

Reflection 3

How have you approached the technical challenges in your project so far?

Challenges

Our project can be subdivided into two systems that work together: ultrasonic sensor subsystem, and LED subsystem. Each of these subsystems have had challenges in integrating all of their components.

Ultrasonic Sensor Subsystem

Arduino and Sensor Integration

This subsystem includes the Arduino Uno microcontroller and two arrays of three ultrasound sensors. These are used to track users within a ring about 30 cm in diameter. The Arduino was used since it is easy to program to work with the ultrasound sensors. The main drawback though is the difficulty in debugging code that runs on microcontrollers. Using Arduino IDE one cannot add breakpoints and the only way to see what was happening in the software was to write multiple print statements. Without too much hassle we wrote a program to run on the Arduino that could read information from three sensors without having them interfere with each other.

Once this was completed, the next phase was to write a new program that could work with six sensors, since we would have two arrays of three sensors. I create a copy of the file and added code for the new sensors while at the same time also refactoring; this was risky because I was adding code while also rewriting existing code increasing the probability of creating a bug. Further increasing the chances of something not working we also rewired the sensors into a new breadboard, so that when we tested the code we were getting '0' from every sensor. Thus began the tedious process of debugging.

This could have been avoided or reduced by first only adding code without refactoring, and testing this on the old circuit we knew was properly wired. Then we could have refactored the code to make it more maintainable, followed by testing it. And finally rewiring the circuit and testing it. This way only one change is made at a time, making it much easier to see where the point of failure is.

Fortunately we never deleted or overwrote the program that worked with three sensors and we used that figure out if the circuits were correctly wired. It was not. We had to rewired the entire circuit until it worked with the code for three sensors. We were back at square one.

I could not figure out where in my refactored code for six sensors I had gone wrong so I decided the best approach would be to rewrite it without the refactoring I had done earlier. This worked perfectly with all six sensors. Then as I was refactoring the working code I saw where I misplaced one line of code. I had inadvertently placed a delay in the wrong place causing all sensor readings to be zero. I fixed this and tested the original refactored code and it worked perfectly. Perhaps if we had not rewired the circuit I would have found this and not spent so much time debugging.

Triangulation

We have also had challenges with writing the script that will triangulate the person's location within the Gather-Ring. This task is primarily Jake's task and he has been working on implementing the mathematic functions and parsing the sensor data. This has been a challenge for him particularly because he has little programming experience, through no fault of his own, but instead because we are barely exposed to it in Electrical Engineering. I saw he was struggling and I sat with him and helped him make progress. Coding in pairs is a great way to learn, as the other person you work with might have a completely different approach to a problem than you; it also ensure there are less bugs because the other person might catch any logical mistakes you might have.

LED Subsystem

The LED subsystem is composed of a Raspberry Pi 3 and a NeoPixel LED light strip with 60 LEDs. We decided to use python on the Raspberry Pi to control the LEDs since it is easy to program in, useful since most team members have limited programming experience, and since there is a python library for the NeoPixel. Unfortunately, this library is outdated and supports neither the newer Raspberry Pi 3 nor our version of the NeoPixel. The two team members assigned to work on this were separated from the main team since they would work in an adjacent room where they could plug the Raspberry Pi into a display with a keyboard and mouse. This unfortunately led to a lack of communication between both subsystems of the team and meant that whenever they encountered an issue they would not ask for help.

For about 2 weeks they tried to get the LEDs to light up and be able to control individual lights. They were successful in getting the lights to light up but encountered issues when running any light pattern they found online, or even turning the LEDs off. This turns out was due to the GPIO pins on the Raspberry Pi 3 behaving different than they did on the original Raspberry Pi 1. When I checked up on them last week I learned that they were stuck but had not asked for help. I spent about 1.5 hours with them trying to figure out what was wrong and eventually found that the only fix was to disable the soundcard on the Raspberry Pi. This – for some odd reason – worked.

This shows the importance of communication within a team. Sometimes people within your team can help you, as they may have already run into this problem before or at least know of an approach to tackle it.