Lizeth Lucero and Lyn Ackert-Smith
Professor Vaccari
Embedded Systems
11 December 2019

**Proximetector360**

**Introduction:**

This project was initially inspired by the way in which we perceive the world around us, specifically how our eyes determine the proximity of an object. However, we believed that we could do something more than simply recreating a model of how our eyes use disparity of images to recognize the distance of an object. We wanted to create something meaningful that can potentially be used in the future and we were especially passionate about creating some kind of accessible device using the skills we learned throughout the semester. Therefore, we decided that not only would we detect the distance of an object using proximity sensors, but we would also use that information to communicate back with the user and let them be able to feel how far an object is through the use of vibration motors, which would vibrate weakly or strongly depending on how far away an object was to the sensor that it was paired up with. Using various sensors in a hat and motors directly next to the sensors would allow the user to feel the world around them, which would be useful for people who cannot see. We used (DroneBot) to help us set up our sensors using Arduino and to set the power of our vibration motors we used examples with timers that we worked on in class.

**Methods:**

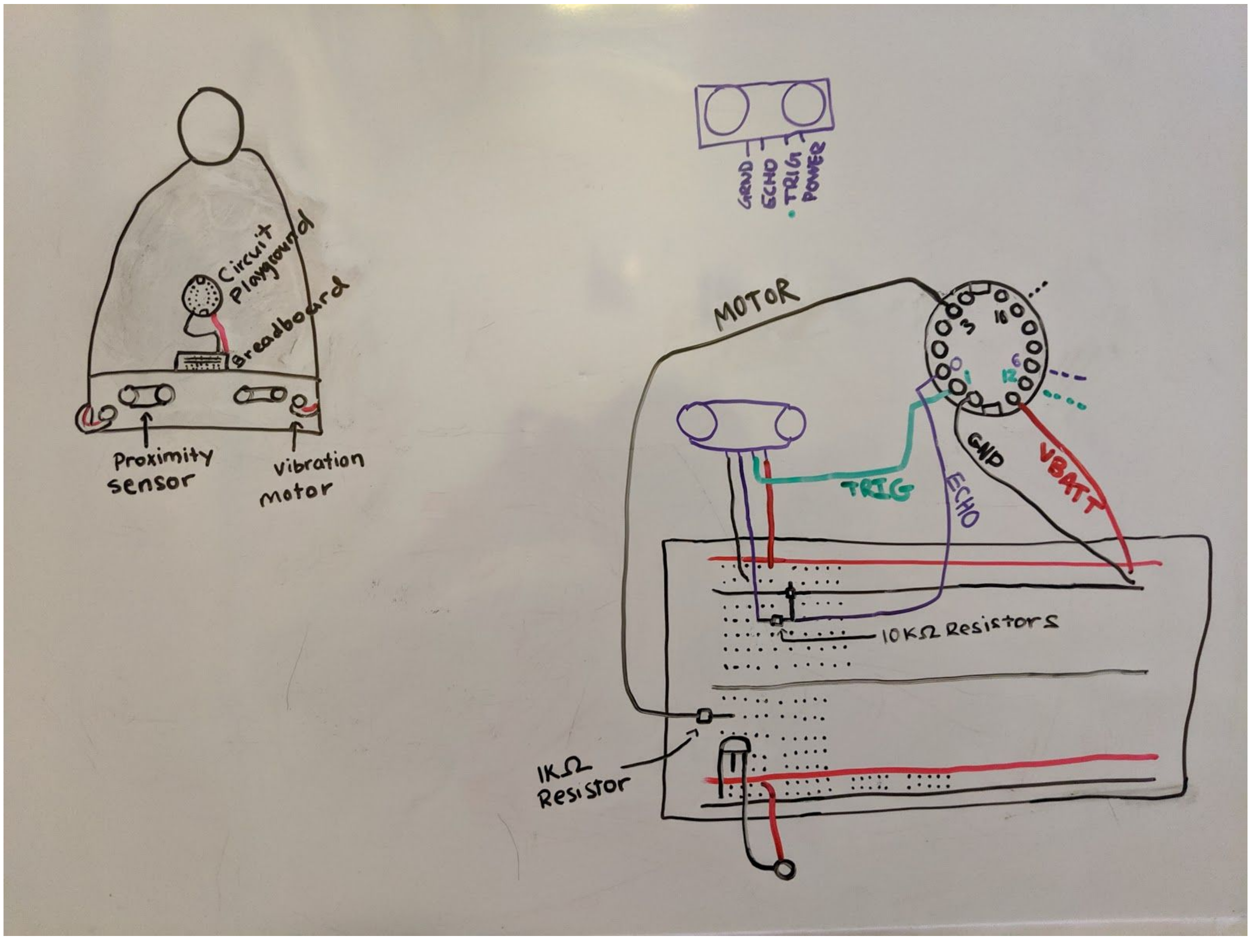
The entire system is split up into two separate subsystems, each one identical to the other and driven by their own adafruit circuit playground board running the same code. To each circuit playground two proximity detectors and two motors are attached. The signals for controlling the trigger of the sensor, reading the proximity, and driving the motor associated with each sensor each using a pad of the circuit playground. Since there were two sensors attached to each circuit playground, this totaled six pads, as well as ground and VBATT.

The readSensor1() and readSensor2() functions were used to read the distance of objects from the proximity sensor 1 and sensor 2 respectively. The sensor works by sending a trigger signal to the proximity sensor trigger pin by writing low for 2 microseconds and then high for 10 microseconds. This initiates the sensor to send a pulse wave and then listen for the time taken for the pulse to bounce back. The digitalWrite() function in the Adafruit library was used to send this signal over pin 0 for what we coined sensor 1 trig and over pin 6 for sensor 2 trig. The time taken for the pulse to return is sent over the echo wire on the sensor, which is attached to pin 1 for sensor 1 and pin 12 for sensor 2. This was read by using the pulseIn() function in the Arduino library, which returned the wavetime from each sensor. The time taken to return was then multiplied by the speed of sound to convert the time into distance that the sensed object was from the sensor, in cm.

This time was then used by dist_motor1() and dist_motor2() to control the vibration motors placed below sensor 1 and sensor 2 respectively, in order to communicate the proximity of objects in terms of the strength of a vibration. The higher the vibration, the closer the object was to the sensor, and thus the user in that particular direction. The dist_motor1() and dist_motor2() used pulse width modulation based on the distance of objects cut off at specific thresholds. The thresholds of distance were 0-50cm, 50-100cm, 100-200cm, 200-300cm, 300-400cm, and 400cm and beyond.

The main loop of the program continuously called the functions sampling() and sampling2() for sensor 1 and sensor 2 respectively, which averaged multiple reads of readSensor1() and readSensor2() in order to help eliminate noise in the data before handing distance read off to dist_motor1() and dist_motor2(). In the case of this code, OVERSMPL determined the number of iterations of averaging and was set to 30.

To interact with the system, the user simply has to wear the headpiece that the circuitry is integrated into. The circuit playground boards should all be wired in such a way that proximity sensors point in different directions, with corresponding haptic motors placed beneath them so that the user may interpret vibrations from the motors with the correct object distance information collected by the sensor above it. The current project only implemented two circuit playground systems for a total of four proximity sensors, but the benefit of having completely independent subsystems is that as many can be paired together as the creator would like to have without fear of interference. The more subsystems placed together on a single headpiece, the higher the resolution of the haptic information, which is preferable for making the system as practical as possible. Future implementations of this project will include a switch to turn the system off and save battery power when not in use, with some way to connect all subsystems together to be able to turn them all on and off from one access point.




**Bill of Materials:**

| Product Name | Product # | Price | Quantity | Total | Link |
|---|---|---|---|---|---|
| HC-SR04 Ultrasonic Sonar Distance Sensor + 2 x 10K resistors | 3942 | $3.95 | 4 | $15.80 | Adafruit |
| Vibration Motors | 1201 | $1.95 | 4 | $7.80 | Adafruit |
| NPN Bipolar Transistors (PN2222) - 10 pack | 756 | $1.95 | 1 | $1.95 | Adafruit |
| 1K Ω Resistor | 4294 | $0.75 | 1 | $0.75 | Adafruit |
| 10K Ω Resistor | 2784 | $0.75 | 1 | $0.75 | Adafruit |
| Circuit Playground Classic Board | 3000 | $19.95 | 2 | $39.90 | Adafruit |
| Bread board | 64 | $5.00 | 2 | $10.00 | Adafruit |
| Alligator Clips | 1008 | $3.95 | 2 | $7.90 | Adafruit |

| | | | | | |
|---|---|---|---|---|---|
| Male to male wires | 758 | $3.95 | 1 | $3.95 | Adafruit |
| Female to female wires | 266 | $3.95 | 1 | $3.95 | Adafruit |

**Results:**
We have a hat with four sensors and four motors installed in which the motors provide feedback to the user as to how far away an object is regarding the sensor it is paired with. Our demo video and additional pictures regarding connections and installation can be found in the videos:images folder.

**Original Schedule:**
- Week 1 (November 11-17)
  - Connect at least two proximity sensors to our arduino by splitting the power and be able to read their distances
- Week 2 (November 18-24)
  - Finish installing other two proximity sensors to our circuit playground
  - Connect vibration motors to our circuit playground and figure out how to control the power it receives
  - Begin to start placing sensors into hat
- Week 3 (November 25-31) (Thanksgiving Break)
  - Present to family and friends for suggestions
  - Finish coding for motors so that their power can be mapped to the distance the sensor corresponding to it
- Week 4 (December 1-4)
  - Work on additional features such as buttons(switch) allowing the user to turn their product on and off when they deem appropriate to do so
  - Using acceleration to detect objects immediately coming at the person

**Modified Schedule:**
- Week 1(November 11-17)
  - Sensors were not in stock so order had not yet been placed
  - Had one sensor to work with and coded for finding the distance of it
- Week 2 (November 18-24)
  - Tried PWM on an LED light to determine if we could map corresponding values of the
  - Implemented transistors into our project so that we would be able to control the power being fed into the vibration motors
  - Worked with two proximity sensors and two motor sensors
- Week 3 (November 25-31) (Thanksgiving Break)
  - Finished coding for the motors

    - Tested our code and we had some interference between the sensors
- Week 4 (December 1-4)
    - Replaced 2 of our motors with 2 LED lights because they broke
    - Transferred our board and wiring into the hat

Our schedules differed a lot mainly due to the fact that we did not receive our sensors until November 22 which was right before Thanksgiving break. Unfortunately we both were not able to work on it much during Thanksgiving break due to certain circumstances and were only able to code for the motors, but could not begin to look at and work on the issues. Once we both were back from break, we were able to begin testing out our sensor and motors. Once week four started we worked on our wiring and code one more time and began to transfer the board to a hat. We ran into a couple of non-function wires so we had to replace those and we also had two motors break, which we replaced with LED's so that we would have something to demo. To place our parts into the hat, we just sewed our parts and board onto the hat and covered it with a ski mask to hide the wiring.

**Issues:**

There were several issues with circuitry that had to be overcome in the project. It was important to supply the proximity sensor with 5v of power for it to function, but the power of the echo signal coming back from the sensor had to be split in two going back into the circuit playground board so as not to overload the board. Getting echo to 2.5v was achieved by using two 10k Ohm resistors, one attached to ground and the other connected to the circuit playground board, with the echo wire connecting these two points.

In addition to this, it was discovered that the amount of power the motors needed in order to feel a perceptible vibration was more than pulse width modulation was supplying. By using a transistor switch, power directly from VBATT could be used, and then the amount of power flowing through the resistor could be modulated with the wire connecting to the middle. This meant that pulse width modulation affecting the central wire gating the transistor could be supplied very little power, and it would be amplified by the transistor so that the haptic motors had enough power to feel their vibration.

There was an issue where objects in front of one proximity detector would seem to be detected from the other sensor on the same circuit playground board through the motor output of the system, although the actual distances ready by each sensor do not have any crosstalk. The way that sensors inadvertabtly interfere with each other is due to the fact that closer sensed objects take far less time to be read by the proximity sensor, and objects further away take much longer to bounce back and be read, so the differences in delay interfere with the pulse width modulation of the motors driven by these functions unless a timer controls for this variable time taken to complete the read cycle.

On the physical aspect of the project, keeping wires in place was challenging, and especially the wired connecting to the haptic motors were difficult to keep intact. The vibration being fed to the motors caused the very brittle and thin wires connected to them to snap frequently. In the future this might be reinforced with solder to prevent them from being as vulnerable to breaking. It was also difficult threading the wires through the hat, so a better design involving sewing wires in place will be designed in the future.

The amount of time it took for the proximity sensors to arrive made it challenging to work within the schedule constraints of the project, since there were delays in their delivery. This set back the whole project, preventing stages of the testing process for occurring as early on as we would have liked.

**Future Work:**

For the future we would like to replace the ultrasonic sensors with a more animal friendly device due to the fact that while we may not be able to hear the ultrasonic wave, animals such as dogs can and it may be very distracting to them. We also worry about how long term use of the device might affect hearing due to the fact that the waves being sent out are 40KHz and despite not being able to hear the sound, our ears still receive the wave and considering the sensors are in a hat, there would be constant exposure to these waves. We also would choose sturdier and higher quality motors due to the fact that they kept snapping. Soldering the wiring in the future would help reduce the bulk and weight of the circuitry and allow a sleeker design that could be sewn in place. Implementing a switch to turn the system on or off, as well as into other modes such as high, medium, and low sensitivity could help with the usability of the device. Adding in the feature of velocity would also be helpful for the user to be able to toggle between, since this can reduce the constant feedback of nearby, yet static, objects which may become annoying over time, only notifying the user if they move from their current position and change velocity.

**References:**

https://www.addicore.com/2N3906-PNP-Amplifier-Transistor-p/152.htm
http://www.learningaboutelectronics.com/Articles/How-to-reduce-voltage-with-resistors.php
http://users.ece.utexas.edu/~valvano/Datasheets/PN2222-D.pdf
https://dronebotworkshop.com/hc-sr04-ultrasonic-distance-sensor-arduino/#1