Near-Field Communication Door Lock

Stefen Sharkey

CSC 202–Fall 2017

Table of Contents

Overview

Section 1.1 – Abstract

Every day, people face the need to unlock a door—generally, the front door to their house and their car door—and are inconvienced with finding their keys and having to manually unlocking the door, sometimes even needing to put down what they are carrying to do so. While many newer houses have keypad entry systems, they still require the person using it to have a free hand. This near field communication (NFC), door lock focuses on not requiring a spare hand to unlock a door this is installed in by allowing the user to bring an NFC-enabled device—credit-card sized NFC chips, keyring chips, NFC stickers, etc.—near the chip reader. This allows for significantly quicker and safer entry.

Section 1.2 – Under the Hood

On a more technical level, the NFC reader is a shield attached to an Arduino Uno. When signaled to, the NFC reader awaits communication with another NFC-enabled chip. When successful communication has been made, the NFC chip's ID is sent via serial communication to the Raspberry Pi. The Raspberry Pi looks the card up in a local SQLite database and determines what type of card it is—user, administrator, programming, deleting, or master—and sends the type ID to the Arduino and Thunderbird12 via serial communication.

If the card is a user or administrator card, the Thunderbird will blink the attached light-emitting diode (LED) green three times, and make a higher pitched tone on the attached piezoelectric, speaker three times; and activate the attached relay, which a lock-style solenoid, powered by a 12V DC power adapter, is attached to, for five seconds. The Arduino will also welcome the user by printing out their name on the attached liquid crystal display (LCD), communicated via Inter-Integrated Circuit (I^2C), or their card ID if a name doesn't exist.

If the card is a programming card, the Thunderbird will turn the attached LED yellow and the Arduino will notify the user that the system is in programming mode via the attached LCD. If any card is read by the NFC shield, its ID will be sent to the Raspberry Pi. If the card is not already in the system, then the card will be inserted into the SQLite database as a user. A status message will be sent back to the Arduino if a card is or is not learned. The system will stay in programming mode until the programming card is brought near near the NFC shield again.

If the card is a deleting card, the Thunderbird will turn the attached LED purple and the Arduino will notify the user that the system is in deleting mode via the attached LCD. If any card is read by the NFC shield, its ID will be sent to the Raspberry Pi. If the card is in the system, and it is not a programming, deleting, or master card, then the card will be removed from the SQLite database. A status message will be sent back to the Arduino if a card is or is not deleted. The system will stay in deleting mode until the deleting card is brought near near the NFC shield again.

If the card is a master card, then the Raspberry Pi will insert the default cards into the SQLite database.

If the card is not in the SQLite database, then the Thunderbird immediately deactivates the relay that the lock-style solenoid is attached to. The Arduino notifies the user via the attached LCD.

Section 2 – Prerequisites

Section 2.1 – Hardware

Section 2.1.1 – Required

- Thunderbird12 - http://www.evbplus.com/ThunderBird12_9s12/ThunderBird12_9s12.html
- Raspberry Pi 2 Model B – https://www.raspberrypi.org/products/raspberry-pi-2-model-b/
- Arduino/Genuino Uno Rev3 – https://store.arduino.cc/arduino-uno-rev3 or https://store.arduino.cc/genuino-uno-rev3
- Adafruit PN532 NFC/RFID Controller Shield – https://www.adafruit.com/product/789
- Adafruit Lock-style Solenoid – https://www.adafruit.com/products/1512
- RGB LED *(substitutable with three single LEDs of respective color, or variation thereof)*
- Piezoelectric speaker
- SRD-05VDC-SL-C Relay
- DC Power Supply 12V 0.75A *(Output amperage may exceed 0.75A)*
- USB Type-B to USB Type-A cable
- One 1x6 female-to-male header
- Two 1x8 female-to-male headers
- One 1x10 female-to-male header

Section 2.1.2 – Required for setup

- USB keyboard and mouse
- Internet
- Ethernet cable *(substitutable with USB Wi-Fi adapter)*
- HDMI cable *(substitutable with HDMI-to-DVI-I cable with compatible monitor)*
- HDMI-compatible monitor *(substitutable with DVI-I-compatible monitor with compatible cable)*
- DC power supply 5V 1.5A *(Output amperage may exceed 1.5A)*
- Soldering iron
- Solder
- Wire cutter
- Wire stripper

Section 2.2 – Software

- Freescale CodeWarrior – http://cache.freescale.com/lgfiles/devsuites/HC12/CW_HC12_v5.1_SPECIAL.exe
- NOOBS – https://www.raspberrypi.org/downloads/noobs/
- Arduino IDE – https://www.arduino.cc/en/Main/Software
- Adafruit-PN532 – https://github.com/adafruit/Adafruit-PN532

- New Liquid Crystal - https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads/
- LBE_Thunderbird – https://github.com/StefenSharkey/NFCDoorLock/tree/master/LBE_Thunderbird
- Arduino project files – https://github.com/StefenSharkey/NFCDoorLock/tree/master/DoorLock_Arduino
- Thunderbird project files – https://github.com/StefenSharkey/NFCDoorLock/tree/master/DoorLock_Thunderbird
- Raspberry Pi project files – https://github.com/StefenSharkey/NFCDoorLock/blob/master/DoorLock_RaspberryPi.py

Section 3 – Setting up the Lock-Style Solenoid

A solenoid is a type of cylindrical magnet that can be used to move or hold another piece of metal inside of it. The most common applications of solenoids are car door locks. This specific solenoid, the lock-style solenoid, has a metal rod inside that is shaped like a door latch which can then, in turn, act as a lock for a door. It also has a spring keeps the rod in the outward position until sufficient electrical current is applied, in which case it retracts into the body.

1. Obtain any unused DC power supply 12V 0.75A. The amperage can be higher, but is unnecessary. Also obtain two flexible jumper wires with at least one male end on each wire, and one with two male ends. It is recommended to use one red and two black wires, with one of the black wires having two male ends, to easily determine the voltage polarity for future use.
2. Cut off the adapter at the end of the DC power supply.
3. Cut off one end of each jumper wire, leaving one male end on each.
4. Using an appropriately sized wire stripper, strip the red and one black jumper wire and the positive and negative leads of the DC power supply such that ½" of metal wire is shown.
5. Grabbing the multimeter, determine either by experience, or by consulting the internet and/or the multimeter's manual, where to plug both probes. For most multimeters, the black probe will plug into the "COM" port and the red probe will plug into the "VΩmA" port.
6. By turning the dial on the multimeter, find the section labled "V⎓," "VDC," or variation thereof, and switch the multimeter to the 20.
7. Plug in the DC power adapter to its respective power source For most power adapters, this is a standard wall outlet.
   Warning: Exercise caution by keeping exposed wire away from any flammable or conductive objects.
8. As many DC power adapters don't clearly mark what positive or negative leads are, connect one lead to one probe on the multimeter, and the other lead to the other probe.
9. If the number displayed on the multimeter is a positive number, then the lead connected to the red probe is positive and the other lead is the negative. If the number displayed is a negative number, then the lead connected to the black probe is positive and the other lead is negative.
10. Take note of which lead is which, but do not damage the cable in any way by doing so.
11. Unplug the DC power adapter.
12. Put 1" of shrink wrap around both jumper wires, but do not apply heat. Slide the shrink wrap to the male end of the wires to avoid it from shrinking due to the soldering process.

13. Twist the positive lead and one of the jumper wire's exposed ends together. Do the same with the negative lead and the other jumper.
    Note: As previously stated, but going into further detail, red wires are generally reserved for voltage sources—the positive lead—while black wires are generally reserved for grounding—the negative lead. It is highly recommended this practice is followed, as it will reduce the potential for accidentally reversing the polarity and damaging the adapter and/or anything it's connected to by making it obvious what each wire's functionality is.
14. If handy, apply flux to both twisted wire pairs. This is to make soldering easier by more easily allowing solder to stick to the wires.
15. Plug in the soldering iron. It will be assumed that the soldering iron does not have any control panel and plugs directly into an AC voltage source.
    Danger: Soldering irons get dangerously hot and, when exercise is not cautioned, can cause fires, injury, or death. Keep soldering iron away from anything aside from designated stand.
16. When the soldering iron reaches the desired temperature (around 600°F/315°C), apply solder to both twisted wire pairs. For tips on how to correctly solder, consult those who are experienced and/or the internet.
17. Unplug the soldering iron.

18. When the soldered joints cool down, slide the shrink wrap over the joint and apply moderate heat until the shrink wrap has shrunk. For tips on how to correctly use shrink wrap, consult someone experienced and/or the internet. Refer to Figure 3-1 for an example of what the finished shrink wrap should look like.

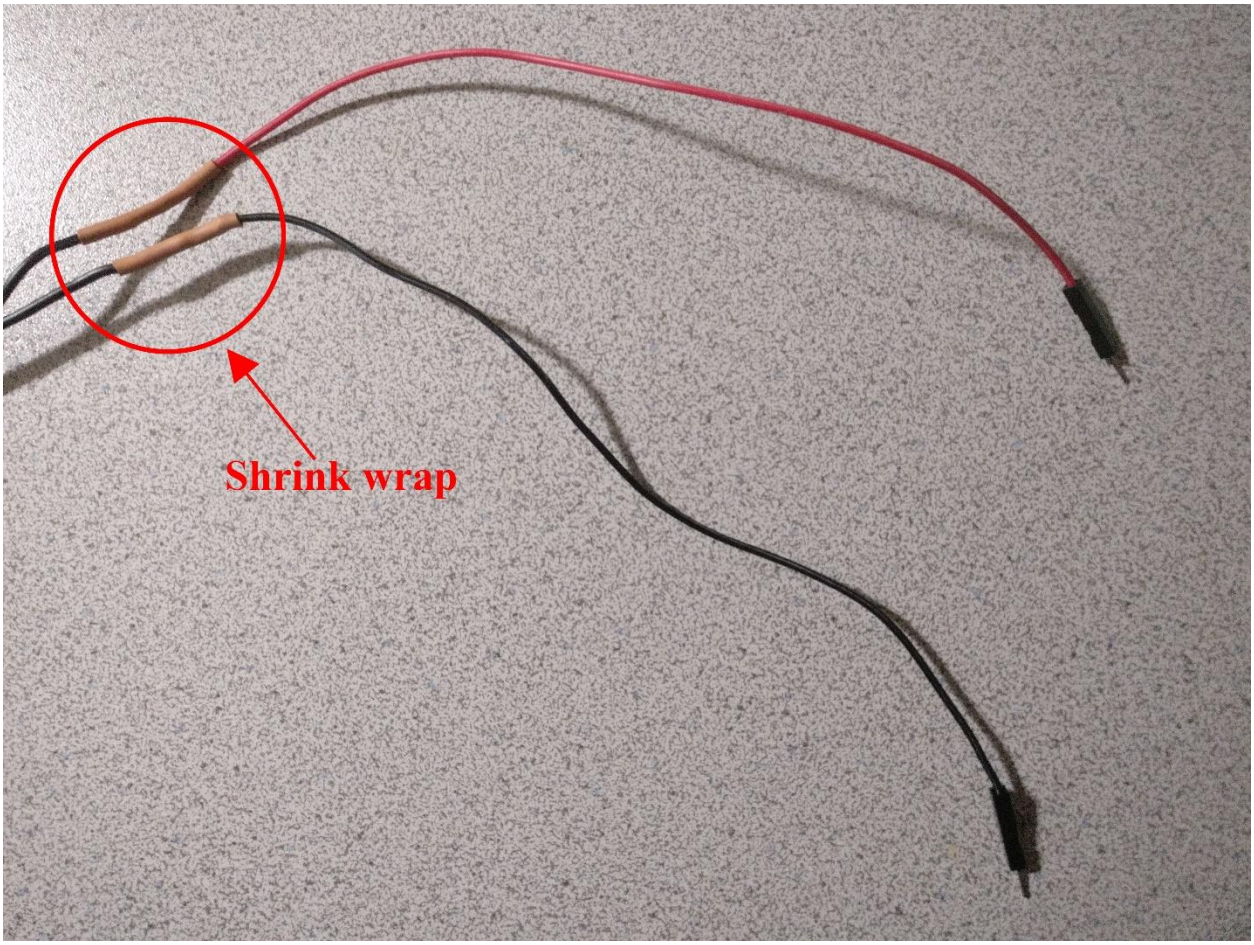


Figure 3-1 – Finished shrink wrap on DC power adapter

19. Plug one end of the unused jumper wire into the blue (negative) side of the solenoid's female connecter. Refer to Figure 3-2 for an example of what the wiring should look like.

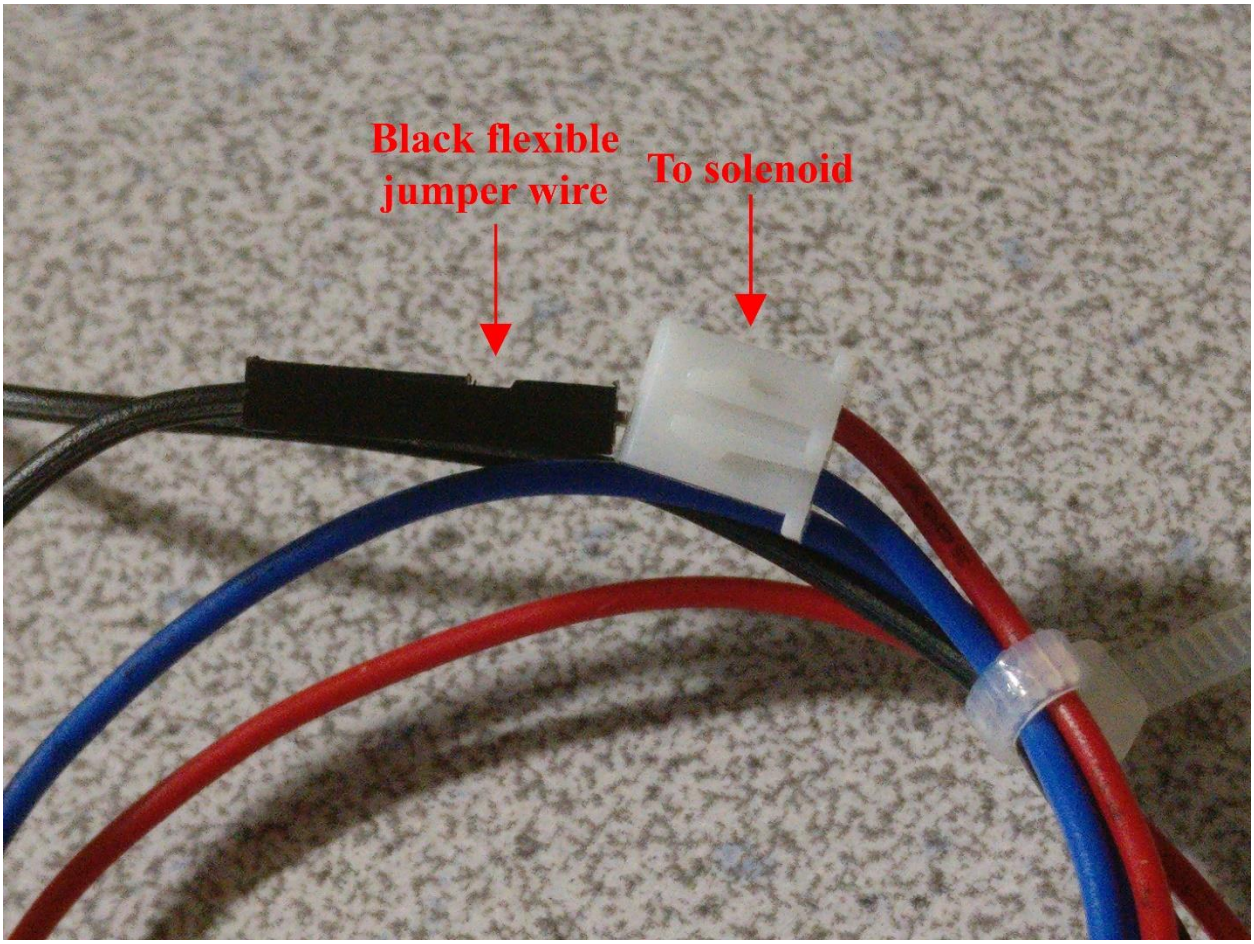


Figure 3-2 – Black flexible jumper wire connected to negative end of solenoid

20. Plug the positive jumper wire soldered to the DC power adapter into the red (positive) side of the solenoid's female connector. Refer to Figure 3-3 for an example of what the wiring should look like.

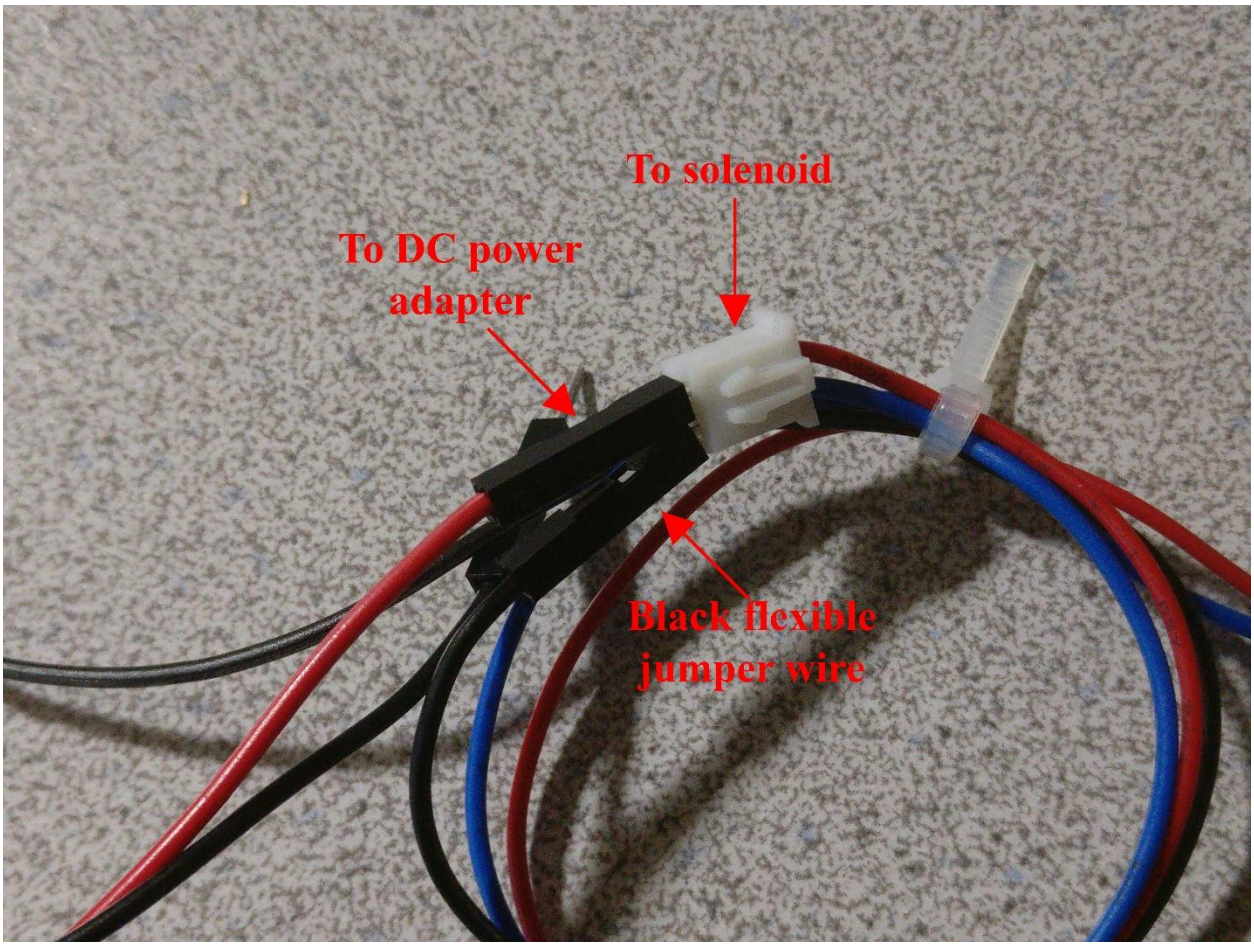


Figure 3-3 – Finished view of solenoid's female connector

Section 4 – Setting up the Adafruit PN532 Shield

The Adafruit PN532 shield for the Arduino is the device used to read NFC chips. NFC is a form of Radio Frequency Identification (RFID), which are commonly used in newer credit cards and certain photo IDs. NFC chips have had varying popularity in the past, with their most notable popularity were in smartphones. It is attached on top of the Arduino Uno, when finished, in order to easily send data to and from the sensor. It works at a range by attaching coils of wire to act as attenae, on both the sending and receiving ends.

1. Plug in the soldering iron. It will be assumed that the soldering iron does not have any control panel and plugs directly into an AC voltage source.
   Danger: Soldering irons get dangerously hot and, when exercise is not cautioned, can cause fires, injury, or death. Keep soldering iron away from anything aside from designated stand.
2. Along the length of the board, along the top and bottom, there will be two rows of holes. The row of holes closest to the outside of the shield will receive the included male-to-male headers. Break off just enough pins to fill in the holes. Figures 3-1 through 3-4 show the completed board.
3. Solder the header on such that the plastic separator is on the underside of the shield and the shorter end is going into the board holes. Repeat this for both sides.
4. Solder the female-to-male headers onto the board such that the female end is on the top of the board. Figures 4-1 through 4-4 show an example of the finished board.

Figure 4-1 – Aerial view of finished Adafruit PN532 shield

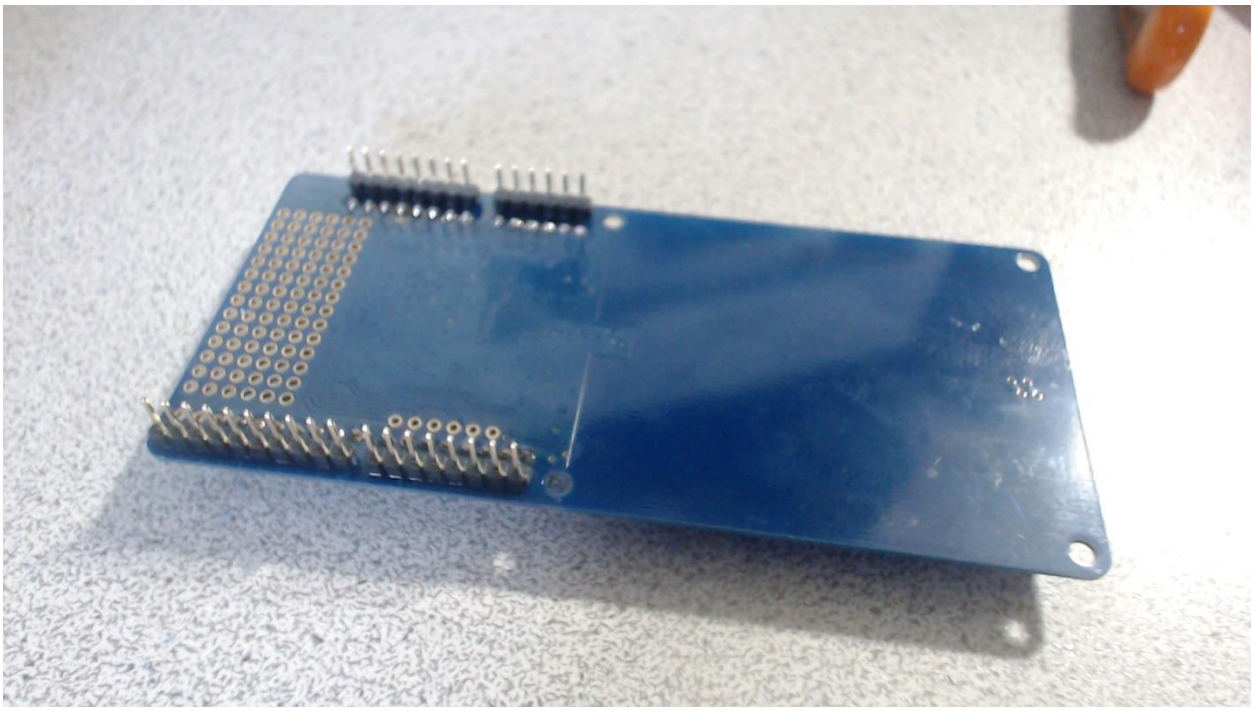


Figure 4-2 – Bottom view of finished Adafruit PN532 shield

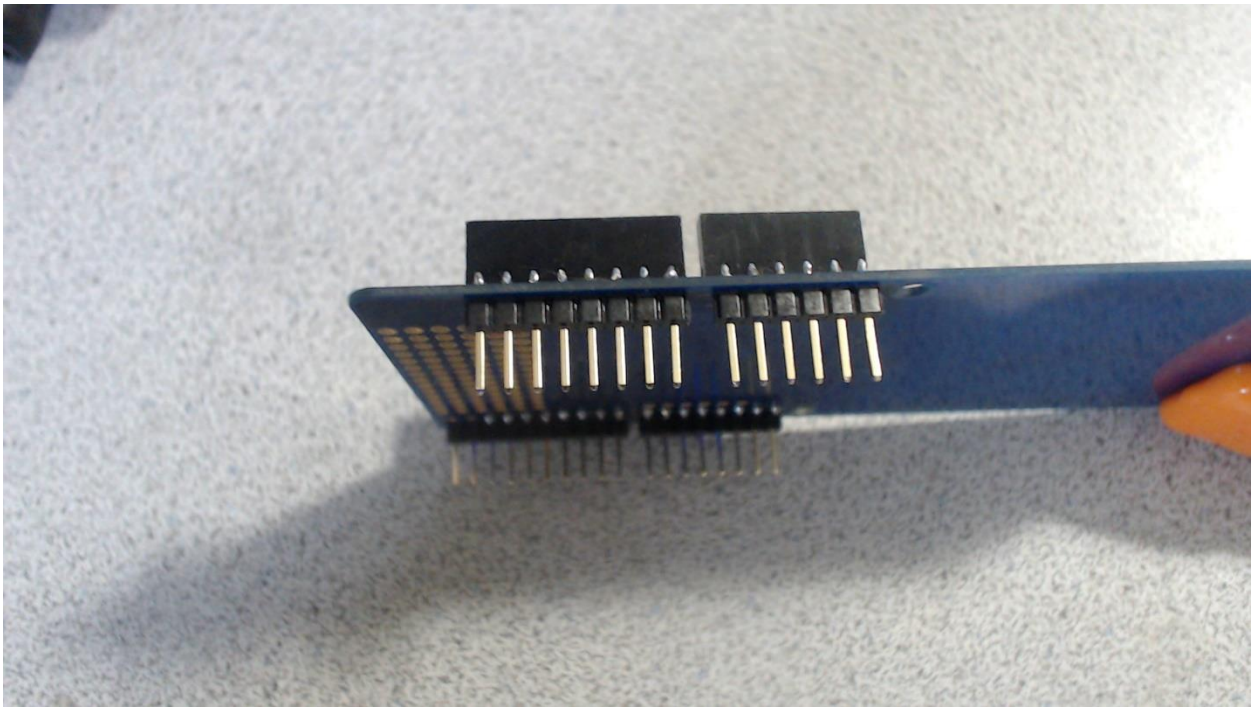


Figure 4-3 – Side view of finished Adafruit PN532 shield

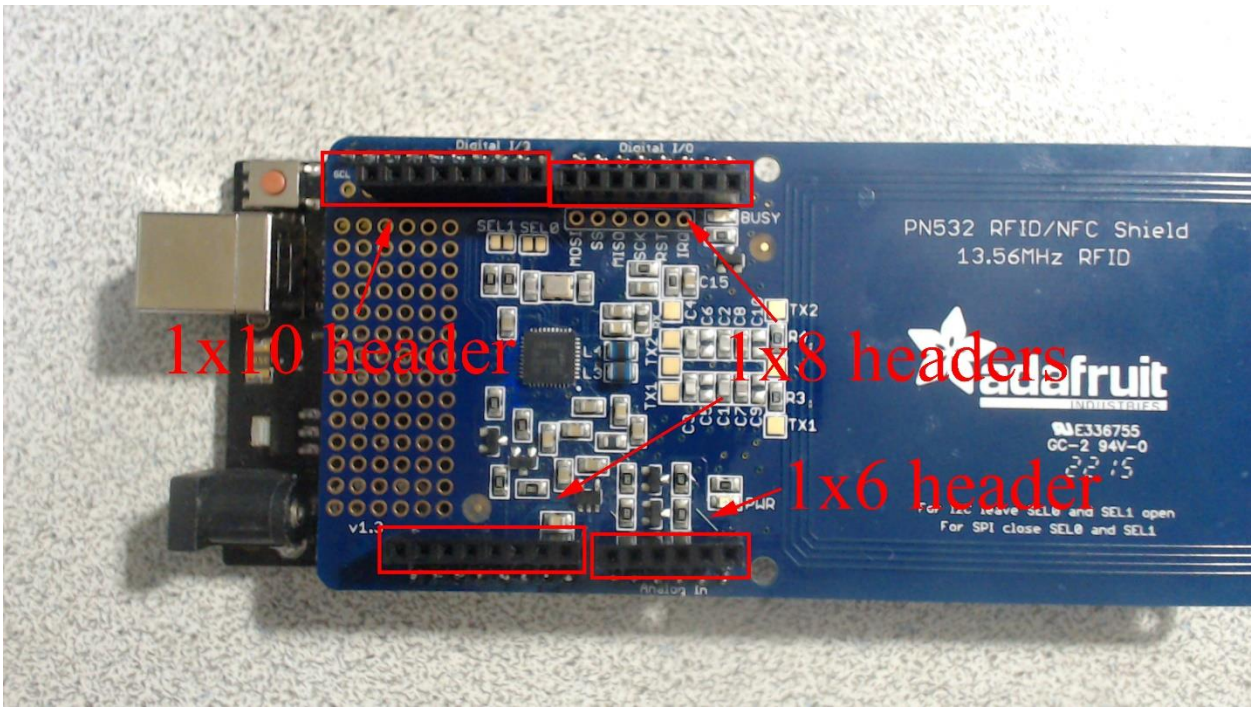


Figure 4-4 – Layout of female-to-male headers on Adafruit PN532 shield

Section 5 – Setting up the Arduino Uno

The Arduino Uno is a consumer-grade and beginner-friendly microcontroller. In this project, the Arduino controls the LCD by displaying status messages to it, as well as the NFC shield, which in turn reads nearby NFC chips.

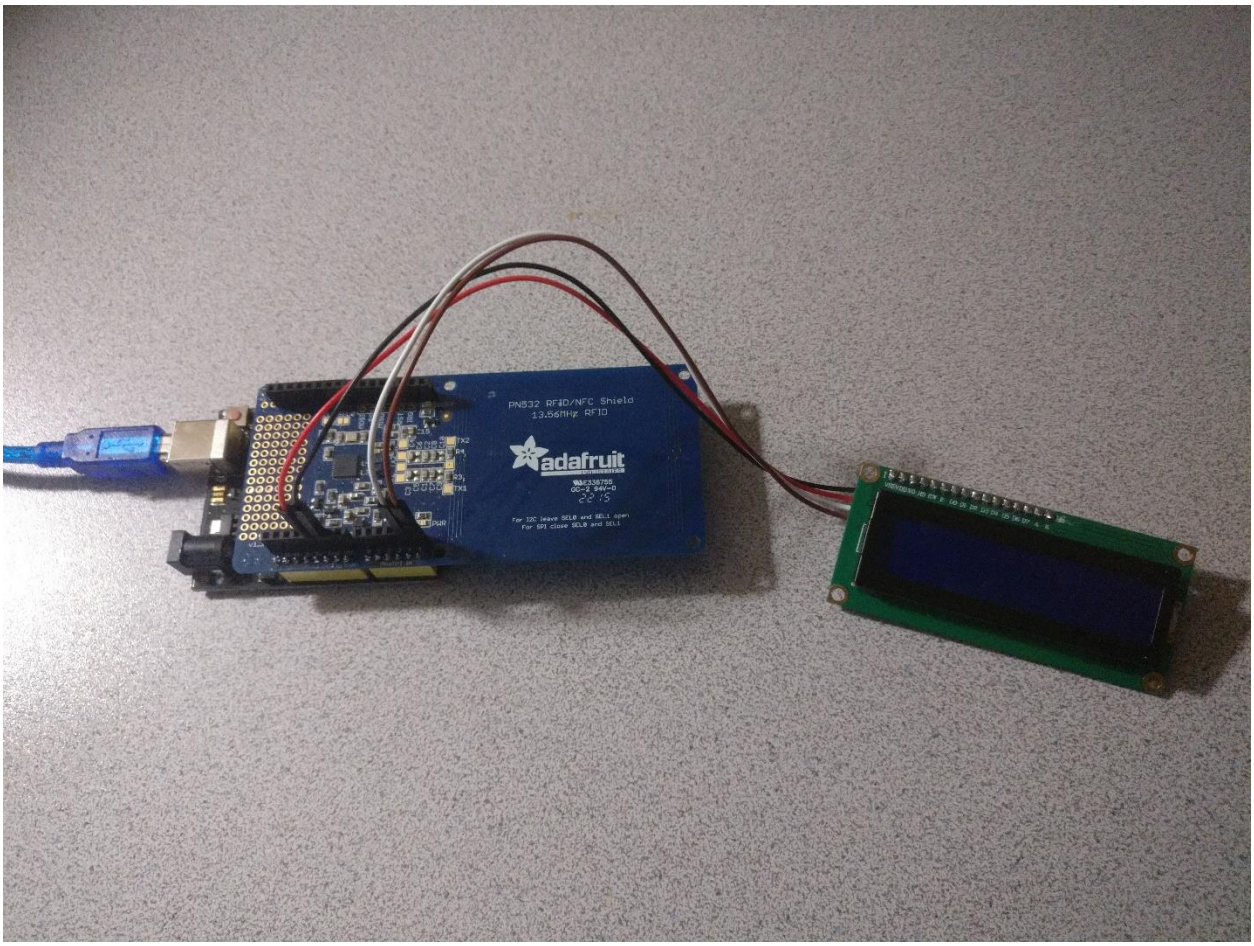


Figure 5-1 – Finished view of Arduino Uno

1. Place the Adafruit PN532 shield on top of the Arduino such that the antenna end of the shield is facing opposite the USB-B port on the Arduino. All the pins on the shield should go in with little resistance and should match the layout on the Arduino. Refer to Figure 5-2 for an example of what the board should look like.

Figure 5-2 – Adafruit PN532 shield attached to Arduino Uno

2. Plug the LCD into the Arduino.

a. Connect the female end of one of the 4-wire male-to-female wires to the "backpack" as shown in Figure 5-3Figure 8-12. Orientation does not matter, but as previously explained, connecting VCC to red and GND to black complies with standard practices.

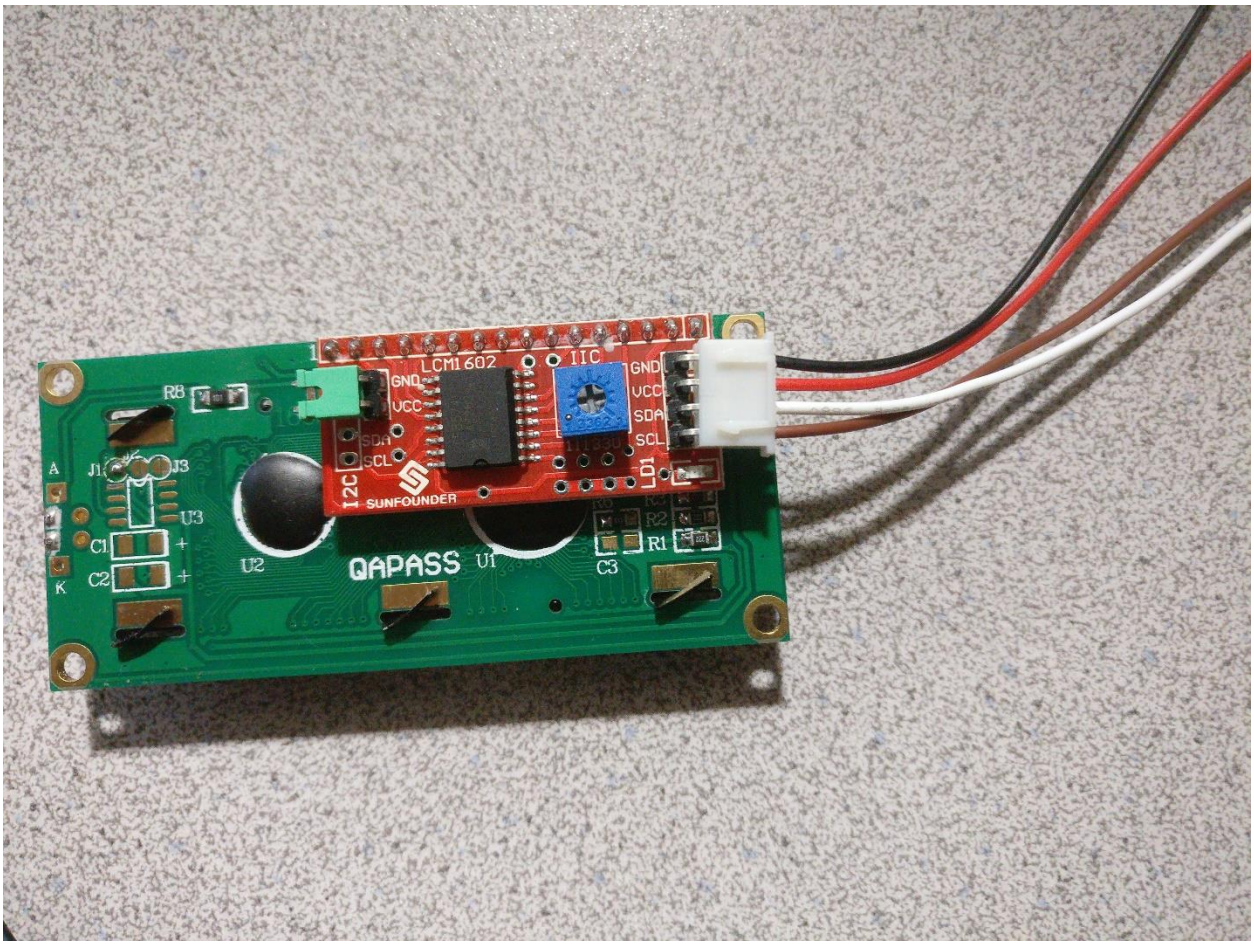


Figure 5-3 - 4-wire male-to-female wire connected to LCD backpack

b. Connect the male end of the 4-wire male-to-female wires as follows. Refer to Figure 5-4 for an example of the completed wiring.

   i. GND to any Gnd pin on the Adafruit PN532 shield

   ii. VCC to the 5v pin on the Adafruit PN532 shield

   iii. SDA to the 4 pin in the “Analog in” section on the Adafruit PN532 shield

   iv. SCL to the 5 pin in the “Analog in” section on the Adafruit PN532 shield

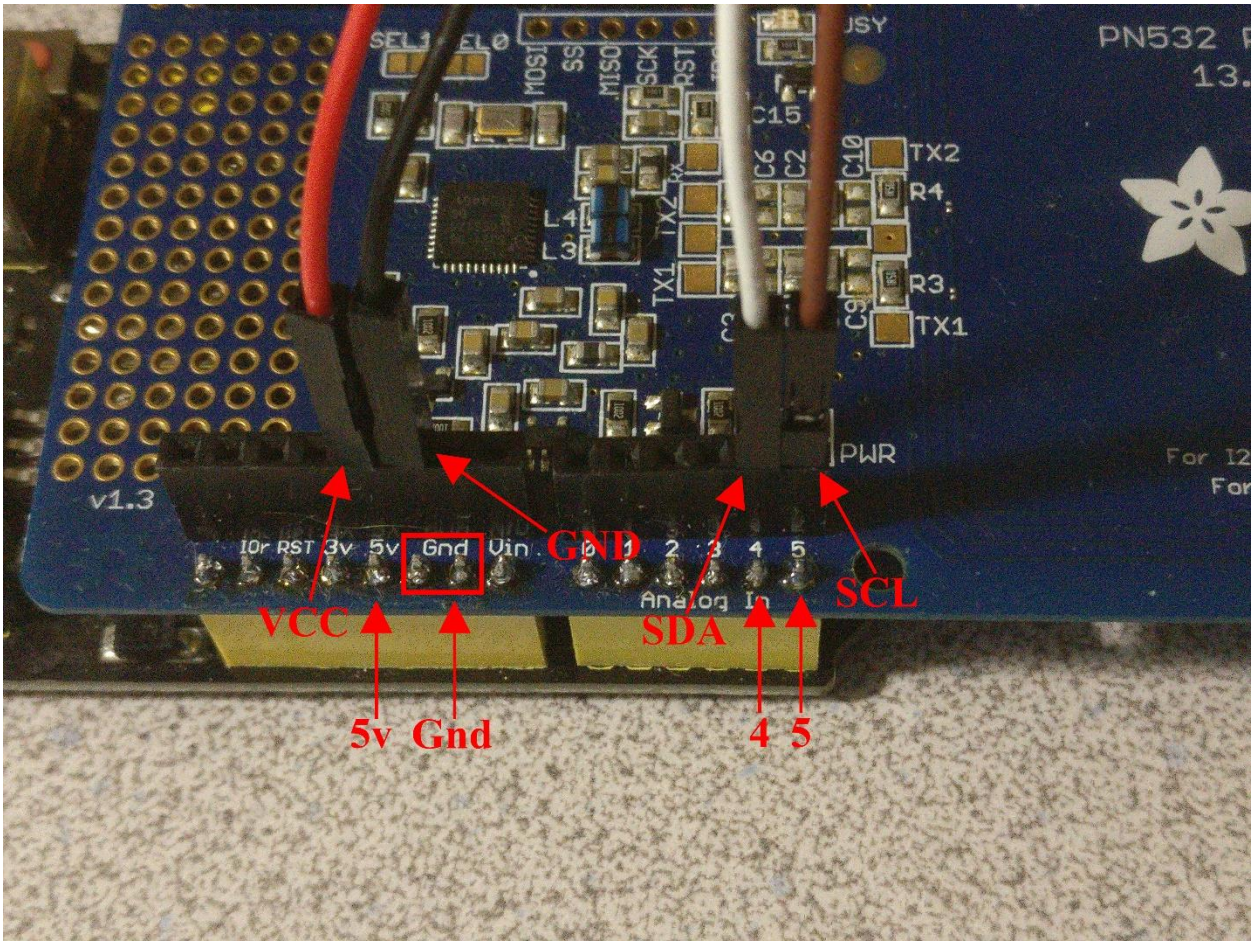


Figure 5-4 – Wiring of LCD backpack into Adafruit PN532 shield

3. Plug the USB-B end of the included cable into the Arduino. Refer to Figure 5-5 for an example of what the cable should look like plugged into the Arduino.

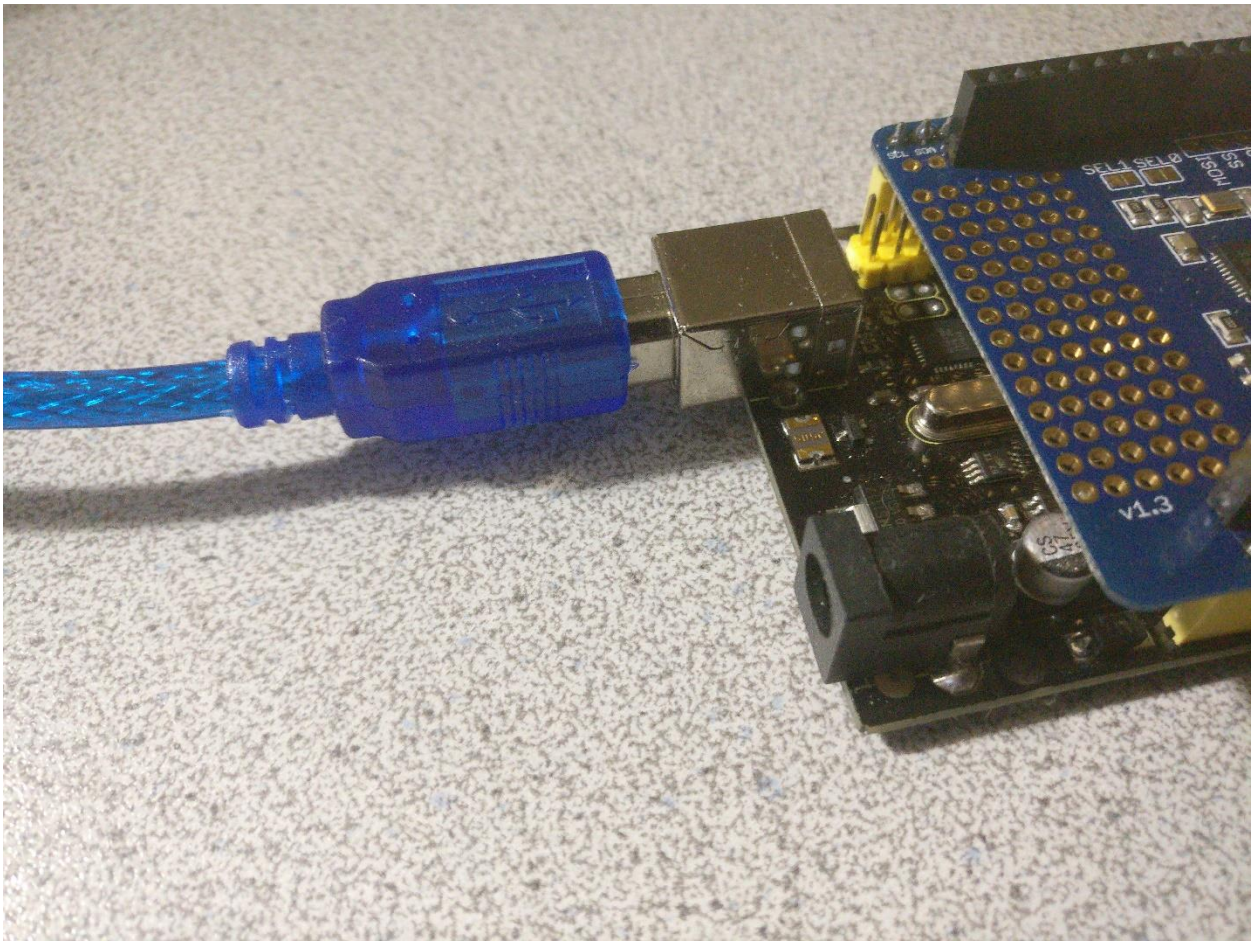


Figure 5-5 – USB-B cable connected to Arduino Uno

Section 6 – Setting up the Arduino IDE

In order to program the Arduino Uno, the Arduino Integrated Development Environment (IDE) is used. The Arduino IDE is a free download from the official Arduino website. One convenient feature about this IDE, as with most, is the ability to add custom libraries. Two custom libraries will used in this project—Adafruit's PN532 library and Francisco Malpartida's New LiquidCrystal library.

1. Download and install the Arduino IDE using default settings. Refer to Figure 6-1 for visual assistance on what file to download.

Figure 6-1 – Arduino IDE download page

2. Download the Adafruit PN532 software library. Refer to Figure 6-2 for visual assistance.

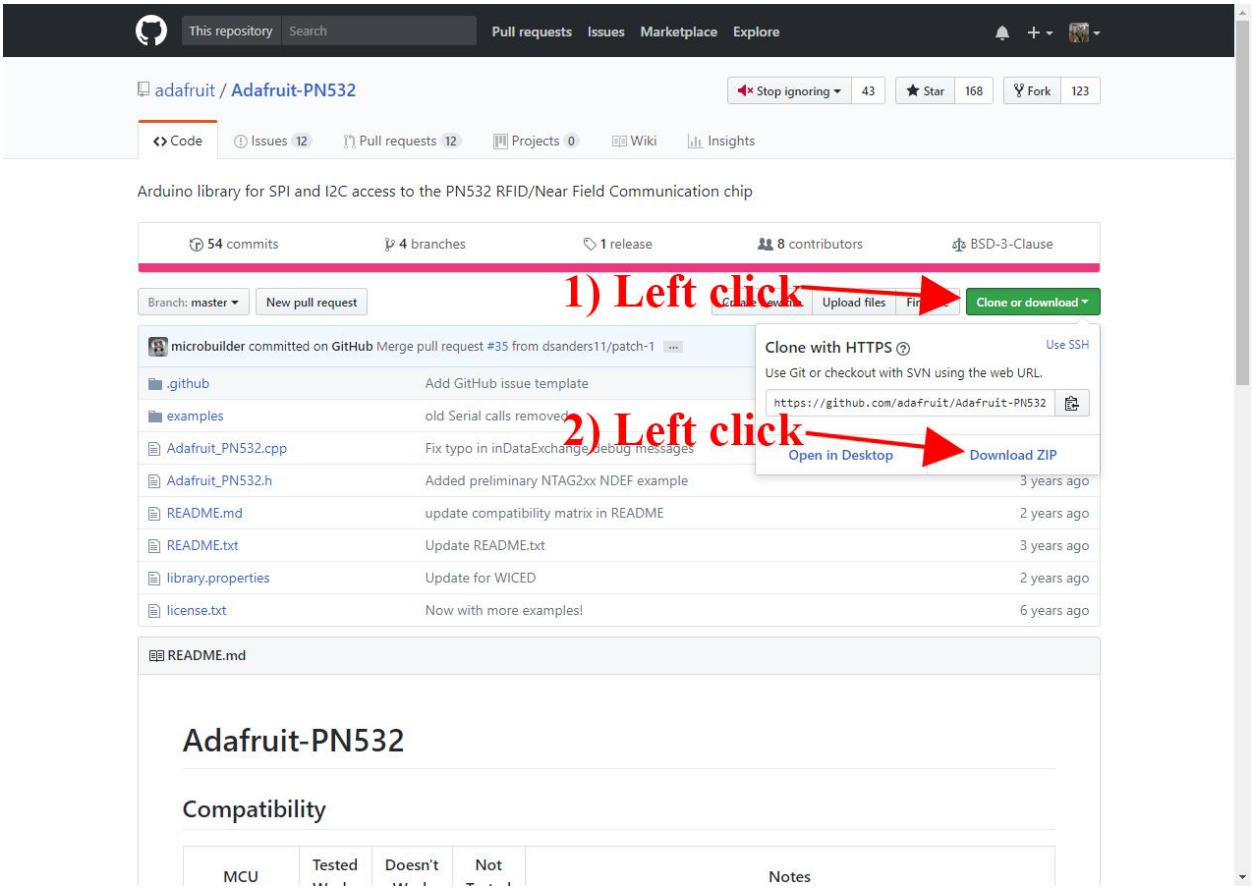


Figure 6-2 – Adafruit PN532 software library download instructions

3. Extract the downloaded ZIP file to the default Arduino user library directory. In this documentation, the Windows archive software will be used.
    a. Left click on the downloaded file to select it.
    b. Right click on the downloaded file.
    c. Left click on the “Extract All…” option.
4. The default extraction directory is the same directory as the ZIP file. To modify it, either type in the desired directory or click “Browse…,” find the desired directory, and click “Select Folder.” For this documentation, the extracted directory will be the user documents folder. Refer to Figures 4-3 through 4-6 for illustrated instructions.

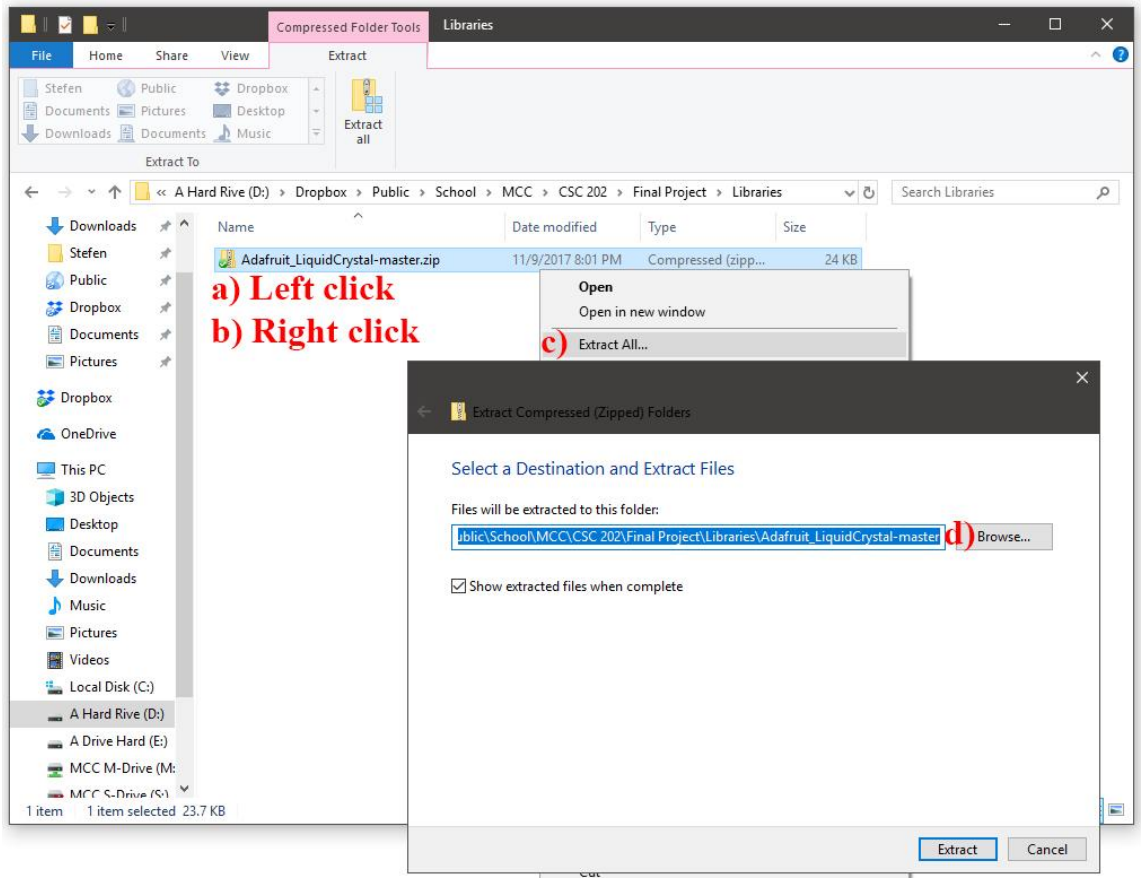


Figure 6-3 – Windows file extraction dialog window

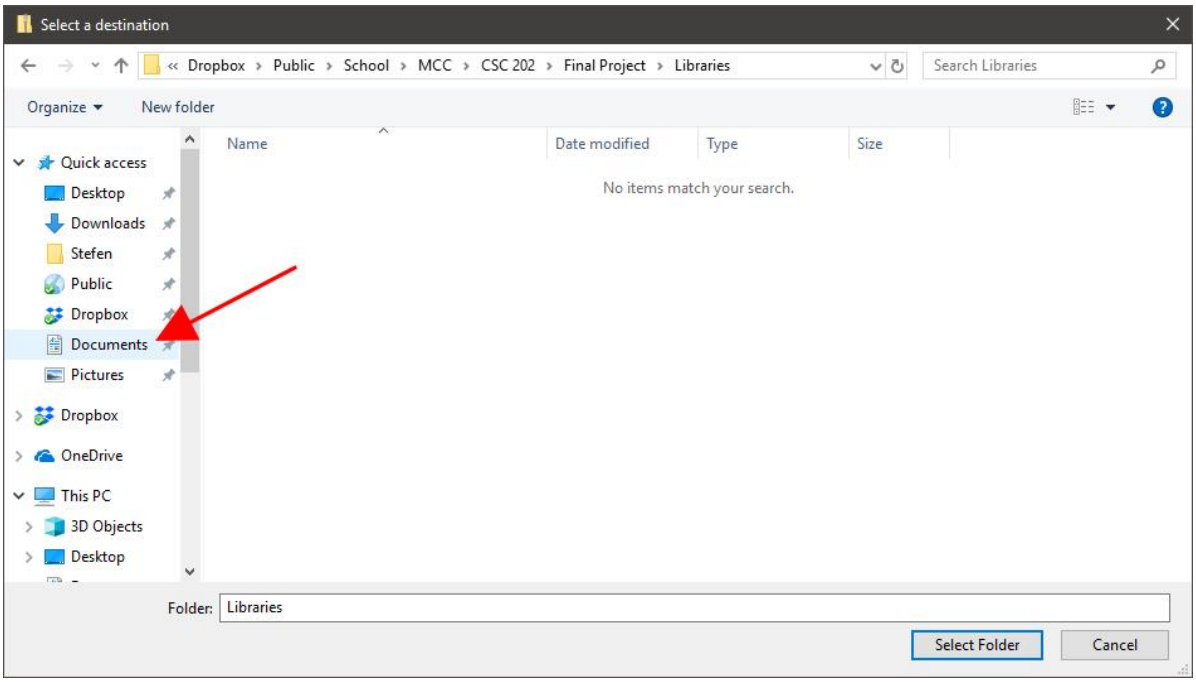


Figure 6-4 – Selecting user documents folder

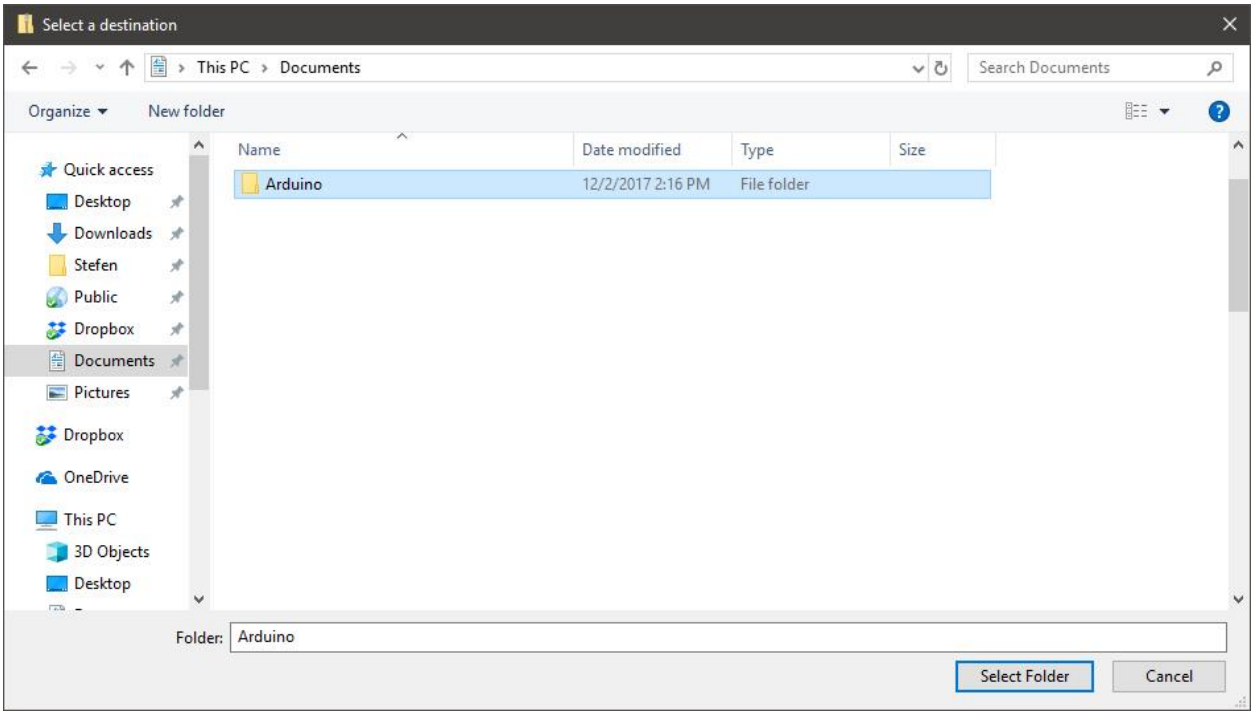


Figure 6-5 – Selecting Arduino documents folder

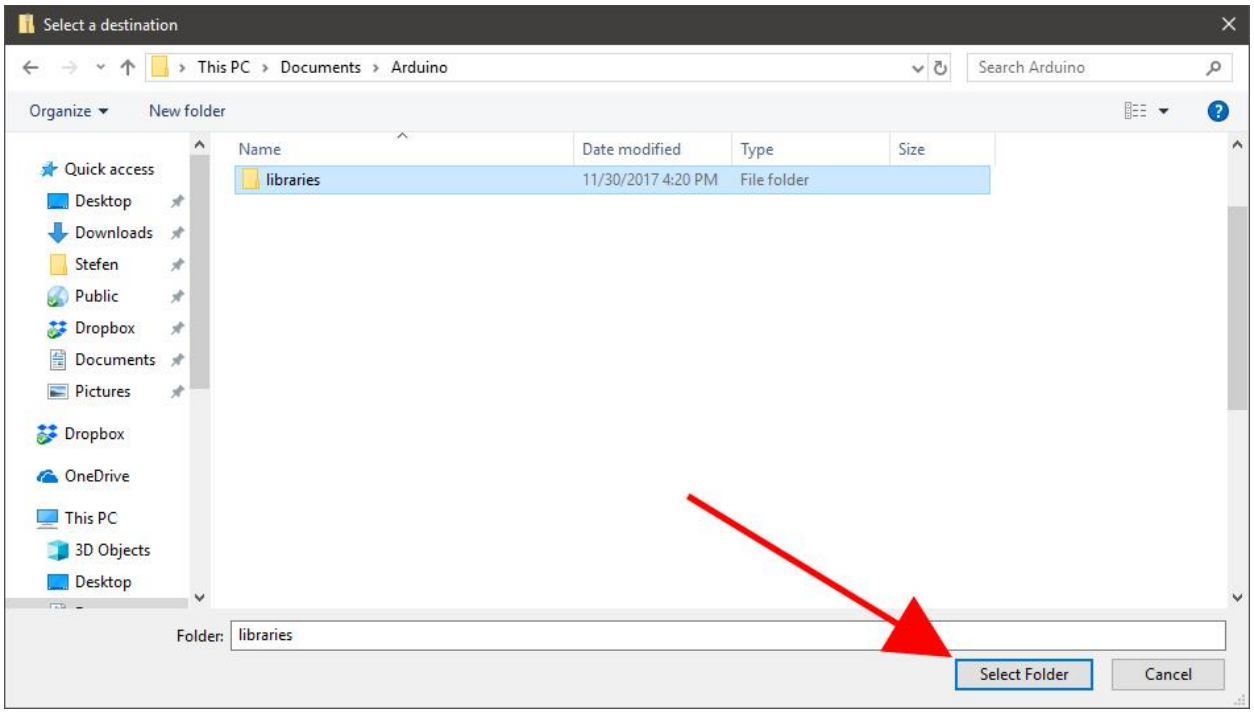


Figure 6-6 – Selecting Arduino user libraries folder

5. Repeat steps 2-4 for the New Liquid Crystal software library. Refer to Figure 4-7 for illustrated instructions.

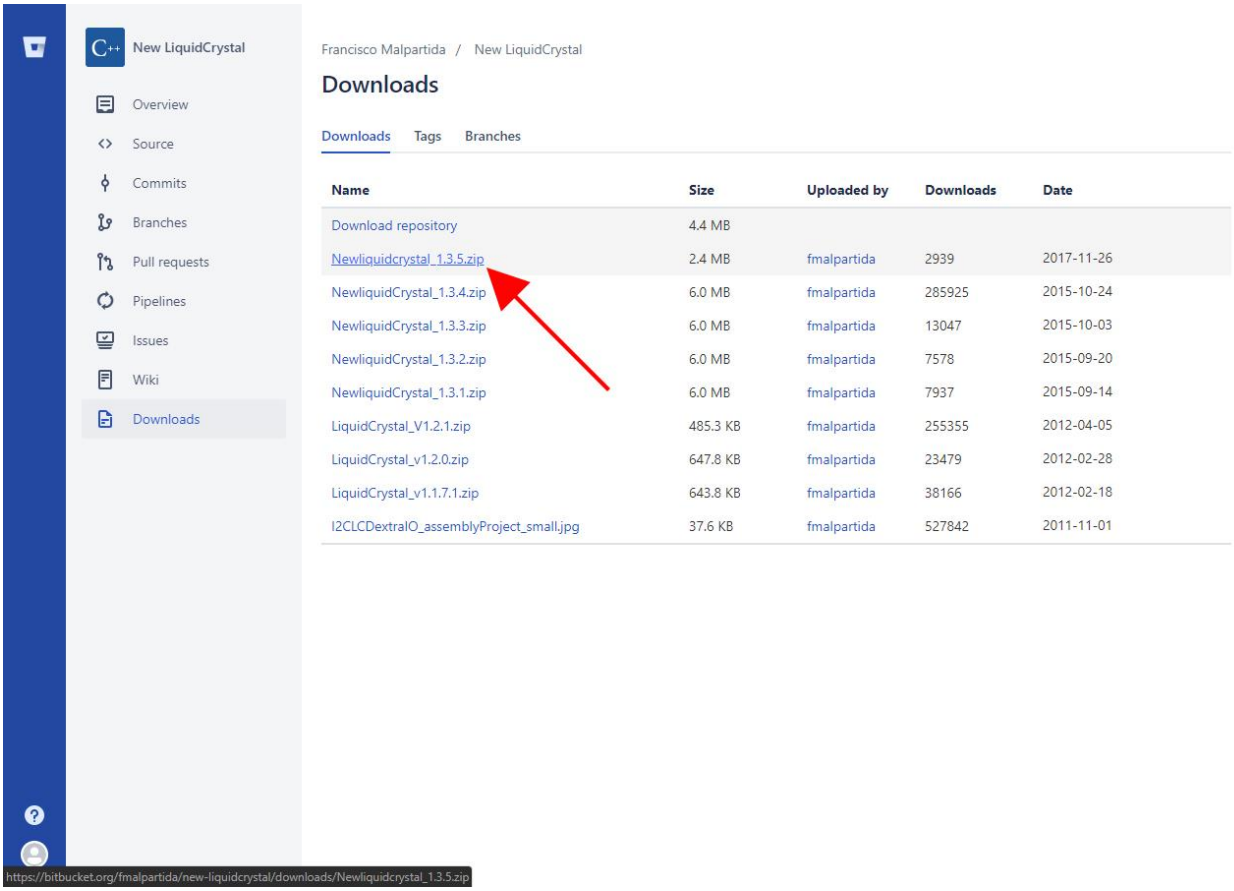


Figure 6-7 – Downloading New Liquid Crystal software library

Section 7 – Programming the Arduino Uno

Once the Arduino Uno and IDE are both set up, the Arduino Uno can be programmed. It is not vital that this be done after setting the Arduino Uno up, but highly recommended, as some code may malfunction if a sensor or other device is not properly connected to the board.

1. Plug the USB-A (“regular” USB) end of the USB-B cable that’s connected to the Arduino into the computer.
2. Download the Arduino project files.
3. Extract the project files to any permanent storage location, such as a hard drive or USB drive. The location of the extracted files does not matter, so long as they are accessible by the Arduino IDE.
4. Enter the “DoorLock_Arduino” directory.
5. Double click on the “DoorLock_Arduino.ino” file.
6. When the Arduino IDE loads, click on “Tools” in the toolbar on the top of the window.
7. Expand the entry near the bottom of the dropdown menu beginning with “Boards” by hovering over the entry.
8. Click on the “Arduino/Genuino Uno” entry within the boards submenu.
9. Click on “Tools” again in the toolbar.
10. Expand the entry beginning with “Port.”
11. Click on the entry that ends with “(Arduino/Genuino Uno).”
12. Click on the “Upload” button near the top of the Arduino IDE window.
13. When the upload completes, unplug the Arduino Uno from the computer.

Section 8 – Setting up the Thunderbird12

The Thunderbird12 is a consumer-grade and beginner-friendly microcontroller, often used in introductory embedded programming courses. In this project, the Thunderbird controls various hardware—a piezoelectric speaker, an RGB (Red, Green, Blue) LED, and a relay. The piezoelectric speaker is used when a key is successfully granted or denied access. The RGB LED also has this function, but also serves as a secondary status indicator as to what state the system is currently in. The relay is used to switch a much higher voltage to allow the solenoid to turn on or off. The BDM serves as a middleman device when connecting to computers, sending and receiving serial data. The BDM is also the Thunderbird12's power source and ground.

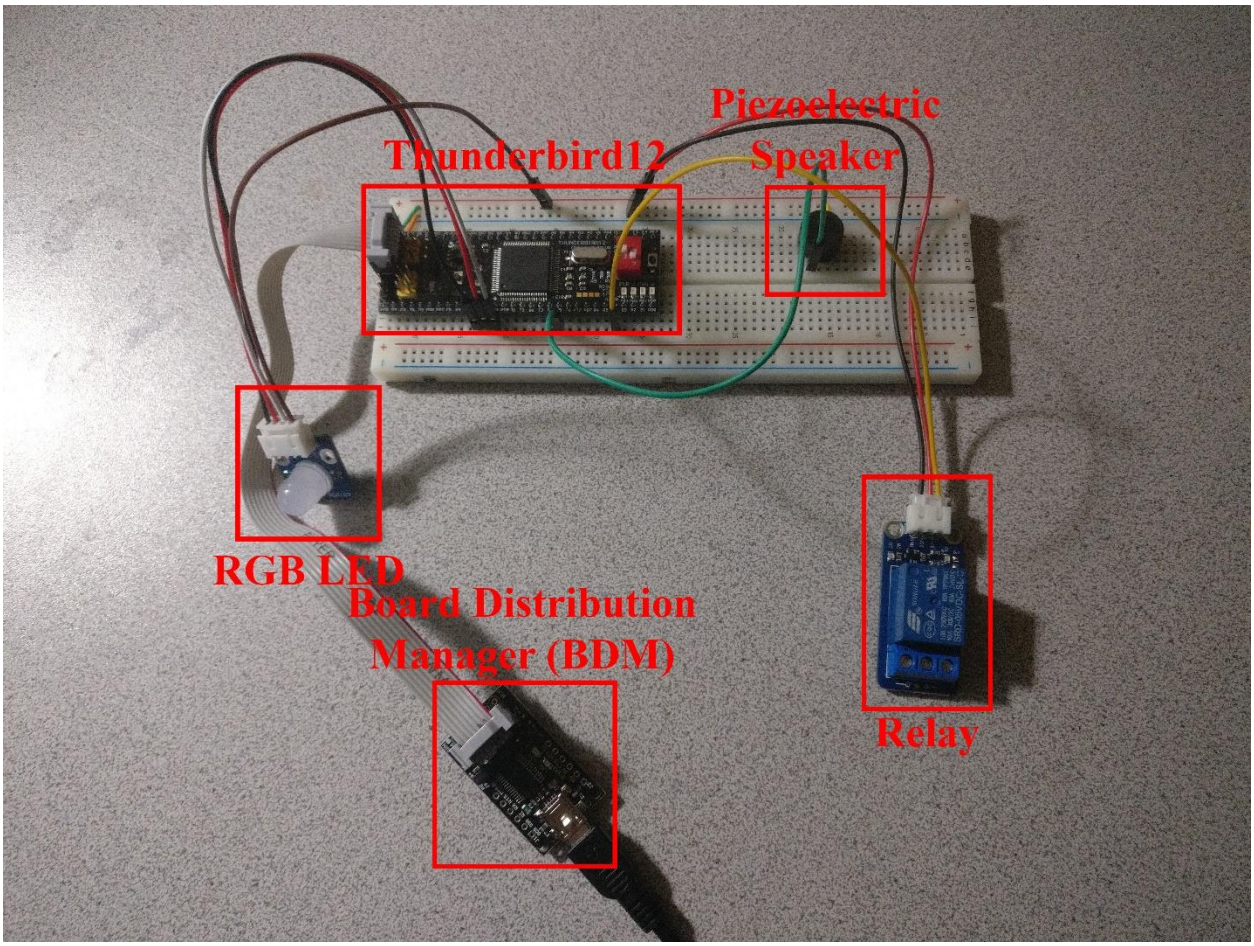


Figure 8-1 – Finished view of the Thunderbird12

1. Insert the Thunderbird into the solderless breadboard. There will be letters and numbers printed on the breadboard on different axes. The top row of pins on the Thunderbird should be inserted into the column b, and the bottom row of pins should be inserted into column h. The board should be inserted as far to the left of the breadboard as possible, left of row 60. Refer to Figure 8-2 for an example of what the breadboard should look like.

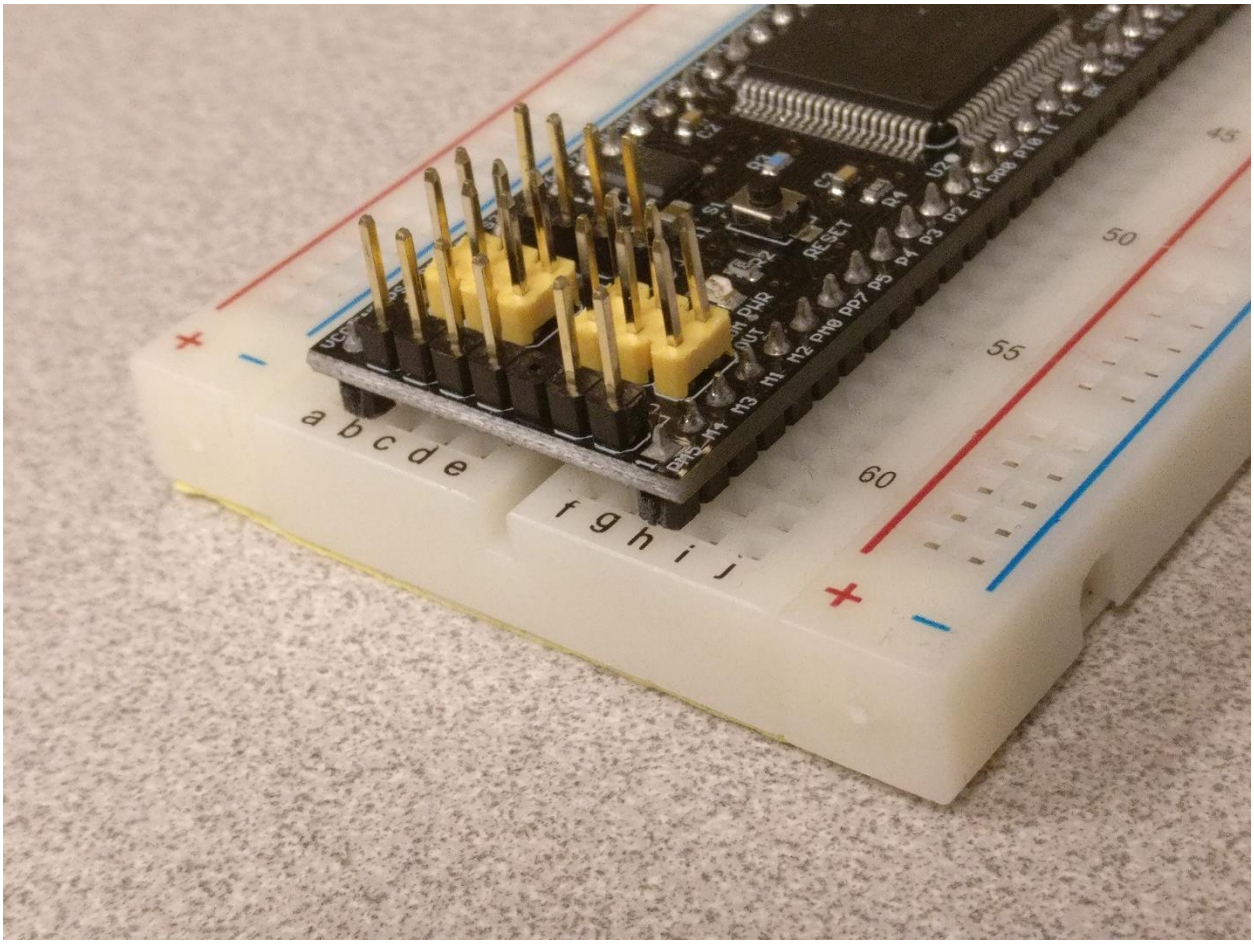


Figure 8-2 – Thunderbird12 inserted in solderless breadboard

2. There will be a red module on the Thunderbird with two white switches. Oriented correctly, the switches will be on the right side of the board with the label “S2” just above it. Ensure that both switches are in the down position. Refer to Figure 8-3 for an example of what the switch configuration should look like.

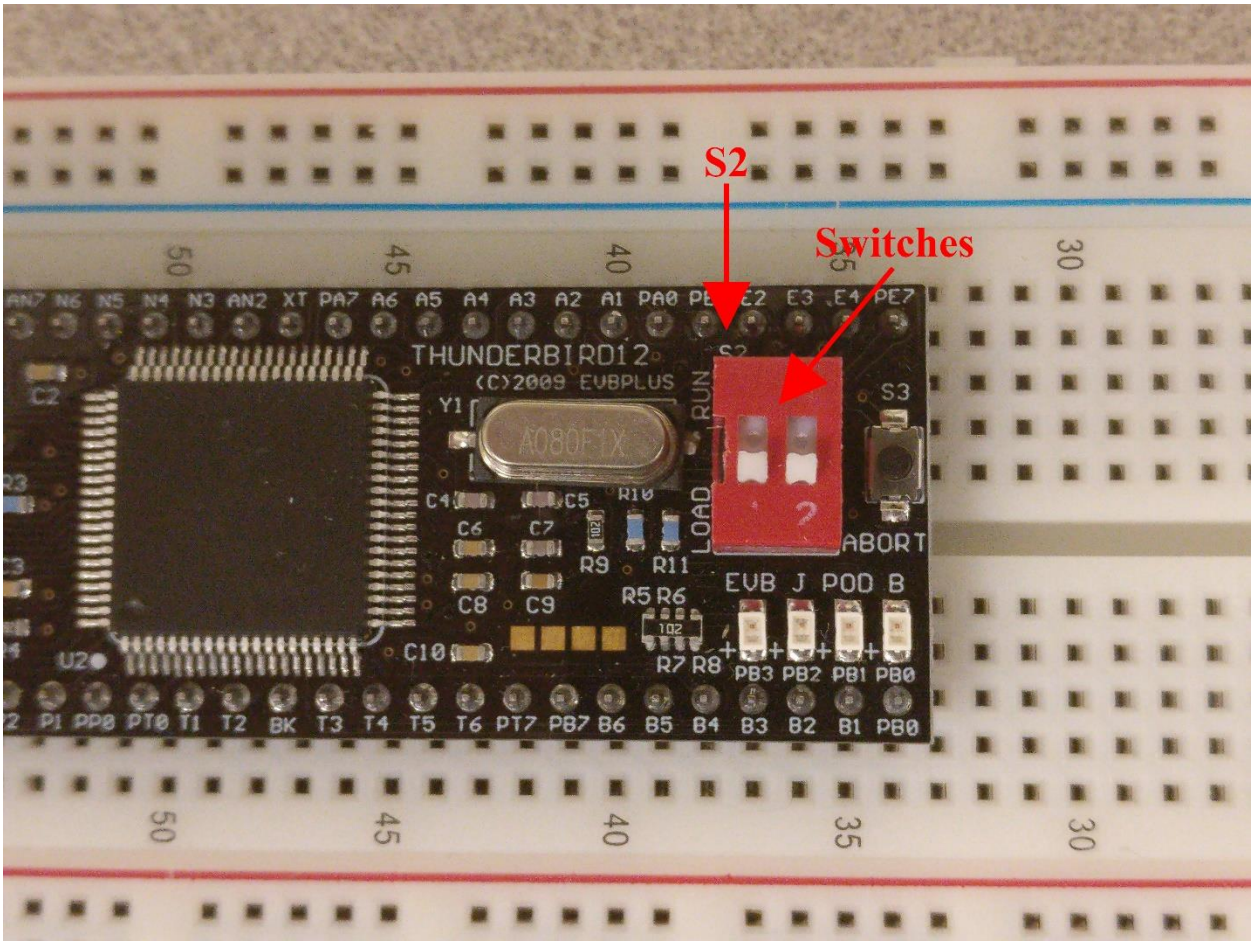


Figure 8-3 – Board configuration showing both on-board switches in down position

3. Connect the Thunderbird12 to the Board Distribution Manager (BDM).
    a. The smaller module that came with the Thunderbird is the BDM. There will be a 4x2 configuration of pins with a yellow plastic base, labled J5 (Jumper 5). Plug the ribbon cable that came with the Thundebird into J5 such that the red line is closer to the J5 silkscreen text. Refer to Figure 8-4 for an example of what the cable should look like plugged into the BDM.

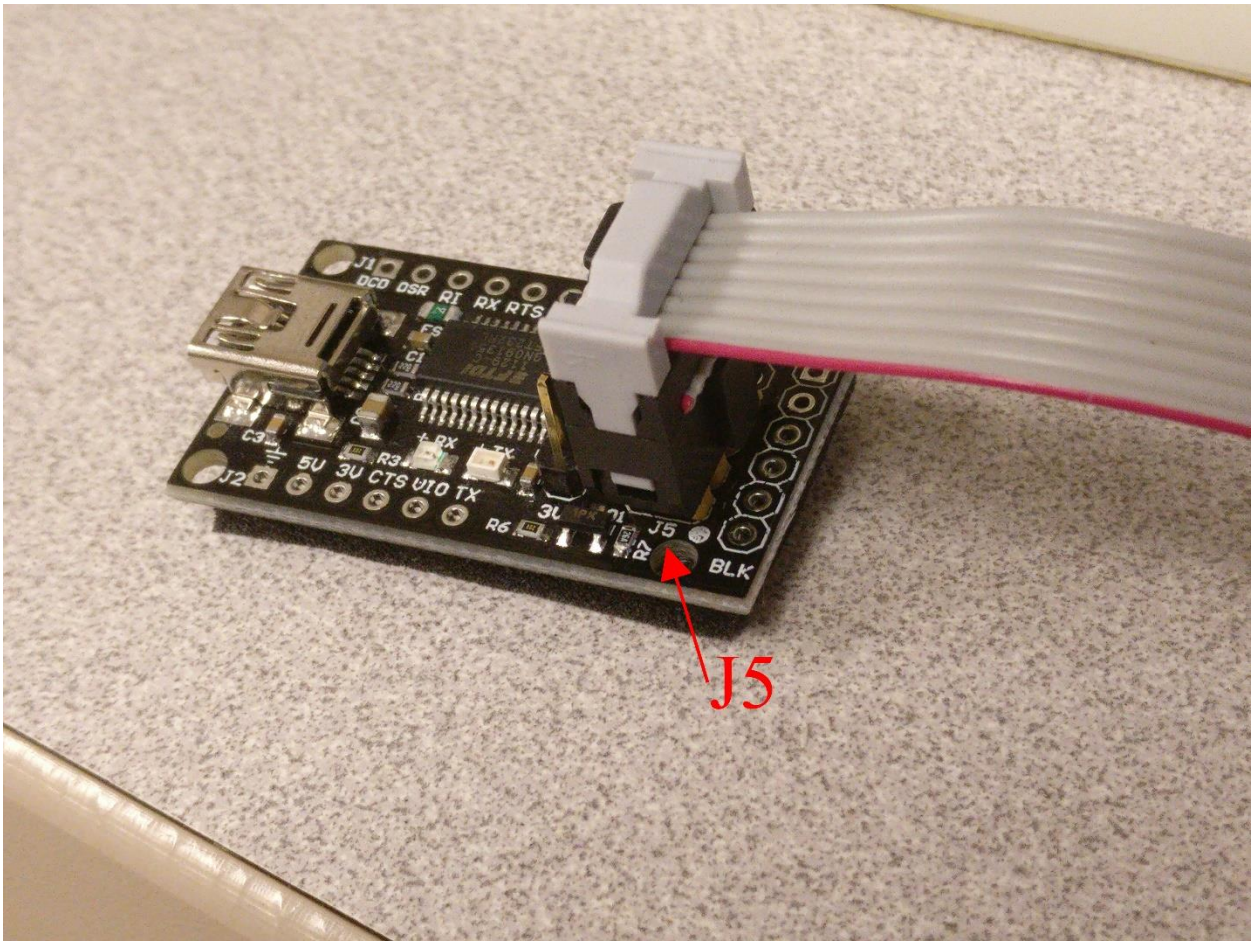


Figure 8-4 – Ribbon cable connected to Board Distrubution Manager (BDM)

b. The other end of the ribbon cable should be plugged into J1 on the Thunderbird such that the red line should be closer to the J1 and VCC silkscreen text, and the unused 4 ports should be closer to J2. Refer to Figure 8-5 for an example of what the cable should look like plugged into the Thunderbird.

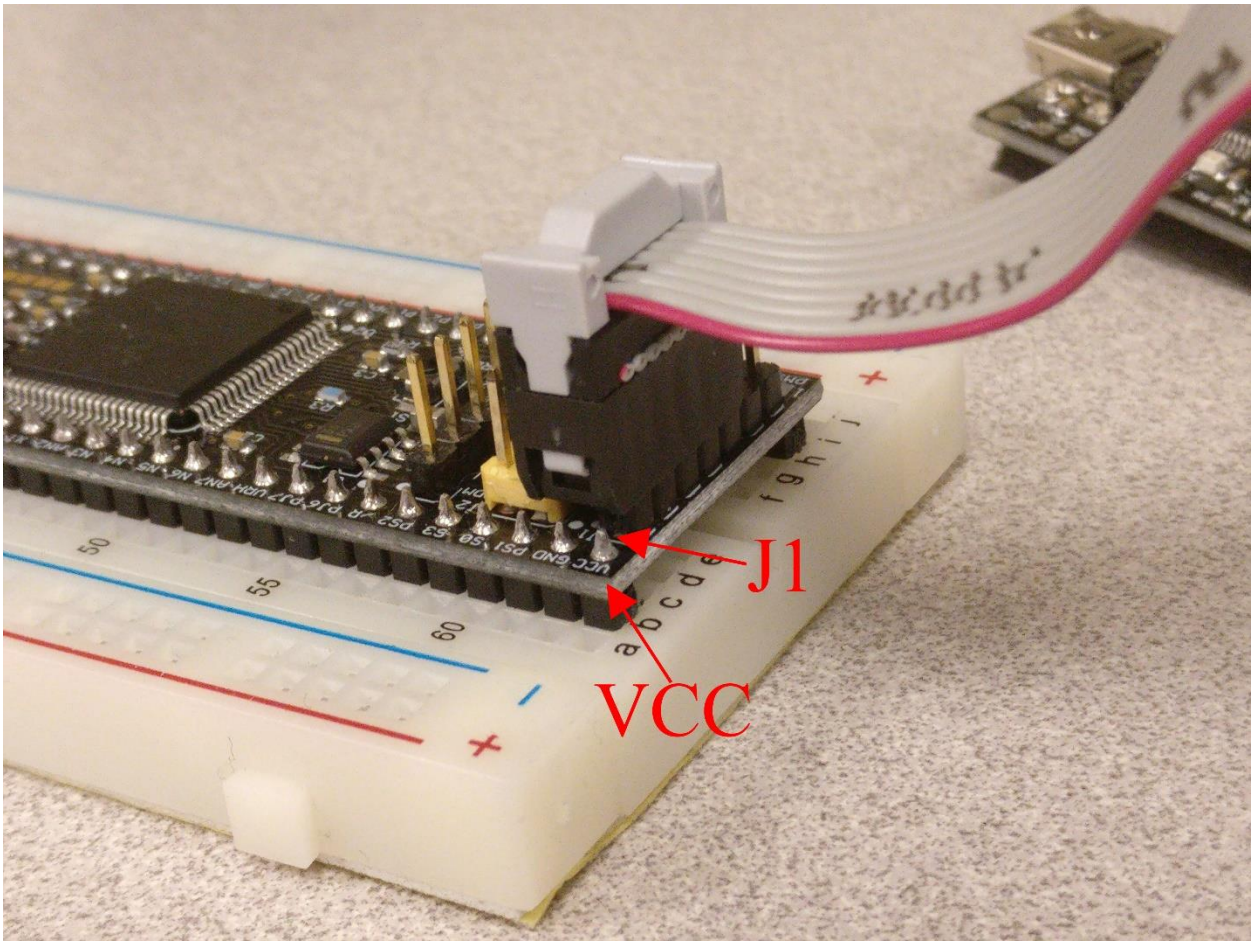


Figure 8-5 – Ribbon cable connected to Thunderbird12

4. Begin wiring the solderless breadboard.
    a. Obtain one 0.75cm (orange) wire, one 1cm (yellow) wire, and one (green) 1.25cm wire from the wire jumper kit, as well as one flexible wire as shown in Figure 8-6.

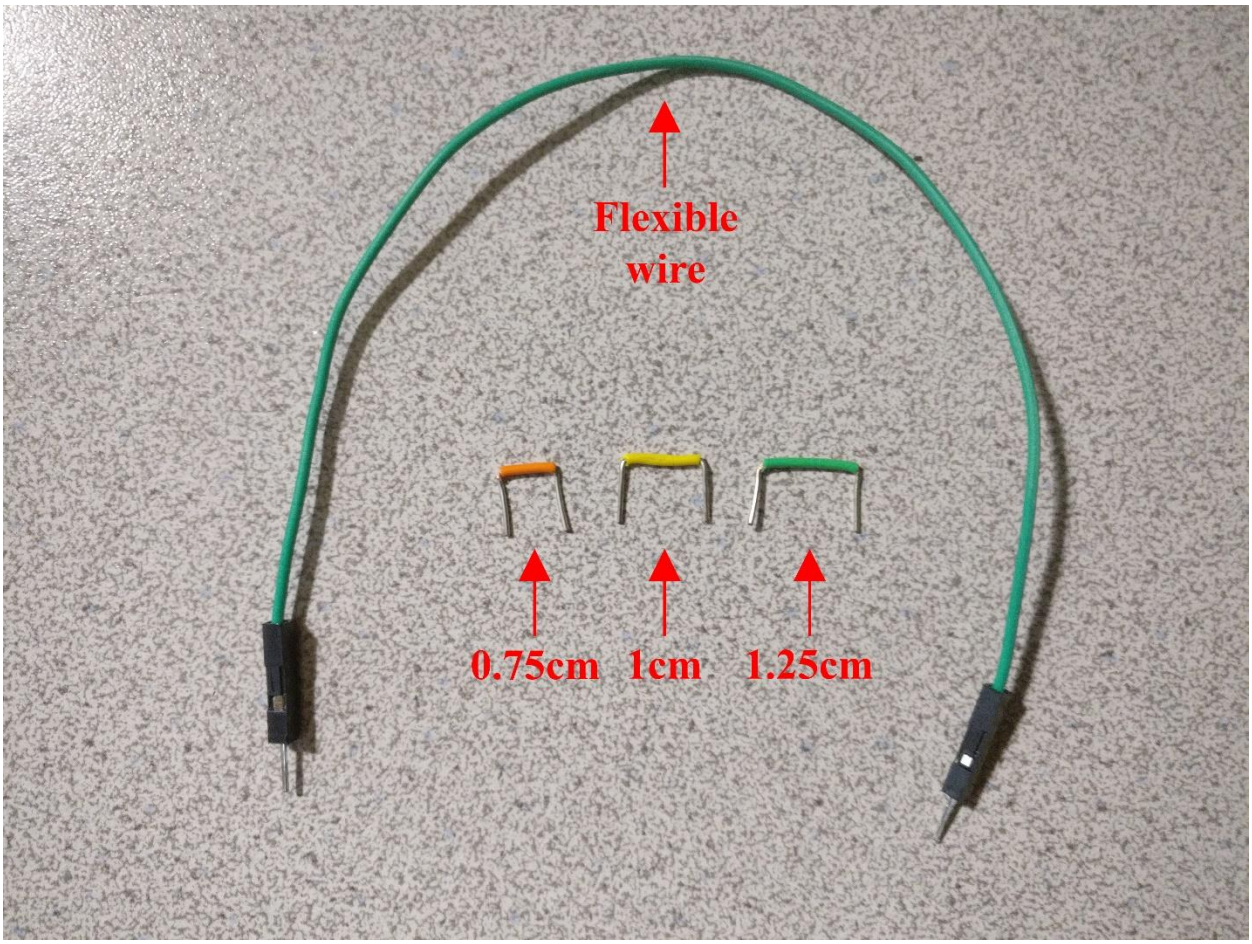


Figure 8-6 – Wires obtained from kit

b. The power rails are the four columns with straight red or blue lines next to them, along with plus or minus signs, respectively. Take one green wire and plug it into the breadboard such that one end is in the same row as VCC on the Thunderbird—also known as plugging a wire “into” VCC—and the other end in the first hole in the “+,” or positive, rail with the red line next to it. Refer to Figure 8-7 for an example of what the wiring should look like.

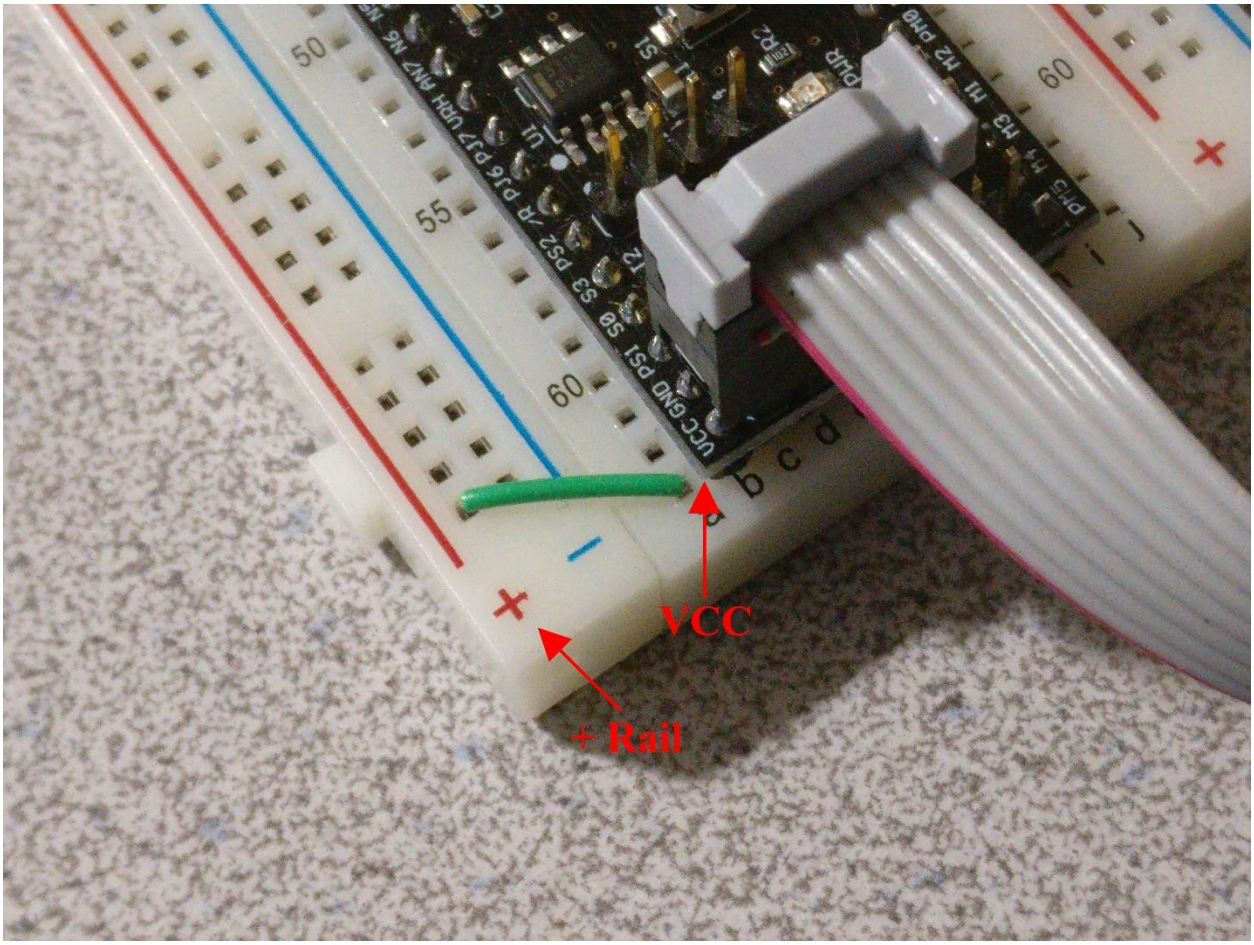


Figure 8-7 – Wire connecting VCC to the positive rail on breadboard

c. Plug the orange wire in such that one end is plugged into GND, which is to the right of VCC, and the other end in the first hole in the "-," or negative, rail with the blue line next to it. Refer to Figure 8-8 for an example of what the wiring should look like.

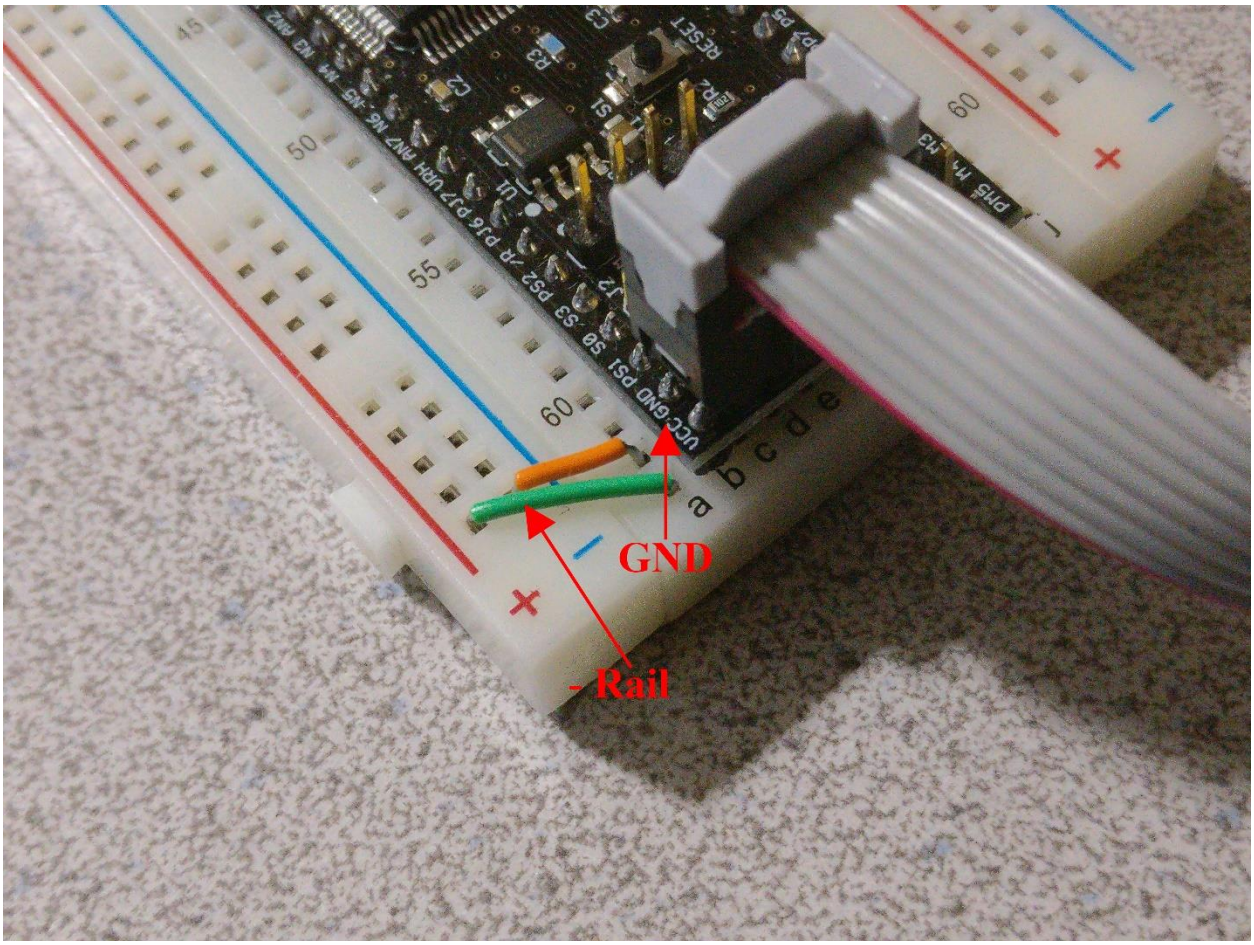


Figure 8-8 – Wire connecting GND to the negative rail on breadboard

5. Connect the piezoelectric speaker to the Thunderbird.
    a. Plug the yellow wire in such that one end is in the “a” column, but to the right of the Thunderbird so that it does not interfere with the Thunderbird, and the other end in the positive rail. Refer to Figure 8-9 for an example of what the wiring should look like.

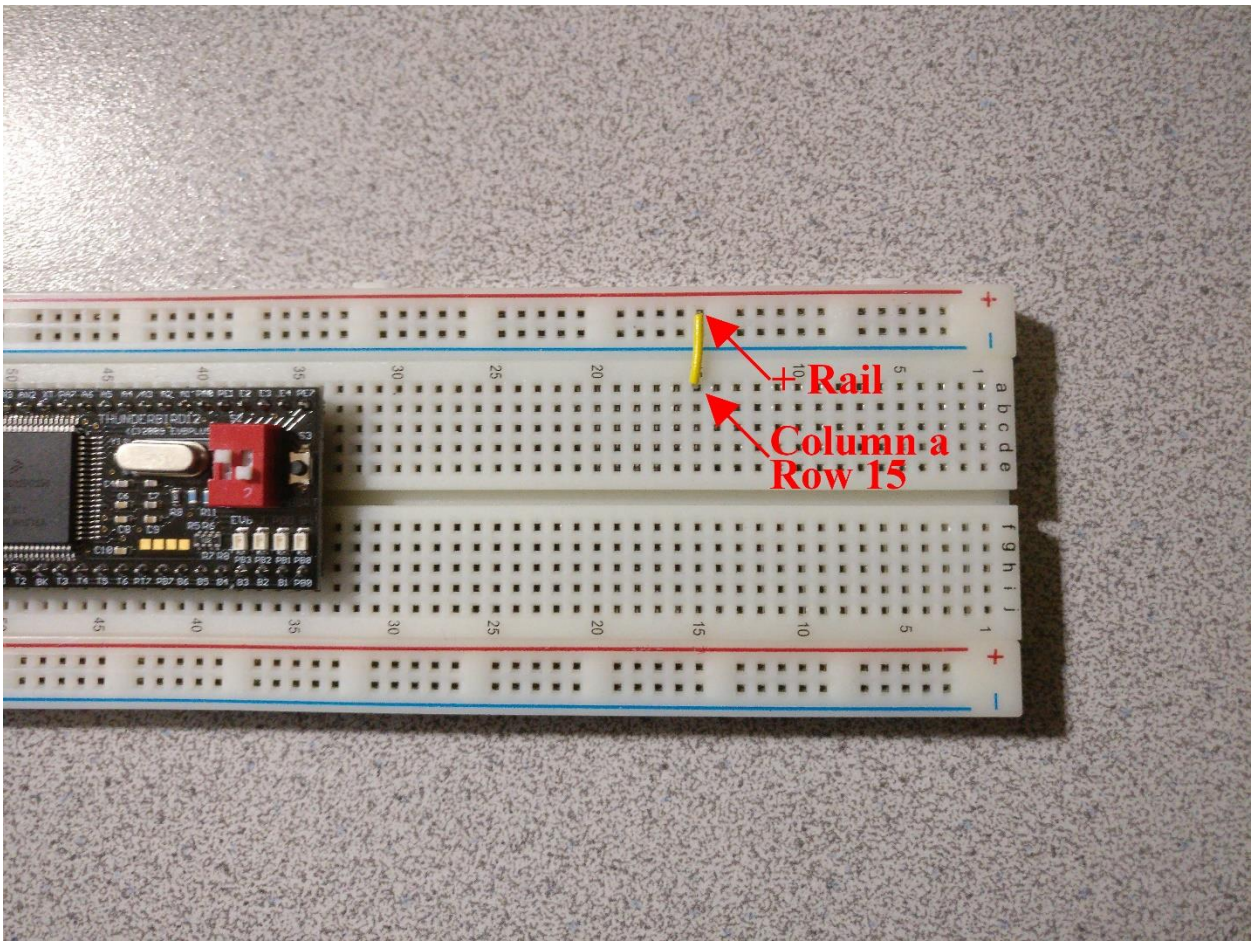


Figure 8-9 – Wire connecting the positive rail to an unused row on breadboard

b. Plug in the piezoelectric speaker such that one pin plugs into the same row as the yellow wire in the previous step, and the other pin plugs into a different row. Refer to Figure 8-10 for an example of what the placement should look like.
Note: The pin must be in columns “b” through “e,” otherwise no current will flow to it, as the center depression in the breadboard seperates two sections of rows.

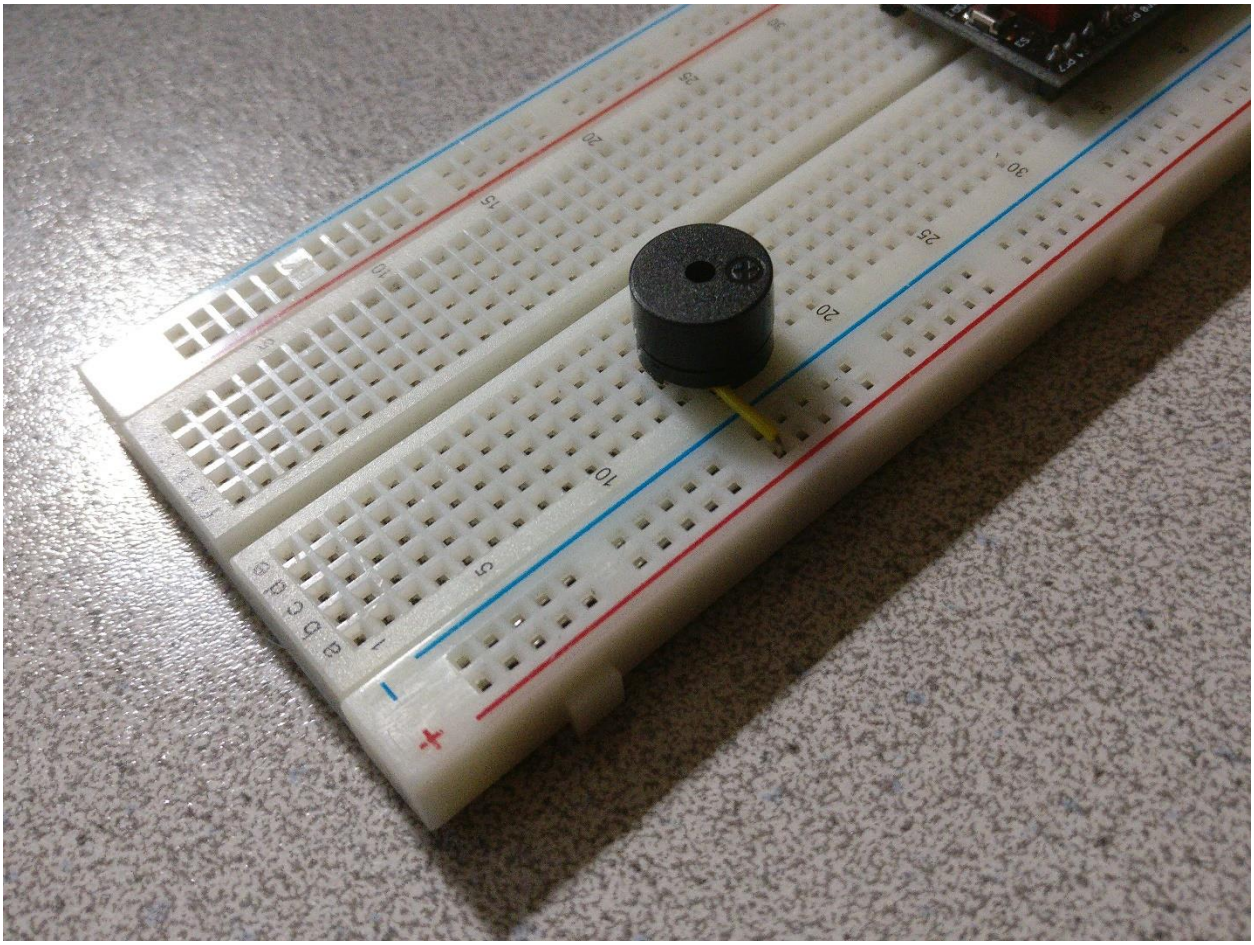


Figure 8-10 – Piezoelectric speaker plugged into breadboard

c. Plug the flexible wire in such that one end is in the same row as the unused pin for the piezoelectric speaker and the other end into T5 on the bottom row of the Thunderbird. Refer to Figure 8-11 for an example of what the wiring should look like.

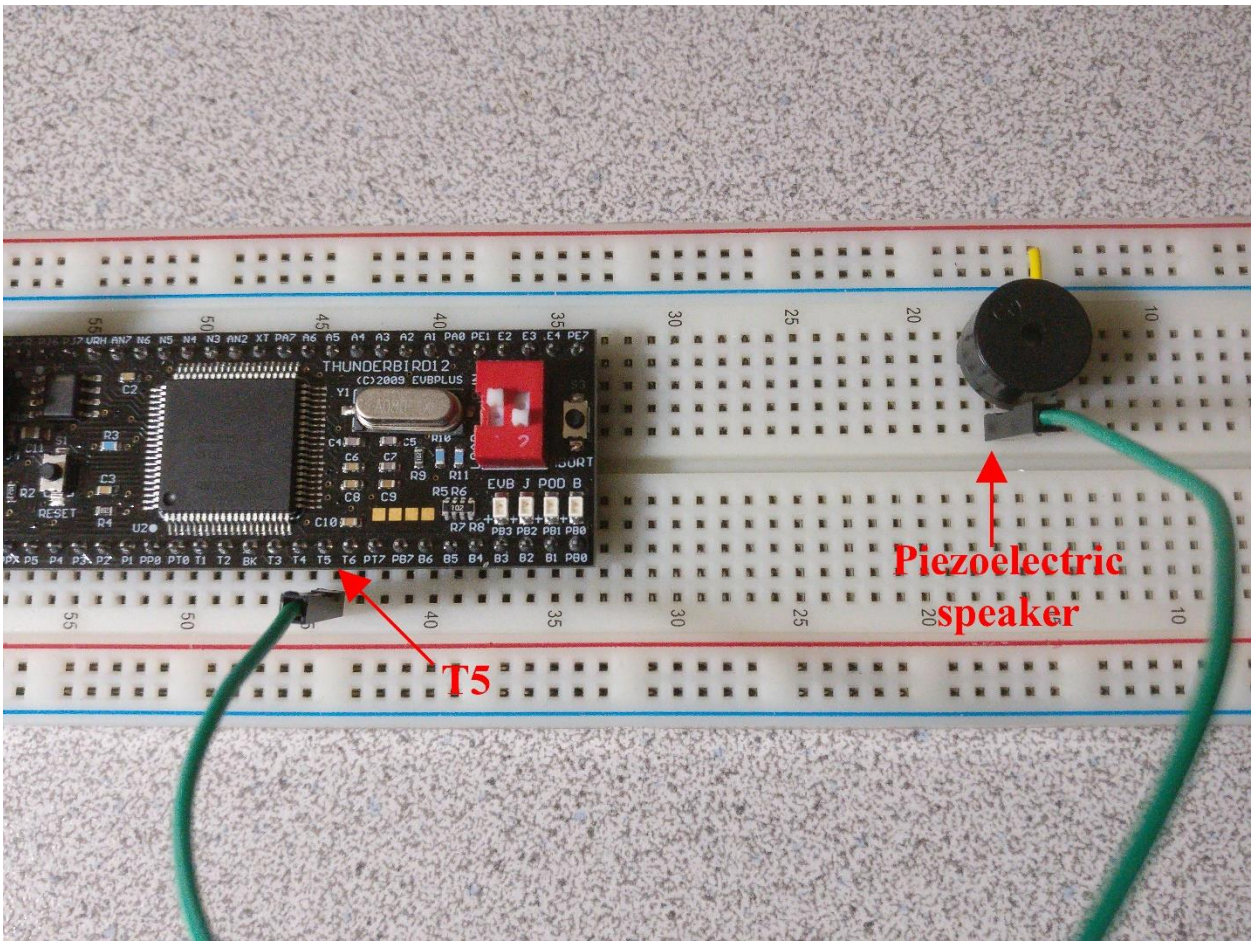


Figure 8-11 – Piezoelectric speaker connected to PT5 on Thunderbird12

6. Connect the RGB LED to the Thunderbird.
    a. Connect the female end of one of the 4-wire male-to-female wires to the RGB LED module as shown in Figure 8-12.

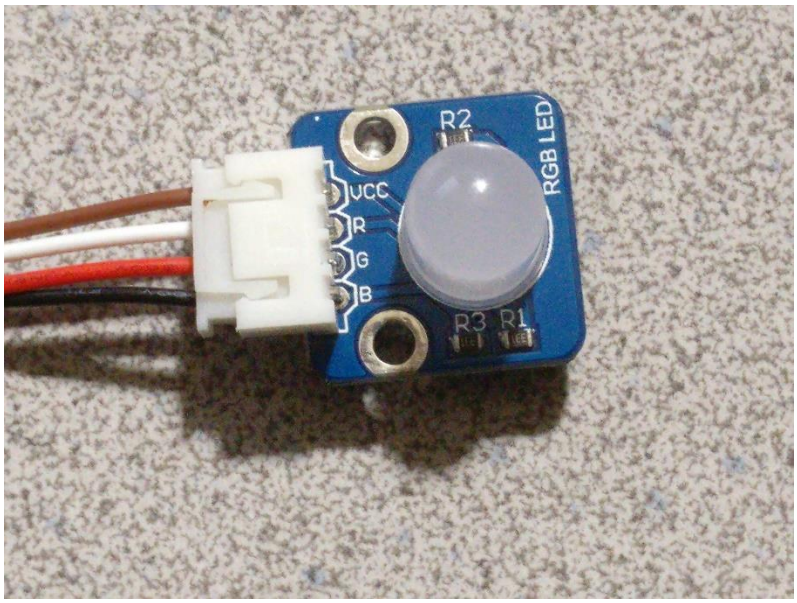


Figure 8-12 – 4-wire male-to-female wire connected to RGB LED module

b. Connect the R, B, and B wires from the RGB LED module to PP0, P1, and P2, respectively, on the bottom row of the Thunderbird. Refer to Figure 8-13 for an example of what the wiring should look like.

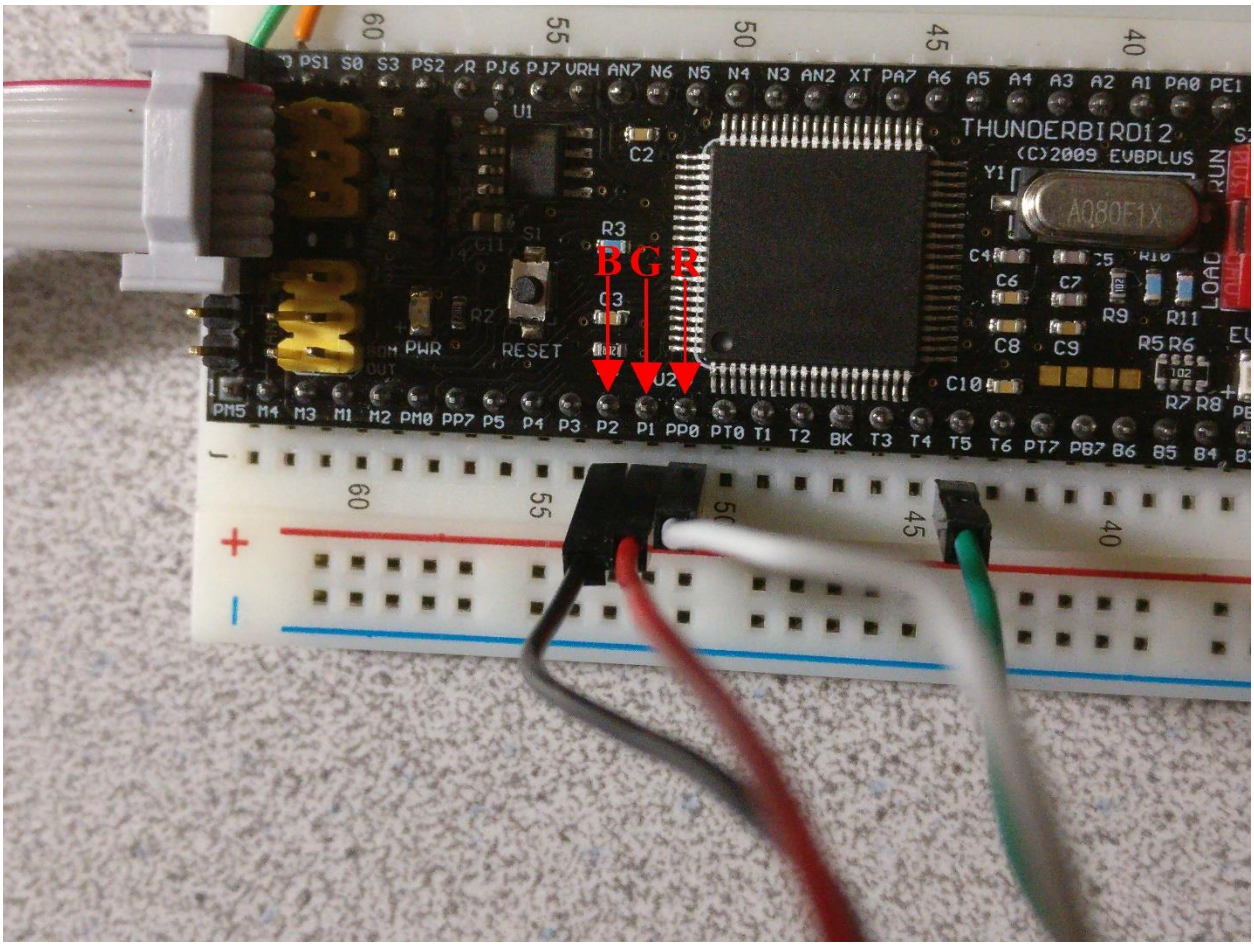


Figure 8-13 – RGB LED module connected to PP0, P1, and P2

c. Connect the VCC wire from the RGB LED module to the positive rail. Refer to Figure 8-14 for an example of what the wiring should look like.

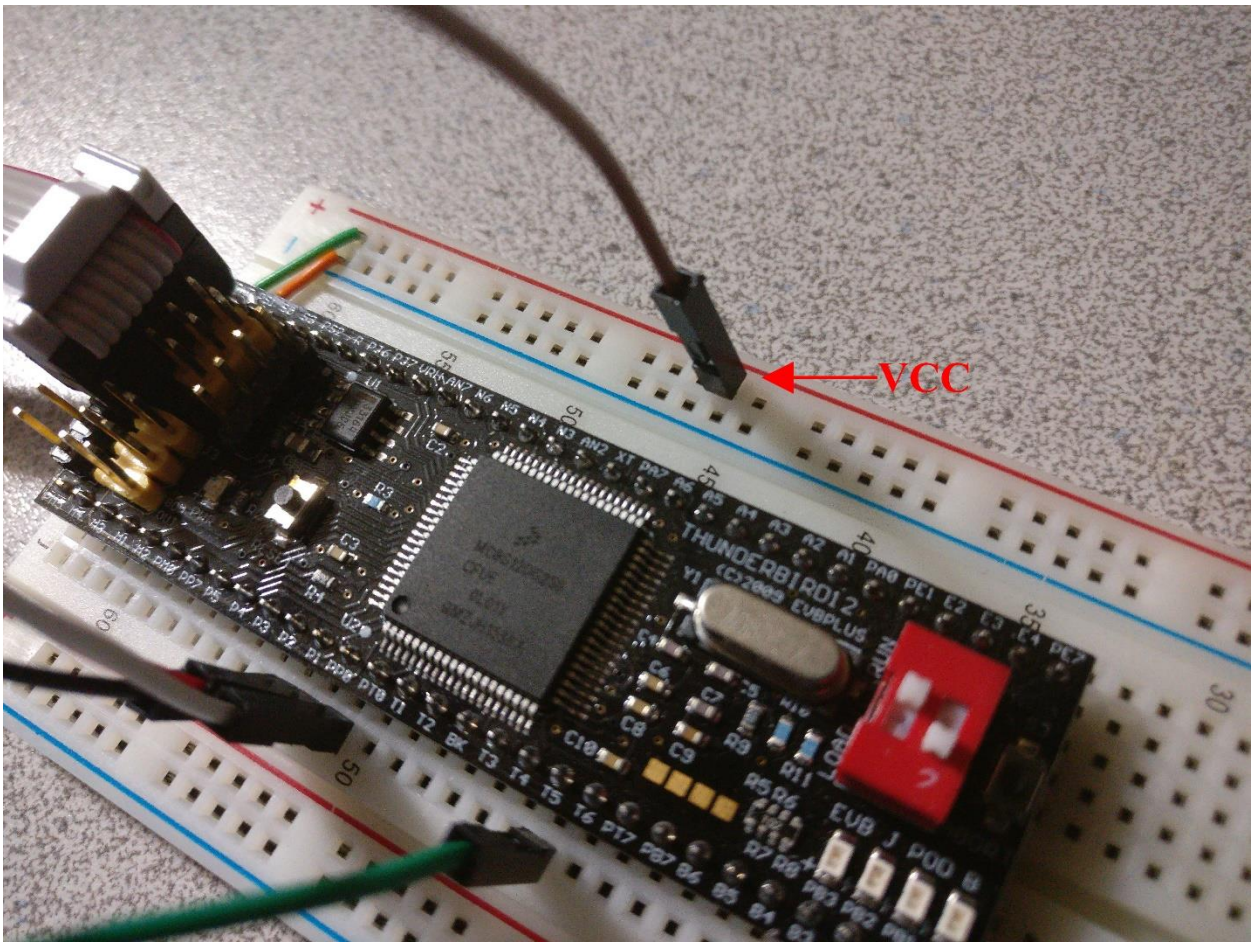


Figure 8-14 – RGB LED module connected to VCC

7. Connect the relay module to the Thunderbird12.
    a. Connect the female end of one of the 3-wire male-to-female wires to the relay module as shown in Figure 8-15.

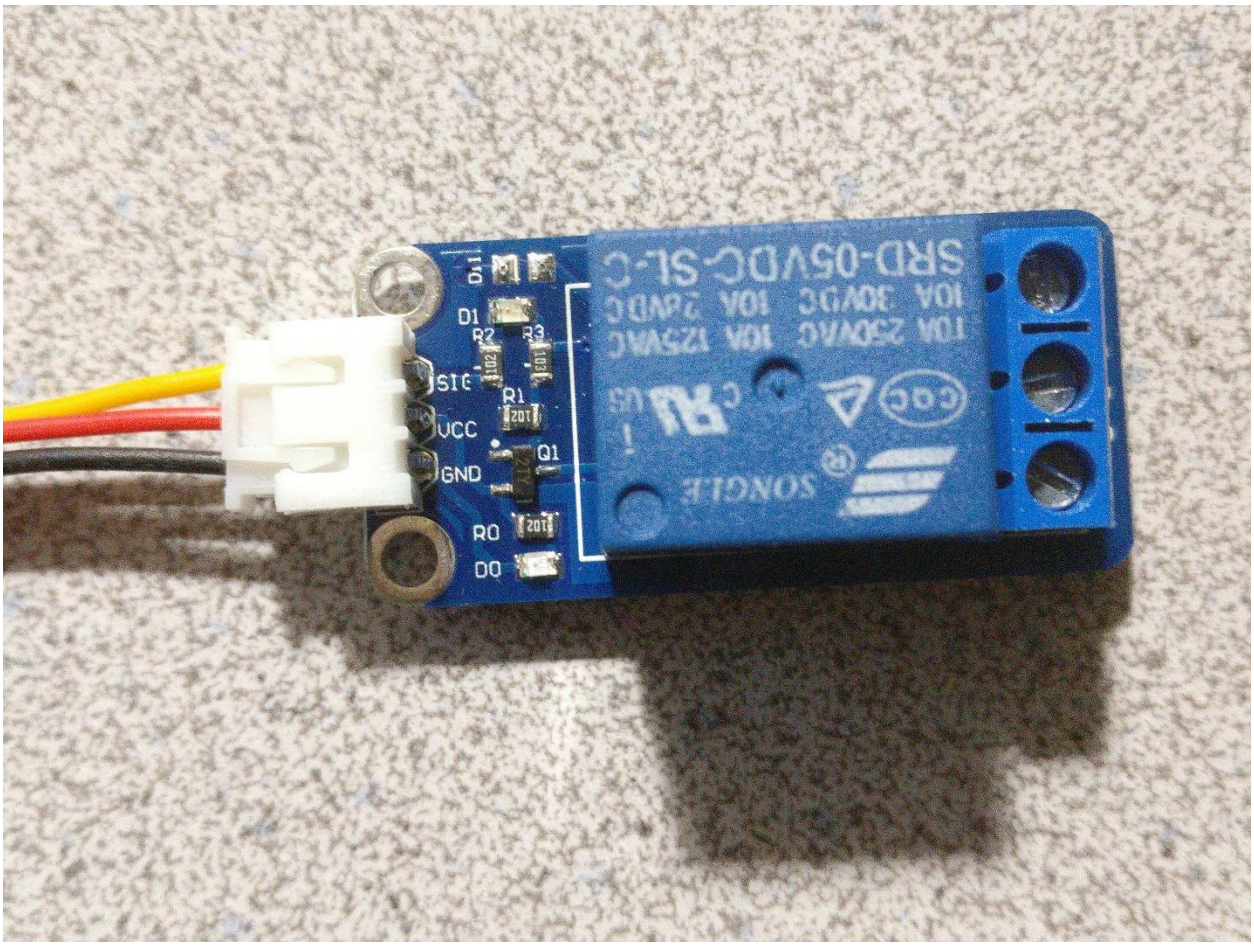


Figure 8-15 – 3-wire male-to-female wire connected to Relay Module (LOW)

b. Connect the VCC and GND wires from the relay module to the positive and negative rails, respectively, and the SIG wire from the relay module to B4 on the bottom row of the Thunderbird. Refer to Figure 8-16 for an example of what the wiring should look like.

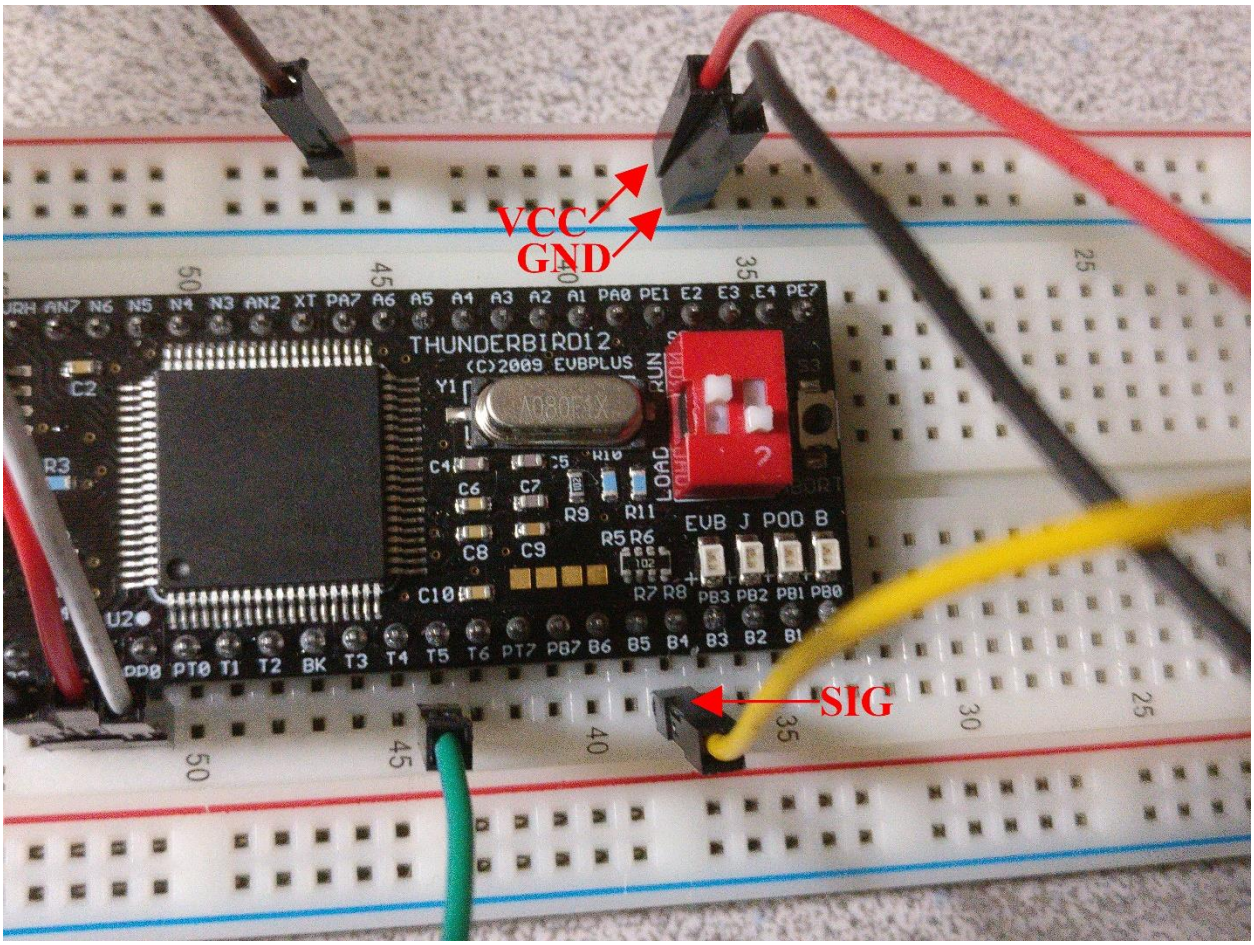


Figure 8-16 – Relay Module (LOW) connected to Thunderbird12

8. Plug the included Mini-B USB cable into the BDM. Refer to Figure 8-17 for an example of what the cable should look like plugged into the BDM.

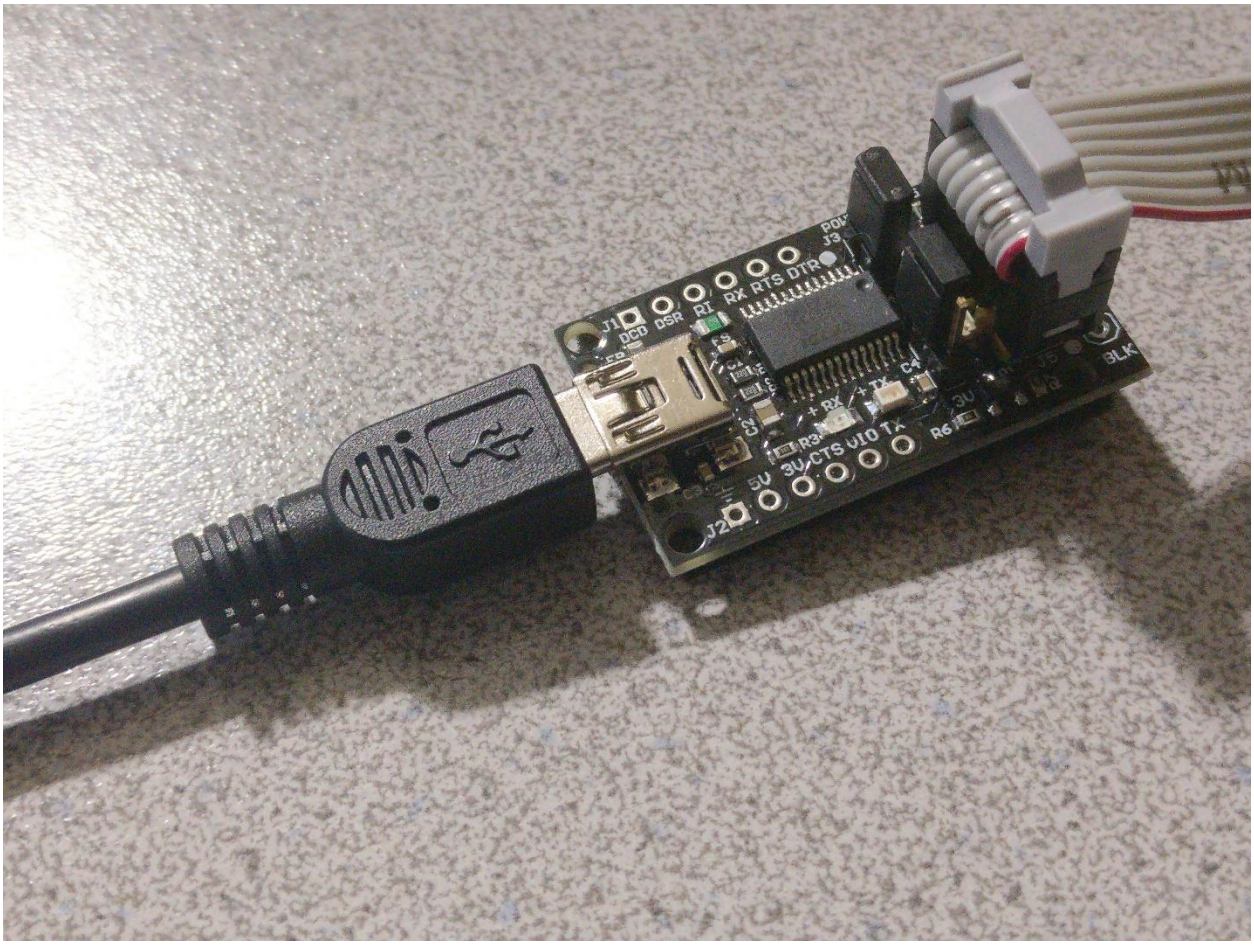


Figure 8-17 – Mini-B USB cable plugged into the Thunderbird12's Board Distribution Manager (BDM)

9. When the upload completes, unplug the Thunderbird12 from the computer.

Section 9 – Setting up CodeWarrior

In order to program the Thunderbird, a legacy, or old, version of CodeWarrior is used. CodeWarrior is a free download from the official NXP website. As with the Arduino IDE, CodeWarrior allows for custom user libraries; however, in this project, they will be added to the project folder itself, rather than the program referencing the library.

1. Download and install CodeWarrior HC12 v5.1 Special Edition using default settings. If prompted to install device or driver software, install that as well.
2. Download the LBE_Thunderbird library.
3. Extract the downloaded ZIP file to any permanent storage location, such as a hard drive or USB drive. In this documentation, the Windows archive software will be used.
    a. Left click on the downloaded file to select it.
    b. Right click on the downloaded file.
    c. Left click on the “Extract All…” option.
    d. The default extraction directory is the same directory as the ZIP file. To modify it, either type in the desired directory or click “Browse…,” find the desired directory, and click “Select Folder.” For this documentation, the extracted directory will be the untouched. Refer to Figure 9-1 for illustrated instructions.

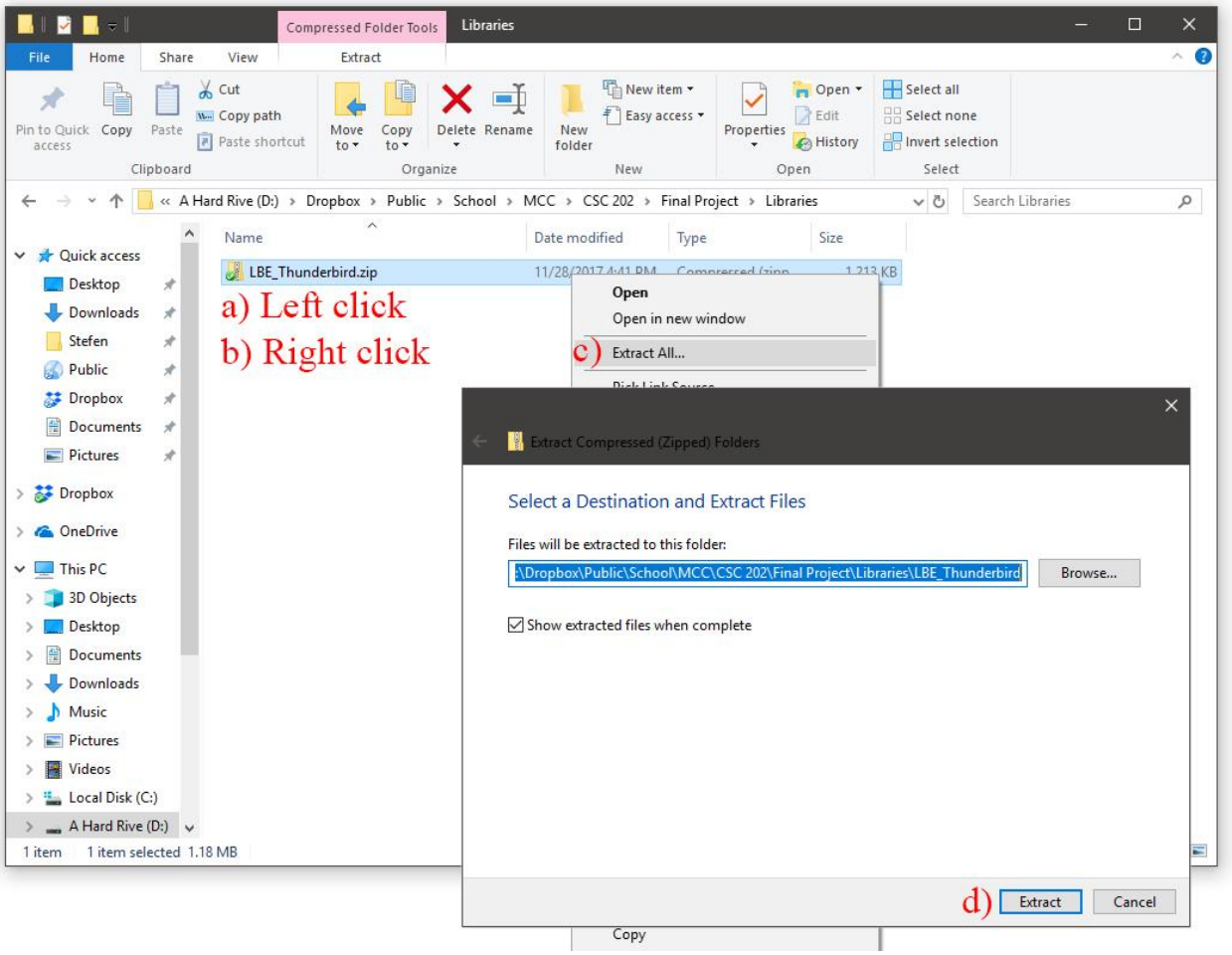


Figure 9-1 – Extracting LCD_Thunderbird software library

Section 10 – Programming the Thunderbird12

Once the Thunderbird12 and Codewarrior are both set up, the Thunderbird can be programmed. It is not vital that this be done after setting the Thunderbird up, but highly recommended, as some code may malfunction if a sensor or other device is not properly connected to the board.

1. To easily determine the COM (Communication) port of the Thunderbird12, the Arduino IDE will be used.
    a. If it is not already open, open up the Arduino IDE.
    b. Within the Arduino IDE, go to “Tools” in the toolbar, go to the option starting with “Ports,” and make note of every entry.
    c. Plug the USB-A (“regular” USB) end of the Mini-B USB cable that’s connected to the Thunderbird12 into the computer.
    d. To force the list to repopulate, close the Tools menu and open it up again, and once again go to the option beginning with “Ports.” Whatever entry was not there before is the COM port of the Thunderbird.
    e. Close the Arduino IDE.

2. Press the button labled “RESET” on the Thunderbird as shown in Figure 10-1.

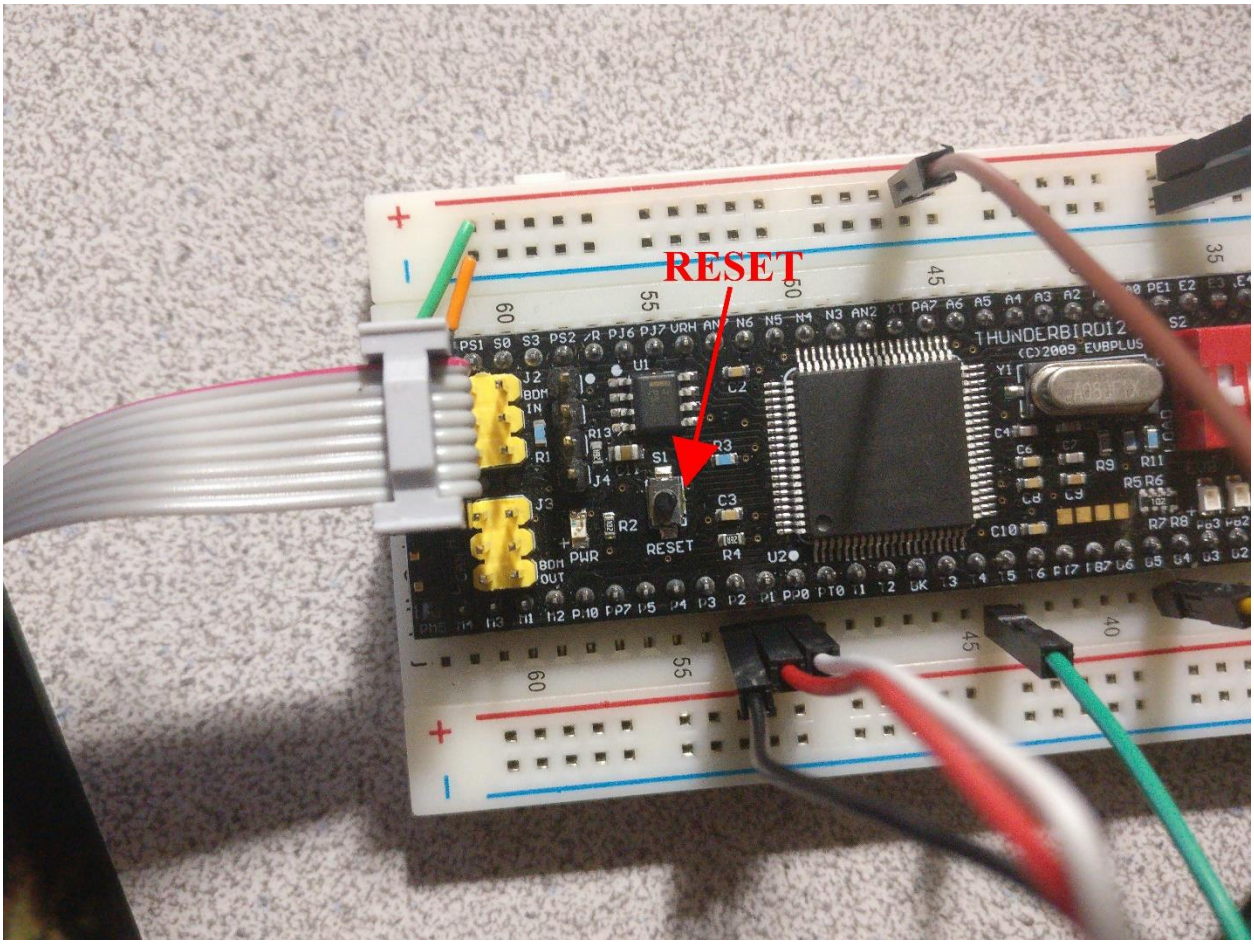


Figure 10-1 – RESET button on Thunderbird12

3. Download the Thunderbird project files.
4. Extract the project files to any permanent storage location, such as a hard drive or USB drive. The location of the extracted files does not matter, so long as they are accessible by CodeWarrior.
5. Enter the “DoorLock_Thunderbird” directory.
6. Double click on the “DoorLock_Thunderbird.mcp” file.
7. When CodeWarrior loads, ensure the communication mode is set to “HCS12 Serial Monitor.”
8. Click on the “Debug” button.
    a. If a dialog pops up with any failure message, click the “OK” button until a dialog pops up with the title “Monitor Setup.”

b. Click on the dropdown menu next to “HOST Serial Communication Port:” and select the COM port number that was determined in step 1. Refer to Figure 10-2 for an illustrated example.

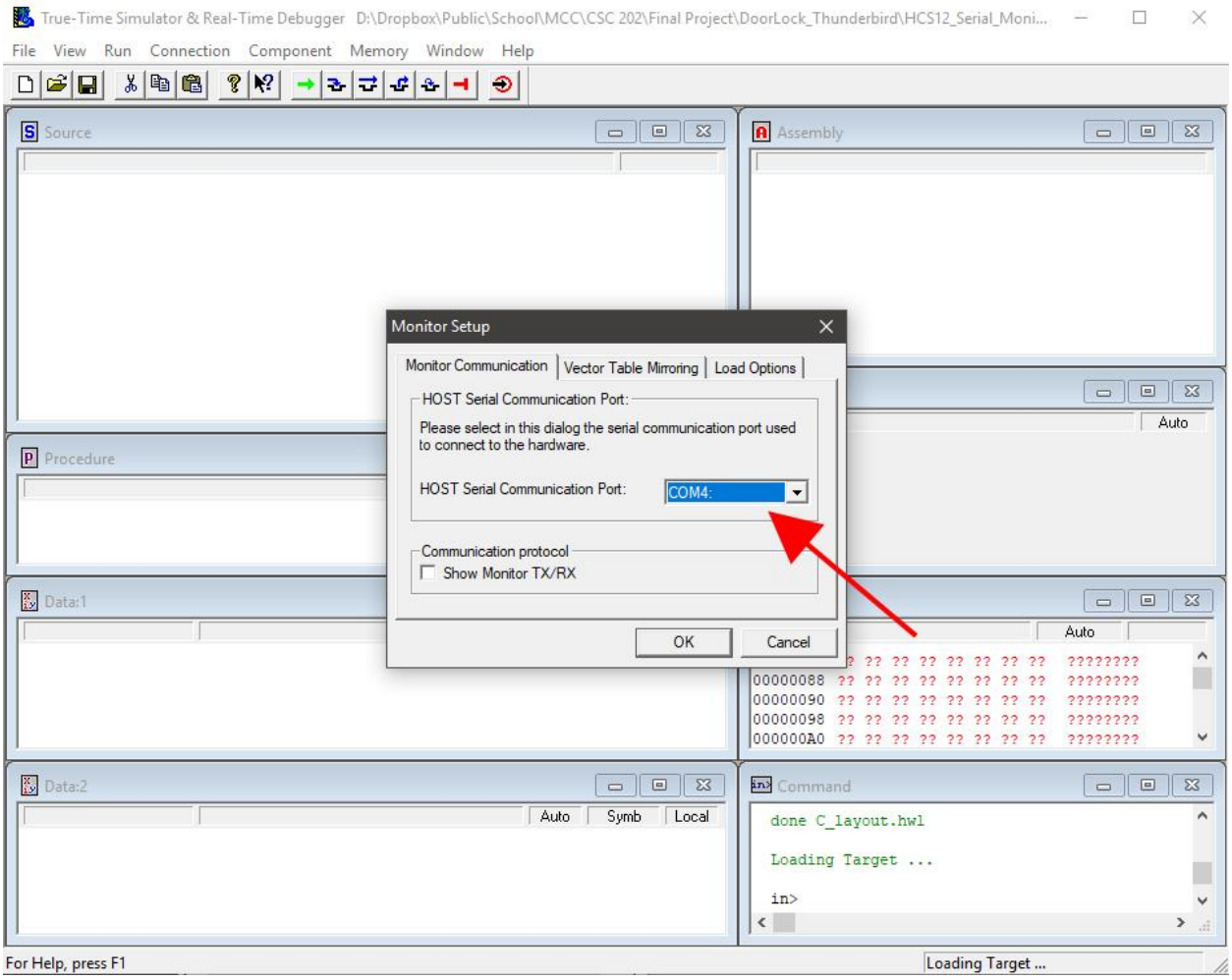


Figure 10-2 – Monitor Setup dialog pop-up in CodeWarrior

c. Hit “OK.”

9. Switch the left switch in the S2 switch section to the up position, towards the “RUN” label. Refer to Figure 10-3 for an example of what the switch configuration should look like.

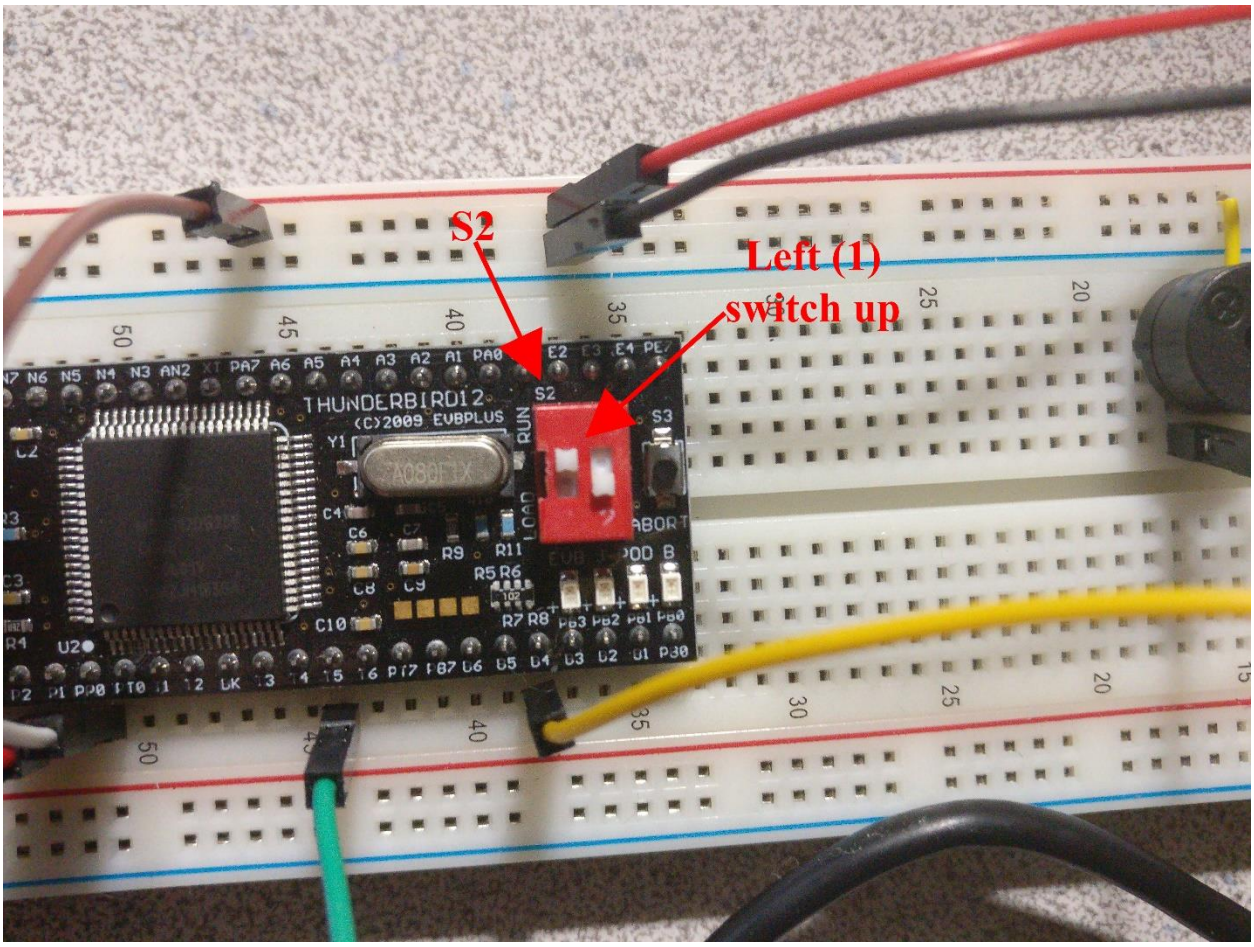


Figure 10-3 – Switch configuration on Thunderbrid12

Section 11 – Setting up the Raspberry Pi

The Raspberry Pi serves as a middleman device.

1. Plug the keyboard and mouse into the Raspberry PI's USB ports, as shown in Figure 11-1.

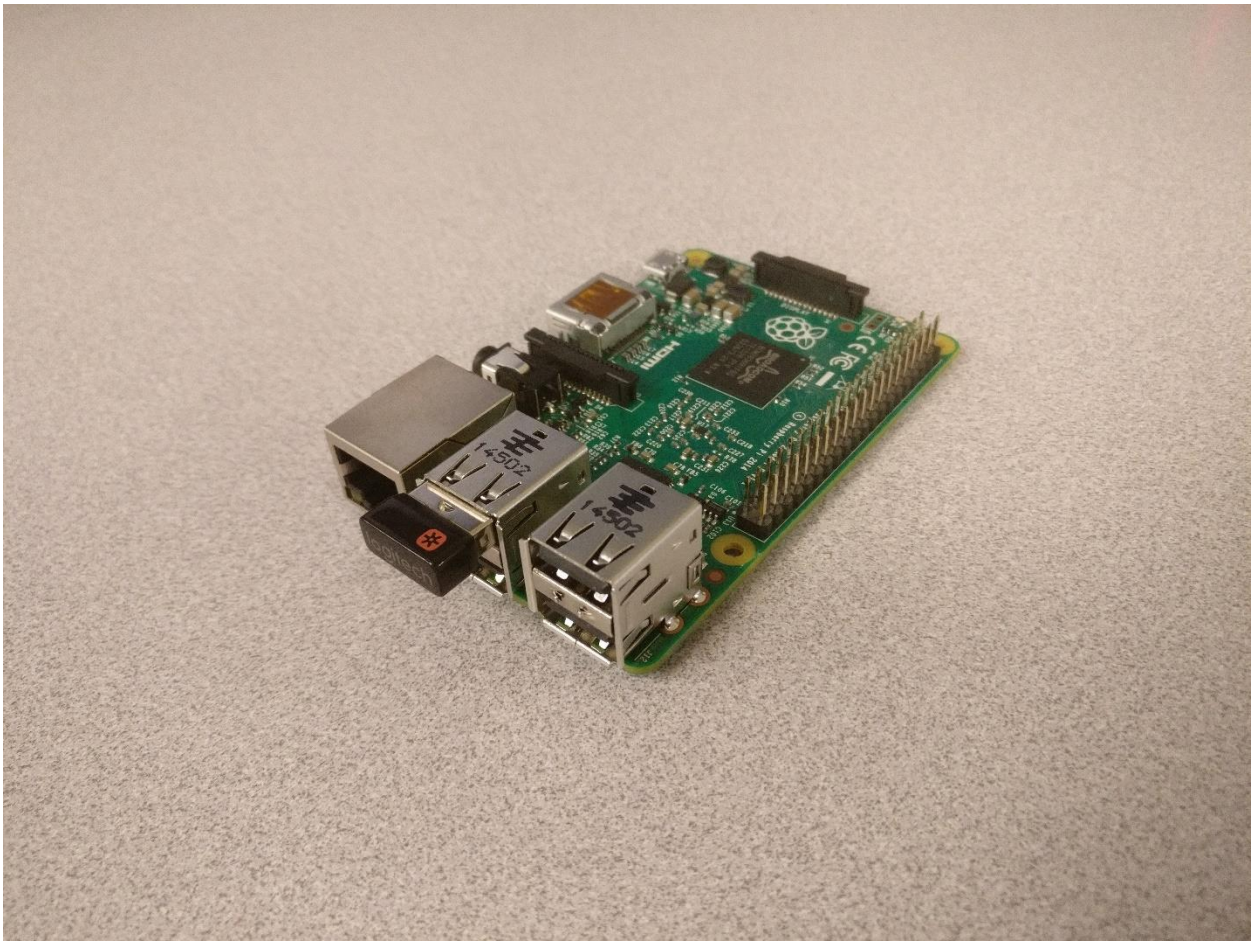


Figure 11-1 – Wireless USB dongle plugged into USB port on Raspberry Pi 2B

2. Plug the monitor's HDMI (or DVI-I-to-HDMI) cable into the Raspberry Pi's HDMI port, as shown in Figure 11-2.

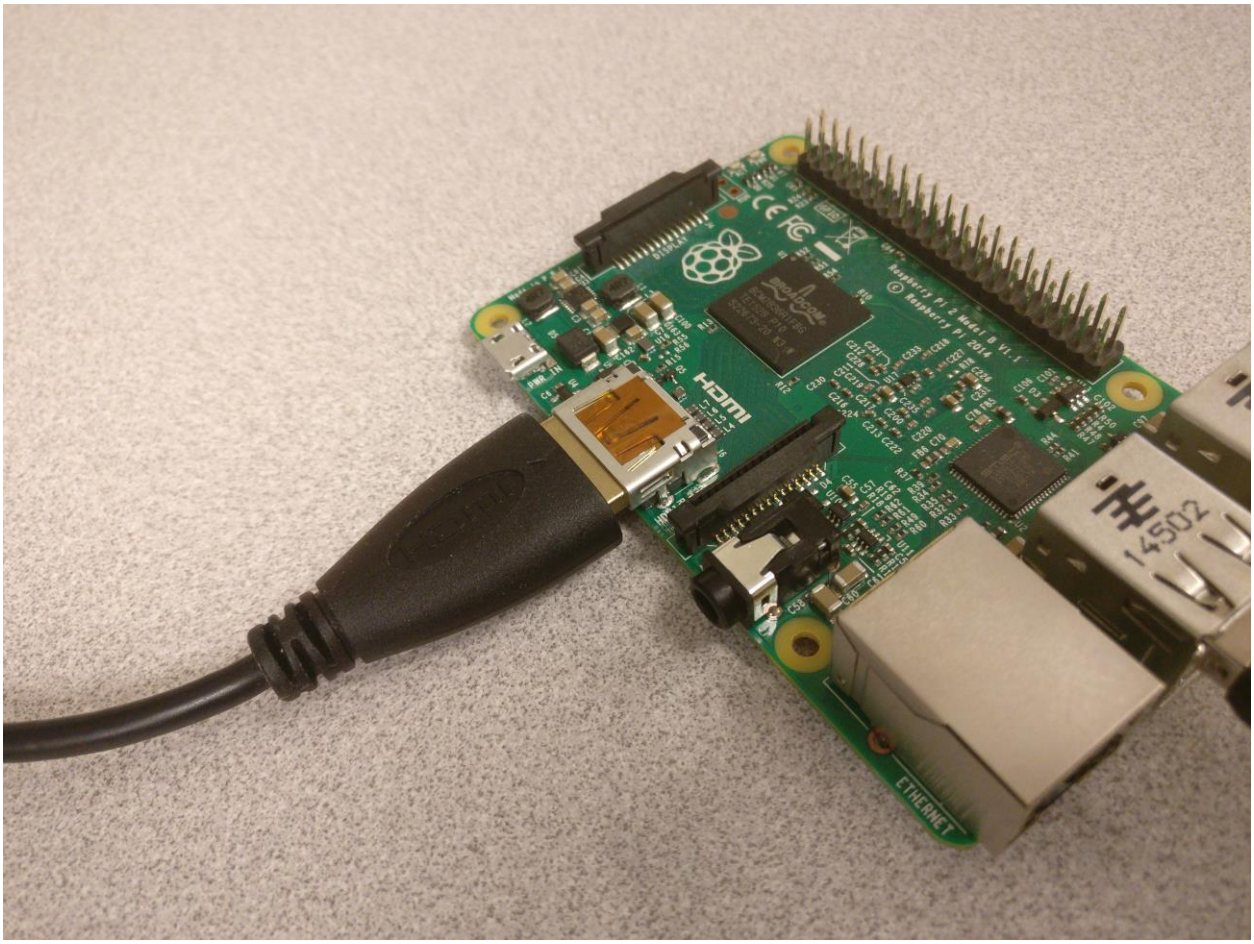


Figure 11-2 – HDMI cable plugged into HDMI port on Raspberry Pi 2B

3. Depending on the networking setup, plug in either an Ethernet cable or a USB Wi-Fi adapter. This documentation will be using an Ethernet cable for simplicity. For information on how to connect to a Wi-Fi network, visit the Raspberry Pi Foundation's official documentation:
https://www.raspberrypi.org/documentation/configuration/wireless/
https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md

4. Plug the Raspberry Pi's DC power adapter in and connect the Micro-B USB end into the Raspberry Pi. Refer to Figure 11-1 for an example of what the Raspberry Pi should look like.

5. Install Raspbian via the NOOBS (New Out Of the Box Software) on-screen prompts. For additional information, visit the Raspberry Pi Foundation's official documentation:
https://www.raspberrypi.org/help/noobs-setup/

6. Open up a terminal session by hitting Ctrl, Alt, and T simultaneously on the keyboard or by finding the terminal icon shown in Figure 11-3.

Figure 11-3 – Terminal icon in Raspbian

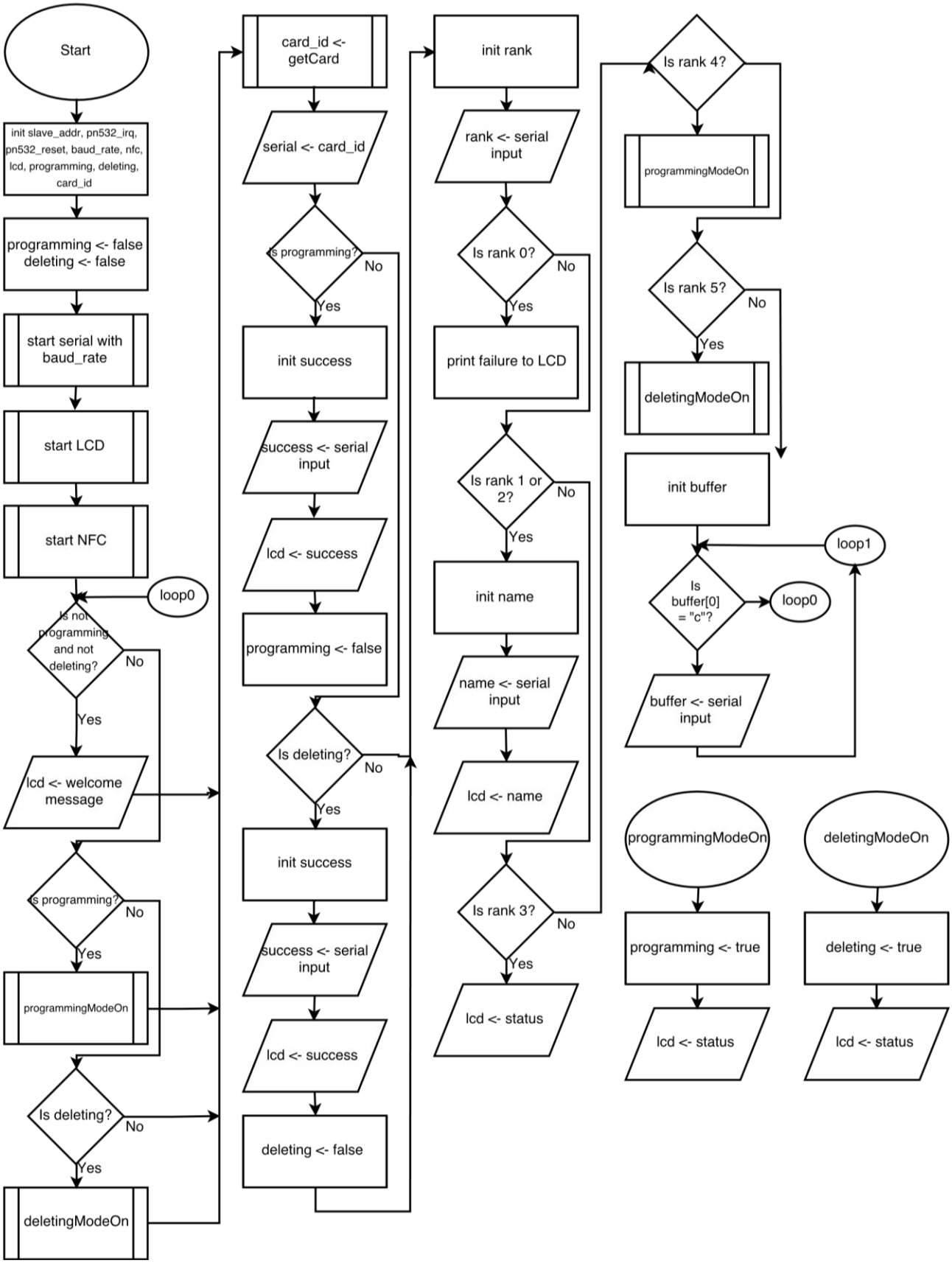
7. Install the required software by typing in the following commands into the terminal in order:

```
sudo apt-get update

sudo apt-get upgrade -y

sudo apt-get install arduino-core wget –y

sudo cd ~/Documents

sudo wget https://goo.gl/4mzkHF

sudo chmod +x DoorLock_RaspberryPi.py

python3 ~/Documents/DoorLock_RaspberryPi.py
```

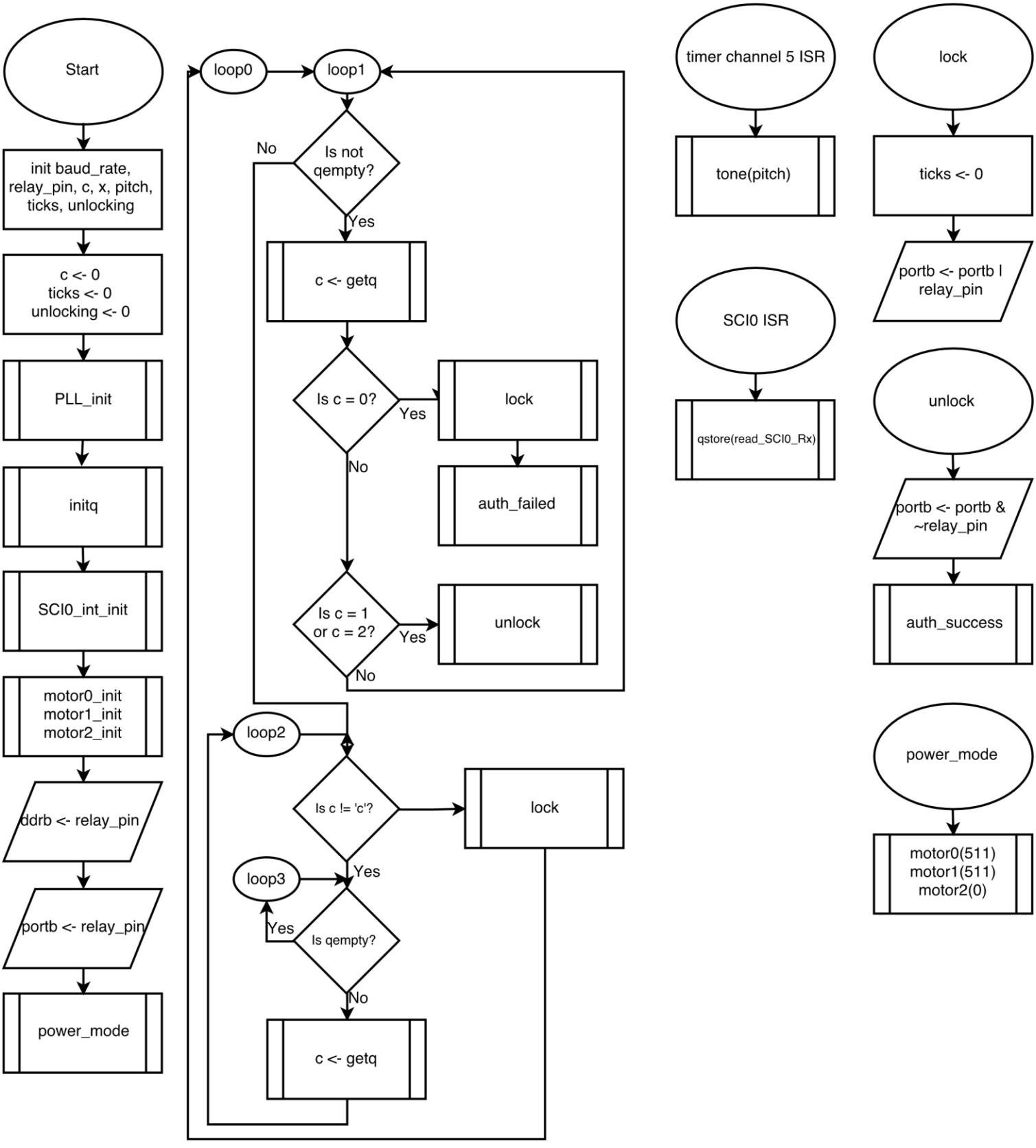
8. Occasionally, the Arduino might reset itself. This is most easily noticed by the text on the LCD disappearing for a few seconds. In the event this happens, allow up to 10 seconds for it to finish.

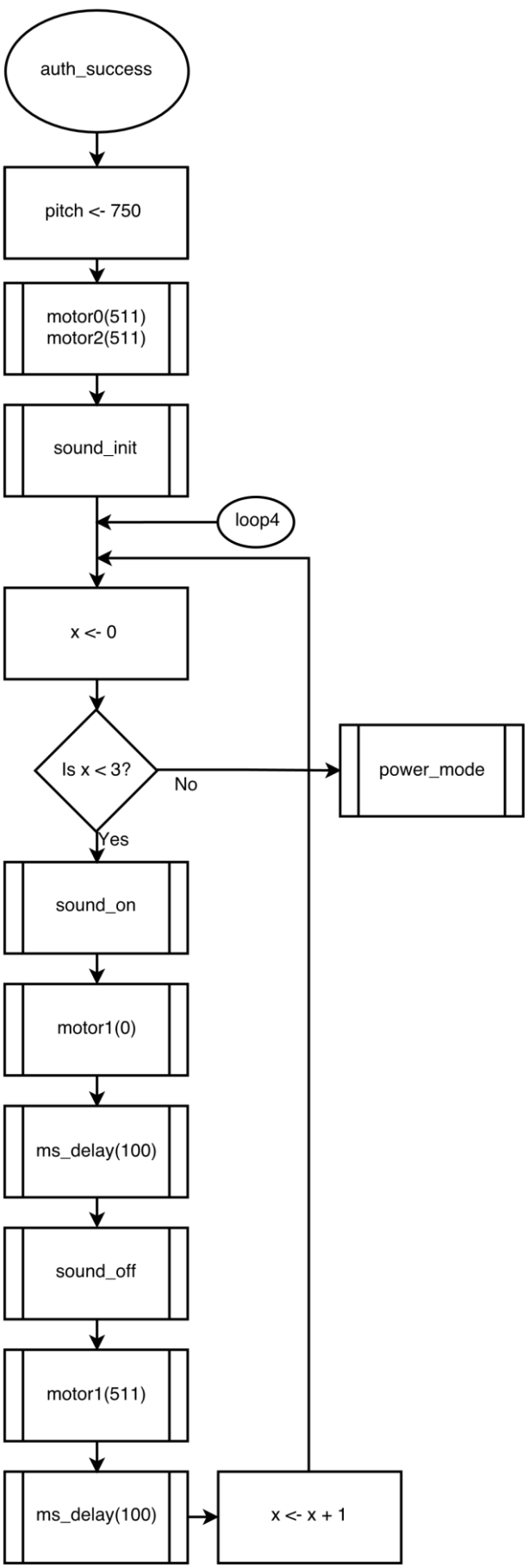
Section 12 – Logic Flow of Source Code

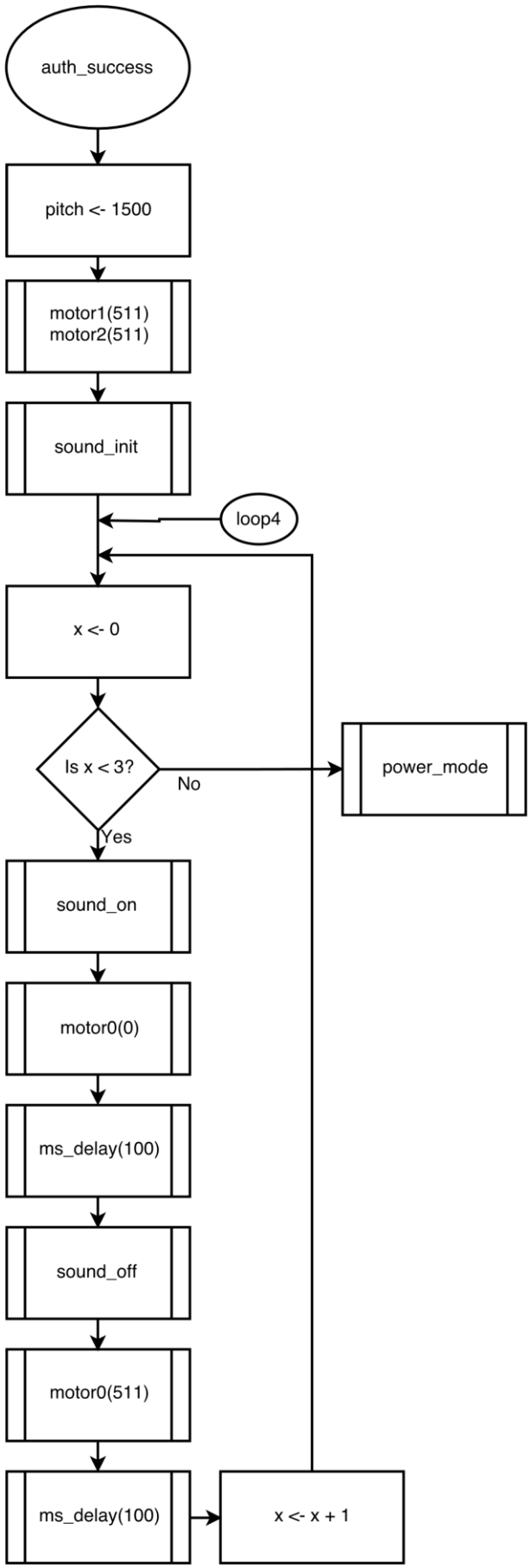
Arduino Uno



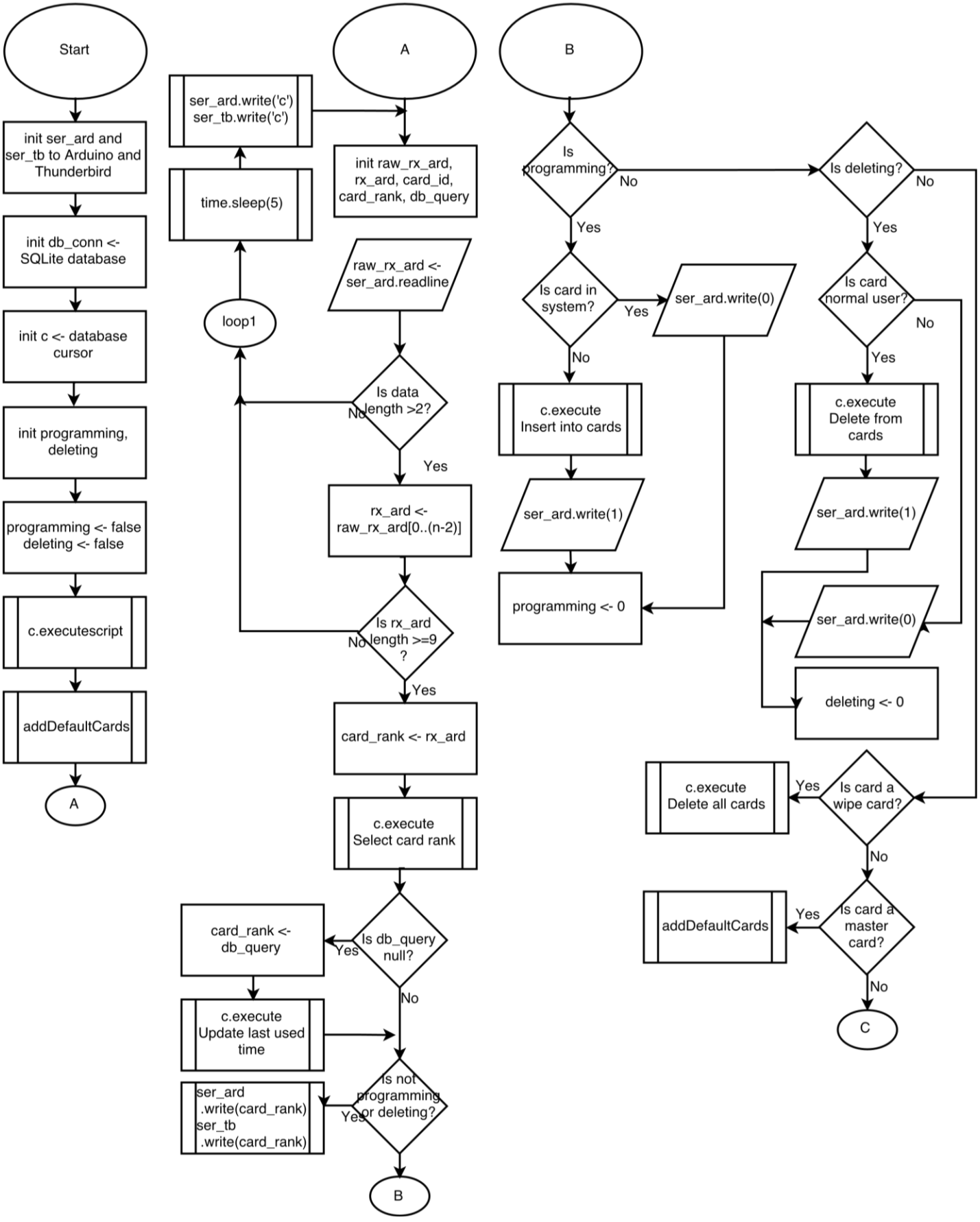
Thunderbird12

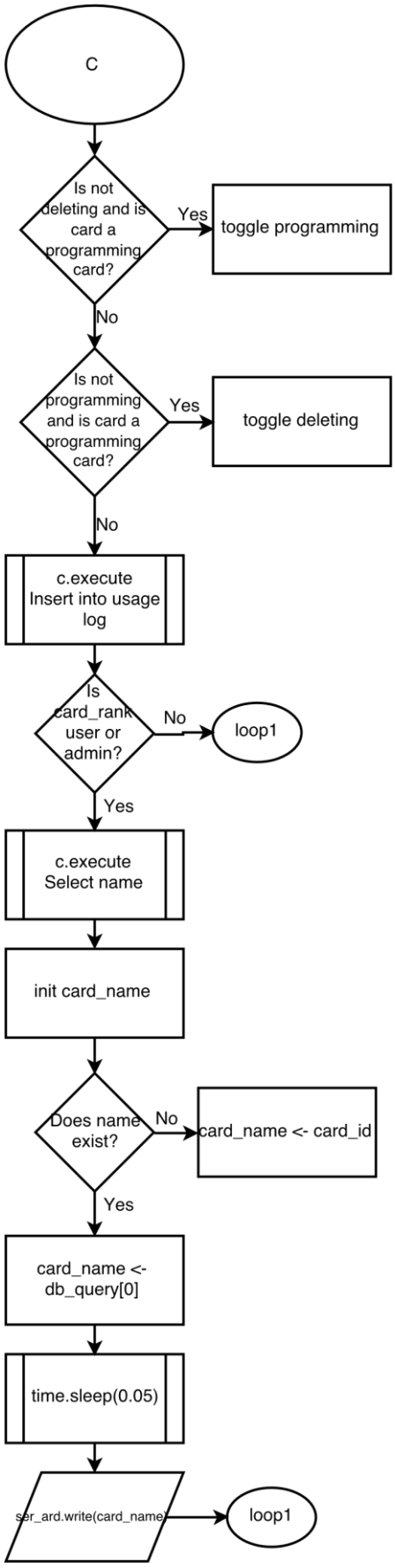


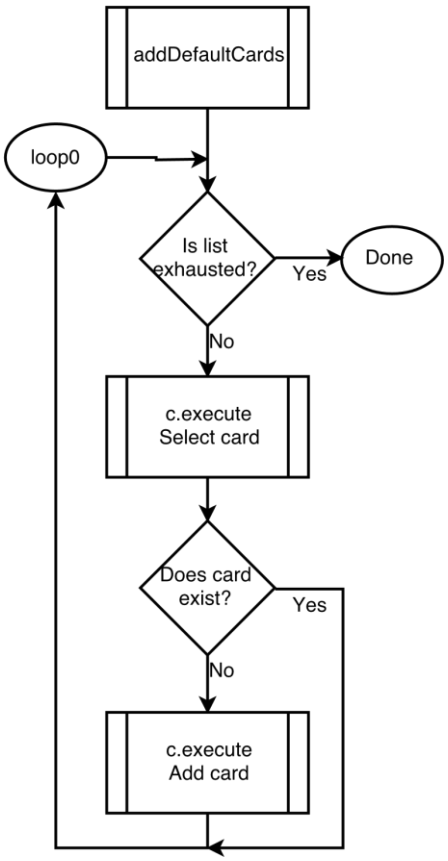

auth_success
pitch <- 750
motor0(511)
motor2(511)
sound_init
loop4
x <- 0
Is x < 3?
No
power_mode
Yes
sound_on
motor1(0)
ms_delay(100)
sound_off
motor1(511)
ms_delay(100)
x <- x + 1


auth_success
pitch <- 1500
motor1(511)
motor2(511)
sound_init
loop4
x <- 0
Is x < 3?
No
power_mode
Yes
sound_on
motor0(0)
ms_delay(100)
sound_off
motor0(511)
ms_delay(100)
x <- x + 1

Raspberry Pi




C
Is not deleting and is card a programming card?
Yes
toggle programming
No
Is not programming and is card a programming card?
Yes
toggle deleting
No
c.execute Insert into usage log
Is card_rank user or admin?
No
loop1
Yes
c.execute Select name
init card_name
Does name exist?
No
card_name <- card_id
Yes
card_name <- db_query[0]
time.sleep(0.05)
ser_ard.write(card_name)
loop1


addDefaultCards
loop0
Is list exhausted?
Yes
Done
No
c.execute Select card
Does card exist?
Yes
No
c.execute Add card