Michael Percival, Aidan Cioppa

I. Documentation

1. Test Scripts and Testing Plan (repository)

a. Current Hardware Testing Verification

Ran simple test code to verify proper function of the sensors/arduino as well as check basic outputs (realistic values for lux from light sensor)(0-5k)

For the LED strip, confirmed how to address individual LEDs as well as change color, turn on off, basic wiring

For the light sensor, confirmed wiring, confirmed how to read values out of the sensor, took notes of realistic lux values that could be read.

For the motion sensor, confirmed wiring, played with range, output duration, discovered lockout period.

Check viability of using light sensor as switch/button/toggle instead of motion sensor.

Link hardware together on arduino and confirm they can function simultaneously and communicate with each other

Confirm full LED strip can be powered via onboard arduino 5v port if dedicated voltage source is plugged into arduino

b. Future Hardware Testing Plan

Experiment with viability of future additions as detailed in presentation. Testing would likely include:

Color toggling alongside on/off toggling (2 PIR sensors)

Exactly how many color options can be added without overloading memory

How to allow for real time color adjustment. Potentially could have 5 saved color presets that are adjustable via separate I/O interface

c. Current Software Testing Plan

Software testing has concluded for the documentation outlined implementation. Located under WorkingImplementations/Base.ino in the repository. This code is a fully functioning implementation.

Additionally began testing out a color toggle implementation where the strip is always on, located in its own folder within the prototypes folder

d. Future Software Testing Plan

Same as future hardware testing plan.
Create branching prototypes for additional future features as development is done.

2. Photos of Prototype

a. Prototype Photos & Explanations

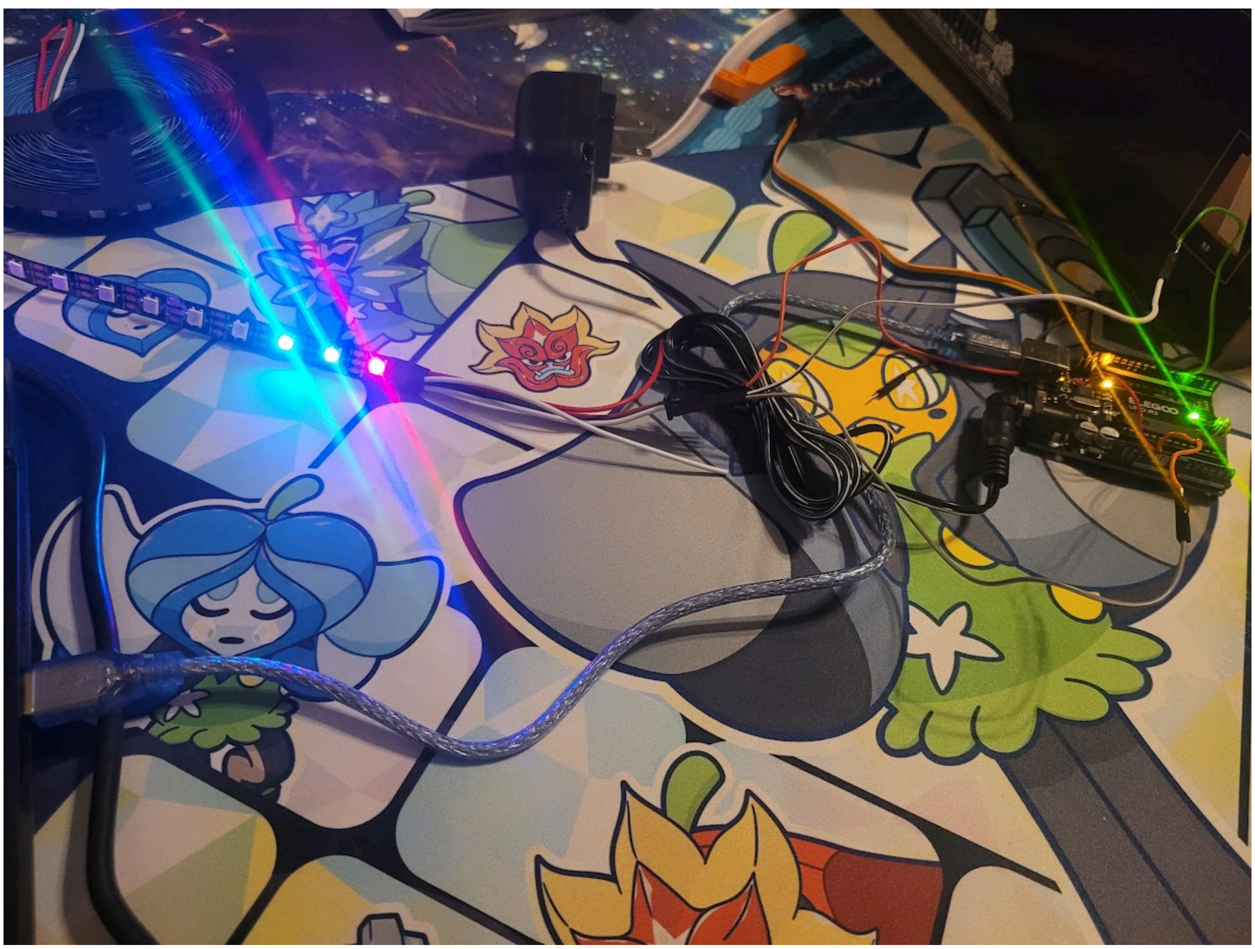

Fig. 1: *LED strip plugged into Arduino to check basic functionality*

This photo is of the Led Strip being plugged into the Arduino to check basic functionality of the FastLED library. The LED was programmed to blink red on the first LED, blue on the second, and green on the third. The DC source adapter can be seen in the photo as well, although it was not used at the time due to the goal being checking the FastLED library, so onboard power was sufficient to power the few LEDs. The code can be found in the repository under Basic_LED_Test_Code.

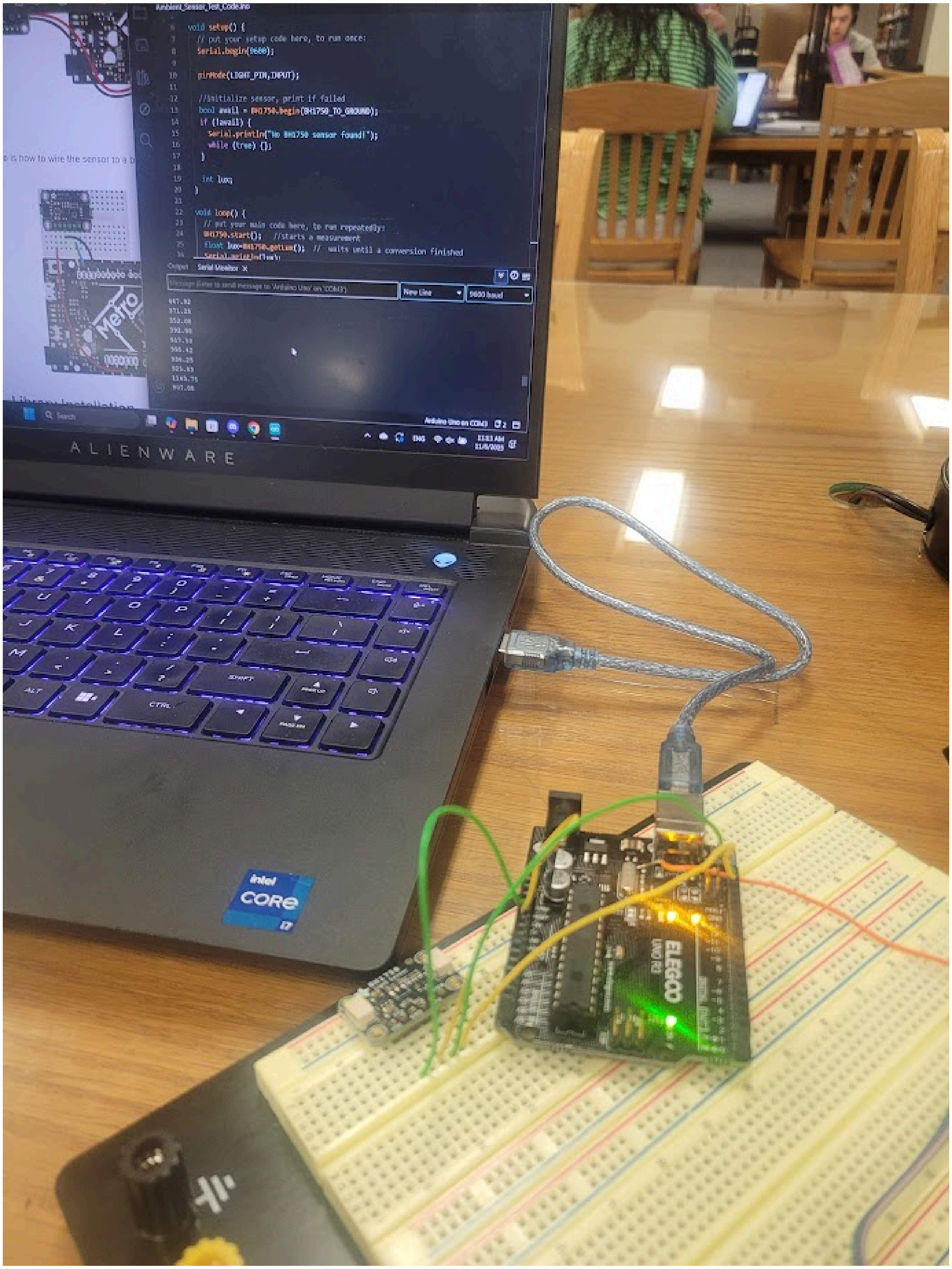

Fig. 2: *Basic testing of the light sensor*

This photo shows basic testing of the BH1750 light sensor. Due to not having the pins soldered onto the sensor, I had to play with the sensor until it gave readings (further discussed in setbacks and solutions). The code used can be found in the repository under Ambient_Sensor_Test_Code

3. Progress and Observations

a. Progress Overview

***Progress has been consistent and a functioning product is expected once the suitable connectors for the light sensor arrive (set for 11/15), issues have been encountered and solutions explored for the light sensor and motion sensor, and an understanding of each of the sensors was quickly developed. Integration into one cohesive system is expected to smoothly progress. Basic test code exists for all sensors and prototype systems have been developed and iterated upon to create a semi polished current iteration. (prototype 3)

The development of the base model has concluded, with explorations into possible future features beginning. Such explorations are located within the repository if interested

b. Setbacks and Solutions

There have been three setbacks, one substantial and two less so.

The minor setback was the Light Sensors not coming with the pins pre-soldered, this made initially testing and reading from the sensors particularly difficult, especially since the output of this sensor is the most difficult to predict. I was able to gather readings from the output but have not been able to exhaustively explore the sensor as a result. In order to solve this issue, stemma qt connectors have been ordered, and will be used to connect the sensors to the arduino in the future. This allows us to not have to solder anything onto the sensors, and is a more secure connection.

The other small setback was the discovery that the motion sensor has a small lockout period, as well as inconsistent readings. The lockout period prevents any readings for about 2-5 seconds after a reading ends. This makes it so you cannot quickly switch from high to low back to high. This is a hardware constraint of the specific sensor and likely unfixable. The sensor also sporadically reads high in its first minute of being initialized. This is extremely annoying to program around and would cause the power to randomly switch between on and off, combined with the very wide range and sensitive nature of the sensor, it would need to be very carefully placed to minimize unwanted readings.

The most substantial setback was the discovery that the code overloads the memory of the arduino UNO. The code was optimized to run within the memory constraints, but was only able to run with the removal of all semaphores and a very limited number of LEDs of the strip being used. The LEDs in the strip being individually addressable takes up a substantial amount of memory, and if we wished to add more color options or expand functionality later, it was clear the available memory of the UNO would not be sufficient. Due to this, we have opted to run the project on an arduino R4 minima, a board with extremely similar functionality to the UNO and with four times as much memory (8KB SRAM). Adjusting the code to this change required minimal updates, there are a few libraries that need to be swapped out, and the scheduler must be prompted to run after setup on the minima.

II. DEMO///

1. Hardware Component

a. Understanding and Explanation of Components

LED strip- has a power line, ground line, and data line. Data line provides all directions for controlling LEDs. LED strip was used as it is a customizable and flexible light source that can easily fit a variety of locations

Motion sensor- has power line, ground line, and data line. Data line provides a 1 or 0 based on whether the motion sensor is detecting motion or not. This sensor additionally has a 2-5 second lockout period where it will stay low after a reading ends, this is very difficult to work around and is causing us to look for an alternative or deal with. Additionally has very wide range and high sensitivity. If working ideally, is a strong option for a basic motion toggle, where every time it detects motion, its output can be interpreted as a signal to toggle between two states.

Light sensor- has power line, ground line, address line, and clock line. Address line provides a value in either raw or lux values. Clock line provides a timer for the sensor to collect data in sync with. This sensor is ideal for detecting changes in ambient brightness allowing it to precisely signal to the LED controller task how bright to make the lights

b. Basic Functionality

(Demo’d during lab)

c. Application to Final Design

LED strip - reacts to the input value of the motion sensor. Initialized to off, when motion is detected from the motion sensor, the LED strip will turn on. The LED strip will remain on until motion is once again detected from the motion sensor in which case the LED strip will turn back off. These LEDs will also vary in brightness depending on the value received from the light sensor.

Motion Sensor - will be the input for turning the LEDs on and off. The user can wave their hand or some object over the motion sensor in order to turn the LEDs on/off.

*We have decided to stick with the motion sensor instead of switching to alternative sensors to fulfill the same function. This decision was made based on memory usage of the sensor as well as the acceptability of a lockout duration due to it solving the issue of accidental repeat inputs, as well as it being very unlikely someone would wish to toggle on off with a high frequency.

Light Sensor - will receive a value of LUX from the light in the room in order to change the brightness of the LEDs. A higher LUX value will make the LEDs brighter, while a lower LUX value will make the LEDs dimmer.

2. Software Component

a. Codebase Storage

https://github.com/AidanCioppa04/smart-LED-project
All project elements are being kept and updated within github, organization is still being finalized so folders may appear later but at the moment all files are just being uploaded
Code contains relevant comments when necessary, variables and function names are sufficient in cases where there is no commenting.

b. Program Flow

The code initializes relevant sensors and objects before creating three tasks which will control the program, there is a motion sensor task that periodically reads from the sensor and signals to the program to turn the lights on or off, a light sensor task that periodically reads from a sensor and signals to the program to increase or decrease brightness, and a LED controller task that responds to the signals sent by the other tasks and controls the on/off/brightness of the LED strip.

c. Testing Code and Demo Scripts

Test code for each sensor is clearly labeled within the repository, verifying functionality for each sensor used as well as the LED strip. Each element is individually tested to isolate issues.

d. Progress on Final Codebase

Prototype 6 was adapted into Base.ino which is the base implementation of the project as described in the documentation. This implementation will provide a working project as long as appropriate hardware is used. The code is commented with additional details and instruction on how to adapt code if you would like to tweak things for aesthetics purposes (color, brightness range, brightness logic)