

The C-ing Hat (2019): A hackathon (Hack UCI) experiment for the visually impaired.

Creative Process:

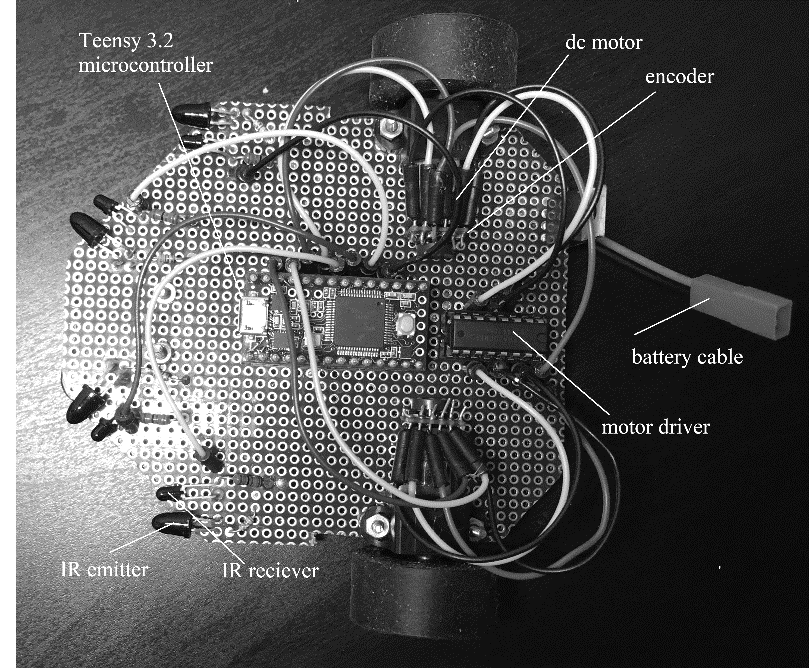
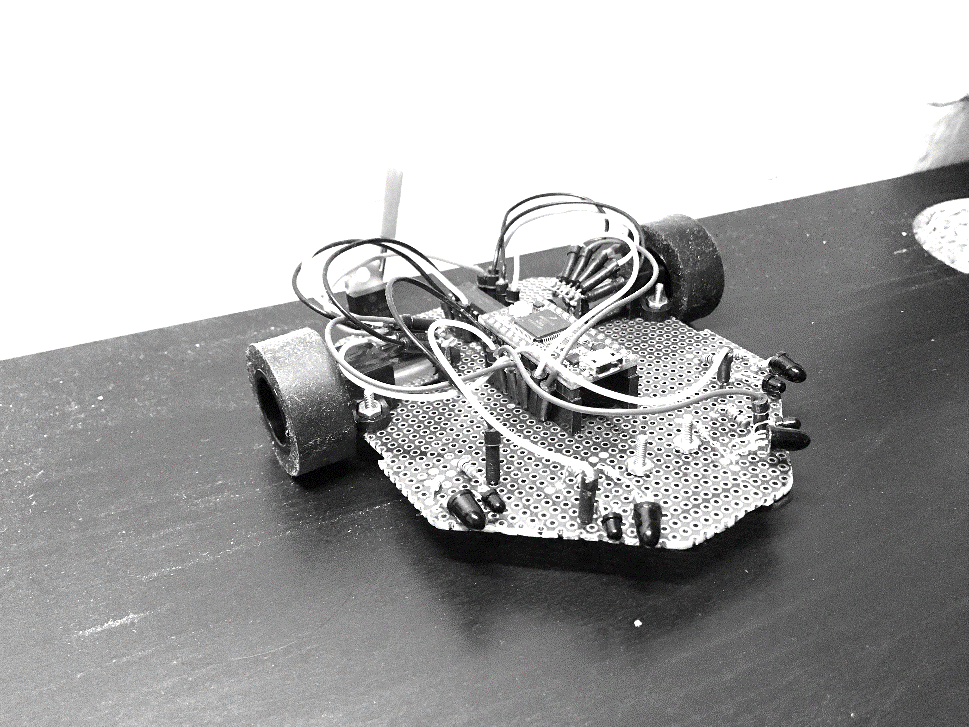
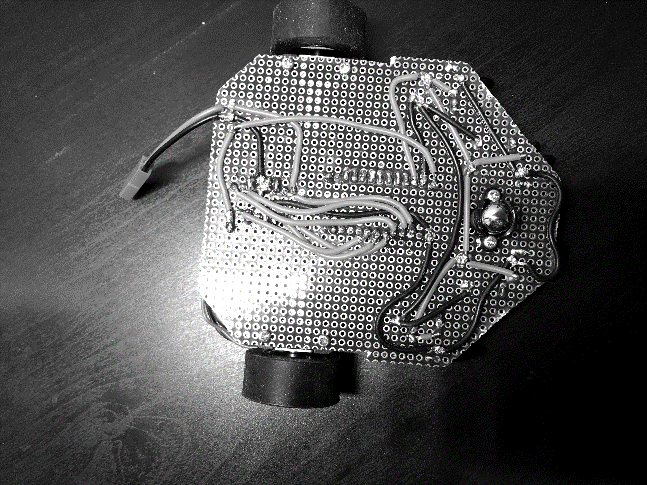
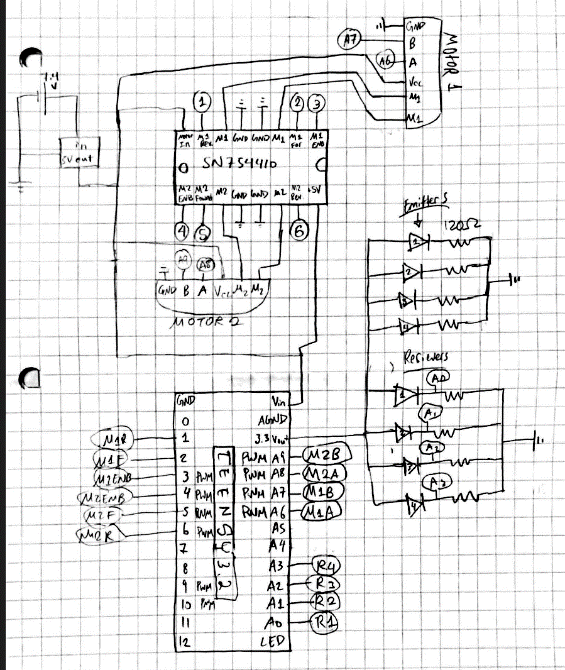
At my first hackathon, my team and I decided to do a hardware project. My team was provided two ultrasonic sensors that birthed the idea of creating a wearable electronic with distance sensing applications. The input would be the distance sensors while the output would be both auditory (for the visually impaired) and visual (for those around the user).

Application:

The hat can detect objects up to 1 meter (adjustable value) in front, left and right of the wearer. It then sounds out a buzzer sound with pitch based on how close the object is to the person. The farther away the object is, the lower in pitch the sound is. The hat also makes longer buzzer sounds if the object is farther away. There are LEDs on the side of the hat to alert people around the wearer.

Construction:

We used an Arduino Uno to control the ultrasonic sensors (distance sensors), buzzer and LEDS. The sensors and outputs are connect via a breadboard and jumper wires. The micro-controller is powered by a 9V battery. The logic was created in the Arduino IDE which uses C++ (hence the name of the hat). The frame of the hat is built of an egg carton and lots of tape.

Micro-Mouse (2019): Autonomous maze solving vehicle

Micro-mouse at UCI is a sub-branch of IEEE. The focus of the club is to create an autonomous “mouse” that can solve a maze without human intervention. There are competitions where students race to see which mice can solve mazes fastest.

Hardware:

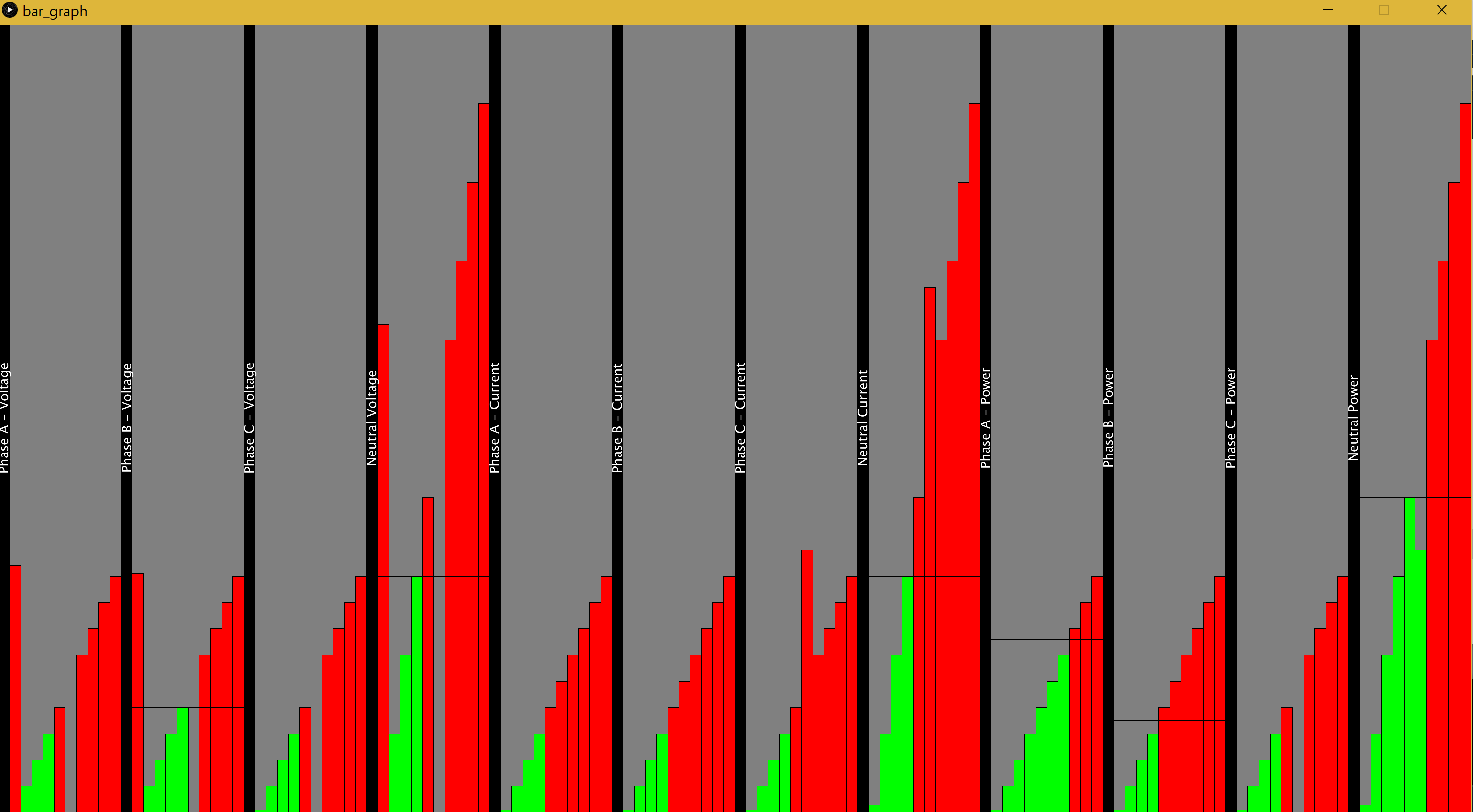
IR receiver emitter diode pairs are used to detect walls. The receivers feed distance variables into a Teensy 3.2 (Arduino based microcontroller). The IR receivers can detect walls in a maze to direct the mouse. The mouse is steered by a pair of 6V DC motors. Encoders are attached to the DC motors to track revolutions in order to maintain straight movement and proper turning. All components were soldered to a through-hole PCB. A parts list was provided by the club.

Firmware: (in progress)

I created sample code in Arduino C++ to test each of the IR receivers, motors, and encoders. Currently I am linking together these components.

Software: (in progress)

After reading and researching different algorithms, I have decided on using the flood fill algorithm that is standard for most micro-mouse. This algorithm is the most efficient compared to other such as dead reckoning or dead end learning. The idea is to imagine that the entire maze has no walls. Each cell is assigned a value based on distance from the center (goal). As the mouse traverses the maze and discovers new walls, the distance value for each cell is updated.



Processing 3 three phase waveform demo