Final Project Report

Light and Proximity Detection Mirror

CSCE 462 Microcomputer Systems

Date: December 8, 2021

Rose Soriano | laurensori@tamu.edu
Kazuoki Tokuno | kazuoki.tokuno@tamu.edu
Nikhitha Reddy Vempati | nikhitha06@tamu.edu

Table of Contents

Section 1: Executive Summary

This document provides all the details about the process of our project that is intended to be a “magic” mirror that detects the amount of light in the room based on current brightness from various factors (sunlight through a window, other lights in the room that are on), and also detect motion when the user walks in front of it in order to be able to turn on its vanity lights to the appropriate brightness. Detecting the amount of light in the room will turn on different amounts of light LED’s surrounding the mirror, meaning that the mirror will turn on all of the lights and give the user full brightness if the room is dark, such as later in the evening, or turn on only some of the lights to give the user only some brightening if there is already some light in the room. The mirror will turn on no lights if it detects that there is ample light already in the room. A photoresistor is the type of sensor that will allow us to read this. We used certain concepts that we learned throughout this course, including but not limited to: working with photoresistors and understanding the functionality of sensors, and programming the MCP3008 8-channel 10 bit ADC chip.

We utilized the GPIO library with a motion sensor that detects when the user approaches. It simply will return to an integer value of 0 or 1 depending on if it detects motion or not, which we can use as a boolean for our program to turn on the mirror or not. Once the user leaves the front proximity of the mirror, it will turn off after about 2.5 seconds of not detecting motion.

Section 2: Introduction

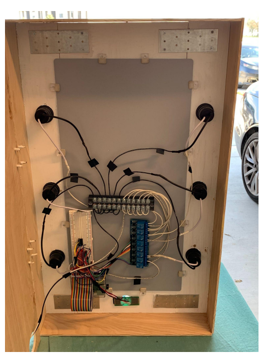
Our project is intended to be used indoors within a room that has lighting adjustment variables, such as lighting devices or simple access to an outside window. Knowing that the lighting of the environment around the mirror will change, the mirror's vanity light bulbs will be at varying percentages of on/off combinations. For example, if the mirror's sensors detect that the room is completely dark, then when someone approaches, it will turn on all six of its light bulbs. If the mirror's sensors detect that it is mostly bright in the room, and that a person walks in front, then it will turn on only two of its lights. The mirror can range from having zero (while in its off state), two, four, and six light bulbs of its six total on at a time.

Some constraints for this project may be handling mains power carefully. This is solved by using a terminal strip to allow better organization of the wiring and to ensure the mechanical and electrical safety of the system by grounding, isolating, and protecting the other components in the electrical circuit. We will also have to use a photoresistor to evaluate how many light bulbs must be turned on, and this we will have to account for any variations due to improper readings from the photoresistor. The photoresistor's light reading could also be affected by any lightbulbs on the project that are turned on, which needed to be accounted for.

People intending to use this project can be users who utilize a mirror to get ready before they head out of their house for the day. It will give them beautifying vanity lights from an automated mechanism, so that they will not have to put any effort while getting ready to either turn it on or adjust the lighting to view themselves. Using no hands to use this mirror will efficiently speed up one's process in getting ready in the morning, and even provide an effortless way to check their appearance before leaving.

Section 3: System Design

The front of our project features a minimalistic design that is meant to simplify any user interaction. The user will only see six lightbulbs, and technically also our hole which shows where the motion sensor and photoresistor are positioned. Other than that, they will not have to see or understand any backend system components, such as the wiring or on/off switches. The only thing that they will need to do to use our product will be to plug it into a mains outlet. Once the program on the raspberry starts running, they will be able to activate and deactivate the mirror just by approaching it.



The back of our product shows all of our individual components that are vital for our electrical circuit. You can see that we placed the raspberry pi close to our frame's hole, so that we can feed the power cord of the raspberry pi easily through the hole (since the included lab cord for the RPi is not very long). After placing the raspberry pi, we decided to place our needed breadboard (for the MCP3008 chip and GPIO header, both from the lab kit) close to the RPi, so that we could use the lab's cables to make necessary connections between the MCP3008 chip digital->analog converter and the Raspberry Pi. The terminal block was placed near these components, but more towards the center of the back of the mirror, so that it could have close-to-equal lengths of wires going from it to the light bulb sockets.

Section 3.1: Hardware

The terminal block allows us to distribute live and neutral ends of our AC-connected outlet cable to parallel light bulb circuits. This was successfully done by daisy chaining the live (white) and neutral (black) ends. The power side of each individual light bulb circuit is connected through one of the channels of the 8-channel relay. The relay acts as a switch which will connect power from the terminal block to power on the light bulb socket when we program it to close. We made sure to plan out our wires and cut them to exact lengths so that our circuit was as neat as possible.

Our bulbs' sockets had already-attached live and neutral wires attached to them. This was a great find at Home Depot since we then did not have to worry about wiring another thing wrong ourselves.

The PIR motion sensor and the photoresistor are soldered onto a PCB board placed into a circle hole in the middle of the bottom of the wooden frame. This allows the sensors to stick through the front. We cut 22 gauge wires and soldered them to the back of the sensors on the PCB board so that we could utilize the breadboard to connect them to the raspberry pi's appropriate pins.

We used velcro straps to attach our components on the back of the mirror, and also placed the raspberry pi in a bought case, so that we could return all lab parts easily without damaging them. The mirror was attached using mirror clamps, and since the slots to hold the mirror were too big, we stuck Gorilla Mounting Putty to help stabilize it against the mirror.

Section 3.2: Software

We programmed the Raspberry Pi using Python, since there are so many useful libraries with Python that allow us to control our relays and sensors easily. After researching the APIs of all necessary components, we first tested individual components in their own scripts so that we could understand their return values and functioning. Then, we compiled all the components (the GPIO outputs for relay, the photoresistor through the MCP3008 chip, and the GPIO controlled motion sensor) in our main script and wrote carefully curated functions and nested loops for our desired end capabilities.

The direction of our software was to first wait until motion was detected for at least 2.5 seconds. We achieved this by using a queue that would have a max size of 10 and push the output value of the motion sensor. If we saw that there were at least five 1's in the queue (which was implemented using a basic list in Python), then we would turn on the light bulbs as this would be considered detected motion.

After detecting motion, our program will immediately read how much light is in the room. We have a carefully curated function to turn on either zero, two, four, or six lightbulbs depending on that photoresistor reading. At this point, the lightbulbs are now in an on state, and the program will set a boolean value to True to keep track of the current on "session", until there is no more detected motion. No detected motion, in our program, is defined by having five or less 1's in our queue. Once this is the current case, then it will turn off all the lights.

The program continuously runs a while loop so that we can keep track of how many 1's and 0's are in our queue, so that we can turn the product on or off depending on if someone is there or not.

```
#import GPIO and time
import RPi.GPIO as GPIO
from time import sleep
from gpiozero import MotionSensor, LED
import os
import time
from time import sleep
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn

# Set up Photoresistor and MCP3008 Chip
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI) # Create the spi bus
cs = digitalio.DigitalInOut(board.D22)                   # Create the CS (chip select)
mcp = MCP.MCP3008(spi, cs)                               # Create the MCP object
chan0 = AnalogIn(mcp, MCP.P0)                            # Create an analog input channel on pin 0

# Set GPIO numbering mode and define output pins
GPIO.setmode(GPIO.BCM)
GPIO.setup(26, GPIO.OUT)
GPIO.setup(19, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(16, GPIO.OUT)
GPIO.setup(6, GPIO.OUT)
GPIO.setup(20, GPIO.OUT)

# Set up PIR Sensor
pir = MotionSensor(18)

# Global variables
motionon = [0] * 10
motionoff = []
sixAreOn = False
```

```
fourAreOn = False
twoAreOn = False
zeroAreOn = True

# Calculates average of 10 elements of array to see if motion was detected (5 consecutive 1's)
def motionDetectionCount(myArr):
    count = 0
    for i in range(len(myArr)):
        if myArr[i] == 1:
            count += 1
    return count

def turnOnSix():
    global sixAreOn
    global fourAreOn
    global twoAreOn
    global zeroAreOn
    GPIO.output(26, True)
    GPIO.output(13, True)
    GPIO.output(19, True)
    GPIO.output(6, True)
    GPIO.output(16, True)
    GPIO.output(20, True)
    sixAreOn = True
    fourAreOn = False
    twoAreOn = False
    zeroAreOn = False

def turnOnFour():
    global sixAreOn
    global fourAreOn
    global twoAreOn
    global zeroAreOn
    GPIO.output(26, True)
    GPIO.output(13, True)
    GPIO.output(19, False)
    GPIO.output(6, False)
    GPIO.output(16, True)
    GPIO.output(20, True)
    sixAreOn = False
    fourAreOn = True
    twoAreOn = False
    zeroAreOn = False

def turnOnTwo():
    global sixAreOn
    global fourAreOn
    global twoAreOn
    global zeroAreOn
    GPIO.output(26, True)
    GPIO.output(13, True)
    GPIO.output(19, False)
    GPIO.output(6, False)
    GPIO.output(16, False)
    GPIO.output(20, False)
    sixAreOn = False
    fourAreOn = False
    twoAreOn = True
    zeroAreOn = False
```

```
def turnOnZero():
    global sixAreOn
    global fourAreOn
    global twoAreOn
    global zeroAreOn
    GPIO.output(26, False)
    GPIO.output(13, False)
    GPIO.output(19, False)
    GPIO.output(6, False)
    GPIO.output(16, False)
    GPIO.output(20, False)
    sixAreOn = False
    fourAreOn = False
    twoAreOn = False
    zeroAreOn = True

def lightIntensity():
    if chan0.value > 60000:
        print('light value w six bulbs ', chan0.value)
        turnOnSix()
    elif chan0.value >= 55000 and chan0.value < 60000:
        print('light value w four bulbs ', chan0.value)
        turnOnFour()
    elif chan0.value >= 40000 and chan0.value < 55000:
        print('light value w two bulbs ', chan0.value)
        turnOnTwo()
    else:
        print('light value w zero bulbs ', chan0.value)
        turnOnZero()

# START; turn off all lights
turnOnZero()

# If motion is detected, append 1. Else append 0. Pop 1st element so list only has 10 elements
hasApproached = False
initialLight = 0
heldFlashlight = 0
while True:
    if pir.value == 1:
        print('motion detected')
        motionon.pop(0)
        motionon.append(1)
        print(motionon)
    else:
        print('no motion')
        motionon.pop(0)
        motionon.append(0)
        print(motionon)
    # If motion was detected from avg
    if motionDetectionCount(motionon) > 6: # Light is on
        if hasApproached == False:
            initialLight = chan0.value
            lightIntensity()
            hasApproached = True
    else:                                   # Light is off
        print('Lights are off')
        hasApproached = False
        turnOnZero()
    sleep(0.5)
    print('Light value: ', chan0.value)
```

```
    print("-------------------")

# Cycle through relay depending on PIR sensor
GPIO.cleanup()
```

Section 4: Conclusion

Our magic mirror is an easy product to use that allows a user to view themselves with vanity light whenever they approach it. To avoid having too much light, or too little, it will adjust its outputting light to the light values of the room. If there is a lot of natural/artificial light, then the mirror will know that the user does not need vanity lights to turn on. If there is only some light in the room, then the mirror will turn on two lights, or four lights, depending on the photoresistor value. Lastly, if there is no light, then all six lights will turn on. Users should not have to touch the mirror at all to turn on or off the functionality of the mirror. They can simply approach it, or walk away, and this will be sufficient enough to check their looks quickly before they head out the door.

Appendix A: Slack Message History

This is the very beginning of the light-and-proximity-detection-mirror channel
You created this channel on October 29th.
Our team's project channel
Rose Soriano 10:22 AM
joined #light-and-proximity-detection-mirror.
Rose Soriano 10:22 AM
set the channel description: Our team's project channel
Kazuoki Tokuno 10:22 AM
was added to #light-and-proximity-detection-mirror by , along with
.
Kazuoki Tokuno 6:07 PM
2 files
amazon mirror.pdf
141 kB — PDF
141 kB — Click to view
amazon small components.pdf
229 kB — PDF
229 kB — Click to view
6:07
Purchased amazon items
Kazuoki Tokuno 6:13 PM
Purchased motion sensor
PDF
PIR.pdf
235 kB — PDF
235 kB — Click to view
6:14
All we have left is to pick up stuff from Home Depot
Kazuoki Tokuno 6:22 PM
I purchased the light bulbs because they weren't available for pick up
PDF
Light Bulbs.pdf
843 kB — PDF
843 kB — Click to view
liu 7:54 AM
joined #light-and-proximity-detection-mirror.
liu 8:00 AM
checked
8:01
For this project, the final prototype quality will be an important consideration. Packaging parts using good project boxes so on from day one will be v important.
Rose Soriano 11:35 AM
Meeting today (11/10) from 1:00-3:00 PM
liu 8:42 AM
So far, I only see a list of shopping, but nothing else. I will escalate the follow up if you remain inactive.
Yuncheng Yu 3:52 PM
was added to #light-and-proximity-detection-mirror by
.
Nikhitha Reddy Vempati 4:44 PM
On Tuesday 11/16 we purchased the rest of the parts and have begun work on the hardware aspect of the project, including assembling the frame of the mirror. We have also begun work on testing the PIR sensor and the photoresistor. So far, we are testing the python modules to understand the output and how to integrate it into our final product. We are meeting today 11/18 to continue our work on the programming of the mirror.
Rose Soriano 5:26 PM

Today 11/18, we wired up our sensors to test their functionality and proximity. So far we know that our PIR motion sensor works very well beyond our expectations, which is great. We have found that we successfully can read different values of light in the environment too.
5:31
We are watching wood working videos to understand how we can use a hole saw to drill holes into our wood pieces to fit the light sockets. Shout out to Blaine at Home Depot for giving us tips on what glue to use for our wood and which brackets can help us stabilize our frame.
Rose Soriano 2:02 AM
Us planning our wood working by marking where we will use a hole saw to create entry points for the light sockets:
image.png

Kazuoki Tokuno 8:46 AM
Meeting on Sunday 11/21 to go over design layout and testing out circuits with light bulb and wall socket
Jyh Liu 10:53 AM
joined #light-and-proximity-detection-mirror.
Jyh Liu 10:59 AM
you will need to speed up actions. Last week appeared to be very thin in actions. Flagged
Rose Soriano 7:16 PM
I agree that we have to speed up some actions of assembly but we spent quite a bit of hours at Home Depot and wood working and software programming to understand the modules we are working with. We will do better to explain our actions so that you can see that. We couldn't just order parts off Amazon since you wanted us to find cheaper parts at Home Depot, but we found that the parts at Home Depot were limited in selection AND more expensive. So we had to spend a lot of time comparing and finding the right parts again.

1 reply
15 days agoView thread
liu 3:55 PM
Okay I do know it takes time and effort to look for right parts
3:55
Good luck for u
3:55
Let me know if I could be of any assistance
Jyh Liu 3:55 PM
no show on monday
Nikhitha Reddy Vempati 3:55 PM
We are in breakout room 12 since the beginning of class.
Rose Soriano 3:56 PM
^ @Jyh Liu
Jyh Liu 3:56 PM
oh okay
3:58
good to hear your progress on zoom
Kazuoki Tokuno 11:59 PM
11/30 Tuesday team meetup at 4:00pm. Measured and marked wood pieces for 6 lightbulb sockets, 16 bracket screws (addition suggested by Home Depot staff to increase stability), and 12 mirror place holders. Drilled the 6 holes (1.5 inch diameter) into our 2 long (30 inch) pieces of wood using a power drill with a hole saw attachment and 2 wood clamps. Encountered a battery issue when drilling holes, which required us to swap drill batteries frequently (at least once every hole) in order to maintain power to saw through the wood. We used a 7/64 inch drill bit to create screw holes for the remaining 28 marks. Applied the first layer of white paint on our 4 pieces of wood (2 x 30 inch, 2 x 11.25 inch). Concluded meetup at 8:00pm.
Kazuoki Tokuno 1:55 PM
Applied 2nd coat of white paint at 11:00am.
Rose Soriano 2:24 PM
12/1 Wednesday the team met up from 12:00-5:00pm. Finished painting a second layer of coat onto the wood, and drilled one more forgotten mirror clamp hole. Then glued together the wood pieces and tested that the holes for the metal brackets lined up. Also recalibrated the light sensor program, and see that it does output values (which before it would only output a value of 0). We still are to run tests on which light gives different readings for

amount of light. Also put together circuit to test one lightbulb, and so we measured out the dimsensions for all the wires we would need to cut. Had to strip the wires and bend them into right angles + twist the ends which was harder than we anticipated. We realized that our 12 gauge wire is very thick and did not fit into our 2 prong plug all the way (since the plug casing would not close entirely which can be a safety hazard). Decided to consider using a 14 gauge, thinner wire, and ask the TAs tomorrow (Thursday) about safety concerns with our components/circuit
Rose Soriano 2:30 PM
12/2 Thursday: The team met up at 12:00, and plans to continue until 4:30, in the EABA lab to talk to the TA about our safety concerns with using high power from the outlet. He told us that we need to be very careful and make sure to possibly use an extension cord to turn off power immediately if anything goes wrong. He told us to make sure to see all the voltage and current capabilities of each of our component, so that we can make sure that the high power of an outlet does not burn out any of our components or lightbulbs. We drew out diagrams more carefully of our electrical circuits, and saw that since our 15A outlet would be split up into 6 parallel lines, each with a relay that could only take 10A, that we should be fine since current is divided once it splits from our terminal block (which can take 25 A). We decided that most of our components still should work since have high voltage and amp ratings. However, we did decide to go buy thinner wire (14 gauge) so that it would fit correctly into our 2-prong plug, and be easier to deal with. We saw that 14 gauge wire is meant for lighting projects anyway, and for 15A, and so we think this is the right direction to take. We definitely understand different gauges of wire now.
Nikhitha Reddy Vempati 6:05 PM
12/2 Thursday: The team meeting was successful. After our conversation with the TA, we adjusted our initial thought process and changed our wires. We also purchased an extension cord to add an additional layer of safety to the project. We safely connected the raspberry pi to the 8-relay, and tested the circuitry of a single lightbulb. The lightbulb is connected to both the relay (which acts as a switch) and the terminal block. The relay is controlled by the program coded on our raspberry pi, and when the program is run, the light bulb is turned on/off systematically as seen in the video demonstration below. Additionally, we have fixed up our wood frame to stabilize it with wood glue and brackets to provide support to the frame.
6:06
QuickTime Movie
RPReplay_Final1638489711.mov
QuickTime Movie from Google Drive
Click to open in Google Drive
6:08
Our next meeting will be Sunday, 12/5 where we complete wiring the rest of the light bulbs. Since we have already tested out PIR sensor and photoresistor successfully, we will have to modify our test program behaviorally to function as the final project is intended.
liu 8:04 AM
If you can, you may want to have some reserve time for this "final prototype" to make sure you do not run into last minute crisis. So far, I only saw two pieces of wood and then this quick flash of lights posted Thursday
8:07
I just saw your power outlet discussion. Please let me know if you need more help.
Nikhitha Reddy Vempati 11:38 AM
Our mirror-woodframe is fully drilled and painted and attached as of 12/1, we only have to attach the lightbulbs into the frame. Once our circuit is fully functional we only have to attach the lightbulbs and circuitry behind the prototype!
liu 2:36 PM
your demo clips?
Kazuoki Tokuno 8:40 PM
Redrilled back to fix alignment issues. Mounted the mirror. Waiting to mount wire system
8:40
4 files

8:41
We will have a running demo tomorrow
liu 7:47 AM
Okay. adequate progress - no need to meet today. But do post your demo once it is available.
Kazuoki Tokuno 1:32 PM
Testing our second bulb which demonstrates the daisy chaining circuit and mounting our components. Our next step is to finish wiring the rest of the bulbs, setting up our sensors, and behavioral coding
Image from iOS
liu 1:44 PM
watch for safety. You are dealing with 120v AC lines
Kazuoki Tokuno 2:17 PM
Yes sir. We went over safety procedures with Mr. Yuncheng
Kazuoki Tokuno 8:31 PM
Circuit system functions. Testing the motion sensor and photoresistor to build behavioral code.
Image from iOS
Nikhitha Reddy Vempati 2:12 AM
12/6/21: We developed a functional circuit system with 6 lightbulbs connected through daisy chaining (using a terminal block) to the relay and powered by the raspberry pi. Through behavioral programming we are able to control ON/OFF of lightbulbs based on certain conditions of the PIR sensor and photoresistor specified in the code. Next, we soldered the photoresistor and PIR motion sensor to a PCB and placed them through a hole in the bottom of the mirror so they are able to detect motion and light to use as conditions for turning the lights ON/OFF. We also completed the basic programming of the photoresistor and PIR sensor that detect motion and amount of light in the room to turn either all bulbs(if no light in the room), 2 lights (medium light in the room), 4 lights (low light in the room), or zero bulbs (high light in the room). The photoresistor is triggered by the condition that there is motion detected near the mirror. We will further discuss the detailed steps that went into programming both sensors. Additionally, we filled any gaps in the frame with wood filler that we will paint over to ensure a clean and finished look. For our next meeting on 12/7, we intend to finish up documentation, clean up our code for the final demo, and ensure that the back of the mirror is organized. We have linked our unofficial demo for our progress below:
2:13
File from iOS (2)
2:13
File from iOS
Rose Soriano 9:16 AM
12/6/21: The programming part of what we completed over the course of today involved:

1. First making sure that our light bulbs were all easily programmable (shown in our 8:31PM video).
2. The second part was to examine the output of the MCP3008 chip that was the digital conversion of the analog value of the photoresistor. This was our initial problem as we did not see an equal distribution of light balance to the room to the digital number it gave us, but we were able to succesfully create different ranges of light values by testing a bunch of different light scenarios (a lot of lights on in the room vs some lights on in the room vs no lights on in the room). Today (12/7) I would like to test what light value it will give us in a room full of natural sunlight.
3. The third component was our motion sensor and evaluating what was the return value of the GPIO library's implementation of the motion sensor. We saw that pir_wait_for_motion() returned True, but was more of an interrupt style statement and so it did not return False when not activated. After some research (since the official API did not easily list what functions were available), we found that "pir.value" was a better function that we could use which would return a 0 or 1 depending on if it was activated or not. This was great for us to use in our if/else/while statements.
4. After putting all of our components together, we used a lot of print() statements in our overall program so that we could see the direction our code was following as we tested different scenarios. Initially, our

program was simple so that it changed how many lights bulbs were on as a person was standing in front of it (due to nested if/else within a continuous while loop). However, we wanted it to only change how many light bulbs were on once, at the beginning as someone stood in front of it. I was able to solve this by adding a boolean variable that was carefully configured to keep track of if motion detection was detected and continuous until someone left the mirror.

New
Kazuoki Tokuno 1:32 AM
12/7-8/21: Added functions in our code to define different states of the light levels. Cleaned up wiring system by: rearranging and trimming wires between the terminal block, relay, and light sockets, and replacing 12 winged wire connectors by soldering each wire connection and encasing the soldered wires with heat shrink tubing. We then nailed plywood to the sides of our wooden frame to create a well surrounding our wire system. We mounted a rectangular plywood cover over the well by using 2 hinges and added a door handle to open the case easily. We opened a hole using a circular hole saw so the wires from the Raspberry Pi and the wires from our terminal block powering the lights could be pulled through.
1:33
Image from iOS

1:33
Image from iOS

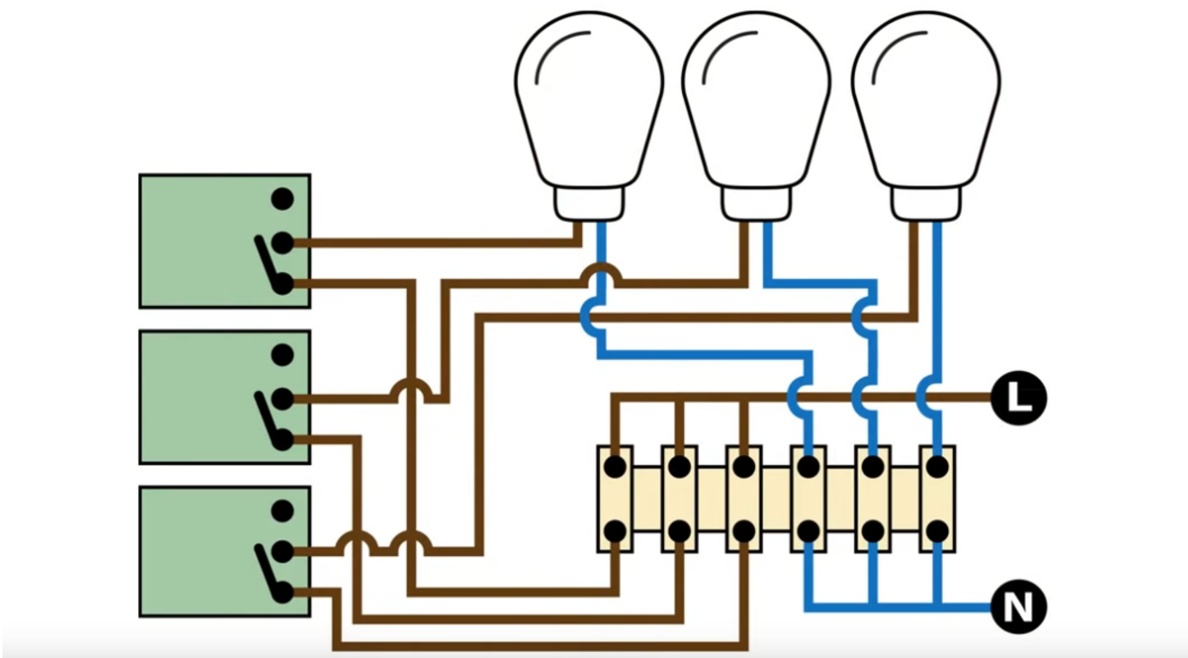
Appendix B: Resources

Sensors:
https://tutorials-raspberrypi.com/photoresistor-brightness-light-sensor-with-raspberry-pi/
https://projects.raspberrypi.org/en/projects/physical-computing/11
https://www.uugear.com/portfolio/using-light-sensor-module-with-raspberry-pi/

Vanity mirror:
https://www.youtube.com/watch?v=vk6WW9FTDE4&ab_channel=LandtoHouse
https://www.youtube.com/watch?v=5Z_WOrhwADE&ab_channel=YuraMenshikoff

Relay:
https://www.youtube.com/watch?v=bOGltcgiXiU&ab_channel=ExplainingComputers

https://www.youtube.com/watch?v=F-yk4Tyc44g&ab_channel=MakerTutor



RaspberryPi and lightbulbs:
https://www.youtube.com/watch?v=bOGltcgiXiU&ab_channel=ExplainingComputers
https://www.youtube.com/watch?v=bOGltcgiXiU&ab_channel=ExplainingComputers