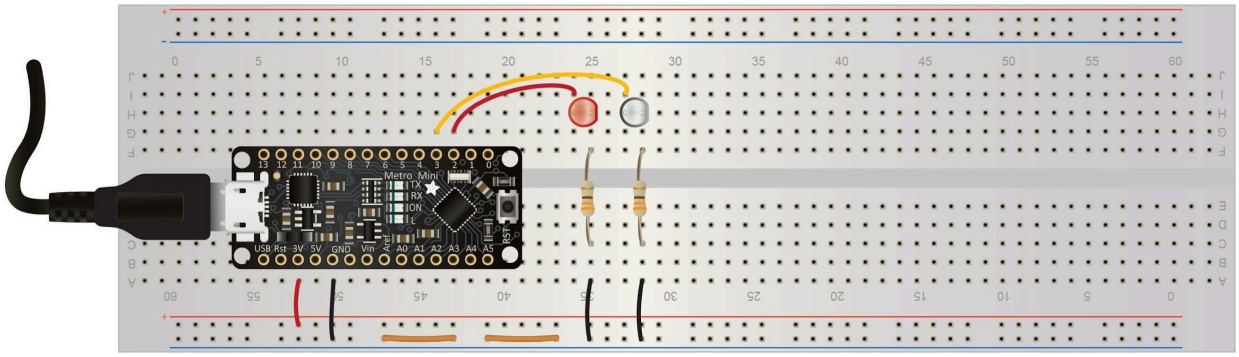
Do-it-Yourself LbyM Material Preparation Guide



This guide assumes that you have had a chance to look through each of the Curriculum Units (1, 2, 3, the Water and Soil experiment unit).

I. Intro
II. Kit overview (include labeled arduino image and resistor chart)
III. Acquiring parts (Arduino, LEDs, resistors, boards, kit boxes, etc, multimeters, tools) [lena chart]
IV. Soldering legs onto Arduino (lena video)
V. LbyM Arduino Sketch Installation
VI. Building an LbyM BasicBoard
VII. Testing an LbyM BasicBoard
VIII. Tips for successful technology implementation [debugging] [troubleshooting] [when all else fails]

I. Intro

Learning by Making makes use of a variety of tools, including computer basic boards, resistors, LED lights of various colors, wire of various colors, batteries of assorted strengths, and even temperature and light sensors. The students will modify their Basic Boards as they follow along with the curriculum, but there is a basic form of the BasicBoard that we suggest building for them ahead of time. This will be the board that they improve upon themselves.

Traditionally, we have offered these materials to our students in the form of kits, which are delivered at the beginning of Unit 1. However, there are no materials *required* to complete Unit 1 other than a laptop or computer with a Chrome internet browser.

Connecting the BasicBoard to the LbyM Web App

- Open serial port (software) that can connect to a USB-2.0 (hardware slot on computer)
    - Note that some serial ports go to different hardware slots on a computer. Your school's IT department may lock your student's serial ports. The serial port connected to the USB connector needs to be open (or opened by your IT department).
- 1 USB-A to USB-C cable (hardware)
- Arduino (metro-mini) Driver (follow link)

II. Kit Overview

There are two different categories of "kits" that should be built to support the LbyM learner. First time set-up of the Arduino and Basicboard is more labor intensive, but those items are easily refurbished for repeated use in the classroom.

Student Kits - Designed to accommodate one student per unit, contains all supplies for ongoing use.

Classroom Supplies Sets - Intended for use in a classroom to accommodate 10-50 students. The Classroom Supply Set Includes enough of each item for each student to use in an individual class.

- **Student Kits:**
    - **Benefits:** Handing out individual items, particularly the small ones, can take far more time than providing each student with what they need for their use throughout the curriculum. Students have a place to store their ongoing projects

and materials (their kit box). Kits also allow the student to go at their own pace, explore the unfamiliar items and play with materials.

- The full list of materials is in the Supplies List, with links to examples. Here are a few additional notes:
  - A small plastic compartment box with a lid is packed with the small parts. This "Tiny box", labeled with a contents list, is packed into a cardboard "Student Kit" box that contains the larger materials, and is, in turn, labeled with the full contents list. At the end of the Curriculum, students return the kit box with all listed items, to be easily refurbished for new students.
  - Contents and packing instructions are listed here.
  - Box labels include a place for the student's name. Putting a name label on the side edge of the box assists with identifying kits when they are stacked for storage.
  - Label templates are here: Label Templates

Class Sets: 30 of each item, or fewer, depending on class size

- **Batteries** - set of AA, AAA, 9V and CR2032 (coin) batteries
  - Note: when packing batteries, make sure the ends don't touch each other, or come into contact with resistors, wires, or other objects. Stacking coin batteries causes them to drain each other very quickly. Battery cases are recommended, or batteries can be stored in a box, all standing upright and held in that position by crumpled paper. Coin batteries can be stored by taping them flat on a piece of cardboard.

- **Wood and Metal Blocks**
- **Multimeters**
- **Gloves** (nitrile exam gloves - not used as gloves)
- **Painter's tape**

Note: Having a few extras on hand of all supplies is recommended.

III. Kit Parts

Part List with example URL links for supplies

Item (with URL)	Quantity and Notes
Blank Breadboard	2 per student, one for pre-built board and one blank
BasicBoard	Built from other items on this list, one per student
Arduino	1 per student, installed on Basicboard
Needlenose Pliers	1 per student
Wire Strippers	1 per student
12" Leashes	4 per student
USB C Cable	1 per student
Temp Sensors	4 per student
Light Sensors	1 per student
LED Bulbs	4 per student Tiny Box (loose), 2 for BasicBoard
Hook up Wire	5 wire bundles per student, wire for basicboard creation, 22 AWG solid core
Hook up Wire	1 wire set for classroom
Tiny Boxes	One per student, holds small supplies
Student Kit box	One per student, holds all student tools/supplies and also the tiny box
Elastic bands	5 per student
Header pins	Strip of (minimum) 12 pins (or sections in groups of 3 to equal 12)
330 Ohm resistors	4 total - 2 packed in Tiny Box, 2 to create BasiscBoard
1K ohm resistors	1 per student
10K ohm resistors	1 per student
100K ohm resistors	1 per student
1M ohm resistors	1 per student
AA batteries	Class set
AAA batteries	Class set
9V batteries	Class set
CR2032 batteries	Class set
Metal Blocks*	Class set

Wood Blocks*	Class set
gloves	Class set of at least one per student (not used as gloves)
Mulitmeters	Class Set
Painter's tape	Class set: One or more roll to be shared
Small Label Stickers	Blank stickers for printing small labels for Tiny Box
Larger Label Stickers	Blank stickers for printing Kit box and Class Set stickers

A few notes on materials:
*The wooden blocks and metal blocks need to be identical in shape in order to properly work in the curriculum. When we were originally designing the lesson, we had to reach out to local woodworkers and smiths. We gave examples of blocks that you can order, but please note that the lesson requires blocks that are identical in all dimensions. We suggest reaching out to your local tradespeople to inquire about custom blocks. It will be far easier to have the blocks made this way than trying to match from two different vendors. Also, the shape can be whatever you like so long as they are both the same.

IV. Soldering

Safety

How to Solder Safely | Soldering This video was created by an independent content creator and covers everything from how to properly hold a soldering iron to how to check and make sure that the solder you are using contains lead or not. If you are using a lead-based solder, you must perform your soldering in a room with proper ventilation. If you are using solder that is *not* lead-based, a table-top fume extractor should be adequate. It is still very important to ensure that the room you are working in has minimal ventilation (open windows and doors, at the very least).

The Arduino is delivered unassembled, so there will likely be some soldering involved before you can use the Arduino. This is also true for the Light Sensor. In this section you will find instructions for soldering the header pins onto the Arduino. The same method can be used for the Light Sensor pins. If you are not familiar with soldering, we have collected several resources here:

1. Getting Jiggy with Hannah and Amber This video was created by the EdEon STEM Learning group in cooperation with Twiggs Space Labs. The video features instructions for how to assemble a Jiggy Bot (sold by Twiggs Space Labs), but the advice and safety suggestions are applicable to any soldering work.

2. How To - Solder Pin Headers to an Arduino Pro Mini This video was created by an independent content creator and is useful because it instructs soldering for the same Arduino we use in the LbyM curriculum. **Note:** This video shows soldering happening with the Arduino situated on a breadboard, but that board is not connected to any power source. Be sure to solder using a breadboard that is unplugged and disconnected from any electrical sources.
3. Helping Hands are devices that can help hold onto parts as you solder. This is an example of one we have found useful.
4. Here is an example of a soldering iron to purchase if you do not already have one. Be sure to have a stand to place the iron on while you are not using it.
5. Here is an example of solder to use that is lead-free.

V. LbyM Arduino sketch installation

In order to use the Arduino with the web app, the Arduino microcontroller needs to be primed with initial software to be useful for use in the curriculum.
Note: These instructions are geared to I.T. or individuals with knowledge of loading sketches on micro-controllers.

A. Required hardware and software to load the initial Arduino software

- An appropriate Computer OS to run Arduino IDE and Google Chrome (i.e. Windows 10 or 11, Apple MacOS, Linux, Google ChromeOS on a Chromebook)
- If using Windows or MacOS, CP210x USB to UART Bridge VCP Drivers are required. They can be obtained from this site.
- Google Chrome browser (Microsoft Edge can also be used)
- A supported Arduino micro-controller (At this time, only the Adafruit Metro Mini V2 is tested and used in the curriculum).
- Arduino IDE 1.8.x (Use the legacy version as this has been fully tested, the newer 2.x has not been tested).
- The initial LbyM Arduino sketch as a zip file. The zip file can be downloaded from here.
- USB A to USB C cable. USB C to USB C cable may be used as well.

B. Installation of drivers

If the computer used to load the initial LbyM Arduino sketch is running Windows OS (10 or 11) or Apple MacOS, software drivers will be required before installation. Make sure the user account has administrative privileges before proceeding. Below are the steps for installing the

driver on a Windows-based PC. Drivers are required for **ANY** computer that is using an Arduino for initial programming and/or experiments, including student computers using Windows (10 or 11) and Apple MacOS.

1. If you haven't downloaded the drivers, take the opportunity to download the drivers from the Silicon Labs site here.
2. Press downloads to show all the downloads
3. Look for the "CP210x Windows Drivers with Serial Enumerator" driver.
4. Click the link to download the driver zip file.
5. Unzip the zip file to a folder.
6. Navigate to the newly unzipped folder and look for the file CP210xVCPInstaller_x64.exe
7. Run the file. Windows UAC may require elevation of privileges to install. Click "Yes" to continue the installation.
8. Click "Next" to continue
9. Accept the agreement by clicking the bubble selector and select "Next"
10. Installer will install the drivers
11. Click "Finish" to complete

Note: Windows Arm drivers are only available in the Windows Universal driver package. Follow the CP210x_Universal_Windows_Driver_ReleaseNotes.txt file in the folder to proceed with installation on Arm hardware.

Below are the instructions for installing drivers on Apple MacOS.

1. If you haven't downloaded the drivers, take the opportunity to download the drivers from the Silicon Labs site here.
2. Press downloads to show all the downloads
3. Look for the "CP210x VCP Mac OSX Driver" driver
4. Unzip the "Mac_OSX_VCP_Driver.zip" file and navigate to the extracted subfolder
5. Mount the DMG file and double click on Silicon Labs VCP Driver. On MacOS 10.13 and later, the installation of the driver may be blocked.
6. To unblock, open the System Preferences Security & Privacy pane and unblock the system extension. See Apple Technical Note TN2459 "User-Approved Kernel Extension Loading" for more information.

Note: Linux and ChromeOS DO NOT require driver installation.

C. Installation of Arduino IDE

The Arduino IDE is required to load the initial sketch. To install the Arduino IDE follow the instructions below:

1. In a browser, navigate to the Arduino IDE download site
2. Scroll down the web page until you find the section “Legacy IDE (1.8.X)”
3. Look to the right of the section and download the installer most appropriate for the operating system that is in use.
4. Run the installer file
   a. For Windows users, run the arduino-1.8.19-windows.exe installer file. Windows UAC may require elevation of privileges to continue. Follow the screen instruction to complete installation
   b. For MacOS users, download and unzip the arduino-1.8.19-macosx.zip file. Copy the Arduino application bundle into the Applications folder.
5. Once complete, proceed to next section for obtaining the LbyM8 Arduino sketch

D. Obtaining and unzipping the initial LbyM8 Arduino sketch

To obtain to LbyM8 Arduino sketch, download the zip file from here. Unzip the file to a convenient location. This could be in your downloads folder or on your desktop. The folder should contain several files. The file needed later is the lbymDriver2021.ino Arduino IDE file. Locate it, and make sure you can easily find it later.

E. Configuring Arduino IDE

Start the Arduino IDE from the Start Menu for Windows users or the Applications folder for MacOS users. The repository to Adafruit’s git repository needs to be added. To begin, navigate to File>Preferences. This will open a window with preferences for the program.

Look for the section called “Additional Boards Manager URLs.” Inside the text box, type the following URL to the Adafruit Boards definition json file:
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

Click the “OK” button to complete this part of the configuration.

Assuming the computer is online, the Arduino IDE will download Adafruit’s board definitions from their git site. Next, we need to ensure the Arduino board libraries are added to the installation.

Navigate to Tools>Board>Board Manager.
Once in the “Board Manager…” window, make sure that Arduino AVR Boards, Arduino MegaAVR Boards, and Adafruit AVR Boards libraries are installed. If not, search for each library, select it, and click install to download and install the library in the Arduino IDE environment. Once installed, the libraries are instantly usable.

Click the “Close” button to complete this part of the installation.

Next make sure the Adafruit development libraries are installed. Navigate Sketch>Include Library>Manage Libraries…

Search for Adafruit by typing in the search bar on the top of the window. It will take time to populate the window. Make sure that the following libraries are installed: AdaFruit BusIO, Adafruit Circuit Playground, AdaFruit GFX Library, Adafruit GPS library, Adafruit ILI9341, Adafruit LED Backpack Library, Adafruit Sleepydog Library, Adafruit STMPE610 library, Adafruit Touchscreen, Adafruit Zero DMA library, Adafruit Zero FFT library, and Adafruit Zero PDM library.

The Arduino IDE interface is sluggish. Please allow some time for the interface to respond to commands. Once all libraries are installed, click the “Close” button to complete the configuration. Configuration only needs to be completed once for the first time.

F. Loading the initial LbyM9 Arduino sketch

Next, load the LbyM9 Arduino sketch by navigating to File>Open. Navigate to where the sketch was unzipped from the prior step. Locate the lbymDriver2021.ino file. Once selected, click the “Open” button to open the sketch.

Your Arduino IDE sketch should now be open with three tabs in the interface. Next, we need to attach the Adafruit Metro Mini V2 to the computer using either a USB A to USB C or USB C to USB C data cable. A green power-on LED should be lit.

From within the Arduino IDE, select Tools>Adafruit Boards>Adafruit Metro. This library is needed to flash the board with the sketch.

Next, select the communications port by navigating Tools>Port>COMx. On a Windows platform, this may look like COMx, where x is the index number for the COM port. Other platforms may look different. For example, on a Linux or Android system, this may look like /dev/usbTTYx.

Check the code by clicking on the verify button; it is the little circle button with a check mark in it. The IDE will now test-compile the code. Should the test be completed without errors, a display should be indicated in the terminal window indicating how much onboard memory is used. Any errors should be resolved prior to continuing.

Next, write the sketch to the Arduino by clicking the upload button. The process will be similar to the verification process. The code will be compiled, written to the non-volatile portion of the on-board memory, and output written to the terminal. The interface will indicate when uploading is done. Once done, testing is usually done to validate the load process.

G. Testing the installation

Open Google Chrome or a Chromium based browser and type http://app.lbym.org in the address bar. Press enter to navigate to the web site. Assuming the Arduino is still connected to the computer via USB cable, press the “Connect” button.

Another window should appear with the COM port of the Arduino listed. Select “CP2102N USB to UART Bridge Controller (COMx).” Click Connect to complete connecting the browser to the Arduino.

If you are testing the Arduino without being attached to a basic board, follow the instructions in this section. In the terminal window, type the following:

`ob1on` [enter]

A red LED on the Arduino should now light up. This is an initial indicator that the Arduino is now properly loaded with the sketch and commands are being sent from the browser. Communication is happening, and the Arduino is now initialized and ready to be used. Type:

`ob1off` [enter]

in the terminal to turn off the red LED. Initial testing is done.

If you have the Arduino attached to a completed basic board. You can test the initial configuration of the basic board before deployment by issuing the following commands in the terminal:

`dp2on` [press enter]

The red LED will turn on.

`dp2off` [press enter]

The red LED will turn off.

`dp3on` [press enter]

The white LED will turn on.

`dp3off` [press enter]

The white LED will turn off. Once the procedure is complete, the basic board with Arduino is now ready for student use in the curriculum.

VI. Building the BasicBoard

Tools you will need:

- Bread board
- Arduino
- Solid core wire (red, black, yellow)
    - **Note:** DO NOT use stranded wire.
- Uninsulated Copper wire
- LEDs
- 330 ohm resistors
- Wire cutters
- Wire strippers
- Needle nose pliers

Step one: When placing the Arduino make sure the breadboard is horizontal and the blue ground rail on top of the red power rail. The Arduino will be placed with the usb connector facing the top of the board, the first pin on the left will be placed on the 8th row down and the 2nd column from the left.

Step two: Using the wire cutters you will need to cut:

- 3 black wires 2.5 cm long
- 2 red wires, one 2 cm and one 4 cm long
- 1 yellow wire 5.5 cm long
- 2 copper wires 2 cm long
- Both legs of 2 resistors in half

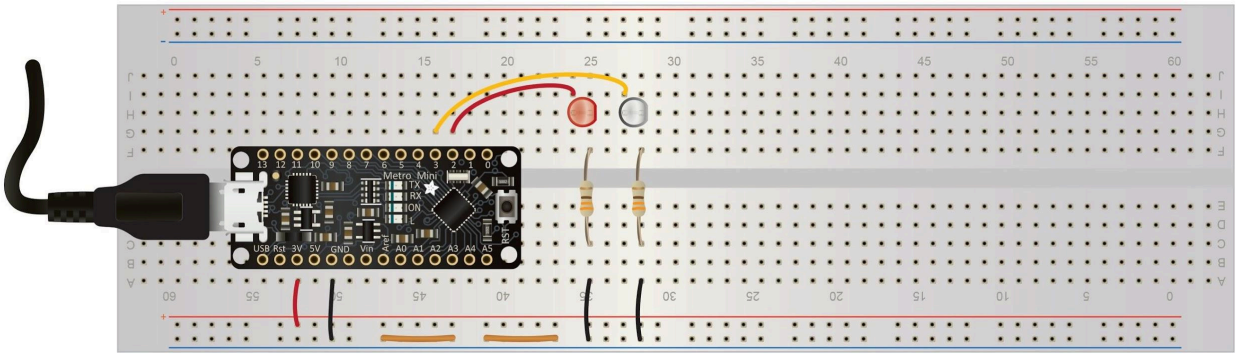
Step three: Using the wire strippers, strip approximately 0.5 cm of insulation off both ends of each wire

Step four: placing the wires, LEDs and resistors. Use pliers to bend and place wire when necessary. The board should now be vertical with the blue ground rail on the left.

1. Place the shorter red wire from the 3V pin on the left side of the Arduino to the red power rail on the left
2. Place one black wires from the first GND pin on the left side to the blue ground rail on the left
3. Place the longer red wire from pin 2 on the right side to the 24th row (if it's one up or down that is ok)
4. Place a red LED with the longer leg in the same row as the red wire and the shorter leg one row down.
5. Place one leg of a resistor in the same row as the short leg of the red LED and the other leg in the same row on the opposite side of the break of the bread board.
6. Place one black wire from the same row of the resistor to the blue ground rail on the left side

7. Place the yellow wire from pin 3 on the right side to the 28th row (again if one up or down that is ok)
8. Repeat 4-6 using a white LED
9. Place cooper wires in ground rail on the left side between the black wires

When you've finished placing the grounding wires the board should look like this:



VII. Testing the BasicBoard

Part 1: Laying out LbyM

The purpose of this testing process is to ensure that the standard red and white LEDs on a default BasicBoard, and more fundamentally the pins transmitting signals to them, both operate correctly. This is done by using the LbyM web app to send commands to the Arduino, a schematic of which is displayed in Figure 1.

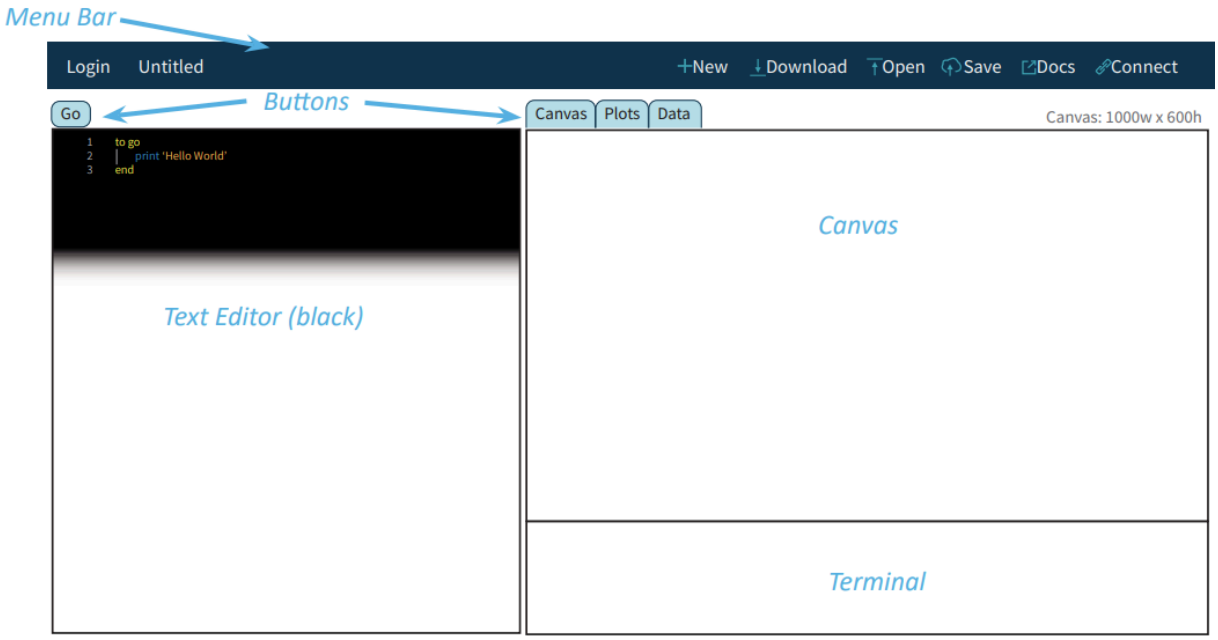


Figure 1: LbyM webapp layout (Source: Experiment UNIT Light and Energy)

Part 2: Codifying the Code

```
71  to test
72    dp2on
73    wait 10
74    dp2off
75    dp3on
76    wait 10
77    dp3off
78  end
```

Figure 2: Test code

*The numbers on the left merely indicate where in the Text Editor each line is, which doesn't matter, although I recommend just placing it below the rest of the code.

1. Begin by going to the LbyM web app, which is currently hosted at **this link**.
2. Log into your account via **Login** in the top left.
3. In the upper middle, click on **+New**.
4. In the list this opens, **LEDs** will provide the ideal code layout for us to work with, so click on that and wait for it to open.

5. Enter Figure 2's code as described in the dark section labeled **Text Editor** from Figure 1:

    1. Scroll below the rest of the code to an empty spot
    2. Type "**to test**"
    3. Press Enter to get to the next line
    4. Press Tab, then type "**dp2on**"
    5. Press Enter to get to the next line, your cursor should be spaced slightly to the right and have the vertical line to its left you can see from positions 72 to 77 in Figure 2.
    6. Type: "**wait 10**"
    7. Repeat this process with until after you have typed "**dp3off**"
    8. Press Enter and then Backspace once to get to the same horizontal position as "**to test**"
    9. Type: "**end**"
6. Click on **Untitled** in the upper left and create a name; "BasicBoard Testing," for instance.
7. Press **Save** in the upper right corner.
8. If you click on your profile image in the top left, this should now be in your program list for future use.

Quick Overview

- **to test:** A header for the code which indicates that you type “test” in the Terminal window (see Figure 1) to initiate it.
- **dp2on:** A command which instructs the pin labeled “2” to begin transmitting power, which on the BasicBoard should be connected to the longer (anode / positive) lead on the red LED and cause it to light up.
- **dp3on:** A command which instructs the pin labeled “3” to begin transmitting power, which on the BasicBoard should be connected to the longer (anode / positive) lead on the white LED and cause it to light up.
- **dp2off:** A command which instructs the pin labeled “2” to cease transmitting power, causing the red LED to darken.
- **dp3off:** A command which instructs the pin labeled “3” to cease transmitting power, causing the white LED to darken.
- **wait 10:** Instructs the code to wait for a time period of 10 deciseconds before continuing with the next command. This corresponds to a time period of one second, but you may adjust this if it is too long or short.

Part 3: Evaluating the LEDs

Now with the code prepared, we can begin our testing. As a reference point, here is a basic schematic of a BasicBoard:

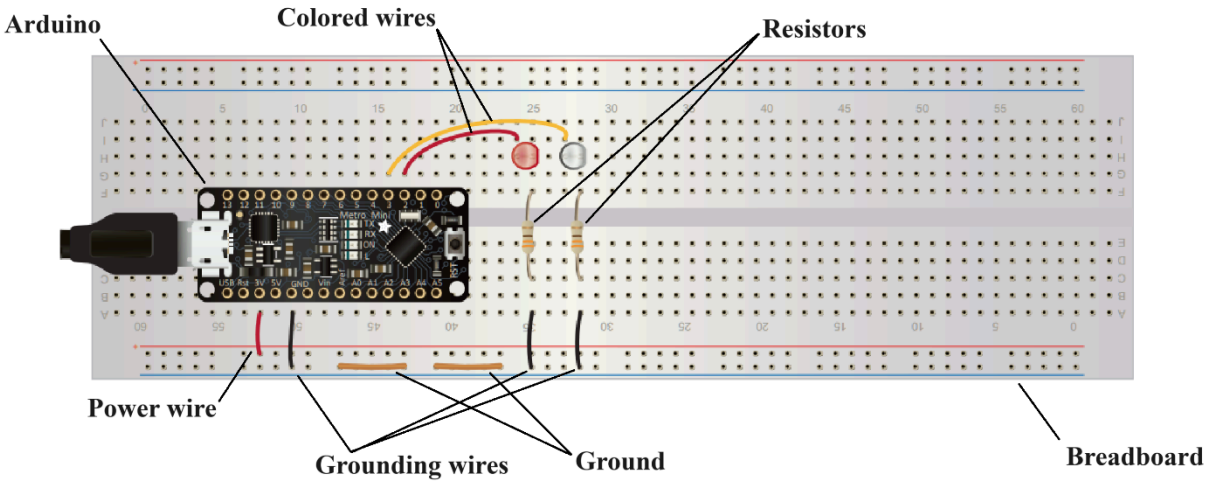


Figure 3: BasicBoard schematic (Unaltered image source: UNIT 3 Doing Science with Sensors)

1. First, check to make sure that the power and grounding wires and ground wires are oriented as shown in Figure 3 above: with red wires in a hole next to the red and black wires next to the blue line. If it is backwards, dismantle the entire BasicBoard and reassemble as shown.
2. Open the LbyM program you saved with the testing code described above.

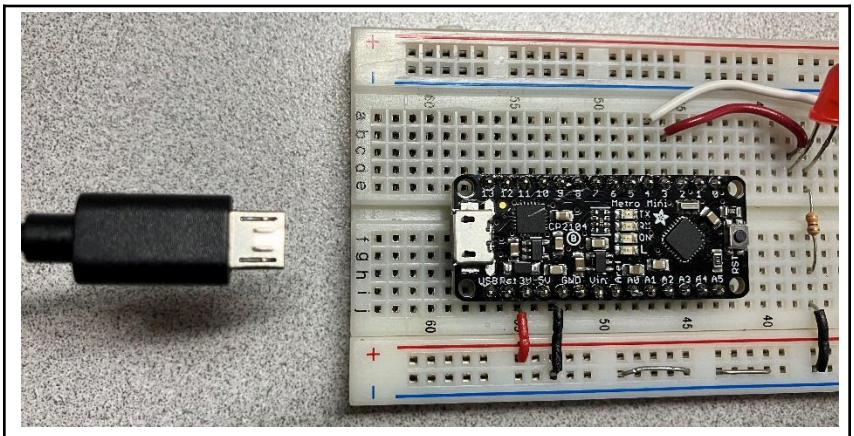


Figure 4: USB orientation

3. Take a USB cable and plug it into the Arduino on the BasicBoard, underside up as shown above in Figure 4, with the other end in your computer's USB port.

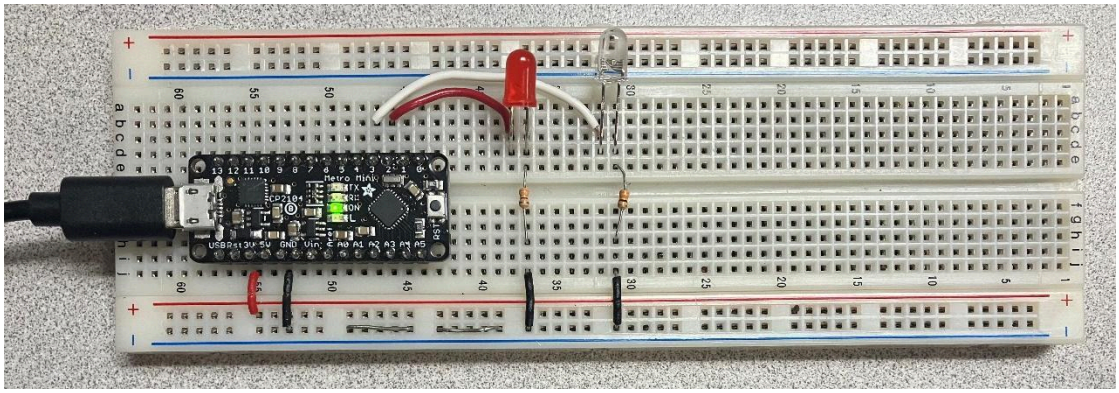


Figure 5: Correctly lit Arduino

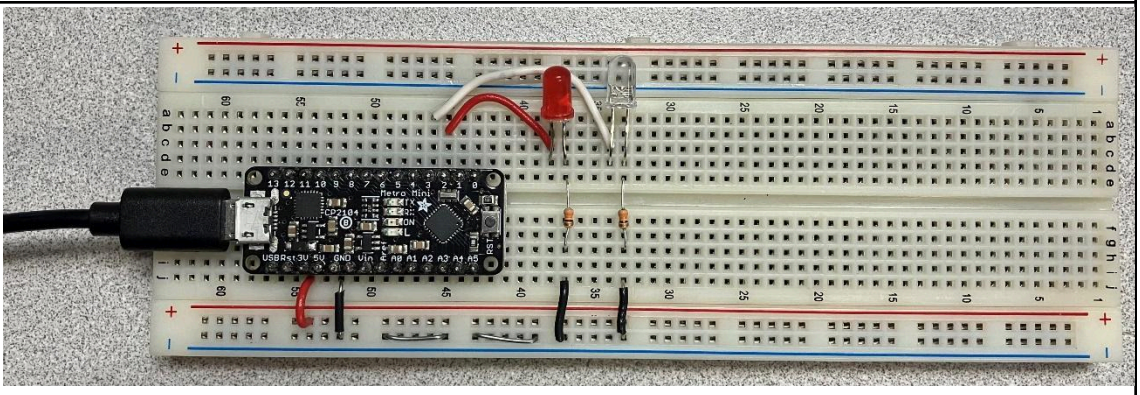


Figure 6: Incorrectly lit Arduino

4. An Arduino in ideal condition should have a green light labeled **ON** activate as seen in Figure 5 once the USB is plugged in.
    a. If the green light doesn't turn on as shown in Figure 6, continue with the rest of the testing. If it passes the rest of the testing, place it in a pile labeled something like "fully works except for green status light".

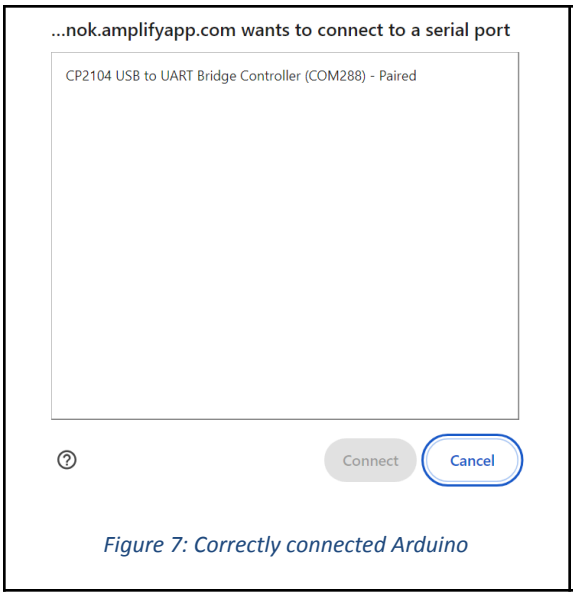


Figure 7: Correctly connected Arduino

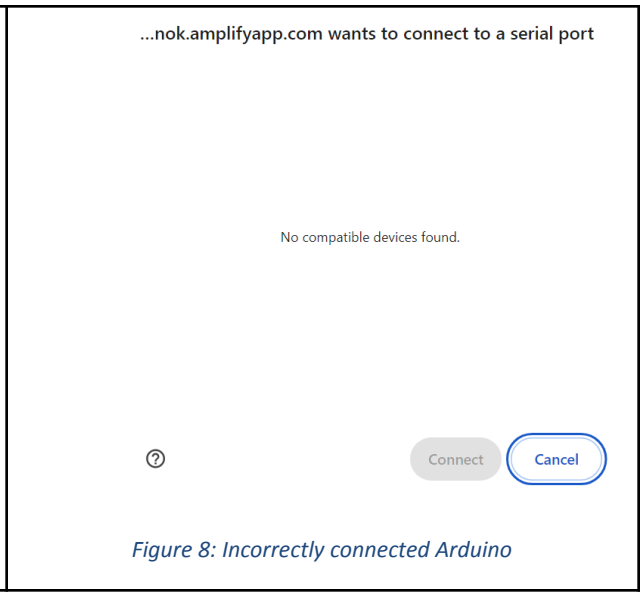


Figure 8: Incorrectly connected Arduino

5. With the Arduino plugged in, click **Connect** in the upper right of the LbyM webapp. This will open the window shown in Figures 7 and 8.
6. If the Arduino properly connects as shown in Figure 7, continue to Step 6.
    a. If the Arduino does not properly connect as shown in Figure 8, see Troubleshooting in Part 4.
7. Click on the listed option and press **Connect**, which will trigger a red light labeled **L** to flash briefly and also change **Connect** to **Disconnect**.

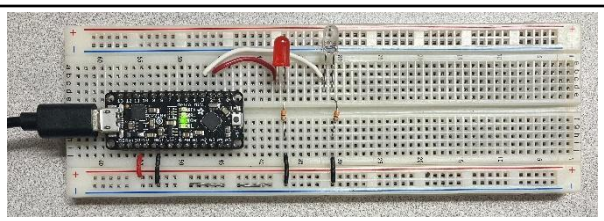


Figure 9: Initial state

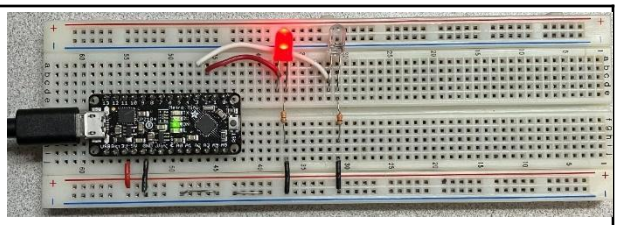


Figure 10: Red LED activated

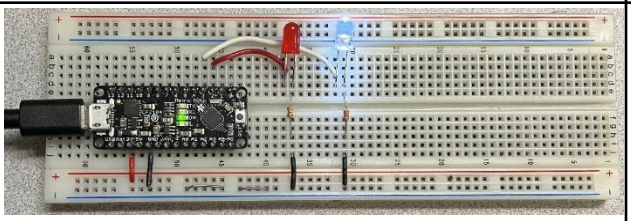


Figure 11: White LED activated

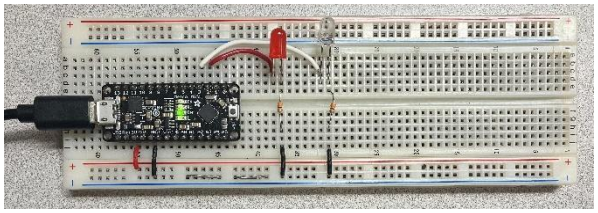


Figure 12: Final state

8. You are now prepared to test the BasicBoard. In the Terminal window (Figure 1), type the word “test” to initiate your code, which should activate the red and white LEDs in the pattern seen in Figures 9-12. If it does, the Arduino has fulfilled its parameters and earned the singular honor of being placed in the “tested and working” pile. If either or both of the LEDs don’t activate, see Troubleshooting in Part 4.
9. Now, repeat with the rest of that pile.

Part 4: Troubleshooting

For this test, our failure condition consists of either the red LED, white LED, or both not lighting up when they are supposed to. The reasons for this happening boil down to five basic causes, which we'll go through:

1. Issues with the code
    i. This is the simplest problem, which you can check by seeing if the code matches what is given in Figure 2.

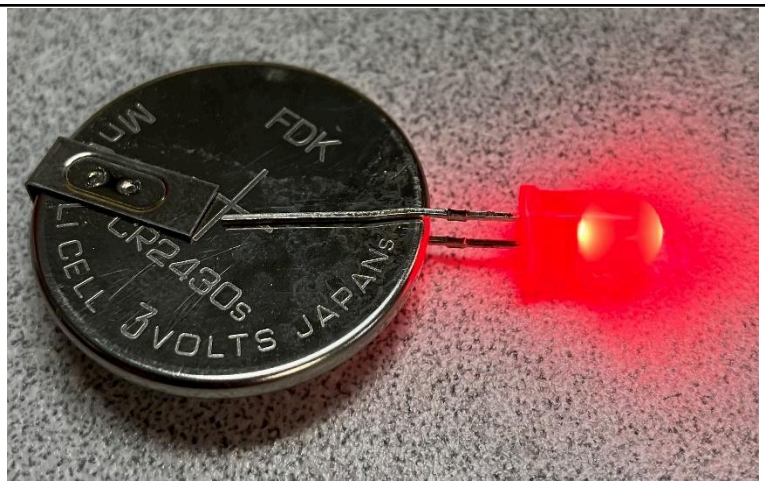


Figure 13: Testing LED with coin battery

2. Issues with the LEDs
    i. Remove the LED that didn't light up and check to ensure that the longer lead is linked to the colored wire and shorter lead is linked to the resistor.
    ii. If they are connected as intended, check with a coin battery as shown in Figure 12 above, with the longer lead on the labeled, positive side.
        1. If it does not light up, flip the LED to double-check. If the LED lights up, reinstall into the BasicBoard as per Figure 4 and continue troubleshooting elsewhere.
        2. If it doesn't, throw away the faulty LED and replace with one of the same color and retest. If it continues to fail to light up, continue troubleshooting elsewhere.

3. Issues with the connections (colored wires, grounding wires, or resistors)
    i. Review the BasicBoard image in Figure 4 and ensure that the connections in the board you test are accurately linking up as shown.
    ii. Common problems may include:
        1. Some connections are simply missing, such as the grounding wires.
        2. Some parts are placed incorrectly, such as the colored wires and resistors being aligned with the same LED lead, see Figure 14.
        3. The power wire being connected to ground (next to blue line) or ground wire(s) being connected to power (next to red line).
        4. The colored wires not being correctly connected to the “2” or “3” pins.
4. Issues with the Arduino
    i. If the LEDs and connections appear to be accurate (along with the code), swap the Arduino with another from a BasicBoard (which I will label BasicBoard X for clarity) which was successfully tested and retest both BasicBoards.
    ii. If the LEDs and connections appear to be accurate, swap the test Arduino with one from a BasicBoard previously established as working and test both BasicBoards. For clarity, I will refer to the board you are testing as BasicBoard A (with Arduino A) and the one which was successfully tested before as BasicBoard B (with Arduino B)
        1. If BasicBoard B with Arduino A succeeds while BasicBoard A with Arduino B fails, then the Arduino is not the problem and you should continue troubleshooting elsewhere.
        2. If BasicBoard A with Arduino B succeeds while BasicBoard B with Arduino A fails, then the Arduino you are testing (A) is not working. Place it in the “not working” pile and either dismantle BasicBoard A or replace it with a new Arduino if there is a ready supply and retest.
5. Issues with the breadboard
    i. If the code, LEDs, connections, and Arduino all appear to be working, the problem may be the breadboard. If there are blank breadboards available, dismantle the BasicBoard and reassemble it there with the same layout as in Figure 4, then retest.
6. If all is lost
    i. If all five cases have been tested and there still appears to be an unknown problem, place it in the “not working” pile and move on to the next one.