EE/CprE/SE 491 WEEKLY REPORT 07

10/25/2024 – 10/31/2024

number: 36

***Project title:* Ultrasonic Object Detector**

Client &/Advisor: Professor Jiming Song

Team Members/Role:
Nathaniel Clarke - Project Software Designer
Brock Dykhuis - Circuit Analysis
Nicholas Jacobs - Electronics
Jonathon Madden - UI Designer & Software Tester

Weekly Summary

This week we looked into how to use the new transducers (the voltage needed and potential placement). We also began looking into which MCU to order for the radar system. The last group used ESP32 D-1 mini, we are looking for a MCU with a similar form factor, and is more efficient if possible. We discussed potentially switching the MCU out for a Raspberry PI if its high clock rate and processing power proves necessary. In terms of software, we took notes on the prior display implementations, and we will likely make a prototype in HTML and PDE to determine which suffers from less delay.

Past week accomplishments

Took notes on the 2023 (sdmay24-24) implementation of the radar display (major variables and functions) - **Nathaniel Clarke**

- Determined the differences of how the displays work (2024 uses a heat map, 2023 shows one point at a time with a line pointing to the new detection). We will need to display multiple points at once, and will need to evaluate the need for a heat map based on the detection accuracy (ideally the streaks will be no more than a cmwide)
- Notes Links: Past Variables and Formulas (2023 project)
- Examples include**:**
    - **myPort**: A serial port connection for data transfer, this transfers data bit by bit, which may prove to be inefficient. Our implementation will attempt to transfer data through WIFI to improve efficiency.
    - **data:** Stores data retrieved from serial port, data is originally in form “distance,angle.” The data variable stores the parsed version without the delimiter (the period).

- **pixsDistance:** distance converted to match pixels of the display. pixsDistance = iDistance*((height-height*0.1666)*0.025); this converts based on the adjusted height with the padding removed, and it is multiplied by 1/400 because the display is tracking up to 40 cm (400 mm).

- Initial research into ESP32 MCUs - **Nathaniel Clarke**
  - Found ESP32-C6-DevKitM-1-N4 on Mouser.com which seems to have a sufficient form factor, and it seems to have sufficient processing power.
  - The Espressif Systems ESP32-S3 Development Kits also seem to viable and have more versatile USB type A ports.
  - The C6 has a primary focus on power use reduction, whereas the S3 emphasizing processing power (while be slightly more expensive). The S3 would likely be better in this application.
  - ESP32-S3-DevKitC-1-N8R8 appears to possibly be the best options due to the 8 MB of PSRAM which can act as a buffer, and allow for the necessary data processing.

- Looking into microcontrollers- **Brock Dykhuis**
  - This can be done over Wi-Fi as both the Raspberry Pi 3B and the ESP32 both have Wi-Fi capabilities.
  - This might not be needed in the final design, as the Raspberry Pi has the ability to be used as a microcontroller.
  - Raspberry Pi does not work as well in “real time” as the ESP32 microcontroller.

- Testing mock displays on processing ide - **Jonathon Madden**
  - Will have to figure out how to get the raspberry pi to communicate with the processing code
- Signal Amplification and Processing **-Nicholas Jacobs**
  - Started adjusting the signal amplification and processing stages to improve radar detection range and accuracy.
- Transducer Testing and Timing **-Nicholas Jacobs**
  - Conducted more testing on the transducers and fine-tuned pulse-echo timing to ensure optimal detection performance.

Individual contributions

NAME	Individual Contributions	Hours this week	HOURS cumulative
Nathaniel Clarke	Took notes on the 2023 implementation of the display, and began looking into a MCU to order.	6	45
Brock Dykhuis	Looked into microcontrollers and how to connect one to the raspberry pi.	5	41
Jonathon Madden	Continued looking over mock displays and worked on figuring out how the raspberry pi will communicate.	6	38

Nicholas Jacobs	Worked on radar detection simulating and adjusting signal amplification, processing, and fine-tuning transducer pulse-echo timing	6	41

Comments and extended discussion

Figure out what microcontroller (ESP32) model we are going to order.

Plans for the upcoming week

Jonathon Madden - Look into how to best send data and receive data from the raspberry pi. Continue going over past groups code to better understand the type of data that will be processed.

Brock Dykhuis - Order a microcontroller and connect it to the raspberry pi.

Nicholas Jacobs - Focus on minimizing signal noise in the amplification circuit to improve radar clarity(Test first, then use actual transducers). Refine the data processing to possibly improve filter and distinguish signals. Continue testing the transducer at various simulated distances to adjust for any delays or distortions, and document each adjustment to keep track of what improves detection.

Nathaniel Clarke - Next week I plan to test creating a display using a PDE sketch, and I will use a text file with randomly generated data as a simulation.

Summary of weekly advisor meeting

During this week's meeting we further discussed the transducers and their specifications (voltage and decibels). We also discussed which microcontroller would be best to use for our project.