# Laboratory # 2 ECE:4880,

# Principles of ECE/CSE Design Fall 2021

# Team CASH

Cole Brooks, Hongyu Zeng, Ann Thomas, & Steve Wasiswa

## Design Documentation:

This portion of the lab report will describe the hardware and software design of the system. It will include relevant visualizations and refer to pictures of our final system in an attempt to help readers better understand our finished design before we begin to explain the design process and engineering decisions made in getting to this final design.

The challenge posed in this lab was straightforward at its face: design an infrared light receiver for a transmitter that already existed. The idea was to simulate a realistic situation that many of us will face in the future. Restrictions such as being unable to change the design of the transmitter are very common in industry, and we need to be ready for that. Some of the major sub-challenges for this lab included dealing with noise, and signal amplification. All these challenges will be discussed in detail in the design process and experimentation portion of this lab report.

Our goal in solving these problems was to meet all the customers’ requirements as well as simplify the design of the receiver as much as possible. We wanted to minimize the hardware component of this lab as much as possible in order to have a sleek, and portable design. We will discuss the hardware and the software components of our solution in the following subsections.

Hardware:

The hardware portion of this lab was intentionally basic (the reasons for which will be described in later sections of this report). Our final solution contains only 4 components: an Arduino Nano 33 IoT (Figure 1.1), a modified infrared obstacle avoidance module found on amazon (Figure 1.2), an LED for use as an alarm light (Figure 1.3), and a magnification apparatus designed using a lens from another obstacle detection component and a 3D printed part used to keep our photoresistor from the obstacle avoidance module lined up with that lens (Figure 1.4).

Schematics for the construction of our final solution can be found in Figure 1.5 in the appendix.

Each of these components will be discussed in more detail in the design process and experimentation portion of this lab report.

Software:

The software portion of our solution can be broken into two parts, the more complicated of which runs on the Arduino. This portion of the software is responsible for receiving the signal from the sensor module, filtering the noise sensed by that sensor module, and alerting the user when the transmitter was no longer detected by the receiver. This portion of the software is entirely written is C++.

The second portion of the software was responsible for sending text messages to users when the receiver did not detect the transmitter.

Both of these portions will be discussed at length in the design process and experimentation portion of this lab report.

## Design Process and Experimentation:

This portion of the report will describe the process by which we arrived at the final design for the Lab. We will compare multiple design choices, evaluate them critically for cost, performance, and complexity, and we will cover the process by which we determined which solution we would choose to go with.

This portion, like in the section before it, will be split up into a section for the hardware and another for the software.

Hardware:

As briefly mentioned in the Design Documentation section of this report, it was a goal of ours to keep the hardware portion of this lab intentionally minimalistic. We did this for a few different reasons.

First of all, all members of our group are CSE students. Although the first lab in this class bolstered our confidence in our hardware skills, we are still much more comfortable and confident with the software we can develop than the hardware we can design.

Additionally, we knew that our final solution could be made much simpler and cheaper if we kept our hardware as simple as possible. If we are to be good software engineers, we must keep the cost and complexity of the systems we are designing in mind at all times.

Even so, we knew that there was no getting around a certain portion of this lab being handled by hardware. We would need something to receive the IR light, as an Arduino cannot do that on its own. We also needed an LED to be attached to the system due to a requirement from the customer.

The first problem we needed to solve was to find a receiver. For this we attempted to use various different components sourced from the electronics shop as well as from Amazon. The first day that the lab was assigned we ordered roughly 5 different IR receivers from Amazon so that we could experimentally figure out which one would work the best. We brought all of these components into the lab as we received them and tried to get them to receive anything from the transmitter. It was interesting seeing which ones worked with the transmitter and which didn’t. In the end, we found that the B07W97H2WS Obstacle Avoidance Module (figure 2.1) performed the best for our solution after some slight modifications. As the name suggests, this component is designed for use in obstacle avoidance. The clear LED sends IR light forward, and the darker LED on the module senses the light. When the darker LED receives a signal, it sets the data line low and turns a red LED on the board on to signal that an obstacle has been detected. Due to both LEDs being pointed in the same direction, only when there is an obstacle in front of the module will the IR light be reflected and picked up.

For us, though, adding additional IR light to the system was a problem. We didn’t need any extra noise to filter out. For this reason we modified the board by cutting off the transmitting LED. This effectively turned our obstacle avoidance detector into a simple IR detector.

The potentiometer onboard the modified obstacle avoidance module was experimentally adjusted in order to maximize distance at which the receiver could detect the transmitter’s signal

The next component was the LED for visibly alerting the user when the signal from the transmitter is not being received (Figure 2.2). This component was quite easy to figure out. We simply used a red LED and a resistor to connect it to the Arduino.

After adjusting the potentiometer onboard the IR receiver module to get the maximum distance, we realized that we could only detect the transmitter from roughly 20 centimeters away. This was a problem, because the prototype receiver we were trying to replace had a working range of over 3 feet. We needed to figure out how to increase the range of our receiver as much as possible to match or exceed the prototype.

Our first approach was to add a simple op amp amplification circuit. Circuit diagrams for which can be found in the appendix as Figure 2.3. We found that while this approach was feasible, it did not align with our goals for this project and we ultimately found a much simpler solution.

The solution we found came from another one of the IR sensors that we had purchased from Amazon. This particular receiver functioned in much the same way that our obstacle avoidance module did. It sent out a red laser from a laser diode, and received the reflected signal with a photoresistor. Additionally this component included a lens that covered the photoresistor in order to increase the effective range of the module. We took this lens and put it over the photodiode in our system and found that it dramatically increased our effective range. Additionally this approach would physically filter out much of the ambient light by focusing our receiver directly in the direction of the transmitter, dramatically decreasing the amount of noise entering our system to begin with.

We quickly found that there was no way to directly attach the lens to the receiver as it was, so we modeled and printed a component to hold the lens and the receiver module in line that held the receiver perfectly in line with the focal point of the lens. Figure 2.4 in the appendix shows the apparatus we manufactured for this system.

Software:

<HONGYU, please describe the process for coming up with the software here>

## Test Report:

This section will include a table that describes all tests conducted to verify the completeness of our solution to this lab.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test** | **User Requirement Address** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Date** | **Correction Action** |
| Length of Temperature Sensor | System has a thermometer sensor at the end of a 2.0 meter cable | Sensor that satisfies customer criteria is connected to the system | As expected | Pass | 9/3 | None |
| Physical  Robustness  Of  Temperature  Sensor | Temperature sensor should be of ‘nice mechanical construction’ (physically robust)   * Capable of bouncing   Sensor is not damaged when placed in ice water | Sensor can be dropped onto a hard surface or placed into ice water with no noticeable decreases in performance or accuracy. | As expected | Pass | 9/3 | None |
| Arduino System Physical Requirements | Box contains a display, a button, a battery, and a power switch. | System has all components required by customer | As expected | Pass | 9/10 | None |
| Enclosure Requirement | Arduino system is enclosed in some way | Arduino system has an enclosure to protect it from physical trauma. | As expected | Pass | 9/5 | None |
| Arduino System is physically robust | When dropped to the floor with cables connected, the connectors or cables should not break. | Arduino system should continue to operate as expected after being dropped from a typical height desk | As expected | Pass | 9//10 | None |
| Temperature Sensor Interchangeability | If the sensor has been unplugged and is then plugged in, the system should resume normal operation without user intervention. | Arduino System should notify user when temperature sensor becomes unplugged, and resume normal operation after the temperature sensor is plugged in again. | System requires users to close an alert before resuming normal operation | Fail | 9/25 | JavaScript implemented to automatically close the alert screen |
| Display Delay | The correct temperature should appear on the display when the button is pressed with no noticeable delay ( < 20 ms ) | Arduino system should display correct temperature within 20 ms of user pressing the red push button | As expected ( time to display was so short it was unmeasureable ( < 1 ms ) | Pass | 9/20 | None |
| Display Readability | Display should be clearly readable under normal indoor lighting conditions for all temperatures within normal range of the device | Display should be readable under a variety of light sources (Indoor with all lights on, indoor with all lights off, outdoors during the day, outdoors at night) | As expected | Pass | 9/25 | None |
| Pushbutton Operation Type | The pushbutton operates in a ‘momentary contact’ way | Pushbutton should turn display on while pressed, and display should turn off when the pushbutton is released | As expected | Pass | 9/3 | None |
| Display Notification for unplugged sensor | Display notifies user when temperature sensor is unplugged or damaged | User should see clearly on the display that the temperature sensor has become unplugged or damaged | As expected | Pass | 9/10 | None |

## Project Retrospective:

This section of the report will include a summary of the outcome of our project. This will include a critical assessment of the effectiveness of the choices described in the ‘Design Process and Experimentation’ section. This section will also include a discussion of factors that contributed to less-than-complete success in meeting our goals and an explanation of how we will change these factors in the future. We will also summarize the role of each team member in this lab, and the workload distribution between team members. We will discuss project management processes that were implemented and outline the timeline for this project.

Overall, we would call this lab a success because we met all of our goals. Our solution performed much better than the prototype, and we successfully minimized the hardware portion required for our system. We feel as though this lab has taught us a lot about filtering noise from a system, and about how modulated light reception works in general.

Although we would call this lab a success, we did encounter a few failures that we will describe in the next few paragraphs. Additionally, we will discuss why these failures occurred, and how we will prevent these failures from occurring in the following labs.

The first point of failure was from a timeline perspective. While we did a good job of getting the lab started early, due to members being busy with midterms and other classes we fell off in terms of productivity at some point in the middle of this lab. We found ourselves pushing hard the last few days before our check off in order to get our solution to a point where we could be happy with it.

Although this is in some ways the nature of college, in the future we would like to address this problem so as to avoid it as much as possible. For the next lab, we will be sure to discuss the timeline early on, and not assume that each members productivity will remain the same throughout the entire lab. We will discuss and estimate the amount of time each member can contribute to the lab on a week by week basis so that we can plan better and better utilize our time.

Another point of failure was in communication. At times in this lab it felt like we were not all on the same page, and this was predominantly due to our means of communication. Although we did a good job of posting updates as much as possible, we weren’t consistent with our use of discord, and texting. This lead to an inability to efficiently refer back to previous conversations because we didn’t know whether the conversation had happened via discord or through text. In the future, we will discuss rules for each method of communication. For example, we may decide that discord is the primary means of communication, and texting/calling is only to be used for emergencies.

Although there were failures in this lab, it was blatantly apparent that our team has grown in terms of communication, efficiency, and otherwise general chemistry from lab 1. The workload felt much more balanced, and communication was much more efficient than in lab 1.

Cole’s first role in this lab was to determine experimentally which components would be used. He took point on ordering all of the components and researching methods for amplification of the signal from the transmitter. Cole also figured out how to connect all of the components together so that the hardware could be passed along as a functioning set to the next person to maintain velocity on this project.

Additionally, after determining that an additional hardware component (the lens and receiver apparatus) was needed for this project, Cole was the person who modelled and printed the 3D component required for this lab. It took many iterations to get this right, and Cole designed each and every one of them.

<Hongyu Contribution to the Lab>

<Steve contribution to the lab>

<Ann contribution the lab>

Project management for this lab followed a much more agile methodology than did our last lab. We created, estimated, and managed workflow using Github’s issues feature. Additionally we did daily standup meetings in the form of daily progress update messages in the discord. We felt like our management style for this lab was a vast improvement on the last lab and lead to better team chemistry, efficiency and performance. After retrospecting with the group we determined that we will take the same approach to the next lab.

Although this lab was outside of our comfort zones, we found that we were more than capable of working through it. It taught us how to approach problems that we are uncomfortable with, and how to work as a team to cover our weaknesses and accentuate our strengths. We feel very confident that what we learned in this lab will contribute to our success in our final Senior Design project.

## Appendix & References:

<Figured referred to above>

<Software state diagrams>

<Hardware Schematics>

<Citations>