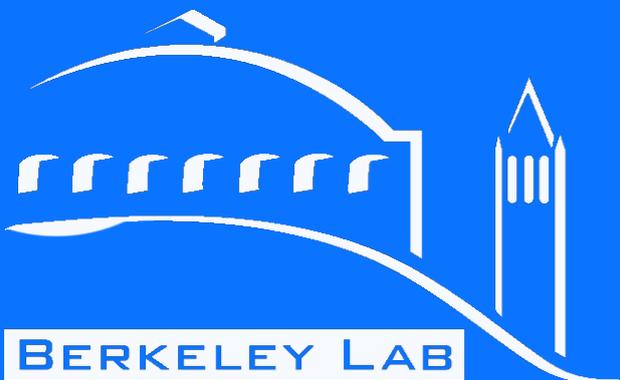
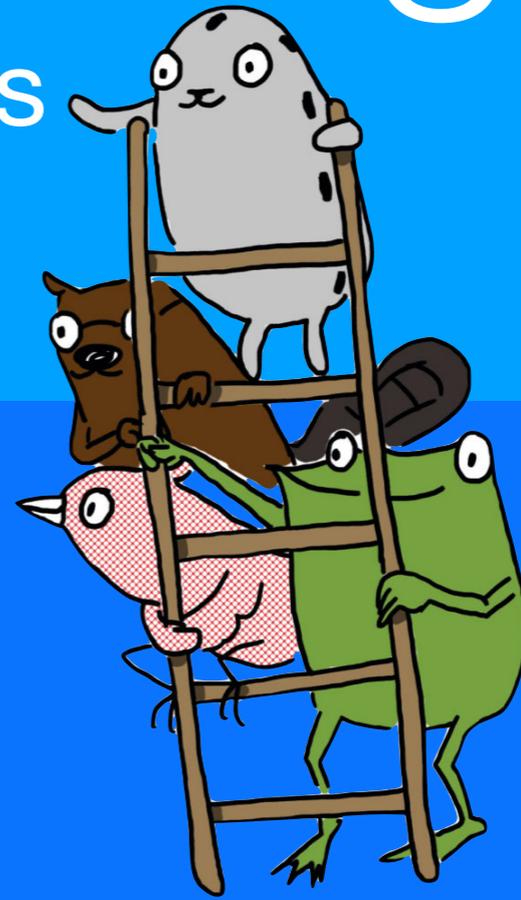


4<sup>th</sup> June 2021

# SAGE electronics & programming

Call for volunteers

Rebecca Carney



[rcarney@lbl.gov](mailto:rcarney@lbl.gov)



### Navigation:

- SAGE Home
- Application
- Program
- Volunteers
- FAQ

### SAGE at Berkeley Lab

Science Accelerating Girls' Engagement in STEM (SAGE) is a one week long summer camp for public high school students (age 14-17) hosted by [National Laboratory](#) scientists and engineers to share what life in STEM (Science, Technology, Engineering, and Mathematics) professions are like.

This program aims to foster innovation, grow the STEM community, and engage intelligent, creative, and passionate young women in the everyday life of scientists and engineers. Throughout the week, students will participate in job shadowing, hands-on projects, professional development, and more!

In 2021, SAGE at Berkeley Lab will be hosted *virtually* from Aug 1 - Aug 7.

### Goals for the camp:

- To encourage college bound students with a passion for discovery and innovation to become a part of the STEM community
- To show that STEM fields offer a diverse set of opportunities: you can balance your interests in artistic expression, historic research, and creative writing with application of your logical thinking, analytic skills, and computational ability.



## 2021 SAGE Camp Schedule

Time	Sunday Aug 1st	Monday Aug 2nd	Tuesday Aug 3rd	Wednesday Aug 4th	Thursday Aug 5th	Friday Aug 6th	Saturday Aug 7th	
8:30-8:45 AM	<b>Opening Day</b> 2pm-4pm	Morning ACTIVE activity (15 mins)					<b>Closing Day</b> 10am-12pm	
8:45 AM		BREAK (5 minutes)						
8:50-9:20 AM		Technical Talks (30 mins)						
9:20 AM		BREAK (10 minutes)						
9:30-10:00 AM		Job Shadow (30 mins)				Projects (100 min)		
10:00 AM		BREAK (5 minutes)						
10:05-11:30 AM		Projects (85 mins)						BREAK (10 mins)
11:30 AM		BREAK (10 minutes)				Presentation Prep (100 min)		
11:40-12:35 AM		Professional Growth (55 mins)						

## Program Activities

### Technical Talks:

The students will hear from a variety of scientists and engineers who will talk about their work, why they chose this profession and how they find it rewarding.

### Small Projects:

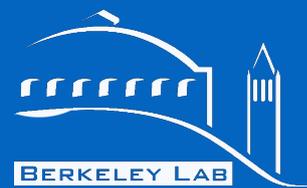
Working on hands-on projects will offer a memorable experience, allow students to gain technical skills and gain a better understanding of the technologies that relate to SLAC and Berkeley Lab work. Project kits will be mailed in advance to all camp participants.

### Job Shadowing:

The students will see and hear about the everyday activities of working professionals in STEM. SLAC professionals will spend time with small groups of students showing them what their daily life is like.

### Professional Growth and Leadership:

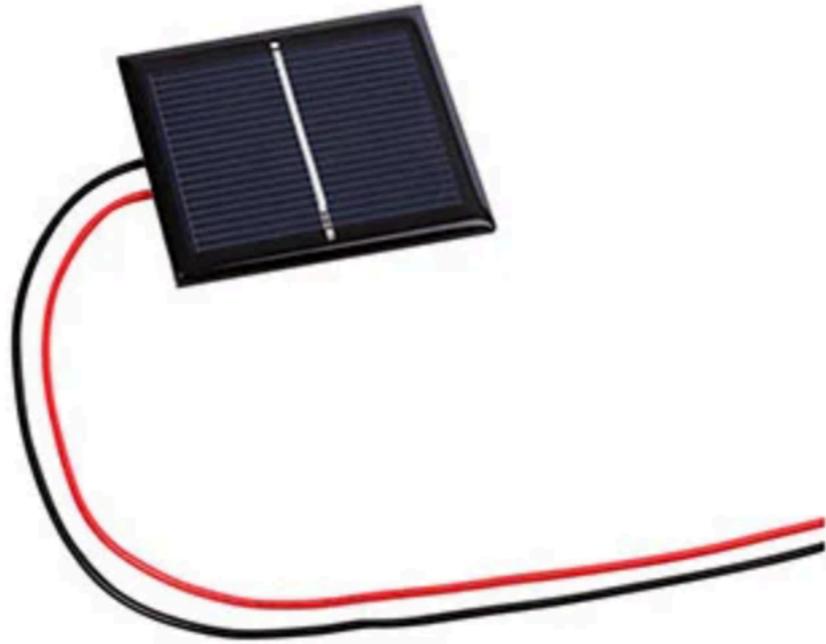
The students will learn skills necessary to enter and stay in STEM fields, as well as practice using tools that help develop confidence and effective communication.





SAGE

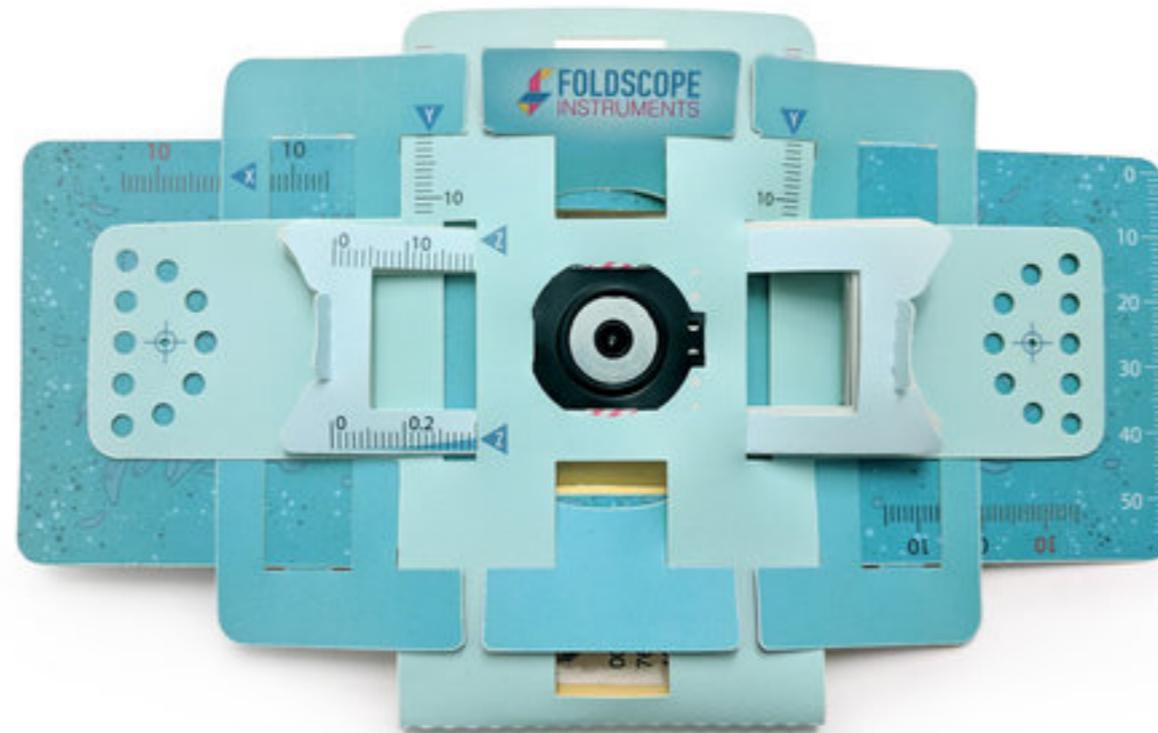
- **Alternative energy:** using photovoltaics and a paper wind-farm to power LEDs





SAGE

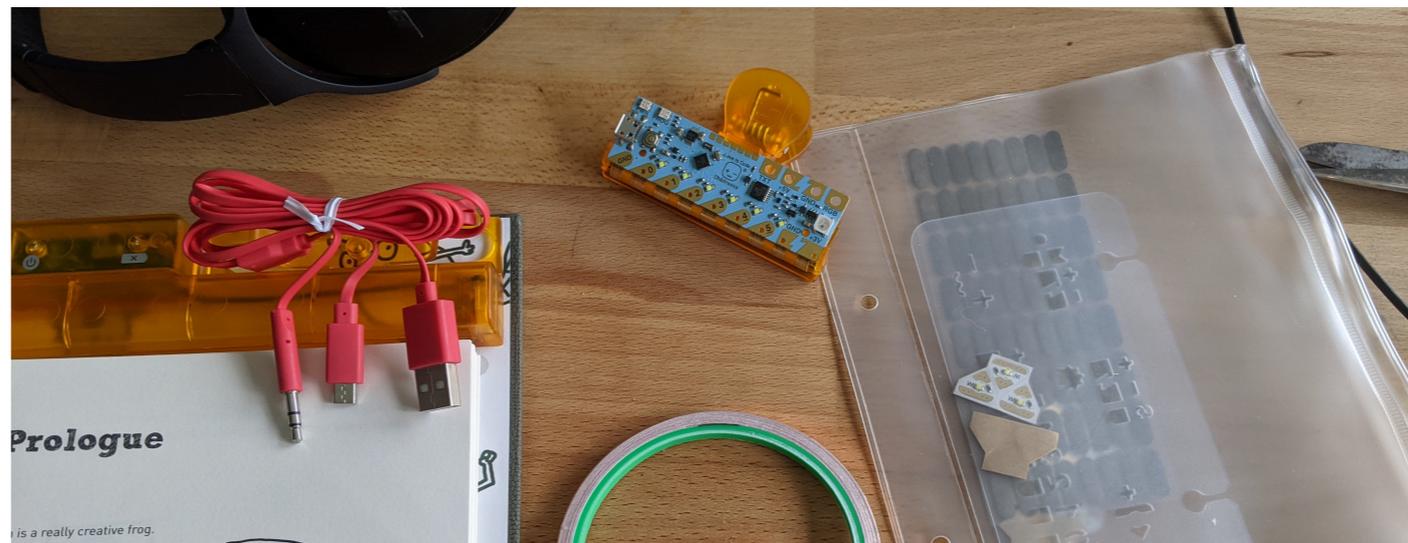
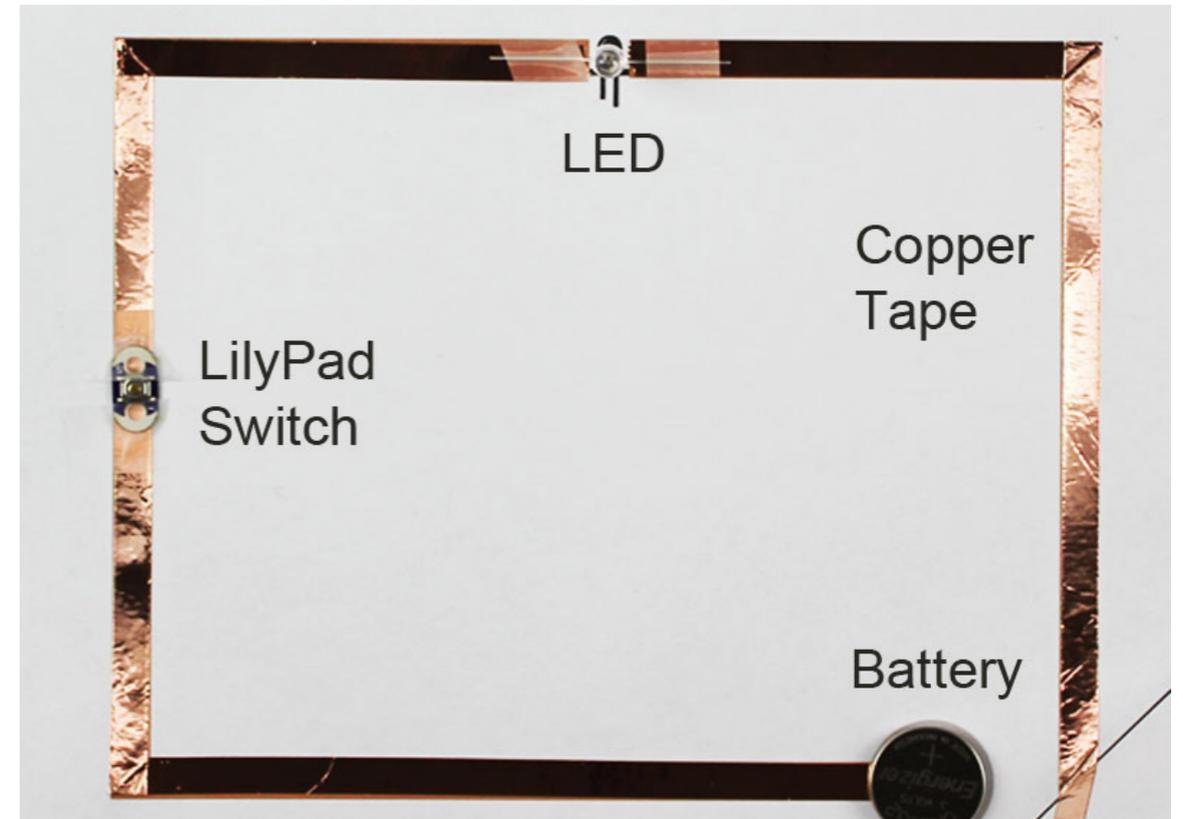
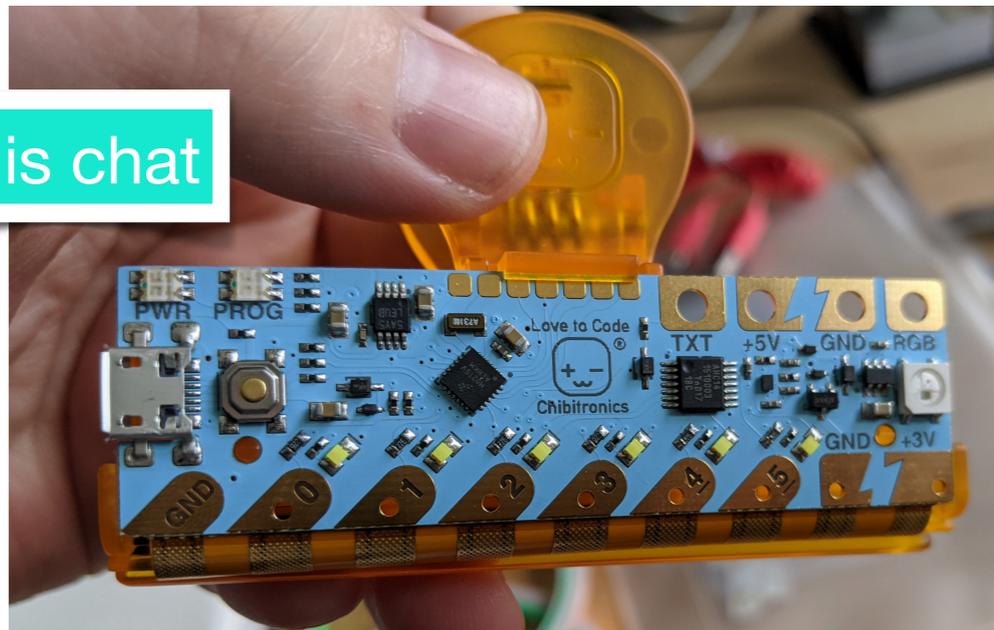
- **Alternative energy:** using photovoltaics and a paper wind-farm to power LEDs
- **Microscopy/optics/cells:** students use a foldscope to transform their cell phones into a microscope. Take samples of dirt and local water and make observations.
- **Crystal growth**





- **Alternative energy:** using photovoltaics and a paper wind-farm to power LEDs
- **Microscopy/optics/cells:** students use a foldscope to transform their cell phones into a microscope. Take samples of dirt and local water and make observations.
- **Coding & electronics:** using chibitronics to explore c-style coding and basic electronics w/ paper circuits + microcontroller

This chat





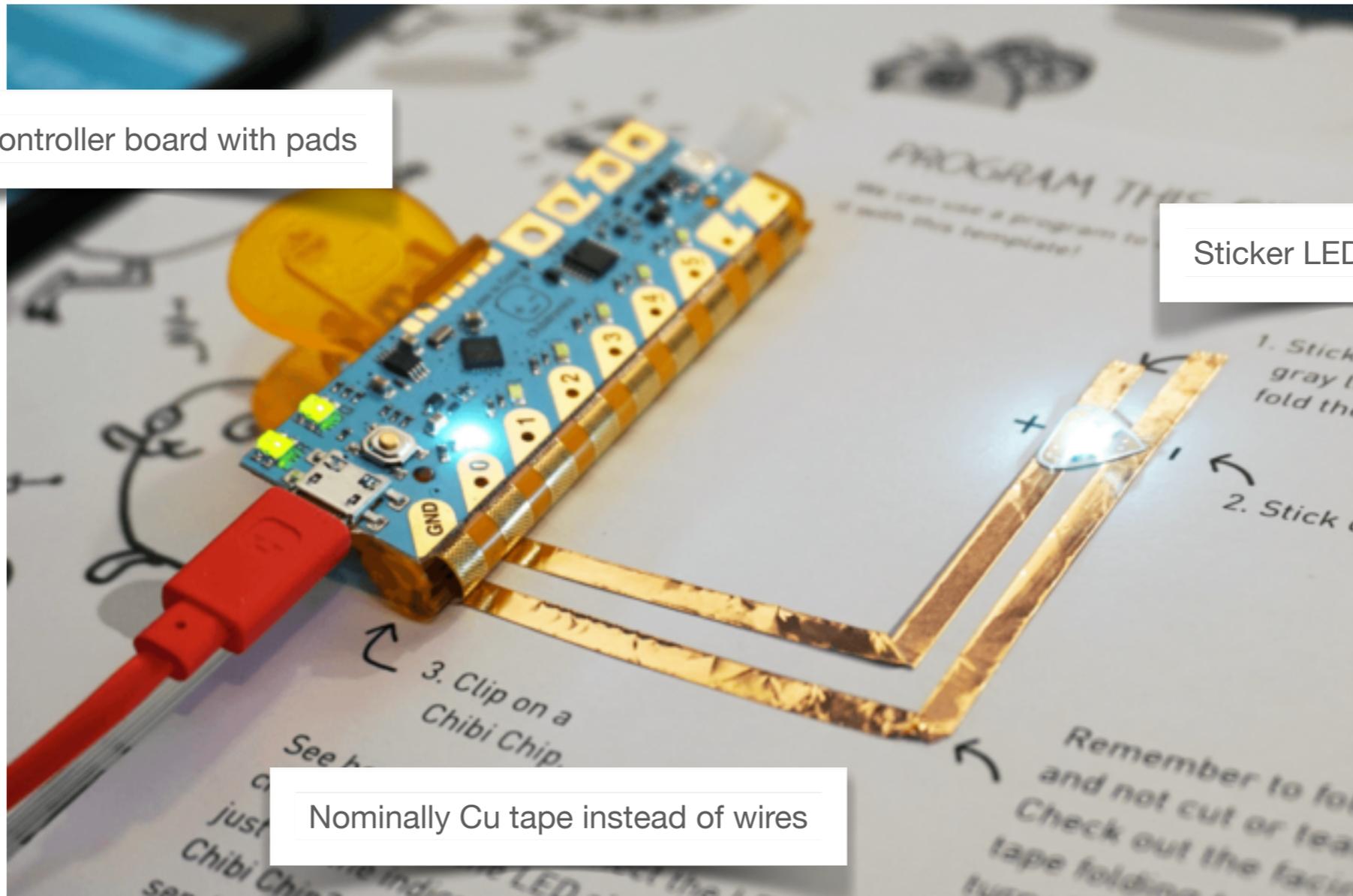
SAGE

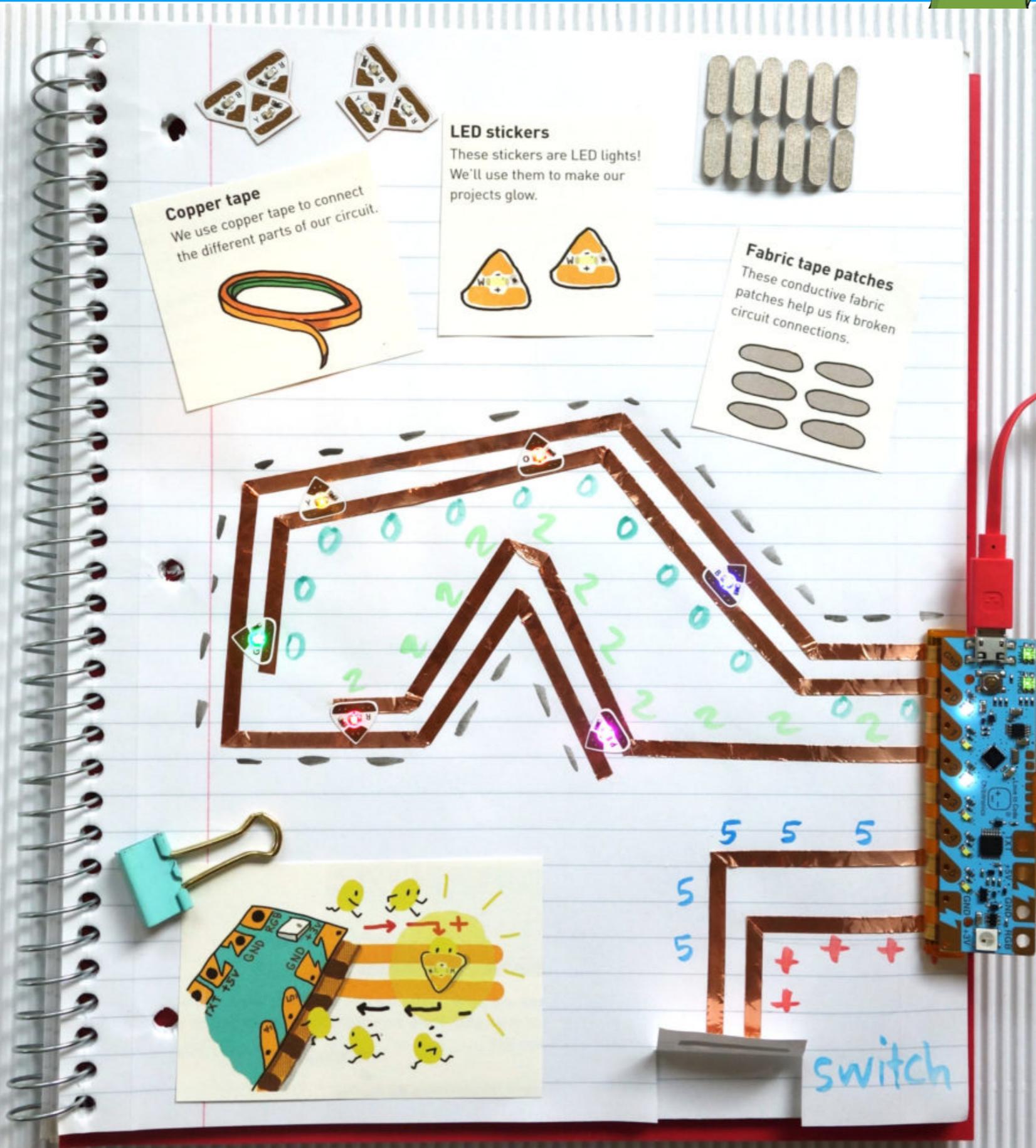
- The children will learn how to:
  - Edit **C-style code** for instant feedback, program microcontroller
  - Relate the coding concepts to a physical process, practical electronics
  - Write pseudo code and use state diagrams to plan work
  - **Build concepts and syntax that translate to real languages (e.g. conditional logic, loops)**
  - **Learn something about electronics in a tangible way: voltage, current, series & parallel, photo-transistors**

Microcontroller board with pads

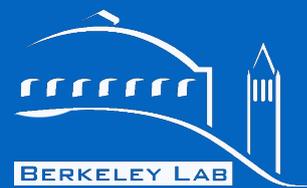
Sticker LEDs & sensors

Nominally Cu tape instead of wires





- Kids can use their own notebook to annotate circuit
- Much easier to use when learning solo: stickers/paper/copper tape easy to connect (and easier to debug on our end!)





- At first I was concerned that this was too juvenile for 14-17 year olds, there is a lot of cutesy illustrations and suggestions to make a coloring book with blinking LEDs.
- However, the content teaches fundamental electronics and C-style syntax!

## DECODE THE CODE

How does the Chibi Chip blink the LED? It's actually following a set of instructions generated from our code! Let's take a look:

```
//Love to Code
//Volume 1: Basic Blink

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

The **comments** are notes to us about what the code does.

The **setup** function happens once at the beginning, when you first turn on the chip.

The **loop** happens over and over after setup is done.

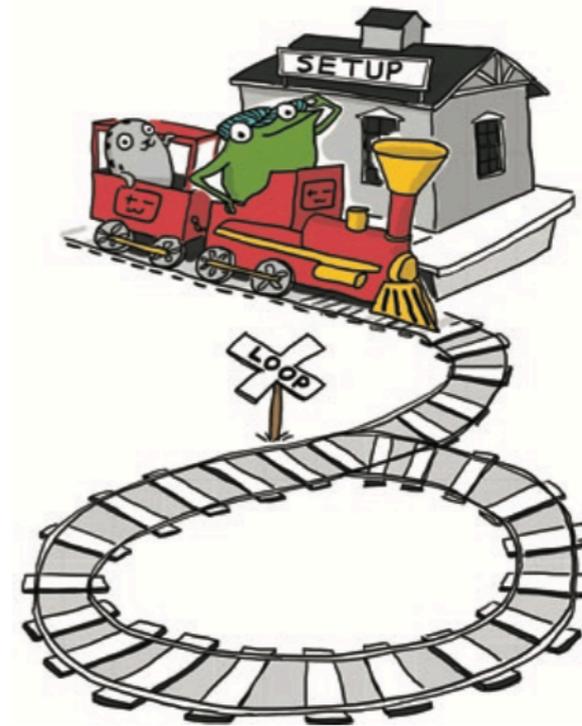
The **comments** describe what the code does. The Chibi Chip will ignore any notes that we leave in the code after a `//` (double slash). Adding comments is a good way to record and remember things.

The main code is split into two parts called **functions**: the "setup" and the "loop" functions. We label the functions first, using `void setup()` and `void loop()`, and then we put the function's code inside squiggly brackets `{` and `}`, after each label. The coded **statements** inside each function are step-by-step instructions for a Chibi Chip. Each statement must end with a `;` semicolon.

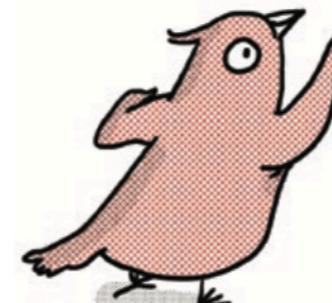
```
void setup() {
  // code here is used to setup the Chibi Chip
  // it's run only once at the beginning
}

void loop() {
  // code here is the loop or 'body'
  // and gets run over and over
}
```

The program runs like a train on a track. First it leaves the station, which is the **setup()**, and then runs around and around on the track, which is your **loop()**.



In other words, when first turned on, the Chibi Chip runs through the **setup()** function once. Then it will run the instructions coded in the **loop()** function over and over, until powered off or reset. The next time it turns on, the Chibi Chip will start with **setup()** again.



The complete set of coded instructions is called a program. The Chibi Chip expects every program to have at least the **setup()** and **loop()** functions. If either is missing, the Chibi Chip won't know how to run the program!



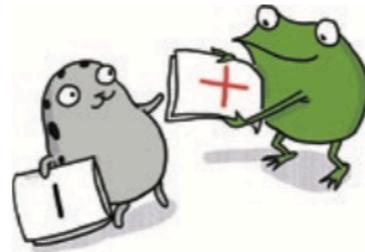


- The characters come across as endearing, NOT patronizing.

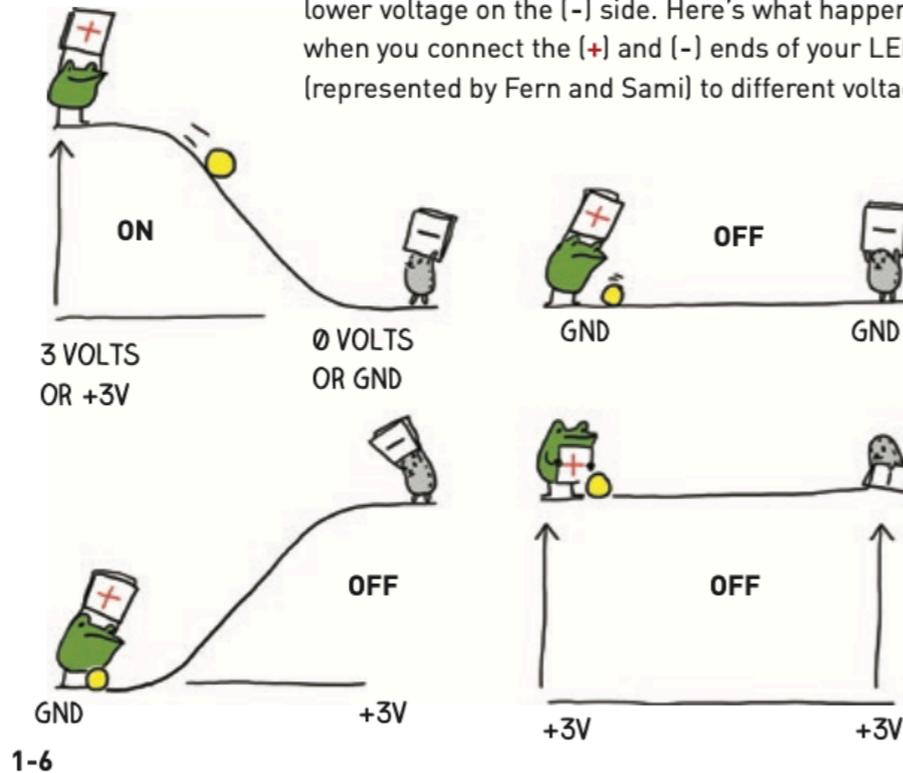


Electricity is a form of **energy** that flows to turn on your light. This flow of energy is called a **current**. Like a ball rolling down a hill, a current will only flow when there is a difference in height, from high to low.

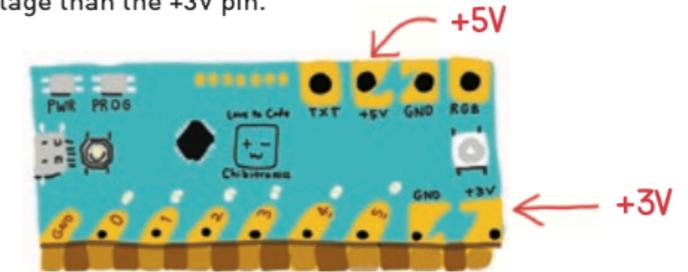
In electronics, the difference in height is called a **voltage**. Voltage is a measurement of potential energy, just like the height of a ball on a hill.



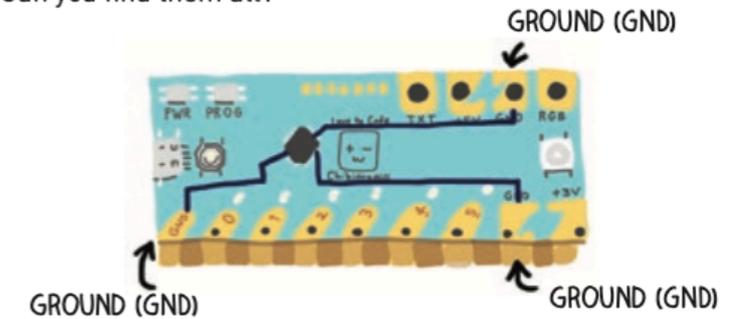
For an LED to turn on, current must flow through the LED from a higher voltage on the (+) side to a lower voltage on the (-) side. Here's what happens when you connect the (+) and (-) ends of your LED (represented by Fern and Sami) to different voltages:



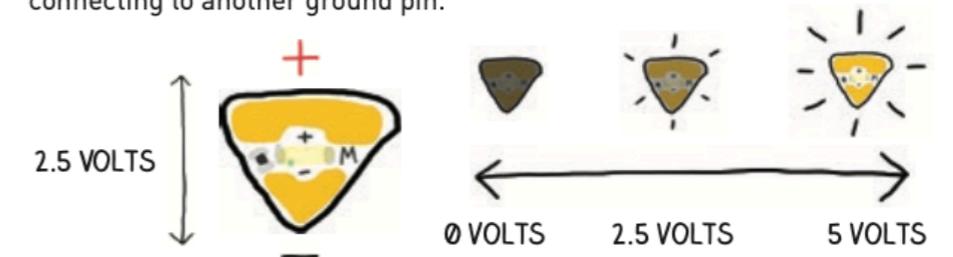
The +3V pin on your Chibi Chip sits at a height of 3 volts. There is another pin on the opposite side labeled +5V. This means it is at 5 volts. This pin provides even more voltage than the +3V pin.



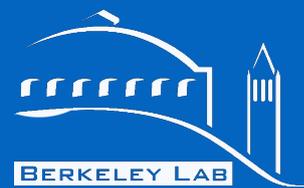
0 volts is called **ground**, and we've labeled the pins with GND. This is where we connect the (-) side of our LED. There are actually three ground pins on a Chip. Can you find them all?



Even though they look like three separate pins, they are actually connected together by copper traces inside the Chibi Chip. This means that they are all at the same voltage (0 volts) so connecting to one ground pin is the same as connecting to another ground pin.



A typical LED sticker needs at least 2.5V to shine brightly, so when you connect it between the +3V pin and GND, there's more than enough voltage to turn the light on. The higher the voltage applied across an LED sticker, the brighter it will shine — up to a point. Be careful, if you give a single LED sticker too much voltage (more than 6V) it can burn out!



4<sup>th</sup> June  
2020

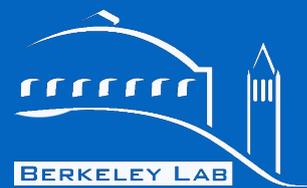


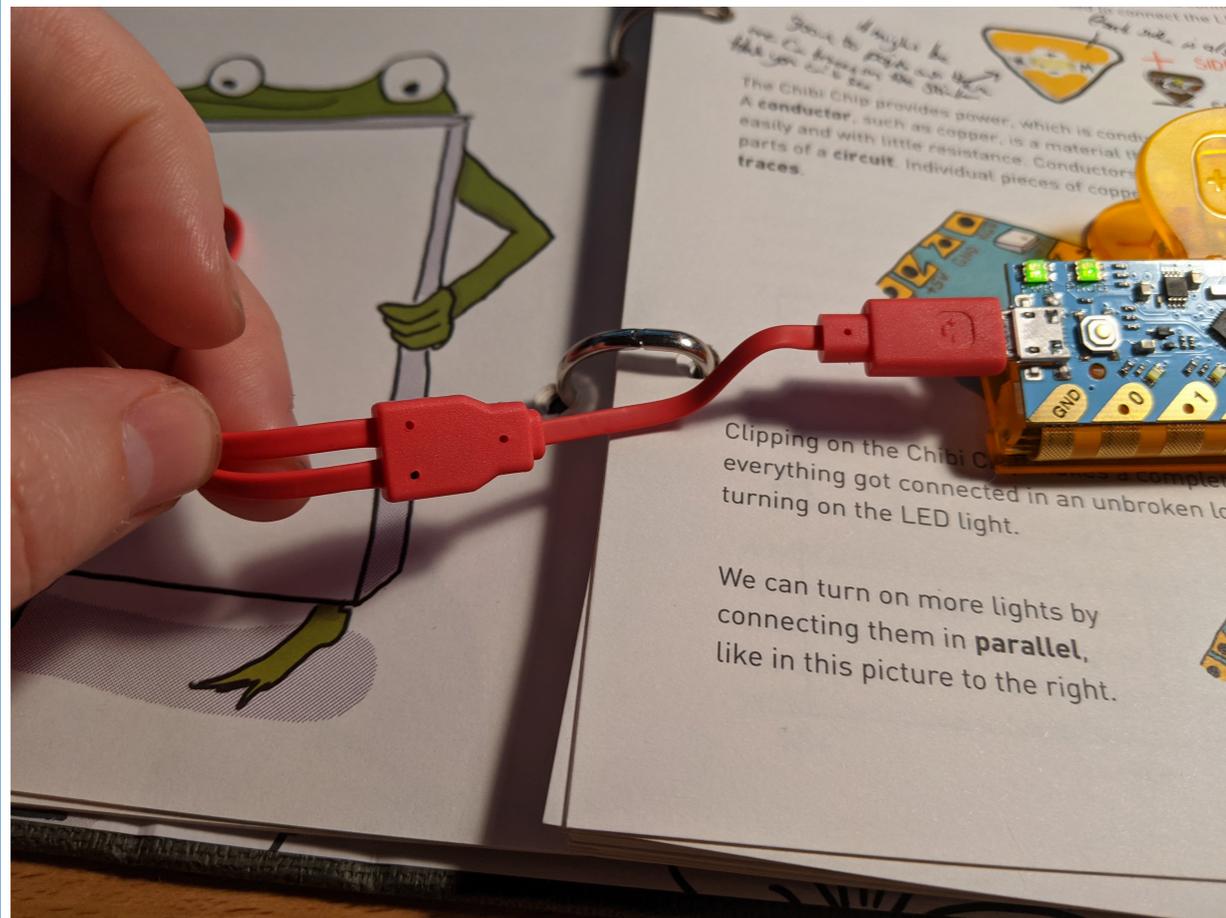
SAGE

rcarney  
@lbl.gov

# Connectors & programming

- **Second chapter** introduces programming with a nice gui interface. Book talks you through each step. But there was one thing I found a bit odd..





- **What was going on with the waveform?**
- Cable on chibiChip is microUSB split to USB-A and.. an audio jack. Also:



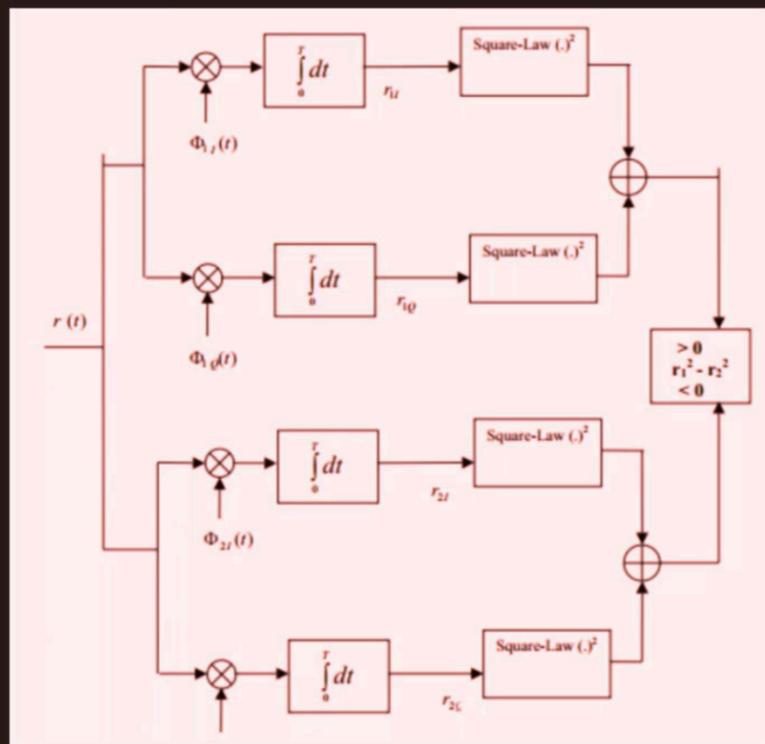
- **“The code starts as text in your browsers which then gets translated by a compiler (nice!) and finally converted into a song that we play to the chibi chip.”**
- Initial reaction: oh man, it sucks that they’re dumbing down a serial or one-wire interface and saying it’s just a ‘song’. This was going so well..
- But then.. why the audio jack?
- Why the little waveform in the GUI?



So I waaaaay underestimated how cool this thing is: they use audio FM to communicate!! (Serial drivers/permissions across platforms)

Deets in this excellent talk by Chibitronics co-found Bunnie: <https://youtu.be/BG--SwqJDPY?t=1230>

## Software: Non-Coherent FSK Demodulation

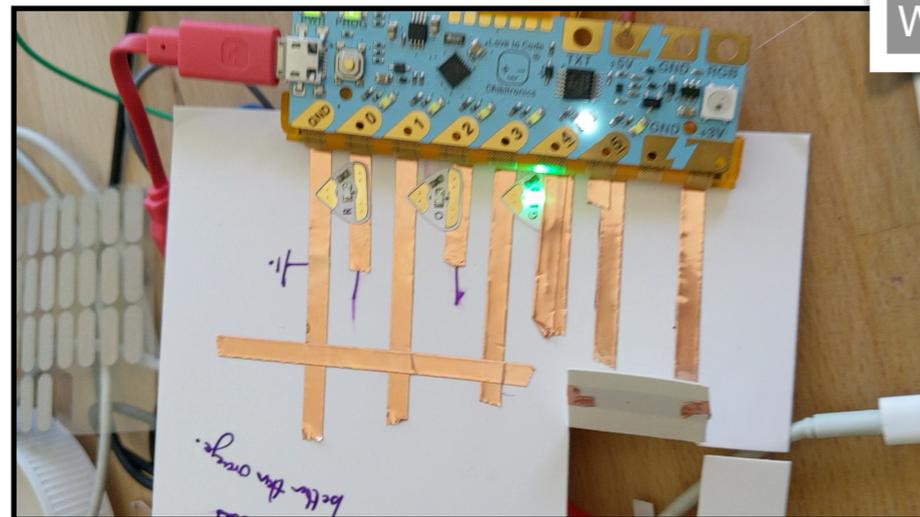


- Convolve incoming samples by sin/cos of each frequency
  - So 4 convolutions
- Each convolution is a series of MACs
- ~250 MACs / sample
- 20MMAC/s at our sampling rate

- The reason this is exciting:
  - This could be an amazing bridge/branching point to talk about **Audio Frequency Shift Keying, DSP, or even encoding protocols!**
  - **Can share coding projects as MP3s.**
  - **The Chibi-literature is not cutting corners.**



- **Pre-camp work** focuses on:
  - Voltage & current
  - Series & parallel circuits
  - Getting to know the equipment incl. LEDs, Cu tape, clip electrodes
  - Programming the microcontroller with web-based script (pre-written)
  - Debugging(!)
  - Ends with a nice open-ended creative project

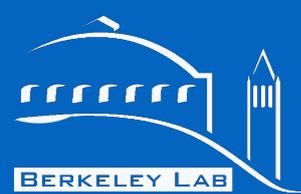
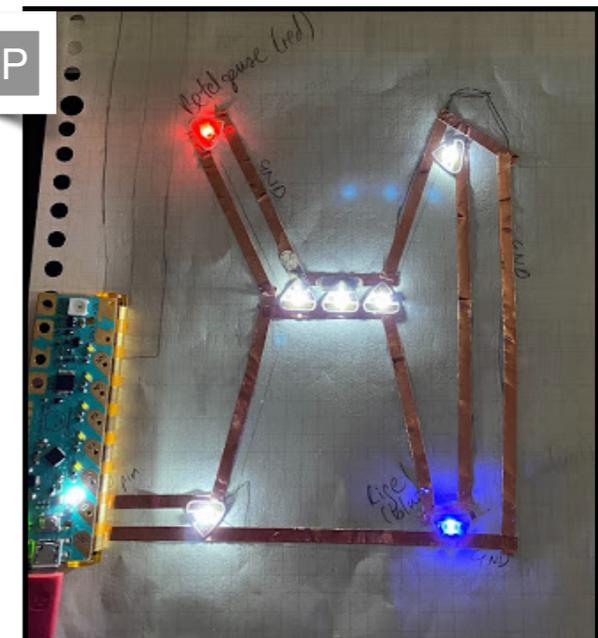


## (2-day) Coding-focused project: Traffic lights

- Layout parallel circuits for traffic light
- Design and build **state-machine** for traffic light
- Include **trigger** for entering state-machine (mechanical paper switch)
- **Stretch Goals:** secondary traffic light (junction or maybe pedestrian walk/don't walk), trigger from light sensor instead.

## (2-day) Sensor-focused project: Constellations

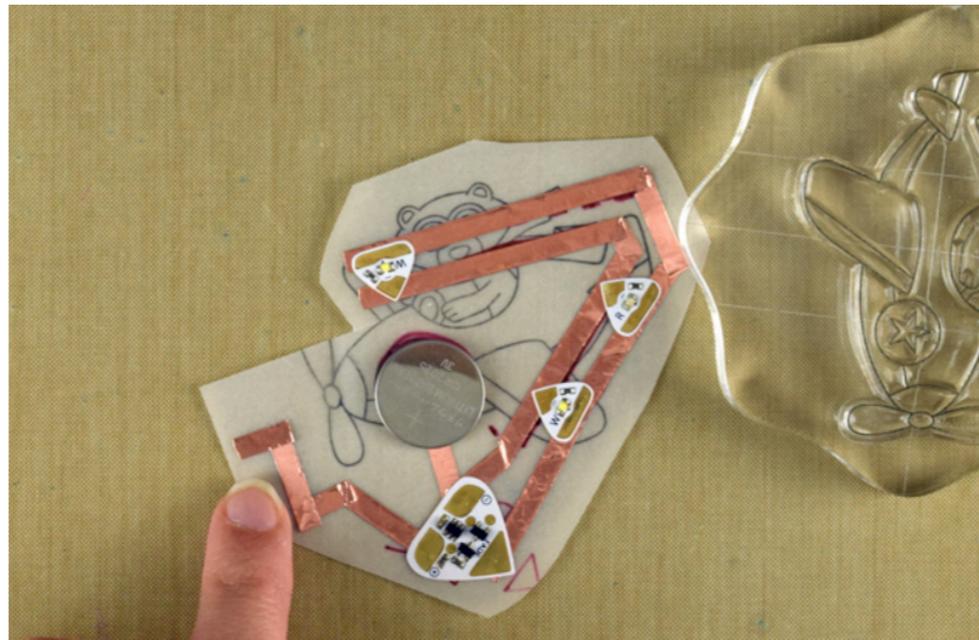
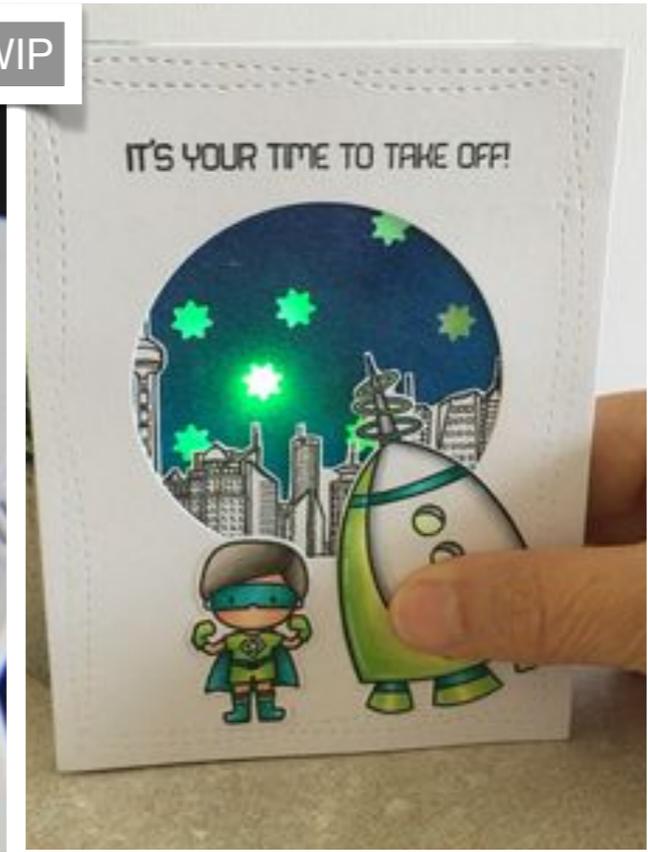
- Layout parallel circuits for a constellation of LEDs
- Learn about CCDs and how they're used in astronomy/astrophysics
- Use a photo-transistor based light sensor to trigger 'stars' coming on in dark
- Have light sensor affect different stars depending on lumi-level (code separated out for parallel pins)
- **Stretch goals:** design circuit with light sensor that turns on LEDs without microcontroller, perhaps some multi-threading.





Pre-camp work focuses on:

- Vol
- Se
- Ge
- el
- Pr
- w
- De
- Er



Any students who had difficulty with the pre-camp work will instead review that material with a **creative paper-project on day 1**, such as a gift card or small illustration.

Following that they will be able to join either of the 2 main tasks on day 2.

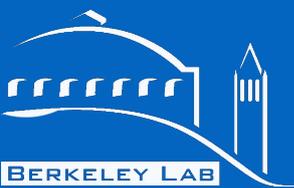
(2-d

- L
- L
- U
- i
- H
- s
- Stretch goals: design circuit with light sensor that turns on LEDs without microcontroller, perhaps some multi-threading.

lights

hanical

r maybe sensor



4<sup>th</sup> June  
2020



SAGE

rcarney  
@lbl.gov



# Organization

- **We have 75 students confirmed for this workshop**
- They will all participate in all the projects (microscopy, energy, chibitronics)
- Chibitronics will use 2 of the 5 project days.
- Split into 2 branches initially: those who were comfortable with the homework and those that weren't. Then it splits further into the traffic light and constellation projects.
- **If we assume ~5 students per group, we need ~15 volunteers for this project alone.**





- **We have 75 students confirmed for this workshop**
- They will all participate in the three projects (microscopy, energy, chibitronics)
- Chibitronics will use 2 of the 5 project days.
- Split into 2 branches initially: those who were comfortable with the homework and those that weren't. Then it splits further into the traffic light and constellation projects.
- **If we assume ~5 students per group, we need ~15 volunteers for this project alone.**

**Short Projects Facilitator:** This facilitator will help lead a project and support students with troubleshooting and facilitation.

(Time commitment: 90 minutes per session plus training sessions for each project.)

- **Camp is 1st week of August**
- **We will have 2 training sessions in July.**

I'm specifically approaching this group because all members have a solid grasp of c-style programming & fundamental electronics. Ideally, that sort of background is what is needed for volunteers.

Sign up here for informational session! [https://docs.google.com/forms/d/e/1FAIpQLSefsyFDXQWgRqHxgKC-XYCBkN54\\_J2bTNLKiRsRwAjpgjmHUqA/viewform](https://docs.google.com/forms/d/e/1FAIpQLSefsyFDXQWgRqHxgKC-XYCBkN54_J2bTNLKiRsRwAjpgjmHUqA/viewform)

