Group Member Names: James Mathlin (Hardware)

Levi Poling (Python Coding)

Beck Lindstrom (Arduino Coder)

Naomi Grant (Documentation)

Course and Quarter: Engineering 114, Winter 2019

Date: 03/05/2019

**To Light Or Not To Light A Fish Tank**

**Problem Statement:**

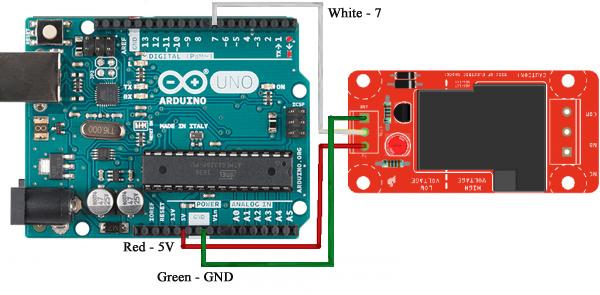
Our group was tasked with finding a way toturn the fish tank light on and off remotely.

**Hardware Setup:**

**Bill of Materials:**

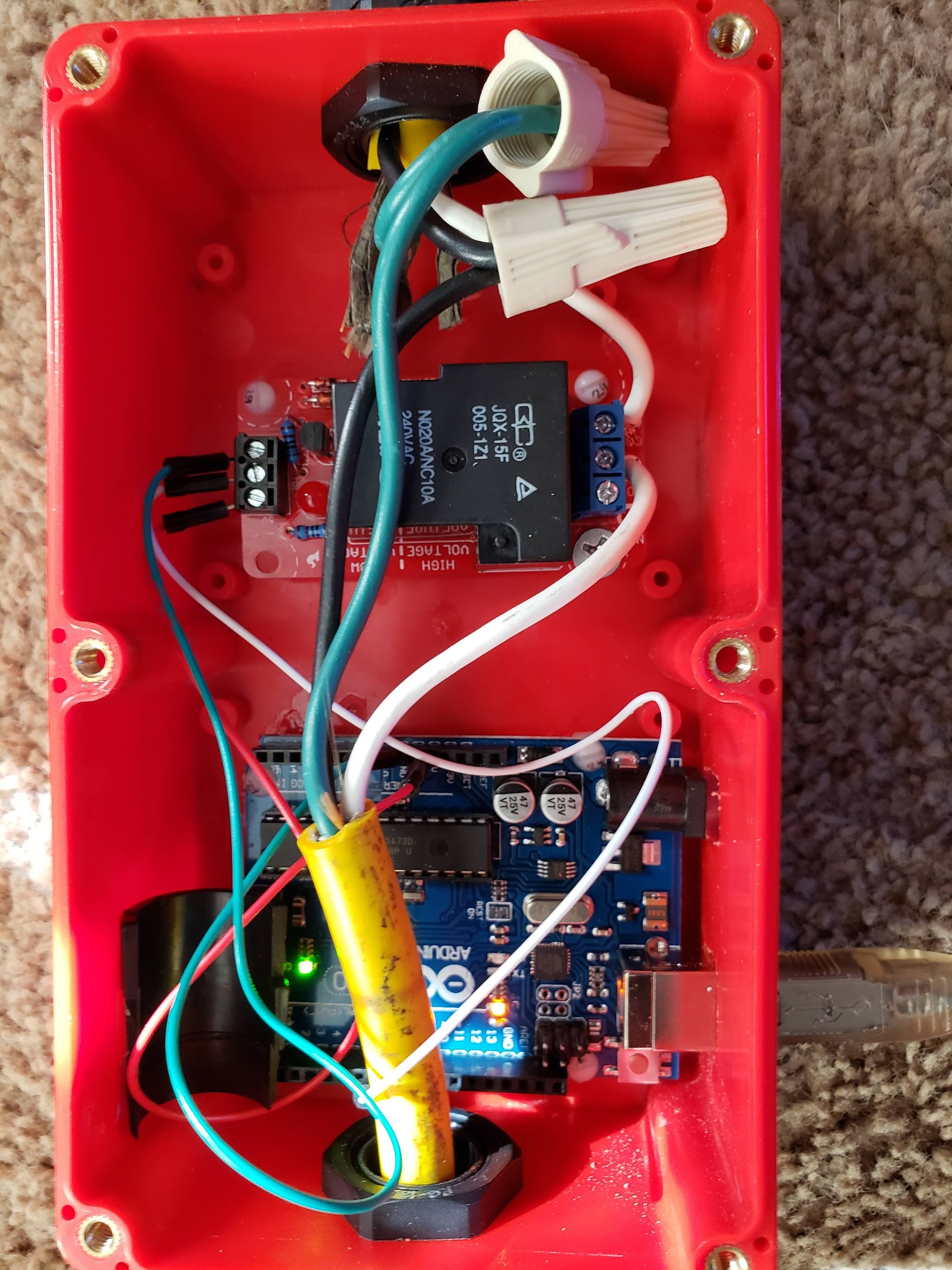
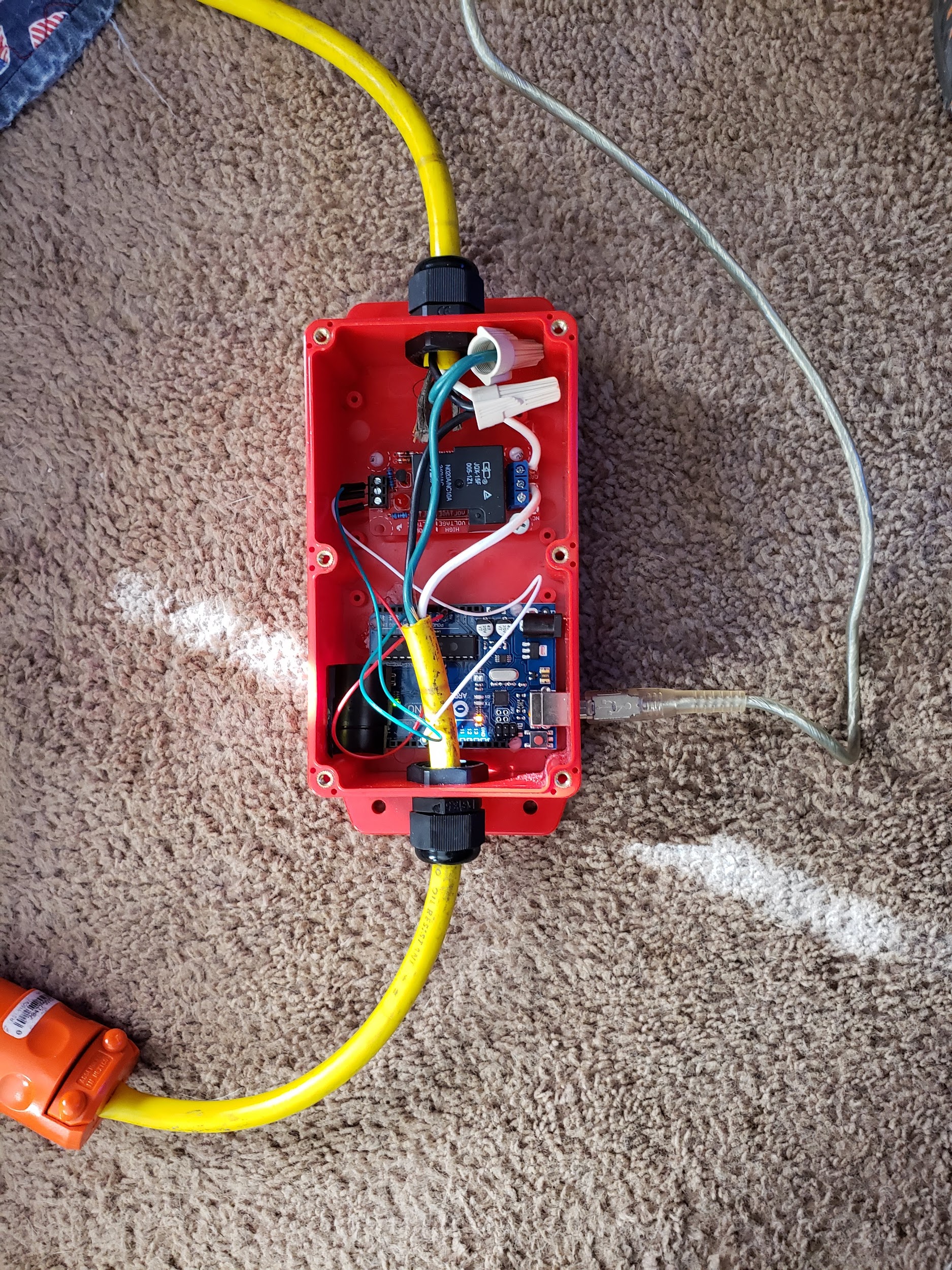
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Part Name** | **Purpose** | **Item Number** | **URL** | **Price** |
| Junction Box | Hide components | 11366 | https://www.sparkfun.com/products/11366 | $8.95 |
| Wire Nuts | Connect Wires | 778329 | https://www.homedepot.com/p/Commercial-Electric-Winged-Wire-Connector-Assortment-30-Pack-778329/302686295 | $3.96 |
| Sparkfun Relay | Convert from 5V to 120V | 13815 | https://www.sparkfun.com/products/13815 | $11.95 |
| Arduino Uno Board | Run code for program | 11021 | https://www.sparkfun.com/products/11021 | $22.95 |
| Power Cord | Supply power | MD261 | https://www.homedepot.com/p/AC-WORKS-AC-Connectors-15-ft-14-3-SJTW-15-Amp-Hospital-Medical-Grade-PowerCord-with-IEC-C19-MD261/303426675 | $25.99 |
| Plastic Nuts | Mounting | 94812A300 | https://www.mcmaster.com/94812a300 | $6.37 |
| Plastic Machine Screws | Mounting | 94735A728 | https://www.mcmaster.com/94735a728 | $6.03 |

**Hardware Schematic:**

****

**Hookup Guide:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Part** | **Pin** | **Connector** | **Pin** | **Part** |
| Arduino Uno | 5V | Red Wire | 5V | Sparkfun Relay |
| Arduino Uno | GND | Green Wire | GND | Sparkfun Relay |
| Arduino Uno | 7 | White Wire | CTRL | Sparkfun Relay |

**Images:**

**Python and Arduino Code:**

Python Code:

**Main Python Code**

- ENGR 114

- Winter 2019

def status(usr\_port):

"""

The status function is used to receive the requested user input from the light function and output the desired status code to the Arduino board.

The status function requires that you set the communication serial port that the hardware is plugged into. Go to your Hardware section of your computer to determine the correct port and enter the port number only. i.e... if the port is COM2, enter 2 into the status function.

The command should be:

\*status(2) \* Make sure to replace the 2 with the specific COM port your computer is using or this function will not do as intended.

"""

import requests

import serial

import time

base\_url = 'https://api.thingspeak.com/channels/714178/fields/1.json?api\_key='

key='O8UW05CXGWM34FYV'

mid\_url = '&results'

end\_url = '='

results\_num = '1'

url = base\_url+key+mid\_url+end\_url+results\_num

ser = serial.Serial(f'COM{usr\_port}',9600)

r = requests.get(url)

json\_data = r.json()

data = int(json\_data['feeds'][0]['field1'])

if data == 1:

time.sleep(2)

ser.write(b'N')

print('The light is currently on!')

init\_data=1

elif data == 0:

time.sleep(2)

ser.write(b'F')

print('The light is currently off!')

init\_data=0

elif data>1:

time.sleep(2)

ser.write(b'F')

print('The light is currently off!')

init\_data=3

while not (data>=5):

r = requests.get(url)

json\_data = r.json()

curr\_data=int(json\_data['feeds'][0]['field1'])

if curr\_data==init\_data:

pass

elif curr\_data == 1:

time.sleep(2)

ser.write(b'N')

print('Your request has been processed and the light is now on!')

init\_data=1

elif curr\_data == 0:

time.sleep(2)

ser.write(b'F')

print('Your request has been processed and the light is now off!')

init\_data=0

elif curr\_data>1:

time.sleep(2)

ser.write(b'F')

ser.close()

time.sleep(10)

r = requests.get('https://api.thingspeak.com/update?api\_key=C20UYZD60U7HNPRJ&field1=0')

json\_data = r.json()

break

print('To recieve external requests, please restart the status() function')

return

def light():

"""

The light function receives an ```on``` or ```off``` command from the user and submits the request to ThingSpeak to switch the hardware to the desired status.

Simply include the desired status:

\*light(on) or light(ON) or light(On)

\*light(off) or light(OFF) or light(Off)

\*light(quit) or light(Quit) or light(Q) or light(q)

The request will be output to ThingSpeak and must be received on the outlet side before functionality will be present. To receive the request, use the ``status(usr\_port)`` function

"""

while True:

import time

usr\_cmd=input('Would you like the light "On" or "Off" or do you want to "Quit"? ')

time.sleep(10)

while not (usr\_cmd=='on' or usr\_cmd=='On' or usr\_cmd=='ON' or usr\_cmd=='off' or usr\_cmd=='Off' or usr\_cmd=='OFF' or usr\_cmd=='Quit' or usr\_cmd=='Q' or usr\_cmd=='q' or usr\_cmd=='quit'):

usr\_cmd=input('You must select on, off or quit! ')

import requests

base\_url = 'https://api.thingspeak.com/update?api\_key='

mid\_url = '&field1='

code='C20UYZD60U7HNPRJ'

if usr\_cmd=='on' or usr\_cmd=='On' or usr\_cmd=='ON':

on\_off = '1'

url = base\_url+code+mid\_url+on\_off

r = requests.get(url)

json\_data = r.json()

print('Must have been too dark for you!')

elif usr\_cmd=='off' or usr\_cmd=='Off' or usr\_cmd=='OFF':

on\_off = '0'

url = base\_url+code+mid\_url+on\_off

r = requests.get(url)

json\_data = r.json()

print("Ok, but the dark scares me!")

elif usr\_cmd=='Quit' or usr\_cmd=='Q' or usr\_cmd=='q' or usr\_cmd=='quit':

on\_off = '3'

url = base\_url+code+mid\_url+on\_off

r = requests.get(url)

json\_data = r.json()

print("You have removed your capability to choose and it must be restarted at the hardware!")

Break

Arduino Code:

/\*

Physical Pixel

An example of using the Arduino board to receive data from the computer. In

this case, the Arduino board turns on an attached light when it receives the character

'N', and turns off the light when it receives the character 'F'.

The data can be sent from the Arduino Serial Monitor, or another program like

Processing (see code below), Flash (via a serial-net proxy), PD, or Max/MSP.

The circuit:

- Relay connected to digital pin 7 to ground with light wired to relay

created 2006

by David A. Mellis

modified 30 Aug 2011

by Tom Igoe and Scott Fitzgerald

modified 8 Mar 2019

by Beck

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/PhysicalPixel

\*/

const int relay = 7; // the pin that the LED is attached to

int incomingByte; // a variable to read incoming serial data into

void setup() {

// initialize serial communication:

Serial.begin(9600);

// initialize the LED pin as an output:

pinMode(relay, OUTPUT);

}

void loop() {

// see if there's incoming serial data:

if (Serial.available() > 0) {

// read the oldest byte in the serial buffer:

incomingByte = Serial.read();

// if it's a capital N (ASCII 72), turn on the LED:

if (incomingByte == 'N') {

digitalWrite(relay, HIGH);

}

// if it's an F (ASCII 76) turn off the LED:

if (incomingByte == 'F') {

digitalWrite(relay, LOW);

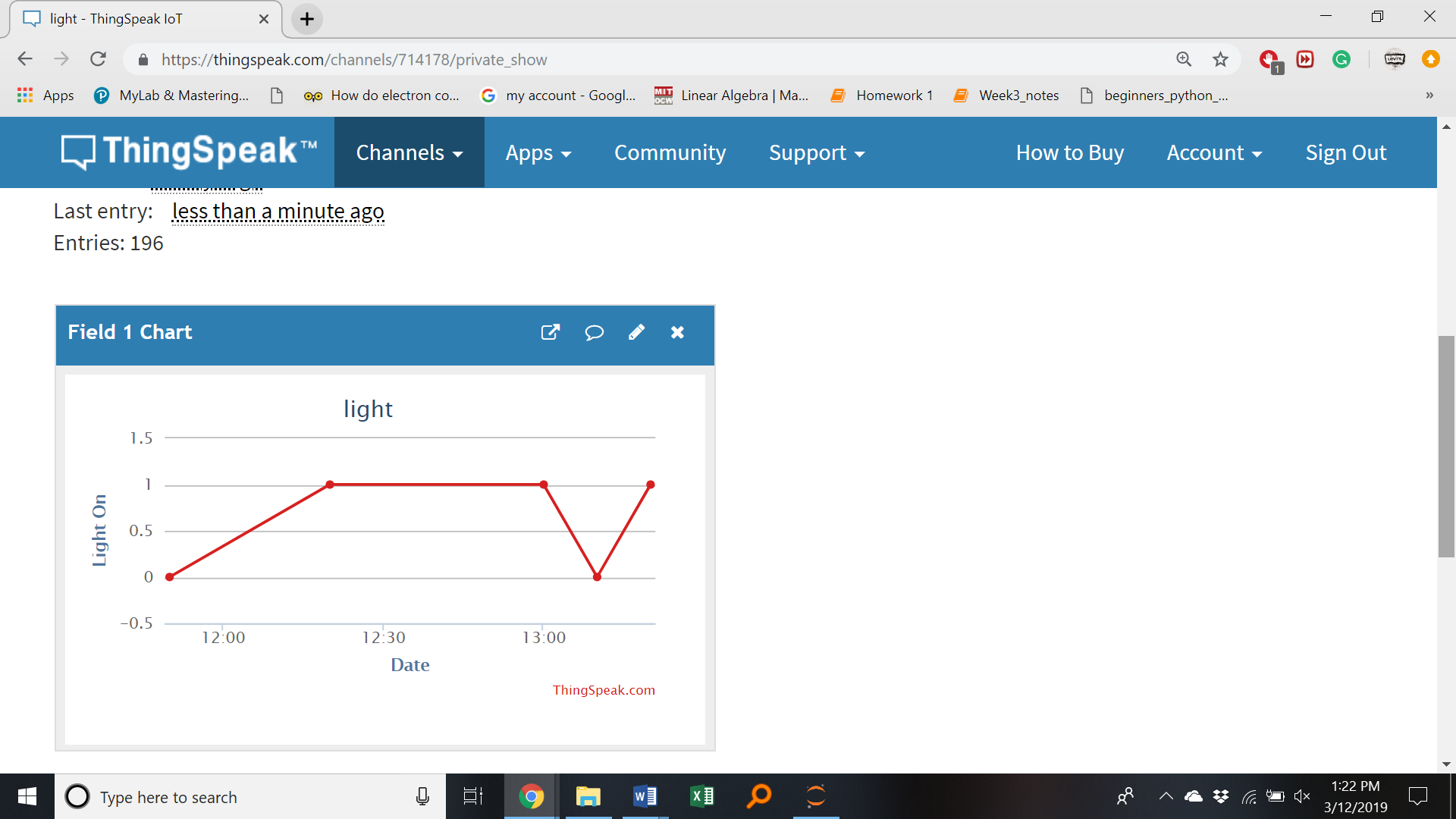
}

}

}

**Results Plot:**

A results plot of whether the light is on or off at the shown times (x-axis). Zero means ‘off’ and 1 means ‘on’.



**Future Work:**

If another group of students were to build on this project, they could add a light sensor so that the light would come on when the other lights in the house go on or when the sun goes up and down. They could use resources in Github to help them.

**License:**

BSD 2-Clause ‘Simplified’ License