Project Phase 2 - Apollo

Parker Tackley

CSE 321: Realtime Embedded Systems

November 11, 2024

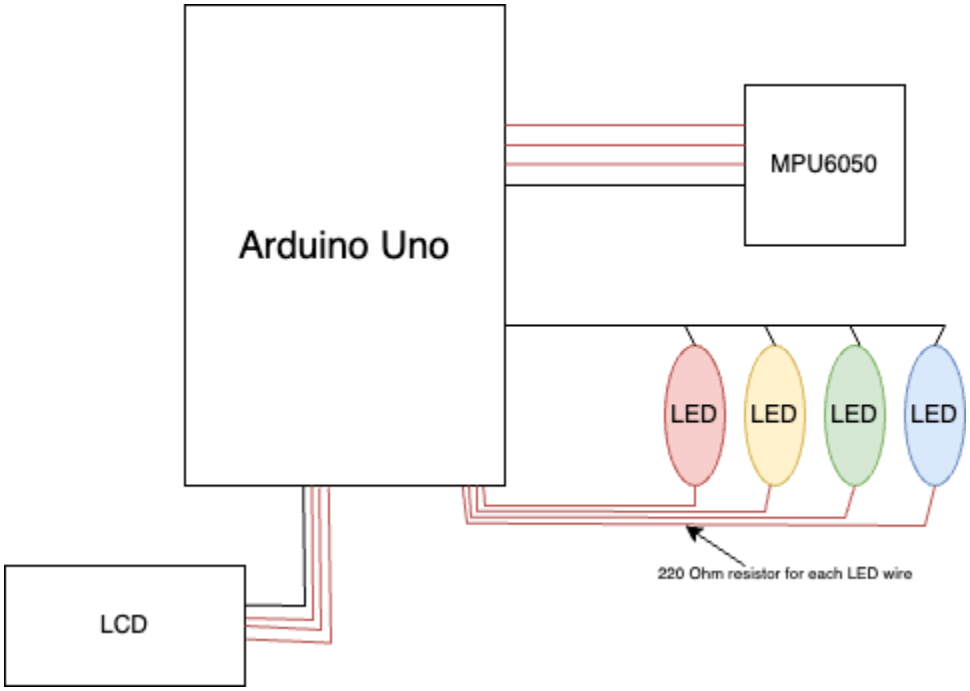
Table of Contents

Hardware Components

Our project consists of an Arduino Uno Rev3 with a few external components which take inputs along with a few outputs. We have our Arduino which is the main component of the system giving us all the necessary computing power, then we have the SHILLEHTEK MPU6050 Accelerometer for measuring the users throwing power in terms of G-force, we then have a few LEDs to display which state the system is currently in, an LCD which displays the inputs along with the highest score, then we also have a speaker which will play sounds based on the measured input.

- ➢ Arduino Uno Rev3
- ➢ SHILLEHTEK MPU6050
- ➢ 4x LED
- ➢ IRC2 LCD screen
- ➢ Speaker

Prototype details



Software Components

Project Design

Currently, our code base is split into three main parts. We have an input section, output section, and a main section.

Our input section handles all of the user inputs and adds the necessary calculations to convert the G-Force into a readable and digestible format. This includes functions which read the inputs from the accelerometer from the X and Y axis based on outside forces, along with a function which converts the reading

Our output section handles the output of the score in regards to the LCD and eventually the speaker. This holds functions related to outputting text onto the display when necessary along with playing sounds through the speaker when a high score has been reached.

Our final section, main, handles all of the basic states that our system will have to navigate through including the play state, the ready state and the celebrate state for a high score. This includes a simple state machine which changes based on the user's input.

Test Scripts and Patterns

In order to test all the components of our system, we have set up a series of tests for each individual part, along with tests to assure that two communicating components work together. We have some simple human interaction tests, along with a few tests which use a computer to ensure correctness.

Current Hardware Tests:

MPU6050 Accelerometer Test:

We connect the MPU6050 to the Arduino and use the dedicated library functions to assure that the correct readings are appearing onscreen by moving the MPU in the correct direction as listed on the physical part.

LED Test:

We connect a few LEDs with 220 Ohm resistors to ensure that they all light up correctly

MPU and LED Test:

Combining the other two tests, we light up certain lights when the MPU6050 is moved in a specific direction.

LCD Test:
Ensure the LCD is displaying text.

Speaker Test:
Play a short sound through the speaker to make sure it is functioning correctly

Future Hardware Tests:

Score Display:
Ensure the LCD displays the user score, along with a message if they've beaten the previous high score or not

Speaker Test:
Ensure the speaker emits sound depending on the state.

Button Test:
Ensure that input is being received when pressing a button.

Current Software Tests:

Timed LEDs:
Using a simple timed state machine to test the readiness of the system at any given time, this is shown by a LED status

Accelerometer High Score:
Uses LEDs to show whether a "high score" was achieved or not

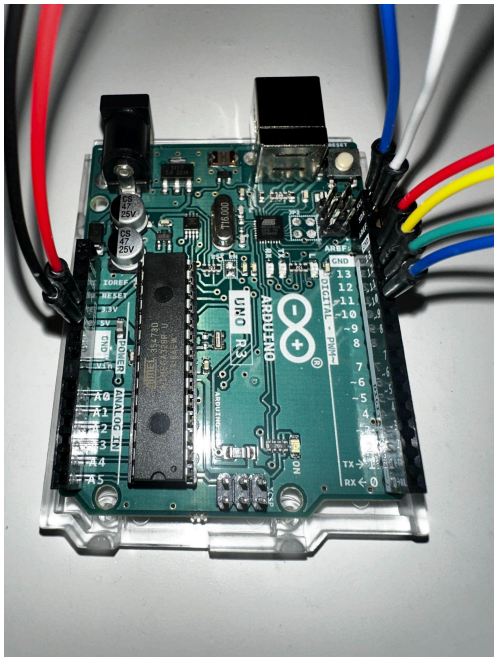
LCD Scores:
Displaying the score and the time left on the LCD to show a countdown of the current state.

Future Software Tests:

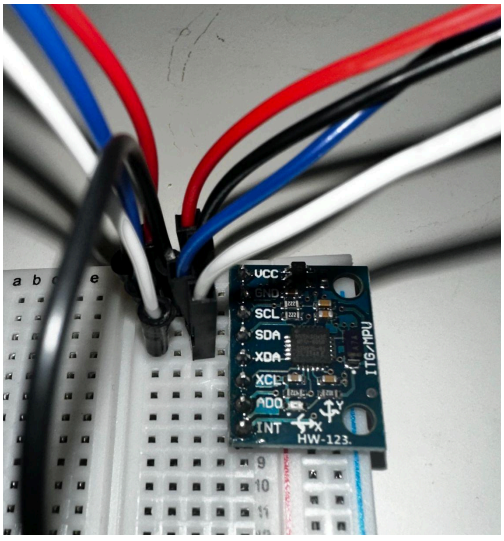
Accelerometer Test:
Make sure the accelerometer is measuring in multiple dimensions to get a better measurement of force.

State Buttons:
Allow for the user to change states by pressing buttons.

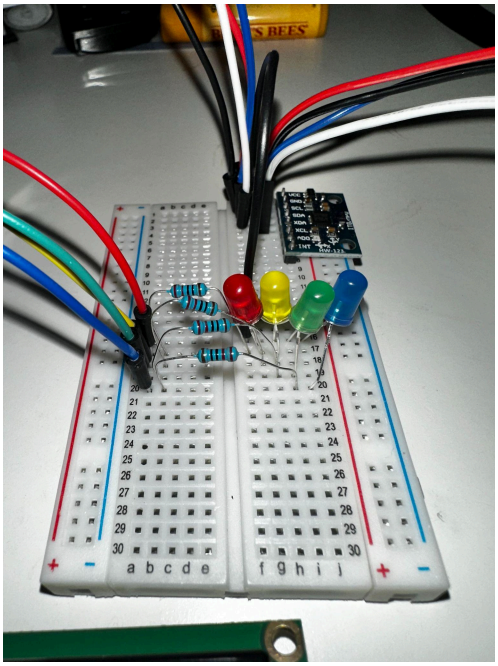
Images



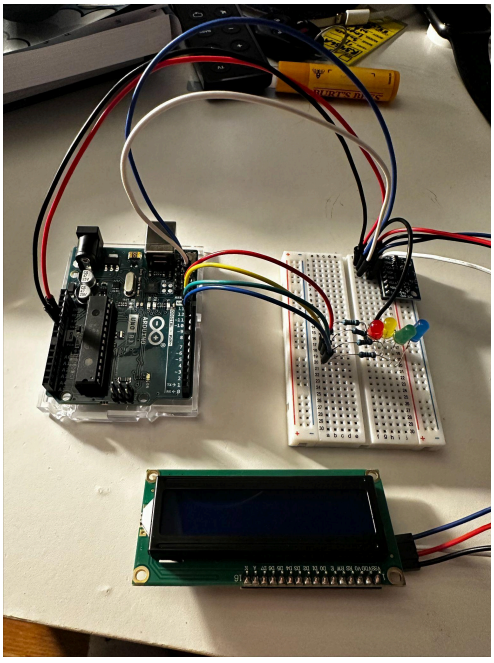
The Arduino with all of the wires connected. On the left, we have the ground wire for all the components along with the 5V output for the accelerometer and the LCD. On the right, the top blue wire is for the SCL/serial clock line and the white one is for the SDA/serial data pin. The bottom red, yellow, green, and blue wires are connected to the 13-9 ports on the arduino for the LEDs given by the color of the wires.



Here, we can see the accelerometer with the red 5V wire to VCC for operating voltage, the blue SCL wire, white SDA wire and the black ground wire.



Here we can see each LED with the corresponding colored wire, where each LED uses a 220 Ohm resistor



An high view of everything connected so far, along with the LCD with the red wire going to VCC, blue to SDL, white to SDA and black to ground

Observations and Notes

Current Challenges

Currently, I'm not facing too many challenges in the physical aspect of the system. Mostly everything is set up and working. Now the biggest challenges that I have are determining how the game, or states, will be chosen. At the moment I just have it set up on a simple timed cycle where there's a play state, a celebration state, and a hold state. I'm wondering if I should implement human interaction to determine these states like a button. My second challenge is how to record and properly measure the force of the player. I'm currently just using the Y-axis force which requires the user to manipulate the system in one dimension. I would like to create a way to make this more accurate. In order to work through these challenges and previous ones we faced, reading the part documentation along with documentation for the various libraries that are being used has helped tremendously with the progress.

What we have working

We have most of the physical parts working in unison with each other. The LCD works, LEDs work, Accelerometer and speakers all work together. We can do pretty simple calculations to measure a rough estimate of the force given through user input. I saw rough because as above, this really only works in respect to one dimension. There's also a very rough state system in place that allows for repeated user input which acts like a testing ground for the components.

What we want to get working

As mentioned above, I want to get a much better state machine set up and a better reading of the forces being measured. Rather than just measuring one dimension, we could take into account multiple dimensions and get the speed along a line. Though this could take some linear algebra. I also want to implement buttons in regards to the state machine. Since I just have everything running on a timeless loop, I want the user to be able to press buttons to choose what they would like to do, either throw or quit the game.

Conclusion

I think I've made some good progress and am at a good track to finish by the end of the semester. Since all of the parts, other than a few buttons, are up and running, it's more about the coding to make everything feel more premium and make sure it all works in unison. I'd like to add many more features, however I don't want to get too ahead of myself since I'm at a pretty good point at the moment.