algorithmic code, and actuators; 4) the actuators, or mechanical devices that can move or control something and thereby implement the change in configuration based on the control algorithm; and 5) structural members allowing the sociotechnical artifact or system of artifacts to perform its primary function.

> For a deeper exploration of machine learning and artificial intelligence, take 11 minutes to watch the video "Machine Learning & Artificial Intelligence" from Carrie Anne Philbin's Crash Course Computer Science series hosted by PBS Digital Studios.

Bringing this together, each of these terms and concepts, from circuits that are mechanical or programmable, to sociotechnical artifacts and systems, to artificial intelligence and intelligent agents, and to smart devices and technologies is shaped by and shapes individual people, scientific disciplines,

1. N. D. R. Goddard, R. M. J. Kemp, and R. Lane, "An Overview of Smart Technology," *Packaging Technology and Science* 10, no. 3 (1997): 129. https://doi.org/10.1002/(SICI)1099-1522(19970501/30)10:3%3C129::AID-PTS393%3E3.0.CO;2-C.

research funding resources and their governmental, corporate, and non-profit funders, journalists and popular media, and many other social, cultural, political, and economic influencers. As the previous sentence is complex, so too is each individual word within it. But only in acknowledging the complexity is it possible to meaningfully enter into true works of liberation of all direct and indirect stakeholders seeking advancement of their individual and communal valued beings and doings.

Bringing Together Sociotechnical Artifacts into an Internet of Things

As we've seen in the Blue Unit, it is possible to bring together microcontroller-based programmed sensors with microcomputer-based processing software using data communications cables such as our UART serial cable. This can be very effective for connecting the microcontroller to the microcomputer as a single device. But a difficulty arises as control algorithms work to facilitate **interdevice internetworking**—that is, to facilitate two or more devices working with each other, whether or not interdevice internetworking code incorporates artificial intelligence functions and thus formally serves as intelligent agents. From farming to the home to the office cubicle to the library shelves, a growing number of independent, small microcontroller/microcomputer sociotechnical artifacts are capturing and routing volumes of data between devices. The code on these devices, while much smaller than centralized cloud computing devices and services, can be very effective at the rapid collection and analysis of data.

> If we had computers that knew everything there was to know about things—using data they gathered without any help from us—we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best.
>
> Kevin Ashton, "That 'Internet of Things' Thing"[2]

It is here that the open protocols and processes of the Internet serve an essential role, leading researchers such as Neil Gershenfeld and others to formally advance an Internet of Things (IoT), allowing myriad devices to intercommunicate and interoperate.[3] The initial IoT concept could incorporate, but did not require, use of broadband connectivity to support much of its functionality. Key is the recognition that the circuits and software, even including a Web server, can be brought together on very low-cost microcomputers. While our educational-focused toolkit includes a $35 USD Raspberry Pi and $25 USD Circuit Playground Express, along with a range of other materials, it's possible to build various everyday devices for as little as $1 USD. Alarmingly, many of our smart devices today are really not intelligent agents, but rather basic data gathering devices that use the Internet to communicate with proprietary controllers located in the "Cloud," storing (and often taking ownership of) data

2. Kevin Ashton, "That 'Internet of Things' Thing," *RFID Journal*, June 22, 2009, https://www.rfidjournal.com/that-internet-of-things-thing.
3. Neil Gershenfeld, Raffi Krikorian, and Danny Cohen, "The Internet of Things," *Scientific American*, October 2004, https://www.scientificamerican.com/article/the-internet-of-things/.

and the information emerging from this data. Management and use of data and information of sensors, which include both known and unknown input from consumers in many cases, is determined by the commercial providers, and then returned to the devices to be implemented. Devices are indirectly interconnected via cloud platforms and software, divorced from the interdevice interconnectivity envisioned within an Internet of Things. Such conflicting meanings also appear in terms such as "smart grid," "smart homes," and "smart cities."

Neil Gershenfeld provides an excellent introduction to the opportunities and challenges of this sociotechnical space within this four-minute segment of a World Sciences Festival panel, on which one of the "fathers of the Internet," Vint Cerf, also sat, and who joins in some humorous bantering at the start. [Watch the video "The Internet of Things" with Neil Gershenfeld online.]

For Neil Gershenfeld and the team at the Center for Bits and Atoms, this has meant bringing together the computer and physical sciences to explore how to turn data into things and things into data, by creating location fabrication facilities, Fab Labs, and a global network for training people and sharing knowledge, a Fab Academy, thereby digitizing fabrication in the same way that we digitized communication over the last two decades. In so doing, the purpose is to shift the focus from plumbing and power to safety and convenience, from pills to proper medication. To do this, it requires shifting focus to the local edges rather than a centralized cloud, exploring the boundaries between the digital and physical worlds.[4] Over the last five decades, the digital revolutions of computing and communication have transformed the world by enabling unprecedented productivity, and have catalyzed remarkable changes in everyday life while also providing opportunities for enormous wealth generation. But with the exciting possibilities for some people have come the exclusion and downward impacts for others. As Neil Gershenfeld, Alan Gershenfeld, and Joel Cutcher-Gershenfeld note in their 2017 book *Designing Reality: How to Survive and Thrive in the Third Digital Revolution:*

> The negative aspects of the first two digital revolutions are not simply accidents. Nor were they driven by some unseen hand. Decisions made (and not made) and priorities set (and not set) early on, as the technologies were being developed and introduced to the market, have had lasting effects. We built breakthrough digital communication capabilities, but we failed to build in cultural norms, feedback loops, and algorithms that could have reinforced civil discourse. We created incredibly efficient new models of digital commerce, but have also introduced new threats to privacy and security. We value the advances made possible with digital automation, even as we struggle with impacts of lost jobs due to technology. (p. 5)

For the Gershenfelds and others within the community of the Fab Lab movement, this third, fabrication revolution brings together the **virtual world of bits** and the **physical world of atoms** in innovative ways in which we can find another chance to shape our societies. Digital fabrication provides

4. Neil Gershenfeld and Jim Euchner, "Atoms and Bits: Rethinking Manufacturing," *Research-Technology Management* 58, no. 5 (September 2015): 16–23. https://doi.org/10.5437/08956308X5805003.

new possibilities for enabling self-sufficient local communities beyond dependency on the centralized cloud of today. In so doing, it also opens up potentials for global sustainability for everyone, rather than just the fortunate few.

Gartner is a global research and advisory firm that has developed the Gartner Hype Cycle model to inform its customers, senior leaders across the enterprise, regarding technology hype versus commercial viability. This can be another helpful way to more broadly recognize the ongoing shaping and reshaping of sociotechnical artifacts over an extended period. Initially, innovation of emerging technologies leads to early proof-of-concept stories and increasing publicity of a product. However, as this often still requires further research and development, increasing wide use of the product can lead to a "Peak of Inflated Expectations" as success stories begin to be accompanied by increasing stories of failure. In many cases, interest continues to wane until the product reaches a "Trough of Disillusionment." Only through ongoing development and testing do we reach a "Slope of Enlightenment" leading to a "Plateau of Productivity." View the Gartner Top 10 Strategic Technology Trends for 2020 video to see an example of this methodology in practice, with indications that we are slowly moving from a BIT-NET of Things, as noted by Gershenfeld, to an increasing plateau of productivity that includes both the edge and the cloud, each in its own way.

Also compare the highlights within this 2020 Gartner Top 10 Strategic Technology Trends to their Press Release introducing the Top 10 Strategic Technology Trends for 2023, which adds to their optimize, scale, and pioneer themes that of sustainability.

> "However, in 2023, delivering technology will not be enough. These themes are impacted by environmental, social and governance (ESG) expectations and regulations, which translate into the shared responsibility to apply sustainable technologies. Every technology investment will need to be set off against its impact on the environment, keeping future generations in mind. 'Sustainable by default' as an objective requires sustainable technology," said David Groombridge, Distinguished VP Analyst at Gartner.

To consider the possibilities and pitfalls of the Internet of Things through a critical lens is to recognize the many seen, unseen, and unforeseen dangers, as Richard Milner brings forward in "Race, Culture, and Researcher Positionality."[5] Furthermore, it is to question "Whose Culture Has Capital?"[6] as asked by Tara Yosso, a question that we explored at the end of the Blue Unit in chapter 4A: Sharing our Counterstories. It is essential that we continually bring not only our positivist and interpretive meta-theoretical assumptions, but also our critical meta-theoretical assumptions into the methodological landscape. For the alternative is to use the normative investigations based on deductive reasoning in

5. Milner, H. Richard. "Race, Culture, and Researcher Positionality: Working Through Dangers Seen, Unseen, and Unforeseen." Educational Researcher 36, no. 7 (October 2007): 388–400. https://doi.org/10.3102/0013189X07309471.
6. Tara J. Yosso, "Whose Culture Has Capital? A Critical Race Theory Discussion of Community Cultural Wealth," Race Ethnicity and Education 8, no. 1 (March 2005): 75. https://doi.org/10.1080/1361332052000341006.

its evaluations of our sociotechnical artifacts, such as the Turing test many have adopted to determine whether a computer running artificial intelligence software can “think.”[7]

Lesson Plan

Ultimately, choice between cloud services, local servers and server farms, or the Internet of Things and smart devices does not require either “this or that” decisions. In many cases, strategic decision making, design, and implementation may reveal that a combination of several different networked information systems is the ideal practical path to facilitate functional diversity in achieving valued beings and doings for the broadest range of stakeholders. The first objective of this session is therefore to introduce us to the concepts and terms, the technical components, and the sociotechnical codifications underlying these networked information artifacts and systems, helping advance the existence and sense of choice, as well as our ability to more effectively execute and achieve that choice to advance our valued beings and doings. While these activities can be done using many types of computers and server farms, we’ll make use of the Raspberry Pi computer throughout to especially advance a greater sense of the choices we do have but often are unaware of having given we are often directed into consumptive patterns of a specific product line. Indeed, some of these activities throughout this book using the Raspberry Pi have evolved from earlier works in communities for whom the sharing of data and information needed to be carefully controlled in acknowledgement that access beyond a room or a carefully controlled local area network could put an individual or groups life at risk. Low-cost Linux-based general-purpose computing devices can provide freedom and safety within environments in which individual or societal dangers otherwise would create barriers to liberating actions.

A second objective as we decodify these materials of this session then is to recognize that the data and information being collected is not just of things. It is often data and information about human and more-than-human persons and their activities. Individually and collectively, people have shaped the devices and systems which collect data and facilitate information processing for a range of objectives. In this process, we gather data on people’s actions and behaviors within various social and environmental contexts which we also are observing. How we reflect upon, investigate, and act upon this determines the form of power that is used: power within ourselves to advance our power over others, or power within others to advance power within all, to every extent possible. How the artifacts are shaped—and how we are shaped by them —also influences how we go on to shape things and persons alike. Considering how we can use our newly developing social justice storytelling skills within a data, information, knowledge, and wisdom pyramid to incorporate diverse wisdom in ways that advance knowledge in support of self-sufficient local communities and global sustainability for everyone is essential as we move forward within the information sciences.

7. Britannica, The Editors of Encyclopaedia. “Turing test”. Encyclopedia Britannica, 13 Mar. 2023, https://www.britannica.com/technology/Turing-test. Accessed 13 April 2023.

As a reminder, the Lesson Plan found in the Introduction of the book, think of our travel through the many session of this book as pre-season training activities within a sports metaphor. Each is meant to strengthen a different aspect of ourselves and ourselves in relation to others. As with pre-season training, by the end of the Rainbow Unit we will have not reached the finish line, but rather the end of pre-season training. In entering the Rainbow Unit, use these four sessions to further advance a clearer and stronger person-centered community inquiry mindset and to also push even further within your action-reflection collective leadership practices to develop stronger muscle memory so that these methods are an active part of your daily practices as an information science professional moving forward.

Essential Resources:

- Burgess, Matt. "What Is the Internet of Things? WIRED Explains." *Wired UK*, February 16, 2018. https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot.
- TED Institute. "Internet of Things: Transforming the routine." YouTube, November 15, 2016, https://www.youtube.com/watch?v=8qjf_VuGW7A.
- Stop Autonomous Weapons. "Slaughterbots." YouTube, November 12, 2017, https://www.youtube.com/watch?v=9CO6M2HsoIA. Note: Watch the following dystopian science-fiction arms-control short film with caution. This film was produced by the non-profit Future of Life Institute and Stuart Russell, who provides short remarks at the end. It first screened to the November 2017 United Nations Convention on Certain Conventional Weapons.
- *Full Frontal with Samantha Bee*. "Black Future Month." Produced by Halcyon Person with Adam Howard. TBS, March 20, 2019. https://www.youtube.com/watch?v=vJrHFBCa4MQ.

Additional Resources:

- Harwood, Trevor. "Internet of Things (IoT) History." Postscapes. Accessed July 16, 2020. https://www.postscapes.com/iot-history/.
- Feather, Katie. "Attack of the Internet of Things." *Science Friday*, October 28, 2016. https://www.sciencefriday.com/segments/attack-of-the-internet-of-things/.
- Gershenfeld, Neil, and JP Vasseur. "As Objects Go Online: The Promise (and Pitfalls) of the Internet of Things." *Foreign Affairs* 93, no. 2 (March 2014): 60–67. https://www.foreignaffairs.com/articles/2014-02-12/objects-go-online.
- Gershenfeld, Neil, and Jim Euchner. "Atoms and Bits: Rethinking Manufacturing." *Research-Technology Management* 58, no. 5 (September 2015): 16–23. https://doi.org/10.5437/08956308X5805003.
- Huckle, Steve, Rituparna Bhattacharya, Martin White, and Natalia Beloff. "Internet of

Things, Blockchain and Shared Economy Applications." *Procedia Computer Science* 98 (January 1, 2016): 461–66. https://doi.org/10.1016/j.procs.2016.09.074.

Key Technical Terms

- Terms associated with more complex **programmable circuits**, including **Algorithm**, **Artificial Intelligence (AI)**, **Machine Learning**, and **Intelligent Agents/Smart Devices**
- The major data transport layers of the **Open Systems Interconnection (OSI) Model**, including the Physical & Data Link Layers, the Network Layer, the Session & Transport Layers, and the Application & Presentation Layers
- The **Internet Protocol (IP)**, **Transmission Control Protocol (TCP)**, and **User Datagram Protocol (UDP)** standards that make up the core of the **TCP/IP suite**
- The **Hypertext Transfer Protocol (HTTP) Client-Server Architecture** and **Hypertext Markup Language (HTML)** that together with the **Uniform Resource Locator (URL)** make up the base components of the World Wide Web
- Key concepts such as the Internet of Things, BITNET of Things, and **Intelligent Agents/ Smart Devices**

Professional Journal Reflections:

1. Review your original list of devices from the opening Background Knowledge Probe. How does it need to be revised, if at all? Why or why not?
2. In what ways do you have greater control than you previously thought over your devices? Less control? Why?
3. How do we recodify the concepts, terms, and technologies of microcontrollers and microcomputers, of smart and intelligent devices, technologies, and systems, of the Internet of Things/Bitnet of Things, so as to advance a more just society for all, and especially for those marginalized and oppressed?

1B: Connecting Our Electronic ‘Thing’ to a Wider World

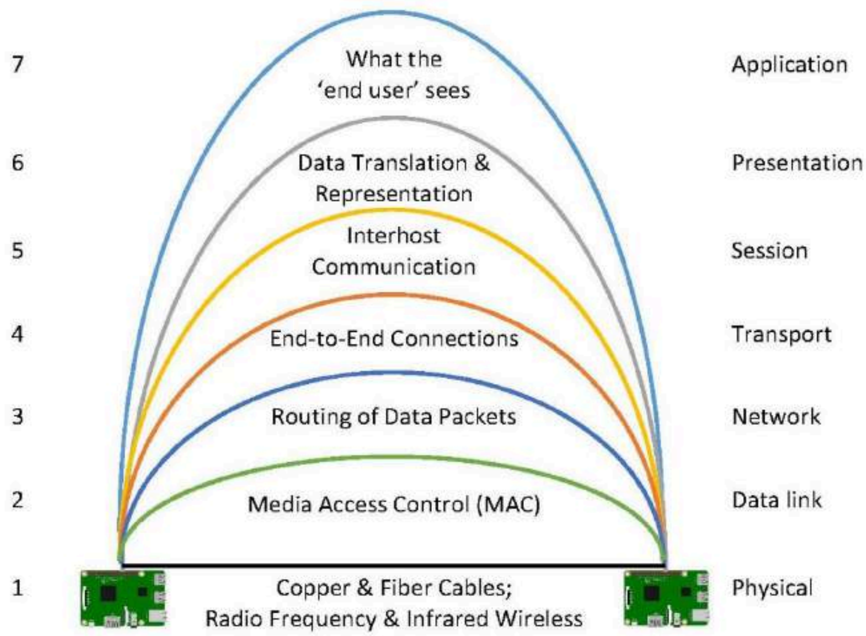
Technical Overview

As we saw at the end of the Orange and Blue Units, there are many ways to provide data communications between devices. In the Orange Unit, we learned more about the ports available on our computers, such as the **USB** and **HDMI** ports, that provide ways for us to connect external devices such as keyboards, mice, and monitors. And in the Blue Unit, we learned about serial data communications interfaces such as the **Universal Asynchronous Receive and Transmit (UART)** interface we’ve been using to transmit and receive data between the Raspberry Pi and the screen or PuTTY console of our laptop, and between the Circuit Playground Express and our Raspberry Pi. While technically a keyboard and a computer or a microcontroller and a microcomputer are two separate devices, via the software we use, they work as one unified system. They are our computer. They are our counterstory trumpet.

But underlying the concept of the Internet of Things—which today is often popularly referred to as Smart Devices coming together to create Smart Homes, Smart Grids, Smart Cities, and more—is an interconnection of electronics. A mechanical keyboard connected to a personal computer is not an IoT device. But a programmable keyboard that uses machine learning to predict changing contexts based on data from a range of local IoT devices and functionally evolve accordingly would be an IoT keyboard. It is to create an Internet of Things, whether or not this is connected to the physical Internet. There are actually many choices that can be made regarding this interconnectivity, only some of which make use of the Internet Protocol, something we’ll learn more about in session two of the Rainbow Unit.

Data Transport

As the evolution of computer networking continued into the 1970s, a number of different government- and corporate-sponsored programs led to the development of protocols, standards, and frameworks for data, information, and communication distribution between computers. This included CYCLADES in France, NPL network in the UK, and ARPANET in the United States. Through extended work by international bodies such as the International Organization for Standardization (ISO) and the Telecommunications Standardization Sector of the International Telecommunication Union (ITU-T), an Open Systems Interconnection (OSI) reference model was created and formally published in 1984 as both ISO 7498 and X.200. A key component of the OSI is a seven-layer model.



With regard to the Internet, these layers are practically combined into four groups, as follows:

Application & Presentation Layers	Provide the physical **client/server** (e.g. web browser client/**Hypertext Transfer Protocol, or HTTP**, server and **Structured Query Language, or SQL**, database server) or **peer-to-peer** (e.g., Bittorrent, Bitcoin) applications with which end users work & the data translation and representation used in support of those applications
Session & Transport Layers	Provide interhost communication and connections using protocols such as the **User Datagram Protocol (UDP)** and **Transmission Control Protocol (TCP)**. While TCP is the most commonly used, leading to the standard **TCP/IP suite** reference, UDP is used when speed is prioritized over occasional dropped packets, for instance in live streaming and online gaming contexts.
Network Layer	The core of the **Internet Protocol (IP)** used to provide end-to-end routing of data packets across the backbone
Physical & Data Link Layers	The unique identifiers associated with key hardware devices and the communications technology used to interconnect these devices

Limor Fried of Adafruit Industries partnered with DigiKey to create the web series "All the Internet of Things," highlighting essential tools for building Internet of Things devices. Traditional Internet-based systems and structures combine the seven layers into four groups as listed in the table above. But Internet of Things applications often focus on the Physical and Data Link, Network, and Transport layers, as is highlighted in the video.

To prepare for our own work in the Rainbow Unit, watch the first 13 and a half minutes of the video "All the Internet of Things – Episode 1 – Transports," Fried's review of Ethernet and Wi-Fi transport (read the full video transcript on Adafruit's website).

Data transfers between Internet of Things devices do not always (or even often) require human-computer interaction. The applications commonly used within Internet of Things devices do typically make use of common protocols such as HTTP and SQL that work within the Session and Presentation layers of the OSI model, but often can do so without the need for human-computer interfaces such as a web browser, or use the human-computer interface only occasionally for testing and configuration/programming purposes. In episode two of "All the Internet of Things," Fried introduces us to commonly used application protocols including HTTP, which we'll be using in the exercises below.

To get started, watch the first 4:10 of the video. For those wanting to do a deeper dive, you may consider watching the video "All the Internet of Things – Episode 2 – Protocols" through minute 11:04 to learn about the REST API (Application Programming Interface) that can be used with HTTP to create more extensive intercommunication between devices (read the full video transcript on Adafruit's website).

Towards Our Own Internet of Things

In the following three exercises, we will work with the key components used to create IoT devices.

1. First, we will return to the MakeCode Serial UART Communications Extension used with the Raspberry Pi Counterstory Digital Little Free Library exercises from session four of the Blue Unit. This time, we'll add in temperature and light sensor data to the A1, A2, and A4 touchpad and A and B push button sensors. We'll also include the current runtime of the Circuit Playground Express since it was last powered on or rebooted, providing five unique data points. We will transfer this data to the Raspberry Pi via the UART connection every second. A simple Python application run on the Raspberry Pi will then print these to a terminal window.
2. Second, instead of displaying this data within a terminal window, we'll display the data within a dynamic Hypertext Markup Language (HTML) document, viewable from a web browser client. Starting with Python version 3, Python code can be used to set up an HTTP server, watching for HTTP client requests for data. When a client request is received, the server dynamically creates an HTML page providing the real-time data, previously col-

lected data, static text, or a combination of the above.

3. Third, we will add the option to power LEDs on and off from a web browser. To accomplish this, we can use Python code, bringing together the HTTP server code from exercise two with code from the Coding Electronics chapter in the Orange Unit. Two options are available: The first uses the push buttons and LEDs as configured in the Orange Unit Coding Electronics chapter. The second uses a diffused RGB (Red, Green, Blue) LED to allow us to choose one of seven different colors for the LED display.

As we get started with these exercises, you may find it helpful to quickly review Blue Unit, Session 4, “Raspberry Pi Counterstory Little Free Library.” In particular, make sure you have your Circuit Playground Express connected to your Raspberry Pi using a TTL/USB Serial Cable. The TTL connects to the Circuit Playground Express using male to alligator clip wires. Disconnect the red wire from power (VOUT) on the Circuit Playground Express as we’ll be connecting the Circuit Playground Express to your personal computer with a microUSB to USB cable to edit the Circuit Playground Express code for these exercises.

If you haven’t been using the Raspberry Pi since the end of the Orange Unit, it might be helpful to review Orange Unit 4B: Meet the Microcomputer, including videos found within the chapter, and Exercises found in Orange Unit 4C: Getting Started with the Raspberry Pi.

- If you need help using RealVNC (**V**irtual **N**etwork **C**omputing) on your laptop to serve as the keyboard, mouse, and monitor interface with the Raspberry Pi, review the later steps in the Exercise: First Boot of the Raspberry Pi OS
- If you are now connected to a different Local Area Network (LAN), you’ll need to connect the Raspberry Pi to a keyboard, mouse, and monitor temporarily to get it to connect to that network. If you have problems getting the monitor to serve as a screen for the Raspberry Pi in this new location, you’ll find FAIL FORWARD Tips in the Exercise: First Boot of the Raspberry Pi OS
- If you need help using the PiOLED to identify the IP address so that you can reconnect the VNC View on your laptop to the VNC server on your Raspberry Pi, review Exercise: Adding an Adafruit PiOLED Text Display

Opportunity to Remix

These activities do not use databases or machine-learning code to create that true IoT device. Further, while both the Circuit Playground Express and Raspberry Pi devices are configured to transmit (TX) and receive (RX) data via the Universal Asynchronous Receive Transmit (UART) protocol, MakeCode currently does not support the RX pads on the Circuit Playground Express. For those wanting to do a deeper dive, follow Adafruit’s tutorial to use CircuitPython on the Circuit Playground Express, instead of MakeCode, to effectively use both the TX and RX pads and create a full serial bi-directional communications with the Raspberry Pi. Note: If you want to return to working in MakeCode after using CircuitPython, Adafruit provides instructions for uninstalling CircuitPython.

Exercise: Using Python to Monitor Sensor Information

Let's return to the circuitplayground-ToolboxTrumpet4Octave-WithUART.uf2 MakeCode[1] used within Blue Unit chapter 4B: Raspberry Pi Counterstory Little Free Library. We can reuse that code, remixing

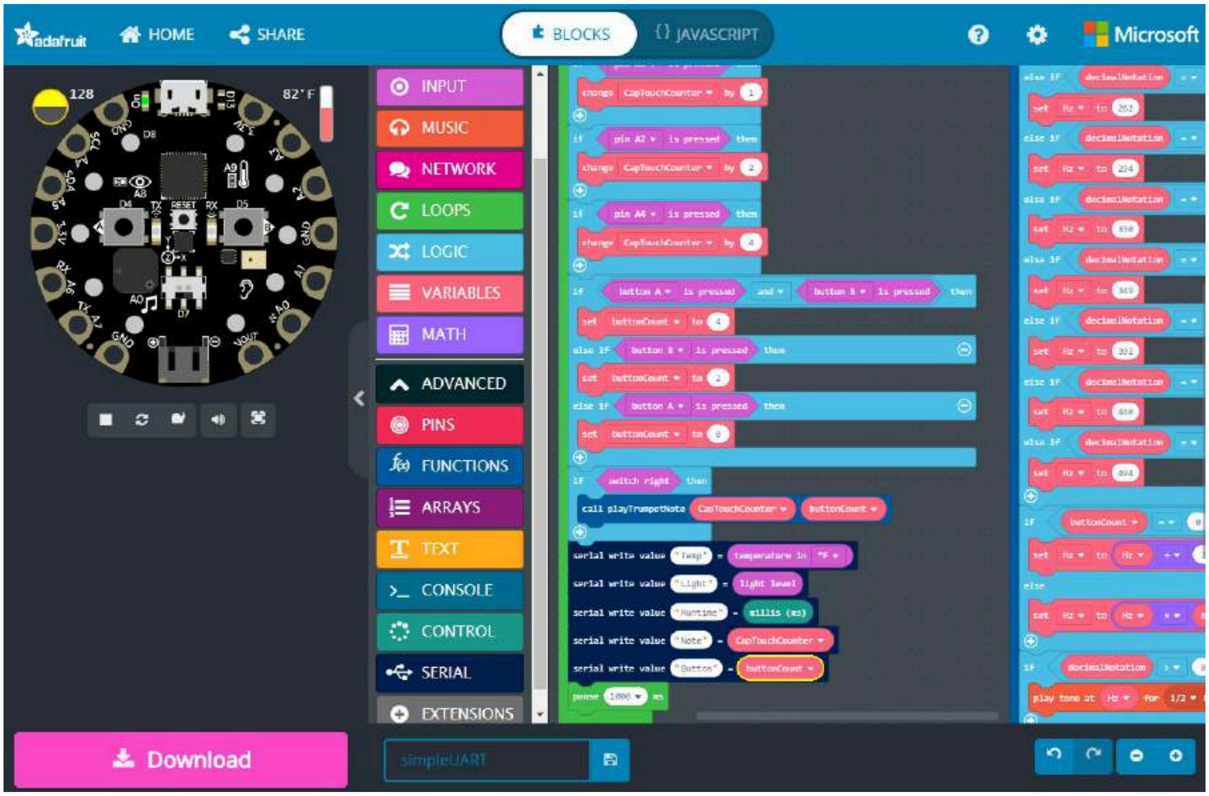
1. MakeCode © 2020 Microsoft and Adafruit. Screenshots used with permission.

it slightly to re-purpose the "serial write" so that it outputs sensor data. We'll test out this modification using a new Python program instead of the previous `cstories.py`.

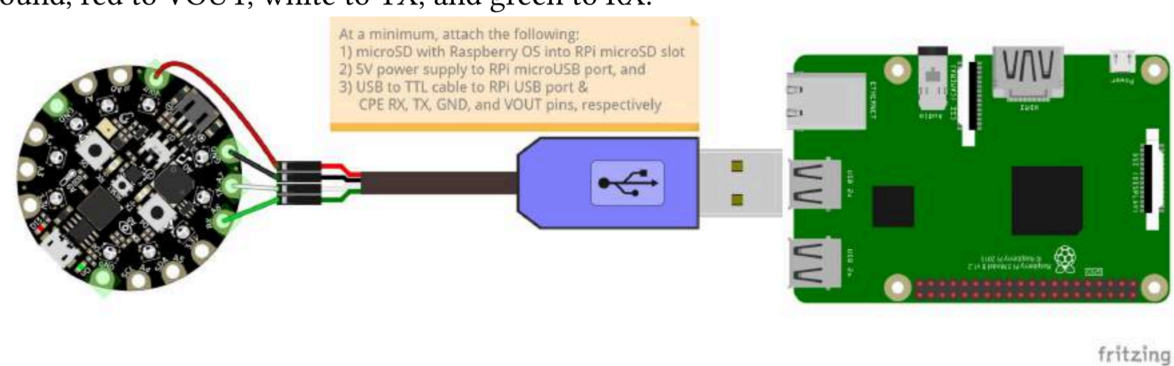
See the above image for where we're beginning.

1. Within the FOREVER loop, delete the "if CapTouchCounter > 0" LOGIC statement. This should also delete the two serial write number blocks within the if statement. Deleting this conditional "If…Then" logic statement is needed as we'll want to transmit the serial write values whether or not the pads A1, A2, or A4 on the Circuit Playground Express have been touched.
2. Add five new "serial write value "x" = 0″ blocks just above the "pause 1000 ms" block. These should be set up as follows. NOTE: value names are case sensitive as this will need to match the Python code run on the Raspberry Pi.
    a. 'serial write value "Temp"' should be setup to transmit the INPUT "temperature." It can be left as ℃ or changed to °F if you prefer Fahrenheit over Celsius.
    b. 'serial write value "Light"' should be setup to transmit the INPUT "light level."
    c. 'serial write value "Runtime"' should be setup to transmit the CONTROL "millis (ms)," that is, the number of milliseconds elapsed since power on.
    d. 'serial write value "Note"' should be setup to transmit the VARIABLE "CapTouch-Counter."
    e. 'serial write value "Button"' should be setup to transmit the VARIABLE "button-Count."

- NOTE: Previously we transmitted a number. These are sent in 1-byte packets (1 byte = 8 bits). These numbers, which were in the range of 0-4 and 0-7, were predictably sent as a single 1-byte packet. Moving forward, values such as Runtime will range, requiring 1 or more byte-sized packets. In MakeCode, the easiest alternative is to send each as a complete line of data that ends with a carriage return and line feed. In this way, they can easily be displayed line by line in a console window, just as if you were typing in the data using a keyboard and then hitting the "Enter" key at the end of each line of data. Your lines should now look something like this. Note the two new sensor icons within the simulator window:



3. Change the name of the file to SimpleUART, hit the save icon, and then flash this to your Circuit Playground Express.
4. With the slide switch on the Circuit Playground Express to the right, you should be able to again play four octaves of the ‘C’ scale. Slide it to the left if you do not want it to play the scales moving forward.
5. With the Circuit Playground Express now running the new code, disconnect the USB cable between the CPE and your laptop or Raspberry Pi. You’ll instead connect your Circuit Playground Express to the Raspberry Pi using the USB to TTL cable. Connect the black to ground, red to VOUT, white to TX, and green to RX.



6. From a web browser, enter: https://go.illinois.edu/simple_uart5.
7. Take a glance at this new Python code called simple_uart5.py:
    a. This code first imports some needed Python library packages accessed through

the Python Package Index[2], including:

- serial, used to receive and transmit serial data
- os, which will let us to run Linux commands and read data
- sleep, which lets us run the time function
- RPi.GPIO, the PyPI project specifically for the Raspberry Pi GPIO.

b. The code creates a variable called `uart` and connects it to the `/dev/ttyUSB0` device. If you needed to change the USB port used for the Blue Unit Toolbox Trumpet Counterstory Player, keep this in mind. You'll need to make that change again after downloading this file.

c. A key addition is a new function, `convert`, that is created to change the Runtime from milliseconds to hours, minutes, and seconds.

d. The last half is a "while True" FOREVER loop that reads in the five lines each time they are sent from the Circuit Playground Express. It fine tunes each of the lines to make them more readable and then prints them to standard out, which typically is a terminal window display of the Raspberry Pi.

e. NOTE: An essential piece of the fine tuning is related to the "decode" process. In particular, the Circuit Playground Express sends lines of data encoded using **UTF-8**. According to Wikipedia, "**U**nicode **T**ransformation **F**ormat **8**-bit is a variable-width encoding that can represent every character in the Unicode character set. It was designed for backward compatibility with ASCII and to avoid the complications of endianness and byte order marks in UTF-16 and UTF-32."[3] **ASCII** is the American Standard Code for Information Interchange. It was the standard for US-based computers and the basic encoding scheme initially within the Internet Protocol. However, this limited usable character sets, something that plagues much of global Internet usage to this day.[4]

8. Right click on the `simple_uart5` programming code and save it as `simple_uart5.py` to your personal computer (if connecting to the Raspberry Pi using a screen, PuTTY, or SSH) or to your Raspberry Pi (if using a keyboard, monitor, and mouse with the Raspberry Pi). Make sure to save it as a Python file with the .py extension, not with the default .txt extension.
    - **If the** `simple_uart5.py` **file was saved to your personal computer**, copy it over to your Raspberry Pi using the scp command.

2. Also known as PyPI, pronounced like "pie-pea-eye." Despite the shared "PI," it is not specifically Raspberry Pi-related.
3. "UTF-8" in Wikipedia, July 20, 2020, https://en.wikipedia.org/w/index.php?title=UTF-8. For more resources on UTF-8, see http://www.utf-8.com/.
4. Endianness defines the ordering of bytes of a word of digital data, with big-endian-based and little-endian-based computer architectures ordering bytes in storage in two significantly different ways. Today's major personal computer providers use the little-endian format.

- **If you have more than one USB device connected to your Raspberry Pi**, you may need to edit the Python code using nano, to match the /dev/ttyUSB0 line to your exact USB.

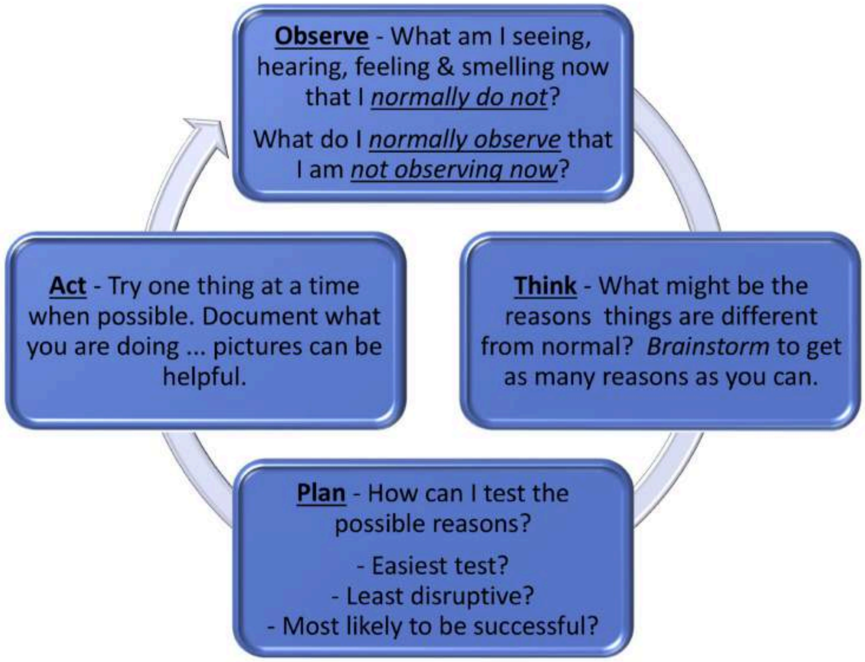
9. In a Raspberry Pi terminal window, type in:

- `pi@raspberrypi:~ $ python3 simple_uart5.py`

Once the program begins, you will see new sensor information printed to the terminal window every second. Try pressing different sensors and buttons to see the data output change. You can end the Python session at any time by hitting CTRL-C. The display will look something like this:

```
pi@raspberrypi:~ $ python3 simple_uart5.py
Raw Response:  e:0
               ['e', '0']
               0
Raw Response:  Button:1
               ['Button', '1']
               1
Raw Response:  Temp:77
               ['Temp', '77']
               77
Raw Response:  Light:21
               ['Light', '21']
               21
Raw Response:  Runtime:2143584
               ['Runtime', '2143584']
               2143584
               0:35:43
-----------------------
Raw Response:  Note:0
               ['Note', '0']
               0
Raw Response:  Button:1
               ['Button', '1']
               1
Raw Response:  Temp:77
               ['Temp', '77']
               77
Raw Response:  Light:21
               ['Light', '21']
               21
Raw Response:  Runtime:2144772
               ['Runtime', '2144772']
               2144772
               0:35:44
-----------------------
^CTraceback (most recent call last):
  File "simple_uart5.py", line 21, in <module>
    response = uart.readline().decode('utf-8')
  File "/home/pi/.local/lib/python3.5/site-packages/serial/serialposix.py", line 483, in read
    ready, _, _ = select.select([self.fd, self.pipe_abort_read_r], [], [], timeout.time_left())
KeyboardInterrupt
pi@raspberrypi:~ $
```

This may take some pair programming and failing forward to get up and running. Remember to work through the troubleshooting basics cycle to work from "not yet" to "yet"!



Key Takeaways

Microcontrollers are common in many of our everyday electronics, from computers to printers to microwaves to cars. Sometimes they only serve to sense device context and then control the motors running a device accordingly. Other times, sensor data is collected and passed on to other microcontrollers and to microcomputers. And sometimes microcontrollers collect data from other sensors first. The Circuit Playground Express has been designed first as an educational tool, but as we've seen first building the Toolbox Trumpet Counterstory Player and now in this exercise, it has a range of sensor capabilities that could be used as datasets informing actions for a range of information services.

What might you do next to remix this?

So far we've used the terminal window to view Python information. But since Python version 3, it is also possible to print this information as Hypertext Transfer Protocol (HTTP) server data instead of, or in addition to, a standard output like we've now sent to the terminal window. With the Python code and UART communications confirmed, let's explore how this data can be made remotely available.

The Hypertext Transfer Protocol is a well-established, highly used TCP/IP-based client/server protocol that we'll explore more formally in Rainbow Unit, Session 2. For now, consider that you make regular use of web browsers which perform a client role putting forward requests to associated servers using the same protocol. Today, web browsers are mostly used to access pages stored on HTTP servers by typing in an http:// or https://, followed by an Internet Protocol name associated with that server, and perhaps further subdirectory names on that server, each separated by another '/' symbol, and finally the resource name at the end of the last '/' symbol if needed. Together this creates a **Uniform Resource Locater (URL)**. The URL is a commonly used form of the more general Uniform Resource Indicator (URI), strings of characters used to unambiguously identify a particular resource. The first part of the URL states the protocol to be used. Web browsers actually support a number of different protocols, but we almost always make use of the HTTP protocol or the secure Hypertext Transfer Protocol, HTTPS. For the following exercises, the remainder of the http:// URL will include the Raspberry Pi's IP address and a special port number we'll define in the Python code to separate it from any other web servers that may be running on your local network.

Exercise: Using http to Remotely Monitor Sensor Information

For this exercise, we do not need to change anything on the Circuit Playground Express or with the TTL/USB cable connection to the Raspberry Pi. The focus instead is on the Raspberry Pi IP Address and the Python code that will be used to read the five lines of serial data sent once every second by the Circuit Playground Express using the TTL/USB cable.

1. From a web browser, enter: https://go.illinois.edu/simple_uartWebserverCPE.
2. Take a glance at this new Python code called simple_uartWebserverCPE.py. This is pretty complex, and do not worry if it doesn't make much sense at this time. Notes have been added, delineated using the comment character "#". Doing a quick read of these comments might be helpful to just highlight key aspects of the Python code. Later, after running this Python code on the Raspberry Pi and then reading data through a web browser, come back and review it again.
3. Right click on the `simple_uartWebserverCPE.py` programming code, and save it as `simple_uartWebserverCPE.py` to your personal computer (if connecting to the Raspberry Pi using a screen, PuTTY, or SSH) or to your Raspberry Pi (if using a keyboard, monitor, and mouse with the Raspberry Pi). Make sure it is not saved with a final .txt extension.
4. This code will now need to be edited using a simple text editor on your laptop, or using Thonny or mousepad on the Raspberry Pi:
   `pi@raspberrypi:~ $ mousepad /home/pi/Code/Python/simple_uartWebserverCPE.py`
   Whichever text editor you use, here are the changes that need to be made:
   - In the third section at the top of the code, note that the host_name is configured to the IP address assigned to my home Raspberry Pi (more on connecting your Raspberry Pi to a Local Area Network (LAN) is provided in the backmatter of this textbook). The host_name variable will need to be changed after downloading the

Python code to match the IP address you find when typing into the Raspberry Pi terminal window the following:

- `pi@raspberrypi:~ $ hostname -I`

Also included in this section is a host_port. Think of a shipping port of call as an analogy. When arriving at a shoreline with designated docking ports for a boat or ship, a set of names or numbers are used. This is the same for the Internet Protocol. By default, HTTP uses port 80 and HTTPS uses port 443. Ports from 0 through 1023 are well-known system ports and are stricter than ports above this range. Port 8080 is actually a common number used for test web servers. The iSchool course for which this textbook is commonly used is IS401, and so 8401 is the chosen host_port listed here. But this can be changed, and indeed should be changed if the same Raspberry Pi is attempting to run multiple different web servers simultaneously.

- **If you have more than one USB device connected to your Raspberry Pi**, you may now also need to revise the /dev/ttyUSB0 line to an appropriate USB device other than 0.

5. Save the changes to the Python code.
    - **If** `simple_uartWebserverCPE.py` **was edited using mousepad on the Raspberry Pi**, save the file and exit mousepad.
    - **If** `simple_uartWebserverCPE.py` **was edited using the Thonny Python IDE on the Raspberry Pi**, save the changes. You can then use the Run icon at the top of Thonny to test out the code.
    - **If** `simple_uartWebserverCPE.py` **was edited on your personal computer**, copy it over to your Raspberry Pi next.

6. With the Python code tweaked to match your IP address and possibly your `/dev/ttypUSB` assignment, type the following in a Raspberry Pi terminal window:

- `pi@raspberrypi:~ $ python3 simple_uartWebserverCPE.py`

7. In a web browser **located on the same local network as your Raspberry Pi**, type in your Raspberry Pi's IP address, followed by `:8401`, like this:

- `http://10.0.0.30:8401`

This is the most complex piece of programming code we've worked with yet. It includes changes needed within Python to match it with the IP address of your Raspberry Pi. For some, this may be the first time you've edited Python. You may have been using the Raspberry Pi in Graphical User Interface mode with a keyboard, mouse, and monitor; and thus it might be the first time you've had to work with the IP address of the Raspberry Pi.

In addition, many new concepts related to networking are just starting to be introduced. In all, there are plenty of "fail forward" opportunities to advance a growth mindset. Maybe too many for the time being, in which case "yet enough" might be working collaboratively with another person to walk one of you through this exercise on one Raspberry Pi.

But I would encourage you to spend at least 45 minutes or so working to fail forward on each Raspberry Pi available, comparing and contrasting what works and what doesn't work as a community of practice.

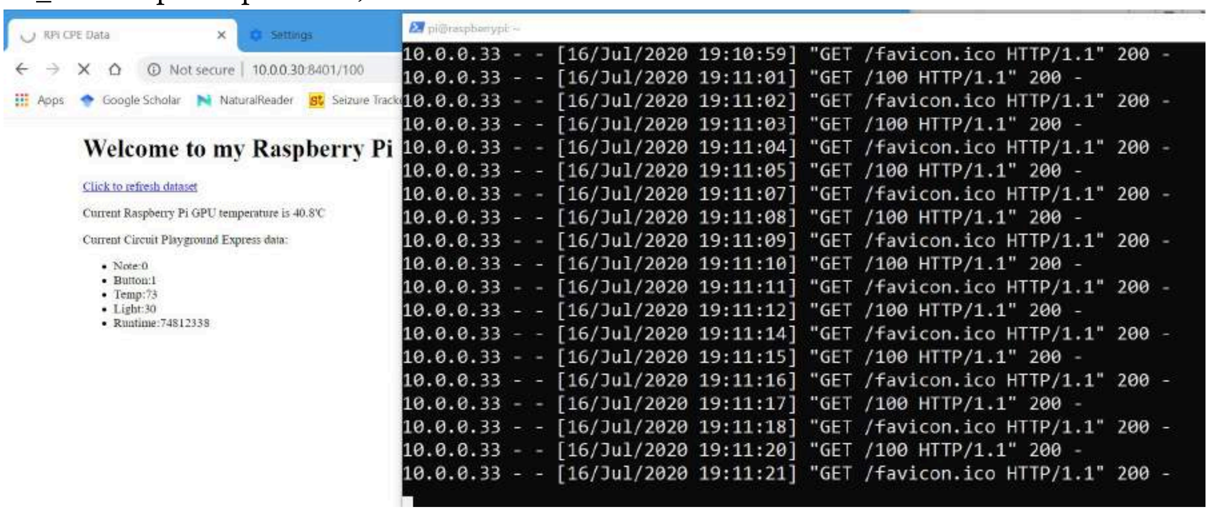
When you're ready, take a closer look at the following highlights of the `simple_uartWebserverCPE.py` code:

- In addition to several other Python libraries, the code imports the http.server module of the base Hypertext Transfer Protocol, or HTTP, request handler and server library package.
- The core part of the Python code is the class BaseHTTPRequestHandler. In Python, "classes provide a means of bundling data and functionality together."[5] BaseHTTPRequestHandler "is used to handle the HTTP requests that arrive at the server. By itself, it cannot respond to any actual HTTP requests; it must be subclassed to handle each request method (e.g. GET or POST)."[6]
- In this case, do_HEAD() sends the headers for a typical GET request sent from a HTTP client such as your web browser, including a 200 OK response and a content type 'text/html' noting text and/or **Hypertext Markup Language (HTML)** is being sent back to the client.
- The do_GET() does the lion's share of the work setting up a dynamic HTML page. It can do this because a special implementation of BaseHTTPRequestHandler is being used for reading data from and controlling GPIO of a Raspberry Pi—note the Import RPi.GPIO at the top of the file.
- At the bottom of the do_GET() function, you'll note a wfile.write statement, written in the stated html.format, encoded in UTF-8. Within the html.format portion of this statement, there are seven items providing data to be integrated into the seven {} tags above, in order from top to bottom. So the "temp" data is collected using the vcgencmd—that is, the Raspberry Pi VideoCore GPU command line utility. The five Circuit Playground Express data items are read next through the USB to TTL serial port, and finally an empty status line,

5. "9. Classes — Python 3.8.4 Documentation," accessed July 20, 2020, https://docs.python.org/3/tutorial/classes.html.
6. "Http.Server — HTTP Servers — Python 3.8.4 Documentation," accessed July 20, 2020, https://docs.python.org/3/library/http.server.html#module-http.server.

something currently unused.

- The last lines of the Python code are the location where the server hangs out waiting for client requests before going into action with the class BaseHTTPRequestHandler code.
- The html text within the code includes a statement:
  <meta http-equiv="refresh" content="1">.
  Many web browsers know what this statement means and are set up by default to follow this meta statement to refresh the content with a new request to the server every second. The terminal window will note who placed the request for information, when the request was placed, and the do_HEAD response provided, while the web server will show the do_GET response provided, as shown below:



  A link stating "Click here to refresh dataset" and javascript refresh script is also provided in the HTML to facilitate manual refreshes of the data, in case a web browser does not follow the meta instructions.

Key Takeaways

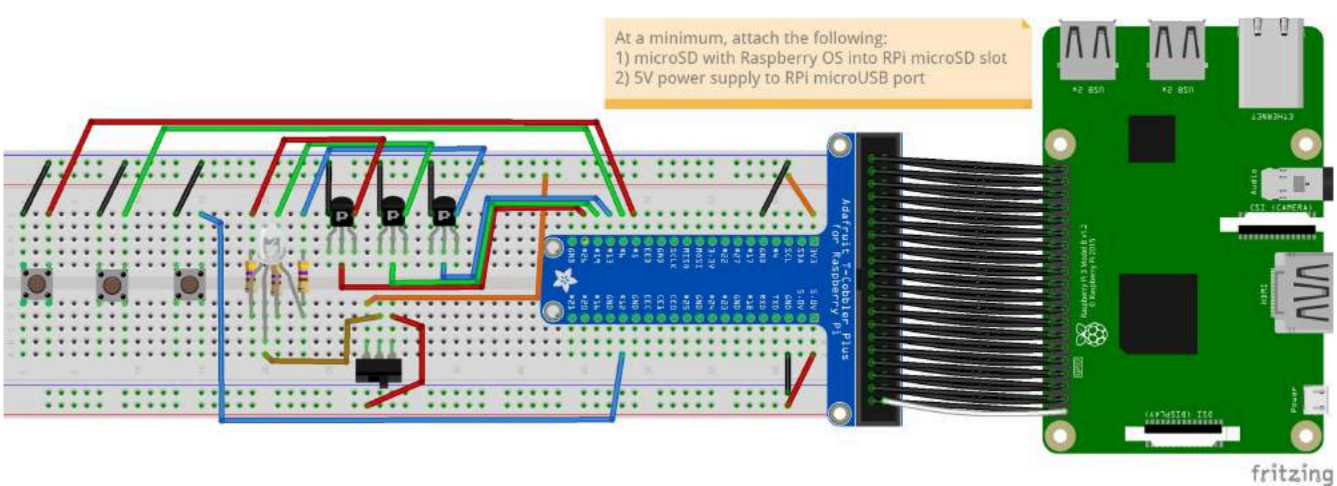
The Hypertext Transfer Protocol (HTTP) is a widely used client/server protocol. For most, you'll encounter the HTTP application as a general-purpose server—a common one being Apache—in which many different parent and child web pages are located through websites located on a single web server. This textbook in its online version is a gathering of many individual web pages located on one of those general-purpose servers. There are other [Illinois Open Publishing Network] "Windsor & Downs Press" books, and also a number of "Publishing Without Walls" books and "IOPN Journals" all located on this general purpose web server.

In this exercise, though, we're running a special-purpose, Python-based HTTP server on a microcomputer and we are setting it up for the specific purpose of distributing data it collects both from its own hardware and data provided over a serial connect from a microcontroller—all gathered within the Python program to share as a single web page. But as is the case with the majority of HTTP-provided data, it is sent using the Hypertext Markup Language (HTML), the standard for web pages.[7] An important takeaway is the distinction between HTML standard and the HTTP protocol. Both are Internet Protocol-based. But while standards serve as guidelines that can or should be followed, protocols serve as rules defining exactly how data is exchanged and the expected behavior to be followed within this change. You can put anything into an HTML page that you want, and it will be sent to the requester—whatever is in there. But if it is something outside of the standard, it may be ignored or even lead to distorted communication between the client and the server using this protocol.

Within an object-oriented programming language, a collection of data variables are acted upon using defined functions that are brought together as an object. A class then serves as the blueprint for that object. In this exercise, the class BaseHTTPRequestHandler brought together two functions, do_HEAD and do_GET, each beginning with the Python function keyword "def". The do_GET() method, or function, implements the standard GET method of HTTP/1.1 used to retrieve whatever information is identified by the Request-URI. The do_HEAD() method, on the other hand, MUST NOT return a message-body but rather the metainformation continued in the HTTP headers. This is different from the <head></head> of the HTML standard, which is instead meta-information as defined within the HTTP protocol.

> For more, take a look first at the [formal protocol definitions page](). This was the original document used by W3C, the World Wide Web Consortium based on sections from RFC9110 and which was referenced in the first edition of this textbook, published in 2020. Subsequently, in June, 2022, the Internet Engineering Task Force (IETF) published [RFC 9110, HTTP Semantics](). This now supersedes the W3C document and the better source to learn more about the semantics used in the above paragraph. The IETF and the **Request for Comments (RFC)** process are formally introduced in [Rainbow Unit 2A: Digital Internets Past and Present]().

Currently MakeCode on the Circuit Playground Express only supports the writing of data using one of the serial write blocks, thereby providing data to the other side of the UART connection. It does not support the serial read blocks. Otherwise, we could enter the data into variables and use those, for

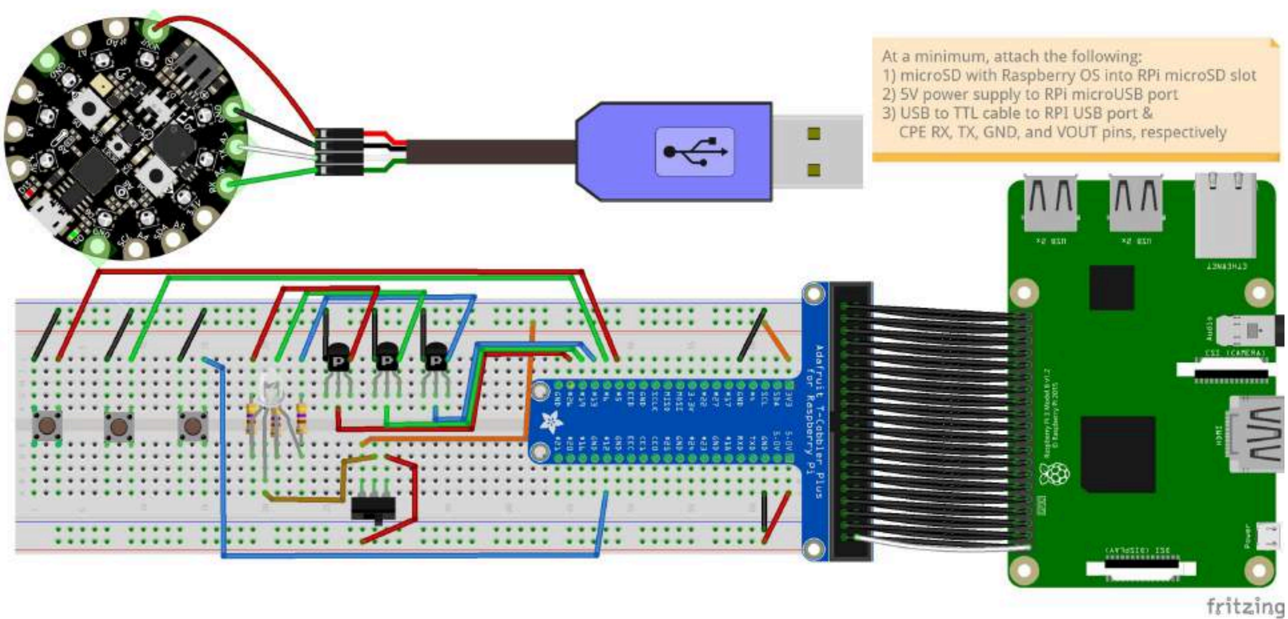


7. A [tutorial for HTML]() is available from [W3Schools.com]().

instance, to set LED pixels to specified colors. But think back to the Orange Unit's [Coding Electronics](#) chapter where we used the [Python command switchLED.py](#) to read the three momentary switch push buttons and turn on or off associated LEDs via transistors for each of the three parallel circuits.

Exercise: Using HTTP to Remotely Control LEDs

As we enter this last exercise, let's consider how we might make use of the Python class BaseHTTPRequestHandler on the Raspberry Pi to not only provide regularly updated Circuit Playground Express data, but also information regarding which LED is currently running on the breadboard based on the status of the Momentary Switch push buttons just reviewed above. While this will bring together two "things" on one Raspberry Pi, this also provides a potential launchpad for considering ways different microcontrollers and microcomputers could ultimately further be brought together to create a range of local "Internet of Things" devices.



This final Python program is a remix of the simple_uartWebserverCPE. A new table has been added to the HTML portion of `def do_GET` from which a selection can be made regarding which of the three LEDs should be on, and which should be off. And below that, additional variables have been set up to read from, and write to, the appropriate GPIO pins.

1. From a web browser, enter: https://go.illinois.edu/simple_uartWebserverLED
2. As before, take a quick glance at this new Python code called simple_uartWebserverLED.py. Much of it is taken directly from the simple_uartWebserverCPE.py program. And as before, after downloading this, you will need to edit the Python file to change the "host_name" from 10.0.0.30, the local IP Address of my Raspberry Pi at my home office, to be whatever IP address you were assigned on your local network. The file is much longer, though, because

of the new HTML table and then the new series of “If…Then…Else” conditionals, one to provide an edited output of the Circuit Playground Express data, the other to interact with the GPIO pins and edit associated variables.

3. Right click on the simple_uartWebserverLED.py programming code and save it as simple_uartWebserverLED.py to your personal computer (if connecting to the Raspberry Pi using a screen, PuTTY, or SSH) or to your Raspberry Pi (if using a keyboard, monitor, and mouse with the Raspberry Pi). Make sure it is not saved with a final .txt extension.
4. This code will now need to be edited using:
    - Thonny Python IDE, in which case you can test the code through the Thonny Run icon, or
    - using mousepad on the Raspberry Pi
      `pi@raspberrypi:~ $ mousepad /home/pi/Code/Python/simple_uartWebserverLED.py`

- Whichever text editor you use, here are the changes that need to be made:
    - **ALL**: Change the host_name to match the IP address assigned to your Raspberry Pi as seen in the hostname command, or as used to ssh into your Raspberry Pi.
    - **If you have more than one USB device connected to your Raspberry Pi**, you may now need also revise the /dev/ttyUSB0 line to an appropriate USB device other than 0.
    - **ALL**: save the changes you just made.

5. Before going to the next step, be sure other Python programs we’ve been running are stopped. This includes:
    - `simple_uartWebserverCPE.py` used in the previous exercise is not still running as it is already using the assigned port number for its own HTTP server.
    - `switchLED.py` which you might have run in the Key Takeaways of the last exercise.
    - To stop a program running on the Linux command line, you typically will type:
        - `pi@raspberrypi:~ $ CTRL-C`
          Where CTRL- indicates for you to press and hold down the control key on your keyboard while typing in the additional key, which in this case is the letter ‘c’.

6. With the simple_uartWebserverLED.py Python code tweaked to match your IP address and possibly your `/dev/ttypUSB` assignment, in a second Raspberry Pi terminal window, type in:

- `pi@raspberrypi:~ $ python3 /home/pi/Code/Python/simple_uartWeb-`

```
serverLED.py
```

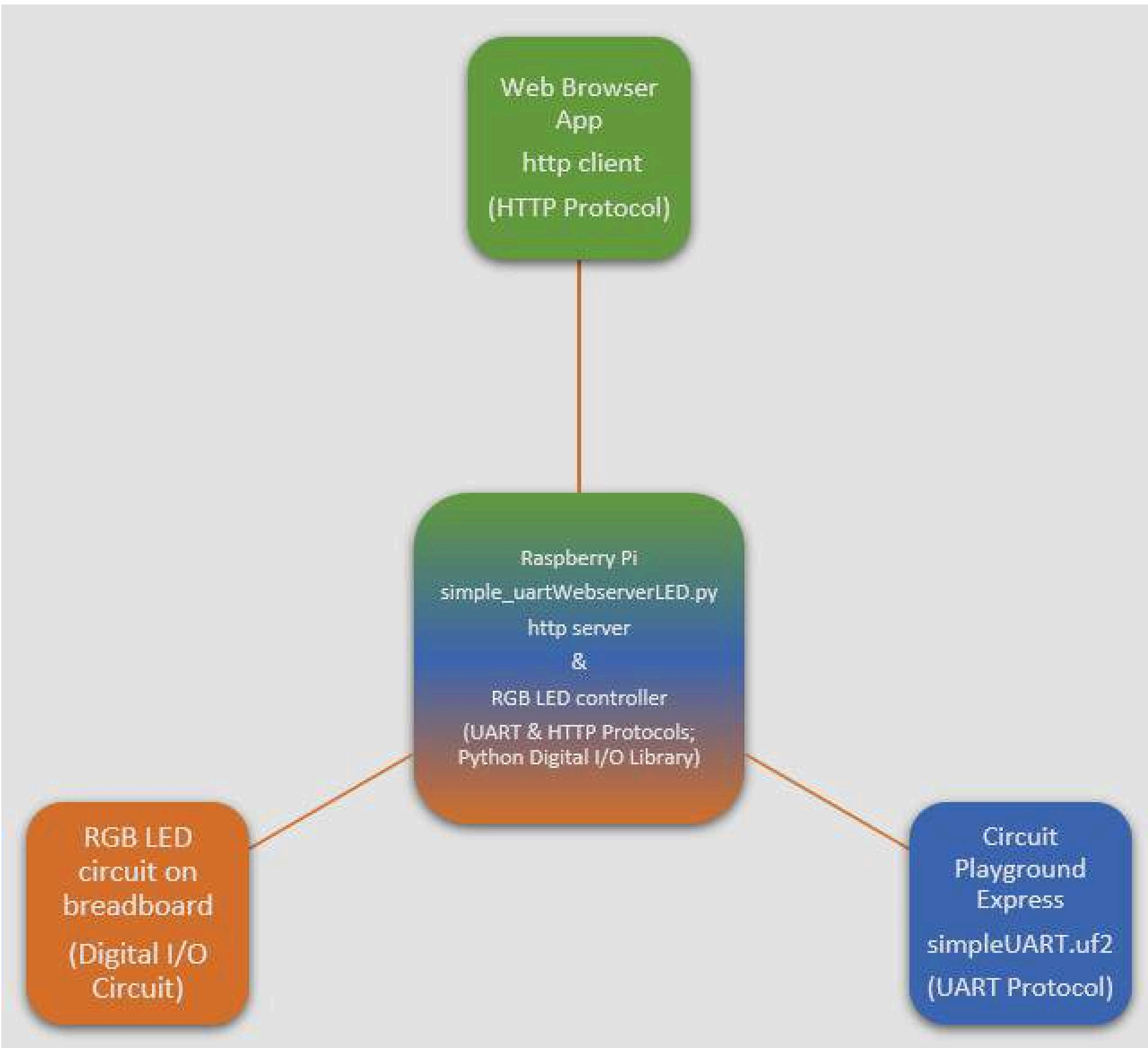
7. In a web browser **located on the same local network as your Raspberry Pi**, type in:

- `http://172.16.112.58:8401` Again, replace the IP address above with your Raspberry Pi’s. As you can see, this example is on a different LAN from the previous examples.

You should now see something like the following:

Key Takeaways

In this exercise, we have revisited an exercise from the Orange Unit in which Python code was used to create a programmable circuit allowing us to selectively turn on and off LEDs using Momentary Switch push buttons. We have continued to use a MakeCode Toolbox Trumpet program on the Circuit Playground Express, which allows us to play four octaves of the C-Scale. And, finally, we expanded the Python code from the previous exercise to both report on data from the Circuit Playground Express and also from the Raspberry Pi GPIO, and also add in the ability to remotely control another red, green, and blue LED set.



An illustration demonstrating the multiple communication channels active simultaneously. These include digital I/O communications between the breadboard and Raspberry Pi, UART communications between the Circuit Playground Express and the Raspberry Pi, and HTTP communications between a web browser and the http server Python code on the Raspberry Pi.

Pause to look again at the two images taken to display the target outcome of this exercise. Now that these are set up and working properly, consider the following probe questions:

- How many different devices are now playing a role as tools within this third exercise?
- How many different computer programs are now playing a role facilitating the use of these tools?
- Which tools could be used independent of the other tools shown in the images? Which ones must be used together to work? Which ones cannot work fully independently, but could be used through slight adjustments, for instance, by just being provided a separate power source?
- Which ones now require access to the Internet periodically, or continuously, to function effectively? Consider the Internet not as the local network, even though it works using the Internet Protocol, but rather the connection to the global Internet provided by your Internet Service Provider.
- Are there times when you see discrepancies in the timeline of information collected vs. displayed? Which components might be shaping these discrepancies, if any?
- Who shaped the different tools, including the electronics and computer programs used to create these tools? How does this shaping go on to shape your use of them?

Wrap Up

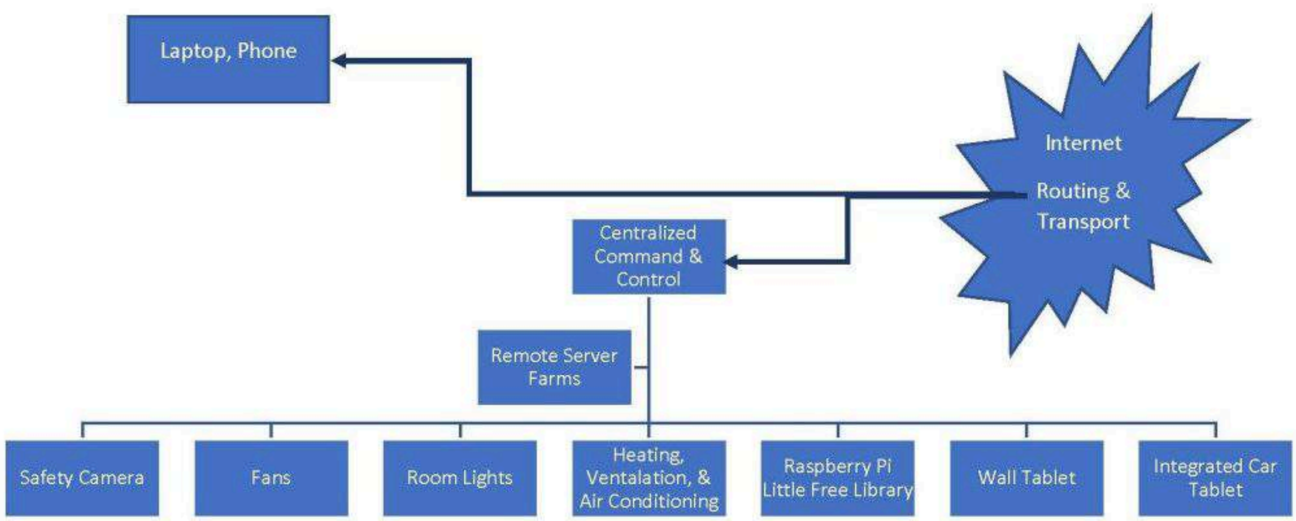
As we've worked through Session One, we've worked to demystify what it means when someone says a thing is an Internet of Things, or Smart device. We've worked to unpack what is actually within these devices, and also what is essential for them to work together. The goal is for this to help us to see that IoT devices can often stand separately and work independently. But as they are brought together on a local network, the data can start being combined to further influence the monitoring and activity of these interrelated devices. Also, as we've seen here, a local microcomputer can be set up to collect the data from multiple other datasets separately collected on devices and consolidated into a single output.

From here, consider that both the Raspberry Pi and Circuit Playground Express themselves can collect data from multiple worker devices as a parent using the I2C serial data transmission protocol. When Circuit Playground Express is programmed using Python or Arduino, this could include both input and output data with a range of remote microcontrollers and integrated circuit sensors and motors, with data being stored in a file instead of, or in addition to, sending it to the Raspberry Pi using the UART serial communications protocol. Likewise, the Raspberry Pi can both provide data dynamically to a web browser and also store this data in a file or database. And while our programs have been used to provide a consistent unconditioned stimulus and response pattern, it is possible that through the use of machine learning algorithms, this can be expanded to create conditionals to reshape stimulus and response patterns. All of this can readily be done without Internet connectivity using local networking technologies.

But it's also important to note that there's the Internet of Things and there's the Bitnet of Things as differentiated by Neil Gershenfeld and his colleagues. If the Raspberry Pi serves exclusively or primarily as a storage center for data, this is more like a centralized Bitnet of Things. Even if this centralized system facilitates remote control of devices, it has not yet moved to being an Internet of Things device, especially if each microcontroller/microcomputer device with which it might communicate remains separate from all other devices within its arena.

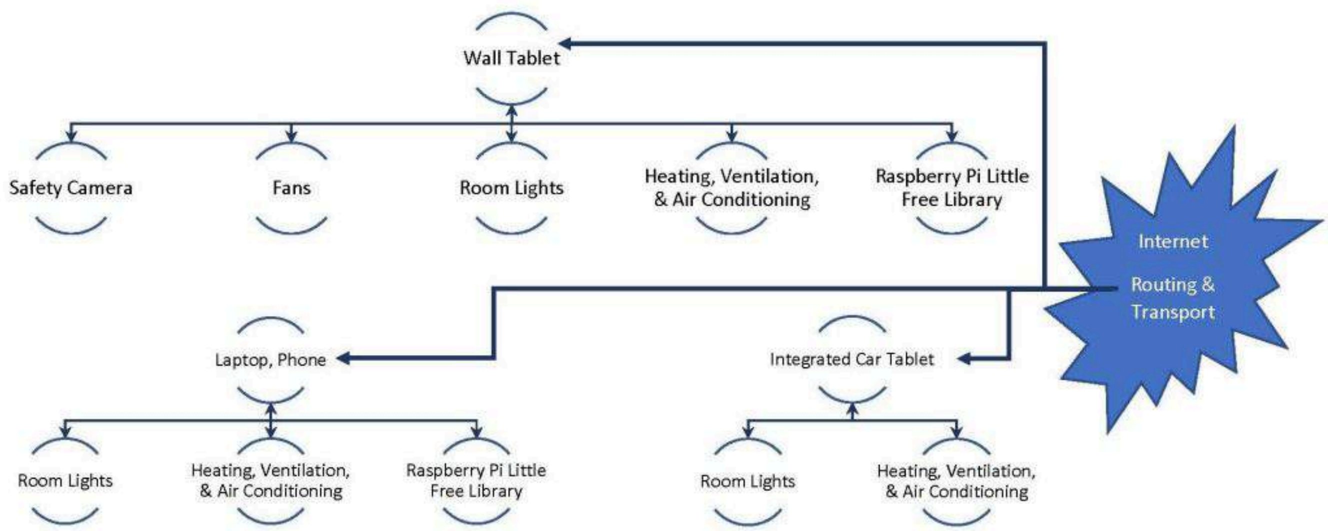
As an example, I found it increasingly important to purchase a window air conditioner for our home woodshop given the rising nighttime temperatures, increased humidity, and the negative impacts this was having on the metal tools and wood used in our carpentry projects. This week I purchased a Toshiba Smart Wi-Fi Window Air Conditioner with Remote and ENERGY STAR. In setting it up, I was able to use the easy to use touch controls. What about some of the other methods for working with the air conditioner specified using the terms Smart, Wi-Fi, and Remote? Toshiba clarifies the Remote Control as a now standard radio frequency handheld remote used within the same room. The Wi-Fi, on the other hand, requires a connection to the Local Area Network, thereby facilitating communications with the associated *smart*phone app and with voice controlled using Amazon Skill. Toshiba has helpfully clarified that their use of the term smart indicates the air conditioner, like our remotely controlled LEDs, can interface with smartphones. And while the Amazon Alexa Skills system does incorporate machine learning and narrow artificial intelligence (AI for a specific task), these are not incorporated within the air conditioner to provide ongoing engineering of the auto settings.

This is the concept behind the Bitnet of Things for Neil Gershenfeld and others.



While many home devices today make use of the Internet for routing and transport, it can often be for communication a centralized "Bitnet of Things" command and control center rather than between a range of Internet of Things devices.

What might be the alternative pathway in which the air conditioner was integrated within an Internet of Things workshop? The table saw, bandsaw, planers, chop saw, woodlathe, air compressor, dust collector, and various other mechanical tools and also the wood resources would include sensors set up as Internet of Things devices. The dust collector, air compressor, and air conditioner would monitor the saws, planers, and lathes to detect usage and performance levels. These would be used to adjust settings on the climatic and air quality standards within the space to increase performance and longevity of machine, human, and more-than-human alike. In part, this would be done through integrated machine learning that worked not only individually on each of the many sensors and controllers, but through end-to-end strategic learning to recognize the best adjustments given the different contexts and current requirements. After all, the woodshop is a very different space when it is also being used to raise a new gathering of soon-to-lay-egg chicks, or when it is used by my students to design, cut, and build computer tables, or when it is used by my family to build gifts, sometimes in rapid succession.



Internet provides routing and transport between a range of Internet of Things devices and associated interface devices.

This is the concept behind the Internet of Things. Machine learning moves from a centralized command and control to each of the various tools that now have as part of the construction electronic sensors, servomotors, and other controllers. These tools then work in an artificial community of practice to achieve objectives the algorithms on each individual device could not do alone.

It is not essential for there to be Internet access, although sometimes it might be useful for there to be. For instance, if we were to configure the home router, which is provided a public IP address, to forward port 8401 requests to the private IP address of the Raspberry Pi, then web browsers around the world could choose the LED color! If done in this way, the Internet is not needed to collect and analyze the data and adjust algorithms accordingly, but instead is used to provide additional data as helpful.

And there are times when it is helpful to send data to remote databases, HTTP servers, and other systems, either as the primary server array for storing and consolidating information, or as a secondary location in addition to the primary local server. This will be the focus of Session Two of the Rainbow Unit.

For now, this and the previous exercise provide baseline examples of the broader framework used for many Internet of Things devices. But it also provided some starting points that you could use to build your own Internet of Things devices right now. For those wanting to dig deeper into the technical work of building Internet of Things devices, you may want to watch further episodes in the series, including:

- All the Internet of Things – Episode 3 – Services (Read the full video transcript on Adafruit's website).
- All the Internet of Things – Episode 4 – Adafruit IO, an IOT Service for Everyone by Adafruit for those starting out with IoT (Full transcript available on Adafruit's website).
- All the Internet of Things – Episode 5 – The S in IoT is for Security (transcript at Adafruit's website).

Comprehension Check

[An interactive element has been excluded from this location for the PDF edition of the text. Complete the activity for this chapter online.]

2A: Digital Internets, Past and Present

Background Knowledge Probe

- What is the first thing that comes to mind when someone says they are on the Internet right now? Is this the same or different from internetworking as we discussed in the last session?
- What quickly comes to mind as the top three transformative advancements that have resulted from the "digital revolution"?
- What might be the concealed stories behind the stock stories regarding these Internet and digital revolution memories?

Some Early Examples of Systems for Networking and Internetworking

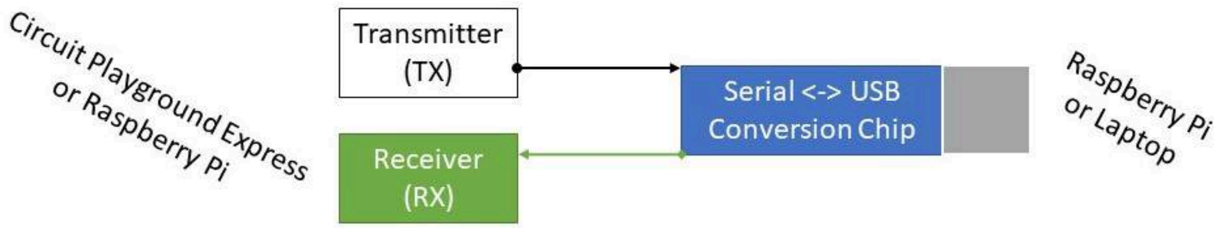
Networking

1. The *exchange of information or services* among individuals, groups, or institutions *specifically*: the cultivation of productive relationships for employment or business
2. The establishment or use of a *computer network*

First known use of networking: 1967, in the meaning defined in sense 1

Merriam-Webster, "Networking"[1]

1. Merriam-Webster, "Networking," accessed June 2, 2020, https://www.merriam-webster.com/dictionary/networking. Emphasis added.

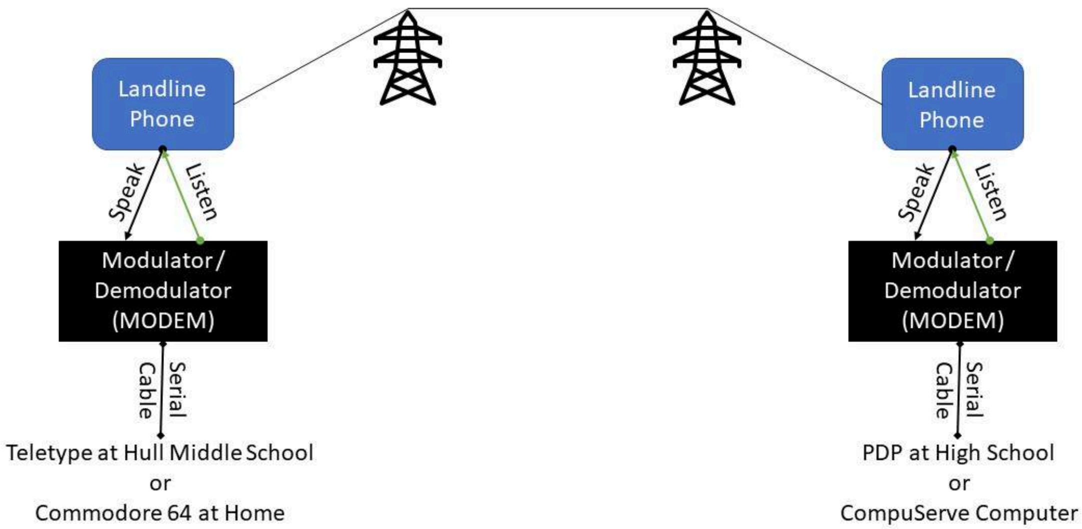


The USB to TTL serial UART connection between a microcontroller and microcomputer.

As we have explored throughout the textbook, there are many ways to connect, or network, two devices together. We've made use of the Universal Asynchronous Receive Transmit, or UART, protocol with our own devices. We have also briefly reviewed the Inter-Integrated Circuit (I2C), which is a synchronous, multi-parent, multi-worker, packet switched, and single-ended serial computer bus.

There are many other ways to network devices together, through physical cables (Ethernet, fiber) or varying wavelength frequencies (WiFi, Bluetooth, GPS, infrared, cellular, satellite, low-powered radio). My own history with networking goes back to the 1970s and the growth of radio-controlled model airplanes. While attending Hull Middle School, I used a teletype device and landline phone to dial in to a minicomputer located at the high school, both to play Star Trek video games in my free time and to write my first BASIC programming language code to calculate the board feet of a log for my family's sawmill. (My dad was not impressed. He preferred continuing to use our analog, hand-held Doyle log rules).

Then in the early 1980s I bought a Commodore 64! The night before I picked it up, I remember using a manual typewriter to complete an English class assignment at Lake Michigan Community College and continually thinking, "This is the last time I will ever need to use whiteout!!" When I got it, I also got a few issues of PC Magazine so that I could read the BASIC programming code stories that I could then enter in and use to play games and to begin creating my own calculating devices for making. And for a short period I also tested out CompuServe, a commercial online service, from which I could join in discussions and download other BASIC programming code. But landline phones were expensive when used to connect from my rural home. So for all of these, there was only one way I effectively internet-worked — sneakernet. **Sneakernet** is an informal term sometimes used tongue-in-cheek to reference those times when the best, or perhaps only, way to transfer electronic data between devices is by physically moving storage media (e.g., magnetic tape, floppy disk, hard drive, USB flash drive) from device to device.



A Modulator-Demodulator, or modem, converts the text of the sender into binary so that the data can be sent over the network. The receiving modem then demodulates the binary back into text.

But the early 1980s also saw the emergence of **BITNET**, an international cooperative network launched by Ira H. Fuchs at the City University of New York and Greydon Freeman at Yale University.[2] It used IBM's **Remote Spooling Communications Subsystem (RSCS) protocol** and a **peer-to-peer (P2P)** network **"store-and-forward" principle** in which data and information at a given BITNET-connected computer was sent to a central mainframe computer, stored, and then forwarded to other mainframe computers. This university-focused network eventually spanned the coasts of the United States and was also joined in 1982 by the European Academic and Research Network, or EARN, providing a means for communications across a broad research community. For me, this first true internetworking as I worked towards my PhD degree at Rutgers, The State University of New Jersey, provided a means to easily email and store and forward research data across campus and between universities. But as with previous networked systems I had used, the primary focus was towards centralized administration and a focus on target audiences. For me to get the support I needed from outside University BITNET members to fix microcontroller or C programming code problems with our supporting small startup businesses, I needed to communicate via landline phone and send digital data and electronics via postal mail.

2. This is not to be confused with the Bitnet of Things Gershenfeld references as the counterpoint to the Internet of Things which he has championed and which was discussed in chapter 1A of the Rainbow Unit. Learn more about historic BITNET.

Emergence of an Open Model for Internetworking

At the same time that commercial service providers such as CompuServe and cooperative university networking systems such as BITNET were emerging with a focus on centralized servers and services, other models for internetworking were also under development. A key design of these open internetworking structures was that *each* computing device connected to that network has responsibility for reliable processing and delivery of data. This pioneering work came from Louis Pouzin and the **French Cyclades Network** in the early 1970s.[3] It was created to allow **end-to-end protocol** transparency of application-specific features even when the devices are on different types of networks. Pouzin selected Cyclades as an inspirational description of between networks, or internets, from the Greek islands with the same name that form a circle in the Aegean Sea. In his presentation of the Cyclades network to French ministers in 1974, Maurice Allegre provided this metaphor:

> One should retain the image [of the Cyclades islands]; the processing centers are still today islands isolated in the middle of an ocean of data, which overwhelms our civilization. Now, thanks to the networks, these islands will be able to connect and thus contribute to a wide circle of information exchanges which will shape the future development of our society.[4]

Ultimately, the French government chose the Transpac telecom monopoly's plan for centralized control by the network provider within the network of data traffic and quality of service until 2012 before fully transitioning to commercial Internet, something for which the Cyclades open model provided essential inspiration.

However, key features of Cyclades influenced the design of the **Internet Protocol (IP)**, another packet switching distributed communications network, this one being developed through the **Advanced Research Projects Agency (ARPA)** established by the United States Department of Defense in 1966 and officially named **ARPANET**. ARPA itself was formed as an agency in 1957 with the objective to establish a United States (US) lead in science and technology applicable to the military after the launch of the satellite Sputnik by the Union of Soviet Socialist Republics (USSR). This agency was in large part a continuation of post-World War II high-technology industries of the Cold War period. Like many agencies of the post-war development period, it included extensive basic research by academia and industry for the development of generic infrastructure, protocols, and standards.

One central development based on this unique US open internetworking sociotechnical system was the launch of a **Request for Comments (RFC)** system in 1969 to keep unofficial notes on the development of ARPANET as well as peer-reviewed official documents of Internet specifications, communications protocols, and procedures.[5] Today it is overseen by the **Internet Engineering Task**

3. Andrew L. Russell, and Valérie Schafer, "In the Shadow of ARPANET and Internet: Louis Pouzin and the Cyclades Network in the 1970s," *Technology and Culture* 55, no. 4 (December 3, 2014): 880–907. https://doi.org/10.1353/tech.2014.0096.
4. Russell and Schafer, "In the Shadow of ARPANET and Internet," 886.
5. The current list of Requests for Comment are available at their website.

Force (IETF), founded in 1986 by the **Internet Architecture Board (IAB)** as part of the United States Department of Defense's **Defense Advanced Research Projects Agency (DARPA)**. In 1992, it became a part of the **Internet Society (ISOC)**, an American nonprofit organization founded as part of a larger transition of the Internet away from being a United States governmental entity.[6]

> Examples of the Request for Comments (RFC) Process
>
> The **Hypertext Transfer Protocol (HTTP)** is one example of how open standards and protocols develop through the RFC process. Its initial development from 1989 to 1992 by Tim Berners-Lee and colleagues happened as a part of their work at CERN in Switzerland. It was first released publicly on **USENET** newsgroups alt.hypertext, comp.sys.next, comp.text.sgml and comp.mail.multi-media. It wasn't until 1996 that an informational RFC 1945, HTTP/1.0 was released. This was followed by a proposed standard, RFC 2068, HTTP/1.1 in 1997 and a draft standard, RFC 2616, HTTP/1.1 in 1999.
>
> For comparison, consider the earlier development of the **Simple Mail Transfer Protocol (SMTP)**. This protocol evolved through early drafts submitted in 1980 (RFC 772) and 1981 (RFC 780 and RFC 788). It became an internet standard with RFC 821 in 1982.
>
> In both examples, the original standard was iteratively extended as needed to fit contemporary Internet contexts.
>
> Take a few minutes to look at the abstracts of these RFC drafts and standards. Read through the introduction of RFC 2616, HTTP/1.1 (published in 1999) and compare it to the introduction and "The SMTP Model" from RFC 821 (published in 1982). How do these two RFCs differ? How are they the same?

In 1993 I joined the Neuronal Pattern Analysis group at the Beckman Institute, University of Illinois at Urbana-Champaign, as a post-doctoral researcher. One project I was a part of was exploring ways we could share raw data between research labs around the globe using the Internet. At the same time, the University's National Center for Supercomputing Applications was developing NCSA Mosaic, the first widely used graphical web browser. Participation as part of this research team seeking to collect, store, and share large raw datasets using HTML through our in-house NCSA Mosaic HTTP server meant spending considerable time poring through USENET discussions and through review of web page raw HTML, looking for examples as the protocol was still in early development. And sometimes it meant getting pre-notifications of changes through hallway conversations with those working on Mosaic.

And for me, this journey through systems for networking and internetworking took another radical turn when in 1995 I decided to test the waters outside of the tenure system and neuroscience to join the team developing Prairienet Community Network at the Graduate School of Library and Information Science, University of Illinois Urbana-Champaign.

6. Learn more about the launch of the Internet Society on their website.

The Free-net and Community Networking Movements

In 1995, an informational Request for Comments, RFC 2235, was submitted by Robert H. Zakon and the Internet Society. As noted in the introduction, "This document presents a history of the Internet in timeline fashion, highlighting some of the key events and technologies which helped shape the Internet as we know it today."[7] This includes a listing of many different national and international internetworking organizations and companies, including the **Cleveland Free-Net**, launched in 1986 by Tom Grundner of Case Western Reserve University.

Cleveland Free-Net used a community-based **bulletin-board system (BBS)** model in which community forums are hosted by a community server to facilitate the sharing of public information of local citizens online and launched a wider free-net movement. Free-nets made use of local voice-grade landline phone dial-ups to provide free or low-cost access from home, school, and library computers to a multiuser computer within a central location of a community. The phone was connected to a Modulator-Demodulator, or **modem**, that turned the text data provided from computing devices into the binary zeros and ones being used by that digital network through connection to a remote set of modems connected to the multiuser computer. The multiuser computer itself ran a range of server applications in addition to BBS software and also provided access to the Internet, including USENET with its store-and-forward framework, that together inspired much of the forums that became available through the emergence of the World Wide Web because of HTML and HTTP.

In a speech delivered to the University of Illinois at Urbana-Champaign in 1994, Grundner brought forward the term "cybercasting" to compare Free-nets to "broadcasting." He goes on to provide the following vision:

> Basically, it's the same service you find in radio or television networks. You might have, let's say, a local radio station here in Urbana and you might have your own radio talk show hosts or disc jockeys. But you might also be taking feeds from ABC Radio. Similarly, we have independent affiliates who operate their community computer systems, drawing upon local people, local information resources, and so forth. Then we try to supplement that with high-quality feeds from the network level—information services and features that supplement what they're able to do locally.[8]

For Grundner and the Free-net movement, community computing was about building stronger citizens through community conversations and information sharing, advances in local K-12 education and government two-way interactions, support for small- and medium-sized businesses who could not otherwise afford the services only major corporations could obtain, the agricultural community who are often left out of conversations with county officials and extension agents, and hundreds of local com-

7. Robert H. Zakon, "Hobbes' Internet Timeline," accessed July 16, 2020. https://tools.ietf.org/html/rfc2235.
8. Tom Grundner, "Seizing the Infosphere: An Alternative Vision for National Computer Networking," Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, 1994. http://hdl.handle.net/2142/365.

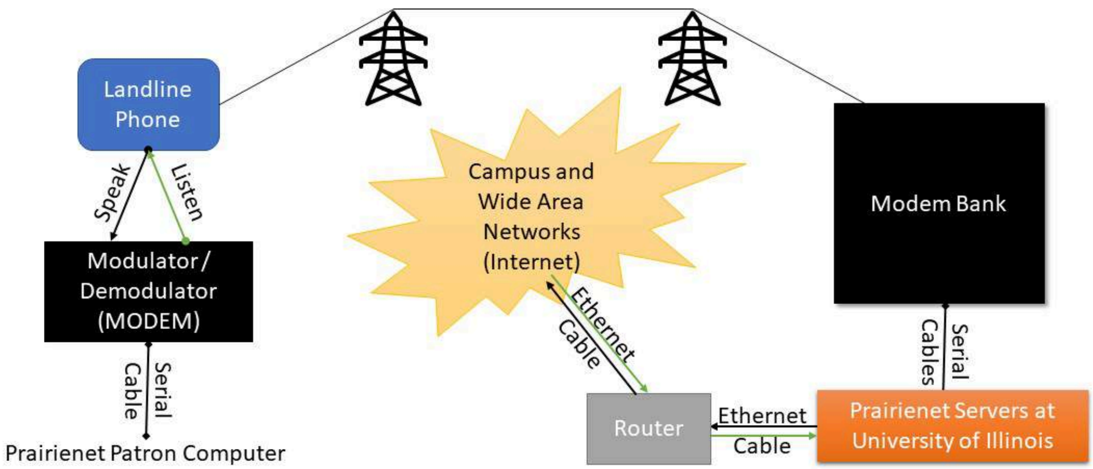
munity organizations. Further, while community media historically has made strong use of Public, Educational, and Government access (PEG) cable television channels to express concerns for local issues, engage in democratic debate, and deliver reliable access to information, art, and music, they too found this community computing movement of great value. And with its official grand opening in June 1994, this embodied the inspiration and dreams of Prairienet Free-net, later called Prairienet Community Network, which I joined in the fall of 1995. Other early networks also were being developed and embedded similar principles, including the Community Memory project from the late 1970s, FidoNet and other public bulletin board systems, and other early community networking systems including PEN (Public Electronic Network) and the WELL (Whole Earth 'Lectronic Link) in California.

> Community Networks
>
> Every community has a system of 'core values' that help maintain its 'web of unity'… A community network supports these core values by supporting the information and communications needs of smaller communities within the larger community and by facilitating the exchange of information between individuals within these small communities … and by encouraging the exchange of information among communities."
>
> Douglas Schuler, "Community networks and the evolution of civic intelligence"[9]
>
> Prairienet
>
> Prairienet is used to make community information available in electronic form and is also used as a free access point to Internet information resources… The fundamental question to be addressed by the research is, "to what extent is Prairienet serving its intended purpose as a community forum for education and information access?"
>
> Gregory B. Newby and Ann P. Bishop, "Community System Users and Uses"[10]

Free-nets and other community networks in part served an essential role at a time when Internet access remained the venue of research projects using the NSFNET-based Internet. In 1995, NSFNET reverted to a research network backbone linking super-computing centers, with United States WAN traffic now routed through interconnected commercial network service providers and backbone providers, opening up full commercial use of the Internet. Throughout the late 1990s and into the mid-2000s, core aspects of Free-nets and centrally-located community computing systems providing dialup access were replaced by commercial Internet Service Providers and larger web service providers. Any Free-nets that remained, such as Austin Freenet, changed their focus to technology training and access to residents of the city. Others, such as Prairienet, ended over a decade of service as

9. Douglas Schuler, "Community networks and the evolution of civic intelligence," *AI & Society* 25, no. 3 (August 2010). https://doi.org/10.1007/s00146-009-0260-z.
10. Gregory B. Newby and Ann P. Bishop, "Community System Users and Uses," *Proceedings of the ASIS Annual Meeting* (1996): 1-2. https://web.archive.org/web/20220213055143/https://petascale.org/papers/asispnet.pdf.

the host School of Information Sciences continues its work to foster community networking through graduate and undergraduate education, community inquiry and participatory action research, service-learning and design studios, and mentoring and internships. In so doing, a range of community-based stakeholders strive to bridge silos by coming together to innovate-in-use Internet-based tools for social and democratic participation. At their core, community networks remain a gathering of schools and libraries, of governmental and small and medium businesses, and of a range of community organizations addressing local issues for all sections of the community, thereby advancing community development for a more just society.



The Prairienet Free-Net. One modem is attached to the patron's computer. The University of Illinois School of Information Sciences hosted a bank of modems, each with a unique phone number, to which patrons could connect through phone lines.

The Commercialization of the Internet

During the early 1980s, some commercial vendors began to incorporate Internet protocols into their digital artifacts used on their competitive, private network services, as noted by some of the early Internet developers in their article "A Brief History of the Internet." Then in 1983, the United States Department of Defense separated from ARPANET to create its own separate network, MILNet. In 1986, the United States National Science Foundation Network (NSFNET) was created, establishing five super-computing centers to provide open high-computing to ARPANET users around the world. Locations included Princeton (John Von Neumann Center, JFNC), Pittsburgh (Pittsburgh Supercomputing Center, PSC), University of California, San Diego (San Diego Supercomputing Center, SDSC), the University of Illinois Urbana-Champaign (National Center for Super-computing Applications, NCSA), and Cornell (Theory Center). In 1988 an NSF-initiated conference at Harvard's Kennedy School of Government entitled "The Commercialization and Privatization of the Internet" built off of a growing number of initial three-day workshops to begin a broader conversation on the topic. This shortly led to the establishment of the first Interop trade show, started in September of that same year. Vendors also started attending periodic formal meetings of IETF meetings to discuss new ideas regarding Internet protocols. The late 1980s and 1990s also saw a strengthening of US intellectual property rights policies. This "pro-patent" posture opened up possibilities of university faculty-founded firms and establishment of new public-private partnerships.[11] Combined, this established a rich, mutually cooperative evolution of the Internet for a range of research, end user, and vendor stakeholders from a primarily military-funded research program to an internetworking suite available both publicly and privately. And so while other global models of internetworking faded, such as the French Cyclades Network which lost out to corporate monopolies, the US Internet model rapidly expanded through this public-private partnership, some of which was introduced through initial first use via local and regional community networks.

In 1995, the **United States National Science and Technology Council's (NSTC)** Committee on Computing, Information and Communications chartered the **Federal Network Council (FNC)**. The NSTC itself is a Cabinet-level council established as Executive Order 12881 in 1993 by President Bill Clinton early in his first term and is part of the Executive Branch's Office of Science and Technology Policy. The membership of the FNC consisted of one representative from each federal agency whose programs utilize interconnected TCP/IP Internet networks. Its charter was two-fold:

- to provide a forum for networking collaborations among Federal agencies to meet their research, education, and operational mission goals.
- to bridge the gap between the advanced networking technologies being developed by research FNC agencies and the ultimate acquisition of mature versions of these technologies from the commercial sector. This helps build the NII, addresses Federal technology transition goals, and allows the operational experiences of the FNC to influence future Federal research agendas.

> *RESOLUTION: The Federal Networking Council (FNC) agrees that the following language reflects our definition of the term "Internet." "Internet" refers to the global information system that — (i) is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions/follow-ons; (ii) is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols; and (iii) provides, uses or makes accessi-*

11. David C. Mowery and Timothy Simcoe, "Is the Internet a US Invention?—An Economic and Technological History of Computer Networking," Research Policy 31, no. 8 (December 1, 2002): 1386. https://doi.org/10.1016/S0048-7333(02)00069-0.

> *ble, either publicly or privately, high level services layered on the communications and related infrastructure described herein.*[12]

As we've worked through this textbook, we've explored the ever-present social shaping of technology in recognition that technologies are not, and can never be, neutral. The systems for internetworking are both deeply technical systems and also deeply social systems. They are sociotechnical artifacts and systems. They have been designed, developed, and implemented within a range of political, economic, social, and cultural structures. The outcomes of this shaping of sociotechnical design and implementation will positively impact and shape some but not others, and also potentially negatively impact and shape some. Remember that tinkering is a research process, one that incorporates various aspects of the methodological landscape. Without incorporating critical paradigms though incorporation of moral and ethical questioning to unmask and expose hidden forms of oppression, false beliefs, and commonly held myths, we continue to advance the giant triplets of racism, materialism, and militarism.

So let's finish this chapter by exploring a limited government, supply-side, market-oriented think tank that has also been one of the shapers of the sociotechnical Internet, one that has influenced both government policy and technology innovation very differently than that of the Free-net and community networking movement model.

In 1993 The Progress & Freedom Foundation, a nonprofit think tank, was launched. The first paragraph of its mission statement notes:

> The Progress & Freedom Foundation is a market-oriented think tank that studies the impact of the digital revolution and its implications for public policy. Its mission is to educate policymakers, opinion leaders, and the public about issues associated with technological change, based on a philosophy of limited government, free markets, and individual sovereignty.

In August 1994, the foundation released the publication "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age." In its preamble, authors Esther Dyson, George Gilder, George Keyworth, and Alvin Toffler begin by stating:

> The central event of the 20th century is the overthrow of matter. In technology, economics, and the politics of nations, wealth in the form of physical resources has been losing value and significance. The powers of mind are everywhere ascendant over the brute force of things.
>
> In a First Wave economy, land and farm labor are the main "factors of production." In a Second Wave economy, the land remains valuable while the "labor" becomes massified around machines and larger industries. In a Third Wave economy, the central resource a single word

12. Federal Networking Council, "FNC Resolution: Definition of 'Internet'," October 24, 1995. https://www.nitrd.gov/historical/fnc/internet_res.pdf.

> broadly encompassing data, information, images, symbols, culture, ideology, and values is actionable knowledge.
>
> The industrial age is not fully over. In fact, classic Second Wave sectors (oil, steel, auto production) have learned how to benefit from Third Wave technological breakthroughs just as the First Wave's agricultural productivity benefited exponentially from the Second Wave's farm mechanization.
>
> But the Third Wave, and the Knowledge Age it has opened, will not deliver on its potential unless it adds social and political dominance to its accelerating technological and economic strength.[13]

Soon after this publication, a range of scholars began introducing the terms technolibertarianism or cyberlibertarianism to explore the ways both the political left and right within the United States were beginning to incorporate these central Third Wave concepts of a Knowledge Age introduced within the Magna Carta. In "Cyberlibertarian Myths and the Prospects for Community," Langdon Winner highlights three key aspects within the Magna Carta:

1. technological determinism — changes that are inevitable, irresistible, and world-transforming;
2. radical individualism — ecstatic self-fulfillment and the putting aside of anything that might hinder the pursuit of rational self-interest;
3. supply-side, free market capitalism as formulated by Milton Friedman.[14]

UK media theorists Richard Barbrook and Andy Cameron explore how the US New Left movement that formed as part of '60s radicalism — collaboratives with campaigns against militarism, racism, sexual discrimination, homophobia, mindless consumerism, and pollution — and the US New Right — those resurrecting an older form of economic liberalism based on liberty of individuals within the marketplace — came together in a hybrid faith they termed the Californian Ideology. In "The Californian Ideology," they go on to underline:

"Crucially, anti-statism provides the means to reconcile radical and reactionary ideas about technological progress. While the New Left resents the government for funding the military-industrial complex,

13. Esther Dyson, George Gilder, George Keyworth, and Alvin Toffler, "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age," The Progress & Freedom Foundation, August 1994. http://www.pff.org/issues-pubs/futureinsights/fi1.2magnacarta.html.
14. Langdon Winner, "Cyberlibertarian Myths and the Prospects for Community," *ACM SIGCAS Computers and Society* 27, no. 3 (September 1, 1997): 14–19. https://www.langdonwinner.com/other-writings/2018/1/15/cyberlibertarian-myths-and-the-prospects-for-community.

the New Right attacks the state for interfering with the spontaneous dissemination of new technologies by market competition."[15]

More recent work by David Golumbia indicates there remains a strong political voice championed by both the political right and left regarding the liberatory nature of mass computerization and "digital practices whose underlying motivations are often explicitly libertarian." For instance, Golumbia states:

"We must not mistake the 'computer revolution' for anything like a political revolution as various leftist traditions have understood it. The only way to achieve the political ends we pursue is to be absolutely clear about what those ends are. Putting the technological means for achieving them ahead of clear consideration of the ends is not merely putting the cart before the horse; it is trusting in a technological determinism that has never been and will never be conducive to the pursuit of true human freedom."[16]

In considering the prospects for community given the broad adoption of cyberlibertarianism, Winner concluded by noting: "Superficially appealing uses of new technology become much more problematic when regarded as seeds of evolving, long term practices. Such practices, we know, eventually become parts of consequential social relationships. Those relationships eventually solidify as lasting institutions. And, of course, such institutions are what provide much of the actual framework for how we live together. That suggests that even the most seemingly inconsequential applications and uses of innovations in networked computing be scrutinized and judged in the light of what could be important moral and political consequences. In the broadest spectrum of awareness about these matters we need to ask: Are the practices, relationships and institutions affected by people's involvement with networked computing ones we wish to foster? Or are they ones we must try to modify or even oppose?"[17]

The last half century has seen a rich exploration of various sociotechnical artifacts and systems for digital networking and internetworking. These tools have provided opportunities to address a range of individual and social issues and meet a range of individual and social needs. The commercialization of the Internet has further opened up amazing possibilities for community networking and the advancement of stronger citizens through community conversations and information sharing, education and government two-way interactions, support for small- and medium-sized businesses, of agriculture, of community media, of Makerspaces, DIY and do-it-together culture, and of many, many more means for advancing civic intelligence. But the social shaping of technology also includes a range of political, economic, and social influencers that have problematically shaped the technologies and our visions for these technologies.

15. R. Barbrook and A. Cameron, "The Californian Ideology," *Science as Culture* 6, no. 1 (January 1, 1996): 56. https://www.tandfonline.com/doi/abs/10.1080/09505439609526455.
16. David Golumbia, "Cyberlibertarians' Digital Deletion of the Left," *Jacobin Magazine*, December 4, 2013. https://www.jacobinmag.com/2013/12/cyberlibertarians-digital-deletion-of-the-left/.
17. Winner, "Cyberlibertarian Myths and the Prospects for Community."

As Eduardo Villanueva-Mansilla notes in "Section *2.3. The early mirage, and the current desert*" of his 2020 First Monday paper "View of ICT policies in Latin America: Long-term inequalities and the role of globalized policy-making"[18]:

> Inequalities, both new and old, find a home on the Internet, and multiply happily. Commercial giants dominate the conversation and fail to pay their due. Public spheres become toxic and uncontrollable through the tools that exist on the Net[19]. Nonetheless, governments dedicate resources to promote access and discuss policy frames in order to provide an attractive environment to invest. This points to a divorce between the reality of the past 25 years of Internet development, and the imagination of its potential. How to deal with this divorce (or may we call it a divide?) requires an understanding of the ideological perspective behind most of the development/promotional efforts about the Internet. This is especially seen by those undertaken by the government themselves, as it has happened in Latin America and specially in Peru for the past decades.

But we do not stand at an "either/or" inflection point. Internetworking can happen in many shapes and forms which regularly incorporate the Internet suite of protocols and applications. The key is the level to which we enter into use of an Internet-based tool from a 'thing-oriented' compared to a 'person-oriented' framing. A dominant narrative is centered within 'thing-oriented' framings such as cyberlibertarianism. But as we work to open new critical lenses that help us see the 'person-centered' counterstories past and present, and enter into communities of practice across difference using collective leadership and community inquiry, we begin to dismantle the many webs of oppression within these sociotechnical systems while effectively using carefully selected aspects to achieve social justice outcome goals, as Judy Wajcman notes in a video segment on the impact of digital technology. [Watch the video "Gearty Grilling: Judy Wajcman on the impact of digital technology" online.]

Lesson Plan

In this session we begin our journey exploring the Internet and other digital internets which serve as the base of many of our Networked Information Systems today. The history of internetworking, the range of organizations that helped in its design, and the influence specifically of the United States in shaping and codifying these designs is important to remember as we move forward in the Rainbow Unit. The mutual shaping, past and present, is both significantly solidified within the black boxes that are our daily smart devices used within our smart homes and smart cities, and also highly fluid within our tinker, maker, and DIY innovation-in-use spaces.

18. Villanueva-Mansilla, E. (2020). ICT policies in Latin America: Long-term inequalities and the role of globalized policy-making. First Monday, 25(7). https://doi.org/10.5210/fm.v25i7.10865.
19. David Nemer, 2018. "The three types of WhatsApp users getting Brazil's Jair Bolsonaro elected: WhatsApp has proved to be the ideal tool for mobilizing political support — and for spreading fake news," Guardian (25 October), at https://www.theguardian.com/world/2018/oct/25/brazil-president-jair-bolsonaro-whatsapp-fake-news. Accessed April 25, 2023.

Essential Resources:

The essential resources for this extended session chapter are many and varied.

First, skim these in connection with related sections of this chapter, bringing in a critical mindset to tease this sociotechnical review apart a bit further as it relates to the broader themes of this textbook. With this as an entry point, carefully choose two or three of these to delve further into the ways we are not confronting our ignorance on the various conflicting impacts based on the social shaping of technologies, but rather the ways our hidden intelligent practices have impacted other individuals and communities.

- Histories of open models for internetworking
    - Russell, Andrew L., and Valérie Schafer. "In the Shadow of ARPANET and Internet: Louis Pouzin and the Cyclades Network in the 1970s." *Technology and Culture* 55, no. 4 (December 3, 2014): 880–907. https://doi.org/10.1353/tech.2014.0096.
    - Leiner, Barry M, Vinton G. Cerf, David D. Clark, Robert E Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. "A Brief History of the Internet." *ACM SIGCOMM Computer Communication Review* 39, no. 5 (October 2009): 22–31. https://doi.org/10.1145/1629607.1629613.
    - Zakon, Robert H. "Hobbes' Internet Timeline." Accessed July 16, 2020. https://tools.ietf.org/html/rfc2235.
- Free-nets and Community Networking
    - Grundner, Tom. "Seizing the Infosphere: An Alternative Vision for National Computer Networking." Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, 1994. http://hdl.handle.net/2142/365.
    - Peter Day. "Community Networks." Public Sphere Project. Accessed July 16, 2020. http://publicsphereproject.org/content/community-networks-1.
    - Schuler, Douglas. "Community networks and the evolution of civic intelligence." *AI & Society* 25, no. 3 (August 2010): 291-307. https://doi.org/10.1007/s00146-009-0260-z.
- Commercialization of the Internet
    - Leiner, Barry M, Vinton G. Cerf, David D. Clark, Robert E Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. "A Brief History of the Internet." *ACM SIGCOMM Computer Communication Review* 39, no. 5 (October 2009): 22–31. https://doi.org/10.1145/1629607.1629613.
    - Mowery, David C., and Timothy Simcoe. "Is the Internet a US Invention?—An Economic and Technological History of Computer Networking." *Research Policy* 31, no. 8 (December 1, 2002): 1369–87. https://doi.org/10.1016/S0048-7333(02)00069-0.
- The Magna Carta and Cyberlibertarianism

- Dyson, Esther, George Gilder, George Keyworth, and Alvin Toffler. "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age." The Progress & Freedom Foundation, August 1994. http://www.pff.org/issues-pubs/futureinsights/fi1.2magnacarta.html.
- More on the Progress and Freedom Foundation can be found on the foundation's website, which remains as an archive. Consider especially their "About" page, which includes their mission statement and statement of who they are, and list of "Supporters" as of 2009.
- Winner, Langdon. "Cyberlibertarian Myths and the Prospects for Community." *ACM SIGCAS Computers and Society* 27, no. 3 (September 1, 1997): 14–19. https://www.langdonwinner.com/other-writings/2018/1/15/cyberlibertarian-myths-and-the-prospects-for-community.
- Barbrook, R. and A. Cameron. "The Californian Ideology." *Science as Culture* 6, no. 1 (January 1, 1996): 56. https://www.tandfonline.com/doi/abs/10.1080/09505439609526455.
- Golumbia, David. "Cyberlibertarians' Digital Deletion of the Left." *Jacobin Magazine*, December 4, 2013. https://www.jacobinmag.com/2013/12/cyberlibertarians-digital-deletion-of-the-left/.

Additional Resources:

- Look for one or two current articles related to applications of the Sociotechnical Internet today. For instance, in May 2020, *The Intercept* published Naomi Klein's article "Screen New Deal: Under Cover of Mass Death, Andrew Cuomo Calls in the Billionaires to Build a High-Tech Dystopia." This article is the first installment in an ongoing series about the shock doctrine and disaster capitalism in the age of Covid-19, highlighting a continuing belief amongst tech billionaires and many others in Silicon Valley that there is no problem that technology cannot fix — and that this provides them a golden opportunity for deference and power that has been unjustly denied to them.

Key Technical Terms

- The **Federation of the Locals**, **End-to-End Principle**, **Store-and-Forward Principle**, **Peer-to-Peer Architecture**, **Client-Server Architecture**, **Request for Comments (RFC)** frameworks, and principles that are core to the Internet
- Basic components used to build a network, including:
  - Node identification using **Media Access Control (MAC)** and **Internet Protocol (IP)** address,
  - Network Interface Controllers (NICs) for wired **Ethernet Port**, wireless Ethernet

card, **Modem**, or Optical Network Terminals (ONT) for fiber optics devices.
    - **Media** used to interconnect devices, including coaxial copper, twisted pair copper, and fiber optics cables, and radio wave transmissions
    - The **Interconnect Device**, such as a wireless Ethernet (Wi-Fi) access point, wired Ethernet **Switch**, and the **routers** used to interconnect the different devices for each local network to create a larger network of networks
- The different levels of internetworking of multiple devices and of multiple networks of devices, including **Personal Area Networks (PAN)**, **Local Area Networks (LAN)**, Campus (CAN) and **Metropolitan Area Networks (MAN)**, and **Wide Area Networks (WAN)**
- Internet service provider technologies, including **Digital Subscriber Line (DSL)**, **Cable Internet**, **Cell-Based Internet**, Satellite-Based Internet, and **Fiber Optics Internet**
- The IP addressing and naming process and procedures, including **Domain Name System (DNS)**, the **Dynamic Host Configuration Protocol (DHCP)**, Network Address Translation (NAT), and Internet Protocols version 4 and version 6
- **Cloud computing** and its relation to the formal **specifications**, standards, protocols, and procedures of the **Internet Protocol (IP)**
- Troubleshooting tools, including **Link Lights**, **Ping**, **Traceroute**, dig, ifconfig/ipconfig, and speedtest. NOTE: You are encouraged to explore the Back Matter chapter Network Troubleshooting for more details on these.

Professional Journal Reflections:

In the Blue Unit chapter 2A: The Methodological Landscape we explored the assumptions that serve as a landscape for the three major research paradigms within Western science today. For those working within **positivist paradigms**, the purpose for research is to discover regularities and causal laws so that people can explain, predict and control events and processes. For those working within the **interpretive paradigms**, the reason for research is to describe and understand phenomena in the social world and their meanings in context. And for those within the **critical paradigms**, the reason for research is to empower people to change their conditions by unmasking and exposing hidden forms of oppression, false beliefs and commonly held myths. We've also discovered that the **deductive reasoning or logic** used within positivist paradigms is essential for tasks such as the writing of programming code which goes on to shape development of machine learning and artificial intelligence. As we saw in the Blue Unit, though, writing instructions integrated into education using **inductive reasoning** can lead to errors in the instructions which themselves need to be done using deductive logic as was seen in writing the first three notes of the middle-C scale as part of 2B: Make Music with Code.

Throughout the book, another core assumption is drawn from critical paradigms which acknowledge that any research is a moral-political and value-based activity which require us to explicitly declare and reflect on their value position(s) and provide arguments for their normative reasoning. As Richard

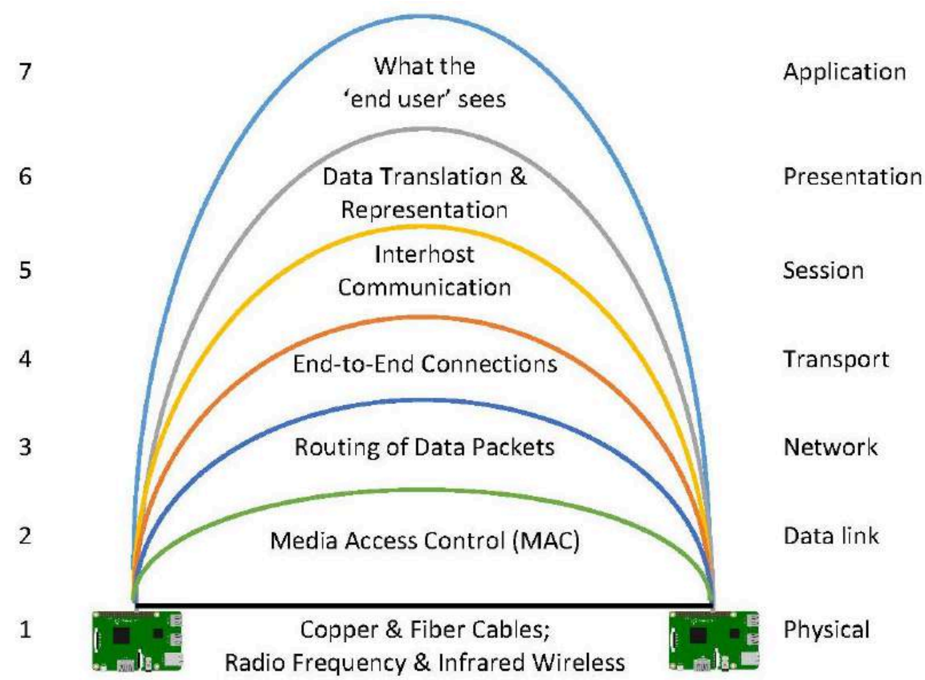
Milner notes, it is important within critical research for us to research the self, research the self in relation to others, to take part in engaged reflection to think what is happening in a particular research community, and to shift inquiry from self to system. I have found this to be relevant as much within tinkering as research as it is within formal research in academia and the business sector.

Thus, the goal of this week's Professional Journal Reflection is to help bring together reflections of our self into conversation with our inquiry into the Internet past and present to advance a more holistic understanding of networked information systems which is shaped by, and which shapes, the relation of self and others.

1. Using a positivist paradigm lens, note several key technical takeaways from the chapters and essential resources from this session of the Rainbow Unit.
2. Using an interpretivist paradigm lens, move from the simple nuts and bolts to note several evocative insights from this session of the Rainbow Unit.
3. From a critical paradigm lens and acknowledgement of a non-neutral stance, what mutual shaping has influenced the development and use of the Internet and major sociotechnical artifacts that are parts of our personal and professional lives? With our professional engagement with others around us near and far?
4. What first thoughts do you have on how we can work together to actively decodify key terms and concepts related to networked information systems to better understand ways in which the oppressive components within these can be countered moving forward?

2B: The Infrastructure of the Internet

Technical Overview



A visualization of the seven conceptual layers of the Open Systems Interconnection (OSI) model.

The hardware infrastructure of the Internet happens at layers 1 and 2 of the OSI model. Layer 1 provides the cable and radio wave **media** that interconnect devices, along with the **network interface controller (NIC)** installed within the computing device to which media connects. When formally connected to an **Internet Protocol (IP)** network the NIC becomes a **node** on the network. Layer two of the OSI model provides the identification mechanisms for the node. A computing device can have one or more than one NIC. For instance, your laptop may be simultaneously connected to a network with both a wired Ethernet media & NIC and a Wi-Fi media and NIC, and your smartphone to a cell radio wave media and NIC and also a Wi-Fi media and NIC. Each NIC is uniquely identifiable so that information is correctly disseminated to the appropriate device. To direct the flow of information between nodes, there must be an **interconnect device** or a combination of devices to facilitate communications. The only exception is when two nodes use the NIC, node identifiers, and media to do direct peer-to-peer communications.

Computer Network Building Blocks

There are *four main components* needed for a computer or other computing device to join as a node on a **local area network (LAN)** within a building such as a home, library, office, or cafe.

Node Identifier

Any device directly connected to the network that has been assigned a **unique identifier** on that network. Examples include:

- MAC address (also known as the hardware, physical, or Ethernet address): The serial number for Ethernet cards.
- IP address: The address used by the Internet protocol.

Network Interface controller (NIC)

The **hardware** necessary for a node to connect to a network. Examples include:

- Ethernet controller (wired Cat5/Cat6 or wireless/Wi-Fi): Used for LANs.
- MODEM (cable, DSL, dialup): Used for traditional internetworking with the Internet.
- Optical Network Terminal (ONT): Used for fiber to the home Internet.

Media

The **communications technology** used to connect nodes. Examples include:

- Copper transmits low-voltage electricity (e.g., wired Ethernet, DSL, cable, dialup).
- Fiber optic transmits light (e.g., fiber to the home).
- Radio waves (e.g., wireless Ethernet or Bluetooth).

Interconnect Device

A **device used to interconnect nodes**. Examples include:

- Switch or hub: Wired Ethernet LAN.
- Access point: Wireless Ethernet (Wi-Fi) LAN.
- Router or gateway: Builds an Internet by connecting different LANs together.
- Hotspot:
    a. A popular alternative term to access point on a Wi-Fi LAN; or
    b. A mobile device (e.g. Smartphone, Coolpad Surf, NetStick USB Modem, or MiFi 8000) that is both a cell-based wireless router and a Wi-Fi access point providing Internet access through the cell-based Internet connection.[1]

Nodes interconnect with other nodes in different ways, depending on how far they reach geographically, how many people are meant to use them, and who primarily owns or controls them. Some cover a very small area and may be used for very specific devices, while others are more general, cover larger areas, and are especially effective for use on the Internet.

Network Areas of Coverage

- **Personal Area Networks (PAN)** provide a simple computer network organized around a few personal devices, allowing the transfer of files, photos, and music without the use of the Internet or your home's local network. Two common examples would be your Bluetooth headset or keyboard and mouse. Depending on the Bluetooth range selected (or chosen for you), this could span three feet, ten feet, or one hundred feet. Beyond Bluetooth, other common PAN connectivities include Infrared (IR), USB, ZigBee, Wi-Fi, and radio frequency (RF, including short-distance AM and FM radio).
- The simplest type of Internet-based area network is a **Local Area Network (LAN)**. A LAN is a network with connected devices in a close geographical range. It is generally owned, managed, and used by people in a building. For example, connecting to a Wi-Fi network at a coffee shop or library would mean that your device would be a node on the café or library's publicly accessible LAN. Many public spaces may have a second,

1. Mobile Beacon and Mobile Citizen are providers for such devices, dedicated to serving schools, libraries, and nonprofits.

private LAN for use by staff only.

- A **Wireless LAN (WLAN)** is another name for a LAN used over Wi-Fi. In some cases, a router is used to strategically isolate the wired Ethernet LAN from the wireless Ethernet LAN and may therefore distinguish between the networks specifically using LAN versus WLAN connectivity.
- A **Metropolitan Area Network (MAN)** is a collection of LANs and devices in an area the size of a city. A version of a MAN is a **Campus Area Network (CAN)**, which would be a network the size of a college, organization, or business campus. These types of networks are typically community-owned and/or managed, and may also provide the infrastructure for one or more local/regional **Community Networks**.
- A **Wide Area Network (WAN)** covers the size of a state, country, hemisphere, or globe. A WAN is comprised of multiple MANs and/or LANs interconnected through a **backbone** or core transmission line or set of lines. The primary WAN we know of today is the **Internet**, a federation of local and regional networks interconnected through transmission lines typically owned and managed by one or more Internet service providers (ISP: the business that provides connections to each LAN), network service providers (NSPs: the business(es) that provide connections between ISPs), and backbone providers (the business(es) that provide the more extended connections between NSPs).

The backbone of the Internet, that part serviced by network service providers and backbone providers, is constructed using a **fiber optic** cable infrastructure. To carry signals rather than using electrical signals, glass fibers are used to carry light, with upwards of a thousand fibers being located within a single cable cladding. It is often the case that more fibers are included within a cable than are needed at the time of installation (called **dark fiber**) to allow for future growth without additional installation expense. Further, **Wave Division Multiplexing (WDM)** is used to allow multiple different wavelengths of light to be distributed on each strand of fiber (multiplexed) and then later separated (de-multiplexed), transmitting multiple communication streams simultaneously though a single light pulse. As technology continues to improve, replacement of multiplexers for newer models is allowing for still further data to be transferred over existing lines without additional installation expense of the cables themselves. The data itself is transferred using pulses of light transmitted using **light-emitting diodes (LEDs) or small lasers**. This can be done at very high speeds and over very long distances with less susceptibility to interference. A few different techniques are used to separate different wavelengths of light in ways that allow multiple communication streams, each at high frequencies, supporting higher capacity in addition to high frequencies. This opens up data transfer rates using fiber optics that are 20 to 1,000 times faster than copper cable and outdoor Wi-Fi Internet service and for a larger customer base. As Susan Crawford points out in her 2018 book, *Fiber: The coming tech revo-*

lution—and why America might miss it, "If the information-carrying capacity of copper wire is like a two-inch-wide pipe, fiber optic is like a river fifteen miles wide."[2]

Within the United States, most Internet service providers, on the other hand, make use of existing communication technologies developed for phone and cable television to also provide Internet access. Indeed, it has often been marketed as the "triple play," a discounted package providing these three at a discounted price compared to the purchase of each one individually from the provider, or from several different providers. In some cases, a provider primarily uses one technology, such as the cable Internet used by Xfinity/Comcast. On the other hand, depending on your geographic location, you can get Internet service from AT&T via copper digital subscriber line (DSL) or fiber optics Internet lines, as well as via radio waves through their wireless phone services.

Internet Service Provider Technologies

Digital Subscriber Line (DSL)

- Adds two channels to standard phone line for Internet
- Hub and spoke (dedicated line) topology; full duplex
- In the U.S., DSL prioritizes download speeds

Cable Internet

- Redirects a cable channel to be used for Internet
- Neighborhood shares bus topology; full duplex
- In the U.S., cable internet prioritizes download speeds

Cell-Based Internet

- 3G adds the EV-DO (Verizon, Sprint/Nextel) or HSDPA (AT&T, T-Mobile) protocol to cell voice's protocol
- 4G adds the WiMax (Sprint) or LTE (Verizon, AT&T) standard to cell's voice protocol

2. Crawford, Susan P. *Fiber: The Coming Tech Revolution and Why America Might Miss It* (New Haven: Yale University Press, 2018), 6.

- 5G is based on previous-generation cellular principles that use current low-band wireless frequencies with slower speed but longer distances[3]
    - 5G-NR (New Radio) can incorporate mid-band and millimeter wave (mmWave) high-band frequency bands.
    - mmWave provides the highest speeds but with a much shorter distance range, while mid-band balances data transfer speeds and distance covered.
    - 5G is capable of transferring data at ten to one hundred times the speed of 4G within these new radio frequencies; Non-Standalone (NSA) 5G allows for the combination of 4G infrastructure for voice communications and mmWave frequencies for increased data capacity
    - 5G upload speeds remain generally slower than download speeds, but have improved significantly over 4G, especially within the mmWave band being directed towards Internet of Things (IoT) applications.
    - Actual performance can vary significantly, as some packages provided by 5G providers for businesses will differ significantly from those provided to the general public, and information on availability and for whom can prove difficult to find.
- Equivalent to bus (shared) topology as radio signals can overlap; half duplex
- Applications of cellular services generally prioritize download speeds

Satellite Internet

- Indoor Unit (IDU) provides a modem connecting premises router to antenna dish (outdoor unit, or ODU).
- The very-small-aperture terminal (VSAT) dish antenna, which can also be used for satellite television service, requires a clear line of sight to facilitate microwave communications directly with the geostationary satellite serving the Internet provider, or to a shared Gateway Earth Station (gateway hub) that then connects to the satellite.
- Broadband speeds have improved considerably, with download speeds now reaching up to 40 Mbps. Download speeds are prioritized over upload speeds.
- Latency of signal, the delay between end node data transfers, is typically over 500 milliseconds. Wired Ethernet on a LAN is typically below 2 milliseconds; regional copper

3. See Nelson Machado Junior, "A Brief Introduction To 5G Technology: What you have to know about the 5th generation of cellular networks", Medium, January 21, 2021. Accessed November 16, 2022, https://medium.com/the-shadow/a-brief-introduction-to-5g-technology-b50c0f453f4

Internet latency is typically below 10 milliseconds. Latencies above 100 milliseconds can be problematic for some Internet applications, such as live stream conferencing and online gaming.

Community Wireless

- Uses standard wireless Ethernet (Wi-Fi) outdoors; anyone can use off-the-shelf equipment to create
- Equivalent to bus (shared) topology; typically half duplex
- Synchronous upload and download speeds

Fiber Optics

- Ultra-high-speed communications technology with one or more channels for Internet
- Hub and spoke (dedicated) topology; full duplex
- Synchronous upload and download speeds, ultra-low latencies

For most homes, community organizations, and small office/home office contexts, a **gateway router** is used that provides a WAN port used to connect the media leading to the first router of the Internet service provider. While sometimes this WAN port may need to first connect to a DSL/Cable **modem** or a fiber optics **Optical Network Terminal (ONT)**, in other cases, this interconnect device is integrated into the router. Typically, a gateway router will also incorporate both wired Ethernet switch and Wi-Fi access point interconnect devices for interconnectivity on the LAN side of the router. In addition, a gateway router typically integrates a **Dynamic Host Configuration Protocol (DHCP)** server that dynamically or statically assigns IP addresses to connected nodes on the LAN. The router will be configured to route essential Internet "phone book" type lookups to a designated ISP or third-party **Domain Name System (DNS)** server that contains a database of **public IP addresses** and **associated IP names**. All of these additional services facilitate its core function as the router between the LAN and the WAN.

We've worked through quite a few underlying concepts related to computer networks. Before moving into our first exercise, take a few minutes to review what we've already covered and also to get a glimpse at materials we'll be covering next by watching Carrie Anne Philbin's introduction to Computer Networks, Crash Course Computer Science episode #28.

Exercise: Listing Building Blocks of Your Computing Devices

Before moving on further, take some time to look at your own Local Area Network (LAN), whether it's the one in your place of residence, the LAN of a family or friend, or that of your workplace, library, community center, etc. To the extent possible, do this at a physical location where you can see the various network building blocks, and maybe do it with the person who took the lead in setting things up if you weren't the one who did that.

Location:	Note 1	Note 2	Note 3	Note 4	Note 5
Internet Service Provider:					
Communication Technology/Media Connection to Premises:					
Specified Upload/Download:					
Specified Monthly Data Caps:					
Additiona Notes:					
Gateway Router:					
Router WAN MAC Address:					
Router WAN IP Address:					
Additional MODEM/ONT?					
Additional integrated interconnect devices?					
Wireless 2.4 GHz Settings					
Wireless 5 GHz Settings					
LAN Switch Information					
LAN TCP/IP Setup					
Router IP Address					
Router Subnet Mask					
Router LAN MAC address					
Router as DHCP Server Starting and Ending IP Address					
Reserved IP Addresses / Device Names and MAC Addresses					
Port Forwarding Service Name(s), Port(s), and Internet IP Address(es)					
Security Access Control and Block Sites/Services					
Additional Notes:					
LAN Attached Device Name	Type	Media (e.g., WiFi, wired Ethernet)	MAC Address	IP Address (dynamic/static?)	Notes

A Local Area Network (LAN) documentation template.

Download the Excel spreadsheet template.

1. Begin with the information regarding the Internet Service Provider connecting the premises to the Internet. What type of communication technology and media is used to connect the premises to the Internet? What are the specified upload and download speeds for the service being provided? Are there any monthly data caps limiting use? As other things come to mind, also include notes on this information.

2. Next, explore the gateway router that provides the first hop between the WAN and the LAN you are documenting. To the extent possible, document the MAC and IP addresses of the router's WAN Network Interface Controller, and also that of the router's LAN NIC (remember, many devices have multiple network interface controllers, and therefore multiple node identifiers). Are there additional devices between the outdoor media and the router, such as a modem or optical network terminal? Does the router provide one or more LAN ports? Does it serve as a wireless access point? What are the settings for these? Does it provide a DHCP server giving IP addresses to other LAN nodes, and if so, what are the settings for this service? Are any IP addresses reserved for specific devices? Does it forward port requests that come to its WAN IP address to specific LAN node IP addresses? Are there any security access controls, block sites, and block services being used?
3. Finally, do all you can to document every node connected on this LAN, noting the type of device, the media being used, the assigned MAC and IP address, and any notes you think would be helpful to keep on record. Remember to that your laptop, printer, Raspberry Pi, and other devices might be connected through multiple means such as via WiFi and also wired Ethernet. You might also document devices that are connected to multiple different networks, for instance a smartphone that is connected to the LAN using WiFi and also to a cell network. And there may be devices like that smartphone that also serve as hotspot routers, connecting some devices to the Internet via the cell network instead of the LAN's ISP.

Key Takeaways

This exercise synthesizes concepts integral to computer and network building blocks. It also demonstrates an important principle regarding documentation. It's easy to forget some of this information until computer network trouble is encountered. Filling out this form and occasionally updating it before again tucking it away in an easy-to-access file folder can prove of significant value when troubleshooting an array of computer network issues.

Also, it is sometimes helpful to have these notes on hand and add to them strategically in combination with the Network Troubleshooting chapter.

More on IP Addresses and IP Names

When we type in a URL, or Uniform Resource Locator, in a web browser, we're almost always typing in an Internet Protocol name. Consider for instance the URL to this book:

```
https://iopn.library.illinois.edu/pressbooks/demystifyingtechnology/
```

Hypertext Transfer Protocol (HTTP) is the ever-present client-server protocol we have used for the last several decades to move information across the Internet. In this case, the first part, https, indicates the resource we're searching for is the secured HTTP protocol (HTTPS). The second part indicates the providing resource is the web server with name iopn.library.illinois.edu. The last two parts indicate the directory and subdirectory in which the specific resources being requested are located. Not specified is the specific file, which in this case probably defaults to index.html, index.php, or something similar.

But as with our phone system, the name doesn't truly get you in. Rather, the IP name needs to be associated with an IP address to pull up a web page, just as a person's or organization's name needs to be associated with a phone number to make a phone call. As of this writing, to get to the website, iopn.libary.illinois.edu is actually first converted to the IP address 130.126.162.192 in order to access the server. We can do this ourselves by typing in:

```
https://130.126.162.192/pressbooks/demystifyingtechnology/
```

Only IP names are converted to IP addresses. The directory and subdirectory listings use whatever characters were used to create those directories.

The Formation of IP Domain Names

The basic structure of the Internet came out of research launched in 1973 through funding from the U.S. Defense Advanced Research Projects Agency (DARPA). Researchers developed came a system of protocols known as the Transmission Control Protocol (TCP) and Internet Protocol (IP), or TCP/IP Protocol Suite.

In 1983, a conceptual framework for domain names was established through the RFC, in order to support the growing number of applications spanning multiple hosts, networks, and finally the Internet.[4] RFCs 883 and 973 expanded the domain name system (DNS) to build an intentionally extensible system. In 1987, two new RFCs made 882, 883, and 884 obsolete. These were "Domain Names – Concepts and Facilities, Request for Comments 1034" and Domain Names – Implementation and Specifications Request for Comments 1035.

These, too, have since had a range of RFC updates related to specific components of DNS: 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936, 8020, 8482.

Design Goals of DNS

The primary design goal of the domain name system (DNS) "is a consistent name space which will be used for referring to resources. In order to avoid the problems caused by ad hoc encodings, names

4. P. Mockapetris, "Domain Names – Concepts and Facilities," Network Working Group, November 1983, https://www.ietf.org/rfc/rfc882.txt.

should not be required to contain network identifiers, addresses, routes, or similar information as part of the name."[5]

Today, we have a range of top-level domains, some of which are based on organization type (e.g., .gov, .edu), geographic location (e.g., .uk, .es), or general category (e.g., .org, .com, .net, .site).

Individuals and organizations can apply for second-level domains they can then use on the Internet (e.g., illinois.edu, raspberrypi.org, adafruit.com, wolske.site).

Individuals and organizations can then create subdomains to extend their DNS tree to represent sub-groupings (e.g., ischool.illinois.edu, makecode.adafruit.com, martin.wolske.site).

Fortunately, another thing the RFCs for domain names came up with was a solution to the pesky phone system.

Consider that to call someone with our phone we need to know a series of numbers in order to dial them, employ our smartphone's contacts app, or create our own name/phone number listing system that informally maps a name we can remember to that series of numbers we need to dial.

By contrast, all IP names that are to be accessible need to be formally mapped within a DNS server. Each registered second-level domain runs its own local DNS server that holds the authoritative mappings. Top-level domain providers then run DNS servers that map the second-level domains, like illinois.edu, with the authoritative DNS servers for that domain. Internet service providers also run DNS servers that can be used to temporarily remember mappings for a set period of time, generally as defined by the second-level authoritative DNS servers. These are used by our local area networks so that we only have to type in illinois.edu in our web browser, and not 192.17.172.3.

IP Addresses

Internet Protocol addresses are the formal identifier of a node on a TCP/IP network. These addresses are used to route messages between source and destination across a network. Introduced in 1983, IP version 4 addresses use a 32-bit number broken into four 8-bit numbers separated by periods. When working in 8-bit binary notation, the decimal equivalent ranges between zero and 255. That is, an IPv4 IP address can range from 0.0.0.0 to 255.255.255.255. Protocols and policies have been developed to provide clear guidance regarding these addresses.

Almost all IP addresses, ones such as 192.17.172.3, are publicly accessible over the Internet. While some are not formally mapped to domain names widely known across the Internet, and some have strong security measures to restrict access, these numbers all can work across the Internet as needed/desired. For this reason, any router that is publicly available over the Internet (linking a Local Area Network

5. P. Mockapetris, "Domain Names – Concepts and Facilities, Request for Comments 1034," Network Working Group, November 1987, https://tools.ietf.org/html/rfc1034.

to Wide Area Networks) must have one of these public IP addresses. The router you use at your home, office, or other organization to connect to the Internet through an Internet Service Provider is typically assigned one of these public IP addresses. Often, it's only given to you temporarily, and may change dynamically in structured or semi-structured ways. But for those needing to ensure reliable access to nodes, for instance to web or database servers, you might purchase a static IP address so that you can set up routing information in a DNS server.

In the late 1990s, the Internet began running into the limits of IP addresses in version 4. As a 32-bit number, the maximum number of addresses available was 4,294,967,294. While that seems like a lot, given an increasing number of people have several different Internet "smart" devices in addition to their own laptop, four billion addresses isn't nearly enough. These protocols were given out in formal ways that suited the 1970s' and 1980s' understanding of the limited uses of the Internet—a far cry from what really evolved. IPv6 was ratified in 2017 but has yet to be fully implemented. It uses 128-bit, allowing 3.4×10^{38} possible addresses.

Private IP Addresses

In creating the Internet Protocol, there were several blocks of IP addresses that were made private. Anyone can have access to any of these without any required registration of them. The only caveat is that they are meant for use on private networks and cannot be routed through the public Internet.

The Internet Engineering Task Force (IETF) directed the Internet Assigned Numbers Authority (IANA) to reserve the following Internet Protocol Version 4 (IPv4) address ranges for use on private networks:

IP address range	Maximum number of addresses available to a single Local Area Network
10.0.0.0 – 10.255.255.255	16, 777, 216
172.16.0.0 – 172.31.255.255	1, 048, 576
192.168.0.0 – 192.168.255.255	65, 546

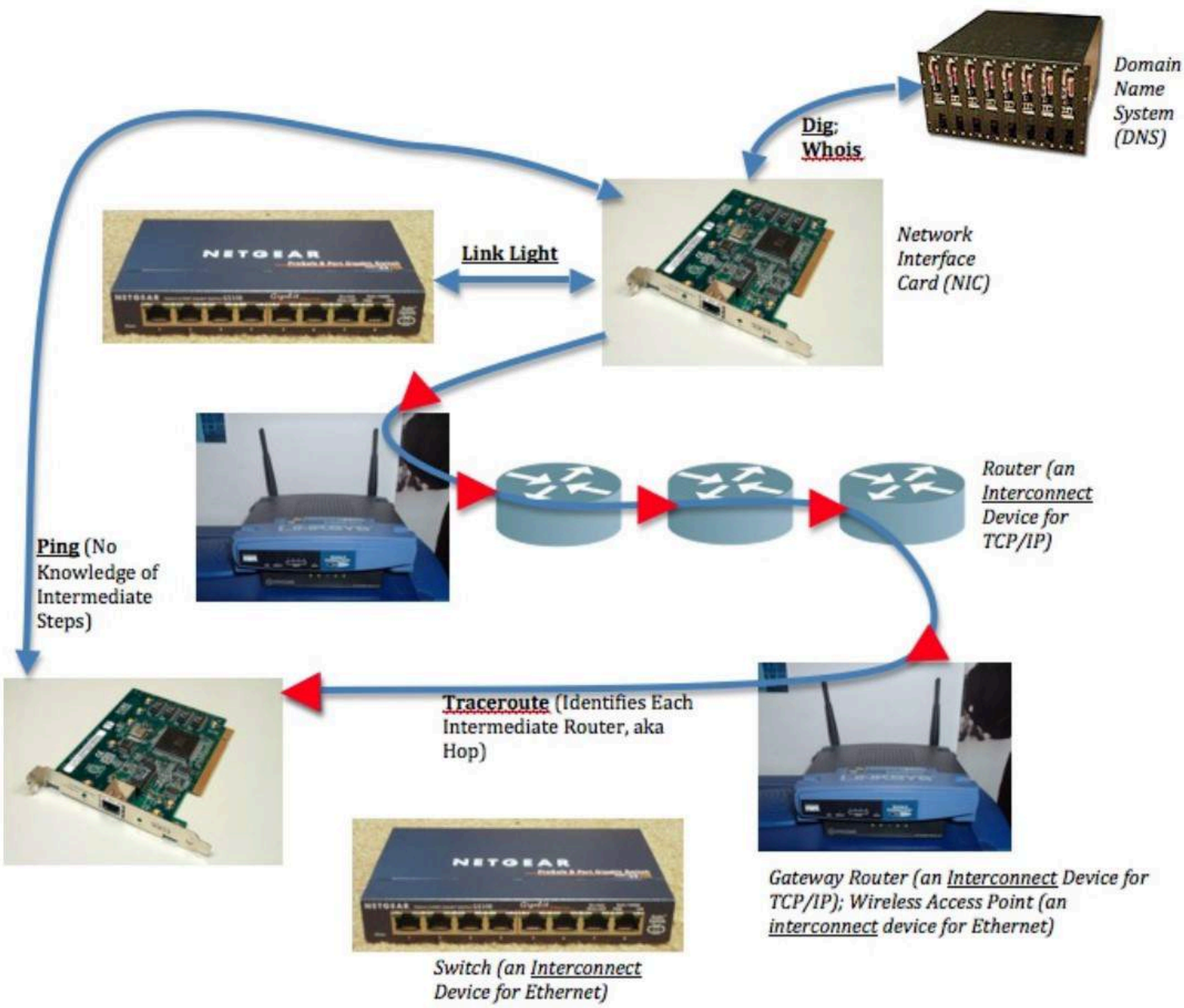
Private IPv4 addresses are widely used today. They allow a home or organization to create personal private networks for internal use and then set up routers to translate traffic (NAT, or Network Address Translation) meant to pass between that private network and the public Internet. Or more likely, when you purchased Internet access through an Internet Service Provider, the router they purchased came set up with a Dynamic Host Configuration Protocol (DHCP) server. This router hands out private IP addresses to nodes like your laptops, desktops, phones, and printers, or to those connecting to the router via WiFi. That router also is a NAT doing the network address translation to its Wide Area Network public IP address assigned to it by that Internet Service Provider.

Take a few minutes to [watch Carrie Anne Philbin's introduction to the Internet, from Crash Course Computer Science episode #29]().

Exercise: Internet Detective

When the Internet is working at expected levels, we generally don't think about the extensive collection of sociotechnical artifacts needed for one node to communicate with another node on a local network, let alone the many more that allow us to connect when the end-points cross the globe. When things are less than optimal, our responses vary from stepping out for coffee in passive acceptance, to pangs of guilt that we must be doing something wrong, to anger that nothing can be done when essential services are being lost.

This exercise equips us with sleuthing tools as we work to demystify the Internet further.



A network flowchart highlighting connection points at which different troubleshooting techniques apply.

As explored in the extension chapter on Network Troubleshooting, there is a range of network troubleshooting tools such as ping, traceroute, and speedtest. And there are a couple of tools that are especially helpful when sleuthing IP names: whois and dig. These tools are often installed or available to be installed on different operating systems. They are also integrated into various webpage tool sets.

Before beginning, we must update and upgrade the Raspberry Pi operating system. This is necessary for The General Purpose Raspberry Pi Web Server exercise later, so let's get a head start. At the same time, we can install whois and dig in the operating system. The advantage to this is ensures the following exercises are completed in a consistent manner.

Enter these commands to update the Raspberry Pi and install the following packages. Note that each of these commands start with the word "sudo." This indicates the command that follows (such as apt) is issued as a superuser: a user with administrative privileges. As you go, you may periodically be required to enter "y" for yes, or hit "q" after reading upgrade information, before you can proceed with the upgrade.

pi@raspberrypi:~ $ sudo apt update

pi@raspberrypi:~ $ sudo apt upgrade

pi@raspberrypi:~ $ sudo apt install dnsutils

pi@raspberrypi:~ $ sudo apt install whois

Take a few minutes to test the strategies for tracking down problems and identifying if there's something we might do about them in Network Troubleshooting. Take a close look especially at traceroute. What can the following tools tell you about a network?

- link lights
- ifconfig/ipconfig
- ping
- traceroute
- speedtest

Let's do a quick initial exploration with the traceroute command to see where the Adafruit.com website might be located physically:

pi@raspberrypi:~ $ traceroute adafruit.com

For those running the traceroute command in a Windows PowerShell terminal, you'll type in tracert adafruit.com instead.

```
pi@raspberrypi:~ $ pi@raspberrypi:~ $ traceroute adafruit.com
traceroute to adafruit.com (104.20.38.240), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  4.810 ms  5.073 ms  4.903 ms
 2  23.249.47.10 (23.249.47.10)  10.308 ms  10.140 ms  15.658 ms
 3  * * *
 4  * * *
 5  ce-0-8-0-3.r08.chcgil09.us.bb.gin.ntt.net (168.143.228.224)  14.359 ms  14.521 ms
 14.577 ms
 6  ae-8.r07.chcgil09.us.bb.gin.ntt.net (129.250.2.221)  14.321 ms  10.104 ms  11.020
ms
 7  ae-0.cloudflare.chcgil09.us.bb.gin.ntt.net (140.174.28.242)  11.067 ms  9.967 ms
16.920 ms
 8  104.20.38.240 (104.20.38.240)  10.983 ms  11.173 ms  11.238 ms
pi@raspberrypi:~ $
```

I see a first hop that takes me to my gateway router on my LAN, and then the first router of my ISP. A few lines down, I see I've reached ntt.net. A quick search on my web browser, typing in ntt.com in the search bar, takes me to NTT Communications, a Global IP network. In my search, it actually defaulted to Japanese which I had my browser translate, indicating this is likely a Japanese-owned corporation providing backbone service in the United States. The 5th hop also includes "chcgil09.us" within the IP name, which suggests the router is probably in Chicago, Illinois, United States. This would make sense as I live and am doing this traceroute just a couple hours south in Champaign, Illinois. Hop seven takes me to an IP address that includes the name "cloudflare" before finally reaching 104.20.38.240, the IP address associated with the IP name adafruit.com as listed at the top of the traceroute.

```
pi@raspberrypi:~ $ pi@raspberrypi:~ $ traceroute adafruit.com
traceroute to adafruit.com (104.20.38.240), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  4.810 ms  5.073 ms  4.903 ms
 2  23.249.47.10 (23.249.47.10)  10.308 ms  10.140 ms  15.658 ms
 3  * * *
 4  * * *
 5  ce-0-8-0-3.r08.chcgil09.us.bb.gin.ntt.net (168.143.228.224)  14.359 ms  14.521 ms
 14.577 ms
 6  ae-8.r07.chcgil09.us.bb.gin.ntt.net (129.250.2.221)  14.321 ms  10.104 ms  11.020
ms
 7  ae-0.cloudflare.chcgil09.us.bb.gin.ntt.net (140.174.28.242)  11.067 ms  9.967 ms
16.920 ms
 8  104.20.38.240 (104.20.38.240)  10.983 ms  11.173 ms  11.238 ms
pi@raspberrypi:~ $ dig -x 23.249.47.10

; <<>> DiG 9.10.3-P4-Raspbian <<>> -x 23.249.47.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 42018
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;10.47.249.23.in-addr.arpa.     IN      PTR

;; AUTHORITY SECTION:
47.249.23.in-addr.arpa. 384     IN      SOA     ns3.iTV-3.com. hostmaster.iTV-3.com. 2
019021200 28800 7200 604800 86400

;; Query time: 10 msec
;; SERVER: 10.0.0.1#53(10.0.0.1)
;; WHEN: Sat Jul 18 17:32:21 UTC 2020
;; MSG SIZE  rcvd: 114

pi@raspberrypi:~ $
```

One thing that isn't fully clear is the details regarding the second hop of the traceroute, the one that leaves my premises and takes the packets of data to my ISP. This is a place where the 'dig' command can sometimes prove very helpful. Before moving on to explore the use of that command, above is a snapshot of a search I did indicating that second hop was to iTV-3.com, the leaser of the fiber optics municipal area backbone network owned by the cities of Champaign and Urbana in collaboration with the University of Illinois Urbana-Champaign.

Once installation for dnsutils (which include the dig command) and whois is complete, proceed. Let's learn more about the dig command, starting with a search of the manual command within Linux before moving on to do a couple of searches specific to the Adafruit.com IP name and IP address.

From the Raspberry Pi command line, type:

pi@raspberrypi:~ $ man dig

Dig (domain information groper) performs Domain Name System (DNS) lookups. It's a flexible tool which can take some time to understand fully. After skimming through the manual entry, let's do a couple of quick tests of the command.

pi@raspberrypi:~ $ dig adafruit.com

Here, we see within the "QUESTION SECTION" that we're looking for type 'A' information. In DNS, 'A' records store the 32-bit IPv4 address(es) associated with a hostname. Here, we see the IP name adafruit.com actually has been assigned two, 104.20.38.240 and 104.20.39.240.

In a web browser, type in first one, then the other, of these IP addresses. What do you get as a web-page? What does this tell you?

Let's do a reverse lookup to get the 'PTR', a pointer from an IP address to the associated canonical name.

pi@raspberrypi:~ $ dig -x 104.20.38.240

Here we see in the "AUTHORITY SECTION" that ARPA provided 'SOA', that is, the start of a zone of authority record, in which this IP address is associated with dns.cloudflare.com, the authoritative Domain Name Server for cloudflare.com. According to their "About" page, Cloudflare is a "service that protects websites from all manner of attacks, while simultaneously optimizing performance."[6] While we don't know specifically how, we do know that Adafruit Industry is associated with, or makes use of in some way, Cloudflare.

Let's now use the whois command to see if we can learn more about Adafruit and Cloudflare. From the command line, type:

6. "About Cloudflare," Cloudflare, accessed July 18, 2020, https://www.cloudflare.com/about-overview/.

pi@raspberrypi:~ $ whois adafruit.com | less

(NOTE: the pipe symbol, found on the upper right between the "enter" and "backspace" keys on US keyboards, takes the output from one command and passes it to the next command, in the case "less" which allows us to view the contents one page at a time, moving back and forth within the text.)

Here, we see that the domain "adafruit.com" is registered through NameCheap, Inc. The name was created in May 2005, was last updated July 2018, and will expire May 2026 — IP names are not owned, but only leased.

We then find that the Domain Name Server for the domain name "adafruit.com" is hosted by Cloudflare.com. Adafruit, Inc. may still host the website in-house, or may use another service to do the hosting using an Infrastructure-as-a-Service or Platform-as-a-Service "cloud" web server. From this, the only thing we do know is that Cloudflare is performing as a Domain Name Server for Adafruit. The DNS system actually has quite a range of record types that can be effectively used to support a wide mix of IaaS, PaaS, and in-house infrastructures simultaneously in support of one domain name. As a result, Adafruit.com can actually be making use of a number of different in-house and remote-located services.

Use the up and down arrows to explore further the Registrant, Admin, Tech, and Name Server information records for the Adafruit.com domain name. Then let's do the same for Cloudflare.com by typing:

pi@raspberrypi:~ $ whois cloudflare.com | less

We do see some information regarding creation, updated, and expiry dates, and also the Name Sever for the domain. But we also see that while early in the establishment of the Internet Protocol all records were kept open, today many items can be held private from the public. So we see far less in whois for Cloudflare than we did for Adafruit.

Before moving on, let's take a quick look at the whois manual:

pi@raspberrypi:~ $ man whois

This description specifies this application searches for an object in the Request for Comments (RFC) 3912 database. Towards the bottom of the manual, NOTES are listed clarifying the search process and some of the underlying official resources searched depending on context.

Key Takeaways

The tools `traceroute`, `dig`, and `whois` can provide us with a considerable amount of information regarding the Internet backbone and the different end-point entitities that comprise the social and technical infrastructures that for the most part we just think of as a website. From the LAN of the personal computer running the web browser to the LAN of the computer or system of computers running the web server, there is an array of other LANs running the Domain Name Systems used to go from IP names to IP addresses, and then to specific server LAN locations that may vary depending on personal computer LAN geographic location, and then to various performance and security services, such as that from Cloudflare, that are used to further advance performance, and the many other systems and services hosted on still other LANs, only some of which are using HTTP web services. This Internet web, something that happens out of sight and mind, transfers untold packets of data around the globe every millisecond, coming back to our personal electronic devices as web pages, email, and audio/video communications in what appears to be a single, consolidated information artifact. What we don't see is the wealth of sociotechnical artifacts all influencing the shaping of this single information artifact we have received or transmitted.

Take a few minutes to check out Warriors of the Net. You can see a unique visualization of packets, routers, and even a guest appearance of the Ping of Death.

Digging Deeper

From Adafruit to Raspberry Pi: We used several tools to explore the hops to Adafruit.com, the IP address of Adafruit, the registrar used to lease the IP name and the Domain Name Server used to associate the IP name with an IP address, and other bits and bobs about the provider of core parts of the toolkit used for hands-on exercises in this textbook. Consider repeating this now to explore further raspberrypi.org. How does this compare and contrast with the adafruit.com exploration? Is there anything new that you learn from this parallel exploration?

The Author's Site: You can find my blog, http://martin.wolske.site, a subdomain under my leased domain name wolske.site. This domain name is hosted by an Infrastructure-as-a-Service (IaaS) provider, Dreamhost. As the iSchool was winding down Prairienet as a regional community network web-hosting service in the early 2000s, we searched for alternative web-hosting services for our non-profit patrons. Dreamhost not only provided web hosting, they also provided domain name registration, both at no cost for 501(c)(3) non-profits. As we helped move many non-profits to Dreamhost, I also moved my own website there and pay a yearly fee so that I can continue leasing the domain name wolske.site and also provide a WordPress service for subdomain martin.wolske.site. I have a second website, http://apcg.wolske.site, that currently redirects to the IOPN home for this book. In the coming years I may again deactivate this redirect and reactivate Dreamhost hosting of a WordPress site and using a PressBooks theme to allow testing of new chapter sections for a 3rd edition if I so choose. What can you learn about these two subdomain websites and the IaaS through the use of traceroute, dig, and whois in addition to the things you can learn by using a web browser to go to these sites?

São Tomé é Príncipe, Africa: I've valued my time doing participatory action research community inquiry projects in São Tomé é Príncipe, Africa, with citizens of this island nation, to advance their community cultural valued beings and doings. We found it technically impossible to set up community wireless on the island because of volcanic deposits that significantly interrupted Wi-Fi signals. We also found a nation that valued analog interactions within the marketplace and that made use of the public and community radio stations available. And their ISP provided wired broadband that valued upload speeds over download speeds so as to bring forward the information of people to others instead of focusing only on centralized corporate information sources being brought down to the people. But to bring this all to a broader audience, the official website of the nation is not located on the island. Where is it located? Who oversees this website?

To do this last Internet detective dig, consider the different top-level domains that might be used in association with this nation. Within the United States, where the Internet was birthed from ARPANET, we would use the ".gov" top-level domain name for a government, and ".edu" top-level domain name for an educational institution. But other nations needed to either apply to the United States for second-level domains associated with a top-level domain (which they couldn't do for .gov but could do for .com), or they needed to use their national top-level domain first (e.g., .co.uk specifies a .com site within the United Kingdom). For this detective task, try out different subdomains, such as stp.gov.st (the site listed within the Wikipedia listing for São Tomé), saotome.st, and saotome.org. Which are trusted sources of information about the nation, if any? Who hosts these websites? Who manages them? What is left unknown through the use of these tools?

Cloud Computing

When discussing Internet-based applications today, the term cloud or cloud computing is often used. Indeed, sometimes the word cloud is used synonymously with the word Internet. Cloud computing, like the Internet of Things, is an evolving paradigm. For this reason, in 2011 the National Institute for Standards and Technology, as part of its statutory responsibilities under the Federal Information Security Management Act of 2002, developed a short document highlighting important aspects of cloud computing. The goal was to provide a baseline for further discussion regarding cloud computing and as a means for comparison of cloud services and deployment strategies. Essential characteristics include on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. Together, these provide a means by which multiple organizational, community, or public consumers to which these services are deployed can each have great individual flexibility and freedom to unilaterally adjust services to fit current and anticipated demands across a range of devices. These essential characteristics and deployment models are associated with one of three service models:

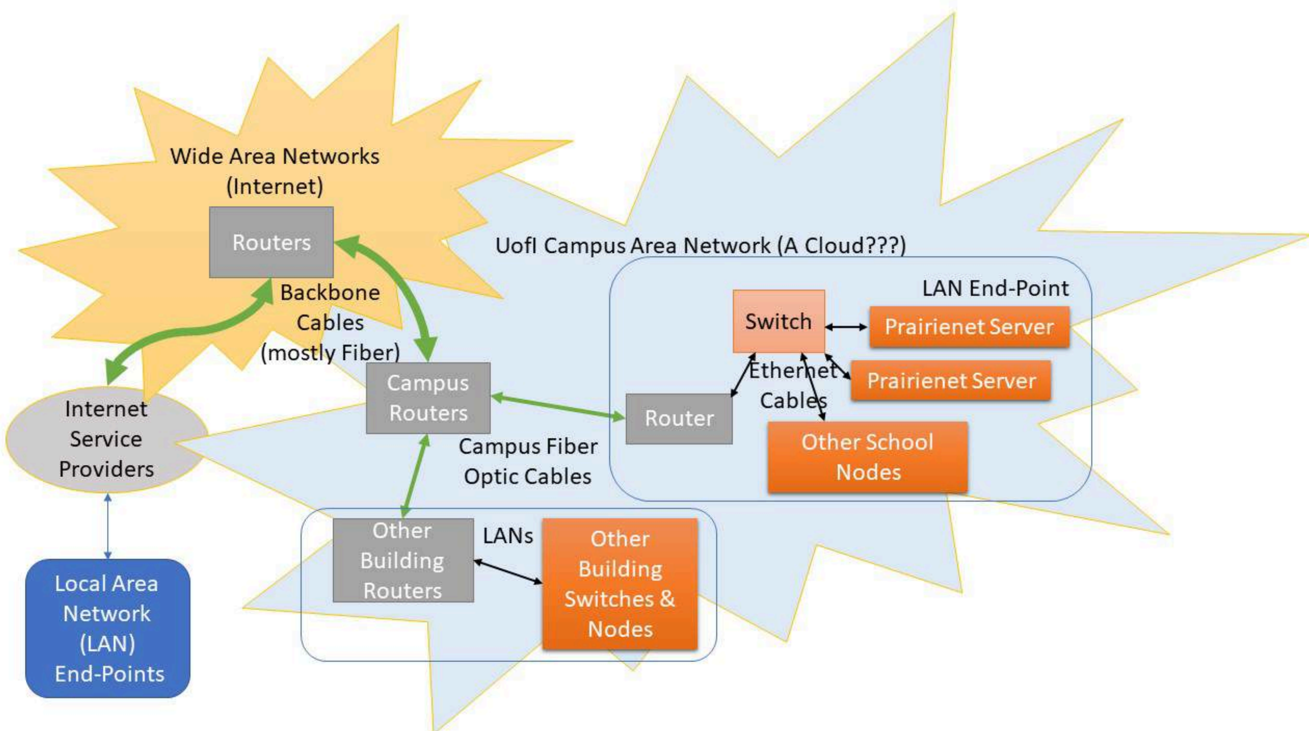
- **Software as a Service (SaaS)**: The consumer runs applications running on a cloud infrastructure.
- **Platform as a Service (PaaS)**: The consumer uses the cloud to deploy applications

they have created or acquired and that make use of programming languages, libraries, services, and tools provided and/or supported by the provider as part of their platform.

- **Infrastructure as a Service (IaaS)**: The consumer is provided a base of computer resources such as processors, storage, and networks upon which they can deploy and run software.

Others have noted that the distinction between higher-level Platform and lower-level Infrastructure as a Service found within a large data center is not a crisp line and should be considered together as *utility computing*.[7]

What is not generally recognized in relation to Cloud Computing is that, as part of the Internet with its foundational concepts of the end-to-end protocol creating a federation of the locals, data centers themselves are housed within their own Local Area Networks (LANs), some of which may be located within broader corporate Campus Area Networks (CANs). As part of a federation of the locals, these data centers should not be seen as centralized servers with overriding authority control, but rather



A visualization of multiple LANs within a Campus Area Network and connected to remote LANs via the Internet Wide Area Network.

7. Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, et al., "A View of Cloud Computing," *Communications of the ACM* 53, no. 4 (April 2010): 50–58. https://doi.org/10.1145/1721654.1721672.

local nodes internetworked with other local nodes to provision hardware and software services, a concept that was underlined within the Essential Characteristics section of the NIST Definition of Cloud Computing.

Wouldn't it therefore be accurate to consider the research servers (e.g., Prairienet Community Network server computers), the University library servers, and University campus infrastructure servers that serve on- and off-campus associates a Cloud service?

Wrap Up

As noted earlier in this session, internetworking can happen in many shapes and forms which regularly incorporate the Internet suite of protocols and applications. The key is the level to which we enter into use of an Internet-based tool from a "thing-oriented" compared to a "person-oriented" framing. Through the exercises within this part of the session, we've made use of different research methods to explore key social and technical aspects of the Internet suite, seeking to decodify key terms and concepts related to networked information systems.

I have been part of the iSchool at Illinois since 1995, where "we believe in the power of information to change the world," as noted at the top of our 2022-2027 Strategic Plan. The strategic plan goes on to highlight our innovative research at the intersections of inquiry and engagement, that includes interdisciplinary partnerships across broad communities. Internetworking happens both within the community and through the cloud, across Internet of Things devices on a local area network and local end-point devices across the Internet, within local servers and on server farms. Socioeconomic and sociopolitical frameworks have been introduced over the years, such as through the The Progress & Freedom Foundation and in publications such as "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age," working to shift to an exclusive supply-side, free market capitalism rooted within technological determinism and radical individualism. Emphasis is placed on removing individuals from their information activist communities, and instead into positions of consumption of data and information resources provided by platforms and services in the Cloud. Bringing together topics introduced in the first two sessions of the Rainbow Unit, we can now see Neil Gershenfeld's championing the bringing together the virtual world of bits and the physical world of atoms in innovative ways, through the Fab Lab movement and Internet of Things devices and through his opposition to the Bitnet of Things as the supply-side, free market solution to the grassroots Internet of Things alternative. For some economic and political framings, accelerating technological and economic strength can only happen through social and political dominance that focuses on this unique free market capitalist form. But as I've presented various aspects of the Rainbow Unit nationally and internationally, there have been a number of companies and community organizations that have provided their vigorous affirmation of the need for work at the intersection of inquiry and engagement beyond such a singular, hyper-individualized socio-political and economic lens.

If we are to address the opportunities and urgent needs of the day, there is a need for both community and corporate stakeholders to join in collective leadership. It may start through service and outreach, but needs to actively work beyond these to engagement that is reciprocal and mutually beneficial. Further, this requires the voices of diverse groups across cultures and disciplines. And as we've explored throughout this book, that requires inquiry that brings in the voices of the marginalized and oppressed. This can often start through works such as social justice storytelling, using story data, information, knowledge, and wisdom to counter the dominant think-oriented narrative through concealed stories to respond to stock stories, resistance stories to highlight injustices, and emerging and transforming stories to (re)construct knowledge built on concealed and resistance stories. But as we move forward within the Rainbow Unit, we'll also see the importance of such inquiry and engagement to move into all realms of creative endeavors. This is true now more than ever, as noted within the iSchool at Illinois's strategic plan's "Intersections of Inquiry and Engagement" section:

> "Information in the form of data underpins vast initiatives, including artificial intelligence (AI), machine learning, data science, big science, and more. Poor quality data, poorly integrated and conserved data, and unacknowledged data biases can corrupt these efforts."

As we've explored in this session, the infrastructure of the Internet provides an amazing opening, but also some significant hurdles, for us to build from "person-centered" digital internets past and present, and to also open new critical lenses to further challenge the dominant "thing-oriented" framings through new pathways as we move forward within the information sciences.

Comprehension Check

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with answers, see the online version.]

1. A network that covers the size of a state, country, and could even be considered to include the entire Internet.
    1. Internet Protocol (IP)
    2. Local Area Network (LAN)
    3. Campus Area Network (CAN)
    4. Wide Area Network (WAN)
2. A network with connected devices in a close geographical range. It is generally owned, managed, and used by people in one building.
    1. Internet Protocol (IP)
    2. Local Area Network (LAN)
    3. Wide Area Network (WAN)
    4. Campus Area Network (CAN)

3. Match the following terms with their definitions: **Interconnect Device**, **Transmission Control Protocol (TCP)**, **Dig**, **Hypertext Transfer Protocol (HTTP)**, **Node**, **Internet Protocol (IP)**, **Traceroute**

Definition	Term
Any device directly connected to the network that has been assigned a unique identifier or address.	
The ever-present application layer protocol used to request and access media (such as webpages) from across the Internet.	
Examples of this include switches, routers, and gateways.	
A key communications protocol for routing data within networks, thus enabling the Internet. This protocol delivers packets from a sender to the destination and requires IP addresses for routing these packets.	
It specifically defines how data intended to be send over a network is broken into packets before transmission and reliably reassembled in the right order at the destination application.	
A tool to test the connection between two devices on a network, testing each intermediate connection between the two.	
A tool used to search through Domain Name System (DNS) servers for various DNS records. As such, it can be very useful for troubleshooting public IP name and IP address problems.	

3A: The Digitization of Divides

Background Knowledge Probe

Across the journey through the Orange, Blue, and now Rainbow units, we've worked to increasingly advance a more holistic and nuanced understanding of both the multiple social layers and the multiple technical layers of networked information systems. Small decisions can lead towards large outcomes which are both shaped by those small decisions, and which go on to shape future small aspects in an iterative, nonlinear process. As we explored in Digital Internets Past and Present, the smallest decision such as the end-to-end principle upon which the Internet Protocol is founded works to assure device-to-device connectivity is front and center. And so there can be a community networking movement or an Internet of Things philosophy allowing different devices to work as needed within their local spaces while also assuring the ability to work with other devices working as needed within their local spaces. Small or large, every other aspect of the Internet Protocol works to assure this internetworking of end-point devices. But the end-to-end principle can also shape other pathways, too, such as when individuals come together to form a nonprofit think tank such as the Progress and Freedom Foundation from which the first publication "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age."

And so it is within our own lives that we are both shapers of, and people being shaped by, the events and materials around us. Bring to mind a few recent events that have happened within your lived experiences. Think broadly, considering your times outdoors as well as indoors, at stores and markets as well as at community centers, with community and family as well as time alone.

- What is one technical strength that was demonstrated by you or another person, human or more-than-human, within one or more of these events?
- What is one economic strength that was demonstrated?
- What social hierarchy and/or social equity was demonstrated by you or another person within one of these events?

The “Digital Divide”

> Digital Divide
>
> The economic, educational, and social inequalities between those who have computers and online access and those who do not.
>
> Merriam-Webster, “Digital Divide” [1]

One central theme during President Bill Clinton’s administration in the 1990s centered around fostering an economic transition based on technology investment national competitiveness. Goals included: long-term economic growth that creates jobs and protects the environment; making government more efficient and more responsive; and world leadership in basic science, mathematics, and engineering.[2] Core within these goals was an advancement of human capital through education and training, and the advancement of an “information superhighway” through general deregulation and targeted public–private partnerships. As part of the assessment, the **National Telecommunications and Information Administration (NTIA)** published a series of four *Falling through the Net* reports between 1995 and 2000. The 1995 report brought forward a survey of “have-nots” in rural and urban America, while the 1998 and 1999 reports were the first to especially highlight the “digital divide.” The final report in this NTIA series moved from reporting on the digital divide to considering ways to move towards digital inclusion.

For the Clinton administration, combating poverty and connecting the “have-nots” to a privatized **National Information Infrastructure (NII)** was considered a national economic emergency. As Daniel Greene notes in “Discovering the Divide: Technology and Poverty in the New Economy”, it is important to frame the digital divide “as a poverty program as well as a technology program.”[3] To this end, a healthy citizenry was redefined as “a bundle of human capitals brought to market by information technology” and digital literacy training became a limited investment in “workforce oriented technology provision and training.”[4] Greene goes on to state:

> Within this first part of the digital divide frame, combating poverty is a problem not of alleviating suffering in the present, but of making the correct investments in “information have-nots” so as to resolve current crises of underutilized labor, realize future capital growth, and achieve post-Cold War international economic hegemony.[5]

1. Merriam-Webster, “Digital Divide,” accessed July 3, 2020, https://www.merriam-webster.com/dictionary/digital%20divide.
2. William J. Clinton and Albert Gore, Jr., “Technology for America’s Economic Growth, A New Direction to Build Economic Strength,” ED 355 929, February 22, 1993. https://files.eric.ed.gov/fulltext/ED355929.pdf.
3. Daniel Greene, “Discovering the Divide: Technology and Poverty in the New Economy,” *International Journal of Communication* 10 (2016): 1216, https://ijoc.org/index.php/ijoc/article/view/3969.
4. Greene, “Discovering the Divide,” 1216.
5. Greene, “Discovering the Divide,” 1217.

The NII originated in 1991 as part of the High Performance Computing Act, created and introduced by then US Senator Al Gore who had spent years working to this point. In 1991, Gore published the following in *Scientific American*:

> The unique way in which the U.S. deals with information has been the real key to our success. Capitalism and representative democracy rely on the freedom of the individual, so these systems operate in a manner similar to the principle behind massively parallel computers. These computers process data not in one central unit but rather in tiny, less powerful units.
>
> Capitalism works on the same principle. People who are free to buy and sell products or services according to their individual calculations of the costs and benefits of each choice process a relatively limited amount of information but do it quickly. When millions of individuals process information simultaneously, the aggregate result is incredibly accurate and efficient decisions.... Communism, by contrast, attempted to bring all the information to a large and powerful central processor, which collapsed when it was overwhelmed by ever more complex information.
>
> Al Gore, "Infrastructure for the Global Village"[6]

Greene concludes by stating: "This conflation of different scales — infrastructure and individual, personal computing and national markets — was not just New Democrat spin, but an overarching regulatory regime emphasizing market competition as the primary political calculus and market citizenship as the primary political unit."[7]

> Take a minute to pause and reflect on how this consideration of a healthy citizenry and market citizenship compares to a key intention of Free-Net and Community Networking movements generally, and the information sciences more specifically, to foster an informed citizenry as highlighted in this textbook.

One significant shaper of the "digital revolution" explored in the last session was the Progress & Freedom Foundation, which sought to educate policymakers, opinion leaders, and the public regarding technological change based on a philosophy of limited government, free markets, and individual sovereignty. A central "futurist" thought leader who played a major role in fostering the creation of the Progress & Freedom Foundation was Newt Gingrich, a professor of history and geography at the University of Western Georgia in the 1970s before serving in the US House of Representatives from 1979 until 1999. His time as a Representative included serving as House Minority Whip from 1989 to 1995, and as Speaker of the House from 1995 to 1999. Further, as noted in the Frontline episode "The Long

6. Al Gore, "Infrastructure for the Global Village," *Scientific American* 265 no. 3 (September 1991): 151. https://www.scientificamerican.com/magazine/sa/1991/09-01/.
7. Greene, "Discovering the Divide," 1218.

March of Newt Gingrich," an initial principal project of the Progress and Freedom Foundation was Gingrich's college course called "Renewing American Civilization".[8] The course, like "Cyberspace and the American Dream," and released at about the same time, highlighted an important need to move from a Second Wave to a Third Wave economy by adding "social and political dominance to its accelerating technological and economic strength."[9] As a professor of history, for Gingrich the renewal of American civilization required something more than the reinvention of government which the Clinton administration was working towards — including through their digital divide framing that was emerging from Gore's extended congressional initiatives. While Gingrich agreed with a significant part of this, from his viewing of American history, the renewal of American civilization required the replacement of the entire structure of the "welfare state" with an "American values-based model" of the government.[10]

It's important to note that while Gingrich and Gore became highly vocal rivals in the 1990s, they both entered congress in the late 1970s and quickly became part of a bipartisan in-house futurist think tank, the "Congressional Clearinghouse on the Future," where they came to know and respect each other for important common ground that they shared. These shared ideals are demonstrated in the similarities seen between the activities of the Clinton administration and those of the Progress & Freedom Foundation. While there was a clear demarcation between the role of government and that of the Foundation, it was more one "between government as referee and government as spectator."[11] It was between reinventing government through advancing a healthy citizenry for market citizenship, part of which included digital literacy and inclusion to help individuals leave the welfare system, and the replacement of the entire structure of the welfare state.

In the preface to her book, *Digital Dead End: Fighting for Social Justice in the Information Age*, Virginia Eubanks notes "magical thinking" as a driver of technology-based economic development programs.[12] It is a deeply "thing-centered" approach. The underlying assumptions within these digital divide initiatives are that some people lack access to technology and/or the skills to effectively use the technologies meaningfully as citizens. Schools, libraries, religious institutions, and other community spaces play an essential underlying role to bridge the digital divide through provision of community technology centers and by advancing digital skills development for individuals. What is not seen is a shift in poverty policies that emerged in the U.S. and some European nations at the same time that included a punitive,

8. The course was taught through Kennesaw State College, Georgia, by Representative Gingrich and was open to all online. A complaint was filed regarding use of non-profit funds in support of staff time with the Committee on Standards of Official Conduct within the House in 1994, for which a settlement was reached.
9. Esther Dyson, George Gilder, George Keyworth, and Alvin Toffler, "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age," The Progress & Freedom Foundation, August 1994. http://www.pff.org/issues-pubs/futureinsights/fi1.2magnacarta.html.
10. Newt Gingrich, "Renewing American Civilization," CSPAN September 10, 1993. https://www.c-span.org/video/?50261-1/renewing-american-civilization. "The Long March of Newt Gingrich," Frontline, January 16, 1996. Accessed July 7, 2020. https://www.pbs.org/wgbh/pages/frontline/newt/index.html.
11. John Heilemann, "The Making of The President 2000," *Wired*, December 1, 1995, https://www.wired.com/1995/12/gore-newt/.
12. Virginia Eubanks, *Digital Dead End: Fighting for Social Justice in the Information Age*, Cambridge, MA: MIT Press, 2012.

paternalistic, racialized bent. Programs funding the bridging of the digital divide centered on educational quick fixes in which short hour-long courses ensuring individuals could turn on a computer, type on a keyboard, connect to the Internet, and fill out a job application form were prioritized over individual and community valued beings and doings. Quarterly reports to grant agencies prioritized head counts and defined skills checkboxes over community-defined learning outcome objectives. Ultimately, these were programs that crossed party lines, and were part of a broader neoliberal agenda. As Greene notes: "By reviewing the links between the digital divide frame and other contemporary neoliberal projects, it becomes clear that a crisis of human capital deficits was articulated in multiple domains and that final result of the frame — redefining access not as available tools or skills but as the opportunity to compete — fit cleanly not just with the demands of post-Fordist capital but with the common sense of laborers who really were excluded from the transition to a knowledge economy, and those helping professionals — librarians, in Stevenson's [2009] account — who really do work for inclusion."[13]

In the end, even as the left edge of neoliberalism under the Clinton administration began to weaken and the right strengthened, the problematic magical thinking of the digital divide framing has persisted, as Greene notes in his conclusion:

> 'Digital divide' stuck in the United States because a frame announcing a national crisis of competitiveness, defined as a human capital deficit and resolved through public-private partnerships for access extension, created a fundamental definition of 'access' that resolved the contradictions between a punitive, paternalistic poverty policy and the premise of the New Economy. If the opportunity to compete was made available by information technology, then investments in those opportunities — publicly encouraged but privately executed so as to not violate the sanctity of competition — were urgently required. Unfortunately, more competition creates not just more winners, but also more losers. Therefore, individual failures of competitiveness had to be excused as lacking initiative or improperly planning, whereas mass failures could be understood as populations surplus to New Economy requirements—thus justifying the expansion of the prison and workfare systems that paralleled Clinton-era digital divide initiatives and bounded their antipoverty aspirations.[14]

At some point in our participatory action research community inquiry projects, we may discover our own complacency within oppressive structures through an over-reliance on a distributive paradigm setting up computer labs, providing digital skills training, and helping "digital immigrants" keep pace within the digital revolution. We thus miss seeing the many non-technical, socio-political, and economic realities of generational oppressions that dismiss the community cultural wealth and local valued beings and doings to instead prioritize neoliberal agendas. One of my own deep nights of the soul occurred in 2008 when, through an abductive leap, I came to see a radically different understanding of the oppressive works I was doing and helping others do through teaching and practice as part of

13. Greene, "Discovering the Divide," 1216.
14. Greene, "Discovering the Divide," 1226.

our bridging the digital divide tasks. But following deep conversations with my community partners of color regarding unspoken truths of my own works centering on white supremacy, and also works towards new processes of reconciliation, our community of practice, oppressed and oppressor, used Freire's popular education approach in a deeper way to continue those aspects of projects that were deemed of value, innovated-in-use further those aspects we considered reparable in some meaningful way, and cast off those things that clearly worked to continue the intersecting webs of oppression. And it has been this continued work that has guided much of the development of this textbook.

As we've explored throughout this textbook, the technical can never be separated from the social. The sociotechnical artifacts and systems are complex webs of interdependent social and technical components. We each need to make use of new critical thinking and expanded digital literacy skills to tease apart aspects using individual action-reflection, and even more within communities of inquiry action-reflection using collective leadership and popular education strategies.

A great visualization of this transformative movement into a more people-centered, popular technology approach in which the marginalized "other" can challenge our preconceptions regarding the Digital Divide can be found in Eubanks' "Trapped in the Digital Divide: The Distributive Paradigm in Community Informatics." Together with YWCA workshop participants in Troy, New York, Eubanks created a series of drawings. She notes how her initial sketch used to answer their question, "What is the digital divide?" a new-to-them term, ultimately helped step workshop participants from the initial codification in which "have-nots" are helped to reach the desired "have" side of the chasm, to popular technology discussion and a new mutual understanding of the term. The revised drawings of the divides by workshop participants were essential in guiding this work of decodification and recodification. Some considered ways in which this needed to be re-articulated as technology for the people as skills, strengths, and resources from both sides of the "divide" are brought together. Others visualized the ways the divide actually represented a conflict between "The Man", those with power who used hoarding and shortcuts to advance personal gain, and "The Rest of Us," the survivors. And over time, this moved to considerations of new paths forward, ones that sought for possibilities of equal exchange in which digital technologies become one tool to bring stuff together within ethics of sharing and community.

These visualizations can provide an example of a recodification of one term, the digital divide. In doing this work of decodification and recodification in action reflection, we discover not an end point, but a new starting point. Alternate images of the "digital divide" serve as question prompts such as: if this is the true divide, what works do we now start, what works do we re-invent, and what works do we put aside as unusable to begin addressing this newly understood digital divide?

Moving From the Digital Divide to Digital Inclusion

As noted earlier, the National Telecommunications and Information Administration (NTIA)'s final *Falling through the Net* report focused on digital inclusion, a theme selected to especially highlight "the progress made and the progress yet to be made."[15] They go on to state:

The rapid uptake of new technologies is occurring among most groups of Americans, regardless of income, education, race or ethnicity, location, age, or gender, suggesting that digital inclusion is a realizable goal. *Groups that have traditionally been digital "have-nots" are now making dramatic gains.*

This first use of digital inclusion left open a more formal definition of the term. For many, the discussion has remained on the digital divides that leave some excluded from full participation online and all that brings. Others, like the nonprofit National Digital Inclusion Alliance (NDIA), have found the term digital divide to be problematic given the multiple intersecting divides within. They have instead forwarded a more formal definition of the key terms **digital inclusion**, **digital equity**, and **digital literacy** as follows:

Digital Inclusion

Digital Inclusion refers to the activities necessary to ensure that all individuals and communities, including the most disadvantaged, have access to and use of Information and Communication Technologies (ICTs). This includes 5 elements: 1) affordable, robust broadband internet service; 2) Internet-enabled devices that meet the needs of the user; 3) access to digital literacy training; 4) quality technical support; and 5) applications and online content designed to enable and encourage self-sufficiency, participation and collaboration. Digital Inclusion must evolve as technology advances. Digital Inclusion requires intentional strategies and investments to reduce and eliminate historical, institutional and structural barriers to access and use technology.

Digital Equity

Digital Equity is a condition in which all individuals and communities have the information technology capacity needed for full participation in our society, democracy and economy. Digital Equity is necessary for civic and cultural participation, employment, lifelong learning, and access to essential services.

15. National Telecommunications and Information Administration, "Falling through the Net IV: Toward digital inclusion," U.S. Department of Commerce. http://www.ntia.doc.gov/files/ntia/publications/fttn00.pdf, xiii.

Digital Literacy

NDIA recommends the American Library Association's definition of Digital Literacy via their Digital Literacy Taskforce:

Digital Literacy is the ability to use information and communication technologies to find, evaluate, create, and communicate information, requiring both cognitive and technical skills.

A Digitally Literate Person:

- Possesses the variety of skills – technical and cognitive – required to find, understand, evaluate, create, and communicate digital information in a wide variety of formats;
- Is able to use diverse technologies appropriately and effectively to retrieve information, interpret results, and judge the quality of that information;
- Understands the relationship between technology, life-long learning, personal privacy, and stewardship of information;
- Uses these skills and the appropriate technology to communicate and collaborate with peers, colleagues, family, and on occasion, the general public; and
- Uses these skills to actively participate in civic society and contribute to a vibrant, informed, and engaged community.

National Digital Inclusion Alliance, "Definitions"[16]

To the above digital literacy skills, as noted in the Orange Unit, I would add progressive community [inter]action and critical social + technical skills, brought forward as part of person- and community-centered deliberative dialogue processes using critical research paradigms, pedagogies, and abductive reasoning. This is embedded throughout this textbook and has been developed through a range of digital literacy-oriented courses and workshops done over the last two decades. And as noted earlier in this chapter, sometimes there have been deep nights of the soul that have been part of a critical abductive leap as a transformative moment reveals unseen or unforeseen works dehumanizing others. But at other times, I have had wonderful insights provided by those who are experiencing these works of marginalization and oppression.

As an example of the latter, consider a five-part workshop that included parents and teachers from a Kindergarten-5th grade elementary school whose constituents come primarily from households with low socio-economic status. The school's motto, "Technology and Literacy for the Community," inspired them to look for creative ways to further engage parents as collaborators in their children's education, knowing that a significant number of households did not have each of the five key elements necessary for full digital inclusion. One African-American parent noted at one point the frustration they

16. National Digital Inclusion Alliance, "Definitions," accessed July 6, 2020. https://www.digitalinclusion.org/definitions/.

feel when questioned regarding their decision to purchase a smartphone rather than a laptop. They went on to state how, as a parent of color, they need to teach their children to call a relative or trusted other whenever a police officer seems to be looking at them closely, and to then put their smartphone in a pocket with the connection open in case a negative encounter develops. As such, the smartphone, while an inferior device to the laptop in some ways for educational purposes, has a unique lifesaving property, the value of which trumps other considerations if the purchase of only one device is possible. Stories similar to this have been shared with me individually or within a class group discussion in a range of workshops by people of color, and highlight the intersectional realities of digital technology and Internet experiences, and the essential need for digital inclusion and equity to continually strive to advance social justice considerations within all aspects of policy and practice.

This highlights the essential need for digital inclusion and equity activities to consider the five key elements within a range of contextualized, dynamic individual, cultural, and community-defined conditions.

To conclude this chapter let's probe further the first element regarding digital inclusion, "affordable, robust broadband internet service." As NDIA notes, digital inclusion must evolve as technology advances. In a myriad of ways, broadband access within the United States has lagged behind that of other nations. In particular, given the advancements in Internet service that today includes **fiber-to-the-home (FTTH) or premises (FTTP)** in many nations, to what extent is FTTH a necessary condition for there to be full Digital Equity? In what ways is full civic and cultural participation, employment, lifelong learning, and access to essential services for the many places within the US where home fiber not available, or not affordably so, achievable?

Fiber is a central media used in building the backbone of the Internet throughout the world. But when it comes to Internet provision by the Internet Service Providers to the home, community centers, and small- to medium-businesses, what is primarily available is the old copper wires that had previously been installed to provide phone and television service. These proved more than adequate for voice service and the downloading of television audio and video. But these significantly underperform if we look at broadband Internet within a broader conceptualization beyond simple information consumption plus some basic social media and email tasks. To do live-stream sharing in support of interactive audio/video, selected asynchronous sharing of large datasets from home to community without use of cloud servers, or an emerging array of IoT device interactions centered within the original Internet Protocol's end-to-end principle high speed (20+ Megabits per seconds; Mbps), low latency (8 milliseconds or less) upload speeds are needed in addition to high speed downloads of online audio/video information and communication. If a musical group wants to do a live performance in which each is performing from a different LAN, but as if they were in the same room, they must have latencies at or under 8 milliseconds. To share video along with audio without lags, they must have 20+ Mbps uploads to share their audio/video with the others rather than just downloading the performance of their band. Adding to this, if there is more than one computing device on the LAN seeking to participate in live streams, asynchronous data sharing of large datasets, and IoT device activities, then each of these devices will each require high speed, low latency upload and download speeds, increasing the requirements for infrastructure service from the WAN to the LAN by the ISP. Digital Subscriber Line (DSL), Cable, WiFi, and cell-based Internet do not meet these needs, and they cannot be engineered to eventually meet such needs. Only fiber optics can, and it continues to be further developed, adding even greater possibilities for distance socializing in ways that advance civic and cultural participation, employment, lifelong learning, and access to essential services.

This highlights a way in which truly just digital inclusion and equity often requires fiber-to-the-home to fully provide the conditions for full participation in civic and cultural participation, employment, lifelong learning, and access to essential services.

Affordable, Robust Internet Service Provision

In her books *Captive Audience: The Telecom Industry and Monopoly Power in the New Gilded Age* (2013) and *FIBER: The Coming Tech Revolution – And Why America Might Miss It* (2018), Susan Crawford brings together her professional experiences as a member of the board of directors of ICANN from 2005-2008, as co-leader of the FCC transition team between the Bush and Obama administrations, and her time within the Law Schools at Michigan and Harvard, among other lived experiences. From this, she richly brings together the history of the Internet with current political contexts nationally and internationally to consider the impacts of the Internet today on our lived experiences. The monopoly power has significant power over health care, education, urban administration and services, agriculture, retail sales, and offices, something that has been a constant even as the emergence of transformative Internet of Things devices is reshaping the landscape. She has traveled around the United States and internationally to see the many different municipalities, regions, and countries that have, in some cases for decades, established ways to implement home fiber Internet as a utility to advance full digital inclusion and equity as defined by individuals and their communities. While I personally have been part of various Community Networking projects, some of which have included fiber-to-the-home grants, I found her interview with Bill Moyers to be a very insightful introduction to the deeper social shapers of key barriers that have been raised keeping us from this true integration of digital equity.[17] [Watch the video "Susan Crawford on Why U.S. Internet Access is Slow, Costly, and Unfair" online.]

As part of the Broadband Technology Opportunities Program (BTOP) grants following the 2008 financial crisis, the Obama administration made available funds for 1) Infrastructure, 2) Public Computing Centers, and 3) Sustainable Adoption programs. The Urbana-Champaign Big Broadband (UC2B) project worked to construct 187 miles of fiber-optic broadband network to provide high-speed connectivity to area community anchor institutions and support fiber-to-the-home services in four low-income neighborhoods. Community Networks continues to provide resources, including case studies,

17. Bill Moyers, "Susan Crawford on Why Our Internet Access Is Slow, Costly and Unfair," BillMoyers.com, February 8, 2013. https://billmoyers.com/segment/susan-crawford-on-why-u-s-internet-access-is-slow-costly-and-unfair/.

fact sheets, videos, and podcasts highlighting ways communities are working to implement cost effective, publicly owned, cooperative models, and other nonprofit approaches for community broadband.[18]

Today, FTTH over the publicly owned UC2B is being provided to a growing number of homes and other premises in the area through its private partner, I3 Broadband. During the COVID-19 pandemic and stay-at-home orders, this proved essential as I and my son sometimes were each participating in separate University Zoom live video sessions while my wife was participating in a HIPAA-compliant Doxy.me live video session with a patient. The FTTH Internet connectivity provided 1 Gbps download and 300 Mbps upload speeds, while our wired Ethernet connectivity to each room provided 100 Mbps upload/download to each of our laptops. We consistently had latencies under 8 milliseconds to the regional I3 Broadband. But there were many times when a Zoom indicator or traceroute test would indicate failures either at the cloud level, or more often at the other end of our connection. When functional diversity requires communication over voice and video rather than text following a Traumatic Brain Injury, as became my case after a 2016 bicycle accident, full participation may be lost with a bad audio/video feed somewhere along the Internet backbone or end-points, and a significant human aspect is lost. When a pair-programming hands-on activity with electronics or a face-to-face therapy session happens online without high quality video, a significant analog, social aspect is lost. When several minutes are lost troubleshooting technical problems, a significant analog, social aspect is lost. When people already experiencing levels of stress find themselves at a loss, problematic social aspects are increased.

This provides an example of how digital inclusion and equity is too often addressed only on one side of the Internet when viewed from the underlying end-to-end principle that arose from the French Cyclades model — a vision of interconnecting a group of islands, a metaphor from the Aegean Sea island group — and later guiding the Internet Protocol development first funded in support of a flexible military during the Cold War.

As one final example, my brother has had epilepsy since early childhood and suffers from refractory seizures that are drug resistant.

> Epilepsy is the fourth most common neurological disorder in the world. If you have epilepsy, surges of electrical activity in your brain can cause recurring seizures. [Watch the video “What is Epilepsy? from epilepsy.com” online.]
>
> Seizures are sudden surges of abnormal and excessive electrical activity in your brain and can affect how you appear or act. Where and how the seizure presents itself can have profound effects.

18. Read the transcript online from a 2013 interview with two participants in the project, Carol Ammons and Brandon Bowersox-Johnson.

> Not all seizures are the same. Many people with epilepsy have more than one type of seizure and may have other symptoms of neurological problems as well.
>
> The Epilepsy Foundation, "About Epilepsy: The Basics"

For my brother, treatment experience has taught him that maintaining a complete personal record of seizures and treatments is an essential complement to, and at times even more informative than, the expensive tests commonly used within the medical profession such as Magnetic Resonance Imaging (MRIs) and Electroencephalogram (EEGs). He demonstrates collective leadership in practice. For thirty years he has used his personal computer to document hundreds of seizures, over a dozen anti-seizure medications and their assigned usage regimens, and other valuable medical details. And today new seizure diary programs are becoming available to assist epilepsy patients in documenting the treatment path (my brother has been finding Seizure Tracker very useful to document and forward to his doctor treatment data). Combined, this rich personal dataset connects these medical-focused notes within the broader personal context in ways that help inform conversations with his care team and with the broader epilepsy community.

Recently, he was provided with an innovative NeuroPace brain-responsive neurostimulation (RNS) system. Tiny wires with electrodes, called leads, are placed in the area of seizure onset and connected to a neurostimulator placed under the scalp. This small battery-powered microcontroller is programmed by the doctor to monitor brainwaves, detect unusual activity, and respond in real time. It is personalized by the doctor for the individual, but this personalization can include additional new analog feedback of the individual to the doctor further expanding the information flow. [Watch the video "NeuroPace: How the RNS System Works" online.]

At least in ideal conditions. My brother still lives in the rural area in which we grew up. And as is the case for many rural areas, Internet service is still inadequate to meet all needs, especially within the context of tele-health. Provision of Internet is currently being provided by a Satellite provider, with latencies consistently over 700 milliseconds. This is far less than the <8 millisecond latency required to feel as if you are in the same room, and something not effectively supported within today's general purpose live streaming services such as Zoom. Further, beyond diminished or lost live stream conversations required as a result of Covid-19 social distancing, essential weekly uploads of NeuroPace data can sometimes fail for extended periods. In his speedtests, he finds that while his download speeds are often >30 Mbps, his upload tests at times have given the message "Upload Test Error". Thus, as my brother notes:

The new RNS treatment demands more than a computer and the skills to operate it–sufficient upload speeds are required to weekly transfer data to the doctor. Also, the COVID-19 outbreak has left me using the telephone to speak with a doctor and therapist more than once instead of using the tele-health option. Will RNS eventually offer the chance for doctors to alter the Neuropace device settings from home? Not without sufficient computer speeds.

Ultimately, information sharing between a distanced community of practice in support of collective leadership fails at multiple levels, even as the 20 Mbps download, 3 Mbps upload (20/3) speed definition of minimum service as defined through the US FCC Lifeline program is recorded as met within corporate and governmental Internet records.

As the Rev. Dr. Martin Luther King, Jr. notes: "Injustice anywhere is a threat to justice everywhere. We are caught in an inescapable network of mutuality, tied in a single garment of destiny. Whatever affects one directly, affects all indirectly."[19] Truly just digital inclusion and equity is more than an individual issue. It is a community and social issue that requires a high-quality infrastructure not only at the backbone of the Internet, but at each end-point as well and all nodes in between. Each missing element as listed within the definition of digital inclusion, and each element that is less than what is needed as defined from within the individual and community's defined valued beings and doings based on a community's cultural wealths, is another intersecting component of the many different analog and digital divides that reinforce the giant triplets of racism, materialism, and militarism.

Lesson Plan

Essential Resources:

- Robinson, Laura, Jeremy Schulz, Matías Dodel, Teresa Correa, Eduardo Villanueva-Mansilla, Sayonara Leal, Claudia Magallanes-Blanco, et al. "Digital Inclusion Across the Americas and Caribbean." *Social Inclusion* 8, no. 2 (May 14, 2020): 244–59. https://doi.org/10.17645/si.v8i2.2632.
- Eubanks, Virginia. "Trapped in the Digital Divide: The Distributive Paradigm in Community Informatics." *The Journal of Community Informatics*, 3, no. 2 (September 14, 2007). https://doi.org/10.15353/joci.v3i2.2373.
- Ferreira, Becky. "Watch Our Documentary on Detroit's Grassroots Internet Network." Vice, November 21, 2017. https://www.vice.com/en_us/article/qv395v/watch-our-documentary-on-detroits-grassroots-internet-network.
- King, Jamilah. "A Tech Innovation in Detroit: Connect People, Not Computers." COLORLINES, October 3, 2012. https://www.colorlines.com/articles/tech-innovation-detroit-connect-people-not-computers.
- Yu, Liangzhi, Lorcan Dempsey, and Sarah Ormes. "Community Networking: Development, Potentials and Implications for Public Libraries." *Journal of Librarianship and Information Science* 31, no. 2 (June 1, 1999): 71–83. https://doi.org/10.1177/096100069903100202.

19. Rev. Dr. Martin Luther King, Jr., "Letter from a Birmingham Jail," Birmingham, AL, April 16, 1963. https://www.africa.upenn.edu/Articles_Gen/Letter_Birmingham.html.

Key Technical Terms

- Digital Divide, Digital Inclusion, and Digital Equity
- **Client-Server Architecture**, **Daemon**, **Hypertext Transfer Protocol (HTTP)**, **Hypertext Markup Language (HTML)**, the **Programming Languages** PHP and SQL
- Sociotechnical Networked Information System

Professional Journal Reflections:

The social shaping of technology includes a range of political, economic, and social influencers and frames shaping technologies and our visions for these technologies. The digital divide further represents a social framing of a range of sociotechnical artifacts as exclusively technical matters, something the digital inclusion and other movements are trying to reframe.

1. Take some time to read through your past Professional Journal Reflections. As you reread your own reflections, what has remained consistent throughout these? What has slowly transformed within these?
2. In what ways do the readings of this session build from the past reflections? In what ways do they counter those past reflections?
3. What are some first thoughts regarding next steps you need to take individually and within your communities of practice in response to your journey through *A Person-Centered Guide to Demystifying Technology*?

3B: A Person-Centered Network Information System Adventure

Technical Overview

> My philosophy is very simple: When you see something that is not right, not fair, not just, you have to stand up, you have to say something, you have to do something.
>
> My mother told me over and over again when I went off to school not to get into trouble, but I told her that I got into a good trouble, necessary trouble. Even today I tell people, "We need to get in good trouble."
>
> John Lewis, interviewed by Valerie Jackson for StoryCorps[1]

We began this session considering the codifications of the digital divide and beginning a process of decodification of this and related terms. We explored some examples specifically related to NDIA's first element of digital inclusion, "affordable, robust broadband internet service." Digital inclusion, equity, and literacy, as brought forward by NDIA, ALA, and others, give a more formal definition that can serve as a framework to move on from the magical thinking that is pervasive within current neoliberal understandings of sociotechnical networked information systems. But this is an ongoing work of community inquiry and action-reflection.

As we move forward in our professional practices, we each will likely make use of a range of sociotechnical digital tools within our daily activities, which, together in work with others, include works towards advancing digital inclusion and equity goals. But in practice, information science professionals have encountered limits and constraints to the social justice parameters given their permitted roles, and given how certain socially structured fabrics and certain community values are placed above oth-

1. John Lewis and Valerie Jackson, "The Boy From Troy: How Dr. King Inspired A Young John Lewis," StoryCorps, February 20, 2018. https://storycorps.org/stories/the-boy-from-troy-how-dr-king-inspired-a-young-john-lewis/.

ers, particularly in light of dependencies on public funding that gives greater power to the middle-class, white population.[2]

As an introductory text, *A Person-Centered Guide to Demystifying Technology* has worked to open up the closed box of digital technologies sufficiently so that we can explore ways in which we can innovate-in-use our everyday technologies so that our choice, [re]design, and use of digital technologies better serve the diverse community knowledge and cultural wealth within the fabric of communities. It is an introductory text seeking to help us see better those aspects of digital technology that are "not right, not fair, not just," to quote John Lewis. As noted throughout the Rainbow Unit, there have been an extraordinary growth in digital computing and communications over the last 50 years. The community networking and community FabLab movements are but two of the many examples of ways digital bits and physical atoms have been brought together with the objective of advancing the real-world goals of local people to address the situations limiting them from being more fully human as active members within civil discourse as part of a community forum, life-long learning, and information access and use. But there have also been key influences and influencers that have prioritized integration of social and political dominance within accelerating technological and economic strengths through works of individualism, technological determinism, and supply-side, trickle-down neoliberalism. These have served to relocate some divides while working to reinforce and expand others, often in the name of digital inclusion.

To move beyond the many historic and ongoing divides fostering existing injustices and the digitization of information systems that are birthing new injustices in many areas of society locally, regionally, and beyond, "you have to say something, you have to do something" in meaningful ways. This digitization of the divides continues even as we remain caught in that inescapable network of mutuality and are thus being affected, whether it be directly or indirectly, as Martin Luther King, Jr., notes.[3] As information science professionals, this can come in many forms, including: as part of social justice storytelling; as part of our selection and construction of electronic components; within our work to integrate, modify, and write new programming code; and in the selection, creation, and use of algorithms, including those incorporating artificial intelligence and machine learning.

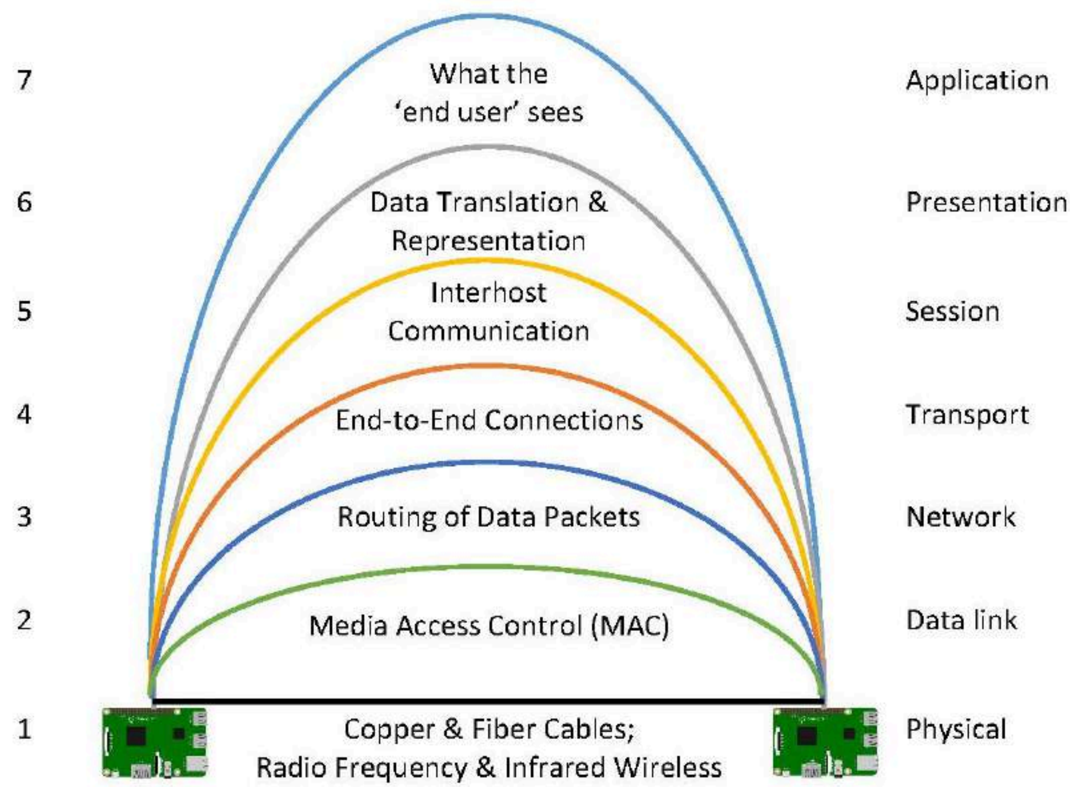
Let's finish sociotechnical session three of the Rainbow Unit by taking some first steps to create a general-purpose Raspberry Pi Server. At a minimum, we'll setup the Raspberry Pi as an Apache Web Server with the WordPress content management system (CMS). From here, it can use this general-purpose space strategically to prototype multiple subcomponents within the larger digital Little Free Library, which can be used to get into good trouble, saying something, doing something to address an injustice. Or maybe you'd like to try out creating a DIY community wireless network, or set up multiple Internet of Things devices to get into that good trouble.

2. Bharat Mehra, Kevin S. Rioux, and Kendra S. Albright, "Social Justice in Library and Information Science," in *Encyclopedia of Library and Information Sciences* (CRC Press, 2009), 4820-4836. https://doi.org/10.1081/E-ELIS3-120044526.
3. Rev. Dr. Martin Luther King, Jr., "Letter from a Birmingham Jail," Birmingham, AL, April 16, 1963. https://www.africa.upenn.edu/Articles_Gen/Letter_Birmingham.html.

As you take on this adventure, keep forefront in your mind ways to ensure that this is a person-centered approach to do something to advance a more just society.

Exercise: The General Purpose Raspberry Pi Web Server

In session one of the Rainbow Unit, we did a few different exercises exploring some of the underlying concepts behind the Internet of Things, also highlighted in Limor Fried's segments "All the Internet of Things." Of special note now, reflect back on the Hyptertext Transport Protocol (HTTP), an example of a client/server protocol that works at layers 6 and 7 of the OSI model.



Let's now test out a more general-purpose installation of HTTP on the Raspberry Pi, setting it up as **web server** that can be used on the **World Wide Web**. While we did create an HTTP-based web server in session 1, the Python library code used created a special-purpose web server. This time, we'll install the Apache web server, a widely used, general-purpose web server.[4] Before starting the installation of Apache, take a few minutes to watch Carrie Anne Philbin's Crash Course Computer Science episode #30 on the World Wide Web.

4. Apache® is a registered trademark or trademark of the Apache Software Foundation in the United States and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of these marks. "Debian" and the Debian Logo are trademarks of Software in the Public Interest, Inc.

About the Apache Web Server

Reflecting back on my time from 1993 to 1995 as a postdoctoral researcher at the Neuronal Pattern Analysis group at the University of Illinois's Beckman Institute, I remember our explorations of ways to make use of this brand-new thing called a "web browser" to share our raw data sets across the Internet with other researchers. One of the challenges we faced was the rapid ongoing changes in the code for the campus's National Center for Supercomputing Applications (NCSA) HTTP server, in support of their just-being-developed Mosaic web browser. What worked on the http server or on the web browser one day might not work the next day. Further, changes made within HTTP or HTML may create conflicts with the newly written code needed on the Sybase database server to effectively share the data by way of the hypertext protocols. I'd actively work on the HTTP/HTML side, while my colleague would work on the database server side. This is the rapid prototyping, fail-forward, growth mindset in action. It also represents those times when innovations go viral while still in their alpha phase, the lab time before an artifact or system enters the beta testing phase and eventual stable release. I was already working with the Linux operating system, version 0.9. I don't remember the 0.x versions of Mosaic and NCSA HTTP, or if they were even listed yet.

The University of Illinois meets the Carnegie Classification R1: Doctoral Universities – Very high research activity in part because the core focus of these research and learning activities are to "pioneer innovative research that tackles global problems and expands the human experience. Our transformative learning experiences, in and out of the classroom, are designed to produce alumni who desire to make a significant, societal impact."[5] For me, innovative research had expanded from my initial grant-funded electrophysiology-based project to also include an HTTP server and web browser project sharing the raw data stored in multiple Neuronal Pattern Analysis group databases. This inspired me to switch tracks, moving from my postdoctoral research at the Beckman Institute, and joining the Prairienet Community Network research project at the School of Information Science in 1995, which was using community inquiry to explore how the Internet and a HTTP-based networked information system could be used in support of various community information, communication, and learning activities. Earlier that year, the Apache HTTP server project began as the NCSA research for which the HTTP server and Mosaic web browser were initially being developed was winding down. The NCSA code was available for use in other projects through what became the free and open-source licensing framework. Robert McCool—who as an Illinois undergraduate student helped lead the initial NCSA HTTP server development in support of Mosaic—joined with a group of other computer programmers who were patching the NCSA HTTP server to meet their own project requirements, such as Brian Behlendorf—a founding member and president for three years of the Apache Software Foundation—to release the Apache HTTP server using the NCSA code as its base. The use of existing code and a series

5. "About Illinois" accessed June 28, 2023, at: https://illinois.edu/about/index.html

of software patches led to the early development team's use of the pun "A PAtCHy" server, a pun that somehow stuck and became the formal name, respelled Apache HTTP.[6]

As Carrie Anne Philbin notes at the end of the Crash Course Computer Science video above with regard to net neutrality, the implications of decisions matter. And so it is with The Apache Way, as listed within the Apache Software Foundation website. It is a 501(c)3 non-profit public charity organization incorporated in the United States of America, formed for open, collaborative software development within a principle of "meritocracy": government by merit. It is one built within the spirit of community, in which various projects were emerging that together become the "projects" by which the separate communities were identified. And so a framework was created within which each community, or project, was designated as the central decision-making organization for their community within the broader Apache world.

But it is also a foundation in which the pun that inspired the name "Apache" came with acknowledged cultural appropriation implications. Co-founder Brian Behlendorf noted in the documentary "Trillions and Trillions Served," as cited by both The Apache Software Foundation Wikipedia page and Natives in Tech Blog post "Apache® Appropriation" [7]: "I suggested the name Apache partly because the web technologies at the time that were launching were being called cyber this or spider that or something on those themes, and I was like we need something a little more interesting, a little more romantic, not to be a cultural appropriator or anything like that, I had just seen a documentary about Geronimo and the last days of a Native American tribe called the Apaches, right, who succumbed to the invasion from the West, from the United States, and they were the last tribe to give up their territory and for me that almost romantically represented what I felt we were doing with this web-server project…" (NOTE: The YouTube video of the "Trillions and Trillions Served" documentary has since been made private, although is still referenced as a source in both the Wikipedia page article and the Natives in Tech Blog post references above. I still include this quote even though I was not able to confirm this citation during the writing of this on April 24, 2023 as even the decision to make private the YouTube video itself adds meaning to this issue of cultural appropriation and the stories that are told in relation to this.) The name "Apache," which is trademarked by a U.S. charity nonprofit foundation, is thus inspired by "the last days of a Native American tribe" without recognition that the Indigenous peoples currently have eight federally recognized tribes with many thousands of living Apache citizens today, without full consultation, and without enumeration. The trademark was awarded to the nonprofit foundation by the same U.S. government that acknowledges the sovereignty of these eight tribes—Apache Tribe of Oklahoma; Fort Sill Apache Tribe of Oklahoma; Jicarilla Apache Nation, New Mexico; Mescalero Apache Tribe of the Mescalero Reservation, New Mexico; San Carlos Apache Tribe of the San Carlos Reservation, Arizona; Tonto Apache Tribe of Arizona; White Mountain Apache Tribe of the Fort Apache Reservation, Arizona; and Yavapai-Apache Nation of the Camp Verde Indian Reser-

6. "Apache Server Frequently Asked Questions," Apache, accessed July 20, 2020, https://web.archive.org/web/19970106233141/http://www.apache.org/docs/misc/FAQ.html#relate.

7. Recvlohe, Adam, Holly Grimm and Desiree Kane. Apache® Appropriation. January 6, 2023. https://blog.nativesintech.org/apache-appropriation/. Accessed April 24, 2024.

vation, Arizona. In so doing, we found two decisions, the first an acknowledgement of the sovereignty of eight Apache tribes, and the second an awarding of the Apache trademark to a foundation with The Apache Way logo of a feather that "was chosen out of reference and appreciation for the people and tribes who refer to themselves as 'Apache.'"[8] But we also find in these the unaddressed need for direct mediation and conflict resolution found within models such as restorative justice, as the trademark was requested and awarded without consultation, and without acknowledgement and respect of the name's provenance and origin from those who had been awarded sovereign nationhood.[9]

As mentioned at the start of the Background Knowledge Probe to begin this session, small decisions can lead towards large outcomes, which both are shaped by these small decisions and go on to shape future small aspects, in an iterative, nonlinear process in which the larger systems and structures become a reflection. The Apache HTTP server is shaped by the various small decisions listed above and many more as well, some of which are community- and collaboration-centered in ways that strongly match core principles of the Internet Protocol. Apache remains commonly used, especially on Linux-based computers, making up over 25 percent of the websites around the world, and is the perfect choice for a general-purpose HTTP server on the Raspberry Pi building community-led networked information systems. The following instructions are a slightly modified version of those found on the Raspberry Pi Foundation documentation site. Using the DIKW framework as introduced in the Orange Unit 4A: Storytelling in the Information Sciences, these modifications are meant to facilitate exploration of this networked information system from both technical and social lenses to connect data with context. In organizing and structuring the instructions in this way, these modifications seek to finetune the relevance of the general-purpose HTTP server set up for the more specific purposes of the learning context at hand. And as part of the shift from a "thing-orientation" to a "person-orientation", the modifications are meant to help your community of inquiry make this information actionable, that is, turn it into knowledge. It is in this way that we've seen such digital information and communication technologies subsequently serve as useful tools for movement work, for standing up to say something, to do something to facilitate a more just society for all, and especially for those who have been pushed to the margins. With creative works comes failure, requiring a fail-forward and growth mindset, as we first learned in Carol Dweck's video[10] shared in the Introduction and which we've explored through the book. As we near the conclusion of this introductory text advancing a more holistic and nuanced understanding of the multiple social and technical layers of networked information systems, it is important to highlight the need for each of us to be open to rebuke and correction within the range of sociotechnical layers, including those of cultural appropriation. For to the extent that this is acknowledged and addressed through direct mediation and conflict resolution, we open more case examples of the arc of the moral universe, which, while long, does bend towards justice, as many

8. The Apache Software Foundation. "History of the Apache Software Foundation." Accessed April 24, 2023. https://www.apache.org/apache-name/
9. Recvlohe, Adam, Holly Grimm, and Desiree Kane. "Apache® Appropriation." Posted January 6, 2023. Accessed April 24, 2023. https://blog.nativesintech.org/apache-appropriation/
10. Dweck, Carol. "The Power of Believing That You Can Improve." YouTube, December 17, 2014. https://www.youtube.com/watch?v=_X0mgOOSpLU.

social justice warriors, including the Reverend Dr. Martin Luther King, Jr., have noted. For as President Barack Obama stated in a 2009 *Time* magazine article, "while you can't necessarily bend history to your will, you can do your part to see that, in the words of Dr. King, it 'bends toward justice.'"[11]

Net neutrality is a critical issue as we work on the World Wide Web through the installation of a general-purpose web server, and is something we as professionals should follow, given its ongoing impacts shaping our work. And it is something we can serve to influence as members of our societies, wherever this may take us moving forward. So, too, is the identification of issues of cultural appropriation, however. As we explored in the Blue Unit chapter 4A: Sharing Our Counterstories, we need to continuously research the self, research the self in relation to others, join in engaged reflection and representation, and shift from self to system, as H. Richard Milner IV notes in the article "Race, Culture, and Researcher Positionality: Working Through Dangers Seen, Unseen, and Unforeseen." In briefly unpacking the many positives, along with a newly emerging exploration of an injustice of cultural appropriation found within the name, we mean to recognize the dangers in a way that can begin a process of restorative justice, and ensure that this process is kept alive and active as we seek ways to do our part. But to learn of this unforeseen danger that comes with installing an actively used, open-source web server like Apache, without adding this to our story library to be strategically used moving forward, we have then swept it under the rug, so to speak, like the dust we want to hide from visitors. It is to make it Unseen and to therefore become part of the ongoing systems that in this context has been centuries of work to erase Indigenous presence, past and present, within the Americas. In my writing of these paragraphs following the exploration of student reading notes on the previous draft of this chapter April 24, 2023, I am responding to my own (re)introduction to the Apache Software Foundation as more than pun, and as something I need to delve into further moving forward. I need to do this while also continuing to actively make use of this server as an example of free and open-source software, which remains foundational to key aspects of my community networking movement and teaching work.

Steps

Install Apache

1. First, update the available packages by typing the following command into the Terminal:

```
pi@raspberrypi:~ $ sudo apt update
```

2. Next, install version two of the Apache web server by typing:

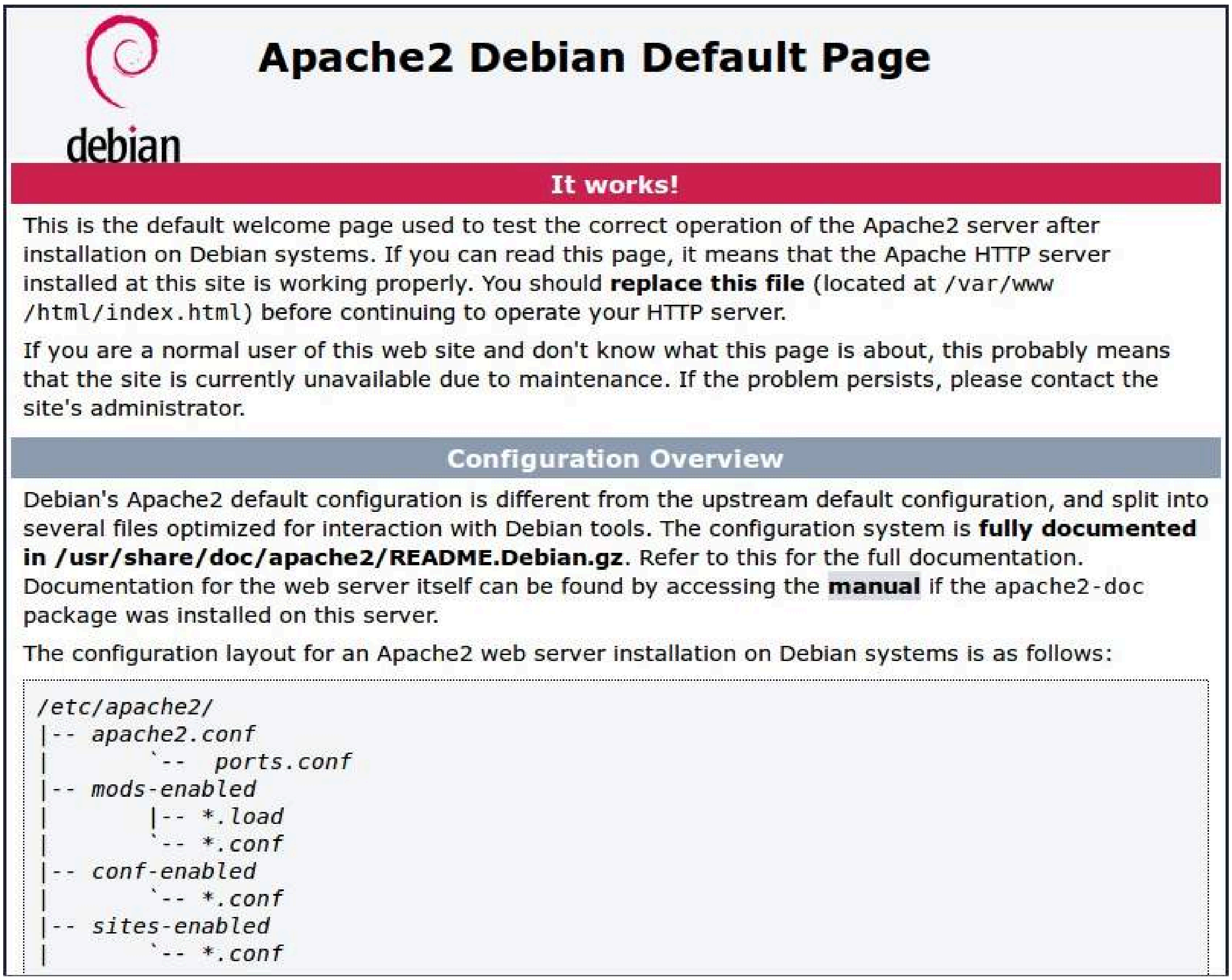
```
pi@raspberrypi:~ $ sudo apt install apache2 -y
```

11. Obama, Barack. "A New Era of Service Across America." *Time* (March 19, 2009). Time Inc., New York. (Online *Time* archive, time.com)

Test the web server

By default, Apache puts a test HTML file in the web folder. This default web page is served when you browse to http://localhost/ using the Chromium Web Browser on the Pi itself, something that works well for those who connect to the Raspberry Pi graphical display using RealVNC, or who have their Raspberry Pi attached directly to a keyboard, mouse, and monitor. For everyone, you can also access the test HTML file using something similar to http://192.168.1.10 (whatever the Pi's IP address is) from another computer on the network. To find the Pi's IP address, type `hostname -I` at the command line (or read more about finding your IP address).

Browse to the default web page either, on the Pi or from another computer on the same LAN, and you should see the following:



Unlike in session one, this time around, we do not add a port number after the IP address. The HTTP protocol has as a default port 80. So you could type in http://192.168.1.10:80, but you do not need to do this, as it is assumed by default.

This also means that you could have two web servers running simultaneously, the Apache general-purpose one on port 80, and one of the Internet of Things Python HTTP servers that use port 8401. Check this out by typing into the terminal window:

```
pi@raspberrypi:~ $ python3 simple_uartWebserverLED.py
```

Now, in a second web browser window, go to your IP address equivalent of http://192.168.1.10:8401. Voilà! Two web servers on one Raspberry Pi!

Initial Reflections

- What is a web browser?
- From where does a web browser get its web pages?
- What's different regarding the web pages retrieved from port 80 versus those retrieved from port 8401? When might this matter?

Changing the default web page

The default web page created during the installation is just an HTML file in the computer's filesystem. When installing servers, typically also installed is one or more configuration file(s). Within Linux operating systems, these are usually installed in the `/etc`, or et cetera, directory. In the terminal window, type:

```
pi@raspberrypi:~ $ ls -al /etc/apache2
```

Notice a file called `apache2.conf`. This is the main configuration file used to guide the launch of an Apache HTTP daemon. This provides the start-up information needed for a server to perform in the specified way. The listed apache2.conf, ports.conf, and additional conf-enabled, mods-enabled, and sites-enabled files provide those who are administering the HTTP with server abilities to further innovate-in-use the base Apache software. Changes can be made on the fly, and then integrated into the server through a refresh without disrupting ongoing and new page requests.

> In computing terms, a **daemon** is a computer program that runs in the background, providing services as needed. In this case, each time it is launched, the HTTP daemon starts itself based on the specifications within the configuration file, and then mostly hangs out, twiddling its thumbs and waiting for a call in asking for something. When the call comes, it gets busy doing its stuff before going back to waiting mode. You'll often see a running HTTP server daemon listed as httpd. These daemon server applications are a way for things to perform a similar manual function to that done by a human call support service where the worker closely follows a checklist. When might it be better for this service to be done

> by the "thing"? When might it be better for this service to be done by the "person"? In what ways might machine learning and artificial intelligence advance to the point where some of the current "person" tasks may be replaced by a "thing"?

To review these configuration files for Apache, head over to `/etc/apache2`. It is especially recommended to skim the `apache2.conf` and `ports.conf` files.

We don't need to review the configuration files to know that the default location for HTML files in Apache is generally under the `/var`, or variables, directory. When we install Apache, a new directory, `/var/www`, is created to specifically host World Wide Web data. And Apache goes on to create a test `index.html` HTML file. (In the /etc/apache2 configuration files, index.html is one of the default files looked for when someone requests a parent page for a website. This is why we can get a webpage by typing in something like "www.raspberrypi.org." This web browser request, combined with the configuration of the HTTP server, results in the return of "www.raspberrypi.org/index.html" or something similar, as specified within the HTTP server configuration file.)

Let's go over to the HTML directory space and edit the index.html file `/var/www/html/index.html`. Even as we do this, let's keep the python3 HTTP server running for the moment, as we'll bring these together in a later step. So, navigate to this directory in a second terminal window so that we can have a look at what's inside of this test page:

```
pi@raspberrypi:~ $ cd /var/www/html
pi@raspberrypi:/var/www/html $ ls -al
```

This shows that by default, there is one file in `/var/www/html/` called `index.html`, and it is owned by the `root` user (as is the enclosing folder). In order to edit the file, you need to change its ownership to your own username. Change the owner of the file (the default `pi` user is assumed here) by typing into the terminal:

```
pi@raspberrypi:/var/www/html $ sudo chown pi: index.html
```

You can now try editing this file using `mousepad` and then refresh the browser to see the web page change. To do so, in the terminal window, type:

```
pi@raspberrypi:/var/www/html $ sudo mousepad index.html
```

Next, scroll through the <head> section, which contains a range of metadata and also various **Cascading Style Sheets (CSS)** code, until you see the tags </head><body> on two separate lines.

Diving Deeper

Consider some quick changes, such as changing the title “Apache2 Debian Default Page” and the note “It works!” to something different. You might also add in additional <p></p> paragraphs sharing your thoughts at this moment.

When done editing the html, save the changes. You don’t need to exit, but can now refresh the browser to see the web page changes. This may take some trial and error to get working—keep failing forward!

Once done testing out changes of the default index.html file, exit mousepad.

Creating our own webpage

Starting at minute 4:05 of the video “The World Wide Web: Crash Course Computer Science,” Carrie Anne Philbin gives a quick example of an HTML page.

To create new web pages, you’ll need to run `mousepad` or your favorite text editor with super user privileges, as the /var/www/html directory is owned by user “root.” For example, to create the page mypage.html, type:

```
sudo mousepad mypage.html
```

Once you’ve opened the text editor, take a few minutes to explore creating a webpage. If you haven’t done this before, you can test out Philbin’s example. Or if you’ve done web page creating directly through text editing before, feel encouraged to use your own creativity to create a new page. Once you think you have something reasonable, save the file (you don’t need to exit, though), and then go to a web browser and go to the new page. In my case, I’d type in:

```
http://10.0.0.30/mypage.html
```

Creating an Internet of Things iframe page

The iframe tag is used to embed another document within the current HTML document. The iframe tag is used throughout this textbook to bring in YouTube videos. And we can now use this frame to bring in a web page from another server—in this case, that created by the HTTP server run through simple_uartWebserverLED.py.

Open up a new text file in the www directory called iot.html. If I’m using mousepad, I’d type:

```
pi@raspberrypi:/var/www $ sudo mousepad iot.html
```

Now, create a page that looks something like:

```
<html>

<head><title>IoT Gathering Page</title></head>

  <body>

        <h1>Internet of Things Gathering Page</h1>

        <p>This page is used to gather datasets from various

                local Internet of Things devices

        </p>

        <iframe width="1024" height="768"

                     src="http://10.0.0.30:8401"

                     frameborder="0" allowfullscreen>

        </iframe>

  </body>

</html>
```

Notice that it is a combination of static HTML and also an iframe tag bringing in source data from the other HTTP server running on the Raspberry Pi. Make sure you've listed the IP address of your own Raspberry Pi instead of my 10.0.0.30 Raspberry Pi address. Then save the document. Be sure you have your breadboard and Circuit Playground Express components attached and that your Python3 program simple_uartWebserverLED.py is up and running successfully. Now go to your web browser and open up both the Python page directly, as we did in session one, and also through the Apache server, as we've just set up now. For my Raspberry Pi, with an IP address of 10.0.0.30, I'd go to:

```
http://10.0.0.30:8401/
http://10.0.0.30/iot.html
```

And remember, if you have other LAN-based computing devices with web browsers installed, such as your smartphone, you can use those to access these pages as well.

Install PHP

When we used the Python3 BaseHTTPRequestHandler library in session one, we were able to bring together some static HTML page tags with some Python programming code to create a dynamic web page whose data was continuously updated. With the iframe tag, we were able to bring this dynamic web page data into an otherwise static web page we were creating. Let's finish this exercise by installing PHP.

PHP is a **preprocessor**: it's code that runs when the server receives a request for a web page via a web browser. It works out what needs to be shown on the page, and then sends that page to the browser. Unlike static HTML, PHP can show different content under different circumstances. Other languages are also capable of doing this, but since WordPress is written in PHP, that's what we need to use this time. PHP is a very popular language on the Web: huge projects like Facebook and Wikipedia are written in PHP.

"PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML."[12] To allow your Apache server to process PHP files, you'll need to install the latest version of PHP and the PHP module for Apache. While still in the `/var/www/html` directory, type the following command to install these:

```
pi@raspberrypi:/var/www/html $ sudo apt install php libapache2-mod-php -y
```

Now, rename the index.html file to index-orig.html:

```
pi@raspberrypi:/var/www/html $ sudo mv index.html index-orig.html
```

Create the file index.php:

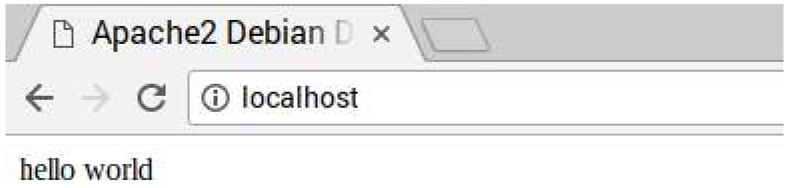
```
pi@raspberrypi:/var/www/html $ sudo mousepad index.php
```

Put some PHP content in it:

```
<?php echo "hello world"; ?>
```

Now, save the index.php file you've edited in mousepad, and refresh your browser. You should see the phrase "hello world" displayed. This is not dynamic, but still served by PHP.

12. "What Is PHP?" PHP Manual, accessed July 21, 2020, https://www.php.net/manual/en/intro-whatis.php.



If you see the raw PHP above instead of "hello world", reload and restart Apache like so:

```
pi@raspberrypi:/var/www/html $ sudo service apache2 restart
```

Now try something dynamic using mousepad to add the following line in index.php:

```
<?php echo date('Y-m-d H:i:s'); ?>
```

Underneath the hood, PHP is now running Linux operating system apps to collect current information and bring it back to a dynamic web page. This is the PHP equivalent to Python's use of `/opt/vc/bin/vcgencmd measure_temp` to get the current temperature of the Raspberry Pi, like we did in session one of the Rainbow Unit.

To add in some static text using standard HTML tags, consider adding an <h1>*Title of page*</h1> line above the PHP code, and maybe some <p>*paragraphs of text*</p> after the PHP to see what it looks like.

Going a step further, you can show a rich listing of your PHP info at the bottom by adding an additional PHP hypertext preprocessor call:

```
<?php phpinfo(); ?>
```

Key Takeaways

We've now used a few technical exercises over the sessions of the Rainbow Unit to explore different ways we could create a Raspberry Pi HTTP server. While at first, we used Python3 BaseHTTPRequestHandler to provide special purpose HTML page data and information, we've now added the general-purpose Apache HTTP server and PHP preprocessor to create a broader range of HTML pages that could be accessible upon request by an HTTP client. While a Python3-based HTTP server can prove of value in support of a specific task, it is the general-purpose HTTP server which has become the foundation of the Internet's World Wide Web.

In this exercise, we've seen how flexible server platforms like Apache have been developed using the open protocols and standards of the Internet and an open community of developers to work on a range of different computers and to meet a range of different use objectives. This design strategy provides opportunities for innovation-in-use, something we tested as we moved from initial installation of Apache, to the creation of a first HTML page, to the addition of an Internet of Things iframe, to the addition of the PHP Hypertext Preprocessor. We've also seen how, through the use of the default port address with the Apache server and an alternate 8401 port for the Python3 server, one computer could run multiple distinct instances of the HTTP server package simultaneously, providing us with even greater flexibility to innovate-in-use.

As we saw in Rainbow Unit 2B: The Infrastructure of the Internet, and as was again highlighted to start this chapter, the Internet Infrastructure itself is primarily comprised of the Network, Transport, and Session layers of the OSI model. It is not designed for a specific purpose, just the general one of end-to-end communication between devices. The HypterText Transport Protocol (HTTP) then works at layers 6 and 7 of the OSI model to facilitate data translation, representation, and "end user" visualization, even if that "end user" is another HTTP server working to link together different data and information. The HTTP server can be single-purpose and run through Python code, or it can be general-purpose, as it is when running through Apache. Add-ons like PHP and javascript can also be added to extend what an HTTP server such as Apache can accomplish. As we saw in Rainbow Unit 1B: Connecting Our Electronic 'Thing' to a Wider World, the small decisions made in the designs of our breadboard circuit, of the Raspberry Pi GPIO port, of the Raspberry Pi OS, of the Python code running on the Raspberry Pi OS, of the Circuit Playground Express, of the CPE controller that includes support of MakeCode .uf2 programs, and of the MakeCode program running on the CPE, each shaped the larger experience of the user interacting with the system that was created. So, too, can we see now the many small design decisions made on the many smaller components that make up the Apache and PHP platforms and the HTML and PHP code running within these platforms. But it is important that we also take a further step back to recognize ways in which the small design decisions regarding the many individual aspects of the infrastructure of the Internet that work at layers 3, 4, and 5 of the OSI model have shaped the decisions that make up the applications and platforms working at layers 6 and 7 of the OSI model.

As John Lewis states, "when we see something that is not right, not fair, not just, we have to stand up, we have to say something, we have to do something." We need to get in good trouble. When we look at the digital divide and take on work to instead foster digital inclusion and digital equity, it is actually by design that we often only see the black box of the Internet, instead of seeing the many small components of which the Internet serves as a reflection. As we look at the Internet using the dominant thing-orientation and technological determinism framing, we miss seeing the social that is as much and more what shapes the technical, even as people are then shaped by the technical in return.

Take a moment to reflect back on your journey through the Orange and Blue Units leading up to the start of the Rainbow Unit. Think especially about the construction of the different technical components that together have become a digital Little Free Library. Think also about the social justice storytelling activity which has been an active part at different times during this journey. Then work to envision the different ways in which these smaller parts have come together, at different times and in different ways, to create larger components and systems that are sociotechnical in nature, rather than just one or the other. Finish by thinking further about ways in which the large reflects the various small components, and the ways that these shape each other.

A WordPress-Based Public Information System

The Raspberry Pi Foundation includes instructions for a project to Build a LAMP Web Server with WordPress. **L**inux, **A**pache, **M**ySQL/**M**ariaDB, and **P**HP/**P**erl/**P**ython (LAMP), a term that became popular in the late 1990s, represents a bringing together of these sociotechnical systems to create a flexible **web stack**, a collection of software applications to perform specific tasks, in this case specific to web development. WordPress is a commonly used extension to the widely used base LAMP stack.

Back in the Orange Unit 4C: Getting Started with the Raspberry Pi, we explored the Raspberry OS, a GNU Linux operating system designed for the Raspberry Pi microcomputer and based on the Debian GNU Linux distribution. Linux itself is only the kernel of the operating system, the part that serves as the interface between the hardware and software of the computer. The GNU (GNU is not Unix) suite is most commonly used to provide the wider range of software used to give us the full operating system platform. The distribution further determines which GNU packages and other Linux-based packages of software to include. Debian was launched in 1993 and is composed only of free and open-source software. It has a more structured versioning release structure that serves some purposes well in providing greater stability, but is subpar for others, as it may slow down change for others. These are factors which were considered when the Raspberry Pi Foundation decided to use Debian as its base.

Linux installation, check. As Raspberry OS also comes with **P**ython, check to that, also.

We've now completed the Apache web server and PHP installation steps from the Raspberry Pi project to Build a LAMP Web Server with WordPress.

Apache installation, check. **P**HP installation, check.

We've now got the **LAP** portions of the **LAMP** web stack installed and running both for single-purpose Python web applications and for general-purpose static and dynamic webpage applications. So far, the instructions have been provided through the textbook. The text for these instructions primarily comes copied and pasted from Build a LAMP Web Server with WordPress. This can legally be done because the Raspberry Pi Foundation published these under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license. You'll also notice that this textbook is released using that same license. This is required when you see the ShareAlike listing within a specific Creative Commons license used. The Creative Commons was already the primary assumption for publications released through the Illinois Open Publishing Network, but as the author, I decided to modify the form of Creative Commons license to be used to have open use of the Raspberry Pi Foundation text, so that I can adapt and remix the Apache and PHP instructions to fit the context of this book. Take a moment to learn more about the Creative Commons Share Alike license as another example of the small shaping the large.

What remains is the installation of popular database engine **M**ySQL and the WordPress CMS.

MySQL is a **database** system. It uses the structured query language identified through its abbreviation, SQL, and spoken as either S-Q-L or "sequel." It is used to manage the data found in relational database management systems. Data itself can be stored in many ways, including in text files or in spreadsheets. Spreadsheets help to organize the data using rows and columns, and can further organize them using multiple sheets. Links can even be made between different cells across different sheets. This is a base level database. But relational database management systems take this up many notches by bringing in data definition, update, retrieval, and administration functions. It is here that SQL also comes in, by providing ways to access multiple records from a database with one command. In part, this is because SQL requests do not need to specify how to reach a record.

MySQL is an open-source relational database management system (RDBMS) founded by the Swedish company MySQL AB in 1994. Sun Microsystems acquired MySQL AB in 2008, and then Oracle acquired Sun Microsystem in 2010. There is now both a GNU General Public License (GPL) version and a proprietary version. In 2010, one of the co-founders of MySQL launched MariaDB, a community-developed, commercially supported fork of MySQL. MariaDB seeks to maintain high compatibility with MySQL, and is the RDBMS now used in the Raspberry Pi foundation project instructions, starting with step 4.

WordPress is a free and open-source content management system (CMS) written in PHP and paired with MySQL/MariaDB. It was first released in 2003 and is used by many websites around the world, either directly or through platforms built using WordPress as their base. WordPress is licensed under the GPLv2 from the Free Software Foundation. Take a few minutes to explore more about this mission, history, and story of WordPress at: https://wordpress.org/about/. Here's where you go to download WordPress for installation, along with a range of themes, plug-ins, and other material. Do so by moving to step 5 of the Raspberry Pi Foundation LAMP with WordPress project.

There is also a hosted version of WordPress at https://wordpress.com/. Go into the about page to compare how the story shared through this site compares to the one in the .org site. How is the story similar? Different? What are their different audiences?

Key Takeaways

Since early in the Orange Unit, we've been exploring aspects of data—the symbols used to present real-world entities created through abstraction—and information—the formatted, structured, and organized outcome of the process of meaning-making and value assessment from data. By the start of the Rainbow Unit, we were able in chapter 1B: Connecting Our Electronic 'Thing' to a Wider World to read data such as 0, 1, 77, 21, and 2144772, read from the Circuit Playground Express every second using Python code on the Raspberry Pi to display this data in a terminal window or a web browser as information using format, structure, organization, and meaning-making names, such as Note, Button, Temp, Light, and Runtime. But since the Orange Unit, we've also worked to step up a major level of information through the ongoing meaning-making through social justice storytelling. As we ended the Blue Unit, chapter 4B: Raspberry Pi Counterstory Little Free Library provided us with the means to readily make available up to seven stories, using the Circuit Playground Express in combination with the Raspberry Pi running Python code. In both Blue Unit 4B and Rainbow Unit 1B, we made use of single-purpose MakeCode and Python programs to accomplish a specific task.

We've now worked through a series of exercises resulting in a LAMP web server with WordPress. We've also worked to identify various decisions influencing the design and implementation of many of the sociotechnical components that make up this web stack, as installed using the base instructions provided by the Raspberry Pi Foundation. We now have the ability to write dynamic HTML using data and information of many types stored within both text and database files. This data and information can be further processed as part of meaning-making by individuals and groups of people, resulting in different format, structure, and organization, and thereby creating yet different information. We concluded this set of exercises by adding the MariaDB Relational Database Management System (RDBMS) and WordPress Content Management System (CMS) to our LAMP server to facilitate the ongoing work of collecting, organizing, formatting, and distributing data and information in collaboration with web clients. In doing so, we've created a higher-level sociotechnical network information system.

From Orange Unit Session one forward, we've worked to demystify the social and technical components that together shape the larger daily use technologies amongst us. These mutually shaped sociotechnical components and systems shape us even as we practice innovation-in-use to shape these devices. We've now got a digital Little Free Library system with electronics, microcontrollers, single-purpose web servers, and a general-purpose LAMP web server with WordPress CMS. But we also now have worked to demystify in significant ways the basic building blocks of these various aspects, opening up agency to innovate-in-use these various components. In so doing, we have hopefully discovered ways that stock stories have been used to create a lens on the sociotechnical components and systems that aren't right, aren't just. Moving further, the goal has been to provide us with base skills for social justice storytelling in which we can discover, develop, and share counterstories that buck the status quo in different ways. These can come through concealed stories that serve as direct responses to stock stories, resistance stories that not only buck against stock stories but also highlight injustices, and emerging and transforming stories that (re)construct knowledge built on concealed and resistance stories. Going further, we ourselves are hearing the counterstories in ways that are (re)constructing knowledge—actionable information—that we can use as we design, build, put into practice, and/or innovate-in-use networked information systems as part of our own professional practices as information professionals getting into good trouble, necessary trouble.

Do Something New!

Set aside some time for reflection now to consider the three different networked information systems we now have on hand for our use for a range of purposes, including as a mobile Digital Little Free Library and Counterstory Player:

- The Raspberry Pi Counterstory Little Free Library
- The Raspberry Pi Special-Purpose Current Statistics Reporting and Remote LED Controller
- The Raspberry Pi General-Purpose LAMP with WordPress Server

1. What are some times when you would choose one of the three over the other two? Why?
2. What is an innovation-in-use you would like to do to ensure that each one better meets the needs of its use as you described above?
3. Are there ways that an innovation-in-use could help each of the networked information systems facilitate its use by different audiences to help them make the data and information actionable—that is, to help people turn it into knowledge?

The following are a few ideas for something new you can do to tighten your understanding of the key learning objectives of the textbook. In taking one of these ideas, it may be that you do not currently have the needed connections with community members to make this real. Further, your work through this textbook may have been done in a restricted timeframe which may be coming to an end. Thus, this may be more of a thought experiment with targeted skillset development, rather than a real-life, critically engaged community inquiry activity. But starting this activity with some imagining of members of a community that find themselves in the margins in a critical way, a way for which you need to stand up and say something, do something, will help you to bring a more holistic sociotechnical exploration into the chosen "Do Something New!" exploration, as you consider ways to professionally get into good trouble, necessary trouble as you move forward.

Expanding Your Digital Little Free Library

You've now set up a WordPress CMS on your Raspberry Pi, running under the Apache Server and connected to a MariaDB Server. How might you make use of these to grow your Digital Little Free Library into something that can facilitate data and information collection, management, analysis, and distribution to serve a specific need or opportunity? Beyond your community of practice, who needs to be brought into the conversation as part of a human-centered design and implementation? There are a range of ways in which today's networked information systems, including but not limited to social media, have incorporated systems, structures, policies, and practices that have been exclusionary for those on the margins of society. How can this system be used to address a community-identified limit situation? How can this be done in a way that allows those community members who have been oppressed and marginalized to become active innovators-in-use of the system, so as to ensure that it provides an actionable way for these members of the community to address the limit situation in a way that matters to them?

To integrate this into a learning opportunity, you might also check out the Raspberry Pi Foundation, which provides a range of other educational projects you can do to learn more about web servers, websites, and webpages using HTML, CSS, and JavaScript. More generally, the Foundation provides projects to further explore different software and coding languages. Some of these might be directly integrated into the WordPress space, while others might be a complement to that space.

What's Next?

For those who choose this adventure, how might you use this space strategically to prototype multiple subcomponents within the larger Digital Little Free Library system in order to "say something, do something" to "get in good trouble" in, with, and for a marginalized community in a way that matters to them?

MAZI Zone DIY Community Network

The core work with the Raspberry Pi within this book has made use of the newest version of the Raspberry Pi OS. But there are now Linux operating systems targeted for embedded devices to do a range of special-purpose tasks. Both the Raspberry Pi and the Circuit Playground Express are made using **embedded devices**. As an example, The OpenWrt Project is a Linux operating system designed to facilitate the routing of network traffic using a range of embedded devices, including the Raspberry Pi. The OpenWrt Linux operating system can thus be used to turn the Raspberry Pi into a router connecting a **Local Area Network (LAN)** to a **Community Area Network (CAN)** using wired and wireless Ethernet ports. For those with a deeper technical background, this may be of interest to explore as a means for building a Community Wireless Network by attaching a Raspberry Pi to a LAN using a wired Ethernet cable, and then setting up the wireless LAN to create a new LAN (e.g., see these instructions from Sparkfun).

Perhaps a better starting point for this introductory text, those wanting to Try Something New! might consider creating a MAZI Zone community network that includes not only a LAMP with WordPress server, but also additional programs, and even the possibility of linking together multiple Raspberry Pi's into a broader community network.

MAZI means "together" in Greek. The MAZI project is part of the Collective Awareness Platforms for Sustainability and Social Innovation initiative, promoted by the European Commission, and funded through the European Union's H2020 framework. Their goal is "to provide technology and knowledge in order to:

- empower those who are in physical proximity, to shape their hybrid urban space, together, according to the local environment and context.
- generate location-based collective awareness as a basis for fostering social cohesion, conviviality, participation in decision-making processes, self-organisation, knowledge sharing, and sustainable living.
- facilitate interdisciplinary interactions around the design of hybrid space and the role of ICTs in society."[13]

According to MAZI, do-it-yourself (DIY) networking using such toolkits "can serve two complementary objectives:

1. to improve Internet connectivity in a certain region or local area
2. to support local interactions and services."[14]

The MAZI Toolkit

The MAZI Toolkit is made up of three elements. All of these are available from the MAZI Toolkit page on this website.

1. Low cost hardware: Currently the Toolkit is using the Raspberry Pi system. Designs for making your own hardware casings will be available soon.
2. Software and applications: These are specifically developed by the MAZI Project, including a set of local web applications ready to be installed on the captive portal. The functionality will range from very simple communication services, like chatting, forums, wikis, and polls to more sophisticated collaborative applications for social networking, deliberations, community organizing, project development, etc.
3. Guidelines and knowledge, including examples and inspirations: Installation scripts and step-by-step guides are part of the toolkit, enabling you to build and deploy your own network zone, to configure a user-facing captive portal, and to select and customize software applications.

In addition, you can directly access the Toolkit guidelines on GitHub, which includes up-to-date documentation.

13. p. 4. *The Mazi Handbook.* http://www.mazizone.eu/wp-content/uploads/2018/10/MAZIEbookOct2018.pdf
14. p. 7. *The Mazi Handbook.* http://www.mazizone.eu/wp-content/uploads/2018/10/MAZIEbookOct2018.pdf

For this adventure, consider flashing a second 16GB MicroSD card with a ready-made MAZI toolkit image that includes the Raspberry Pi Operating System and the rest of the LAMP + WordPress stack, along with a number of other software applications that can be used by choice as part of your own MAZI Zone. This MAZI Zone can be used offline on its own, as part of a LAN or CAN intranet, or online with the Internet. You can use it exclusively as a web-based ICT, or you could also include Python-based IoT communications, integrating and innovating-in-use into your MAZI Zone exercises from session one of the Rainbow Unit. The MAZI website lists several test examples, but consider other ways this might be used as part of an Urban or Rural Community Network, or perhaps even in support of digital agriculture on that five acre farm seeking sustainable agriculture pathways.

What's Next?

For those who choose this adventure, how might you now use your MAZI Zone to prototype ways this can be designed, so that you could "say something, do something" to "get in good trouble" in, with, and for a marginalized community or communities within a geographic region?

From an HTTP Thing to Multiple Internet of Things Devices

Through the exercises of the Rainbow Unit, we've explored a few different ways we could bring together the Orange Unit electronics and the Blue Unit code with the Rainbow Unit internetworking to build a basic Internet thing. How might you now start combining multiple basic Internet things together to create a plural Internet of Things? This adventure could be done using the Python3 and/or Apache HTTP server setup already in place, done in combination with one of the above "choose your adventure" possibilities above to provide a richer web interface, or using the Adafruit IO. [Watch the video "All the Internet of Things – Episode 4 – Adafruit IO: An IoT Service for Everyone" online.]

The Adafruit Learning site also has instructions on how you can take a newly emerging pathway to set up your Raspberry Pi as a Home Assistant. Unlike IFTTT or Samsung Smart Things, your data can remain local, not needing to tie in with the Internet, or can be used in combination with the secured Adafruit IO to add in remote access.

And for those comfortable in CircuitPython who want to test out an open-source web application that includes data cleaning and transformation, numerical simulation, statistical modeling, data visualization, and machine learning, still another adventure could include CircuitPython with Jupyter Notebooks as a first exploration of how machine learning and weak artificial intelligence could be brought into these works. [Watch the video "Machine Learning & Artificial Intelligence: Crash Course Computer Science #34" online.]

What's Next?

For those who choose this adventure, how might you now use your Internet of Things to prototype a way you could "say something, do something" to "get in good trouble" in support of a marginalized community? This is a more technical activity but there may very well be members of the oppressed community who are ready and able to be an active member of your community of practice. However, it may also be that they do not have available time as they find themselves working multiple jobs and serving different community leadership roles because of the limit situations keeping them in the margins. In this case, you might work in collaboration with some to design for design after design so that community members can use these in ways that allow innovation-in-use to assure the IoT prototype can be adapted to advance a given activity overcoming the limit situation specific to the current community context.

Wrap Up

To the extent possible, working on the various activities of this textbook have been done within a community of practice. Bringing investigations together with reflection and discussion with others whose lived experiences are different than yours is essential if we are to identify unknown and unseen meanings, values, and intentions that are embedded within the words of text and within the design of hardware and software. As we come to better understand the codification within these, we can begin a process of decodifcation and recodification, making the data and information potentially actionable in some new way. It is to re-center our work as information scientists from a thing-orientation towards a person-orientation. And as noted several times throughout the textbook, it is to use this as an opportunity for doing pre-season training activities within a sports metaphor. Each is meant to strengthen a different aspect of ourselves and ourselves in relation to others.

But core through each of the activities within this textbook has been the advancement of sociotechnical skills in ways that prepare each member of our community of inquiry for a deeper, critically-engaged community inquiry. Ongoing pre-season action-reflection collective leadership practices are working to develop stronger muscle memory so that these methods are an active part of your daily practices as an information science professional moving forward. Such adventures will include in different ways at different times active bottom-up participatory action research in community, with community, for community throughout your lifetime. This is something we'll explore further in Rainbow Unit 4A: Recovering Community: Designing for Social Justice.

Comprehension Check

The Orange Unit: A Person-Centered Launch, Blue Unit: Computational Tinkering, and Rainbow Unit: Networks Big and Small each include a Key Learning Outcomes Self-Check covering various technical, information, cognitive, socio-emotional, progressive community engagement, and critical sociotechnical skills. These are again included within the Review chapter for each unit.

- Take a few minutes now to look back over these to consider your own progression from understanding less to understanding more the foundations from which development of these skills arise.

The Rainbow Unit 3B: A Person-Centered Network Information System Adventure brings these many different skills together as a concluding sociotechnical networked information system.

- In what ways have the various skills been comprehended sufficiently to take on this concluding adventure? What skills might be further advanced sufficiently through discussion and reflection within your current community of inquiry?

The Rainbow Unit 3B has also worked to explore ways one set of hardware can be used in a range of ways to collect, analyze, organize, format, and distribute data and information. How this is done matters as it is shaped by, and shapes, ways data and information are made actionable, that is, ways data and information are turned into knowledge by those who interact with this data and information. Unconsidered, we likely further algorithms of oppression in our personal and professional practices. If you currently do not have rich diversity within your community of practice, work to outline clearly how diversity would be brought into the collective leadership to assure this is the “good trouble” to which the late John Lewis refers.

- Take a few minutes now to create your own comprehension check that you can use in the coming days and weeks to affirm you are effectively meeting your learning objectives so that you can say something and do something moving forward personally and professionally so as to get in good trouble in support of personal and community social justice goals.
- Keep these on your desk or wall to glance at regularly as you work towards completion of the textbook activities, using these to help you keep interest and specificity of thought and actions in this last stretch.

4A: Recovering Community: Designing for Social Justice

Background Knowledge Probe

Design

As transitive verb:

1. To create, fashion, execute, or construct according to plan
    a. To conceive and plan out in the mind
    b. To have as a purpose
    c. To devise for a specific function or end

As intransitive verb:

1. To conceive or execute a plan
2. To draw, lay out, or prepare a design

As noun:

1.
    a. A particular purpose or intention held in view by an individual or group
    b. deliberate purposive planning
2. A mental project or scheme in which means to an end are laid down

   a. A deliberate undercover project or scheme
   b. **Designs** *plural*: aggressive or evil intent — used with *on* or *against*
3. A preliminary sketch or outline showing the main features of something to be executed
   a. An underlying scheme that governs functioning, developing, or unfolding
   b. A plan or protocol for carrying out or accomplishing something (such as a scientific experiment); *also*: the process of preparing this
4. The arrangement of elements or details in a product or work of art
5. A decorative pattern
6. The creative art of executing aesthetic or functional designs

Merriam-Webster, "Design"[1]

1. After reading this definition, put everything aside and pause for several minutes to reflect back on your journey through *A Person-Centered Guide to Demystifying Technology*. Explore especially the ways you've done aspects of design, whether as noun or verb, during this journey.
2. When you return, review the titles of each social and technical session within the Orange, Blue, and Rainbow Units. How does this compare to, and further shape, your initial reflections on your journey through this book?
3. Now skim the "Introduction to the Book" and the Orange, Blue, and Rainbow Overviews. How do these further shape your initial reflections on your journey through this book?
4. With these reflections in mind, take a few minutes to brainstorm and creatively document a vision or two you may have of a person-centered collective leadership design of a networked information system.

The Designer in Each of Us!

> What if our discipline made the shift from a science approach (organizing our professional knowledge in the form of testable explanations and predictions about the world) towards a design approach (identifying problems and addressing them with human-centered solutions)?
>
> Miguel A. Figueroa, Foreword to *Design Thinking*[2]

1. Merriam-Webster, "Design," accessed July 3, 2020, https://www.merriam-webster.com/dictionary/design.
2. Figueroa, Miguel A. Foreword to *Design Thinking* by Rachel Ivy Clarke. (Chicago: ALA Neal-Schuman, 2020), vii.

> **To conclude,** there are two possible futures for libraries: one is a passive future, in which libraries sit back and let others design solutions to information problems. The second one, and the one I know I prefer, is the one that libraries design. Design thinking is so pervasive in librarianship that libraries and librarians already have the power to create unique, powerful, value-laden experiences and help individuals, communities, and even societies solve information problems. It's time to embrace design as a fundamental component of librarianship in order to create a better future for all.
>
> Rachel Ivy Clarke, *Design Thinking*[3]

As noted by Merriam-Webster, design is a word with a range of definitions. To begin, let's consider this within a very broad framing: to make something, it first needs to be implicitly or explicitly designed. When we are trained within a profession to move from rote use of tools and techniques at our disposal in order to accomplish tasks, to instead expanding upon the tools and techniques to assure they work for different populations and contexts, we are being taught to design and make new things or to further innovate-in-use and remix existing things. We have moved from use to design. We all design personally and professionally to some extent or another. But only some of us are formally given the job description of designer. As we've journeyed through the Units of this textbook, we've been immersed within a deep hands-on social + technical dive into sociotechnical artifacts including electronics, software, and networks culminating with what is hopefully now a more holistic understanding of networked information systems. We've met, and made active use of, skillsets, frameworks, and standards employed by a wide range of information professionals in selecting, co-designing, appropriating, and innovating-in-use networked information systems. Throughout this deep dive, a central objective has been the development of a critical approach to sociotechnical artifacts and the advancement of community agency in appropriating technology to achieve individual and community development goals. Core is an intent to move away from information science professionals as passive recipients of networked information systems designed by others to address information problems, and instead build from our already existing professional "power to create unique, powerful, value-laden experiences and help individuals, communities, and even societies solve information problems."[4] Information science professionals *and* our patrons *each* move from use to design and back as needed to effectively take on information problems. In *Design Thinking*, Rachel Ivy Clarke notes:

> Supporting patrons has always been a key component of librarianship, which has prided itself on being a user-centered profession. Early American librarianship was rooted in service to library users with the goal of improving their lives through exposure to books, reading, and literacy. Reading was emphasized as a means of intellectual, moral, and social education and improvement. But this meant that the tools and services created by libraries for users were often based less on what users *wanted* to read and more on what librarians thought users *should* read; usually this meant classic, culturally uplifting texts in the Western

3. Rachel Ivy Clarke, *Design Thinking*, Library Futures 4 (Chicago: ALA Neal-Schuman, 2020), 54.
4. Clarke, *Design Thinking*, 54.

> literary canon. Although this agenda certainly had users at its center, today it is criticized for its cultural presumption and its imposition of certain values.[5]

Especially beginning in the mid-twentieth century, efforts have been made to move from *assumptions* to *investigations* of what is actually needed and wanted, such as through the use of "needs assessments." This form of design has been heavily influenced through **user-centered design (UCD)** in which "design is based upon an explicit understanding of users, tasks, and environments; is driven and refined by user-centered evaluation; and addresses the whole user experience. The process involves users throughout the design and development process and it is iterative. And finally, the team includes multidisciplinary skills and perspectives."[6]

However, as noted by Sasha Costanza-Chock in *Design Justice: Community-Led Practices to Build the Worlds We Need*, "Design always involves centering the desires and needs of some users over others. The choice of which users are at the center of any given UCD process is political, and it produces outcomes (designed interfaces, products, processes) that are better for some people than others (sometimes very much better, sometimes only marginally so). This is not in and of itself a problem. The problem is that, too often, this choice is not made explicit."[7] Clarke adds, "User-centered approaches, by their very name, focus on use: things like completing a task or accomplishing a goal. The word *user* reduces people to the use of a thing, rather than engaging with their experiences as human beings."[8]

While user-centered practices are works of design, user-centered design too easily results in the user being identified as an entity completing a task, ultimately a work of dehumanization. An alternate holistic approach incorporating empathetic techniques has emerged attempting to re-center human beings as individuals. "It's a process that starts with the people you're designing for and ends with new solutions that are tailor made to suit their needs. **Human-centered design (HCD)** is all about building a deep empathy with the people you're designing for; generating tons of ideas; building a bunch of prototypes; sharing what you've made with the people you're designing for; and eventually putting your innovative new solution out in the world."[9] A central aspect of human-centered design is to consider not only the actions of design work, but the thought processes that underly these actions. **Design thinking** emerged as a term in the 1960s to describe this key aspect of design found across a wide range of design disciplines.

Indeed, consider our own work throughout the sessions and units of this textbook, and you'll find ongoing cycles of action-reflection praxis — reading, watching, discussing + hands-on actions alone

5. Clarke, *Design Thinking*, 22-23.
6. "User-Centered Design Basics | Usability.Gov," U.S. Department of Health and Human Services, April 3, 2017. https://www.usability.gov/what-and-why/user-centered-design.html.
7. Costanza-Chock, Sasha. *Design Justice: Community-Led Practices to Build the Worlds We Need,* Information Policy (Cambridge, MA: The MIT Press, 2020), 77. https://designjustice.mitpress.mit.edu/.
8. Clarke, *Design Thinking*, 23-24.
9. Design Kit, "What is Human-Centered Design?", accessed July 17, 2020. https://www.designkit.org/human-centered-design.html.

and within pairs/small groups + individual and corporate reflections. Together, these move us from use to design to broader thinking regarding design processes and the shapers of design and product. There is the risk that as with so many other concepts that reach popular use, design thinking is taken out of the more rigorous, extended work of design as a mental pat of congratulations for being thoughtful and empathetic while continuing a user-centered approach. And there is a risk that this design thinking is done in ways that shape design to conform to dominant narratives. But critically, strategically, and communally used, design thinking becomes a valuable *mindset* while working to resolve ill-defined problems using critical abductive reasoning within a broader design justice approach as we'll explore in the next section. [Watch the video "What is Design Thinking?" online.]

In the 1990s, one of the lead faculty at the Stanford University Design School, David Kelley, also worked to bring design thinking into the field through the founding of IDEO, a global design and consultancy company. And through more recent funding by the Global Libraries program at the Bill & Melinda Gates Foundation, IDEO led to the creation of a Design Thinking for Libraries toolkit in collaboration with Chicago Public Library in the United States and Aarhus Public Libraries in Denmark.

Libraries are a center for information access and serve to benefit their people in a wide range of ways. Over time, approaches and perspective have had to change to adapt to new opportunities and needs. But the challenges are real, complex, and varied within a rapidly evolving information landscape. Design thinking within the library context seeks to foster deeper understanding of the needs of patrons and to engage with communities in new ways. The design thinking processes listed in the toolkit bring forward a few formulas that might prove useful in moving from learning about the terms and concepts of digital technologies to considering ways these technologies can be further designed and remixed by the communities of practice of which you are a part or whom you are serving within a support and resource role. And as information scientists, the activity toolkit may also prove a useful base as we enter into collaborations with design professionals who bring to the collective leadership deeper formal training in design techniques, including form, composition, balance, typography, visual literacy, color theory, etc. In practice, it's also helpful to include within the toolkit alternate terms to those within the "Design Thinking" framing, such as:

- Inspiration: Discovery, Interpretation, Empathize, Define
- Ideation: Ideate, Create, Prototype, Rapid Prototyping
- Iteration: Implementation, Experimentation, Deliverable, Test, Evolution

And consider, too, that this design thinking framework itself is another way of considering the ongoing inquiry cycle done in community, with community, and for community that includes: … ⟶ Ask ⟶ Investigate ⟶ Create ⟶ Discuss ⟶ Reflection ⟶ Ask ⟶ …

Design Justice

Library and information science is a profession dedicated to advancing the informed decision-making abilities of individuals, communities, and societies. Throughout the first half of the twentieth century:

> Libraries played a significant role by portraying themselves as socially uplifting agents, developing library services to meet the needs of European immigrant populations during the foundation years (1900–1917), or during the era of depression and war (1930–1945), and afterwards, where the focus was on economic recovery and revitalization. During these times, the library provided liberating directions for social justice outcomes in nurturing just and fair ideals, expanding the base of impact to include outreach populations, creating a service-based ethics in the profession, and forging partnerships with community-based social justice agencies towards common goals.[10]
>
> Bharat Mehra, Kevin S. Rioux, and Kendra S. Albright, "Social Justice in Library and Information Science"

As the twentieth century came to an end and we entered the twenty-first century, these works also included community networking, community technology centers, digital literacy training, and addressing the digital divide. "These included social justice activities and initiatives, even though the rhetoric and vocabulary of 'social justice' had not been significantly incorporated into the mainstream profession."[11] Mehra, Rioux, and Albright acknowledge there is no universally accepted definition for social justice, and moreover, social justice theory specific to library and information science is still within its infancy:

> Historically, social justice has been concerned with the tensions between: 1) the individual's right to choose her/his own ends; 2) conflicts with other individuals' rights to make similar choices; and 3) the debate on individual rights vs. the good of the community.[12]
>
> The lessons from the past are today providing libraries directions to develop a new approach that recognizes:
>
> 1. importance of outcome-based, socially relevant evaluation methods in assessing library services;
> 2. value of local experiences and ontologies and their representation into formalized organizational tools of information; and

10. Mehra, Bharat, Kevin S. Rioux, and Kendra S. Albright, "Social Justice in Library and Information Science," in *Encyclopedia of Library and Information Sciences* (CRC Press, 2009), 4823. https://doi.org/10.1081/E-ELIS3-120044526.
11. Mehra, Rioux, and Albright, "Social Justice in Library and Information Science," 4823.
12. Mehra, Rioux, and Albright, "Social Justice in Library and Information Science," 4820.

3. necessity in building equitable partnering efforts with disenfranchised constituencies.[13]

Human-centered design has served as an important transition point from previous user-centered approaches to design in helping the information science profession move towards a person-centering of sociotechnical design, implementation, and use. This textbook builds from over 20 years of social justice-oriented teaching, research, and practice that has included extensive use of service projects, community inquiry, and engaged scholarship to bring together the school and community to advance community goals. But a 2008 ethnographic study of my course "Introduction to Networked Information Systems" conducted by Junghyun An found that without greater criticality, incorporation of human-centered design techniques and social justice-oriented pedagogies and goals into design studio service projects too easily left true action-reflection praxis at a superficial level.[14] An's valuable study underlined important aspects, and highlighted key shortcomings, of my own ongoing research that has emphasized the essential need for critical ongoing thinking regarding the various social issues and problems within our own technology design and use practices, as well as those of others shaping these technologies. And of central importance is that these reflections include collaborative discussions within our own profession and related professions, and also very importantly with the wider range of stakeholders, especially those marginalized and oppressed via the shaping that is embedded within and emerges from these technologies.

This textbook was birthed through deep nights of the soul that especially emerged beginning in the late 2000s. Rather than learning-by-doing specifically as a service-learning action, it now seeks to be a learning-by-doing action-reflection praxis as a primer for true ongoing professional design justice works. The action-reflection learning-by-doing praxis of this textbook happens as we use our deepening understanding of the components and concepts underlying electronics, software, and networks to build a picture, or codification, of these artifacts within real situations and people, including those forgotten and marginalized. It happens as we then work to decodify the artifacts and associated codes by looking at certain aspects in order to build a new lens on the larger sociotechnical artifacts. It continues to happen as, through this work of action and reflection, we reach towards a recodification to move from magical thinking about networked information systems that are designed by others for our passive professional use to meet patron needs, and instead envision an alternate, social justice path for our profession that centers around community inquiry and collective leadership engaging all members of local communities.

But for this to happen, we need to assure our human-centered design thinking is centered within a **design justice** approach. Costanza-Chock argues that this work is:

13. Mehra, Rioux, and Albright, "Social Justice in Library and Information Science," 4824.
14. An, Junghyun. "Service Learning in Postsecondary Technology Education: Educational Promises and Challenges in Student Values Development," University of Illinois at Urbana-Champaign, 2008. https://hdl.handle.net/2142/17387.

> …about the relationship between design and power. It's about the growing community of designers, developers, technologists, scholars, educators, community organizers, and many others who are working to examine and transform design values, practices, narratives, sites, and pedagogies so that they don't continue to reinforce interlocking systems of structural inequality. It's about design, social justice, and the dynamics of domination and resistance at personal, community, and institutional levels. In essence, it's a call for us to heed the growing critiques of the ways that design (of images, objects, software, algorithms, sociotechnical systems, the built environment, indeed, everything we make) too often contributes to the reproduction of systemic oppression. Most of all, it is an invitation to build a better world, a world where many worlds fit; linked worlds of collective liberation and ecological sustainability.[15]

Costanza-Chock works towards a recognition that development of new information and communication technologies (ICT) needs to "not only take shape in Silicon Valley, they also emerge from marginalized communities and social movements, both during waves of spectacular protest activity and also in everyday life," in order to "advance the growing conversation about the pitfalls and possibilities of design as a tool for social transformation."[16] The Design Justice Network and its underlying principles are central to Costanza-Chock's vision. Considered part of a living document, these principles emerged over several years of work as part of the Allied Media Conference held in Detroit. As of a summer 2018 update, the Design Justice Network Principles are:

> Design mediates so much of our realities and has tremendous impact on our lives, yet very few of us participate in design processes. In particular, the people who are most adversely affected by design decisions — about visual culture, new technologies, the planning of our communities, or the structure of our political and economic systems — tend to have the least influence on those decisions and how they are made.
>
> Design justice rethinks design processes, centers people who are normally marginalized by design, and uses collaborative, creative practices to address the deepest challenges our communities face.
>
> Principle 1: We use design to **sustain, heal, and empower** our communities, as well as to seek liberation from exploitative and oppressive systems.
>
> Principle 2: We **center the voices of those who are directly impacted** by the outcomes of the design process.
>
> Principle 3: We **prioritize design's impact on the community** over the intentions of the designer.

15. Costanza-Chock, *Design Justice*, preface, https://designjustice.mitpress.mit.edu/.
16. Costanza-Chock, *Design Justice*, preface, https://designjustice.mitpress.mit.edu/.

> Principle 4: We view **change as emergent from an accountable, accessible, and collaborative process**, rather than as a point at the end of a process.[17]
>
> Principle 5: We see the role of the **designer as a facilitator rather than an expert**.
>
> Principle 6: We believe that **everyone is an expert based on their own lived experience**, and that we all have unique and brilliant contributions to bring to a design process.
>
> Principle 7: We **share design knowledge and tools** with our communities.
>
> Principle 8: We work towards **sustainable, community-led and -controlled** outcomes.
>
> Principle 9: We work towards **non-exploitative solutions** that reconnect us to the earth and to each other.
>
> Principle 10: Before seeking new design solutions, **we look for what is already working** at the community level. We honor and uplift traditional, indigenous, and local knowledge and practices.
>
> Design Justice Network, "Design Justice Network Principles", emphasis from document[18]

Design justice re-centers the role of digital literacy training using what Virginia Eubanks calls **popular technology**. Introduced in her 2007 article "Popular technology: exploring inequality in the information economy" and expanded further in her books *Digital Dead End* (2011) and *Automating Inequality* (2017), Eubanks systematically investigates the ways high-tech tools and jobs continue and expand social inequalities within the United States. To move beyond this dominant narrative, Eubanks brings forward an alternative digital literacy agenda that "make[s] technology analysis and development both relevant and empowering to people who live in persistent poverty, by drawing on traditions of popular education in South America (articulated by Paulo Freire) and the United States (articulated by Myles Horton and the Highlander Research and Education Center)."[19] Eubanks underlines that as a digital literacy strategy, "popular technology reminds us that technology is not a destiny but a site of struggle."[20] A popular technology approach:

- Resists oppression in the form of exploitation, marginalization, powerlessness, cultural imperialism, and violence;
- Draws on cognitive, cultural, and institutional difference as a resource; and

17. This principle was inspired by and adapted from the Allied Media Network Principles.
18. Design Justice Network, "Design Justice Network Principles," accessed July 17, 2020. https://designjustice.org/read-the-principles.
19. Eubanks, Virginia. "Popular Technology: Exploring Inequality in the Information Economy." Science and Public Policy 34, no. 2 (March 1, 2007): 131. https://doi.org/10.3152/030234207X193592.
20. Eubanks, Virginia. *Digital Dead End: Fighting for Social Justice in the Information Age* (Cambridge, Mass: MIT Press, 2011), 155.

- Engages in participatory decision making in agenda setting, design, implementation, and evaluation.

In the Laura Flanders Show episode below, Eubanks further introduces *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor* before describing the 2018 Allied Media Conference and the "Our Data Bodies" project, developed in collaboration with members of the Detroit Community Technology Project. These all serve as examples of coalition work and collaboration that should be the essential centering of all radically reconsidered digital literacy training, technology design, and sociotechnical artifact and system implementation. Eubanks highlights the Allied Media Conference as an "incredible space where we have difficult conversations about media, technology, and social justice. But the folks who are here are makers. They're organizers, they're activists, so they make spaces and they make movements. But they also make stuff." She highlights the math-washing that leads us to believe things, issues, and solutions are more complicated than we can handle: "It's true there might be a really technically complicated system that takes a little bit of breaking down for us to understand, but we understand the problems and we understand the solutions and we understand actually even some of the really complicated technologies that are out there once we have the language, the basic language to do it." [View the video "Automating Inequality: Virgina Eubanks" to hear her story.]

The Studio Space

Makerspaces, Tinkering Studios, Fab Labs, and other creative spaces have become increasingly commonplace as workspaces bringing people together to engage in conceiving, designing, and developing new artifacts. They often incorporate both traditional crafts and also emerging digitally based crafts. Beyond general action-reflection that includes various inspiration, ideation, iteration, and "act, investigate, create, discuss, reflect" cycles, a key component of effective design is the *critique* process. Clarke describes the critique in *Design Thinking*:

> **Critique** is a rigorous form of evaluation that is central to the discipline of design. Critique may call to mind scary memories of harsh, negative criticism, perhaps in front of peers, like reading a poem aloud in a creative writing class only to have the instructor and classmates rip it to shreds. However, well-executed design critique is not subjective negativity. Critique is not about whether you "like" a design or approve of its aesthetics. A good critique says *why* something does or does not work for that individual. It asks questions and elicits the rationale for a design decision. A critique is about discovering what's not working in order to make a design better. It can be a hard process to learn to take critique well, and even harder to learn to give a critique well. The critique of a design—especially one you are emotionally invested in—can feel like a personal blow. This is why it's important to seek critique throughout the design process.[21]

21. Clarke, *Design Thinking*, 44.

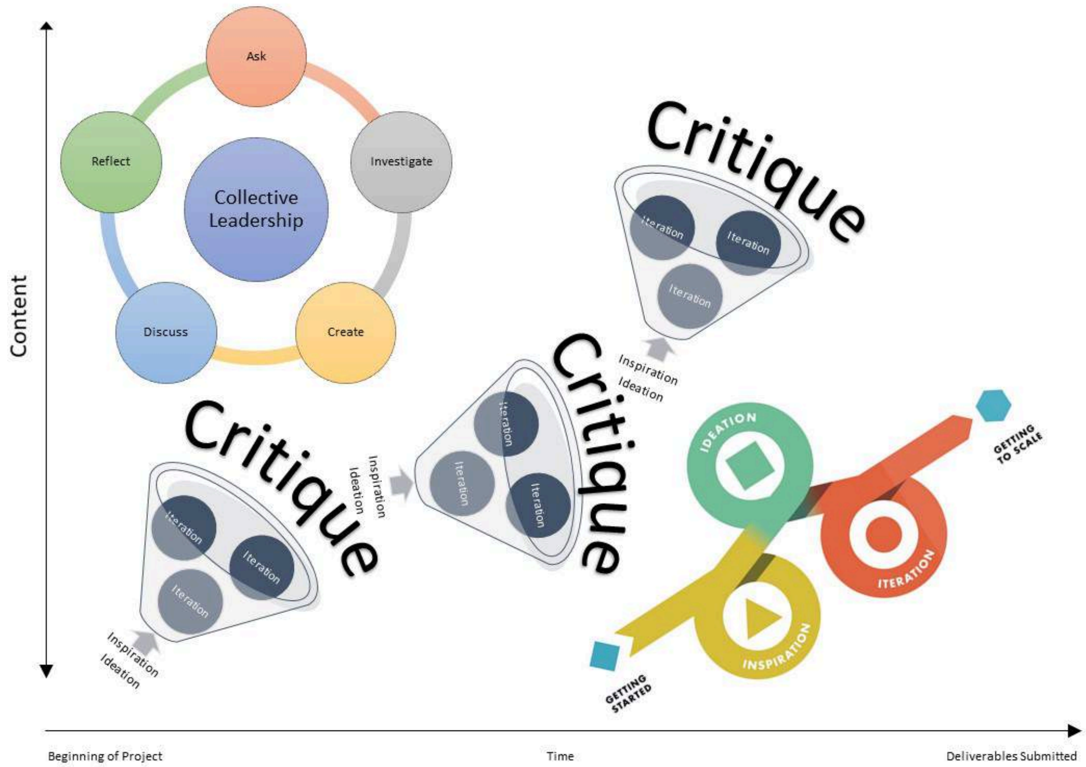
Critique can take many different forms, and it is often helpful to make strategic use of selected forms at different points within the design process. The following is a list of some of these I've found helpful within my community informatics studios:

Table 2. The design critique as a model of distributed learning.[22]

Critique Type	Brief Definition	Mapped to Traditional Academic Practice
Formal	Public, invited, summative, evaluation events	Structured and summative presentation formats: peer-reviewed journals, advanced degree oral defenses
Seminar/ Group	Semipublic, less formal, more generative, but occasionally evaluative	Symposium format in which development of ideas or products is stressed as a group
Desk	Conducted with mentor or instructor, highly generative, modeling of cognitive behavior potential for incidental learning	Modeling for graduate students, professional participation, individualized instruction
Peer	Highly informal, generative, voluntary or on request	Structured, scaffolded critique; informal sharing or practice of work; 3-person critique

Rooted in the apprentice model of learning in which students study with master craftspeople or artists to develop their craft, **studio classes** emphasize learning by doing, often through iterative, multimodal analysis, proposition building, design, making, and critique of different alternatives that might address community-based problems. It has been an integral pedagogy in practitioner-based fields such as architecture, urban planning, and fine and applied arts. Studio-based learning aligns closely with John Dewey's concept of experiential learning. For instance, Dewey emphasized the importance of helping students shape their purpose for a given activity by constructing a plan based on their impulses, past experiences, and community knowledge to maximally shape the current learning environment. In this way, teachers act more like "guides" to assist students in developing and implementing their design and making choices. Students and instructors work together within a studio space that serves as a model of professional practice. Regular written reflections also help students think more deeply about the paths that lead them to their final project processes, outputs, and outcomes.

22. Adapted from "Application of critique elements in academic practice and instructional design" © Springer Science+Business Media, LLC 2012. From: Brad Hokanson, "The Design Critique as a Model for Distributed Learning," in *The Next Generation of Distance Education: Unconstrained Learning*, eds. Leslie Moller and Jason B. Huett (Boston, MA: Springer US, 2012), 71–83. https://doi.org/10.1007/978-1-4614-1785-9_5.



Scaling ideas through multiple community inquiry cycles using collective leadership as a foundation for iterative testing and critique spirals

Much of the service-learning coursework that has guided the creation of this textbook emerged originally through my work as part of the East St. Louis Action Research Project (ESLARP), one of the oldest community-university partnership programs in the United States. The University of Illinois Urbana-Champaign was pushed to engage with the East St. Louis community, situated 175 miles away, when in 1987 State Representative Wyvetter Younge aggressively worked to bring the University into this most devastated city in the state, in order to provide essential assistance through **participatory action research (PAR)** projects in community, with community, doing collective inquiry and experimentation grounded in lived experiences and social histories. Later, Recreation Sports and Tourism, Library and Information Science, and Law joined the collaboration.[23] It was here that I first began unwittingly making use of studio techniques for analysis, proposition building, design, making, and critique as I watched the works of other university instructors. I also later began a series of community-university collaborations to develop and introduce to the LIS profession a formal **Community**

23. A very insightful reflection on this history of ESLARP can be found here: Sorensen, Janni and Laura Lawson, "Evolution in Partnership: Lessons from the East St Louis Action Research Project," *Action Research* 10, no. 2 (June 2012): 150–69. https://doi.org/10.1177/1476750311424944.

Informatics Studio course that centralizes place-making and popular technology workshops, within which sociotechnical artifact design and implementation serves a valuable supporting role.[24][25] This aligns strongly with the person-centered, hands-on guide for working with information and communications technologies as covered in this textbook, and provides a potential tool for establishing and further advancing design justice as a part of our ongoing community inquiry activities in daily professional field practices.

Whether in the Makerspace or Tinker Studio, as part of design justice projects and networks, or in a community center meeting room, you have knowingly or unknowingly joined into a studio which includes various aspects of design and making. While some will have had formal training as designers, all of us bring in our lived experiences doing design practices. These movements are working to break down barriers to full participation in the analog and digital realms of sociotechnical artifacts, and to advance empowerment for all. But as Angela Calabrese Barton and Edna Tan note in *STEM-Rich Maker Learning: Designing for Equity with Youth of Color*:

> At the same time, it is important to ask questions about how the movement defines who makers are, what makers do, and what kinds of access makers need to tools and opportunities to keep making. These questions cannot be divorced from considering the social, racial, gendered, economic, and political conditions in which particular makers are bound. Espousing an egalitarian vision of making may symbolically level the playing field, while the reality is that access and opportunities to make for some groups of the population continue to remain sporadic.[26]

Barton and Tan further note: "It is always rooted in the history and geographies of young people's lives and in the broader context of making and makerspaces in the United States and beyond."[27] Problem identification and analysis, design, and making practices and expertise build through iterative and incremental works. But what types of practices and expertise are built? How does this go on to impact the longer-term shaping of people, communities, and societies? For us to move from social justice intent to truer forms of social justice practices, it is essential for the information science profession to continue our progression implementing design justice principles and practices.

This chapter serves as a brief general introduction to design thinking, design principles, and design processes, and the ways that these need to center around design justice as one aspect of addressing

24. Wolske, Martin, Deven Gibbs, Adam Kehoe, Vera Jones, and Sharon Irish, "Outcome of Applying Evidence-Based Design to Public Computing Centers: A Preliminary Study," *The Journal of Community Informatics* 9, no. 1 (November 25, 2012). https://doi.org/10.15353/joci.v9i1.3188.
25. Wolske, Martin, Colin Rhinesmith, and Beth Kumar, "Community Informatics Studio: Designing Experiential Learning to Support Teaching, Research, and Practice," *Journal of Education for Library and Information Science* 55, no. 2 (April 2014): 166–77. http://hdl.handle.net/2142/48952.
26. Barton, Angela Calabrese and Edna Tan, *STEM-Rich Maker Learning: Designing for Equity with Youth of Color* (New York: Teachers College Press, 2018), 1-2.
27. Barton and Tan, *STEM-Rich Maker Learning*, 107.

real-world problems with community-based and -led social justice action and reflection praxis. To do this, Costanza-Chock notes that "design justice builds on, but also differs in important ways from, related approaches such as **value-sensitive design**, **universal design**, and **inclusive design**."[28] Others, too, are finding it important to bring together a breadth of design approaches while also considering ways in which designing for values will need to differ from certain approaches given current contexts.

Located in The Netherlands, [Delft Design for Values Institute] brings together a diversity of design approaches, theoretical backgrounds, considered values, and application domains within the design field more broadly. Internal and external value dynamics across the broad spectrum of stakeholders need to be grasped and connected to achieve a more inclusive and successful design. [Watch the video "Designing with Values in Complex Projects" online.]

Pieter Vermaas et al. note how "increased attention in design for values of users and of society at large has found its way also to design methodology."[29] Modern design methods have moved from a previous focus on functionalities, affordances, and disaffordances, to contemporary methods that use methodologies such as ethnographic research to clarify the values at play in the problems of users, for instance regarding conflicting values of safety and privacy. To this end, a common design method bringing together user perspectives and user-driven design is **participatory design**.

But as the scale of design outcomes expands, it is sometimes necessary to move beyond user-driven user perspective change, to broader user-driven social perspective change using **transformation design** methods to broaden the distribution of power within the decision-making process and bring forward a broader social perspective, as may be the case in the design of backbone Internet services used to connect multiple First Nations first-mile local area networks. As the complexity continues to increase, it becomes ever more challenging for design to be directly user-driven, and therefore design for values sometimes uses larger-scale, designer-driven **social implication design** methods. Still, even as complexity increases, as Fischer and Herrmann note in "Meta-Design: Transforming and Enriching the Design and Use of Socio-Technical Systems," a **meta-design** framing can be used to *design for design after design*, facilitating **user-as-designer**, or what Bruce et al. call **innovation-in-use**, at the use time of a product.[30][31]

28. Costanza-Chock, *Design Justice*, 46.
29. Vermaas, Pieter E., Paul Hekkert, Noëmi Manders-Huits, and Nynke Tromp, "Design Methods in Design for Values," in *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains*, eds. Jeroen van den Hoven, Pieter E. Vermaas, and Ibo van de Poel (Dordrecht: Springer Netherlands, 2015), 179–201. https://doi.org/10.1007/978-94-007-6970-0_10.
30. Fischer, Gerhard and Thomas Herrmann, "Socio-Technical Systems: A Meta-Design Perspective," *International Journal for Sociotechnology and Knowledge Development* 3, no. 1 (2011): 1–33. https://doi.org/10.4018/jskd.2011010101. Also available at https://l3d.colorado.edu/wp-content/uploads/2021/01/Published-JOURNAL-version.pdf.
31. Bruce, Bertram C., Andee Rubin, and Junghyun An, "Situated Evaluation of Socio-Technical Systems," in *Handbook of Research on Socio-Technical Design and Social Networking Systems*, eds. Brian Whitworth and Aldo de Moor, 2:685–98. Information Science Reference (Hershey, PA: IGI Global), 2009. https://doi.org/10.4018/978-1-60566-264-0.ch045.

Community technology design, as with many other areas of design, requires the bringing together of many different fields of research and practice that also include human-computer interaction (HCI), design studies, computer science, community studies, and community informatics. This is not something the information sciences profession can do in a silo. For all involved, concerns continue to grow regarding the cultural black box in community technology design, expanding on the existing range of social shaping of technology concerns that have been raised throughout this textbook.[32] It is for this reason that design justice might provide an essential central gathering point if we are to rapidly transition from a "thing-oriented" society to a "person-centered" society.

Lesson Plan

In the last chapter, you were introduced to a famous quote by the late John Lewis, an American civil rights icon who was part of the 1961 Freedom Rides, served as the first chairperson of the Student Nonviolent Coordinating Committee (SNCC), and was a leader with the Reverend Dr. Martin Luther King, Jr., of the march on the Edmund Pettus Bridge in Selma, Alabama, in 1965, that was violently halted by police and later became known as "Bloody Sunday." John Lewis went on to direct the Voter Education Project before being appointed by then-President Carter to lead ACTION, the umbrella federal volunteer agency that included the Peace Corps and Volunteers in Service to America (VISTA). In the 1980s, Lewis became a politician in the Atlanta city council, and then served as a representative in the U.S. Congress. For Lewis, each of these were part of his larger philosophy of a life well lived.

> My philosophy is very simple: When you see something that is not right, not fair, not just, you have to stand up, you have to say something, you have to do something.
>
> My mother told me over and over again when I went off to school not to get into trouble but I told her that I got into a good trouble, necessary trouble. Even today I tell people, "We need to get in good trouble."
>
> John Lewis, interviewed by Valerie Jackson for StoryCorps[33]

Getting into "good trouble" can take many forms throughout one's life as we participate in ongoing inquiry cycles of asking the better question, investigating possibilities, doing creative acts to address limit situations, taking part in community discussions, and doing ongoing works of reflection. These are the works of recovering community through design for justice.

32. For more on this idea, see: Sabiescu, Amalia, Aldo de Moor, and Nemanja Memarovic, "Opening up the Culture Black Box in Community Technology Design," *AI & Society* 34, no. 3 (September 1, 2019): 393–402. https://doi.org/10.1007/s00146-019-00904-z.

33. Lewis, John and Valerie Jackson, "The Boy From Troy: How Dr. King Inspired a Young John Lewis," StoryCorps, February 20, 2018. https://storycorps.org/stories/the-boy-from-troy-how-dr-king-inspired-a-young-john-lewis/.

This session especially focuses on design frameworks and approaches that 1) recognize the designer in each of us, and 2) the essential need for an integration and practice of design to advance the recovery of community. As with all aspects of popular education, these cycles do not bring us to an end point, but rather bring us to new starting points as we continue true social justice works. It is for this reason that popular technology training and design-justice-related processes cannot be done exclusively, or sometimes even primarily, within formal education and professional practices. But where and how this should be done is not something that can be taught. It is something that needs to be discovered in community, with community, and for community. Core to the lesson plan for this session, then, is a goal of opening up possibility for you to discover individually and, to the extent possible within your communities, what introductory development of design frameworks and processes you specifically still need.

Essential Resources:

This article is a 2014 Keynote given at the Illinois State University Critical Media Literacy Conference, and was written by five University of Illinois students who tested the feasibility of several of the key aspects of the De-mystifying Technology workshop framework that is now represented in this book. This feasibility was tested as part of a participatory action research process in community and with community that included design as one part of a Community Informatics Studio course, and gives a valuable example of the critical design process in action within its broader community inquiry context.

- Stangl, Angela; Haniya, Samaa; Naples, Kim; McCoy, Casey; and Ransberger, Rebecca, "Promoting Digital Literacy: The De-mystifying Technology Workshop for Families" (2020). Critical Media Literacy Conference. https://ir.library.illinoisstate.edu/criticalmedialiteracy/2014/Keynote/2

The following case studies and toolkits should be seen as some possible pathways from which you could choose in support of your own lesson plan. Some of these have a specific design focus, while others are community-centered activities in which design justice principles and practices have played a role.

- Lorini, Maria Rosa, Amalia Sabiescu, and Nemanja Memarovic. "Collective Digital Storytelling in Community-Based Co-Design Projects: An Emergent Approach." *The Journal of Community Informatics* 13, no. 1 (2017): 109–36. https://doi.org/10.15353/joci.v13i1.3296. This resource from the Orange Unit is a case study not just of digital storytelling, but also more broadly within co-design projects.

- Strohmayer, Angelika, and Janis Meissner. "'We Had Tough Times, but We've Sort of Sewn Our Way through It': The Partnership Quilt." *XRDS: Crossroads, The ACM Magazine for Students* 24, no. 2 (December 19, 2017): 48–51. https://doi.org/10.1145/3155128. Also from the Orange Unit, this case study especially highlights a feminist framework for human-centered computing and HCI.

- Cumbula, Salomao David, Amalia Sabiescu, and Lorenzo Cantoni. "Community Design: A Collaborative Approach for Social Integration." *The Journal of Community Informatics* 13, no. 1 (March 22, 2017). https://doi.org/10.15353/joci.v13i1.3299. This case study highlights a successful co-design and implementation of a community multimedia center, and notes ways that broader-scale research and development projects can be done in ways that also specifically bring local stakeholders into the design process.

- FirstMile. "Community Stories," 2016. http://firstmile.ca/community-stories-2/ and FirstMile. "Free Online Course: Colonialism and the e-Community," 2016. http://firstmile.ca/free-online-course/. The First Mile consortium of First Nations have worked since 2005 to bring Information and Communication Technologies (ICT) to remote and rural First Nations using a critical lens. As noted in the parent webpage, these "stories outline innovative, industry-leading uses of ICTs and broadband by First Nations communities and organizations in areas including art, education, and health." The free online course on Indigenous peoples and the e-Community is a living resource that provides essential background information for all seeking to reframe ICT within this critical lens.

- de Moor, Aldo. "Citizen Sensing Communities: From Individual Empowerment to Collective Impact," In *Proceedings of the 17th CIRN Conference*, November 6-8, 2019, 91–101. Monash Centre, Prato, Italy. https://www.monash.edu/__data/assets/pdf_file/0004/2219800/CIRN2019-complete.pdf and de Moor, Aldo. "Increasing the Collective Impact of Climate Action with Participatory Community Network Mapping." *Livingmaps Review* no. 8 (April 29, 2020). http://livingmaps.review/journal/index.php/LMR/article/view/197. These two articles provide case studies highlighting how a research consultancy firm can serve an essential role in building collaborative common ground between stakeholders in an organization, network, or community.

- Commotion Wireless. "Where It's Used." Accessed July 22, 2020. https://commotionwireless.net/about/where-its-used/ and Commotion Wireless. "Commotion Construction Kit – Planning." Accessed July 22, 2020. https://commotionwireless.net/docs/cck/planning/. These case studies and design activity are part of a larger toolkit of free, open-source communication technologies and design processes leading to establishment of community wireless mesh networks.

- M-Lab. "What Is Measurement Lab?" Accessed July 22, 2020. https://www.measurementlab.net/about/ and M-Lab. "Papers, Presentations, and Regulator Filings," Accessed July 22, 2020. https://www.measurementlab.net/publications/.M-Lab. "Internet Measurement Tests." Accessed July 22, 2020. https://www.measurementlab.net/tests/. These tools and publications are not specific to design processes, but rather provide an essential toolkit for information collection as part of the information search process. Its use has included work within public library settings. This information can then be incorporated into data storytelling that both helps within a collective leadership design process and also within the structuring of community group critiques.

Professional Journal Reflections:

1. Return now to the reflections, brainstorming, and vision documentation of the Background Knowledge Probe at the start of this session. What has remained constant and what has transformed within those as you've reviewed the session materials and entered into a critique?
2. Review your reflections from session three of the Rainbow Unit. Is there anything you would further revise as you explored past reflections within the context of session three and now session four?
3. What are some additional thoughts regarding next steps you need to take individually and within your communities of practice in response to your journey through A Person-Centered Guide to Demystifying Technology?

4B: Community-Centered Design: An Emergent Strategy for Community Organizing and Action

As a teenager working in the family sawmill, I found myself working closely with one of my dad's cousins who worked at the sawmill over the winter months, when he wasn't working as a road worker the rest of the year, when the roads and the land weren't frozen. He started referring to me as a buffull—a cross between a bull and a buffalo. While I could often be a bull in a china shop, knocking down the beautiful plates, bowls, and cups all around me, I also was someone who would join in community, our backs to each other to take on whatever might be out there. Over time I came to see the deeper meaning of the term.

I was an individual bull from a line of bulls that had a knack for tearing things apart to figure out how to put them back together. We were tinkers. While at times we left a wake of destruction behind us, ultimately, we seemed to find a way to fail forward. The larger something was, the more small parts could be found inside that needed to be torn apart and rebuilt. The bull side of "buffull" was meant to serve as a reminder to temper the vigor in my movements, but to not lose it, either. And so it's been session by session over this book, as we've learned to tear down and rebuild parts, whether on a breadboard, within a microcontroller, or as part of a server. The large is a reflection of the small components of which it is built. The building blocks of a circuit on the breadboard are the base of the building blocks for a microcontroller, which are the base of the building blocks computer, which are the base of a network of controllers and computers. Or so it is within the tinker mindset of a bull.

I am also of the buffalo clan. My cousin was a Kublick, as was my grandmother Wolske, who bore my dad after her arrival from Eastern Europe to the United States. My mom was a Henkelmann born from the womb of a Rosler in Eastern Europe. All four lines were ethnically Germanic, but never had formal citizenship within a nation-state, as they roamed looking for places of community and space for growing and crafting. The buffalo know how to bring backs together to form a circle of protection. But they also provide a wide viewing through many different lenses. Different members are able to see different distances and depths, depending on their angle of view. Together, there is a growing holistic understanding regarding the lay of the land, and also what is being told between the lines of that text. And so it's been session by session, as we have worked to build a community of practice in which we could build a circle of protection to support and nurture each other's growth with the intent to have as an outcome a whole that is greater than the sum of its parts. And it is also to give us a diverse set of lenses with which to see the various social and technical aspects, which themselves create a whole greater than the sum of its parts through mutual shaping. Or so it is within the mindset of a buffalo clan.

Being able to identify the electronics that together become networked information systems provides new insights into the raw data—the symbols used to represent real-world entities—that are eventually converted into digital form to be transmitted or processed using binary. And this seeks to help us to see how the large then reflects these smallest of components. But as the networked interchange of data over the Internet included at its origins and continued to require for an extended time the use of the American Standard Communication Information Interchange (ASCII) codes, characters, and symbols, we also can read more fully the lay of the land and the story being told between the lines. Story data is the ability to identify and interpret data from which information emerges that can be communicated in story. Social justice storytelling means to understand the codification within this data sufficiently to decodify it, in order to better understand the various social and technical aspects that may be contributing to the stock story and to facilitate a sharing of the concealed, resistance, and emerging and transforming stories that respond to the stock stories, highlight the injustices, and reconstruct knowledge built on the counterstories bucking against the stock stories.

As we worked through the Orange Unit and then the Blue Unit, a goal was to increasingly communicate data with context as story in both form and narrative experience. This required an increasing merging of the social and technical aspects to structure story information, the second part of the Story-DIKW framework proposed by Kate McDowell. The Rainbow Unit has sought to further bring a range of different story data and information together as budding story knowledge. As noted within the Story-DIKW framework, this would be the ability to convey knowledge as complex, actionable information through the construction and telling of a story that importantly ensures the incorporation of cultural and contextual cues. The Rainbow Unit thus worked to bring together programmable electronics and the Internet of Things with its counterpart, today's often centralized structures that serve as a primary master device to its many remote slave devices, or what Neil Gerschenfeld calls the Bitnet of Things. The unit has further worked to bring together internetworking and the digital Internet, which includes multiple and often conflicting paths, including those primarily community-led and those primarily corporately led, along with many pathways in between. From here, the unit has worked to explore ways in which a restructuring of social welfare systems has been done in part through definitions of haves and have-nots and through terms such as "digital divide" and "digital inclusion;" this restructuring served as a core aspect of workforce development efforts in United States policies and practices. At the same time we worked to explore the social restructuring, we also worked to discover that in demystifying technology through a deep, hands-on dive into the codification of the sociotechnical Internet, new opportunities open to reflexively decodify it, so as to facilitate knowledge creation and acts of person-centered "good trouble, necessary trouble"[1] in communities, with communities, and for communities of the oppressed as part of our social justice sociotechnical praxis.

We're now coming to the finish as we also reconsider the design aspect of the tinker, adding a design justice framework to the framework of information sciences data, information, knowledge, and wisdom. It is important to see design not as a start point, nor as a finish point, but rather as itself but one tool within a toolbelt. For design to occur justly and lead effectively towards good trouble, necessary trouble, it is essential that the tools of design are used in support of community and movement work that centers critical, person-centered connections, conversations with the people in a room in a moment, the building and preserving of trusting relationships with others, and still other aspects of building and sustaining community. As we've discovered, it is then that design and innovation can occur. And even here, it is not primarily product-focused, but more one of process, in which that which is designed can go on to be further designed through innovation-in-use.[2]

It is exactly here that we find the strength of the Internet Protocols and Request for Proposals processes from which the core of the Internet has arisen, not as solution, but as the starting stage for ongoing creative works to solve the needs of local people and their communities. This is the **End-to-End Principle** first introduced by Louis Pouzin and the French Cyclades Network in the early 1970s. But others, such as the Freedom and Progress Foundation, have championed, and often successfully, for federal policies and corporate practices that would allow one nation, the United States, to use the technological breakthroughs to not only accelerate technological and economic strengths, but also social and political dominance[3]. But the core of the Internet and the potentials for networked information systems remain intact and provide us with the existence of choice. This textbook strives to facilitate the awareness of this choice, along with an introduction into the ways which we can exercise and achieve this choice[4] as part of our work as information professionals, when design and innova-

1. Lewis, John, and Valerie Jackson. "The Boy From Troy: How Dr. King Inspired a Young John Lewis," January 17, 2020. https://storycorps.org/stories/the-boy-from-troy-how-dr-king-inspired-a-young-john-lewis/.
2. Bruce, Bertram C., Andee Rubin, and Junghyun An. "Situated Evaluation of Socio-Technical Systems." In *Handbook of Research on Socio-Technical Design and Social Networking Systems*, edited by Brian Whitworth and Aldo de Moor, 2: 685–98. Information Science Reference. Hershey, PA: IGI Global, 2009. https://doi.org/10.4018/978-1-60566-264-0.ch045; Fischer, Gerhard, and Thomas Herrmann. "Socio-Technical Systems: A Meta-Design Perspective." *International Journal for Sociotechnology and Knowledge Development* 3, no. 1 (2011): 1–33. https://doi.org/10.4018/jskd.2011010101. Also available at https://l3d.colorado.edu/wp-content/uploads/2021/01/Published-JOURNAL-version.pdf
3. Dyson, Esther, George Gilder, George Keyworth, and Alvin Toffler, "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age," The Progress & Freedom Foundation, August 1994. http://www.pff.org/issues-pubs/futureinsights/fi1.2magnacarta.html
4. Kleine, Dorothea. "The Capability Approach and the 'Medium of Choice': Steps towards Conceptualising Information and Communication Technologies for Development." *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 119–30. https://doi.org/10.1007/s10676-010-9251-5.

tion become part of community and movement work to address the needs and opportunities which the member of these communities and movements have identified.

Reflecting Back on the Learning Objectives of this Journey

Let's take a moment to reflect back on a few key highlights from the Introduction to the Book. Let's start with the general learning outcome objectives of this book:

- Develop a clear, hands-on working understanding of the physical and software layers of computers and networks. As learners journey through the units of this book, they will hopefully develop a growing comfort and competency: working with the basic nuts and bolts of computers and networks; appropriately integrating components to serve as tools for computational and information processing; and performing basic troubleshooting.
- Evolve a more holistic and nuanced understanding of the sociotechnical artifacts we use as a daily part of our professional lives. The hardware, software, human, and social whole that is a digital artifact is greater than the sum of the parts. Beyond developing technical competencies, we need to develop an awareness of, and skillsets to influence, the emergent properties that come from specific combinations of the different social and technical building blocks of information systems.
- Develop a critical approach to sociotechnical artifacts. Social systems are constructs of economy, politics, race, class, gender, social institutions, and other cultural dynamics. The design, diffusion, and implementation of technical innovations both reflect and shape these social systems. Critically examining social and technical information systems from multiple individual and societal perspectives opens up consideration of idealized expectations versus actual positive and negative impacts within specific user communities.
- Advance community agency in appropriating technology to achieve our individual and community development goals through a reconsidered digital literacy learning and practice. Far from being just passive adopters of different digital technology artifacts used to find, evaluate, create, and communicate information, as Information Science professionals, we have opportunities to initiate and lead communities of practice, leveraging the plurality of our community's social and technical insights.

To achieve these outcome objectives, this is a book meant to be done as part of a community of practice studio. The different units of the book and the sessions within each unit work within a virtuous cycle that includes asking initial questions, joining in active co-learning, reading the words and worlds of knowledge shared in a range of ways, internal reflection, group discussion, critical questions, and journaling. Collective leadership, community inquiry, and action and reflection directed at the structures to be transformed have been some of the key frameworks used within the textbook, and hopefully within your community of practice studio, to collectively work within the virtuous cycle. Core to all of this is the call of the Rev. Dr. Martin Luther King, Jr.:

> We must rapidly begin the shift from a 'thing-oriented' society to a 'person-oriented' society. When machines and computers, profit motives and property rights are considered more important than people, the giant triplets of racism, materialism, and militarism are incapable of being conquered[5].

To help us advance our full range of skills—social and technical—needed to achieve our works as craftspeople, each session in this book includes two thematically-linked chapters, one more social-oriented and one more technical-oriented. Each of these thematically-linked chapters, along with the various activities, group discussions, and your Reading Notes and Professional Journal Reflections, form one session. Sessions are brought together into several larger units:

- **Orange Unit** with social chapters providing underlying perspectives and mindsets and technical chapters introducing electronics.
- **Blue Unit** with four social and technical sessions challenging us to exploring histories, mutual shaping, and innovations-in-use through a clearer grasp of the meta-theoretical landscape found within Western science and within Indigenous ways of knowing, while bringing together electronics and programming code within microcomputer and microcontroller systems.
- **Rainbow Unit** with four sociotechnical sessions seeking to tighten our growing holistic and nuanced understanding of the tools and technologies we use as a daily part of our professional, community, and personal lives.

Time to Reflect Back on Your Actual Journey

Before moving on, take some extended time now to look through your packet of Reading Notes and Professional Journal Reflections. Work to write down an introductory section of a Summative Professional Journal Reflection as you consider your journey as part of this community of practice.

- How have you changed and evolved as a tinker?
  - As a member of a community of practice?
  - As a social justice storyteller?
  - As an information scientist?

5. King, Martin Luther. "Beyond Vietnam." In *A Call to Conscience*, edited by : Carson, Clayborne and Kris Shepard, eds. 139-164. Grand Central Publishing, 2001.

Emergent Strategy and its Grounding of Design Justice

In the previous chapter, we were introduced to a range of design principles and practices. Design plays an active role in the mutual shaping of sociotechnical products. But it is not the starting or ending point of sociotechnical artifacts. Rather, a range of individual, community, social, and/or technical contexts first come to play in launching a design activity, as possibilities for addressing identified needs and opportunities are explored. And while design may include rapid prototyping, these are generally insufficient to address the underlying individual, community, social, and/or technical contexts which first inspired the design activity. Design is necessary, but rarely if ever sufficient to addressing critical issues impacting people, nature, and society. As Sasha Costanza-Chock notes in her chapter, "Design Practices: 'Nothing about Us without Us'":

> Ultimately, at its best, a design justice process is a form of community organizing. Design justice practitioners, like community organizers, approach the question of who gets to speak for the community from a community asset perspective. This is rooted in the principle that wherever people face challenges, they are always already working to deal with those challenges; wherever a community is oppressed, they are always already developing strategies to resist oppression. This principle underpins what Black feminist author adrienne maree brown calls *emergent strategy*. Emergent strategy grounds design justice practitioners' commitment to work with community-based organizations that are led by, and have strong accountability mechanisms to, people from marginalized communities. This contrasts with most other design approaches; even those that aim to involve users, citizens, or community members typically do so in a consultative process that ultimately is led by the professional designers.[6]

The concept of emergence has been used to understand how many relatively simple interactions become the connective tissue out of which complex systems and patterns arise. The critical connections are thus emphasized over critical mass as strategies are brought forward to build authentic relationships using all the senses of the body and the mind. This is something that is not just a human-centered and -controlled process, but also includes more-than-human individuals and ecosystems as well. **Emergent strategy**, then, is a way of noticing the small actions and connections that create complex systems and patterns. As adrienne maree brown notes on pages 16-17 (Kindle Edition) of her book *Emergent Strategy*:

> Nothing is wasted, or a failure. Emergence is a system that makes use of everything in the iterative process. It's all data.

6. Costanza-Chock, Shasha. *Design Justice: Community-Led Practices to Build the Worlds We Need*, 92-93.

Octavia Butler	**(adrienne maree brown)**
All successful life is	(Fractal)
Adaptable,	(Adaptive)
Opportunistic,	(Nonlinear/Iterative)
Tenacious,	(Resilient/Transformative Justice)
Interconnected, and	(Interdependent/Decentralized)
Fecund.	(Creates More Possibilities)
Understand this.	(Scholarship, Reflection)
Use it.	(Practice/Experiment)
Shape God.	(Intention)

> brown, adrienne maree. Emergent Strategy (pp. 16-17). AK Press. Kindle Edition. Edited to replace (amb) with (adrienne maree brown) for clarity in this context.

There are a number of revisions made in the second edition of this textbook that have at their root the principles and practices of emergent strategy to which I was introduced soon after the publication of the first edition. It serves as a complement to the Western sciences, which is in part why it has informed the creation of the design justice principles, which themselves are rooted in large part within Western framings of design. In so doing, brown brings a Black feminist voice into the conversation, similar to the way Robin Wall-Kimmerer brings an Indigenous voice into the conversation, as Kimmerer brings together her Western science and her evolving understanding of her forgotten roots found in Indigenous ways of knowing together. And I believe emergent strategy resonates so strongly with me personally. It brings together my various cultural, environmental, ecumenical, and Western science foundations, some of which I've shared throughout this book, into a more unified understanding of what I've taught, planned, written, and practiced throughout my career as a critically engaged, community-engaged scholar with a special focus on using the discipline of community informatics. Each of the principles of emergent strategy captures well how I seek to live, think, and work to build a better world. And each is something I hope will also prove fruitful for others as well:

> Small is good, small is all. (The large is a reflection of the small.)
> Change is constant. (Be like water).
> There is always enough time for the right work.
> There is a conversation in the room that only these people at this moment can have. Find it.
> Never a failure, always a lesson.
> Trust the People. (If you trust the people, they become trustworthy).
> Move at the speed of trust.
> Focus on critical connections more than critical mass—build the resilience by building the relationships.

> Less prep, more presence.
> What you pay attention to grows.[7]

After a long day of activity with others outdoors, whether canoeing, sailing, backpacking, or cross-country skiing, we try to ensure time at the campfire to quietly reflect on the day while watching the beauty of the flames constantly changing shape and form. Most nights, this leads to someone sharing something beautiful they saw or experienced over the course of the day—a rose moment. Or maybe it was an experience that hurt them or made them really struggle in some way—a thorn moment. It is to acknowledge that the beauty of roses also includes the risk of a prick by the thorn along their stems. As these experiences start coming forward, we also strive to include the hopes we have for tomorrow—our bud moments.

After multiple days, we try to set aside a down day to also just sit quietly within nature for extended time. Finding ways to tune into one small part of the whole of nature around me helps connect to the small that exists within, and makes up one part of, the complex ecosystem within which we find ourselves to have been navigating. As I sit in quiet contemplation, I see the constant of change within the longer arc that is change across centuries and millennia. I do so by finding the opportunities for conversation with nature. But I also then find it helpful to eventually move from hearing the music coming from the one to hear the choir that results from the voices of the many within the ecosystem in which I find myself.

Listening to the Parts, Listening to the Choir

Take time now to reflect back on your journey and write down a second session of your Summative Professional Journal Reflection, beginning with those from sessions three and four of the Rainbow Unit. What are some of the rose, thorn, and bud moments you've experienced as we've worked to unpack ways that digital technologies have been designed and deployed in ways that have continued or even expanded the web of oppressions found within the social divides around us?

Reflecting further back to the start of the Rainbow Unit, we worked to better understand the infrastructure of the Internet and the possibilities and challenges in broadening our application of internetworking into other realms than just the globalized and corporatized aspects. This was a long journey, but hopefully there were various rose and bud moments amongst the thorns as we traveled through the Rainbow Unit. In what ways did we succeed in demystifying the clouds that keep network information systems within a mist? In what ways did we open up a visioning of the revolution of values that are person-oriented, rather than thing-oriented, as King called us to do over forty years ago?

7. brown, adrienne maree. *Emergent Strategy* (pp. 44-45). AK Press. Kindle Edition.

Now look through the whole of your notes, reflections, discussions, and hands-on activities. As you stand nearing the end of this journey, what are your special rose moments that you will keep preserved as a keepsake? What thorns especially pricked you along this journey or are still pricking you now? What are some buds you see as potentially ready to bloom in the coming months?

Planning Forward

Let's return now to the Person-Centered Network Information System Adventure started in session three of the Rainbow Unit as your pre-season training comes to an end. The chapter ended with a Wrap Up and Comprehension Check. In what ways has your journey sufficiently advanced your sociotechnical skills in ways that prepare each member of our community of inquiry for a deeper, critically engaged community inquiry during the upcoming season of your life? Such adventures will include in different ways at different times active bottom-up participatory action research in community, with community, and for community throughout your lifetime. We've worked to advance muscle memory through ongoing pre-season, action-reflection, collective leadership practices, so that these methods are an active part of your daily practices as an information science professional moving forward. To this we could now add design skills centered within principles of design justice and emergent strategy.

It is time to join together in small groups to find that conversation that can only happen in that room with the people there in this moment. How can you build off the work to date to imagine the work to come next?

- Nimbleness in action is essential, but can easily lead to the embracing of a "do-acracy" that comes with significant risk of unintended consequences and loss of a sustained and truly systemic impact. True, direct democracy, on the other hand, requires times and places to slow it down if we are to expose issues of power, inclusion, and equity. How is the balance between action and reflection within your community of practice at this time? In what ways, if any, do you need to work in some way to address the ever-present tension between process and product?
- Over the course of the textbook, we've worked to first understand the many small components and structures—such as electronic circuits, microcontrollers and microcomputers, basic variables, statements, arrays, conditionals and logic—that make up programming code, and that base standards and structures that are used to do internetworking. From there, we've worked to understand the ways these come together to form more complex structures. While change is constant, understanding these underlying small aspects, which have remained more stable even as complex systems have changed, helps us to demystify these black boxes. To this, we added a range of social components, which together created a more complex social web within which the complex technical structures reside. Together

there exists a mutual shaping of sociotechnical artifacts and systems. In what ways, if any, might you bring this to your future work as an information professional? How can you work to bring others into the more demystified and person-centered approach in the next community of inquiry in which you find yourself, within a thing-oriented society that uses the black box to grow people's dependency on corporations and their interests?

- Women and those who are Black, Indigenous, or within other groups of people of color are often excluded from fail-forward and growth mindsets as they are instead called on to be perfect. Instead of seeing failure as something that is wrong, we need to see it as a necessity and as part of learning. Hearing from the voices of others, and especially the marginalized and oppressed, leads to critical connections, moving the data and information towards that which is actionable—towards knowledge and power within and with those around us to advance good trouble, necessary trouble. This requires trust in the people, and it requires us to move at the speed of trust. Together we have a more holistic lens with which we can see those things that are not right, not fair, not just. With this more holistic lens, we can also stand up, say something, do something. This requires collective leadership as we work to proceed from failure towards lessons that help us ask the better question. It contributes to a virtuous cycle of investigation, design, and creative actions combined with critical conversations that can only happen in that moment with those people. It requires us to be present in that space using a person-orientation, a being-mode of action-reflection praxis, rather than being caught up within a doing-mode in which we are locked within a thing-orientation. In what ways, if any, might you bring this to your future work as an information professional? How can this help your community of inquiry get into the good trouble, necessary trouble to build a better world within the space you find yourselves in that moment?

Diving Deeper

For there to be design justice, it is important to consider all of the ways possible to bring diverse community knowledge and cultural wealth to bear in the process. Indeed, as we've seen throughout the textbook, innovative design and redesign of technologies so often forces into hiding the expansive, diverse works of design and creations of technology done by each human. As Costanza-Chock notes:

> Design justice as a framework recognizes the universality of design as a human activity. As noted in the introduction, design means to make a mark, make a plan, or problem-solve; all human beings thus participate in design. However, though all humans design, not everyone gets paid to do so. Intersectional inequality systematically structures paid professional design work. Professional design jobs in nearly all fields are disproportionately allocated to people who occupy highly privileged locations within the matrix of domination. At the same time, the numerous expert designers and technologists who are not wealthy and/or educa-

tionally privileged white cis men have often been ignored, their labor appropriated, and their stories erased from the history of technology.[8]

As information science professionals, we are paid not to just passively and strictly follow a list of tasks verbatim, but to creatively innovate-in-use our designated activities in ways that meet the needs of our patrons. If we are to truly advance existing and emerging social justice parameters within our profession, we first need to move from user-centered design to human-centered design as part of our daily practices. But from here, we need to also move from general design thinking to an essential design justice framing. Funding to pay community members as members of the design team is an essential feature to advance social justice objectives, but one too often difficult to carry out in practice. In addition, many designers have other commitments to work, community, and family that keep them from daily participation. I've found at times that an effective middle ground is to embed collective leadership within this process through invitation (with provision of funds, transport, and food to every extent possible) of community members who bring essential community knowledge and cultural wealth to bear within the critique process itself.

Finding the Conversation That Can Only Happen Now with the People in this Room

What are some things you should put into your action plan for the days, weeks, and months to come, as a community of practice, as a learning cohort within your degree program or community workshop space, and as individuals? Return again to the comprehension check you developed for yourselves at the end of session three. Take some time now to add to this some specifics related to design. What are the specific design-centered comprehension checks you are seeking to ensure that you are reaching the "yet enough" point of introductory development?

- Often, we make use of activities like think-pair-share or two-four-all to do creative works.
- This time you might consider flipping the script. Perhaps start within larger groups to do some initial brainstorming, using a whiteboard to map a broad action plan. From there, begin considering what smaller cohorts should gather to work on specific aspects especially relevant to their own development as professionals moving forward.
- Whichever path you choose, a final Summative Professional Journal Reflection will be important, to bring the various thoughts that have emerged over the course of the textbook and as part of this last session of the Rainbow Unit into conversation.

8. Costanza-Chock, Sasha. *Design Justice: Community-Led Practices to Build the Worlds We Need*, Information Policy (Cambridge, MA: The MIT Press, 2020), 73. https://designjustice.mitpress.mit.edu/.

However the ideas that form from this action plan brainstorming activity get jotted down, be sure to find a way to copy and paste them into your Summative Professional Journal Reflection as a third component.

Wrap Up

Many technical instructions leave unconsidered the embedded social dimensions of technologies that are continuously in tension with the limits and constraints to social justice parameters. Unconsidered, we likely further algorithms of oppression rooted within deductive logic and positivist paradigms in our personal and professional practices. This is also true of design processes, and is why design justice rooted within critical paradigms has been emphasized so strongly throughout session four. And as noted in the session three Wrap Up, if you currently do not have rich diversity within your community of practice, work to outline clearly how diversity would be brought into the collective leadership to ensure that this is the “good trouble” to which the late John Lewis refers.

Put aside your Summative Professional Journal Reflection for a few days, but keep it relatively close. After a short break, pick it up one last time to read it through. Don’t make any edits—that’s not what journaling is about. But do feel encouraged to add a final section in which you note some reflections on your reflections—that’s the work of recodifying that which you’ve worked to decodify. For only through cycles of action and reflection can the work of community inquiry take on the various situations limiting people from being fully human through good trouble, necessary trouble, overcoming these in a strategic way.

Rainbow Unit Review

[The following is an interactive element from the online edition that has been remediated into a different form for the PDF. For an interactive version with the option to export your responses, use the online version.]

Key Learning Outcomes Self-Check

Listed below are the key learning outcome objectives for the various skill sets we sought to advance in the **Rainbow Unit** of the book. Using the scale below, take a moment to create a second ***1-4 rating*** of these items based on where you think you are in your skills development now that you've completed a pass through the Rainbow Unit cycle. Add additional notes as helpful to keep further specific reflections on your development of these specific skills.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough
4: Yet

Technical Skills

You can (rate 1-4):

- Understand the seven layers of the Open System Interconnection (OSI) model and their practical uses in networking.
- Understand Local Area Network components, and how to design, troubleshoot, and maintain those components.
- Understand Wide Area Networks like the Internet, how to select between Internet Service Providers, and how to join the Internet by connecting to an Internet Service Provider.

- Understand the Internet Protocol, IP addresses, IP names, and how they apply to nodes on your LAN.
- Understand the peer-to-peer and client/server architecture and open protocols at a basic level.
- Understand Infrastructure-, Platform-, and Software-as-a-Service as an operational definition of cloud computing.
- Distinguish between locally controlled Internet devices using open hardware, software, data, and/or protocols, and centrally controlled Internet devices using closed hardware, software, data, and/or protocols.
- Use multiple methods to setup a Hypertext Transport Protocol (HTTP) server.
- Use serial communication to collect, analyze, store, and share data from microcontrollers and microcomputers, and make them accessible across a LAN and from the Internet.

Cognitive Skills

You can (rate 1-4):

- Generate a wide variety of ways to accomplish project and logically analyze divergent pathways to assess potential impacts.
- Logically analyze and organize project tasks in ways that allow use of digital tools to help accomplish them.
- Apply different critique evaluation methods periodically throughout the design process to advance effective human-centered design.
- Turn personal and community strengths, opportunities, and aspirations into project ideas.

Socio-Emotional Skills

You can (rate 1-4):

- Empathize with individuals and social groups across a wide spectrum of social and cultural contexts.
- Progress towards implementation of stages four, five, and six within the Information Search Process as applied to a design.

Critical Sociotechnical Skills

You can (rate 1-4):

- Recognize the historical development of the Internet within the United States and internationally, its further socio-cultural, -economic, and -political development through commissions, associations, think tanks, non- and for-profit groups, and the influences this has had on the design of its sociotechnical artifacts and systems.

- Integrate a basic technical understanding of Internet technologies with a critical sociotechnical perspective to advocate for more just Internet policies and implementations.
- Critically, strategically, and communally use a design thinking mindset to unpack ill-defined problems using critical abductive reasoning within a broader design justice approach.

Your Personal Learning Outcome Objectives

Add the personal learning outcome objectives you created at the beginning of this Unit.

Rate Your Personal Learning Outcome Objectives

Take a moment to create a second ***1-3 rating*** of these items based on where you think you are in your skills development now that you've completed a pass through the Rainbow Unit cycle. Add additional notes as helpful to keep further specific reflections on these specific skills development.

Possible ratings:

1: Unknown
2: Not Yet
3: Yet Enough

Goals	Rating
Your first goal...	

Introducing the Unix Command Line

Dinesh Rathi; Henry Grob; Vandana Singh; and Martin Wolske

Jump to a section:

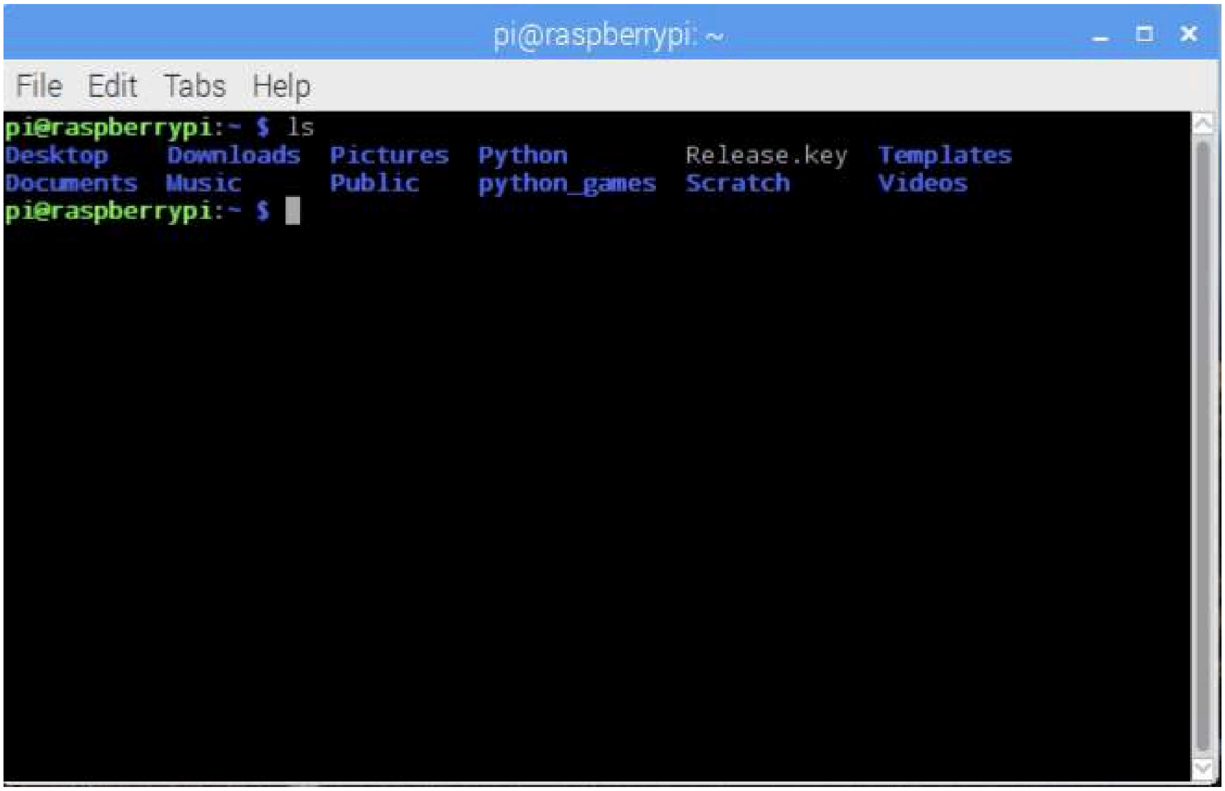
Most of the time, our daily work on computers is done through a graphical user interface (GUI) in which we do some typing of text in combination with the clicking of drop-down menus and option buttons. Those drop-downs and option buttons hide the wide range of text that is actually executed through those mostly simpler clicks we make. But sometimes we need greater control than is provided through those menus and buttons, which have been shaped in part to simplify at the cost of reducing flexibility. And sometimes we need to just quickly hop into a computer to do something and then pop right back out. In these cases, and especially when done remotely from another computer, the terminal interface is a much faster alternative. For this reason, Unix and Unix-based Operating Systems like MacOS and Linux have always included at their core a **Command Line Interface (CLI)** such as a terminal window. While for most of its existence Microsoft pushed people to ignore their command prompt alternative and to instead strictly work within the graphical user interface, beginning with Windows 7, Microsoft has included their PowerShell command-line shell and scripting language.

Listing and Creating Files and Directories

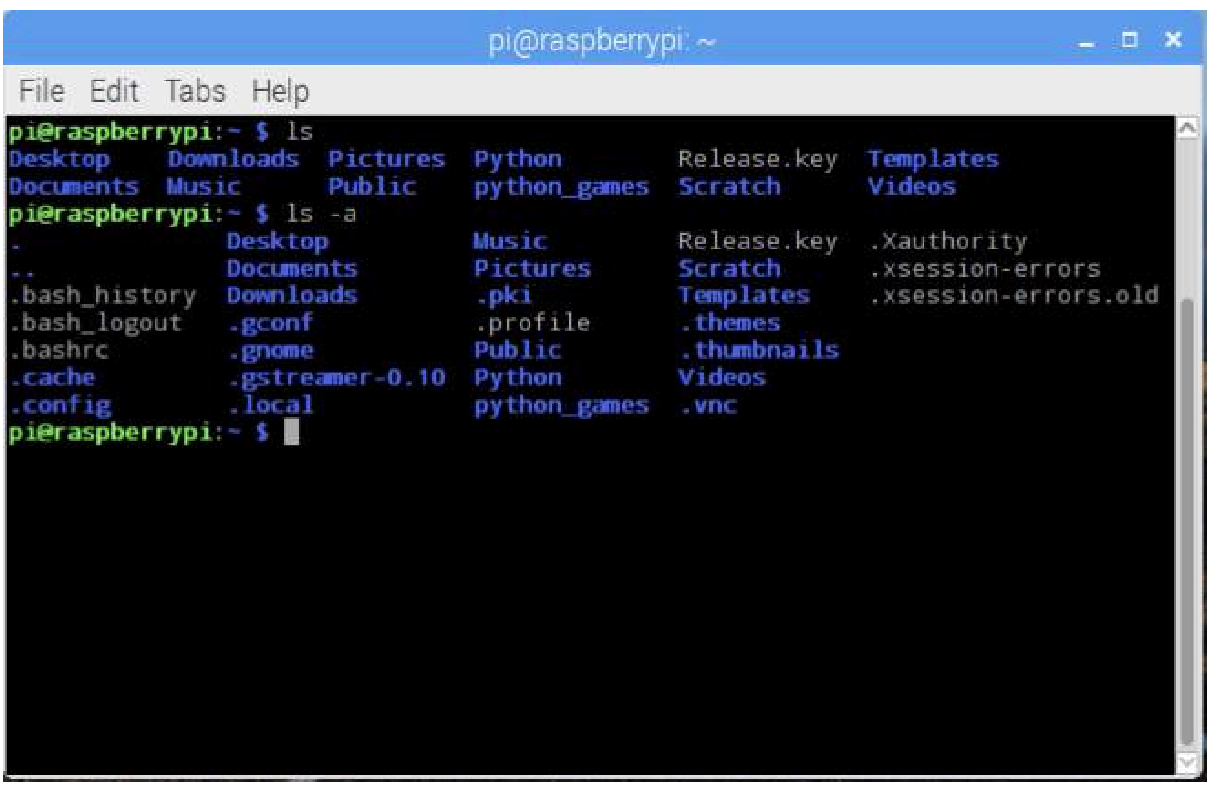
In Linux and Unix Operating Systems, commands start with the command itself, for instance:

```
ls
```

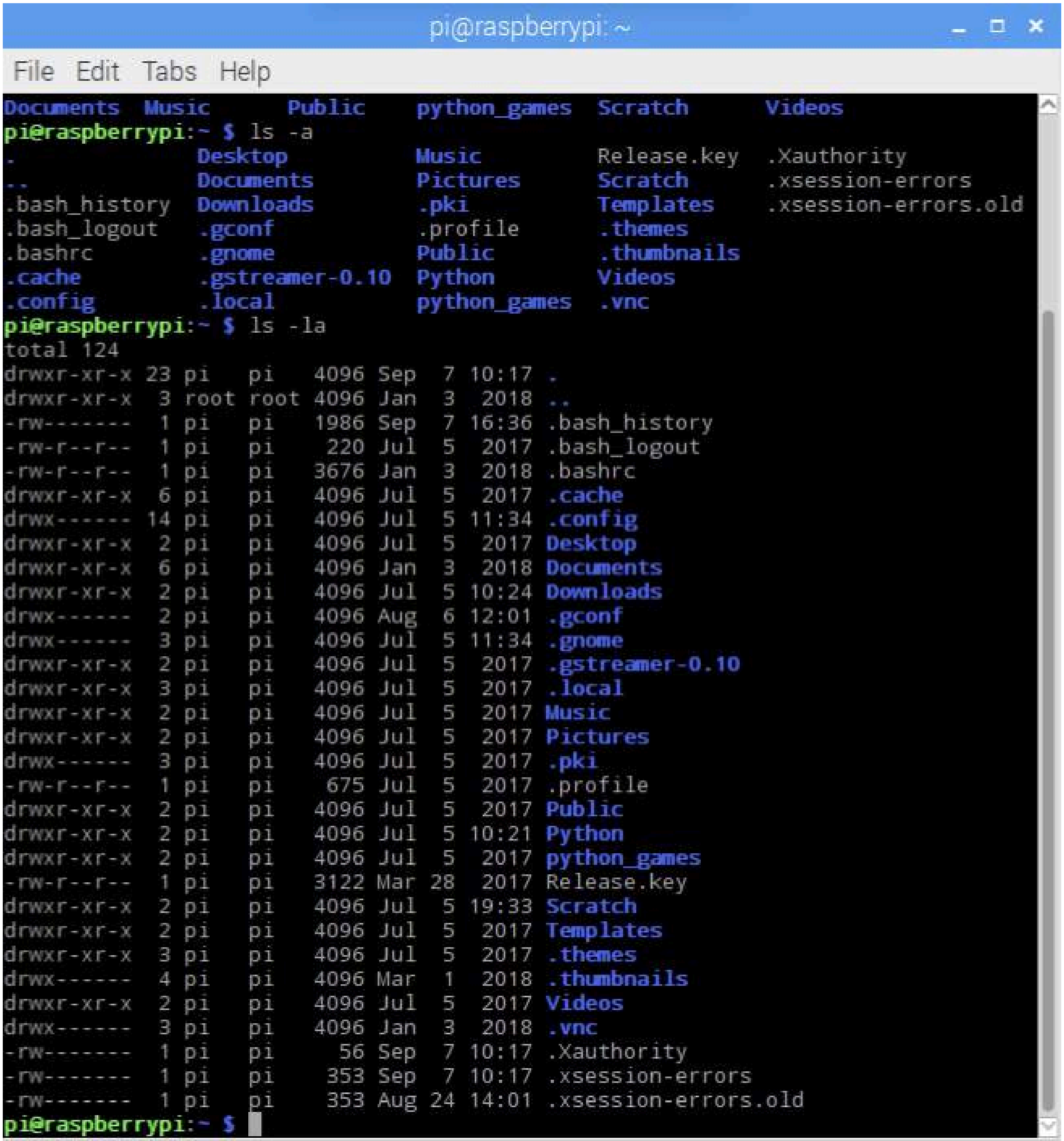
If you type in the `ls` command, you'll get a list of files, folders, and directories.



Many commands also have options, switches, or flags. For instance, the `ls` command has many options you can use, including the `-a` to list all files, including those that start with a period to keep them a little more hidden.



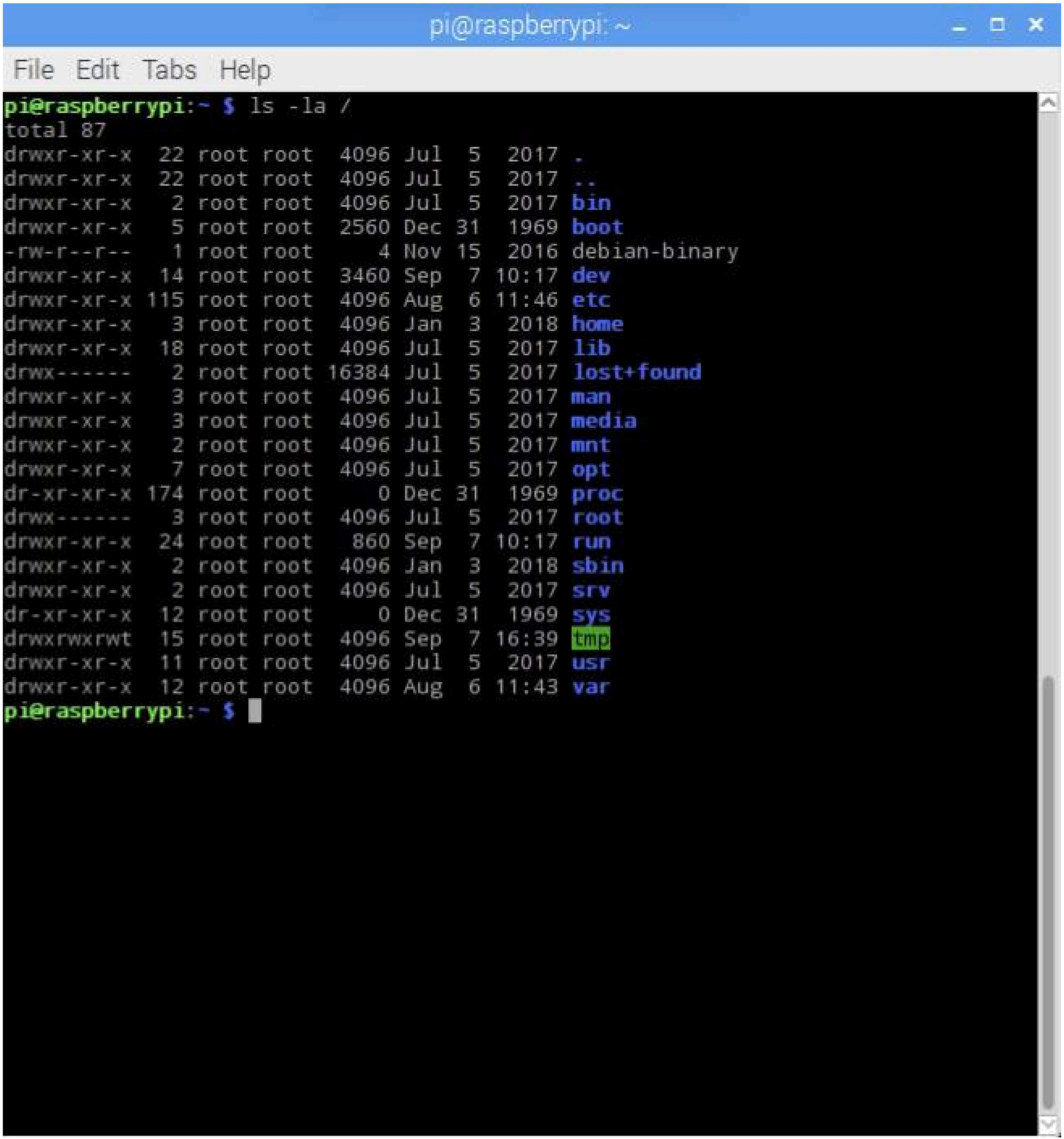
As another example, if we add not just a `-a` but also a `-l` to an `ls`, we can have all files listed in a long format using option grouping.



While `ls` is the basic command, and `ls -la` is the command with options, we can also add in arguments at the end, for instance, to get a listing of files, folders, and directories within a specific directory. As an example:

```
ls -la /
```

This command would give us the list of files, folders, and directories within the very top of the tree structure.



To create an empty or blank file, use the `touch` command, followed by the name and extension of the new file:

```
touch mynewfile.txt
```

To create a new directory, use the `mkdir` command, followed by the name of the new directory you would like to create:

```
mkdir cstories
```

Copying Files

To move a file from one place to another, use the `mv` command. This has the effect of moving rather than copying the file, so you end up with only one file rather than two. It can also be used to rename a file by moving the file to the same directory but giving it a different name.

```
mv file1 file2
```

This command moves (or renames) file1 to file2.

Remove Files and Directories

To delete (remove) a file, use the `rm` command. As an example, we are going to create a file and then delete it. Type the following:

```
$ touch tempfile.txt
$ ls
$ rm tempfile.txt
$ ls
```

The `ls` command is used first to check if it has created the file and second to check if it has deleted the file.

You can use the `rmdir` command to remove a directory (make sure it is empty first). Note that Unix will not let you remove a non-empty directory.

Create a directory called tempstuff using `mkdir`, then remove it using the `rmdir` command.

Searching the Contents of a File

Using `less`, you can search though a text file for a keyword (pattern). For example, to search through a file called science.txt for the word science, type:

```
$ less science.txt
```

Then, while still running `less` (i.e., don't press [q] to quit), type a forward slash [/] followed by the word to search:

```
/science
```

As you can see, less finds and highlights the keyword. Type [n] to search for the next occurrence of the word.

Another powerful search tool is `grep`, one of many standard Unix utilities. It searches files for specified words or patterns. First clear the screen, then type:

```
$ grep science science.txt
```

As you can see, grep has printed out each line containing the word science. Or has it? Try typing:

```
$ grep Science science.txt
```

The grep command is case sensitive; it distinguishes between Science and science.

To ignore upper/lower case distinctions, use the `-i` flag:

```
$ grep -i science science.txt
```

To search for a phrase or pattern, you must enclose it in single quotes (the apostrophe symbol). For example, to search for spinning top, type:

```
$ grep -i 'spinning top' science.txt
```

Some of the other options of grep are:

- `-v` display those lines that do NOT match.
- `-n` precede each matching line with the line number.
- `-c` print only the total count of matched lines.

Try some of them and see the different results. Don't forget, you can use more than one option at a time, for example, the command to find the number of lines without the words science or Science is:

```
$ grep -ivc science science.txt
```

Redirection

Most processes initiated by Unix commands write to the standard output (that is, they write to the terminal screen). Many take their input from the standard input (that is, they read it from the keyboard).

There is also the standard error, where processes write their error messages, by default, to the terminal screen. We have already seen one use of the `cat` command to write the contents of a file to the screen. Now type `cat` without specifing a file to read:

```
$ cat
```

Then type a few words on the keyboard and press the [Return] key. Finally hold the [Ctrl] key down and press [d] (written as ^D for short) to end the input. What has happened?

If you run the cat command without specifing a file to read, it reads the standard input (the keyboard), and on receiving the'end of file' (^D), copies it to the standard output (the screen). In Unix, we can redirect both the input and the output of commands.

Redirecting the Output

We use the > symbol to redirect the output of a command. For example, to create a file called list1, containing a list of fruit, type:

```
$ cat > list1
```

Then type in the names of some fruit. Press [Return] after each one.

pear
banana
apple
^D (Control D to stop)

What happens is the `cat` command reads the standard input (the keyboard) and the > redirects the output, which normally goes to the screen, into a file called list1.

To read the contents of the file, type:

```
$ cat list1
```

Using the above method, create another file called list2.

Add the following fruit to the file: orange, plum, mango, grapefruit.

The form >> appends additional output to a file. So to add more items to the file list1, type:

```
$ cat >> list1
```

Then type in the names of more fruit:

peach
grape
orange
^D (Control D to stop)

To read the contents of the file, type:

```
$ cat list1
```

You should now have two files. One contains six fruit, the other contains four fruit. We will now use the `cat` command to join (concatenate) list1 and list2 into a new file called biglist. Type:

```
$ cat list1 list2 > biglist
```

What this is doing is reading the contents of list1 and list2 in turn, then outputting the text to the file biglist.

To read the contents of the new file, type:

```
$ cat biglist
```

Redirecting the Input

We use the < symbol to redirect the input of a command. The command sort alphabetically or numerically sorts a list. Type:

```
$ sort
```

Then type in the names of some vegetables. Press [Return] after each one.

carrot
beetroot
artichoke
^D (control d to stop)

The output will be:
artichoke
beetroot
carrot

Using < you can redirect the input to come from a file rather than the keyboard. For example, to sort the list of fruit, type:

```
$ sort < biglist
```

And the sorted list will be output to the screen.

To output the sorted list to a file, type:

```
$ sort < biglist > slist
```

Use `cat` to read the contents of the new sorted file.

Pipes

To see who is on the system with you, type:

```
$ who
```

One method to get a sorted list of names is to type:

```
$ who > names.txt
$ sort < names.txt
```

This is a bit slow and you have to remember to remove the temporary intermediary file called names when you have finished. What you really want to do is connect the output of the who command directly to the input of the sort command. This is exactly what pipes do. The symbol for a pipe is the vertical bar |.

For example, type in:

```
$ who | sort
```

This will give the same result as above, but quicker and cleaner. To find out how many users are logged on, type:

```
$ who | wc -l
```

Wildcards

The character * is called a wildcard, and will match against none or more character(s) in a file (or directory) name. For example, in your current directory, type:

```
$ ls list*
```

This will list all files in the current directory starting with list. Try typing:

```
$ ls *list
```

This will list all files in the current directory ending with list.

The character ? will match exactly one character. So the command `ls ?ouse` will match files like 'house' and 'mouse'. But it will not match 'grouse'.

Try typing:

```
$ ls ?list
```

Filename Conventions

A directory is merely a special type of file. So the rules and conventions for naming files apply also to directories. In naming files, characters with special meanings such as these should be avoided:

```
/ * & %
```

Also, avoid using spaces within names. The safest way to name a file is to use only alphanumeric characters, that is, letters and numbers, together with _ (underscore) and . (dot).

File names conventionally start with a lower-case letter and may end with a dot followed by the extension, a group of letters indicating the contents of the file. For example, all files consisting of C code may be named with the ending .c, for example, program1.c. Then, in order to list all files containing C code in your home directory, you need only type `ls *.c` in that directory.

Beware: some applications give the same name to all the output files they generate. For example, some compilers, unless given the appropriate option, produce compiled files named a.out. Should you forget to use that option, you are advised to rename the compiled file immediately, otherwise the next such file will overwrite it and the original file will be lost.

Getting Help

There are online manuals which give information about most commands. The manual pages tell you which options a particular command can take, and how each option modifies the behavior of the command. Type the `man` command to read the manual page for a particular command.

For example, to find out more about the wc (word count) command, type:

```
$ man wc
```

Alternatively, the following command gives a one-line description of the command, but omits any information about options.

```
$ whatis wc
```

When you are not sure of the exact name of a command, the apropos command will give you the commands with keyword in their manual page header.

```
$ apropos keyword
```

For example, try typing:

```
$ apropos copy
```

File System Security

In your current directory, type:

```
$ ls -l
```

The letter L stands for long listing. You will see that you now get lots of details about the contents of your directory, similar to the example below.

Also, the command `ls -lg` gives additional information as to which group owns the file (beng95 in this example):

```
-rwxrw-r-- 1 ee51ab beng95 2450 Sept29 11:52 file1
```

Each file (and directory) has associated access rights. In the left-hand column is a 10 symbol string consisting of the symbols d, r, w, x, -, and occasionally s or S.

The first character in the line of output specifies the type: A d indicates a directory. Otherwise, the character will be a dash (-). The 9 remaining symbols indicate the permissions, or access rights, and are taken as three groups of 3.

- The first character, -, notes that this is a file.
- Next, the first group of three characters gives the file permissions for the user that owns the file (or directory). In the above example, `rwx` are the permissions for the user `ee51ab`.
- The middle group of three characters gives the permissions for the group of users to whom the file (or directory) belongs. In the above example, `rw-` are the permissions for the group `eebeng95`.
- The rightmost group of three characters gives the permissions for all others. In the above

example, `r--` are the permissions for any user who does not own the file or is not in the group to whom the file belongs.

The symbols r, w, and – have slightly different meanings depending on whether they refer to a simple file or to a directory.

Access rights on files are:

- r (or -), indicates read permission (or otherwise), that is, the presence or absence of permission to read and copy the file.
- w (or -), indicates write permission (or otherwise), that is, the permission (or otherwise) to change a file.
- x (or -), indicates execution permission (or otherwise), that is, the permission to execute a file, where appropriate.

Access rights on directories are:

- r allows users to list files in the directory.
- w means that users may delete files from the directory or move files into it.
- x means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files. So, in order to read a file, you must have execute permission on the directory containing that file, and hence on any directory containing that directory as a subdirectory, and so on, up the tree.

Some examples:

A file that everyone can read, write, and execute (and delete):

```
-rwxrwxrwx
```

A file that only the owner can read and write. No one else can read or write and no one has execution rights (e.g., your mailbox file):

```
-rw-------
```

Changing access rights

To "change a file mode," meaning to change the access rights of a file, use `chmod`.

Only the owner of a file can use chmod to change the permissions of a file. The options of chmod are as follows:

Symbol	Meaning
u	User
g	Group
o	Other
a	All
r	Read
w	Write (and delete)
x	Execute (and access directory)
+	Add permission
–	Take away permission

For example, to remove read, write, and execute permissions on the file biglist for the group and others, type:

```
$ chmod go-rwx biglist
```

This will leave the other permissions unaffected. To give read and write permissions on the file biglist to all, run:

```
$ chmod a+rw biglist
```

Processes and Jobs

A process is an executing program identified by a unique PID (process identifier). To see information about your processes, with their associated PID and status, type:

```
$ ps
```

A process may be in the foreground, in the background, or be suspended. In general, the shell does not return the Unix prompt until the current process has finished executing. Some processes take a long time to run and hold up the terminal. Backgrounding a long process has the effect that the Unix prompt is returned immediately, and other tasks can be carried out while the original process continues executing.

For example, the command `sleep` waits a given number of seconds before continuing. To wait 10 seconds, type:

```
$ sleep 10
```

After 10 seconds, it will return to the command prompt. Until the command prompt is returned, you can do nothing except wait. To background a process, type an ampersand (&) at the end of the command line. To run sleep in the background, type:

```
$ sleep 10 &
```

The & runs the job in the background and returns the prompt straight away, allowing you to run other programs while waiting for that one to finish. The first line in the above example is typed in by the user; the next line, indicating job number and PID, is returned by the machine. The user is notified of a job number (numbered from 1) enclosed in square brackets, together with a PID, and is notified when a background process is finished. Backgrounding is useful for jobs which will take a long time to complete.

To background a current foreground process, type at the prompt:

```
$ sleep 100
```

You can suspend the process running in the foreground by holding down the [control] key and typing [z] (written as ^Z) Then to put it in the background, type:

```
$ bg
```

Note: do not background programs that require user interaction; e.g., pine.

Listing suspended and background processes

When a process is running, backgrounded or suspended, it will be entered onto a list along with a job number. To examine this list, type:

```
$ jobs
```

An example of a job list could be:

- [1] Suspended sleep 100
- [2] Running netscape
- [3] Running nedit

To restart (foreground) a suspended process, type:

```
$ fg %jobnumber
```

For example, to restart sleep 100, type:

```
$ fg %1
```

Typing `fg` with no job number foregrounds the last suspended process.

Killing a process

It is sometimes necessary to kill (terminate or signal) a process, for example, when an executing program is in an infinite loop. To kill a job running in the foreground, type ^C (control c). For example, run:

```
$ sleep 100
^C
```

To kill a suspended or background process, type:

```
$ kill %jobnumber
```

For example, run the following, and suppose that by running jobs, you find that sleep is job number 4:

```
$ sleep 100 &
$ jobs
$ kill %4
```

To check whether this has worked, examine the job list again to see if the process has been removed. Alternatively, processes can be killed by finding their process numbers (PIDs) and using kill PID_number. The command `ps` stands for process status.

```
$ sleep 100 &
$ ps
PID TT S TIME COMMAND
20077 pts/5 S 0:05 sleep 100
21563 pts/5 T 0:00 netscape
21873 pts/5 S 0:25 nedit
```

To kill off the process sleep 100, type the following command. Then type `ps` again to see if it has been removed from the list.

```
$ kill 20077
```

If a process refuses to be killed, uses the -9 option:

```
$ kill -9 20077
```

Note: It is not possible to kill off other users' processes.

A Starting List of Common Terminal Commands

|

- The pipe key | on the keyboard is used to take the output of one command and send it to another command. For instance, to view through the history of commands run, screen by screen, you would combine the history command and the less command as follows:

  ```
  history | less
  ```

apt-get

- The Advanced Package Tool has been developed by the Debian GNU/Linux branch of Linux, and is incorporated into all child Linux operating systems, like Ubuntu and Raspberry Pi OS, that use the Debian core.
  - Synchronize the list of packages available by typing:

    ```
    sudo apt-get update
    ```

  - Upgrade all the software packages currently installed by typing:

    ```
    sudo apt-get upgrade
    ```

  - Install new applications, for instance the Apache web server, version two, by adding the following to apt-get:

    ```
    sudo apt-get install apache2
    ```

cat

- The cat command, short for concatenate, lists the content of a text file. In the case of very long file contents, the less command works better.

  ```
  cat science.txt
  ```

cd

- The cd, or change directory, command helps us to move around the file hierarchy tree structure of Linux.

/			the root of the tree, containing all other main directories
	/bin		the essential binary executable files of the system
	/boot		the startup files for Linux
	/dev		the entry point for peripherals
	/etc		the commands and files necessary for system administration
		/etc/vnc	as one example, the subdirectory for commands and files necessary to administer the VNC server
	/home		personal user directories
		/home/pi	as one example, the subdirectory for the base user "pi"
	/lib		shared libraries essential for the system during startup
	/mnt		mount points for temporary partitions like a USB hard drive
	/opt		packages for some supplementary applications like minecraft-pi, pigpio, sonic-pi, and Wolfram
	/root		directory of the root administrator
	/sbin		essential high-level binary executable files
	/tmp		temporary files
	/usr		secondary hierarchy
		/usr/bin	majority of binary executable files and user commands
		/usr/include	header files for the C and C++ programming language
		/usr/lib	shared libraries for the system
		/usr/local	data pertaining to the programs installed on the local machine by the root user
		/usr/sbin	the binary executable files that are not essential for the system but that are reserved for the system administrator
		/usr/share	reserved for non-dependent data of the architecture
		/usr/src	code source files
	/var		variable data

clear

- The clear command is helpful when your terminal has become cluttered with lots of output from commands and you want to start with a clean slate. This will clear all text and leave you with the $ prompt at the top of the window.

df -h

- Use this to find out the size of your available disk spaces, how much is currently being used, and how much is available. Many are tmp, or temporary spaces, while others are formally defined /dev devices.

dig

- Dig, the **D**omain **I**nformation **G**roper, is a tool used to search through Domain Name System (DNS) servers for various DNS records. As such, it can be very useful for troubleshooting public IP name and IP address problems. (NOTE: IP addresses in the ranges 10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, and 192.168.0.0 to 192.168.255.255 are all private. That is, they do not need to be registered to be used on local networks, and therefore do not appear in the DNS system of servers.)
- To find the public IP address for the IP name google.com:

```
dig google.com
```

- To find the public IP address for the IP name ischool.illinois.edu:

```
dig ischool.illinois.edu
```

- To find the IP name for the public IP address 216.58.192.174:

```
dig -x 216.58.192.174
```

- To find the IP name for the public IP address 192.17.172.3:

```
dig -x 192.17.172.3
```

find

- The find command is used to find files using their filename.

grep

- The grep, or global regular expression print, command is a powerful tool to search for a given string in a file, or in a range of files within a directory, or even a directory and subdirectories. For fun, try running this command in the pi home directory:

```
grep -r "the" *
```

head

- The head command writes the first ten lines of a file to the screen. First clear the screen then type:

```
head science.txt
```

history

- The history command will list the history of commands you've written, at least back a certain amount of time or a certain number of commands run. A sweet little trick is to use the **!** character followed by the number of a command to repeat that command, for instance, the 137th command you ran:

```
!137
```

hostname -I

- Shows the IP address(es) of your Raspberry Pi

ifconfig

- Used to check the status of wired Ethernet (eth0) and wireless Ethernet (wlan0) Internet connections.
    - inet = the assigned Internet Protocol version 4 (IPv4) address, if any
    - inet6 = the assigned Internet Protocol version 6 (IPV6) address, if any
    - ether = the MAC address, roughly equivalent to a serial number for that specific network adapter

iwconfig

- Indicates which WiFi network the wireless adapter is using at the moment.

iwlist

- Used to get more detailed wireless information than is provided from iwconfig about a wireless adapter interface. The most commonly used argument with iwlist is scan, giving a list of Access Points and Ad-Hoc cells in range at the moment.

```
iwlist wlan0 scan
```

- To narrow down the information provided, it can be useful to combine this with a grep command to search just for the ESSID, or Extended Service Set Identification identifying a specific 802.11 wireless network:

```
iwlist wlan0 scan | grep ESSID
```

kill

- Terminate the process based on its Process ID (PID) number. By default, the kill command sends a request to the PID specified, officially known as a SIGTERM, asking for it to shut itself down:

```
kill 12345
```

- To forcefully quit a PID specified, officially known as a SIGKILL, you need to add in an additional kill statement.

```
kill -s KILL 12345
```

- You can choose to kill several PIDs all at once, also.

```
kill 12345 12346 543 139
```

- To kill all PIDs with the same name, for instance, all PIDs running apache2 sessions, you run the killall command instead.

```
killall apache2
```

less

- The less command is considered to be a more powerful version of the older more command. That's right, the creators thought less is more! Instead of using the cat command to list a file, you can use the less command. It opens the file for viewing. From there, you can use the space bar to jump ahead one screen, the b character to scroll back one screen, the down and up arrows to go down and up one line, and many other tags to do a range of other movements. Hit [Q] to quit out of the file. Less is also used instead of `cat` for reading long files.

ls

- The ls, or list, command is the most common way to get a list of files. This is often combined with the options `-la` and then an argument to search a specific directory.

lsusb

- Lists the USB hardware currently connected to your Raspberry Pi.

man

- The man, or manual, command is used to show the basic operating manual for an available command, including why to use it, the options available in its use, defaults, some examples, and the authors of the command.

mkdir

- The mkdir, or make directory, command allows you to create a new directory. As an example, let's use the `cd` command to make sure we're in our home directory (the easiest way to do that is to use the ~ argument), then use the `mkdir` command to make a Python subdirectory.

```
cd ~
mkdir Python
```

mv

- the mv, or move, command allows users to move a file to another directory, with the first argument indicating which file is to be moved, and the second where it is to be moved. For instance, if you create a new Python command in the pi home directory and later want to move it to a directory called Python in the pi home directory:

```
mv /home/pi/somepythoncode.py /home/pi/Python
```

nano

- A commonly used default text editor. For instance, to edit a file you found using `ls` that is called example.txt, type:

```
nano example.txt
```

ping

- A core command for testing the connection between two devices on a network, whether local or over a wider network (e.g., campus network, Internet). This can be done using an Internet Protocol address, or an IP address number. For instance, both of the following check the connection between your device and the same other device:

```
ping illinois.edu
ping 192.17.172.3
```

ps

- The ps (processes status) command is one of the earliest native Unix/Linux utilities for viewing information about running processes, the operations on computer data within programs being currently run.
- Print all processes currently running by **a**ll users, listed by **u**ser, e**x**tended to include TTY commands and also internal commands:

```
ps aux
```

- Print all processes currently running in wide, or double wide format:

```
ps auxw
ps auxww
```

raspi-config

- The core configuration command for the Raspberry Pi OS as run from the terminal interface. This can also be run from the graphical user interface.

reboot

- Use this command to do an immediate reboot of your computer.

rm

- The rm, or remove, command only removes files. However, the **-r** option will recursively remove a directory, any subdirectories, and any files within those directories. To go even further, you can add in the **-f** option to forcefully remove items. Use this command with great care:

```
rm -rf somedirectory
```

- **And do not, under any circumstances, ever do this command, as you'll put an end to the operating system and all files within:**

```
rm -rf /
```

rmdir

- The rmdir, or remove directory, allows you to remove a directory and files held within that directory.

scp

- Short for "secure copy." Uses the secure shell, or SSH, to copy files to and from a remote device. For instance, to copy the example.py Python file from your laptop to the user pi's home directory on the Raspberry Pi, type in the following (example uses an IP address in the IllinoisNet_Guest local area network domain):

```
scp example.py pi@172.16.108.44:/home/pi
```

- To copy the file anotherexample.py Python file from the user pi's home directory on the Raspberry Pi to your laptop, type in:

```
scp pi@172.16.108.44:/home/pi/anotherexample.py
```

screen

- The screen command detaches a terminal window, creating a separate window within which you can access and control the Raspberry Pi.

shutdown

- Use this command to shut down your Raspberry Pi computer.
    - Shutdown immediately by typing "sudo shutdown -h now"
    - Schedule a shutdown time, for instance 11:30 PM, by typing "sudo shutdown -h 23:30"

ssh

- The secure shell, or SSH, provides remote login access to another computer. On MacOS laptops, to access your Raspberry Pi open the terminal and type the following (example uses an IP address in the IllinoisNet_Guest local area network domain). Before proceeding, you'll need to have written down the WiFi IP address of your Raspberry Pi, or determined it via the console using the USB to TTL cable.
- On Windows, you can use PowerShell to SSH. You can also use PuTTY, the free and open source software which supports several network protocols for connecting two devices. [Download PuTTY](#).

startx

- When a Raspberry Pi or other Linux computer boots straight to a console window, you can prompt it to further boot to reach the GUI by typing:

```
sudo startx
```

sudo

- Placing sudo in front of any of the above commands asserts increased authority if the user logged in has been granted such in the `/etc/sudoers` configuration file. Working in your home directory generally isn't a problem. But often you'll find running higher order control commands and editing of files in system-wide directories will require the running commands like `ls, rm, mkdir` and others with a leading sudo in front of it to exercise your advanced powers of control.

tail

- The tail command writes the last ten lines of a file to the screen. Clear the screen and type:

```
tail science.txt
```

touch

- The touch command allows you to create an empty file. This can be useful if you want to then fill the file with notes, output data from a command, or a range of other miscellany. You can specify a file extension (such as .txt or .py) or name the file without an extension.

traceroute

- A core command to test the connection between two devices on a network, testing each intermediate connection between the two. For example:

```
traceroute illinois.edu
```

UP key

- This is not a command typed into the terminal, but rather a key on the keyboard you can hit to list previous commands you've entered. Much simpler than fully typing in a command again to rerun it.

write

- Open a chat session with another user connected via terminal, console, or other TTY (officially, teletype) connection. For instance, if a user named "visitor" has logged into the Raspberry Pi using SSH, you could say hello to them by typing the following:

```
write visitor
Hello, visitor. How can I help?
```

- This chat could then be continued back and forth until you end the write session by holding down the CTRL key and hitting the letter C on your keyboard. This is known as CTRL-C, stated as "control c."

Raspberry Pi Networking 1010

While the world originated analog, with physical data such as audio and visual signals that change continuously, we now also make regular use of digital signal processing to turn the continuous analog signals into number samples, often transmitted using binary 0/1 sets. A leading mechanism for sharing binary digital data streams of all kinds is the Internet Protocol. In this set of exercises, you'll get a chance to connect your Raspberry Pi to a Local Area Network (LAN) and test out performance of the Raspberry Pi on the Internet.

Some choices of interest may include:

- Exercise: Connect a Raspberry Pi to a Wired Ethernet Port
  - In many cases in which reliable, high-performing Internet is needed, the best choice is to make use of wired Ethernet.
- Exercise: Connect to a WiFi Network from the Raspberry Pi Desktop
  - Recommended if you are running your Raspberry Pi with desktop GUI and have it connected to a keyboard, mouse, and monitor.
- Exercise: Connect to a WiFi Network via the Command Line
  - Recommended if you are running your Raspberry Pi headless, that is, without a monitor (NOTE: the Orange Unit chapter 4C: Getting Started with the Raspberry Pi provides instructions on using Real VNC for headless connectivity[1] between a personal computer and a Raspberry Pi, and instructions on adding an Adafruit PiOLED to display the IP address assigned currently to the Raspberry Pi for remote access.)
- Exercise: Unusual Additional Registration Requirements
  - Some Wi-Fi networks employ measures to restrict access to enhance security. Here's one example of setting up a reliable network connection used at the University of Illinois at Urbana-Champaign.
- Configuring Networking on the microSD card (Raspberry Pi Website)
  - It is possible to create or further edit a wpa_supplicant.conf file in the boot folder of the microSD card on which the Raspberry OS is installed. You use plain text to enter in the Local Area Network's identifier (its SSID, which is the Wi-Fi name you often see listed in drop down menus) and passphrase, and therefore can be easily be accessed by anyone who picks up the microSD card. Use this option only if you're OK with others being able to log into this LAN.

1. Instructions for Real VNC start at step 6 of Exercise: First Boot of the Raspberry Pi OS.

We live today in a wireless society. In some cases, even our phones no longer provide a plug-in for our wired headsets, moving us to instead primarily use Bluetooth headsets. When our mobile phones are located someplace in a building with limited cell service, we can make use of the building's Wi-Fi connection to get us to that cell service. Our laptops and even many desktops today depend on Wi-Fi Internet access. After all, it's much simpler and cleaner-looking than yet another wire.

When mobility, flexibility, and simplicity are important, wireless Ethernet connectivity is a solid choice.

But here are some key reasons why wired Ethernet remains the better choice for networking more often than you might think.

- Performance: Wired Ethernet makes use of Cat5 or Cat6 cables that provide synchronous upload and download of data at either 100 Mbps or 1 Gbps. The 802.11a/b/g speeds are improving and can now provide creative forms of synchronous transfers of data at upwards of 54 Mbps. Wired Ethernet provides synchronous communication from a single computer to a switch. Meanwhile, the wireless radio waves are shared by all connected devices simultaneously, thereby changing performance without notice.
- Reliability: Cat5/6 cables are designed to limit interference, and are often installed within walls or ceilings to further improve reliability. While WiFi has improved over the years, it remains susceptible to interference from a range of fixed and dynamic factors.
- Security: While the wireless 802.11 protocol and interconnected devices used to provide network access to Wi-Fi devices have improved significantly, it remains that a wired Ethernet connection is less susceptible to security attacks than wireless Ethernet.

When doing some home improvements many years ago, I used the opportunity to run wired Ethernet to a few locations in each major room of the house. This helps improve performance of all devices, as the high-demand Roku, solar panel interface, and office computer all make use of the wired Ethernet Cat5b cables. And especially at times when many devices are making active use of the network, the 1-Gbps wired Ethernet connection has proved essential when I am teaching or collaborating online from my home office.

When consulted on new construction and Ethernet, I strongly recommend installation of as much wired Ethernet as possible, especially in libraries, schools, and community centers where there will be more statically located networked devices. This both improves performance, reliability, and security for these static devices and reduces demand for the many wireless devices within the space.

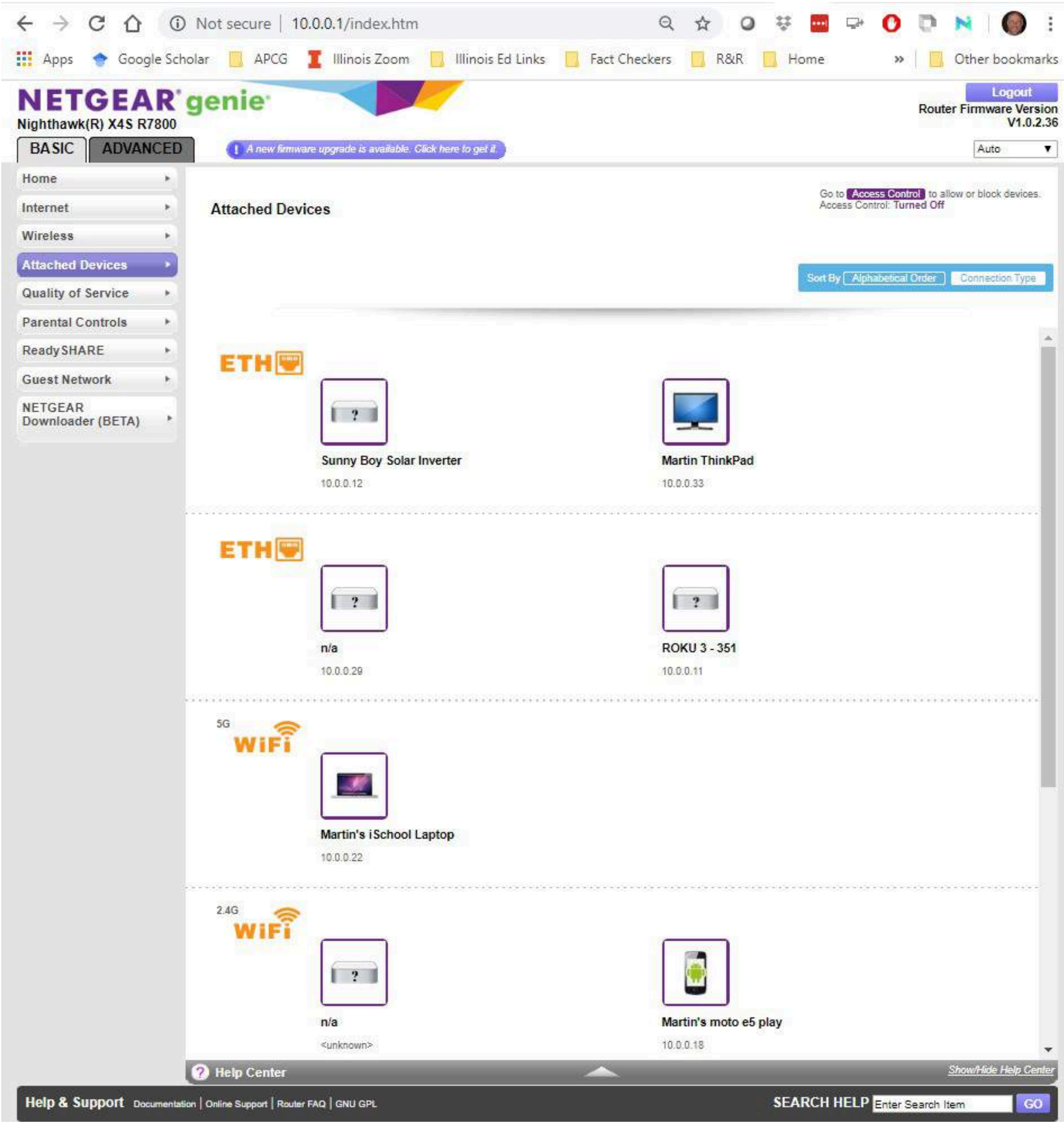
What about the Raspberry Pi? While it is capable of both wired and wireless networking, the choice may vary by project.

For basic educational and prototyping activities, especially when close access or mobility is important to work with breadboarding and electronic testing in a range of locations, Wi-Fi continues to be the preferred means of connecting to the LAN. However, when prototyping moves to the next level and is used for ongoing data collection and distribution—for instance, as a soil and weather sensor database system accessible via web server—I move the Raspberry Pi to a waterproof case and connect it to the LAN via an outdoor Cat5 cable. When the Raspberry Pi is used as a multimedia server or a location-based collective networking toolkit, I've found it ideal to locate the Raspberry Pi right next to the gateway router, connected via a short Cat5 cable.

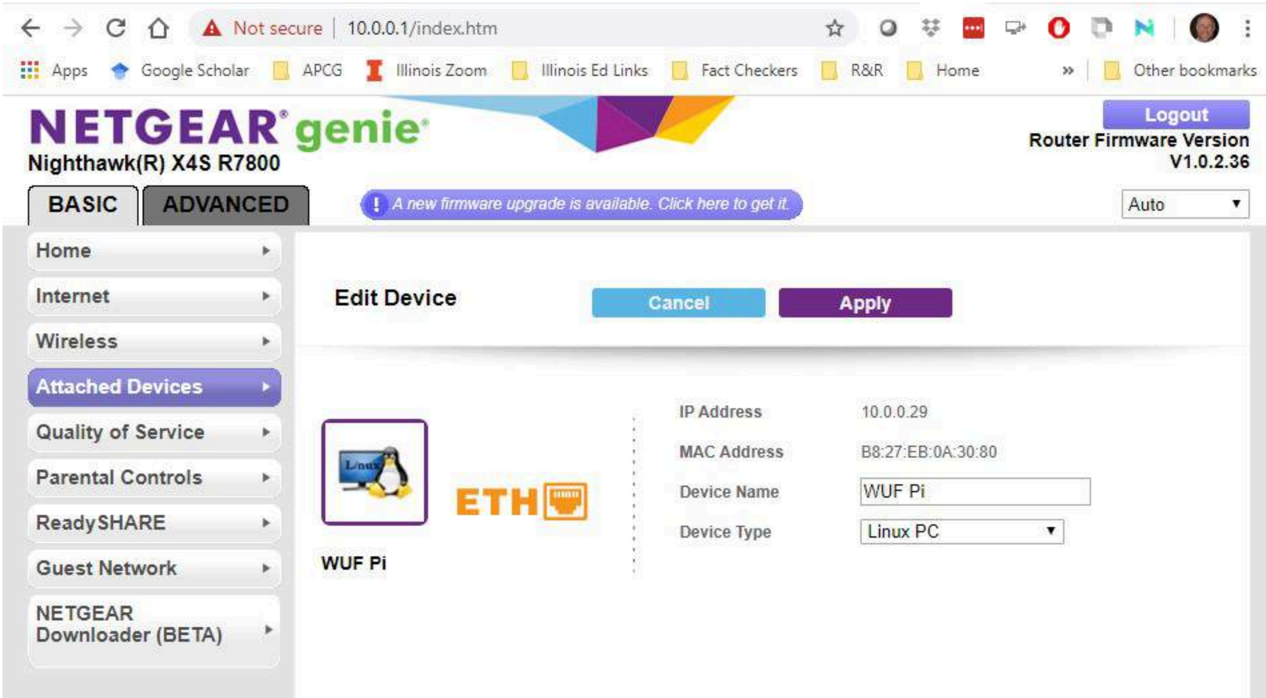
When the Raspberry Pi is running headless (without a keyboard, mouse, and monitor), Secure Shell (SSH) is a quick and easy way to connect to and control the microcomputer. The Pi's microSD card must be edited to enable SSH access within the Raspberry Pi Operating System, but after this, I can connect the Raspberry Pi to a router with a wired Ethernet cable. This will automatically provide it with an IP address that I can use to SSH into the Pi on its very first boot.

Exercise: Connect a Raspberry Pi to a Wired Ethernet Port

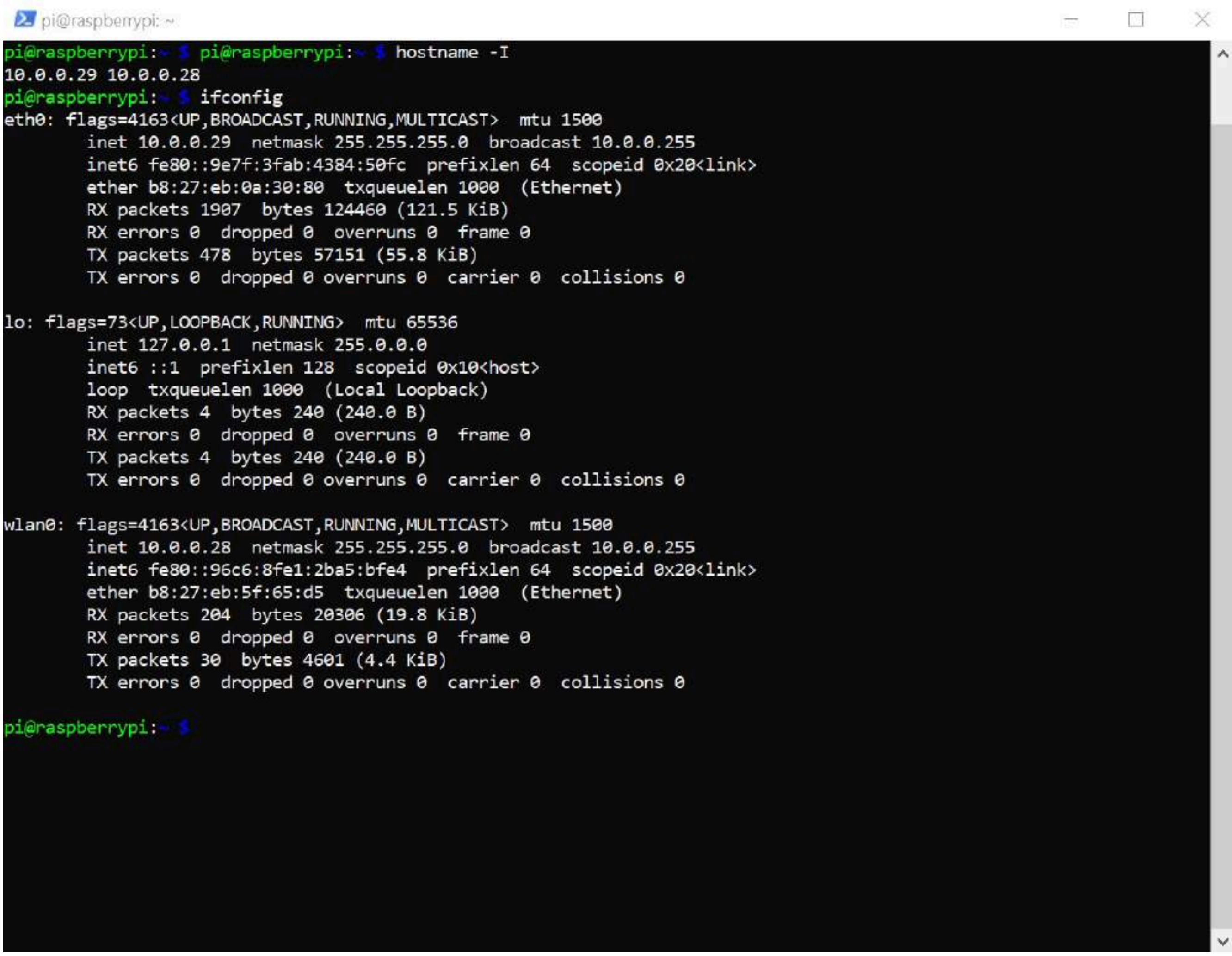
1. Connect one end of an Ethernet cable to the wired Ethernet port on your Raspberry Pi.
2. Connect the other end of the Ethernet cable to an available LAN Ethernet port. Some places where these may be found include:
    - An Ethernet port on the wall.
    - One of the four LAN ports commonly found on home and small office gateway routers used to connect to the Internet service provider.
    - An Ethernet switch that itself connects to a wall connection or to the gateway router for the LAN.
3. Most home and small office gateway routers provide a quick guide with login and password information to help you connect to their web servers to determine the assigned IP address for connected devices.



A first look at the Netgear gateway router's attached devices.



Editing the newly connected, unknown device—the wired Ethernet Raspberry Pi.



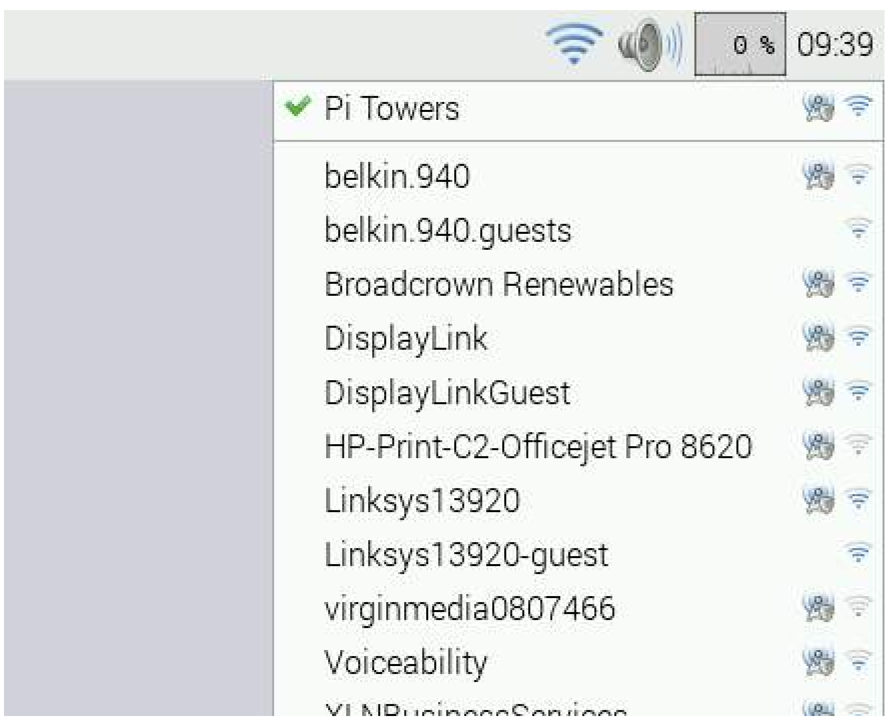
The Raspberry Pi is connected with both wired and wireless Ethernet to the home gateway router, as viewed with a first ssh connection to the 10.0.0.29 address identifed using the Netgear web server's Attached Devices tool. Note that eth0 is assigned the IP address 10.0.0.29, while the wlan0 is assigned the IP address 10.0.0.28.

Exercise: Connect to a Wi-Fi Network from the Raspberry Pi Desktop

A GUI is provided for setting up wireless connections with the Raspberry Pi OS. If you are not using the Raspberry Pi Desktop, you can set up wireless networking from the command line in the next exercise.[2]

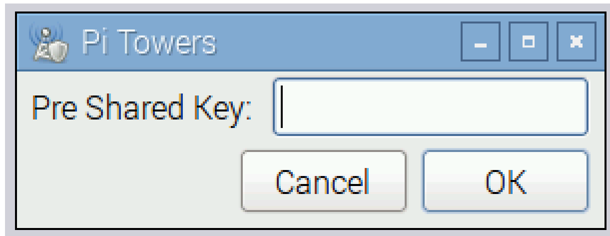
Wireless connections can be made via the network icon at the right-hand side of the menu bar. If you are using a Pi with built-in wireless connectivity, or if a wireless dongle is plugged in, left-clicking this icon will bring up a list of available wireless networks, as shown below. If no networks are found, it will show the message "No APs found—scanning…" Wait a few seconds without closing the menu, and it should find your network.

Note that on the Pi 3B+, which supports the 5G protocol, wireless networking is disabled for regulatory reasons, until the country code has been set. To set the country code, open the Raspberry Pi Configuration application from the Preferences Menu, select **Localisation**, and set the appropriate code.



2. Raspberry Pi learning resources and documentation are licensed under CC-BY-SA 4.0. This exercise is a remix of the "Wireless connectivity in the Raspberry Pi Desktop" tutorial.

The icons on the right show whether a network is secured or not, and give an indication of its signal strength. Click the network that you want to connect to. If it is secured, a dialogue box will prompt you to enter the network key.



Enter the key and click **OK**, and then wait a couple of seconds. The network icon will flash briefly to show that a connection is being made. When it is ready, the icon will stop flashing and show the signal strength.

Exercise: Connect to a Wi-Fi Network via the Command Line

This method is suitable if you don't have access to the graphical user interface normally used to set up Wi-Fi on the Raspberry Pi. It is particularly suitable for use with a serial console cable if you don't have access to a screen or wired Ethernet.[3]

Step 1: Get Wi-Fi network details

1. Connect your laptop to your Raspberry Pi via serial console and begin a remote terminal session to scan for Wi-Fi networks.
    1. Using a TTL to USB cable, connect the black TTL wire to a GPIO Ground pin, the white TTL wire to GPIO 14 (TXD or transmit), and the green TTL wire to GPIO 15 (RXD or receive).
    2. Attach the USB side of the cable to your laptop.
    3. Adafruit provides a lesson on how to run software in the Mac, Windows, and Linux operating systems to connect your computer to the Raspberry Pi from here.
2. To scan for wireless networks, use the command:

- `sudo iwlist wlan0 scan`

3. This will list all available wireless networks, along with other useful information. Look out for:
    - "ESSID:"testing"" is the name of the Wi-Fi network "testing."
    - "IE: IEEE 802.11i/WPA2 Version 1" is the authentication used. In this case, it's

3. Raspberry Pi learning resources and documentation are licensed under CC-BY-SA 4.0. This exercise is a remix of the "Setting up a wireless LAN via the command line" tutorial.

WPA2, the newer and more secure wireless standard which replaces WPA. This guide should work for WPA or WPA2, but may not work for WPA2 enterprise. Visit the FreeBSD manual for WEP hex keys, scrolling down to see the last example on the webpage.

4. You'll also need the password for the wireless network. For most home routers, this is found on a sticker on the back of the router.

Step 2: Connect to a network through raspi-config

1. Once you know the network name (SSID) and password, the quickest way to enable wireless networking is to use the command line raspi-config tool.
2. From the Raspberry Pi terminal window, enter:

- `sudo raspi-config`

3. Select the **Network Options** item from the menu, and then the **Wi-Fi** option.
4. On a fresh install, for regulatory purposes, you will need to specify the country in which the device is being used.
5. Then set the SSID of the network, and the passphrase for the network.

Exercise: Unusual Additional Registration Requirements

Businesses, schools, libraries, and other organizations use various measures to restrict access to their network to enhance security. This often means that a password is required to access a network, whether wired or wireless; two-factor authentication may also be required. Sometimes, only registered devices can access a network. And sometimes, the level of registration determines what can be done over the network.

At the University of Illinois at Urbana-Champaign, two primary wireless Ethernet networks are available for general use: IllinoisNet and IllinoisNet_Guest. For the most part, IllinoisNet_Guest is used for easy first access to the core technical services. Visitors can use this to create a 24-hour login, while regular users can use this to gain up to 12-month access for certain devices, such as gaming consoles, streaming devices, and your Raspberry Pi. As one example, the following steps will help those who are part of the University of Illinois at Urbana-Champaign system to register their Wi-Fi nodes, such as a Raspberry Pi, to the IllinoisNet_Guest network.

MAC Address Overview

Before we start, let us take a step back and look at how devices are differentiated. In the Orange Unit, we learned how different integrated circuits perform different tasks in a device. For a device to connect to a network, it requires an integrated circuit called a **Network Interface Card (NIC)**. There is a NIC for each type of network connection: Ethernet, Wi-Fi, and Bluetooth. The Raspberry Pi 3 includes all three NICs.

To differentiate between different NICs and different devices, each NIC is assigned a **Machine Access Control (MAC) Address**. The MAC address is comprised of 6 groups of 2 hexadecimal digits. Hexidecimals contain the numbers between 0-9, but also include the uppercase letters A-F, to give 16 possible digits. Some information about your device is contained in the MAC address, such as the vendor or manufacturer of your device.

Register Device

The following steps will walk you through registering your device.

First, you will need to find the MAC address of your Raspberry Pi's Wi-Fi network card. To do so, open a terminal window in your Raspberry Pi and type the following command:

```
$ ifconfig wlan0
```

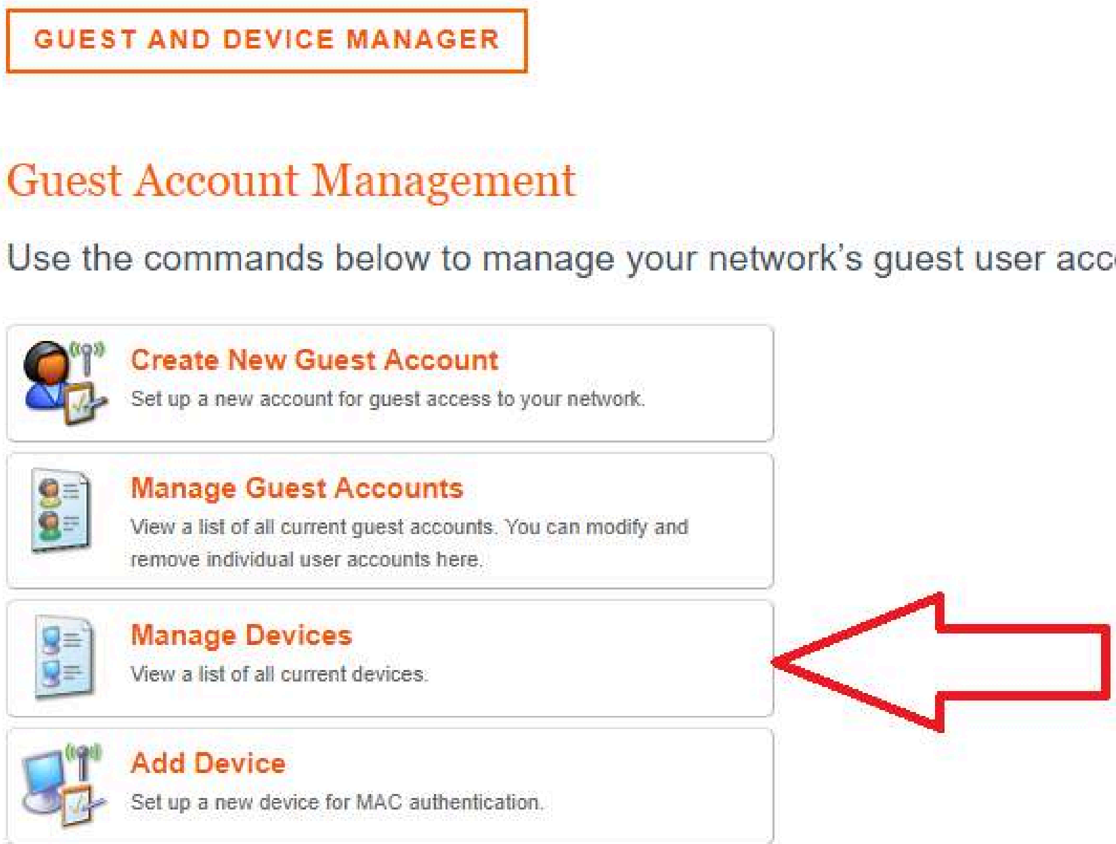
This will output the data related to your wireless network interface. The first line should contain:

Link encap:Ethernet   HWaddr XX:XX:XX:XX:XX

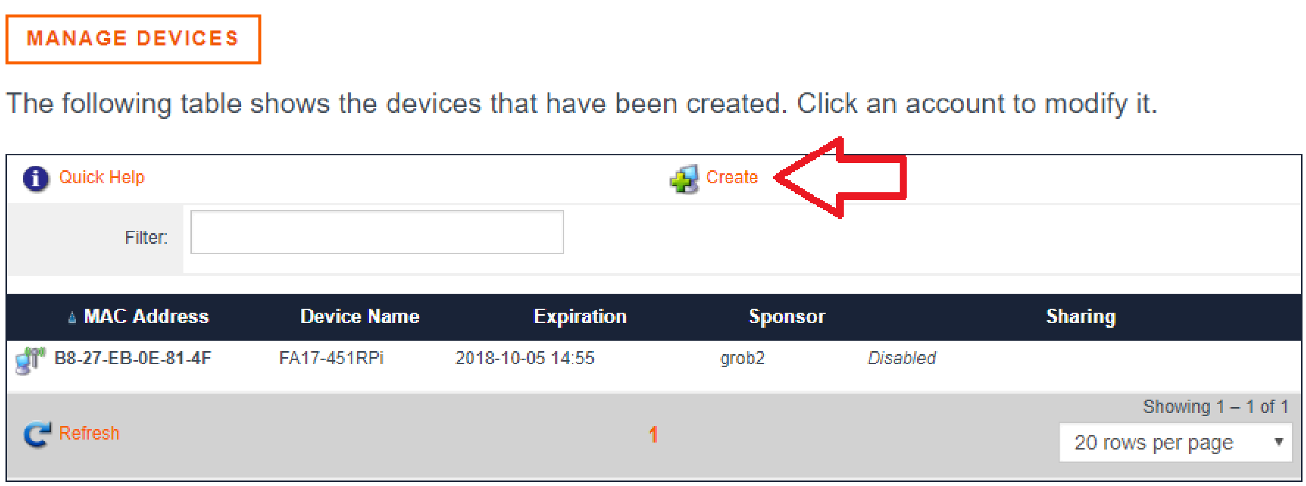
Instead of Xs, your MAC address for your Wi-Fi NIC will be printed.

Either write this down or leave the window open, as you will need this information to register.

Next, using your laptop, navigate to http://go.illinois.edu/illinoisnetdevices and log in using your NetID and password. Click the "Manage Devices" button.



Click the "Create" button to create a device account.



Insert the following information for each required field:

1. MAC Address: Copy or type the MAC address found in the steps above.
2. Device Name: *IS401-RaspberryPi*
3. AirGroup: Leave the box unchecked.

4. Account Activation: *Now*
5. Notes: Leave empty unless otherwise needed.
6. Terms of Use: Check box to accept the terms of use.

Click “Create.” Restart your Raspberry Pi before attempting to connect to the network.

Network Troubleshooting

Initial and ongoing evaluation and documentation are two strategies at the core of effective troubleshooting. These can be applied to both your Local Area Network (LAN) and the interconnection of your LAN with the Internet through your Internet Service Provider (ISP) interconnect device(s). Here are some key techniques I use, which help me to rapidly move up the tiers of support when a problem is encountered.

Essential Network Troubleshooting Tools

In some cases, we connect computers and other electronic devices using serial communications protocols such as Universal Asynchronous Receive Transmit (UART) and Inter-Integrated Circuit (I2C). But today it is very common to instead or also use the Internet Protocol (IP), whether just within a Local Area Network (LAN), as part of a Community Network, or as part of the Internet. Below are some of the key investigative tools we can use to assess these IP network connections.

Link Lights

- LED light found on wired Ethernet cards and switches.
- When lit, indicates that the card and switch are on and working properly and the Ethernet cable is plugged in and working properly.

Network Manager

- Operating system utility that shows network configuration.
- An IP address of 0.0.0.0 or one that starts with 169.254 indicates you are connected to the media but aren't getting a proper IP address from the address server (usually the gateway router).

Ping

- Run from the command line or terminal application to test if you can communicate with another node.
- ping IP_ADDR or ping IP_NAME (e.g. ping google.com)
- Times of 1 ms or less are good on a local network; 10-50 ms or less are good on the Internet.

- Hint: ping different nodes to track down faulty component.

Traceroute

- Run from the command line or terminal application to test the performance of each router between nodes.
- Windows: tracert IP_NAME
- Linux: traceroute IP_NAME
- macOS: traceroute IP_NAME
- Smartphone apps available

Note:

- Asterisks (*) indicate failure or firewall
- Hop 1 is usually your router; hop 2 is your ISP

Speed Test

Web-based Internet connection speed test:

- https://speed.measurementlab.net/
- https://www.speedtest.net
- https://www.speakeasy.net/speedtest/

To access the command line:

- **Windows**: Click on the Start button, choose run, type in *cmd* in the open bar and hit the enter key.
- **macOS**: Click on the spotlight, type in *terminal* and choose Terminal under Applications.
- **Linux**: Click on the Start button, choose Terminal under Accessories.

Previously, we saw how the "data beanbag" could be passed between just two nodes, your laptop and the Raspberry Pi. But even there, the server and client VNC (Virtual Network Computing) software are standing in the circle along with the hardware (computers). The Ethernet cable served to carry the data beanbag between these players.

For most of our work on computers, smartphones, and other networked devices, the beanbag needs to pass between many more interconnect devices to get between the starting and ending node. And when gathering information from something like a web server, the server itself may first act as a client, dynamically collecting data from a range of other servers, again using interconnect devices, before packaging this into a web page that it passes back to you.

Let's use some essential network troubleshooting tools to follow the data beanbag as it's passed around the circle. Along the way, let's do a little bit of testing to make sure our networking is performing as expected, and if not, where it might be failing. This is a great opportunity to work collaboratively.

Exercise: Identify Link Lights

The most general of tests is to see if a wired Ethernet media is making a connection between a node and a network switch – or in the case of our Raspberry Pi/laptop connection, two nodes. Look for a couple of little LEDs on the Raspberry Pi where it connects to the Ethernet cable. When the cable is fully plugged in and the Raspberry Pi is turned on, you should see one or both come on (red indicates it's connected, green indicates it's transferring data).

Look at your laptop. Can you see one or two little LEDs where the Ethernet cable is connected? Some laptops with Ethernet ports, particularly those from Apple, no longer include the link lights, figuring if you want to know, you can open up software to find the answer. And some laptop producers change around the color of the LEDs, or use them for different purposes. Lastly, some laptops no longer provide Ethernet ports.

But regardless, even seeing these LED link lights on one of the nodes generally indicates you are fully and properly connected at both sides of the Ethernet cable. If you see active link lights, you're good to go. If you don't see a link light active, or even if you do, unplug the Ethernet cable and look again. Has anything changed? What about if you plug it back in? If you use a dongle to connect the Ethernet cable to your laptop, what happens if you leave the Ethernet cable plugged into the dongle but unplug the dongle from your laptop? Take a few minutes to compare notes with others who are doing the same but with other types of laptops.

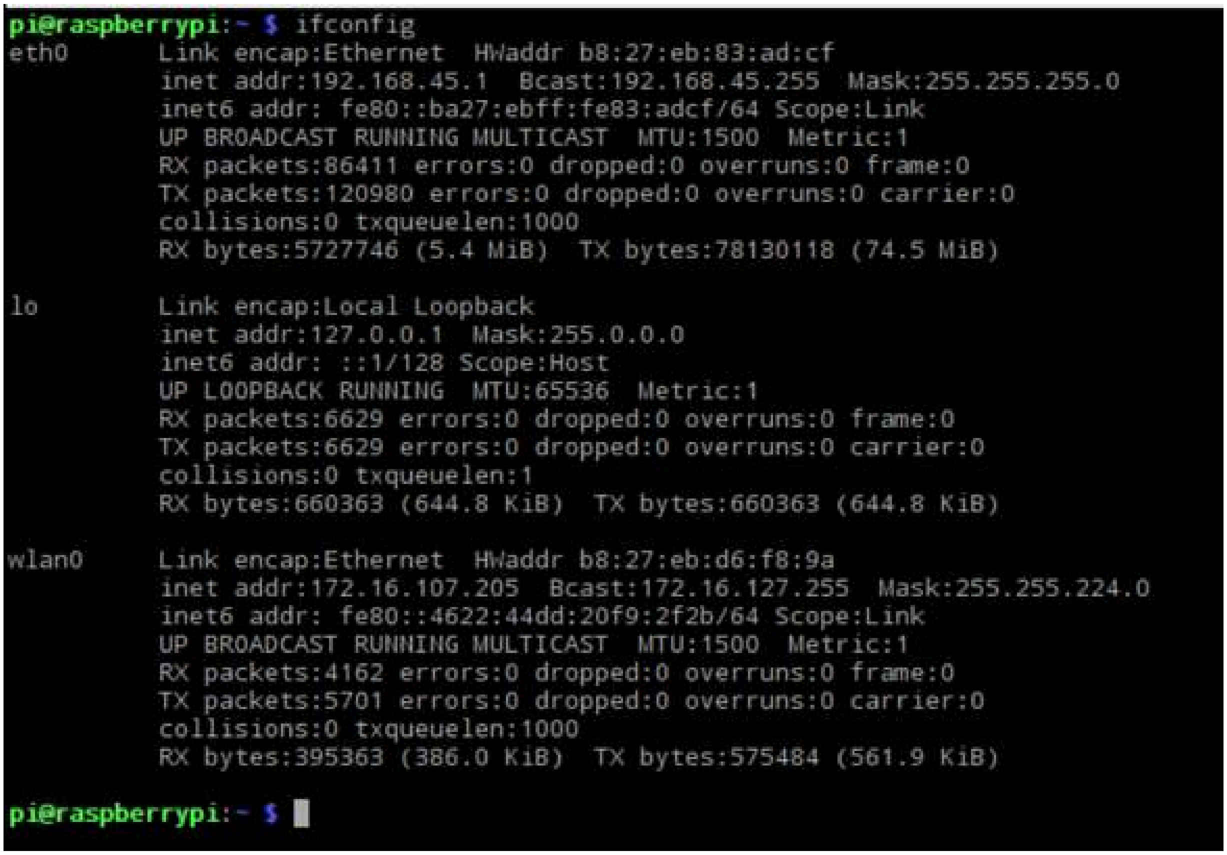
Exercise: Explore Your Network Configuration

All computers have some form of network management software that you should be able to access to see the network configuration. This varies considerably between computer operating systems, and even between versions of the same operating system. And sometimes you need administrative access to actually have significant access. All the same, try to track down how you might do this on your computer. Pair up with someone using the same type of computer as yours and see if you can track it down. Take some notes.

Another way to do this is from the command line or terminal window. On the Raspberry Pi or on a Mac you would type in:

```
$ ifconfig
```

In response, you should see something like the following:



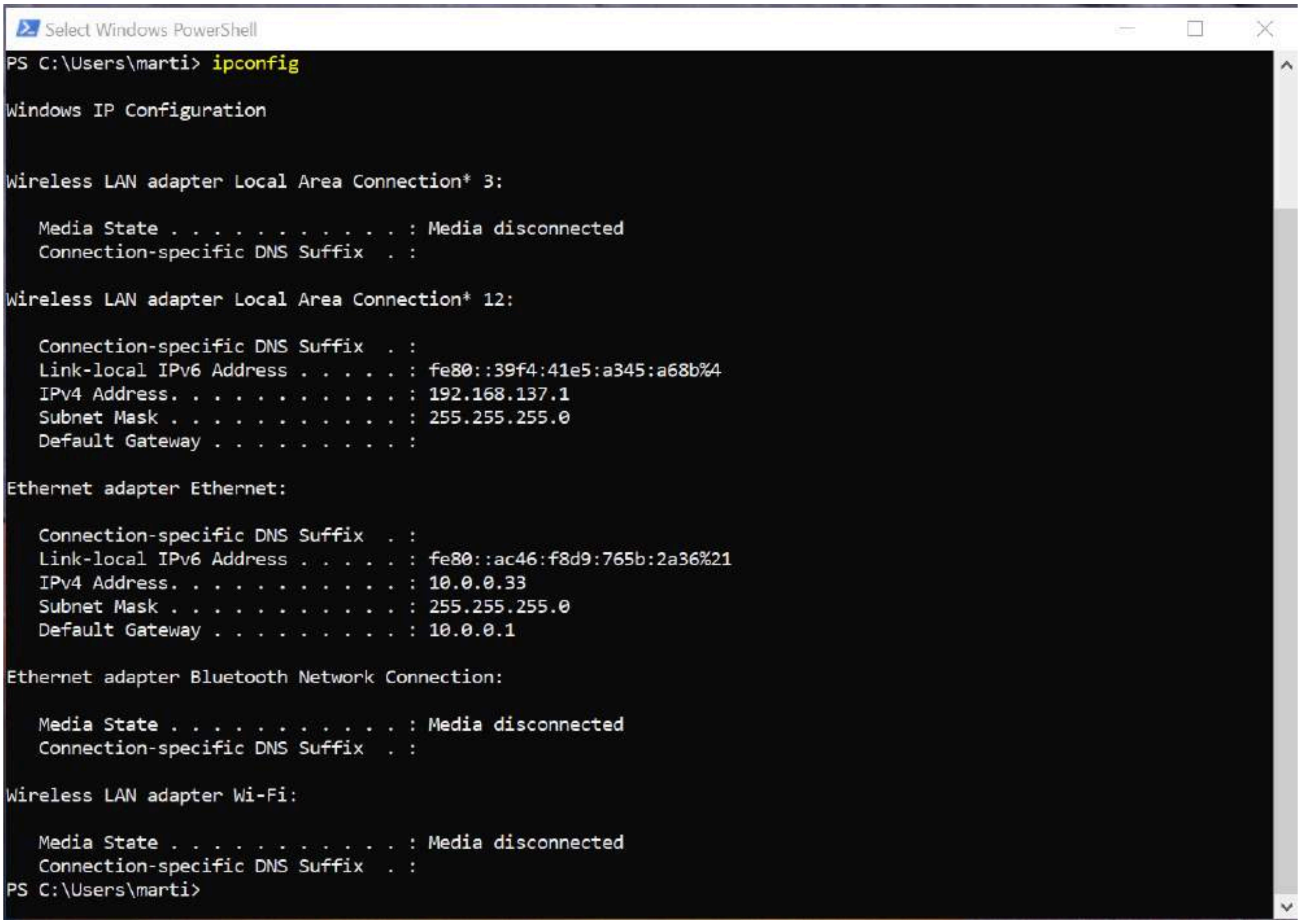
```
pi@raspberrypi:~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:83:ad:cf
          inet addr:192.168.45.1  Bcast:192.168.45.255  Mask:255.255.255.0
          inet6 addr: fe80::ba27:ebff:fe83:adcf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:86411 errors:0 dropped:0 overruns:0 frame:0
          TX packets:120980 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5727746 (5.4 MiB)  TX bytes:78130118 (74.5 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6629 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6629 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:660363 (644.8 KiB)  TX bytes:660363 (644.8 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:d6:f8:9a
          inet addr:172.16.107.205  Bcast:172.16.127.255  Mask:255.255.224.0
          inet6 addr: fe80::4622:44dd:20f9:2f2b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4162 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5701 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:395363 (386.0 KiB)  TX bytes:575484 (561.9 KiB)

pi@raspberrypi:~ $
```

In the above ifconfig, you can see the Raspberry Pi has been assigned an eth0 (wired Ethernet) IP address of 192.168.45.1 and a wlan0 (wireless Ethernet) IP address of 172.16.107.205.

On a Microsoft Windows command line or PowerShell terminal you would type in:

```
> ipconfig
```

In response, you should see something like the following:

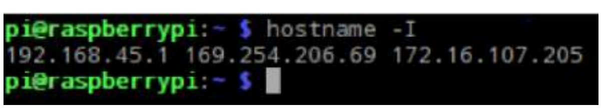


In the above ipconfig, you can see the Microsoft Windows 10 laptop has been assigned a wired Ethernet IP address of 10.0.0.33 and a wireless Ethernet IP address of 192.168.137.1.

In the Raspberry Pi terminal window, you can also type in:

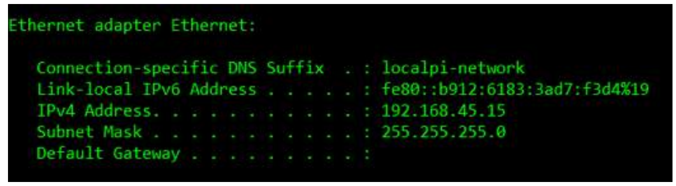
```
$ hostname -I
```

The output from that command should look something like the following:



In the above example, we see the Raspberry Pi has the IP addresses we found using the ifconfig: 192.168.45.1 and 172.16.107.205. But in this case we cannot identify which are wired Ethernet and which are wireless Ethernet. We also cannot define using hostname the new listed IP address of 169.254.206.69 (this is actually part of the private IP address range used for Automatic Private IP Addressing, or APIPA, a failsafe used by some devices when no other IP address is assigned to a device).

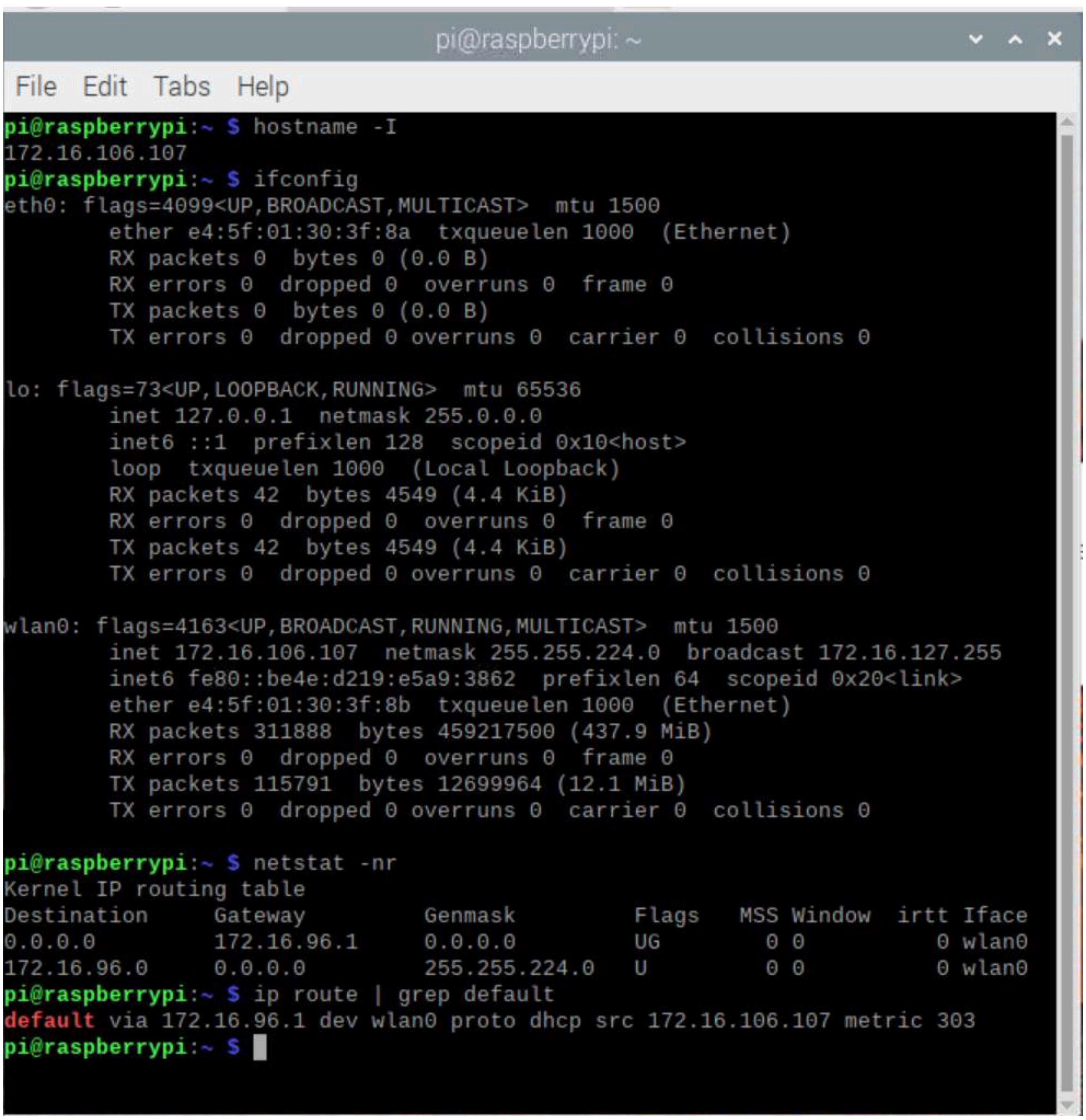
Whether you run ifconfig on a Mac, or ipconfig on Windows, compare the outputs with those you see on your Raspberry Pi to develop a first picture of your LAN. Indeed, you can find a range of apps for your smartphone to run similar commands to identify the IP addresses these are also assigned by the LAN's router or other Dynamic Host Configuration Protocol Server (DHCPS).



```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : localpi-network
   Link-local IPv6 Address . . . . . : fe80::b912:6183:3ad7:f3d4%19
   IPv4 Address. . . . . . . . . . . : 192.168.45.15
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

There are a couple of other helpful Linux commands to further track down information about your LAN, including the Local Area Network IP Address assigned to the router which then connects you to the your Internet Service Provider (ISP), including "netstat" and "ip route":

```
$ netstat -nr
$ ip route | grep default
```



```
pi@raspberrypi: ~
File  Edit  Tabs  Help
pi@raspberrypi:~ $ hostname -I
172.16.106.107
pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether e4:5f:01:30:3f:8a  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 42  bytes 4549 (4.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 4549 (4.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.106.107  netmask 255.255.224.0  broadcast 172.16.127.255
        inet6 fe80::be4e:d219:e5a9:3862  prefixlen 64  scopeid 0x20<link>
        ether e4:5f:01:30:3f:8b  txqueuelen 1000  (Ethernet)
        RX packets 311888  bytes 459217500 (437.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 115791  bytes 12699964 (12.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         172.16.96.1     0.0.0.0         UG        0 0          0 wlan0
172.16.96.0     0.0.0.0         255.255.224.0   U         0 0          0 wlan0
pi@raspberrypi:~ $ ip route | grep default
default via 172.16.96.1 dev wlan0 proto dhcp src 172.16.106.107 metric 303
pi@raspberrypi:~ $
```

Exercise: Test the Network Node-to-Node Path

When your request for a web page is sent to a web server, and when the web server sends back that web page, the information is broken into a bunch of much smaller “data beanbags” called packets. While packets often take the same route along the circle between starting and ending nodes, each packet actually might change its course around the circle, depending on how the different interconnect devices it must pass through are performing at a given moment.

Let’s imagine I’m on an extended vacation, traveling by bicycle, and that I’m doing this without a physical or software-based map of any kind. I just know the name of my starting point and ending point, and I have a general sense of direction to get there.

To start, I might go to the local bike store/club and ask for suggestions. They may give me a perfect set of roads to get me from here to some intermediary destination that they’ve biked before. Great! Off I go. On arriving at that intermediary location, I’ll find the local bike store/club and get a new set of directions. They may mention that the usual route is preferred, but that there’s some construction and so it’s more of a rough ride. As a result, they recommend an alternate route. Great! Off I go again. As I near my destination town, I need to find one more stop-off site to get directions to reach my goal destination: Beer, Bed and Breakfast. With directions in hand, I complete the ride.

This is the way of internets. While some interconnect devices, for instance your home gateway router, have only one way in and one way out if you are to leave one network and join another, the larger routers that make up an Intranet or Internet can provide multiple different routes in case of decreased or failed performance of one route at the given moment.

There are three widely used pieces of software to do network testing. Two, ping and speed test, give you a general overview of the connection between two nodes. But the third, traceroute, will show you each of the router hops between two nodes.

Ping

Ping started as a text-based software installed by default on all major computers. There are now some graphical ping apps available, including ones for smartphones. First, if you’ve identified your gateway router, such as the gateway of 10.0.0.1, let’s use ping to test that connection. On your laptop, open a terminal (Mac) or cmd (Windows) window and type:

```
$ ping  172.16.96.1
```

The command ping sends a ping to a destination node. If all is good, the destination node returns back a ping. Well-functioning LANs should have times listed of less than 1 ms, and should show 0% loss.

On the Mac and Raspberry Pi computers, the default is to continue sending, and hopefully receiving, test pings until you hit CTRL-c. On Windows computers, the default is to run four test pings, and then to stop automatically.

Let's test ping, this time from the Raspberry Pi. Open up a terminal on the Pi and do a ping to some of these sites (hit CTRL-c between each, or open up three terminal windows and run the three simultaneously):

```
$ ping illinois.edu
$ ping adafruit.com
$ ping raspberrypi.org
```

Ping sends the information from the node on which you are running it to a destination node, which then returns a ping. You will be told where the ping is returned from if all is well. It will also return a sequence number, and the time in milliseconds (ms) it took to go there and back again. In general, times under 10 ms occur on LANs and ultrafast WANs. Times above 10 ms but under 100 ms are typically found on fast WANs. Times over 100 ms reflect slower WANs, or networks of all types not working at expected speeds.

Sometimes the failure to get successful pings does not represent a failure in the network, but rather represents a security firewall that does not permit ping traffic to pass to and fro along a certain portion of the network path.

Speedtest

Speedtest is a general descriptor for websites such as the M-Lab Speed Test, Ookla Speedtest, and Speakeasy Speed Test, each of which are useful tools for testing a network's speed and connectivity. When troubleshooting problems with your Internet Service Provider, they may ask you to employ such a tool. As a default, these sites don't let you select a destination node to which you will test the network path. But by collecting an extended set of speedtests, it becomes possible to see if the upload and download speeds change at certain times or under certain conditions.

Speedtests generally provide the following key data points:

- Ping, which is identical to what we get from the command line ping application;
- Download, the speed in Mbps that describes the rate of data, measured as millions of bits (0 or 1) passed per second, traveling from a remote destination to your own location.
- Upload, the speed in Mbps of data traveling from you to the remote site.

The United States Federal Communications Committee (FCC) has a Lifeline program in which low-income households are assured ways to get low-cost Internet to the home. In 2020, the minimum download/upload speed has been set to either 25/3 or 100/10. But often promised IP speeds are much higher than the actual speeds at which networks are currently performing. Speedtests provide a very helpful measurement tool in which to collect baseline and current data when filing reports on real Internet access.

Traceroute

Traceroute is another text-based software installed by default on all major computers. Depending on operating system, the program will use one of several underlying network protocols. As a result, some succeed more than others at going through the full set of hops, as sometimes security firewalls are set up to stop one type of network protocol or another at some hop along the way. But while results vary a little, overall traceroute is an essential troubleshooting tool for networking.

One key point before doing the following activities using traceroute: While on Mac and Linux computers you type in `traceroute` in the terminal window, on Windows computers you type in `tracert` in the command line window.

To start, open two terminals on the Raspberry Pi. In one terminal, run ping 10 times and stop:

```
$ ping -c 10 illinois.edu
```

In the second terminal, run traceroute to the same location:

```
$ traceroute illinois.edu
```

Whereas ping does a general test from one node to another independent of the routers along the way, traceroute does a test of each hop between routers between the starting node and ending node. Typically, traceroute does three tests at each hop.

- The first hop indicates the time it took to pass from the starting node to the first, gateway router that moves us from the LAN to the next network between that node and the destination node.
- The second hop indicates the time it took between the gateway router and the second router in the path. For small to mid-sized LANs, this is usually the hop that takes us from the LAN to the ISP. For MANs and CANs, this is often the next hop in this community-based network.
- Subsequent hops move us further along in a MAN, CAN, or WAN until…
- The final hop brings us to the destination node.

When troubleshooting using traceroute, the first hop is a great indicator as to whether the problem is somewhere on our LAN and needs our attention or the attention of our internal technical support team. The second hop often means we need to contact either our ISP or our campus or neighborhood technical support team. Generally, the further we go along, the more the troubleshooting moves out of our hands.

To review, the first hop typically indicates an in-building issue, the second hop a problem with our direct network provider, and further hops a problem outside our realm of control. But if we do a traceroute test of just one destination node, it's still questionable where on the path the problem is occurring.

To this end, traceroute really shines when we perform multiple traceroute tests. We can do this on one node, say a laptop, going to a multiple different sites, some more local and some more distant. We can take it a step further by repeating the range of traceroute tests on different nodes, say on two different laptops. We can take it a step further still by repeating the range of traceroute tests on a node connected to one network and then to another, say on a node connected to the LAN WiFi, then to your local cellular service. (That last test could be done using a smartphone that can connect to either, or by using the smartphone as a hotspot allowing a laptop to connect to the Internet through cellular.)

For this next test, open one terminal or cmd window on your laptop and two terminals on the Raspberry Pi. As a bonus, if you can, open a traceroute app or terminal that can run traceroute on your smartphone. Run the following traceroute tests on each device, and in the second terminal on the Raspberry Pi, a ping to those sites. Compare and contrast the results.

```
$ traceroute illinois.edu
$ ping illinois.edu
$ traceroute zoom.com
$ ping zoom.com
$ traceroute adafruit.com
$ ping adafruit.com
$ traceroute raspberrypi.org
$ ping raspberrypi.org
```

Compare the different number of hops, the performance of hops, and even some of the possible router location identifiers you find. For instance, below are two traceroute results, one going from my home office to adafruit.com, the other to raspberrypi.org. In both cases, the first hop is the the home's gateway router with IP address of 10.0.0.1. The second hop is the ISP's router, 23.249.47.10. The next two hops go through routers with firewalls and so do not give performance reports (I know this because it is consistent over time and over destination).

In the adafruit.com example, the next three hops travel along the ntt.net backbone within the Chicagoland area before getting to a destination router with IP address 104.20.38.240. This means Adafruit Industries has a server themselves or a provider with a location in the Chicagoland area. All

Internet tests are under or just above 10 ms, indicating a high performing Internet connection at every router interconnect device between my house and the site hosting the Adafruit.com server.

In the raspberrypi.org example, we see a journey of routers that moves from the Chicagoland area to somewhere else, possibly VA, that is, Virginia, in the United States, before going to The Netherlands, likely in Amsterdam, before ending up with Mythic Beasts networks, a privately owned hosting ISP. In this case, note that once travel moves from Chicago to the East Coast of the United States, travel speeds move from below 10 ms to 20-30 ms. Then as the route passes over the Atlantic Ocean, it jumps up again to the lower 100 ms performance level. This is much better performance than even a few years ago, and is an example of why it is possible to have more audio and video communications than ever before!

But how does this compare to your own Internet service?

Key Takeaways

Using ping, speedtest, and traceroute, you will build a strong portfolio of normal and problem network performance data for use when things go wrong on your network. Combining a continuous ping written to a file with regular timestamps, you can have a clear picture of when and at what level network performance is impacted. Speedtest provides a second tier of data, and is often the data ISPs want to see themselves.

Using traceroute, some tests will only have one hop as they stay on the local network. Other tests will have multiple hops as they leave one local area network and move to another, and perhaps yet more local area networks until they arrive at the destination local area network. Routers serve as the go-between, passing **packets** between local area networks. Intranets and the Internet are simply multiple interconnected local area networks.

Usually, hop one in a traceroute is from your node to the interconnect device (such as a gateway) in the building hosting your local area network. A decrease or failure in network performance in hop one generally rests on you to troubleshoot.

Hops two and three are more variable, and may indicate a problem on your Intranet or the hop between you and your Internet Service Provider (ISP). If the former, the troubleshooting onus may still be yours, or that of your IT support team. If the problem is at the level of connectivity between your network and the ISP, then it might be time to give them a call. Or, you might need to first collect data on performance variability over time, as ISPs can often shift the burden of testing back on you.

Regardless, the expected results are that you will increase your level of success at troubleshooting.

Security and Privacy

Sara Rasmussen

Jump to a section:

- Overview
- Curriculum for Young People
- Resources for Adults
- Resources for Educators
- Workshop Facilitation Tools
- Go Deeper

Overview

"Security" holds multiple, varied meanings: Digital information security intersects with personal, physical, and financial security and overlaps with personal, political, and economic issues. With respect to digital tools and systems, security seeks to achieve confidentiality, integrity, and availability for all users. Security manifests within technology as a stand-alone product, such as a password manager, but more often it is embedded within everyday technologies. Security becomes especially important with respect to Internet-connected devices. It is a necessary consideration when connecting to a network and engaging in online activities, such as logging into accounts, sending email, securely banking, shopping online, or sharing personal information on social media.

Privacy, according to NYC Digital Safety, is "the extent to which one's personal data is observed, shared, revealed, exploited, or misused in digital spaces and systems, such as the Internet and the ability to exercise control over that personal data."[1]

Through participant observation and participatory action research, Seeta Peña Gangadharan studies digital literacy programs at libraries and community centers serving marginalized groups. Gangadharan found that new internet users, while optimistic about the potential of the internet, encounter a "privacy-poor, surveillance-rich experience," and come to feel a "sense of inevitability" that if they use the internet, they will eventually experience a breach of their privacy and security.[2] At the time of the study (2012-2013), Gangadharan found that these programs barely covered topics of security and privacy, and moreover, instructors were unable to sufficiently answer the questions of program

1. "Privacy and Security in the Library," NYC Digital Safety, accessed June 10, 2020, https://nycdigitalsafety.org/module/privacy-and-security-in-the-library/.
2. Seeta Peña Gangadharan, "The Downside of Digital Inclusion: Expectations and Experiences of Privacy and Surveillance among Marginal Internet Users," *New Media & Society 19*, no. 4 (April 2017): 597, 605. https://doi.org/10.1177/1461444815614053.

participants. Gangadharan warns that, through our analog social systems, marginalized communities have historically been "watched by default," which feeds into the experience of using the Internet. Gangadharan further notes that, even while being offline is rarely an option, being digitally included "means participation in the potentially harmful consequences arising from inappropriate and asymmetric flows of information."[3] This highlights the tension between the necessity and the risks of digital participation.

From their origins, the disciplines of information security and privacy have focused on technical solutions to such risks. This traditional approach either assumes users to be expert technologists, or identifies them as "the weakest link" in the security chain.[4] Security is implicitly designed with an "expert user" in mind, but that's regularly not the reality. Because security and privacy are most often attributes of a technological product, rather than its primary purpose, they are inherently embedded into larger and more complex systems. This makes the topics of security and privacy nearly impossible to isolate; they must be carefully considered with respect to broader digital skill-building efforts and within the user's and community's context.

Moreover, security and privacy awareness are not inherently linked to digital literacy: It is possible to use technologies in a variety of ways without accounting for the threats they present. And likewise, it's reasonable that non-users of a particular technology may choose this path for their own security or privacy-driven reasons.

While the concepts are often abstract, security and privacy are critical facets of community-centered explorations of digital technologies. For a brief introduction, watch Carrie Anne Philbin's Crash Course Computer Science video on Cybersecurity. Below is a curated list of resources on security and privacy, segmented by audience.

Curriculum for Young People

Common Sense Education from Common Sense Media

- **Intended Audience:** Families, kids, and educators
- Quality content, broken down by age group.
- Provides activities for kids, readings for parents, and lesson plans for educators.
- This nonprofit is affiliated with Comcast.
- Looking for lesson plans? Common Sense offers this overview of "23 Great Lesson Plans for Internet Safety."

3. Gangadharan, "The Downside of Digital Inclusion," 608.
4. Mary Ellen Zurko and Richard T. Simon, "User-Centered Security," in *Proceedings of the 1996 Workshop on New Security Paradigms – NSPW '96* (Lake Arrowhead: ACM Press, 1996), 27. https://doi.org/10.1145/304851.304859. Matt Bishop, *Introduction to Computer Security* (Addison-Wesley Professional, 2004), 1.

Protecting Children's Privacy Online – A Guide for Parents, Carers and Educators from CompariTech.com

- **Intended Audience:** Families and educators
- A long but comprehensive article containing both safety strategies and specific instructions for families in protecting their kids.

My Future from The Boys & Girls Club

- **Intended Audience:** Kids
- The interactive features are only available to members of the Boys & Girls Club, but the training modules provide good inspiration as well as links to relevant free online resources.
- There are resources on privacy, online presence, trolls, bullying, etc.

Civic Online Reasoning from Stanford University and MediaWise

- **Intended Audience:** Ages 10-20
- The curriculum includes 30 free lessons and is accompanied by short videos for teachers.
- The modules are for educators to teach students fact-checking and how to evaluate online sources. According to the website, "The lessons and assessments that make up the curriculum provide students with opportunities to apply fact checkers' questions to real-world examples."
- Topics covered: Wikipedia, social media, and determining how real a website is, with practice opportunities built in.

Resources for Adults

TechBoomers

Intended Audience: Targeted to older adults, but good for everyone

TechBoomers provides a general resource for adults who want to learn to use technology and covers many aspects of safety, such as:

- Privacy settings on social media
- Online shopping safety tips
- Tutorials on popular websites and apps

- What is the cloud?
- Data privacy law

Oasis Connections

- **Intended Audience:** Similarly targeted to older adults
- Quick and easy to watch videos with tips on navigating the internet safely, this series covers a lot of topics with broad relevance beyond the specific concerns of older adults.

Surveillance Self-Defense from the Electronic Frontier Foundation

- **Intended Audience:** Adults
- An extremely comprehensive collection of how-tos as well as more detailed explanations of how information security and privacy works.
- Popular guides include how to create strong passwords, how to enable 2FA, how to use encrypted text messaging, and how to secure your social media presence.
- The writing can get a little technical, but they include a glossary, so when you see an unfamiliar term, you can hover over it and get a definition.

Navigating Digital Information from PBS Crash Course

- **Intended Audience:** Teens and adults
- The ability to interpret and assess online information is foundational to online security because it lends itself towards being able to identify and avoid scams.
- Narrated by author/YouTuber John Green, and part of the PBS "Crash Course" series of videos. The name sounds boring, but they make it fun!
- Described as a "practical take" on the Media Literacy series below.
- This YouTube series covers how to fact-check information online, how to interpret the authority of information, and how search engines and social media feeds work. The goal of the series is to teach healthy online information consumption habits.

Resources for Educators

Cyber Safety for Schools Fact Sheet from Readiness and Emergency Management, U.S. Dept. of Education

- **Intended Audience:** Educators and school administrators

- Provides a summary of online safety concerns, legal and policy considerations.
- Provides guidance on planning for online threat incidents that impact students.
- Links to further learning resources (some are a little outdated).

NYC Digital Safety from the New York Public Library

- **Intended Audience:** Designed for a librarian audience, but useful for anyone teaching these concepts to others
- Quality video series to "train the trainers."
- Starts at the basics, and covers internet technologies, online data collection, account security, tracking, scams, and malware.
- Features quizzes for readers to test their learning.
- Includes an extensive bibliography of news articles and more educational resources at the end of each "chapter."

ALA Toolkits for Social Networking from the American Library Association

- **Intended Audience:** Librarians
- A bit dated, but these two links provide a general overview of what's at stake for young people across age groups and provides introductory definitions and further reading on a variety of topics, including social media.

Workshop Facilitation Tools

These are all-in-one toolkits for people who want to lead workshops to educate coworkers or community members on digital safety.

Stronger NYC Communities, Organizational Digital Security for Immigration Justice from the Mozilla Foundation

- **Intended Audience:** Organizational leaders and facilitators; for example, school administrators, especially those who work with immigrant communities
- This website provides everything you need to run a digital security workshop (facilitator guides, participant workbooks, resource guides, etc.).
- From the workbook: "The Stronger NYC Communities (SCNYC) project was designed to advance the digital security capacities of community-based organizations that work directly with immigrant populations."

- The purpose of this resource is to "support organizations to implement organizational security."

Digital Defense Playbook from Our Data Bodies, available in English and Spanish

- **Intended Audience:** People connected to and working closely with families and the community
- A tool for communities to discuss and explore what online internet use, and subsequently data collection, means to them.
- This workbook, published as a zine by Our Data Bodies, contains popular education activities intended for nonprofits and community groups to use for reflection on data, surveillance, and community safety in their particular context. Through sharing these resources, the authors develop the argument that all groups (formal or informal) should be aware of and concerned about how data about the community is collected and used.

Go Deeper

Ready to explore further into the history of security attacks and preventive mechanisms? Here are two more videos from Carrie Anne Philbin's Crash Course Computer Science series:

- Watch the video "Hackers and Cyber Attacks: Crash Course Computer Science #32" online.
- Watch the video "Cryptography: Crash Course Computer Science #33" online.

Glossary

“If…Then,” “If…Then…Else,” and “If…Then…Else If” Statements

A Control Flow Statement that specifies the conditions under which one path or another should be taken.

Abstraction

The level of abstraction is the level of complexity to which an electronic system or computer program is created. At the lowest level of abstraction, every individual component is identifiable. As the level of abstraction increases, few details related to the system can be identified. At the highest level of abstraction, only the entire system is identifiable. Within programming, the ideal is to determine the most appropriate level of abstraction at which to incorporate the appropriate collection of blocks of code, within modules stored within libraries to facilitate more general-level writing of code.

Algorithm

A procedure developed to solve a problem or accomplish a task in a strategic way. While not exclusive to computer programming, it is often paired with data and information processing procedures.

Analog

Specific to electronics, an analog signal is any continuous electric pulse of varying amplitude. If we view a tone playing from a speaker as a sine wave, an increase in amplitude is equivalent to a louder sound from the speaker. An increase in the number of waves within a certain period of time is equivalent to a higher-pitched tone (perhaps from a middle C note to a middle E note). In these graphs of two different sine waves, the left sine wave plot shows increasing amplitude of the same tone while, the right sine wave plot shows a stable tone held at stable amplitude.

Anode

The positively charged electrodes conducting electric current from a cell into a device like a diode.

Artificial Intelligence (AI)

The use of computers not just to collect and store data, but to run algorithms which learn from data in order to make predictions and decisions about the data.

ASCII

The American Standard Code for Information Interchange (ASCII) uses 7 binary 0s and 1s (bits) to represent characters. The first 32 characters (00000000 to 00011111) are unprintable control codes. Codes 32-127 (00100000 to 01111111) are printable American English codes, including everything seen in this glossary term definition.

Audio/Video Interfaces

Computers incorporate a range of audio/video interfaces, depending on vendor. Some of the most common interfaces currently in use include:

High-Definition Multimedia Interface (**HDMI**), a proprietary audio/video interface standard, currently at version 2.X. This is included as the display port on the Raspberry Pi Model 3 and Model 4 microcomputers.

DisplayPort (DP) is a VESA standard digital display interface, and is backward compatible with other interfaces such as HDMI. Apple computers use the Apple-designed **Mini DisplayPort (mDP)**, although beginning in 2016, Apple began moving towards use of the USB-C connector.

Thunderbolt is the brand name of an interface standard developed by Intel and Apple. The standard allows the multiplex of individual data lanes from multiple connected devices to a single DisplayPort (DP), Mini DisplayPort (mDP) (Thunderbolt versions 1 and 2), or USB-C (Thunderbolt version 3) connector.

Basic Input Output System (BIOS)

Software, generally stored within its own memory chip on the systems board of a computer, which provides initial self-diagnosis of the computer before beginning the boot loading process. The original BIOS framework has mostly been replaced by the newer Unified Extensible Firmware Interface (UEFI), although the term BIOS is still sometimes used.

Binary Notation

Binary notation is based on the base-2 numeral system using only two symbols. The symbols used are the digits 0 and 1. Binary is the standard notation for Boolean true or false logic tests. This is in contrast to the base-10 numeral system commonly used in the English language, and which uses the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

Boolean Logic

A form of algebra used in mathematics, electronics, and computer hardware and software. At its base, all values are found to be either true or false using the Boolean operators "OR," "AND," and "NOT."

Breadboard

A breadboard used to be a board (sometimes literally a board for cutting bread) with nails pounded into it so that you could wrap wires around them to make experimental models of electric circuits. Today, the breadboard is a piece of plastic with holes in it. Underneath each hole is a metal clip. These metal clips connect together a specified set of holes ordered into a row or column. This way, pushing a piece of conductive material into one hole right away connects that material to things pushed into other holes that are joined together by that clip.

A **perfboard** is a thin, rigid sheet with holes but no metal clips on the other side. Instead, copper pads are used, to which conductors can be soldered. In some cases, as with the **perma-proto breadboards** from Adafruit, copper is further used to group together certain holes, mimicking the breadboard in a way that provides greater durability for prototyping work.

Bundled Software

Applications like file managers, user managers, and software managers, and applications like web browsers, text editors, photo and movie editors. Device and print drivers determine which hardware is "plug and play" ready.

Cable Internet

Redirects a cable channel to be used for Internet. Neighborhood shares bus topology. In the U.S., cable internet prioritizes download speeds.

Campus Area Network (CAN)

A version of a Metropolitan Area Network, the CAN is a network the size of a college, organization, or business campus. These types of networks are typically community-owned and/or -managed.

Cascading Style Sheets (CSS)

A style sheet language is used to format the expression of a structured document. While HTML itself provides a standard markup language for web page creation, alone, a web browser would make key decisions regarding the expression of the various tag elements within an HTML file.

The Cascading Style Sheets (CSS) is a modern style sheet language for use with HTML, XML, and other common markup languages used on the Internet today. Different relevant CSS files can be used to separate out content and presentation so that layout, colors, and fonts can be addressed independently, reducing complexity and repetition. This also allows distinct structures for different presentation formats such as web, e-book, and PDF displays of the same HTML or XML document.

Cathode

The negatively charged electrodes conducting electric current out of a device, like a diode, and back to a cell.

Cell-Based Internet

3G adds the EV-DO (Verizon, Sprint/Nextel) or HSDPA (AT&T, T-Mobile) protocol to cell voice's protocol. 4G adds the WiMax (Sprint) or LTE (Verizon, AT&T) standard to cell's voice protocol. Equivalent to bus (shared) topology. Prioritizes download speeds.

Central Processing Unit (CPU)

The CPU is the central processor of a computer. It is built of an ever-growing number of electronic circuits, enabling it to carry out the basic arithmetic, logic, controlling, and input/output operation instructions for the computer. Over the decades, more electronic components have been placed on a single and increasingly small integrated circuit (IC) chip. Today's CPUs are actually multi-core processors, with one chip being constructed of two or more CPUs called "cores." It is on the CPU that instructions for the vast majority of programming code is actually run. The CPU uses three main steps: fetch, decode, and execute, after which the instruction is returned to a memory integrated circuit.

Circuit

When working with electrical components, a circuit is the complete path that allows an electric current to flow from source voltage back to the source ground. Circuits generally include one or more electrical components along this path which are powered by the source.

A **closed circuit** is one in which current can flow fully from its source voltage to its ground return uninterrupted.

An **open circuit** is one in which there is an interruption in the flow of current from its source voltage to its ground return.

Client-Server Architecture

A computing structure which separates the work between a resource provider, or server, and a resource requestor, or client. The client-server model is distinct from centralized computing, in which servers reside in a central location or a set of regional locations. It is true that when there are many simultaneous client programs initiating requests, the server computer program(s) providing resources and services resides on centralized computer hardware called a server, specifically designed for computation, storage, and networked sharing. That is, the hardware called a server is just a computer specially designated to run one or more different applications designed to provide the server side of the client-server protocol.

Cloud Computing

Cloud computing is a form of client-server architecture, where a distributed network of data centers ensures the regular availability of computer storage and power, which is accessed over the Internet.

Cobbler

A Cobbler, used in electronics in combination with a prototyping board such as a breadboard, is a plastic header to which a ribbon cable can be attached, which can then be inserted into the prototyping board. A T-cobbler extension is sometimes included using a printed circuit board, allowing the header to which the ribbon cable is attached to be separated from the male pins that are inserted into the breadboard. Labels on all cobblers identifying each of the electrical circuits associated with each pin can then also be included on the printed circuit board.

Coding

Coding is the practical work of using a system of words, letters, figures, and other symbols to substitute for another set of words, letters, figures, and other symbols. Coding can include the creation of codes of practice and codes of ethics within a profession, codes of classification or identification, and codes of information and instruction within a technology. Decoding is the work of moving a code back to the original system of words, letters, figures, and other symbols. This is compared to programming, which is the creation of a planned series of events, items, or performances using or guiding the creation of codes. Coding is more context-oriented, while programming is oriented towards the bigger picture. DifferenceBetween.net has a helpful page providing further comparisons between coding and programming.

Combinational Logic

Boolean logic in which the output is based only on the present input.

Community Networks

Called the "public space in cyberspace" by Doug Schuler, a community network facilitates the sharing of public information between residents within a geographic area online. Community wireless networking uses standard wireless Ethernet (Wi-Fi) outdoors; anyone can use off-the-shelf equipment to create. Equivalent to bus (shared) topology, with synchronous upload and download speeds.

Computer

An electronic device using a program of instructions to collect, store, process, and transmit data. Many of our daily use devices, including automobiles, mobile phones, home routers connecting a building's network to the wider Internet, the growing number of "smart" internet-connected devices, such as watches; building heating, ventilation, and air conditioning (HVAC); light and sound systems; and our laptops and desktops, are all built using computers. While some use a more significant combination of integrated circuit processors and potentially multiple printed circuit boards, increasingly, these parts are more tightly integrated into a single printed circuit board and a reduced number of integrated circuits. Depending on design and use specifications and marketing, these may be referred to as **microcomputers**, **microcontrollers**, **microprocessors**, or **system on a chip (SoC)**. The Raspberry Pi is a general-purpose microcomputer with integrated system-on-chip central processor and other microprocessors.

Conditional

A command that determines under which specific coding pathway should be executed based on the occurrence of another sequence in the program.

Conductors

Electrical conductivity is the capacity to transmit electricity from one place to another. There are many different substances, such as copper wires and other bare metal pins, that are used to build electronic components. Examples of conductors include breadboards, jumper wires, printed circuit boards, and ribbon cables.

Control Flow Statement

A control flow statement provides a choice between two or more paths, and defines the specific sequence in which individual statements within that path should be executed or evaluated.

Controller

An integrated circuit that interfaces with one or more other devices on a printed circuit board. Besides the CPU, this can include input/output controller(s) and a graphic processing unit (GPU) controller. The Raspberry Pi Model 3 has two controllers, the system-on-chip (SoC) controller that includes the CPU and computer memory, and a controller interfacing with the wired Ethernet and four USB ports.

Current (A or Ampere)

Current is a flow of electrons from relatively positive points to relatively negative points, and is listed in amperes, or amps. Different electronics are capable of using different maximum currents, so it is sometimes necessary to provide resistance to reduce the current passing through the component.

Daemon

A computer program that runs in the background, providing services as needed. In this case, each time it is launched, the HTTP daemon starts itself based on the specifications within the configuration file, and then mostly hangs out, twiddling its thumbs and waiting for a call asking for something. When the call comes, it gets busy doing its stuff before going back to waiting mode. You'll often see a running HTTP server daemon listed as httpd.

Database

A **database** is a collection of organized data. A database uses a similar starting format to that of the rows and columns spreadsheet, this time calling them **records** and **fields**, respectively, which are stored within tables. But the vast majority of databases used today also add in a **relational** aspect through the use of a key which links tables together. And indeed, many of today's spreadsheets have implemented forms of linked cells across sheets as an elementary form of relational database.

To more formally make use of relational databases on a personal desktop, it's possible to move from something like Microsoft Excel to Microsoft Access, which includes a **Database Management System (DBMS)**. The DBMS is the software that interacts with end users, applications, and the database itself to capture and analyze the data. A **query language** is needed to allow communications within the database and from outside agents into the database. A commonly used one is the **Structured Query Language (SQL)**, often pronounced sequel. Together, the database, DBMS, and any associated applications make up a **database system**.

While many databases are proprietary, non-free software, some, like **MySQL**, include both a **GNU General Public License (GPL)** version and a **Proprietary** version. When MySQL was acquired from the founding Swedish company by Oracle, co-founder Michael Widenius launched a new fork of MySQL called **MariaDB**, a community-developed, commercially supported fork of MySQL, with GPL v2 license for most of the database system and a **GNU Lesser General Public License (LGPL)** for the client libraries.

Both MySQL and MariaDB make use of SQL and include standalone clients, allowing users to interact directly with the database itself. Both are highly compatible and intend to remain so. Both can be used as part of a **Linux, Apache, MySQL/MariaDB, or Perl/PHP/Python (LAMP)** base infrastructure to serve a range of database-driven web applications, including Drupal, Joomla, and WordPress, and websites including Facebook, Twitter, and YouTube.

Desktop Environment

The desktop environment is a collection of software that provides a predictable look and feel. This includes a Window Manager that controls the placement and appearance of windows, support icons, menus, etc. For some operating systems, such as Windows and macOS, the desktop environment is the branded look of the operating system and cannot be changed.

Digital

Specific to electronics, a digital signal is a representation of a physical quantity expressed as a series of the digits 0 and 1, that is, binary. The range of decimal numbers representing physical quantities in our work, for instance 12 seconds, is converted to a binary equivalent--in this case, 1100. In this illustration, the dots along the analog sine wave are data points collected to create the digital representation of the physical quantities.

Digital Subscriber Line (DSL)

Adds two channels to standard phone line for Internet. Hub and spoke (dedicated line) topology; full duplex. In the U.S., DSL prioritizes download speeds.

Diode

A semiconductor that passes current from one terminal to another terminal and in which current can only flow in one direction, known as rectification. Some common uses of diodes include reverse current protection, to clip or clamp circuits, and to provide logic gates. Another common usage is as a source for generating light, known as a **Light-Emitting Diode**, or **LED**. Different LEDs work at different wavelengths (the measure of distance between the peak and the trough in a wave), associated with different recognized colors of light. Some LEDs are made to be especially bright, such as a car headlamp made to help us see the road more clearly. Others are meant to be more diffuse, thereby working more as a source of information, like a car brake light or turn signal. Multiple light-emitting diodes can be packaged together in groups of three that include a red, a green, and a blue LED. These **RGB LEDs** can be further packaged together to create a full **LED matrix display**.

Domain Name System (DNS)

A naming system which translates domain names to IP addresses. This ensures a consistent name space for information resources.

Dynamic Host Configuration Protocol (DHCP)

As part of the Internet protocol suite, DHCP is a network management protocol. DHCP servers dynamically or statically assign IP addresses to connected nodes on the local area network (LAN) so that they can communicate with other IP networks.

Embedded Device

An integrated circuit, also known as a chip or a board, that brings together several components, such as a processor, memory, controllers, and input/output peripherals. It can then be embedded into a larger microcomputer or microcontroller system.

End-to-End Principle

According to this principle, network features should be implemented as close to the end nodes of the network as possible. Everything that can be done within the client or server application should be done there. Only those things that interconnecting devices must do should be done there.

Ethernet Port

An Ethernet port is an Institute of Electrical and Electronics Engineers (IEEE) standard connector for 100BASE-Tx/1000BASE-T full duplex Fast and Gigabit wired Ethernet communications, using twisted pair CAT-5 and CAT-6 copper cabling.

Federation of the Locals

The principle that networks begin and end with local area networks (LANs). The other types of area networks, including the wide area networks that comprise the Internet, then serve as a bridge between the local area networks.

Fiber Optics

Ultra-high-speed communications technology with one or more channels for Internet. Hub and spoke (dedicated) topology with synchronous upload and download speeds.

Fiber-to-the-Home (FTTH)/Fiber-to-the-Premises (FTTP)

When fiber-optic communications infrastructure reaches from the infrastructure (such as cable laid underground in the street) into privately owned property to reach an ISP subscriber's home or workspace.

File Manager

A file manager is a software program that helps you manage all the files on your computer. For example, file managers allow you to view, edit, copy, and delete the files on your computer's storage devices. It is known as the File Manager within the Windows Operating System and as the Finder on the Mac Operating System. The file manager makes use of the Graphical User Interface (GUI). The same tasks you can complete with the file manager can also be done through the Command Line Interface (CLI) terminal.

For Loop

A control flow statement that specifies the conditions under which a sequence of events should be executed.

Forever Loop

A control flow statement used to create a function that keeps running a sequence of code infinitely.

Function

A unit of code defined by its role within a more general code structure. To execute the function, it is provided one or more inputs, and produces a concrete result. This result may, but is not necessarily required to, be returned back to the caller of the function.

General Purpose Input/Output (GPIO)

A general purpose input/output, or GPIO, is a common, open-ended transmission mechanism used with integrated circuits on microcontrollers, microcomputers, and other printed circuit board electronics. Pins are provided to which female-ended wires can be connected. GPIO pins can be programmed as input (e.g., sensor data; ground) or output (e.g., state change; power) sources. GPIO pins can be preconfigured for special purposes, or can remain undefined until specified by a user at run time. In general, GPIO provides a means to tailor pins to fit specific design goals within applications, as well as reusability across applications.

Ground

In electrical engineering, ground, also called earth, provides a physical reference point in an electrical circuit from which voltage can be measured.

For electronic circuits, electrical ground, also called common, is the return path to a power supply. Today, most building 120- and 240-volt outlets have power, common, and ground. Many microelectronics make use of only power and common, while certain of our electronics also need earth-source ground in addition to power and common, or electrical ground.

Hypertext Markup Language (HTML)

The standard markup language for data, structured into documents, intended to be displayed on a web browser. Using a standard syntax, HTML annotates a document with a semantic structure. For example, wrapping a phrase in HTML tags denotes that those words are italicized when rendered in the web browser.

Hypertext Transfer Protocol (HTTP)

The ever-present application layer protocol used to request and access media (such as webpages) from across the Internet. Website URLs begin with “http” or “https” to signify this protocol (the added ‘s’ denotes the secured HTTP protocol, which is increasingly common). The Hypertext Transfer Protocol is part of the Internet protocol suite.

I2C

The Inter-Integrated-Circuit serial communications bus.

Unlike UART and SPI which provide send and receive communications to one or two attached devices, respectively, I2C is a bus and can work with device addresses to do serial send and receive communications with multiple devices.

Initialization

The assignment of an initial value for a variable.

Input/Output Devices

Input devices let users control the computer (e.g., keyboard, mouse, touch-sensitive devices). Output devices inform the user of what is happening (e.g., monitor, printer).

Integrated Circuit

Integrated circuits (ICs) are semiconductor wafers which contain a collection of tiny resistors, capacitors, and transistors. These can then be built to serve a wide range of electronic functions. In practice, larger-sized electronic components used to build circuits are first tested using materials like breadboards for rapid prototyping. They are then redesigned to be built into integrated circuits and optimized for regular, more standardized use. At times, a mix of electronic components along with integrated circuits are themselves used on breadboards to do further rapid prototyping of yet larger circuitry. The 5-Key Capacitive Touch Sensor included in the kit for this book contains a mix of integrated circuits. Examples of integrated circuits include processors, memory, and controllers.

Intelligent Agents/Smart Devices

Autonomous devices with software that facilitates the device's perception of its environment and execution of actions that maximize chances of achieving target goals.

Interconnect Device

A device used to connect nodes together. A switch or hub is used with wired Ethernet, an access point is used for wireless Ethernet (WiFi), and a router or gateway builds an Internet by connecting different LANs together.

Interdevice Internetworking

The facilitation of two or more devices working with each other.

Internet Protocol (IP)

A key communications protocol for routing data within networks, thus enabling the Internet. This protocol delivers packets from a sender to the destination and requires IP addresses for routing these packets. Domain Name Server (DNS) services are used to associate IP names with specific IP addresses.

Iteration

Repeating one step in a program multiple times.

Jumper Wires

Wires are made of either a thicker solid metal or thinner strands of multiple wires, placed within a non-conductive material. The exposed ends of the wire can then be inserted into two different holes on the breadboard to safely conduct current from one hole to another, helping to extend the circuit between different electrical components. These are sometimes attached to a plastic holder to provide greater strength. If a solid metal wire end has been soldered into the other side of that plastic holder, it is known as a **male end**. If a metal wire can be temporarily inserted into the other side of that plastic holder, it is known as a **female end**. If a pair of metal clips attached with springs is provided, it is known as an **alligator clip**.

Kernel

The heart of the operating system. It is the go-between from applications between input/output controllers, memory, the central processing unit (CPU), and storage devices.

Link Lights

LED light found on wired Ethernet cards and switches. When lit, the lights indicate that the card and switch are on and working properly and that the Ethernet cable is plugged in and working properly.

Lists and Arrays

Lists and arrays are two commonly used ways to store data. A list is a set of data arranged in some order. Lists are mutable, meaning that the order of the items within the list can change. This allows for sorting, shuffling, adding, and deleting items from the list. One form of lists, called an array, is an indexed set of related elements associating one thing with another; these are especially useful in control flow loops, such as button counters. However, insertion or deletion of items within an array's list can prove difficult.

Local Area Network (LAN)

The simplest type of Internet-based area networks is a **local area network (LAN)**. A LAN is a network with connected devices in a close geographical range. It is generally owned, managed, and used by people in a building. For example, connecting to a Wi-Fi network at a coffee shop or library would mean that your device would be a node on the coffee shop or library's publicly accessible LAN. Many businesses and institutions have a second, private LAN for use by staff only.

Machine Learning

A branch of artificial intelligence (AI) in which algorithms include "teaching" models that continually improve machines' predictions when fed data.

Media

Media are used to interconnect devices on a network, and are made of four primary materials: coaxial copper cable, twisted pair copper cable, fiber optics cable, and radio waves.

Media Access Control (MAC)

Media access control (MAC) addresses are unique identifiers assigned to network interface controllers (NICs) as part of the data link layer of the OSI model.

Memory

Memory is the part of a computer in which data or programming instructions are stored and retrieved. Memory is constructed using high-performance integrated circuit chips to increase performance, as compared to storage devices such as hard drives and lower-performance but less expensive flash drive integrated circuit chips. Much of this type of memory is volatile **random-access memory (RAM)**, in which data is lost when power to the computer is turned off. There are multiple types of RAM: The higher-cost but faster-performing **static RAM (SRAM)**, is used with the CPU to increase CPU performance, while the less costly but slightly slower **dynamic RAM (DRAM)**, or more recently **double data rate synchronous dynamic RAM (DDR SDRAM)** is the main memory of computers. The Raspberry Pi Model 3 comes with 1 GB (Gigabyte) of DDR, while the Raspberry Pi Model 4 is available with 1GB, 2GB, or 4GB of DDR, making it very similar to today's smartphones and Chromebook computers. Many laptops today come with 4GB to 8GB of system memory.

Metropolitan Area Network (MAN)

A collection of LANs and devices in an area the size of a city.

Modem

A portmanteau of the words Modulator-Demodulator. This device converts data provided from computing devices into binary zeros and ones so that it can be transmitted through a network.

Modularization

Including blocks of code over and over into a program, or having a very long list of code in a single instruction set, can become confusing and hard to debug. A module is a block of code, separated out into a manageable chunk. Functions and subroutines are examples of modularization.

Motherboard

The printed circuit board found in a computer, which contains the principal components, along with ports and connectors to other devices. The term "motherboard" arose at a time when it was paired with one or more supporting printed circuit expansion boards, often referred to as **daughterboards** or **expansion cards**, in order to achieve the full general functioning of the computer. The motherboard is sometimes also referred to as the **system board** or the **mainboard**.

Network Interface Card (NIC)

The hardware necessary for a node to connect to a network. For example, an Ethernet card (wired or wireless) is used for a LAN connection. A modem (cable, DSL, dialup) is used for traditional Internet. Optical network terminals (ONTs) are used for fiber to the home.

Network Manager

Operating system utility that shows network configuration. An IP address of 0.0.0.0 or one that starts with 169.254 indicates that you are connected to the media, but aren't getting a proper IP address from the address server (usually the gateway router).

Node

Any device directly connected to the network that has been assigned a unique identifier or address on that network, such as a MAC address (also known as the hardware, physical, or Ethernet address), the serial number for Ethernet cards, or an IP address (the address used by the Internet protocol).

Ohm's Law

German scientist Georg Simon Ohm developed this simple, linear mathematical principle relating current (I), resistance (R), and voltage (V). There are three equations that are used in direct current circuits. If needed, these can help us to determine, for instance, which voltage source or which resistor ohms we should use to achieve a certain passage of current through our circuit.

$V = IR$ … Electrical voltage equals amperes times ohms
$I = V/R$ … Current amperes equals voltage divided by ohms
$R = V/I$ … Resistance ohms equals voltage divided by amperes

Open Systems Interconnection (OSI) Model

The Open Systems Interconnection (OSI) model expresses the relationship between the seven layers of computer and telecommunications systems. These layers are:

1. Physical: Copper & fiber cables, radio frequency & infrared wireless.
2. Data link: Media Access Control (MAC).
3. Network: Routing of data packets.
4. Transport: End-to-end connections.
5. Session: Interhost communication.
6. Presentation: Data translation & representation.
7. Application: What the 'end user' sees.

Operating System (OS)

The core software bundle of a computer. The OS supports different aspects of a computer's basic functions, overseeing the coordination between the physical electronics of the computer and the many software applications that allow us to do our daily computer-based activities. While the physical electronic components of most personal computers support multiple different operating systems, only one OS can be run on a computer at a given time, since it is the central control system of those physical electronics. That is, the OS has the final word on how software interacts with hardware when the computer is turned on and functioning.

The **kernel** is the heart of the OS, and for the most part remains hidden from view. On the opposite side, the **graphical user interface (GUI)** serves as the visual mechanism for interacting with a computer, whether via keyboard, mouse, and monitor, touchpad, or something else. All modern computers also include a **command-line interface (CLI)**, such as a terminal window in which applications can be run and data is displayed using text. Indeed, for higher-level programming, research and development, and systems management, a combination of GUI and CLI interfaces used in parallel prove essential.

Packets

When information and communications data sent is sent from the node and onto the LAN and often to the WAN, it is not sent as a single file, such as that we store on a hard drive or flash drive. Rather, it is divided into pieces, called packets. Each packet contains a header and a payload. The header contains information including the source IP address, the destination IP address, and an identification tag to order the packets. The payload is the actual information that is being transmitted, such as web page data or an email. When a packet reaches its destination, the software application will piece all the packets together.

Pair/Triplet Programming

Pair programming (or triplet programming) is common in software development. Two (or three) programmers collaborate on design, coding, and testing, with qualitative evidence suggesting the subsequent design is better, resulting in simpler code that is easier to extend. Further, whether the pair programming occurs between two novice programmers, between a novice programmer and a more experienced programmer, or between two experienced programmers, people learn significantly more about the system and about software development, as both participants bring unique insights. Conversation between the programming pair can occur at many levels as the driver working at the keyboard takes charge of all changes made in the program and the navigator observes all the code that is entered, considers coding options, works to spot and address problems, considers and recommends simplifications, helps with programming style, and designs and verifies testing.

Parallel Circuit

A complete, closed circuit in which current divides into multiple paths to ground. A failure on one path does not impact electronic components passing along another path running in parallel.

Peer-to-Peer Architecture

A computing structure in which individual devices can share information and resources directly without relying on a dedicated central server. Each device can perform a mix of resource provision (e.g., computation, storage, and networked sharing) and resource requesting (e.g., web page, new email or social media posts) tasks. Compare to client-server architecture, in which the client device performs requests while a server device performs resource provision.

Personal Area Networks (PAN)

Personal area networks (PANs) provide a simple computer network organized around a few personal devices, allowing the transfer of files, photos, and music without the use of the Internet or your home's local network. Two common examples would be your Bluetooth headset or your keyboard and mouse. Depending on the Bluetooth range selected (or chosen for you), this could span 3 feet, 10 feet, or 100 feet. Beyond Bluetooth, other common PAN connectivity includes Infrared (IR), USB, ZigBee, Wi-Fi, and radio frequency (RF, including short-distance AM and FM radio).

Ping

Run from the command line or terminal application to test if you can communicate with another node.

Power Adapter

Our homes, businesses, libraries, and other community spaces typically include wall outlets from which we can plug in our electronics. These are usually either installed to supply 120-volt or 240-volt alternating current (AC) power source to our devices. But many of our devices use a much lower-voltage direct current (DC) power source. It is for this reason that we often need a power adapter that has a plug-in to connect to the wall outlet, and another that has another form of plugging, increasingly today a micro USB connector, to your electronic device. For the kit used in this book, we use a power adapter that works with 120-to-240-volt AC power on one side, and provides 5-volt, 2.4-amp power via a micro USB connector on the other side.

Printed Circuit Board

A printed circuit board is a board base made of fiberglass or glass-reinforced plastic, with one or more layers of copper or other conductive traces placed on one or both sides of the board base. If you've ever tie-dyed a shirt, you've got a good starting point for considering the printing process of a circuit board. Individual electronic components and integrated circuits are then soldered onto specific trace paths to build complex electronics. A small printed circuit board was used to build the 5-Key Capacitive Touch Sensor used in our kit.

Program

A **program** is software containing a series of coded instructions to control the operation of an electronic device (noun), and the activity of creating coded instructions in support of a particular task of an electronic device (verb). Programs are created using a **programming language** that includes rules and a system of symbols. The language must conform to these rules of syntax and semantics, but unlike many commonly used physical electronics, they often are not created as part of a standards body. A programming language may be developed from scratch, but more often they are built within a programming family, using previous programming languages as a base starting point, and may rely on another programming language to effectively function on a machine. This book explores several popular programming languages:

MakeCode: A web-based code editor developed by Microsoft, adapted specifically for the Circuit Playground Express in collaboration with Adafruit. It provides a block editor similar to the Scratch visual programming language developed by MIT Media Lab Lifelong Kindergarten Group. (Scratch is installed by default in the Raspberry Pi OS and with the available extension can be used with the Raspberry Pi GPIO.) The MakeCode programming language itself is TypeScript, a high-level programming language superset of another language, **JavaScript**. MakeCode is edited in a web browser at https://makecode.adafruit.com/ and provides a simulator of a Circuit Playground Express. Programs can be downloaded and flashed, that is, written to the read-only memory (ROM) of the Circuit Playground Express, overwriting the previous MakeCode program on the physical device.

Python: A popular general-purpose programming language with a relatively easy-to-use syntax. A range of integrated development environments (IDEs) are available in which code is written, tested, and debugged. The Raspberry Pi comes with IDLE and Thonny, two commonly used Python IDE bundles.

PHP: Hypertext Preprocessor: A scripting language used with web server applications in support of web development, as well as a general-purpose programming language.

SQL (Structured Query Language): Used within database management system servers, like the open-source relational database management system MySQL, which can be installed and used in the Raspberry Pi OS.

Programmable Circuit

An electronic component with an undefined function, allowing it to be programmed and used in reconfigurable ways.

Prototype

The first, preliminary model of a circuit or a system of circuits used when developing a product. Used both as a noun describing a model and as a verb describing the activity moving towards the creation of the model.

Request for Comments (RFC)

RFCs are technical and organizational notes about the Internet and cover many aspects of computer networking, including protocols, procedures, programs, and concepts.

Resistance (Ω or Ohm)

Electrical **resistance** reduces the flow of current through a circuit. **Resistors** are the typical electrical components used to provide resistance in a circuit, and are listed in ohms. For instance, the exercises in the book mainly use 470- or 560-ohm resistors, also abbreviated as 470 Ω or 560 Ω resistors.

Ribbon Cable

When a number of insulated wires are brought together to create a flat ribbon of wires, this is called a ribbon cable. Ribbon cables may have a plastic box with solid metal pins extending out from the other side, in which case the box is known as a male header. Other times, a plastic box is made with holes into which metal pins can be inserted, in which case it is known as a female header.

Router

A router is an interconnect device used to transfer data from one local area network (LAN) to another LAN connected to the router.

Schematic

A symbolic and simplified diagram or other representation of a circuit. Throughout this book, schematics are used when an illustration of a circuit is needed without specifying exactly how these would be physically built using a breadboard or other prototyping platform. In this schematic illustration, we see the formal representation of the electronics used to create a complete and functioning circuit that include a 560 ohm resistor, a 5mm LED circuit, and a battery. In the prototype illustration, we see one example of how this circuit could be constructed using a tiny breadboard and a double-A battery.

Semiconductor

A substance that can conduct electricity under some conditions, but not under others. Many electrical components are built using such semi-conductive materials, including diodes, sensors, and integrated circuits.

Sensor

Electrical sensors are devices used to detect input from the physical environment. Light sensors might automatically turn on the headlights of your car, while heat and chemical sensors might turn on a fire alarm. And motion sensors might turn up the heating or cooling in a house, or might count how many people have passed through a gate to and from a library. The Circuit Playground Express that comes with the toolkit has a range of sensor inputs within this printed circuit board, including:

Analog light sensor to detect ambient light, with similar spectral response to the human eye.
Temperature sensor calculated using the analog voltage at a given moment.
Microphone audio sensor, which is a digital microphone that works similarly to the microphones built into our laptops and smartphones, but is smaller and less expensive, well-suited for this educational microcontroller.
Motion sensor using an accelerometer, detecting both motion and gravitational pull.
Seven capacitive touch sensors that can sense the touch of a person, human or otherwise.

Sequence

A particular set of events, and the specific ordering of those events, to complete one step in a program.

Sequential Logic

Boolean logic in which the output depends on the present input as well as a set history of the input.

Series Circuit

A complete, closed circuit in which there is one path along which current flows. When one electronic component along the path fails or is interrupted in some way, all components enter an open state and stop working.

Simple Mail Transfer Protocol (SMTP)

As part of the Internet protocol suite, SMTP is a communications protocol for email servers and clients. SMTP is used to send and receive email.

Sociotechnical

Social and technical aspects of devices and systems are not two separate side-by-side items, but different interdependent aspects of the sociotechnical whole that have emergent properties beyond the sum of their parts. Sociotechnical information systems include a range of hardware, software, and networking technical layers, as well as individual and group social layers.

Specifications

Often called "specs," specifications are formalized practices created using accredited technical standards developed and adopted using an open consensus process under guidelines of a standards body, or using de facto technical standards developed and owned by a single group or company.

Statement

A statement is a single instruction given to the computer based on the rules of syntax and semantics of a programming language. The statement may include both the specific command and clarifying details, such as the color all pixels should be set to.

Storage (Optical, Magnetic, and Flash)

There are a range of different **data storage devices**: hard drives (HDDs) constructed using magnetic charges placed on a set of spinning platters; DVDs using pits on a spinning platter read via laser; and flash memory integrated circuit chips, such as solid-state drives (SSDs), USB flash drives, and SD cards, which provide computers with long-term storage of data and software applications. Storage devices are relatively slow, so programming instructions need to be moved from storage to memory using integrated circuits to be fetched, decoded, and executed by the processors of the computer. The Raspberry Pi kit includes an 8 GB MicroSD card storage device, although 16GB, 32GB, and 128GB MicroSD cards are available and can be used with the Raspberry Pi. In addition, it is sometimes helpful to make use of USB flash drives or USB hard drives with the Raspberry Pi to provide even more storage.

Store-and-Forward Principle

A telecommunications technique in which information is stored at each intermediate node on the path to a specified destination.

Structured Query Language (SQL)

A language for creating, deleting, and manipulating data in a relational database. Although SQL is a standard language, many vendors, such as MySQL, have their own “dialect.” Relational database management systems are often a key component within the “client-server” architecture.

Subroutine

A unit of code defined by its role within a more general code structure. A subroutine does not need an input and does not provide an output, but simply executes the sequence of code within it.

Switch

Electronic switches are used to control the flow of current on a circuit. They can be used to switch between the closed position, in which a current continues its flow through a circuit, and the open position, in which the current is not passed through. A classic example of this type is a light switch, in which a closed position would turn on a light and an open position would turn off that light. Closing the switch completes the circuit. Other switches are used to control the amount of current that flows across the circuit. A classic example of this type is a dimmer switch for a light used to brighten or dim the brightness of a light.

While the above are common mechanical switches, it is also possible to use programming code to switch the flow of current along a circuit. A common method for doing this is through use of a transistor, which uses a low current signal to one leg of the transistor provided by the program to determine the flow of a larger amount of current through the transistors other two legs.

Themes and Skins

Pre-set packages containing graphical appearance details. Supported in some versions of Windows and Linux. Available as third-party applications in other cases. Supported for many applications, like web browsers. Changes the look and feel of many features at once (e.g., background colors, text font and size, icons, mouse cursor, etc.).

Traceroute

Run from the command line or terminal application to test the performance of each router between nodes.

Transistor

Small electronic switches allowing control of electrical flow within a circuit without using programming code.

Transmission Control Protocol (TCP)

Together with the Internet Protocol (IP), TCP is one of the primary protocols of the Internet protocol suite. TCP specifically defines how data intended to be sent over a network is broken into packets before transmission and reliably reassembled in the right order at the destination application. Many applications, including the web, email, and file transfer, rely on TCP as a foundation.

Uniform Resource Locator (URL)

Also known as a web address, the URL is a commonly used form of the more general Uniform Resource Indicator (URI), strings of characters used to unambiguously identify a resource, such as "https://www.wikipedia.org." The first part of the URL states the protocol to be used. While web browsers support several different protocols, we almost always make use of the HTTP protocol or the secure Hypertext Transfer Protocol, HTTPS.

Universal Asynchronous Receive and Transmit (UART)

Pronounced "u-art," the universal asynchronous receiver-transmitter is a commonly used hardware supporting serial communication between two devices, such as between a microcomputer and a microcontroller. As an asynchronous device, UART data can be sent at different periods of time as needed. This contrasts with synchronous data communications technologies such as the dynamic random access memory (RAM) found in dual in-line memory modules (DIMM) of a computer, in which data is exchanged at precise intervals. "Universal" indicates the configurability of UART, allowing for unique code to be written supporting different data types and sizes to meet specific data communication needs. An independent driver circuit manages transmission levels (e.g., 115,200 kilobits per second speed) and methods (e.g., USB to TTL) used within a specific communication cable.

USB

Universal Serial Bus (**USB**) is a formal industry standard with specifications for cables and connectors, and also protocols for connections, communication, and power supply between electronic devices such as computers and peripherals. USB 1.x, 2.0, 3.x, and the upcoming 4 are generations of the standard and associated specifications. The USB port connector on a device is called the receptacle, while the connector on the cable is called the plug. There is a range of different USB connectors approved within USB committee specifications, including:

Standard format **USB Type-A** (common on computers, and used on the Raspberry Pi) and Type-B connectors (seen in a range of removable devices, such as external hard drives and printers). Type-A receptacles on devices are color-coded white, indicating USB version 1 or 2 compliant low/full/high speed ports, or blue, indicating USB version 3.x SuperSpeed compliant ports.
Mini-A (common in cameras and some tablet computers) and Mini-B connectors.
Micro-A and **Micro-B** connectors, common in many mobile devices. For USB 3.x ports, a thin but wider **Micro-B SuperSpeed** plug is used.
USB-C is a formal connector specification, and does not necessarily implement USB 3.x standards.

User Datagram Protocol (UDP)

Used together with the Internet Protocol (IP), UDP is an alternative to TCP when speed is essential and error-correction is not. For example, live video streaming applications use the UDP protocol, acknowledging that video may occasionally freeze or appear jumpy for brief periods.

UTF-8

UTF-8, the 8-bit Unicode Transformation Format encoding system, is the dominant encoding for the World Wide Web. It is backward compatible with ASCII, but provides support for a wide range of languages beyond American Standard encoding of ASCII.

Variables

A convenient name to represent numbers that change from time to time. Look at the top of your smartphone, and you may see the battery percentage charge remaining. A program exists on the smartphone to read this variable and provide a visual representation of it. Variables are often used within code to evaluate changes in patterns to determine which series of code should be executed at a given moment.

Voltage (V or Volt)

Voltage is a quantitative expression of the electromotive force required for a charge to pass between two points in an electrical field. Common household voltages include 120- and 240-volt circuits. Common computer and other microelectronic voltages include 12-, 5-, and 3.3-volt circuits. Electrical components are designed for specific voltages and need power adapters if the supply of energy does not meet the component requirements.

Watts

A unit of power used to quantify the rate of energy transfer.

While Loop

A control flow statement that specifies the conditions under which a sequence of events should continue to be executed.

Wide Area Network (WAN)

Covers the size of a state, country, and could even be considered to include the entire Internet. A WAN is comprised of many interconnected MANs and LANs. The WAN that is used to internetwork these is typically owned and managed by one or more Internet service providers (ISPs: the business that provides connections to each LAN), network service providers (NSPs: the business(es) that provide connections between ISPs), and backbone providers (the business(es) that provide the more extended connections between NSPs).

Window System

This is the part of the graphical user interface that communicates with the kernel. Many operating systems allow remote interfacing with the window system, either directly or through third-party applications.

Bibliography

Adafruit Industries. "About Us." Adafruit. Accessed February 9, 2020. https://www.adafruit.com/about.

Adafruit Industries. "Adafruit Learning System." Adafruit Learning System. Accessed February 9, 2020. https://learn.adafruit.com/.

Adafruit Industries. "Adafruit LED Sequins – Multicolor Pack of 5." Adafruit. Accessed February 9, 2020. https://www.adafruit.com/product/3377.

Adafruit Industries. "CircuitPython with Jupyter Notebooks." Adafruit Learning System. Accessed July 21, 2020. https://learn.adafruit.com/circuitpython-with-jupyter-notebooks/overview.

Adafruit Industries. "Code.Org Circuit Playground Individual Kit Pack." Adafruit. Accessed February 9, 2020. https://www.adafruit.com/product/3795.

Adafruit Industries. "Raspberry Pi 3 Model B Starter Pack – Includes a Raspberry Pi 3." Adafruit. Accessed February 9, 2020. https://www.adafruit.com/product/3058.

Adams, Genetta M. "17 Black Internet Pioneers." The Root, February 12, 2012. https://www.theroot.com/17-black-internet-pioneers-1790868134.

Albritton, Richard. "Set up Home Assistant with a Raspberry Pi." Adafruit Learning System. Accessed July 21, 2020. https://learn.adafruit.com/set-up-home-assistant-with-a-raspberry-pi/intro.

American Library Association. "Libraries and the Internet Toolkit." May 29, 2007. http://www.ala.org/advocacy/intfreedom/resources.

American Library Association. "Resources for Librarians about Online Social Networking." September 29, 2006. http://www.ala.org/yalsa/guidelines/socialnetworking.

An, Junghyun. "Service Learning in Postsecondary Technology Education: Educational Promises and Challenges in Student Values Development," University of Illinois at Urbana-Champaign, 2008. https://hdl.handle.net/2142/17387.

Apache. "Apache Server Frequently Asked Questions," January 6, 1997. Accessible from Internet Archive Wayback Machine. https://web.archive.org/web/19970106233141/http://www.apache.org/docs/misc/FAQ.html#relate.

Arduino. "What Is Arduino?" Accessed February 9, 2020. https://www.arduino.cc/en/Guide/Introduction.

Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, et al. "A View of Cloud Computing." *Communications of the ACM* 53, no. 4 (April 2010): 50–58. https://doi.org/10.1145/1721654.1721672.

Ashton, Kevin. "That 'Internet of Things' Thing," *RFID Journal*, June 22, 2009, https://www.rfidjournal.com/that-internet-of-things-thing.

Audacity. "Audacity." Accessed February 9, 2020. https://www.audacityteam.org/.

Barbrook, Richard, and Andy Cameron. "The Californian Ideology." *Science as Culture* 6, no. 1 (January 1, 1996): 44–72. https://doi.org/10.1080/09505439609526455.

Barela, Anne. "Arduino to CircuitPython." Adafruit Learning System, October 22, 2018. https://learn.adafruit.com/arduino-to-circuitpython.

Bate, Alex. "Electronics 101.1: Electricity Basics." Raspberry Pi, October 16, 2018. https://www.raspberrypi.org/blog/electronics-101-1-electricity-basics/.

Biography.com. "Black Inventors." Accessed August 3, 2023. https://www.biography.com/black-inventors.

Bishop, Matt. *Introduction to Computer Security*. 1st Edition. Addison-Wesley Professional, 2004.

Black Enterprise. "List of the Most Influential Blacks in Technology." November 9, 2018. https://www.blackenterprise.com/list-of-the-most-influential-blacks-in-technology/.

Bray, Devon. "PiPlanter 2 – Esologic." Blog. PiPlanter 2. Accessed February 9, 2020. http://www.esologic.com/piplanter-2/.

Britannica, The Editors of Encyclopaedia. "Turing test". Encyclopedia Britannica, 13 Mar. 2023, https://www.britannica.com/technology/Turing-test. Accessed 13 April 2023.

brown, adrienne maree. *Emergent Strategy: Shaping Change, Changing Worlds*. AK Press, 2017.

Bruce, Bertram C. "Technology as Social Practice." *Educational Foundations* 10, no. 4 (1996): 51–58.

Bruce, Bertram C. "What Is Community Inquiry?" Blog. Chip's journey. Accessed February 9, 2020. https://chipbruce.net/resources/community-inquiry-bibliography/what-is-community-inquiry/.

Bruce, Bertram C. *Beyond the Classroom Walls: Imagining the Future of Education, from Community Schools to Communiversities.* Rowman & Littlefield, 2022

Bruce, Bertram C., and Ann P. Bishop. "New Literacies and Community Inquiry." In *The Handbook of Research in New Literacies*, edited by J. Coiro, C. Knobel, C. Lankshear, and D. Leu, 699–742. New York: Routledge, 2008. http://hdl.handle.net/2142/15133.

Bruce, Bertram C., Andee Rubin, and Junghyun An. "Situated Evaluation of Socio-Technical Systems." In *Handbook of Research on Socio-Technical Design and Social Networking Systems*, edited by Brian Whitworth and Aldo de Moor, 2:685–98. Information Science Reference. Hershey, PA: IGI Global, 2009. https://doi.org/10.4018/978-1-60566-264-0.ch045.

Bruce, Bertram C., Ann Person Bishop, and Nama R. Budhathoki, eds. *Youth Community Inquiry: New Media for Community and Personal Growth*. New Literacies and Digital Epistemologies. New York: Peter Lang, 2014. https://doi.org/10.3726/978-1-4539-1201-0.

Budiu, Raluca. "Can Users Control and Understand a UI Driven by Machine Learning?" Nielsen Norman Group, December 16, 2018. https://www.nngroup.com/articles/machine-learning-ux/.

Bull, Glen, and Joe Garofalo. "Internet Access: The Last Mile." *Learning & Leading with Technology* 32, no. 1 (2004).

Burgess, Matt. "What Is the Internet of Things? WIRED Explains." *Wired UK*, February 16, 2018. https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot.

Calabrese Barton, Angela, and Edna Tan. *STEM-Rich Maker Learning: Designing for Equity with Youth of Color*. New York: Teachers College Press, 2018.

CAST. "About Universal Design for Learning," 2020. http://www.cast.org/our-work/about-udl.html.

Caswell, Michelle. Seeing yourself in history: Community archives and the fight against symbolic annihilation. The Public Historian, 36, no. 4 (2014): 26–37. https://doi.org/10.1525/tph.2014.36.4.26.

Caswell, M., Marika Cifor, and Mario H. Ramirez. "To suddenly discover yourself existing": Uncovering the impact of community archives. The American Archivist, 79, no. 1 (2016): 56–81. https://doi.org/10.17723/0360-9081.79.1.56.

Cecez-Kecmanovic, Dubravka, and Mary Anne Kennan. "The Methodological Landscape." In *Research Methods: Information, Systems, and Contexts*, edited by Kirsty Williamson and Graeme Johanson, Second Edition., 127–55, 2018. https://doi.org/10.1016/B978-0-08-102220-7.00005-4.

Center for Technology in Learning, and SRI International. "Resources." Principled Assessment of Computational Thinking. Accessed June 5, 2020. https://pact.sri.com/resources.html.

Chambers, Drew W. "Is Freire Incoherent? Reconciling Directiveness and Dialogue in Freirean Pedagogy." *Journal of Philosophy of Education* 53, no. 1 (February 2019): 21–47. https://doi.org/10.1111/1467-9752.12340.

Clarke, Rachel Ivy. *Design Thinking*. Library Futures 4. Chicago: ALA Neal-Schuman, 2020.

Clinton, William J. and Albert Gore, Jr., "Technology for America's Economic Growth, A New Direction to Build Economic Strength," ED 355 929, February 22, 1993. https://files.eric.ed.gov/fulltext/ED355929.pdf.

Clinton, William J. "Technology for America's Economic Growth: A New Direction to Build Economic Strength." *The Bottom Line* 6, no. 3/4 (March 1993): 4–17. https://doi.org/10.1108/eb025379.

Cloudflare. "About Us." Accessed July 21, 2020. https://www.cloudflare.com/about-overview/.

Common Sense. "23 Great Lesson Plans for Internet Safety," February 7, 2020. https://www.commonsense.org/education/articles/23-great-lesson-plans-for-internet-safety.

Commotion Wireless. "About Commotion." Accessed February 9, 2020. https://www.commotionwireless.net/about/.

Commotion Wireless. "Commotion Contruction Kit – Planning." Accessed July 22, 2020. https://commotionwireless.net/docs/cck/planning/.

Commotion Wireless. "Where It's Used." Accessed July 22, 2020. https://commotionwireless.net/about/where-its-used/.

CommunitySense. "CommunitySense – For Working Communities." Accessed July 22, 2020. https://www.communitysense.nl/.

Comparitech. "Protecting Children's Privacy Online – a Guide for Parents," November 19, 2018. https://www.comparitech.com/blog/vpn-privacy/protecting-childrens-privacy/.

Cooke, Nicole. "Becoming New Storytellers: Counterstorytelling in LIS." In *Information Services to Diverse Populations: Developing Culturally Competent Library Professionals*, 113–36. Santa Barbara, CA: Libraries Unlimited, 2017.

Costanza-Chock, Sasha. *Design Justice: Community-Led Practices to Build the Worlds We Need*. Information Policy. Cambridge, MA: The MIT Press, 2020. https://designjustice.mitpress.mit.edu/.

Crane, Beverly. "Digital Storytelling Changes the Way We Write Stories." *Information Searcher* 18, no. 1 (2008): 3–9.

Crawford, Susan P. *Captive Audience: The Telecom Industry and Monopoly Power in the New Gilded Age*. New Haven: Yale University Press, 2013.

Crawford, Susan P. *Fiber: The Coming Tech Revolution and Why America Might Miss It*. New Haven: Yale University Press, 2018.

Creative Technology Research Lab, "Project TACTIC," accessed August 3, 2023, https://ctrl.education.ufl.edu/projects/tactic/.

C-SPAN. "Renewing American Civilization." Accessed July 17, 2020. https://www.c-span.org/video/?50261-1/renewing-american-civilization.

Cumbula, Salomao David, Amalia Sabiescu, and Lorenzo Cantoni. "Community Design: A Collaborative Approach for Social Integration." *The Journal of Community Informatics* 13, no. 1 (March 22, 2017). https://doi.org/10.15353/joci.v13i1.3299.

Dalton, Jeff. "Pair Programming." In *Great Big Agile*, by Jeff Dalton, 199–200. Berkeley, CA: Apress, 2019. https://doi.org/10.1007/978-1-4842-4206-3_42.

de la Peña, Carolyn. "The History of Technology, the Resistance of Archives, and the Whiteness of Race." *Technology and Culture* 51, no. 4 (October 2010): 919–37.

Delgado, Richard and Jean Stefancic. *Critical race theory: An introduction.* New York University press, 2023.

de Moor, Aldo. "Citizen Sensing Communities: From Individual Empowerment To Collective Impact," 91–101. Monash Centre, Prato, Italy, 2019. https://www.monash.edu/__data/assets/pdf_file/0004/2219800/CIRN2019-complete.pdf.

de Moor, Aldo. "Increasing the Collective Impact of Climate Action with Participatory Community Network Mapping." *Livingmaps Review* no. 8 (April 29, 2020). http://livingmaps.review/journal/index.php/LMR/article/view/197.

de Moor, Aldo. Common Sense. "Common Sense Education." Accessed June 10, 2020. https://www.common-sense.org/education/.

Design Kit. "What Is Human-Centered Design?" Accessed July 17, 2020. https://www.designkit.org/human-centered-design.html.

Dewey, John. *Experience and Education.* 1. ed. The Kappa Delta Pi Lecture Series. New York: Simon & Schuster, 1997.

DigiKey Electronics. "4 Band Resistor Color Code Calculator and Chart." Accessed February 13, 2020. https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band.

Drotner, Kirsten. "Boundaries and Bridges: Digital Storytelling in Education Studies and Media Studies." In *Digital Storytelling, Mediatized Stories: Self-Representations in New Media*, edited by Knut Lundby, 61–84. New York: Peter Lang, 2008.

Dweck, Carol. "The Power of Believing That You Can Improve." YouTube, December 17, 2014. https://www.youtube.com/watch?v=_X0mgOOSpLU&feature=emb_title.

Dyson, Esther, George Gilder, George Keyworth, and Alvin Toffler. "Cyberspace and the American Dream: A Magna Carta for the Knowledge Age." *Future Insight* 1, no. 2 (August 1994). http://www.pff.org/issues-pubs/futureinsights/fi1.2magnacarta.html.

Electronic Frontier Foundation. “Surveillance Self-Defense.” Surveillance Self-Defense. Accessed June 10, 2020. https://ssd.eff.org/en.

ElectronicsWeekly.com. “How They Make Raspberry Pi in the UK – Electronics Weekly Visits Pencoed. Metropolis Multimedia.” YouTube, September 11, 2013. https://www.youtube.com/watch?v=Tza6Hl8wSJ0&feature=emb_title.

Epilepsy Foundation. “About Epilepsy: The Basics.” Accessed July 17, 2020. https://www.epilepsy.com/learn/about-epilepsy-basics.

Eubanks, Virginia. “Popular Technology: Exploring Inequality in the Information Economy.” *Science and Public Policy* 34, no. 2 (March 1, 2007): 127–38. https://doi.org/10.3152/030234207X193592.

Eubanks, Virginia. “Trapped in the Digital Divide: The Distributive Paradigm in Community Informatics.” The Journal of Community Informatics, 3, no. 2 (2007). https://doi.org/10.15353/joci.v3i2.2373.

Eubanks, Virginia. *Digital Dead End: Fighting for Social Justice in the Information Age*. Cambridge, Mass: MIT Press, 2011.

Feather, Katie. “Attack of the Internet of Things.” 10/28/2016. *Science Friday* (blog). Accessed February 17, 2020. https://www.sciencefriday.com/segments/attack-of-the-internet-of-things/.

Federal Networking Council, “FNC Resolution: Definition of ‘Internet’,” October 24, 1995. https://www.nitrd.gov/fnc/internet_res.pdf.

Ferreira, Becky. “Watch Our Documentary on Detroit’s Grassroots Internet Network.” Vice, November 21, 2017. https://www.vice.com/en_us/article/qv395v/watch-our-documentary-on-detroits-grassroots-internet-network.

First Mile Connectivity Consortium. *Stories from the First Mile: Digital Technologies in Remote and Rural Indigenous Communities*. Fredericton: First Nations Innovation and First Mile Connectivity Consortium, 2018. http://firstmile.ca/wp-content/uploads/Stories-from-the-First-MIle-2018.pdf.

First Mile. “Community Stories.” Accessed February 17, 2020. http://firstmile.ca/category/community-stories/.

First Mile. “Free Online Course: Colonialism and the e-Community,” 2016. http://firstmile.ca/free-online-course/.

Fischer, Gerhard, and Thomas Herrmann. “Socio-Technical Systems: A Meta-Design Perspective.” *International Journal for Sociotechnology and Knowledge Development* 3, no. 1 (2011): 1–33. https://doi.org/10.4018/jskd.2011010101. Also available at https://l3d.colorado.edu/wp-content/uploads/2021/01/Published-JOURNAL-version.pdf.

Fitzpatrick, Jason. “How to Turn a Raspberry Pi into an Always-On BitTorrent Box.” How-To Geek, July 3, 2017. https://www.howtogeek.com/142044/how-to-turn-a-raspberry-pi-into-an-always-on-bittorrent-box/.

Fitzpatrick, Jason. "How to Turn a Raspberry Pi into an Always-On Usenet Machine." How-To Geek, May 21, 2016. https://www.howtogeek.com/142249/how-to-turn-a-raspberry-pi-into-an-always-on-usenet-machine/.

Freire Institute. "Paulo Freire." Accessed February 9, 2020. https://www.freire.org/paulo-freire/.

Freire, Paulo. "First Letter: Reading the World/Reading the Word." In *Teachers as Cultural Workers: Letters to Those Who Dare Teach*, 31–47. Boulder, CO: Westview Press, 2005.

Freire, Paulo. *Pedagogy of the Oppressed*. 50th anniversary edition. New York: Bloomsbury Academic, 2018.

Fritzing. "Fritzing." Accessed February 9, 2020. https://fritzing.org/home/.

Frontline. "The Long March of Newt Gingrich," January 16, 1996. Accessed July 7, 2020. https://www.pbs.org/wgbh/pages/frontline/newt/index.html.

Full Frontal with Samantha Bee. "Black Future Month | March 20, 2019 Act 3" March 20, 2019. https://www.youtube.com/watch?v=vJrHFBCa4MQ.

Galco Industrial Electronics. "What Is a Capacitor?" YouTube, October 17, 2014. https://www.youtube.com/watch?v=B-m78bU-xxc&feature=youtu.be.

Gangadharan, Seeta Peña. "The Downside of Digital Inclusion: Expectations and Experiences of Privacy and Surveillance among Marginal Internet Users." *New Media & Society* 19, no. 4 (April 2017): 597–615. https://doi.org/10.1177/1461444815614053.

Gershenfeld, Neil, and Jim Euchner. "Atoms and Bits: Rethinking Manufacturing." *Research-Technology Management* 58, no. 5 (September 1, 2015): 16–23. https://doi.org/10.5437/08956308X5805003.

Gershenfeld, Neil, Raffi Krikorian, and Danny Cohen, "The Internet of Things," *Scientific American*, October 2004, https://www.scientificamerican.com/article/the-internet-of-things/.

Gershenfeld, Neil, and JP Vasseur. "As Objects Go Online: The Promise (and Pitfalls) of the Internet of Things." *Foreign Affairs* 93, no. 2 (March 2014): 60–67.

Gingrich, Newt. "Renewing American Civilization," CSPAN September 10, 1993. https://www.c-span.org/video/?50261-1/renewing-american-civilization.

Goddard, N. D. R., R. M. J. Kemp, and R. Lane, "An Overview of Smart Technology," *Packaging Technology and Science* 10, no. 3 (1997): 129. https://doi.org/10.1002/(SICI)1099-1522(19970501/30)10:3%3C129::AID-PTS393%3E3.0.CO;2-C.

Golumbia, "Cyberlibertarians' Digital Deletion of the Left," *Jacobin Magazine*, December 4, 2013. https://www.jacobinmag.com/2013/12/cyberlibertarians-digital-deletion-of-the-left/.

Gore, Al. "Infrastructure for the Global Village." *Scientific American*, September 1991. https://www.scientificamerican.com/magazine/sa/1991/09-01/.

Greene, Daniel. "Discovering the Divide: Technology and Poverty in the New Economy." *International Journal of Communication* 10 (2016): 1212–31.

Grundner, Tom. *Seizing the Infosphere: An Alternative Vision for National Computer Networking*. Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, 1994. https://www.ideals.illinois.edu/handle/2142/365.

Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions." *Future Generation Computer Systems*, Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond, 29, no. 7 (September 1, 2013): 1645–60. https://doi.org/10.1016/j.future.2013.01.010.

Gubrium, Aline. "Digital Storytelling: An Emergent Method for Health Promotion Research and Practice." *Health Promotion Practice* 10, no. 2 (2009): 186–91.

Gulyas, Alex. "Make a Mail Server Out of Your RPi3." Hackster.io, April 23, 2017. https://www.hackster.io/gulyasal/make-a-mail-server-out-of-your-rpi3-5829f0.

Gus. "Raspberry Pi Email Server Using Citadel." Pi My Life Up, September 11, 2019. https://pimylifeup.com/raspberry-pi-email-server/.

Gus. "Raspberry Pi IRC Server: Setup Your Own Chat Network." Pi My Life Up, September 10, 2017. https://pimylifeup.com/raspberry-pi-irc-server/.

Harris, Tristan. "How Technology Is Hijacking Your Mind — from a Former Insider." *Medium* (blog), October 16, 2019. https://medium.com/thrive-global/how-technology-hijacks-peoples-minds-from-a-magician-and-google-s-design-ethicist-56d62ef5edf3.

Hearne, Betsy. Story, from fireplace to cyberspace: Connecting children and narrative. In B. G. Hearne (Ed.), Allerton Park Institute 39th (p. 143). Urbana-Champaign, IL: University of Illinois Graduate School of Library and Information Science, 1997.

Heilemann, John. "The Making of The President 2000." *Wired*, December 1, 1995. https://www.wired.com/1995/12/gorenewt/.

Hokanson, Brad. "The Design Critique as a Model for Distributed Learning." In *The Next Generation of Distance Education: Unconstrained Learning*, edited by Leslie Moller and Jason B. Huett, 71–83. Boston, MA: Springer US, 2012. https://doi.org/10.1007/978-1-4614-1785-9_5.

Hollingsworth, Gordon. "The Raspberry Pi PiServer Tool." Raspberry Pi, January 4, 2018. https://www.raspberrypi.org/blog/piserver/.

Hoven, Jeroen van den, Pieter E. Vermaas, and Ibo van de Poel. "Design for Values: An Introduction." In *Handbook of Ethics, Values, and Technological Design*, edited by Jeroen van den Hoven, Pieter E. Vermaas, and Ibo van de Poel, 1–7. Dordrecht: Springer Netherlands, 2015. https://doi.org/10.1007/978-94-007-6970-0_40.

Huckle, Steve, Rituparna Bhattacharya, Martin White, and Natalia Beloff. "Internet of Things, Blockchain and Shared Economy Applications." *Procedia Computer Science*, The 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)/The 6th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2016)/Affiliated Workshops, 98 (January 1, 2016): 461–66. https://doi.org/10.1016/j.procs.2016.09.074.

IDEO. "Design Thinking for Libraries." Design Thinking for Libraries, 2015. https://designthinkingforlibraries.com/.

International Society for Technology in Education (ISTE), and Computer Science Teachers Association (CSTA). "Operational Definition of Computational Thinking for K-12 Education," 2011. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf.

ISTE. "ISTE Standards." Accessed February 18, 2020. https://www.iste.org/standards.

Johnson, Eric. "How Technology Is Designed to Addict Us." Vox, January 30, 2017. https://www.vox.com/2017/1/30/14432142/technology-app-addiction-time-well-spent-tristan-harris-recode-podcast.

Kim, Joshua. "'Fiber' Is a Wakeup Call to Our Digital Learning Community." Inside Higher Ed, January 13, 2019. https://www.insidehighered.com/blogs/technology-and-learning/fiber-wakeup-call-our-digital-learning-community.

Kimmerer, Robin Wall. "Reclaiming the Honorable Harvest: Robin Kimmerer at TEDxSitka." YouTube, August 18, 2012. https://www.youtube.com/embed/Lz1vgfZ3etE.

Kimmerer, Robin Wall. *Braiding Sweetgrass: Indigenous Wisdom, Scientific Knowledge, and the Teachings of Plants*, 48–59. Minneapolis, MN: Milkweed Editions, 2013.

Kimmerer, Robin Wall. *Mapping a New Geography of Hope: Robin Wall Kimmerer Keynote*, 2015. https://www.youtube.com/watch?v=QhQKdJHLDcw&feature=youtu.be.

King, Jamilah. "A Tech Innovation in Detroit: Connect People, Not Computers." COLORLINES, October 3, 2012. https://www.colorlines.com/articles/tech-innovation-detroit-connect-people-not-computers.

King, Rev. Dr. Martin Luther, Jr. "Letter From a Birmingham Jail." Birmingham, AL, 1963. https://www.africa.upenn.edu/Articles_Gen/Letter_Birmingham.html.

King, Martin Luther. "Beyond Vietnam." In: Carson, Clayborne and Shepard, Kris, eds. A Call to Conscience, 139-164. Grand Central Publishing, 2001.

Kleine, Dorothea. "The Capability Approach and the 'Medium of Choice': Steps towards Conceptualising Information and Communication Technologies for Development." *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 119–30. https://doi.org/10.1007/s10676-010-9251-5.

Kuhlthau, Carol Collier. "Guided Inquiry: School Libraries in the 21st Century." *School Libraries Worldwide* 16, no. 1 (January 2009): 17–28.

Ladson-Billings, Gloria. "Toward a Theory of Culturally Relevant Pedagogy." *American Educational Research Journal* 32, no. 3 (September 1, 1995): 465–91. https://doi.org/10.3102/00028312032003465.

Ladson-Billings, Gloria. Critical race theory—What it is not! In Handbook of critical race theory in education (pp. 32–43). Routledge, 2021.

Lady Ada. "CircuitPython on Linux and Raspberry Pi." Adafruit Learning System, February 8, 2020. https://learn.adafruit.com/circuitpython-on-raspberrypi-linux.

Lady Ada. "Introducing ItsyBitsy M0 Express." Adafruit Learning System, February 8, 2020. https://learn.adafruit.com/introducing-itsy-bitsy-m0.

Lady Ada. "Photocells." Adafruit Learning System, July 29, 2012. https://learn.adafruit.com/photocells/overview.

Lessnullvoid. "Pulsum-Plantae." GitHub. Accessed February 9, 2020. https://github.com/Lessnullvoid/Pulsum-Plantae.

Lewis, John, and Valerie Jackson. "The Boy From Troy: How Dr. King Inspired A Young John Lewis," January 17, 2020. https://storycorps.org/stories/the-boy-from-troy-how-dr-king-inspired-a-young-john-lewis/.

Lewis, T, Seeta Peña Gangadharan, M Saba, and T Petty. *Digital Defense Playbook: Community Power Tools for Reclaiming Data.* Detroit: Our Data Bodies, 2018.

Liukas, Linda. "A Delightful Way to Teach Kids about Computers | Linda Liukas." YouTube, February 23, 2016. https://www.youtube.com/watch?v=vcxwcWuq7KQ&feature=emb_title.

Lorini, Maria Rosa, Amalia Sabiescu, and Nemanja Memarovic. "Collective Digital Storytelling in Community-Based Co-Design Projects: An Emergent Approach." *The Journal of Community Informatics* 13, no. 1 (2017): 109–36. https://doi.org/10.15353/joci.v13i1.3296.

Machado Junior, Nelson. "A Brief Introduction To 5G Technology: What you have to know about the 5th generation of cellular networks", Medium, January 21, 2021. Accessed November 16, 2022, https://medium.com/the-shadow/a-brief-introduction-to-5g-technology-b50c0f453f4

MAZI. "CAPS in H2020." Mazi Project. Accessed July 21, 2020. http://www.mazizone.eu/about/caps-in-h2020/.

MAZI. "Mazi-Project/Guides." GitHub. Accessed July 21, 2020. https://github.com/mazi-project/guides.

McDonald, Clare. "Computer Weekly Announces the Most Influential Women in UK IT 2018." ComputerWeekly.com, September 26, 2018. https://www.computerweekly.com/news/252449081/Computer-Weekly-announces-the-Most-Influential-Women-in-UK-IT-2018.

McDowell, Kate. "Storytelling wisdom: Story, information, and DIKW." *Journal of the Association for Information Science and Technology* 72, no. 10 (2021: 1223-1233. https://doi.org/10.1002/asi.24466.

McDowell, Kate and Nicole A. Cooke. Social Justice Storytelling: A Pedagogical Imperative. The Library Quarterly: Information, Community, Policy, volume 92, no 4 (2022). https://doi.org/10.1086/721391

Mehra, Bharat, Kevin S. Rioux, and Kendra S. Albright. "Social Justice in Library and Information Science." In *Encyclopedia of Library and Information Sciences*, 4820–36. CRC Press, 2009. https://doi.org/10.1081/E-ELIS3-120044526.

Mell, Peter, and Timothy Grance. "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, September 2011.

Milner, H. Richard. "Race, Culture, and Researcher Positionality: Working Through Dangers Seen, Unseen, and Unforeseen." *Educational Researcher* 36, no. 7 (October 2007): 388–400. https://doi.org/10.3102/0013189X07309471.

Merriam-Webster, "Design," accessed July 3, 2020, https://www.merriam-webster.com/dictionary/design.

Merriam-Webster, "Digital Divide," accessed July 3, 2020, https://www.merriam-webster.com/dictionary/digital%20divide.

Merriam-Webster, "Networking," accessed June 2, 2020, https://www.merriam-webster.com/dictionary/networking.

M-Lab. "Internet Measurement Tests." Accessed July 22, 2020. https://www.measurementlab.net/tests/.

M-Lab. "Papers, Presentations, and Regulator Filings." Accessed July 22, 2020. https://www.measurementlab.net/publications/.

M-Lab. "What Is Measurement Lab?" Accessed July 22, 2020. https://www.measurementlab.net/about/.

Mockapetris, P. V. "Domain Names – Concepts and Facilities, Request for Comments: 1034." Network Working Group. Accessed July 21, 2020. https://tools.ietf.org/html/rfc1034.

Mockapetris, P. V. "Request for Comments: 1035." Network Working Group, November 1987. https://www.ietf.org/rfc/rfc1035.txt.

Mockapetris, P. V. “Request for Comments: 882.” Network Working Group, November 1983. https://www.ietf.org/rfc/rfc882.txt.

Mowery, David C., and Timothy Simcoe. “Is the Internet a US Invention?—An Economic and Technological History of Computer Networking.” *Research Policy* 31, no. 8 (December 1, 2002): 1369–87. https://doi.org/10.1016/S0048-7333(02)00069-0.

Moyers, Bill. “Susan Crawford on Why Our Internet Access Is Slow, Costly and Unfair.” *BillMoyers.Com* (blog), February 8, 2013. https://billmoyers.com/segment/susan-crawford-on-why-u-s-internet-access-is-slow-costly-and-unfair/.

Mozilla Foundation. “Download the Guides.” Stronger NYC Communities. Accessed June 10, 2020. https://foundation.mozilla.org/en/initiatives/stronger-nyc-communities/whats-in-the-guides-and-workshops/.

National Digital Inclusion Alliance, “Definitions,” accessed July 6, 2020. https://www.digitalinclusion.org/definitions/.

Nemer, David. “The three types of WhatsApp users getting Brazil’s Jair Bolsonaro elected: WhatsApp has proved to be the ideal tool for mobilizing political support — and for spreading fake news,” *Guardian* (25 October), 2018, at https://www.theguardian.com/world/2018/oct/25/brazil-president-jair-bolsonaro-whatsapp-fake-news

Newby, Gregory B. and Ann P. Bishop. “Community System Users and Uses.” *Proceedings of the ASIS Annual Meeting*. January 1996. https://www.petascale.org/papers/asispnet.pdf.

Next Generation Science Standards. “4-PS3 Energy.” Accessed February 9, 2020. https://www.nextgenscience.org/dci-arrangement/4-ps3-energy.

Nielsen Norman Group. “Usability 101: Introduction to Usability.” Accessed May 3, 2020. https://www.nngroup.com/articles/usability-101-introduction-to-usability/.

Nussbaum, Martha. *Women and Human Development: The Capabilities Approach* (Cambridge: Cambridge University Press, 2000), 78-80.

NYC Digital Safety. “Privacy & Security.” Accessed June 10, 2020. https://nycdigitalsafety.org/.

NYC Mayor’s Office of Media and Entertainment. “Her Big Idea, Featuring Limor Fried.” YouTube, November 6, 2017. https://www.youtube.com/watch?v=cwpIx7n6UYg&feature=emb_title.

Oasis Connections. “Oasis Connections.” YouTube. Accessed June 10, 2020. https://www.youtube.com/channel/UCYuc4hQe-Dlgah_Zw2g7FJg/featured.

Obama, Barack. A New Era of Service Across America, Time Magazine March 19, 2009. Time Inc., New York. (Online Time archive time.com)

Oberhaus, Daniel. "'Master/Slave' Terminology Was Removed from Python Programming Language." *Vice* (blog), September 13, 2018. https://www.vice.com/en_us/article/8x7akv/masterslave-terminology-was-removed-from-python-programming-language.

Oosterlaken, Ilse. "Human Capabilities in Design for Values." In *Handbook of Ethics, Values, and Technological Design*, edited by Jeroen van den Hoven, Pieter E. Vermaas, and Ibo van de Poel, 221–50. Dordrecht: Springer Netherlands, 2015. https://doi.org/10.1007/978-94-007-6970-0_7.

Ostberg, René. "transhumanism". *Encyclopedia Britannica*, 3 Nov. 2022, https://www.britannica.com/topic/transhumanism. Accessed 13 March 2023.

Oxford Internet Institute, University of Oxford. "OII Awards 2018: In Conversation with Judy Wajcman." YouTube. Accessed June 12, 2020. https://www.youtube.com/watch?v=pf7-93ioYsE&feature=youtu.be.

Parallax, Inc. "How to Wire Circuits from Schematics." YouTube, May 25, 2010. https://www.youtube.com/watch?v=vJUX9cvyYjU&feature=youtu.be.

Phelps, Kirstin and Martin Wolske. "Pause for Pedagogy: Perspectives on Digital Leadership Development from Technology Education." International Leadership Development Interview, June 13, 2018. https://youtu.be/Q4dCUie0mUw

PHP.net. "What Is PHP?" PHP Manual. Accessed July 21, 2020. https://www.php.net/manual/en/intro-whatis.php.

Postscapes. "Internet of Things (IoT) History." Accessed July 16, 2020. https://www.postscapes.com/iot-history.

Public Broadcasting Service. "More About Newt | The Long March Of Newt Gingrich." Frontline. Accessed July 17, 2020. https://www.pbs.org/wgbh/pages/frontline/newt/aboutnewt.html.

Public Broadcasting Service. "Renewing American Civilization | The Long March Of Newt Gingrich." Frontline. Accessed July 17, 2020. https://www.pbs.org/wgbh/pages/frontline/newt/newtcollege.html.

Public Sphere Project. "Community Networks." Accessed July 16, 2020. http://publicsphereproject.org/content/community-networks-1.

Python Software Foundation. "9. Classes." Python 3.8.4 Documentation. Accessed July 20, 2020. https://docs.python.org/3/tutorial/classes.html.

Python Software Foundation. "Http.Server — HTTP Servers." Python 3.8.4 Documentation. Accessed July 20, 2020. https://docs.python.org/3/library/http.server.html#module-http.server.

Raspberry Pi Foundation. "About Us." Accessed February 9, 2020. https://www.raspberrypi.org/about/.

Raspberry Pi Foundation. "Build a LAMP Web Server with WordPress." Accessed February 9, 2020. https://projects.raspberrypi.org/en/projects/lamp-web-server-with-wordpress.

Raspberry Pi Foundation. "Help Guides and Resources – How to Use Raspberry Pi." Accessed February 10, 2020. https://www.raspberrypi.org/help/.

Raspberry Pi Foundation. "Raspberry Pi Foundation Annual Review 2018." Raspberry Pi, 2018. https://static.raspberrypi.org/files/about/RaspberryPiFoundationReview2018.pdf.

Raspberry Pi Foundation. "Raspberry Pi Foundation Strategy 2018-2020." Raspberry Pi, 2018. https://static.raspberrypi.org/files/about/RaspberryPiFoundationStrategy2018%E2%80%932020.pdf.

Raspberry Pi Foundation. "Setting up an Apache Web Server on a Raspberry Pi." Raspberry Pi Documentation. Accessed July 21, 2020. https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md.

Raspberry Pi Foundation. "VCGENCMD." Raspberry Pi Documentation. Accessed July 20, 2020. https://www.raspberrypi.org/documentation/raspbian/applications/vcgencmd.md.

Recvlohe, Adam, Holly Grimm and Desiree Kane. Apache® Appropriation. January 6, 2023. https://blog.nativesintech.org/apache-appropriation/. Accessed April 24, 2024.

Rembor, Kattni. "CircuitPython Essentials." Adafruit Learning System, April 2, 2018. https://learn.adafruit.com/circuitpython-essentials.

Rembor, Kattni. "Welcome to CircuitPython!" Adafruit Learning System, December 19, 2017. https://learn.adafruit.com/welcome-to-circuitpython.

Rhinesmith, Colin, and Martin Wolske. "Community Informatics Studio: A Conceptual Framework." In *CIRN Community Informatics Conference "Challenges and Solutions."* Prato, Italy: Centre for Community and Social Informatics, Faculty of IT, Monash University, 2014. https://shareok.org/handle/11244/13631.

Rodgers, Carol R. "Defining Reflection: Another Look at John Dewey and Reflective Thinking." *Teachers College Record* 104, no. 4 (June 2002): 842–66. https://doi.org/10.1111/1467-9620.00181.

Rowley, Jennifer. The wisdom hierarchy: Representations of the DIKW hierarchy. Journal of Information Science, 33, no. 2 (2007): 163–180. https://doi.org/10.1177/0165551506070706

Rubell, Brent. "Adafruit IO Environmental Monitor for Feather or Raspberry Pi." Adafruit Learning System, August 16, 2018. https://learn.adafruit.com/adafruit-io-air-quality-monitor.

Ruiz Brothers. "Boomy Pi Airplay Boombox." Adafruit Learning System, November 8, 2017. https://learn.adafruit.com/boomy-pi-airplay?embeds=allow.

Russell, Andrew L., and Valérie Schafer. "In the Shadow of ARPANET and Internet: Louis Pouzin and the Cyclades Network in the 1970s." *Technology and Culture* 55, no. 4 (December 3, 2014): 880–907. https://doi.org/10.1353/tech.2014.0096.

Sabiescu, Amalia, Aldo de Moor, and Nemanja Memarovic. "Opening up the Culture Black Box in Community Technology Design." *AI & SOCIETY* 34, no. 3 (September 1, 2019): 393–402. https://doi.org/10.1007/s00146-019-00904-z.

Saladhouse. "What Is a Raspberry Pi?" YouTube, April 14, 2015. https://www.youtube.com/watch?v=uXUjwk2-qx4.

Saujani, Reshma. "Teach Girls Bravery, Not Perfection." YouTube, March 28, 2016. https://www.youtube.com/watch?v=fC9da6eqaqg&feature=emb_title.

Sawyer, Ruth. *The way of the storyteller.* New York: The Viking Press, 1942.

Schuler, Douglas. "Community Networks and the Evolution of Civic Intelligence." *AI & SOCIETY* 25, no. 3 (August 1, 2010): 291–307. https://doi.org/10.1007/s00146-009-0260-z.

Sen, Amartya. *Development as Freedom (*New York: Alfred A. Knopf, 1999).

Sinclair, Bruce. "Integrating the Histories of Race and Technology." In *Technology and the African-American Experience: Needs and Opportunities for Study*, 1–17. Boston: MIT Press, 2004. https://mitpress.mit.edu/9780262195041/.

Snow, Eric, Carol Tate, Daisy Rutstein, and Marie Bienkowski. "Assessment Design Patterns for Computational Thinking Practices in Exploring Computer Science," August 2017. https://pact.sri.com/downloads/AssessmentDesignPatternsforComputationalThinking%20PracticesinECS.pdf.

Solórzano, D. G., & Yosso, T. J. (2002). Critical race methodology: Counter-storytelling as an analytical framework for education research. Qualitative Inquiry, 8(1), 23–44.

Sorensen, Janni, and Laura Lawson. "Evolution in Partnership: Lessons from the East St Louis Action Research Project." *Action Research* 10, no. 2 (June 2012): 150–69. https://doi.org/10.1177/1476750311424944.

Stanford History Education Group. "Civic Online Reasoning." Accessed June 10, 2020. https://cor.stanford.edu/.

Stillman, Larry, and Tom Denison. "The Capability Approach Community Informatics." *The Information Society* 30, no. 3 (May 27, 2014): 200–211. https://doi.org/10.1080/01972243.2014.896687.

Stop Autonomous Weapons. "Slaughterbots." YouTube, November 12, 2017. https://www.youtube.com/embed/9CO6M2HsoIA.

StoryCorps. "StoryCorps DIY." Accessed February 10, 2020. https://storycorps.org/participate/storycorps-diy/.

StoryCorps. "The Great Thanksgiving Listen." Accessed February 10, 2020. https://storycorps.org/participate/the-great-thanksgiving-listen/.

Strohmayer, Angelika, and Janis Meissner. ""We Had Tough Times, but We've Sort of Sewn Our Way through It: The Partnership Quilt." *XRDS: Crossroads, The ACM Magazine for Students* 24, no. 2 (December 19, 2017): 48–51. https://doi.org/10.1145/3155128.

Sutherland, T. (2017). Archival amnesty: In search of Black American transitional and restorative justice. Journal of Critical Library and Information Studies, 1(2).

Sweeney, Miriam E. "How to Read for Grad School." Miriam E. Sweeney (blog), June 20, 2012. https://miriamsweeney.net/2012/06/20/readforgradschool/.

Sykora, Carolyn. "Computational Thinking for All." ISTE, September 11, 2014. https://www.iste.org/explore/Solutions/Computational-thinking-for-all?articleid=152.

TechBoomers.com. "Techboomers.Com: Learn How To Use Websites, Apps, and Devices." Accessed June 10, 2020. https://techboomers.com/.

TED Institute. "Internet of Things: Transforming the Routine," 2016. https://www.youtube.com/watch?v=8qjf_VuGW7A.

Toboso, Mario. "Rethinking Disability in Amartya Sen's Approach: ICT and Equality of Opportunity," *Ethics and Information Technology* 13, no. 2 (June 1, 2011): 107–18. https://doi.org/10.1007/s10676-010-9254-2.

The Berkman Klein Center for Internet & Society. "Susan Crawford on Captive Audience: The Telecom Industry & Monopoly Power in the New Gilded Age." Accessed April 6, 2020. https://www.youtube.com/watch?v=7R4xhwy-1oI.

The White House: President Barack Obama. "The Untold History of Women in Science and Technology." Accessed February 9, 2020. https://obamawhitehouse.archives.gov/women-in-stem.

Toyama, Kentaro. "TEDxTokyo – Kentaro Toyama – 05/15/10 – (English)." YouTube, May 10, 2010. https://www.youtube.com/watch?v=cxutDM2r534&feature=emb_title.

Toyama, Kentaro. *Geek Heresy: Rescuing Social Change from the Cult of Technology*. New York: PublicAffairs, 2015.

U.S. Department of Commerce, National Telecommunications and Information Administration, and Economic and Statistics Administration. "Falling Through The Net: A Survey of the "Have Notes" in Rural and Urban America." July 1995. https://ntia.gov/page/falling-through-net-survey-have-nots-rural-and-urban-america.

U.S. Department of Commerce, National Telecommunications and Information Administration, and Economic and Statistics Administration. "Falling Through The Net II: New Data on the Digital Divide." July 1998. https://www.ntia.doc.gov/legacy/ntiahome/net2/falling.html.

U.S. Department of Commerce, National Telecommunications and Information Administration, and Economic and Statistics Administration. "Falling Through The Net: Defining the Digital Divide." July 1999. https://ntia.gov/legacy/ntiahome/fttn99/FTTN.pdf.

U.S. Department of Commerce, National Telecommunications and Information Administration, and Economic and Statistics Administration. “Falling Through The Net: Towards Digital Inclusion.” October 2000. https://www.ntia.doc.gov/files/ntia/publications/fttn00.pdf.

U.S. Department of Health and Human Services. “User-Centered Design Basics.” Usability.Gov, April 3, 2017. https://www.usability.gov/what-and-why/user-centered-design.html.

UIL UNESCO. “Learning Cities and Education for Sustainable Development.” YouTube, December 11, 2018. https://www.youtube.com/watch?v=gPUK1VifD_U&feature=emb_title.

University of Illinois at Urbana-Champaign. “Wireless Guest and Device Management Portal.” Accessed June 12, 2020. https://go.illinois.edu/illinoisnet.

Upton, Liz. “PiPlanter: A Plant Growth Monitor.” Raspberry Pi, July 9, 2013. https://www.raspberrypi.org/blog/piplanter-a-plant-growth-monitor/.

Urban, Ivette Bayo. “We Are Our Stories: Is Technology Rewriting Our Values? | Ivette Bayo Urban | TEDxSnoIsleLibraries.” YouTube, December 23, 2016. https://www.youtube.com/watch?v=G7oN1zdS8HE&feature=emb_title.

Urquhart, Lachlan, and Tom Rodden. “A Legal Turn in Human Computer Interaction? Towards ‘Regulation by Design’ for the Internet of Things.” SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, March 11, 2016. https://papers.ssrn.com/abstract=2746467.

VanGenderen, Sean. “What Is Design Thinking?” YouTube, April 4, 2014. https://www.youtube.com/watch?v=a7sEoEvT8l8&feature=youtu.be.

Velden, Maja van der, and Christina Mörtberg. “Participatory Design and Design for Values.” In *Handbook of Ethics, Values, and Technological Design*, edited by Jeroen van den Hoven, Pieter E. Vermaas, and Ibo van de Poel, 41–66. Dordrecht: Springer Netherlands, 2014. https://doi.org/10.1007/978-94-007-6994-6_33-1.

Vermaas, Pieter E., Paul Hekkert, Noëmi Manders-Huits, and Nynke Tromp. “Design Methods in Design for Values.” In *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains*, edited by Jeroen van den Hoven, Pieter E. Vermaas, and Ibo van de Poel, 179–201. Dordrecht: Springer Netherlands, 2015. https://doi.org/10.1007/978-94-007-6970-0_10.

Villanueva-Mansilla, Eduardo. ICT policies in Latin America: Long-term inequalities and the role of globalized policy-making. *First Monday*, *25*(7), 2020. https://doi.org/10.5210/fm.v25i7.10865

Vis, Peter. “Ceramic Disc Capacitor Values / Code / Label.” Petervis.com. Accessed February 13, 2020. https://www.petervis.com/electronics/Ceramic_Disc_Capacitor_Values/Ceramic_Disc_Capacitor_Values.html.

Vlahos, James. "Smart Talking: Are Our Devices Threatening Our Privacy?" *The Guardian*, March 26, 2019. https://www.theguardian.com/technology/2019/mar/26/smart-talking-are-our-devices-threatening-our-privacy.

Wajcman, Judy. "Feminist Theories of Technology." *Cambridge Journal of Economics* 34, no. 1 (January 1, 2010): 143–52. https://doi.org/10.1093/cje/ben057.

Wenger, Etienne. "Communities of Practice: A Brief Introduction." *Communities of Practice*, October 20, 2011, 1–7.

Wikipedia. "São Tomé." July 17, 2020. https://en.wikipedia.org/w/index.php?title=S%C3%A3o_Tom%C3%A9&oldid=968141494.

Wikipedia. "UTF-8." July 20, 2020. https://en.wikipedia.org/w/index.php?title=UTF-8&oldid=968648724.

Winner, Langdon. "Cyberlibertarian Myths and the Prospects for Community." *ACM SIGCAS Computers and Society* 27, no. 3 (September 1, 1997): 14–19. https://doi.org/10.1145/270858.270864.

W.K. Kellogg Foundation. Truth, Racial Healing and Transformation Implementation Guide, December 2016. http://www.racialequityresourceguide.org.

Wolske, Martin, and Colin Rhinesmith. "Critical Questions for Community Informatics in Practice from an Ethical Perspective." *The Journal of Community Informatics* 12, no. 3 (2016): 236–42.

Wolske, Martin, Colin Rhinesmith, and Beth Kumar. "Community Informatics Studio: Designing Experiential Learning to Support Teaching, Research, and Practice." *Journal of Education for Library and Information Science* 55, no. 2 (April 2014): 166–77.

Wolske, Martin, Deven Gibbs, Adam Kehoe, Vera Jones, and Shanon Irish. "Outcome of Applying Evidence-Based Design to Public Computing Centers: A Preliminary Study." *The Journal of Community Informatics* 9, no. 1 (November 25, 2012). https://doi.org/10.15353/joci.v9i1.3188.

Wolske, Martin. "Digital Storytelling Workshop." Presented at the Faculty Summer Institute, University of Illinois at Urbana-Champaign, 2011. https://uofi.box.com/s/77jgdy33acn6ezsllqh87472xnux9d56.

Wolske, Martin. "Ethical Electronic Consumerism." Martin Wolske's Blog, August 15, 2013. https://martin.wolske.site/2013/08/15/ethicalelectronicconsumerism/.

Wolske, Martin. "Going Wireless: Introduction." Illinois Media Space, April 15, 2019. https://mediaspace.illinois.edu/media/t/1_p3kk2v2f.

Wolske, Martin. "Intro to Networks." Illinois Media Space, April 15, 2019. https://mediaspace.illinois.edu/media/t/1_d0tj0lc6.

Yosso, Tara J. "Whose Culture Has Capital? A Critical Race Theory Discussion of Community Cultural Wealth," *Race Ethnicity and Education* 8, no. 1 (March 2005): 75. https://doi.org/10.1080/1361332052000341006.

Yu, Liangzhi, Lorcan Dempsey, and Sarah Ormes. “Community Networking: Development, Potentials and Implications for Public Libraries.” *Journal of Librarianship and Information Science* 31, no. 2 (June 1, 1999): 71–83. https://doi.org/10.1177/096100069903100202.

Zakon, Robert H. “Hobbes’ Internet Timeline.” Accessed July 16, 2020. https://tools.ietf.org/html/rfc2235.

Zheng, Yingqin and Bernd Carsten Stahl, “Technology, Capabilities and Critical Perspectives: What Can Critical Theory Contribute to Sen’s Capability Approach?” *Ethics and Information Technology*, 77.

Zieris, Franz, and Lutz Prechelt. “Does Pair Programming Pay Off?” In *Rethinking Productivity in Software Engineering*, edited by Caitlin Sadowski and Thomas Zimmermann, 251–59. Berkeley, CA: Apress, 2019. https://doi.org/10.1007/978-1-4842-4221-6_21.

Zurko, Mary Ellen, and Richard T. Simon. “User-Centered Security.” In *Proceedings of the 1996 Workshop on New Security Paradigms – NSPW ’96*, 27–33. Lake Arrowhead, California, United States: ACM Press, 1996. https://doi.org/10.1145/304851.304859.

Zwass, Vladimir. “Information System.” In *Encyclopædia Britannica*. December 28, 2017. https://www.britannica.com/topic/information-system.

Print Handout

Download the Demystifying Technology Manual

This “cheat sheet” provides key charts and diagrams from *A Person-Centered Guide to Demystifying Technology* in an easy-to-print format.

Acknowledgements

Acknowledgments begin through a reflection that this textbook is the latest edition of a living document arising from my 25 years of engagement, teaching, and research at the School of Information Sciences, University of Illinois Urbana-Champaign. Ann Bishop and Greg Newby, founders of Prairienet Community Network, both welcomed me in and challenged me bring my research and engagement interests into the information science space. Leigh Estabrook, then dean of the School, further opened doors by adding teaching and leadership opportunities to the participatory research of Prairienet. Many of the other faculty and staff of the School have guided my pedagogical development and community inquiry practices, especially Chip Bruce, Ann Bishop, Jon Gant, Nicole Cooke, Manette Messenger, Paul Adams, Sharon Irish, and Lisa Bievenue. But there are so many others that I feel should be listed but worry this acknowledgment would become unwieldy and there's still more to include.

For this PressBooks edition of the living document, special thanks go out to the Center for Digital Inclusion who provided ongoing funding, and the Publishing Without Walls project, which includes the University of Illinois Library, School of Information Sciences, Illinois Program for Research in the Humanities, and Department of African American Studies, and which is generously supported by the Andrew W. Mellon Foundation.

Moving from School to campus, there are many across the University of Illinois that I want to especially recognize. Thanks to all within the East St. Louis Action Research project with whom I have spent rich days (and nights) traveling, discussing, and doing as part of our extended time in community, with community, for community. And much thanks also to the campus engagement scholarship and the broader Outreach and Engagement Practitioners Network, CTRL-Shift, CITL, and Gadget Garage campus/community space!

Extended thanks especially for those deeply valued hours in Prato with the Community Informatics community, and our gatherings between conferences as well!

Colin Rhinesmith, you fit into so many of these categories that I will instead put you into a line by yourself. ?

But real change requires community, and so I want to especially acknowledge members of the Champaign-Urbana, East St. Louis, and São Tomé communities with whom I've collaborated. You have accepted me as I am with such open arms, and have spoken truth to me in so many transformative ways. Thanks to Sally, Joel, James, Vera, Essie, Martha, Christina and Eddie, Brad, Bill, Sister Marge, Earl, Kaitroy, Jorge, James, Elvis – this list should go on, but I'll stop here with apologies to the so many I've left out. I am so very different, my personal and professional engagement is so very different, my work with community technology and networks is so very different, this book is so very different because of you all. Thank you!

Living documents like this develop through extended works in a community of practice. At the top of that list come the many who have spent extended hours as my research and teaching assistants during the time the Demystifying Technology worksheets and handouts turned into Word documents and then into HTML and PressBooks items. Thanks go out to Dinesh, Vandana, Beth, Chris, Jake, Deven, Damon, Jeanie, Sarah, Hailley, Kim, Sarah, Travis, Samaa, Angela, Casey, Rebecca, Rachel, Dan, Brandon, Hila, and Stephanie. And special thanks go out for the deep commitment and extended hours that Henry and Sara put into this work.

I continue to challenge myself to expand the popular education of Freire, and this especially re-centers me as instructor-student, giving space to the student-instructors who have so much from their lived experiences, unique knowledge, and community cultural wealth to bring to the table. This, too, is a deep part of this living document. Thank you.

But the start, the end, and all between comes deeply from the closest of my community, my family. And at the center of that is Angie.