ENVIRONMENTAL SENSING FOR CYCLISTS

Team #36 Presentation

Yanda Hu, Zhaoxin Gu, Jiayi Ke

TA: Yuchen He

Outline

- Overview
- Block Diagram
- Voltage Regulator
- Motor Sub-Circuit
- Control System
- Power Distribution Board
- Summary

Introduction

- Cyclists killed rose from 729 in 2014 to 818 in 2015 and still rising
- Estimated cyclist injuries at around 45,000 in 2015
- Total cost of cyclist death and injury is over $4 billion per year
- Our project aims to help improve cyclist safety by alerting them of hazards behind them while cycling

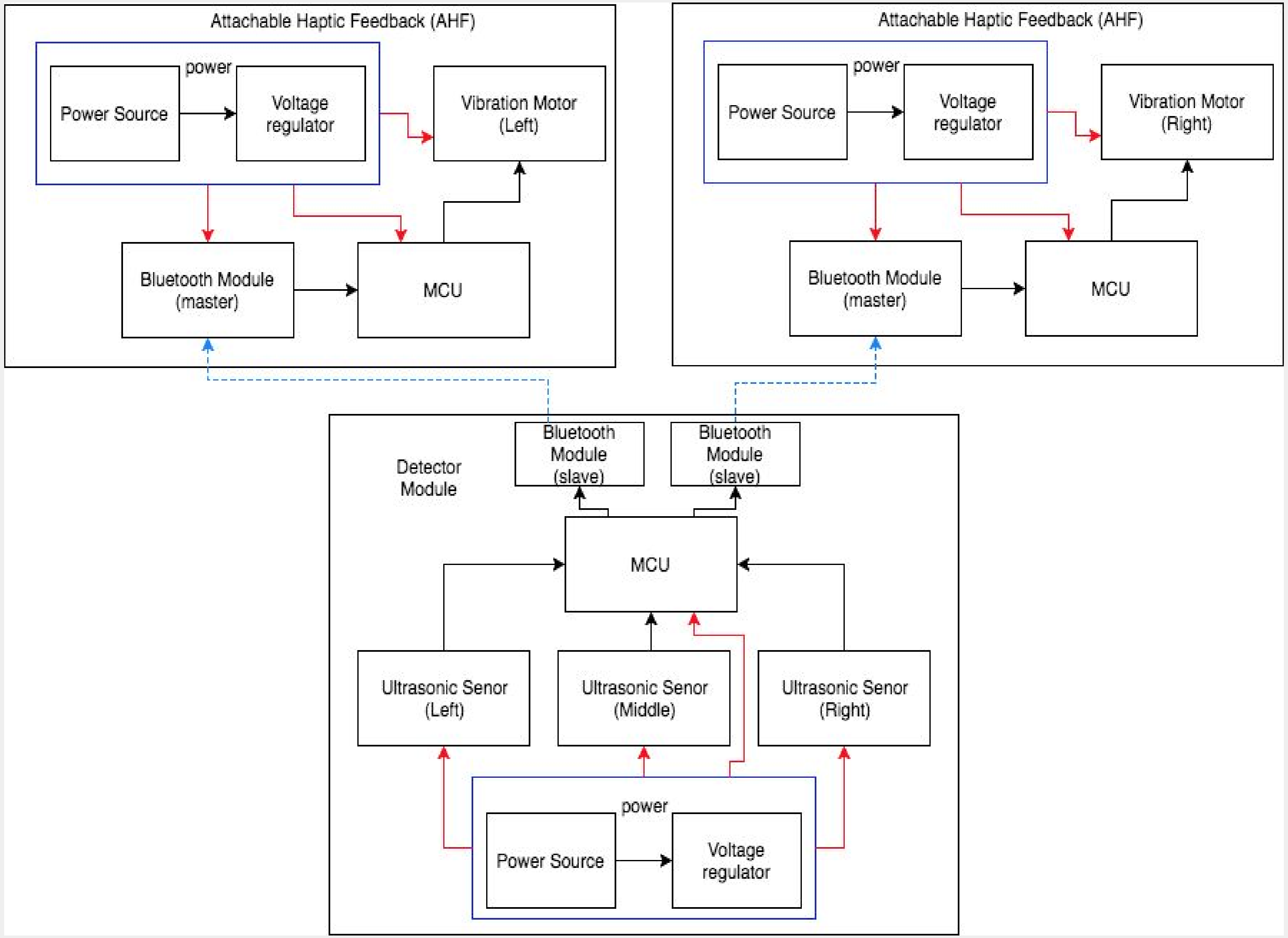
Objective

- Sensors will need to be able to detect hazards behind the user up to 4m
- Communication between sensors and the AHF will need to be possible
- Vibration motors on the AHF will need to vibrate at varying intensities depending on how far away hazards are. They will be installed in attachable cases that the user will wear on their forearm.

Block Diagram

Product consists of two types of modules communicating with one another:

AHF (attachable haptic feedback)
and the Detector Module



Voltage Regulator

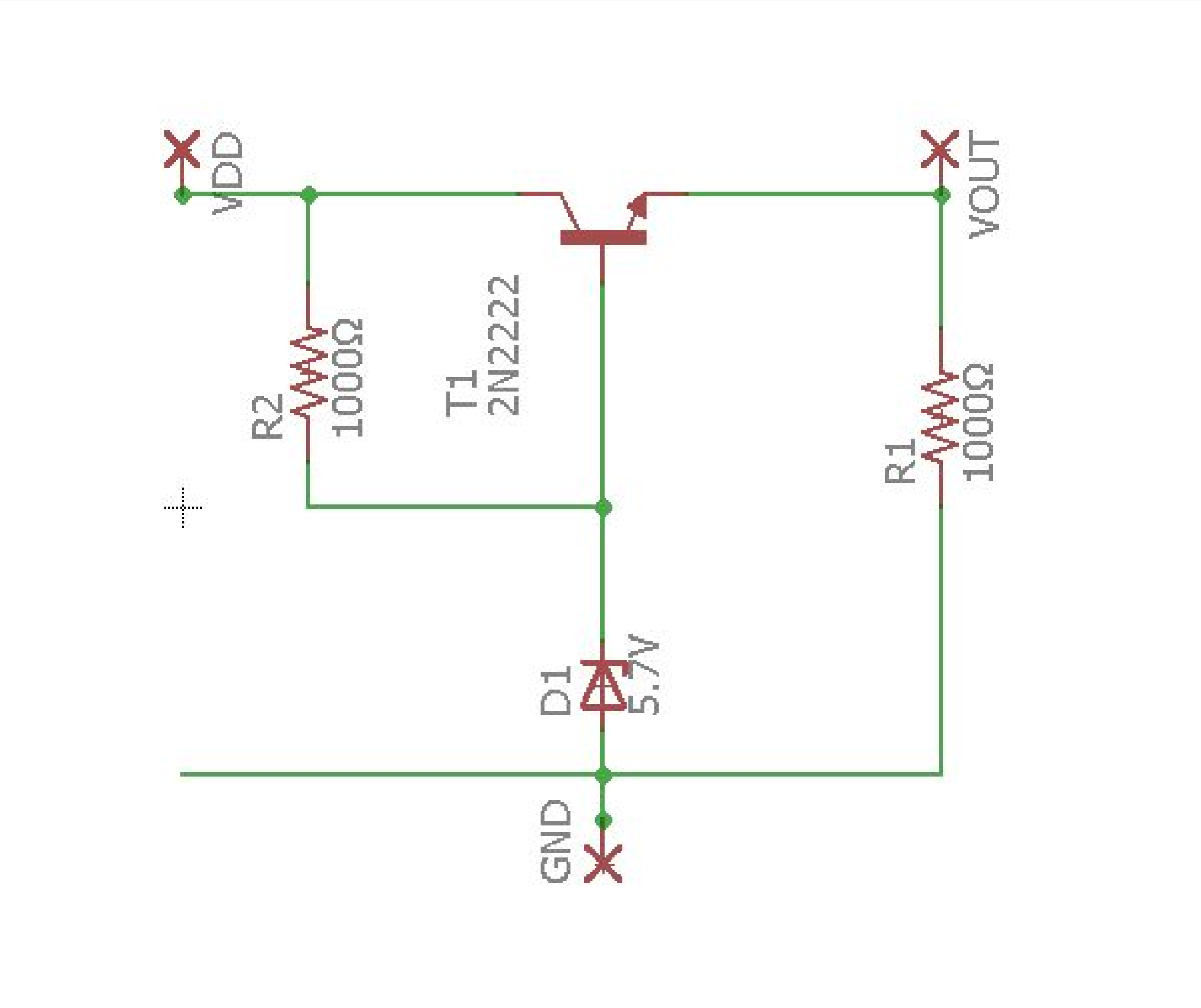
Battery life - 9V lithium ion battery

Time elapsed (min)	Battery voltage level (V)
0	8.956
30	8.723
60	8.607
90	8.590
120	8.580

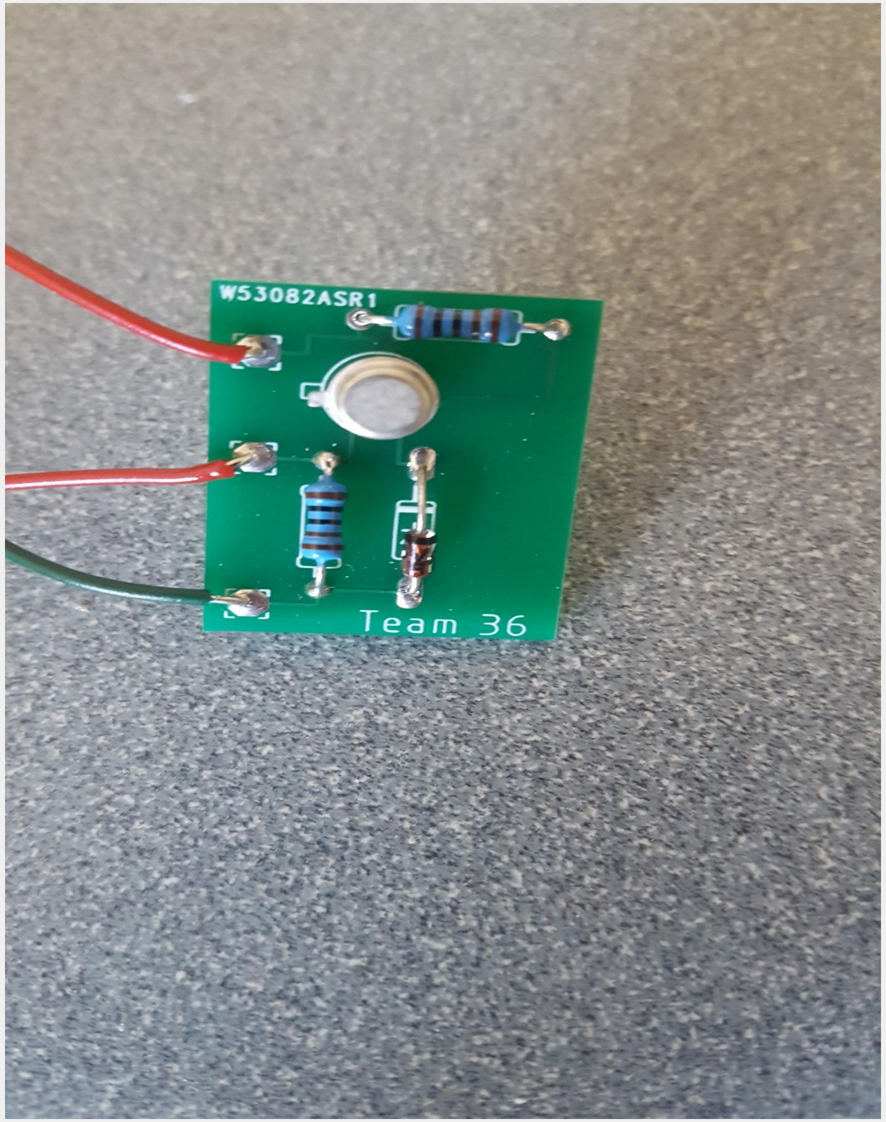
Even after 2 hours of powering the motors, the battery level did not drop significantly, thus, the battery will be able to operate the product for a long period of time efficiently.

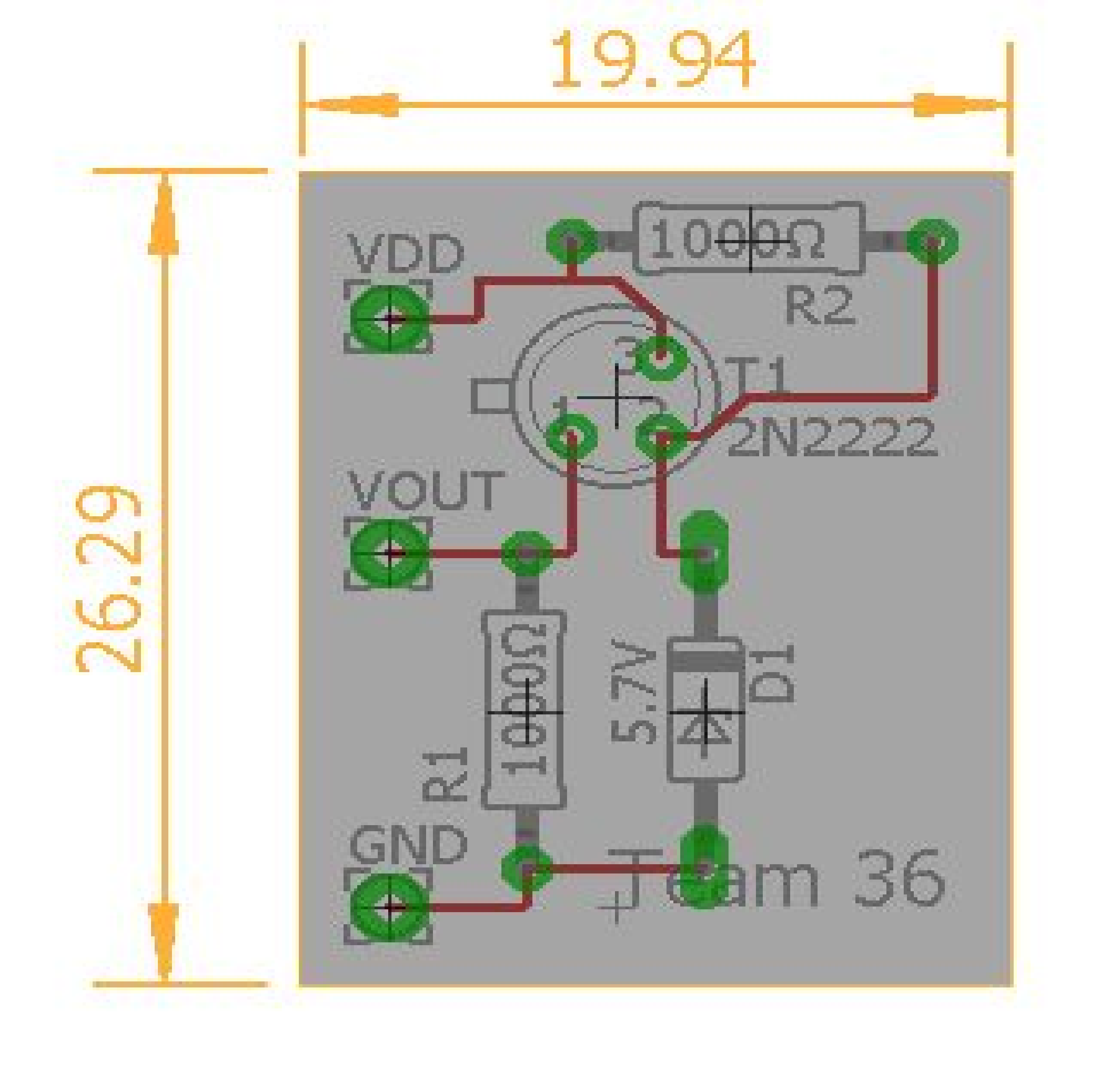
Voltage regulator Requirement

- Regulates a 9V input voltage to be ~5V Vout
- Vout is dependent on the breakdown voltage of zener diode
- Vout also dependent on Vbe of BJT, which is 0.7V.
- Ideal zener diode breakdown is 5.7V, however, the closest value we could find on market was 5.6V



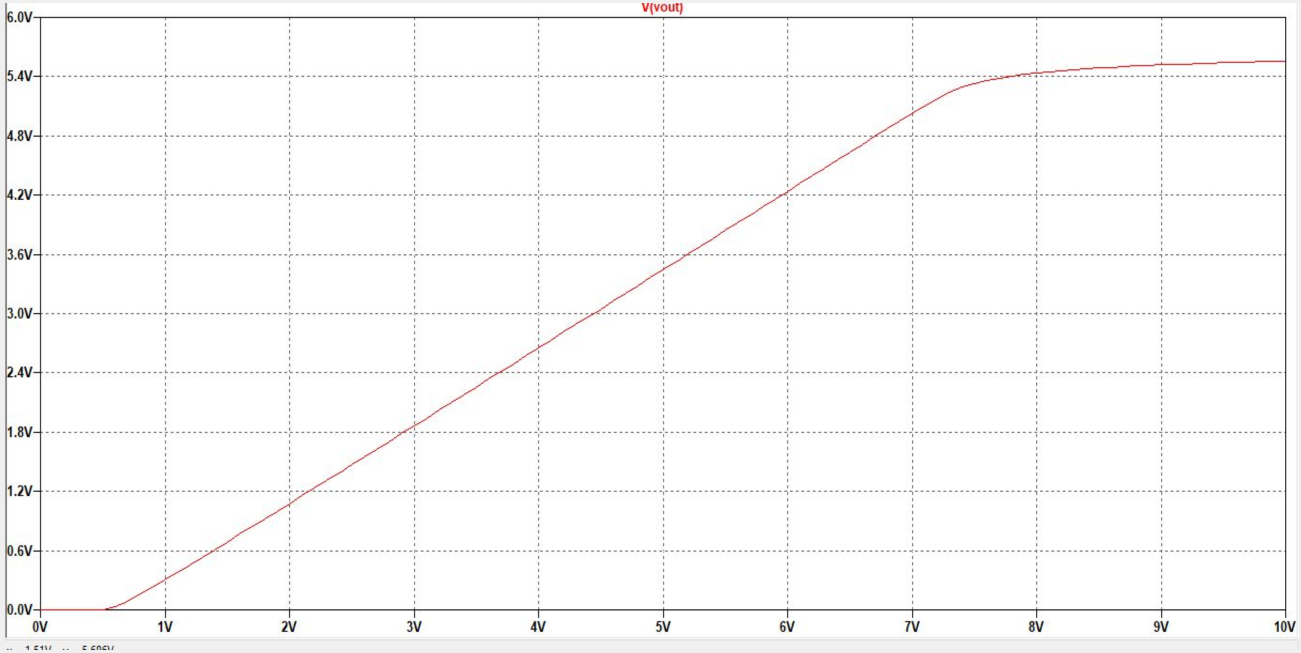
Voltage Regulator PCB

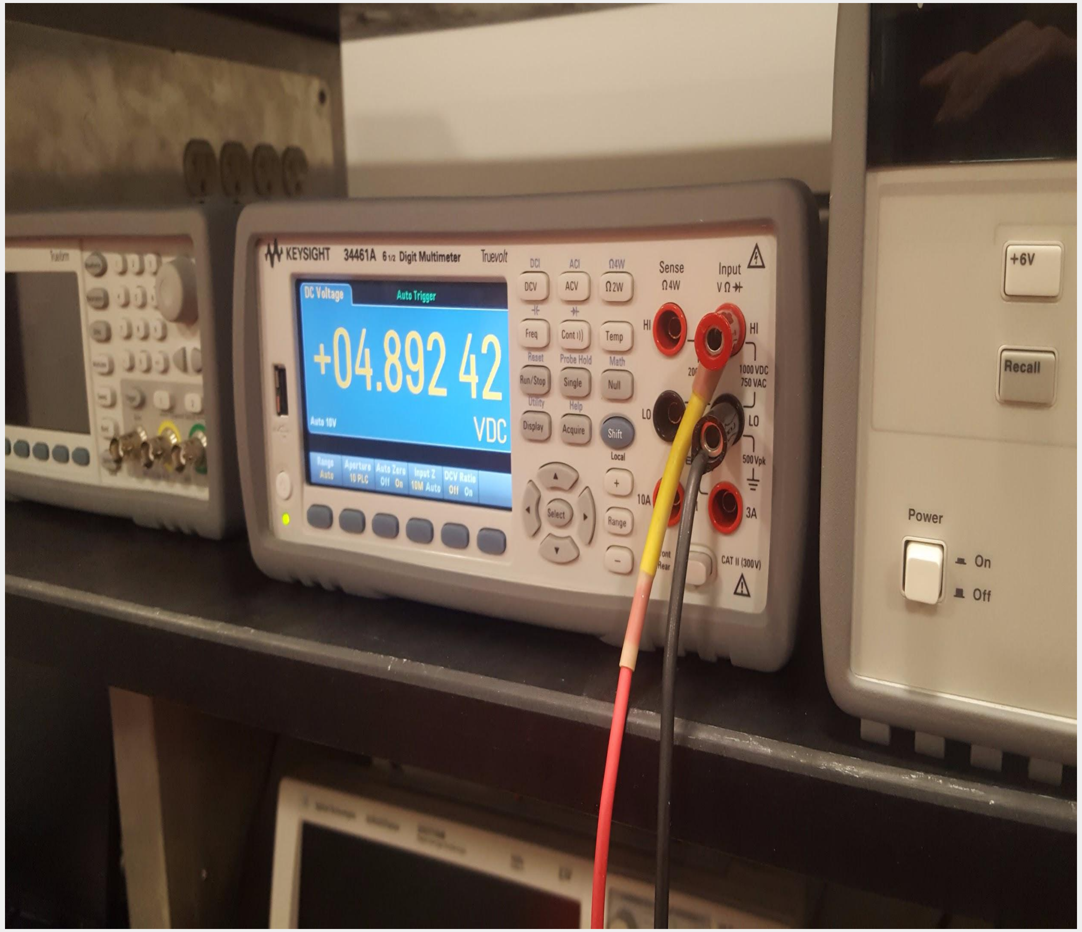




Verification of Voltage regulator

- With Vin of 9V, the multimeter will show Vout as ~4.9V
- Graph below uses the circuit shown before and sweeps Vin from 0 to 9. Vout is on vertical axis and will plateau at ~5V.





LTSpice did not have 5.7V zener diode either, so I used 6.2V zener diode instead which results in a Vout of ~5.4V.

Motor Sub-Circuit

Vibration motor Sub-Circuit

Responsible for :

- Driving the Motor safely according to the specifications of the motor
- Controlling the Motor's on/off based on the PWM signal from the micro-controller

Requirements:

- Takes 5V power supply from Voltage regulator module to supply the motor
- Max Collector Current no larger than 80 mA
- Turns the Motor On/Off according to Signal from Microcontroller

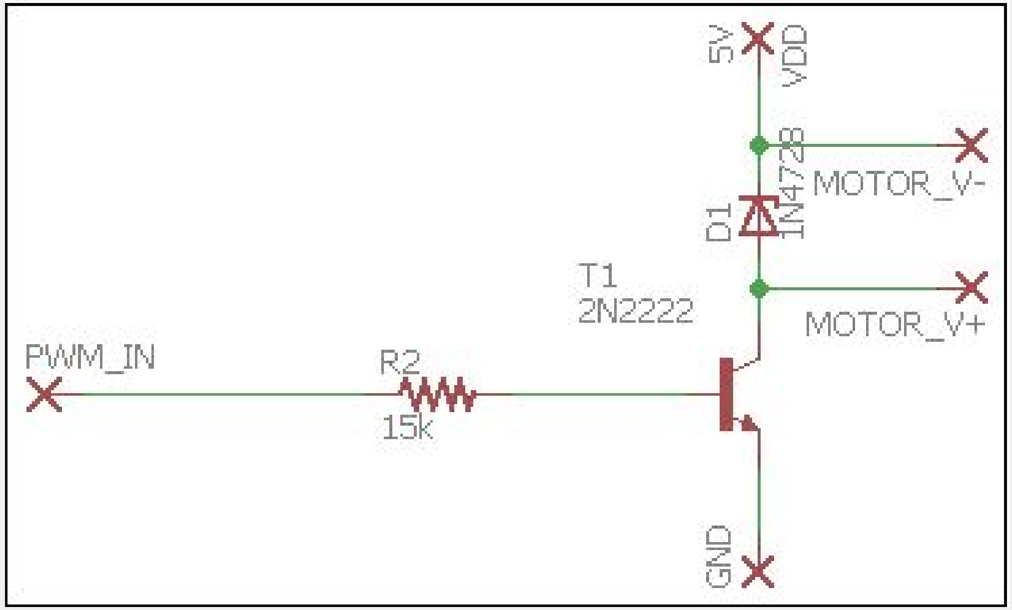
Vibration motor Sub-Circuit

$I_C = \beta\ I_B$

$I_B = V_{PWM\_IN} / (R_2 + R_{BE})$

With $R_{BE} = 1k\Omega$, $\beta = 200$, $V_{PWM\_IN} = 5V$, and $I_C <= 80$ mA while operating

We have $R_2 = 11.5k\Omega$

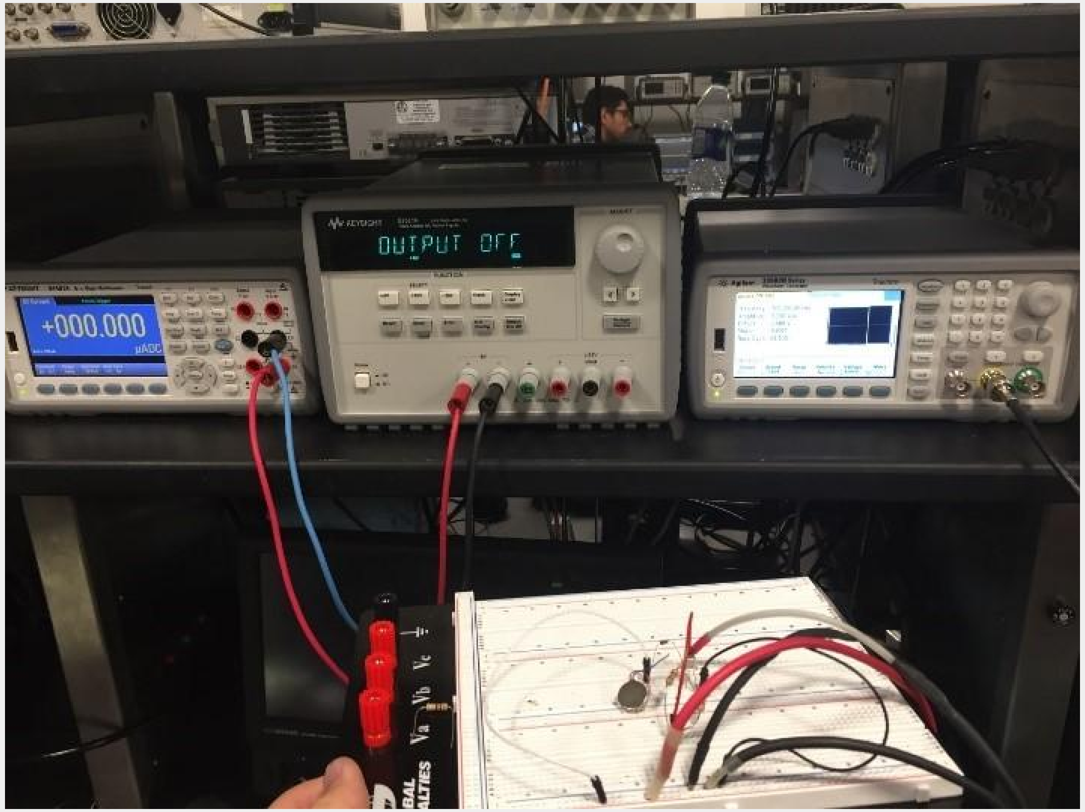


Vibration motor Sub-Circuit
- verification

Actual value obtained:
$\beta = 216$
$R_{BE} = 1.13\ k\Omega$

Actual Resistance used for R_2:
$R_2 = 15\ k\Omega$

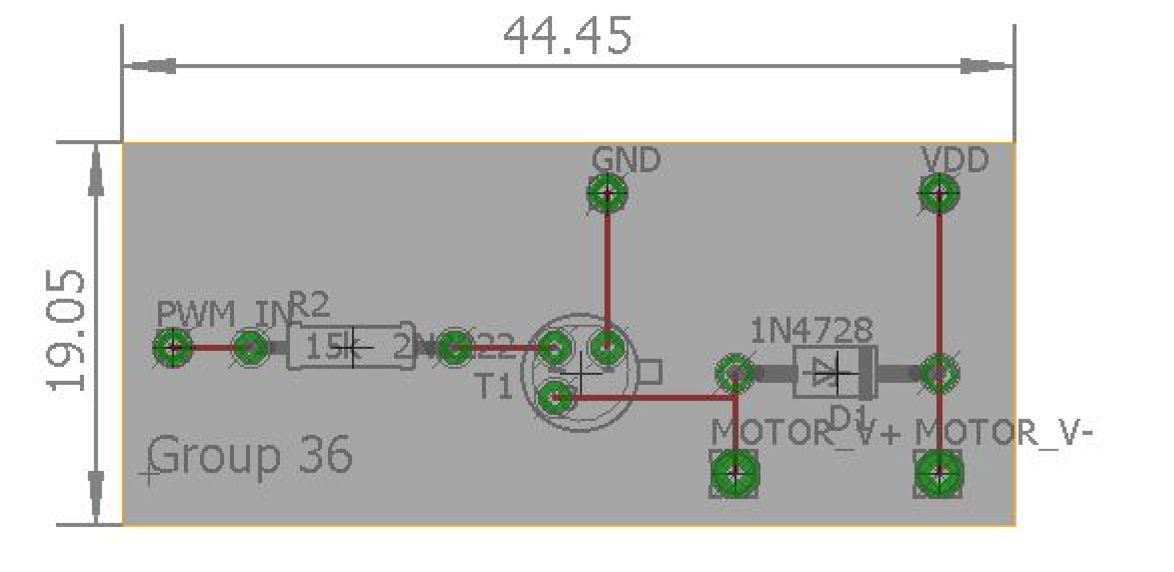
Measured Current:
$I_B = 0.31mA$
$I_C = 67\ mA\ (< 80\ mA)$

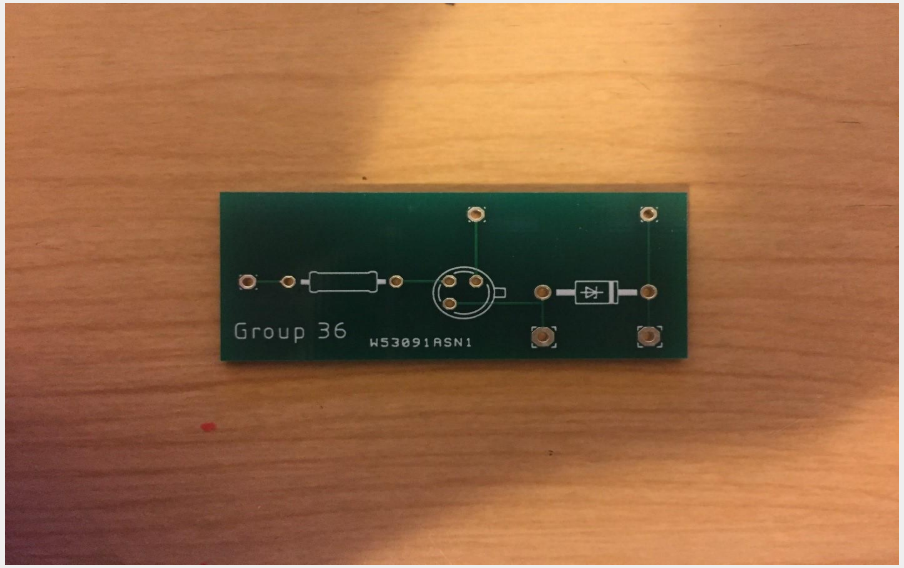


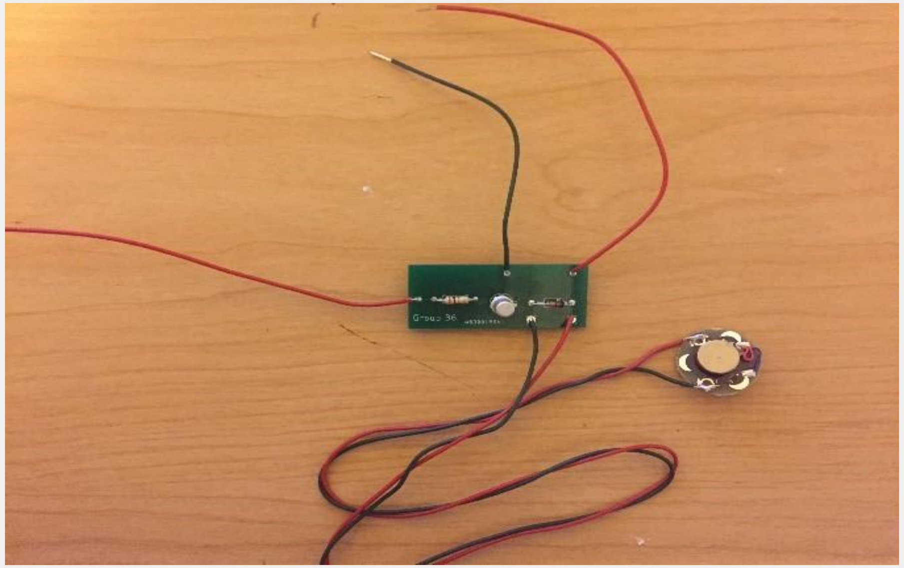
Vibration motor Sub-Circuit
- verification

Frequency	Vibration Strength 20% Duty Cycle	Vibration Strength 50% Duty Cycle	Vibration Strength 80% Duty Cycle	Comment
1 Hz	Low	Medium	Strong	Switching on and off become very noticeable
100 Hz	Low	Medium	Strong	Noticeable change in Vibration Strength
1000Hz	Medium	Medium	Less Strong	Change in Vibration Strength become less obvious

Vibration motor Sub-Circuit

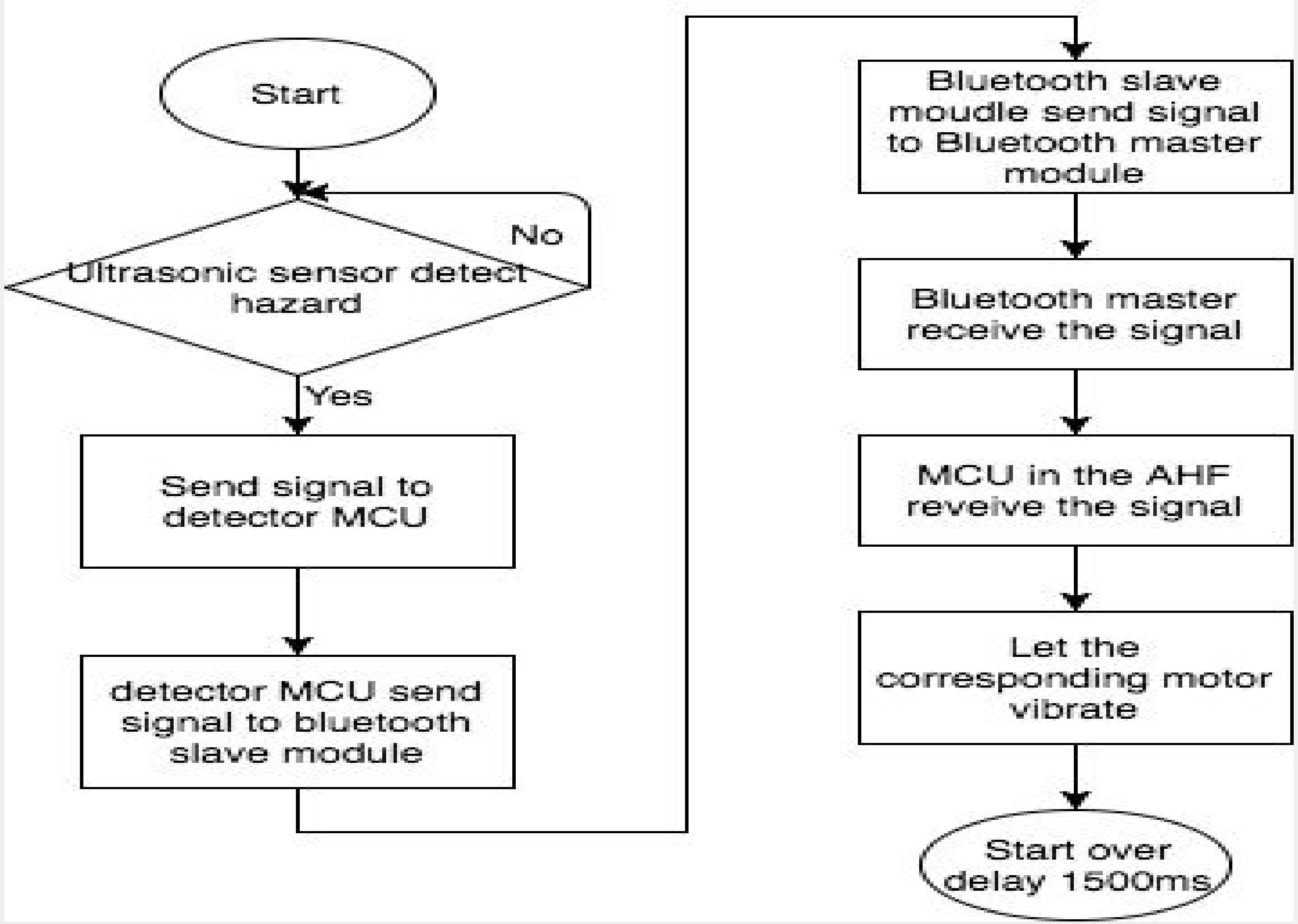






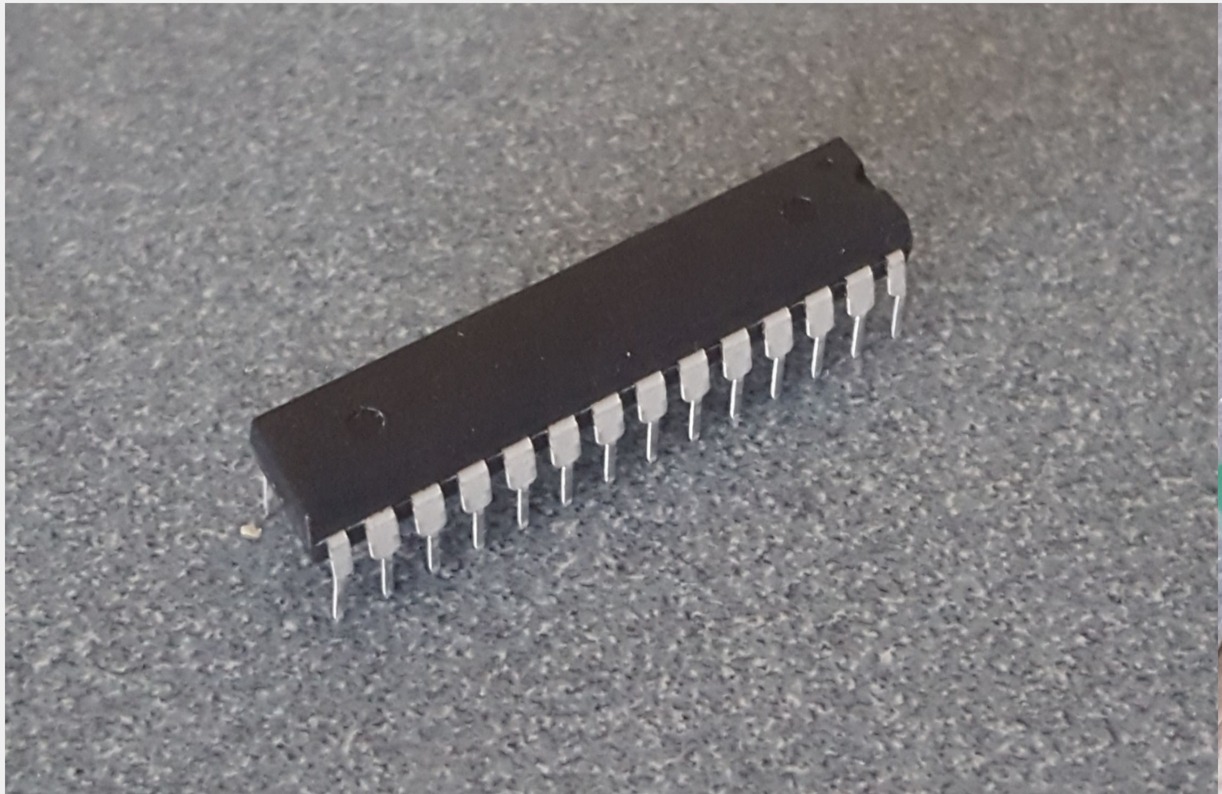
Control System

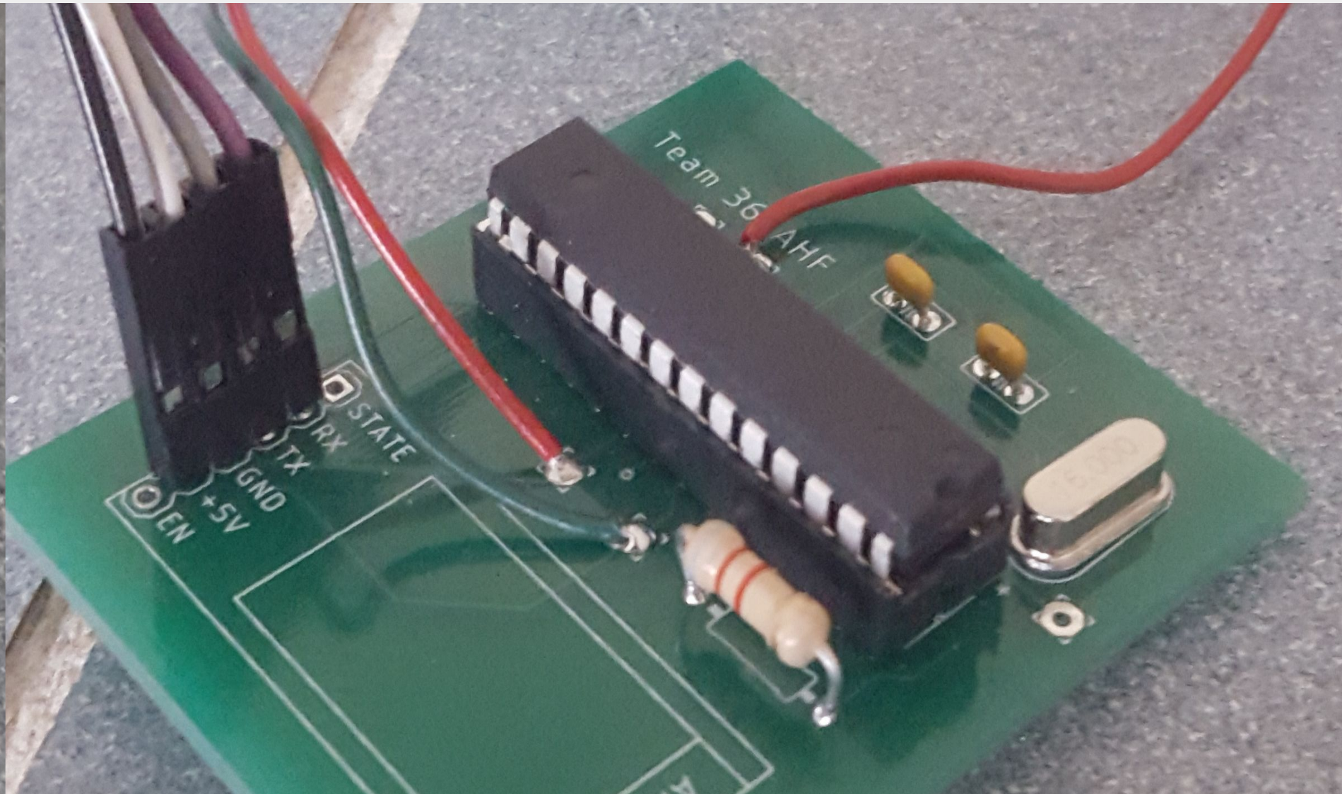
Program Flow chart



Microcontroller

We are using Atmega328P as our microcontroller, which is compatible with Arduino.



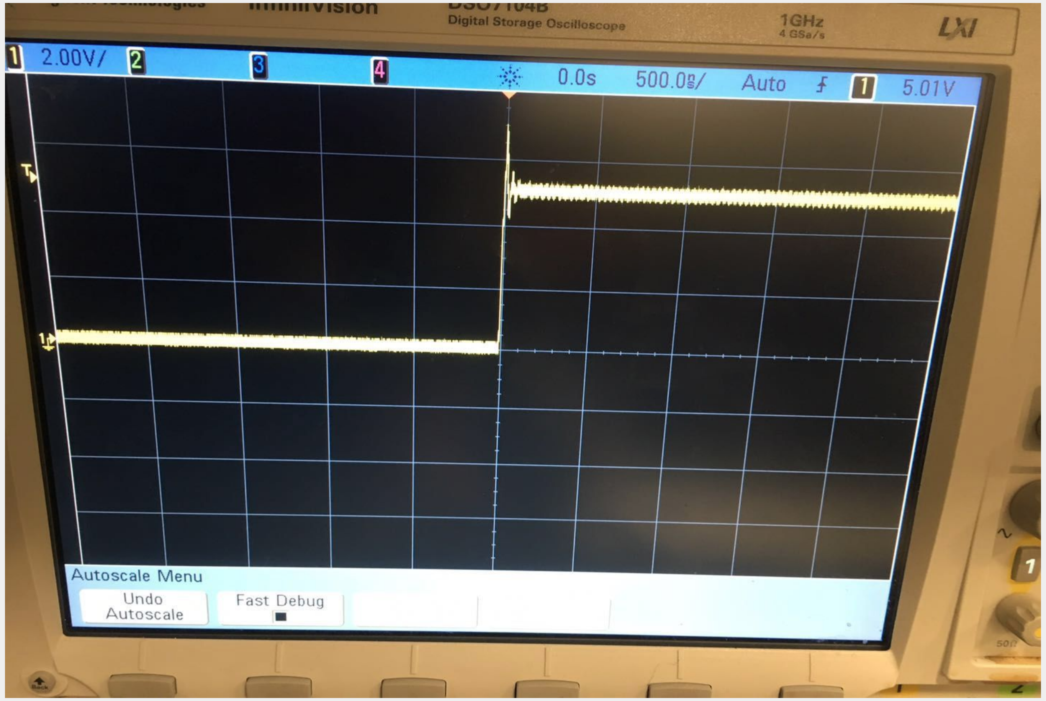


Microcontroller Requirement

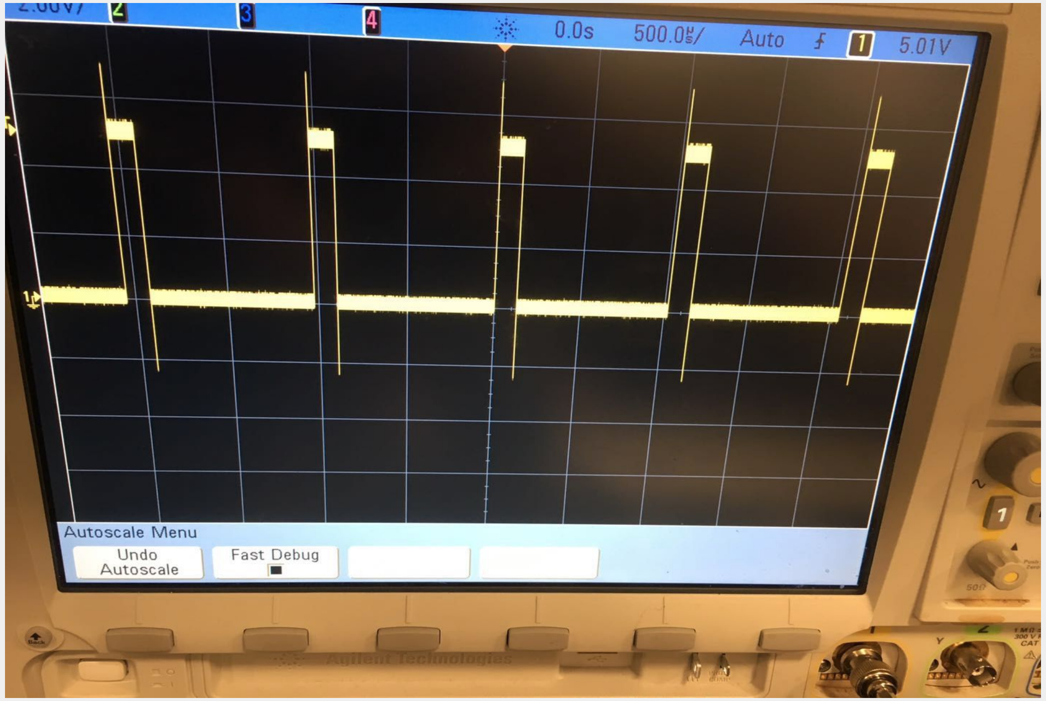
1. When connected to PCB, microcontroller should control Bluetooth module to receive and send data
2. For the PWM signal using digital output, the pins 3, 5, 6, 10, 11 in the MCU, the duty cycle can be modified to control the intensity of the motor by analogWrite() function, analogWrite(255) should be a duty cycle of 100%.
3. For those pins do not have a PWM output, we can use digitalWrite() function and to change the duration of the HIGH and LOW voltage to get the effect of a PWM.

Microcontroller Verification

2. For the PWM signal using digital output, the pins 3, 5, 6, 10, 11 in the MCU, the duty cycle can be modified to control the intensity of the motor by analogWrite() function, analogWrite(255) should be a duty cycle of 100%.



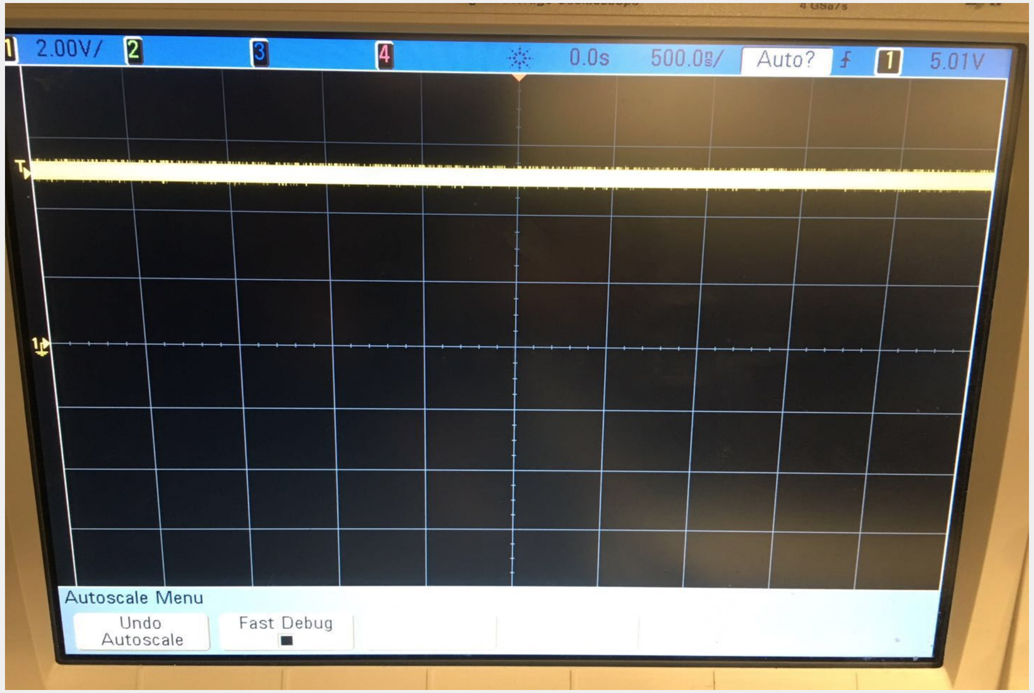
analogWite(128) 50% duty cycle



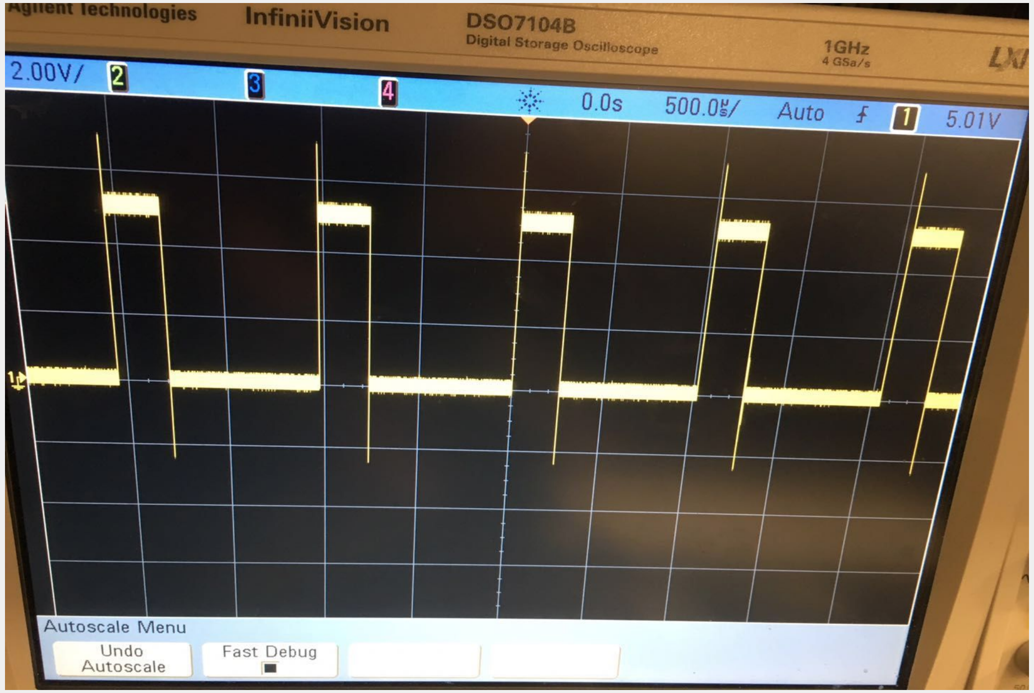
analogWite(32) 12.5% duty cycle

Microcontroller Verification

3. For those pins do not have a PWM output, we can use digitalWrite() function and to change the duration of the HIGH and LOW to get the effect of a PWM.



Digital output with no pull-LOW delay

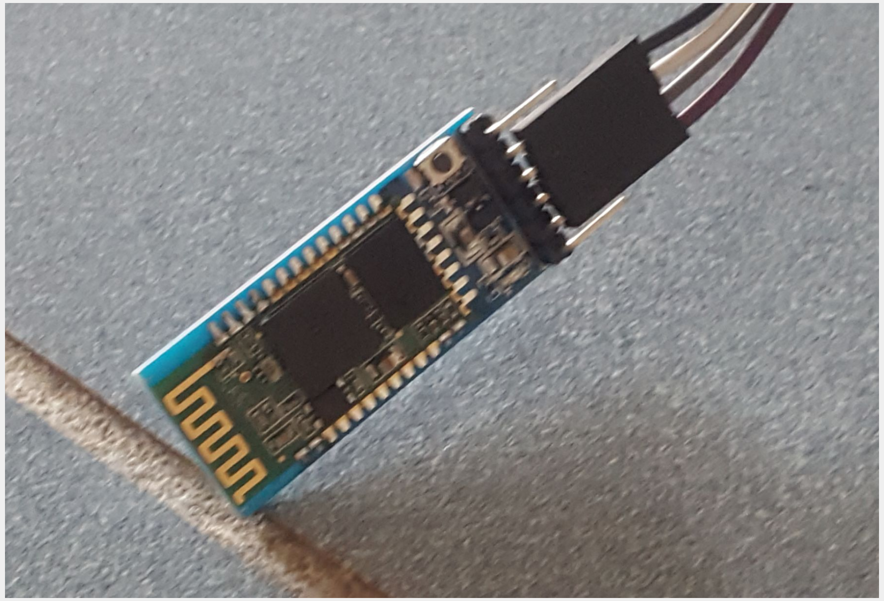


Output HIGH for 3ms, LOW for 9ms, 25% duty cycle

Bluetooth

We are using HC-05 as our master and slave bluetooth module.

Since it can set mode between master and slave mode, we set the mode of the four bluetooth into two pairs.



Bluetooth (two pairs)

Bluetooth encoding: 4 bits binary numbers.
Left two bits represent the intensity of the left motor,
Right two bits represent the intensity of the right motor.

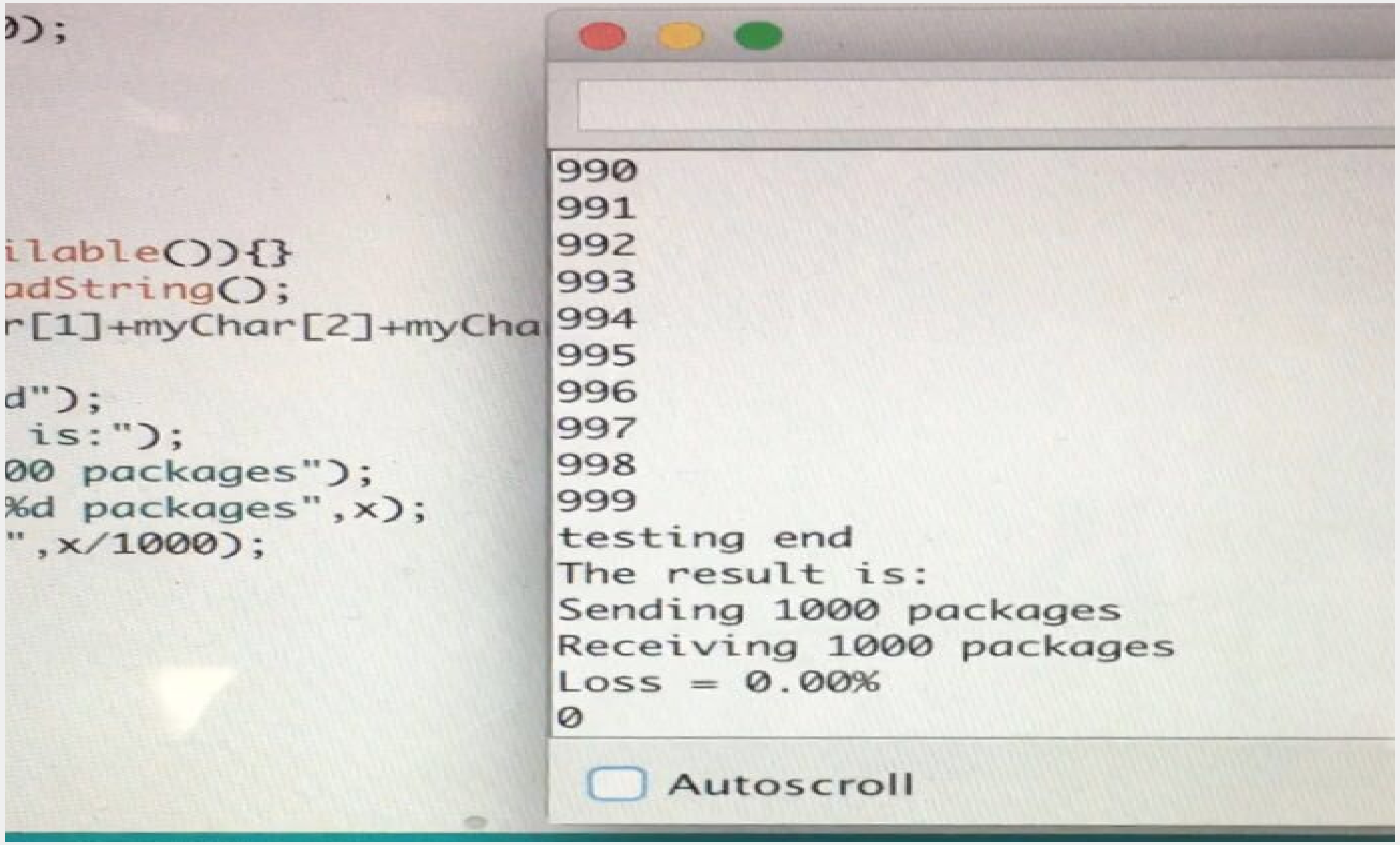
0000 (No vibration)	Both motor	Left motor	Right motor
Low intensity	0101	0100	0001
Medium intensity	1010	1000	0010
High intensity	1111	1100	0011

Bluetooth Requirement

1. Can always receive coherent signal (>99% accuracy, which is <1% loss) from the bluetooth master module
2. Receive bluetooth signals within 100cm.
3. The delay should be within 0.3 seconds since we need time for the cyclists to react to the hazard.

Verification of Bluetooth

1. Can always receive coherent signal (>99% accuracy, which is <1% loss) from the bluetooth master module

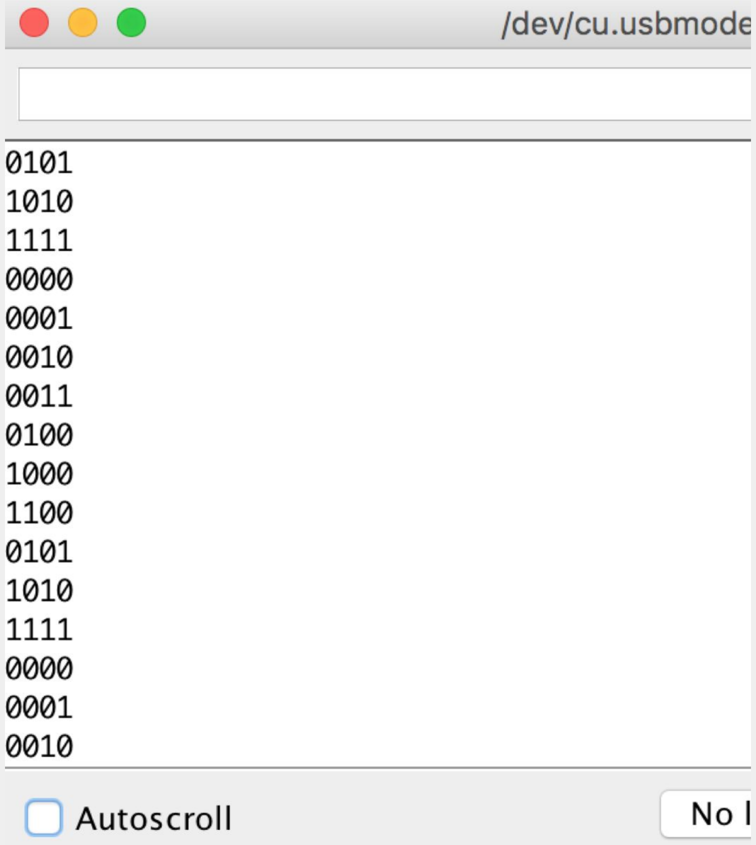


Verification of Bluetooth

2. Receive bluetooth signals within 100cm.

```
BTSerialSlave.write("0000");
delay(1500);
BTSerialSlave.write("0001");
delay(1500);
BTSerialSlave.write("0010");
delay(1500);
BTSerialSlave.write("0011");
delay(1500);
BTSerialSlave.write("0100");
delay(1500);
BTSerialSlave.write("1000");
delay(1500);
BTSerialSlave.write("1100");
delay(1500);
BTSerialSlave.write("0101");
delay(1500);
BTSerialSlave.write("1010");
delay(1500);
BTSerialSlave.write("1111");
delay(1500);
```

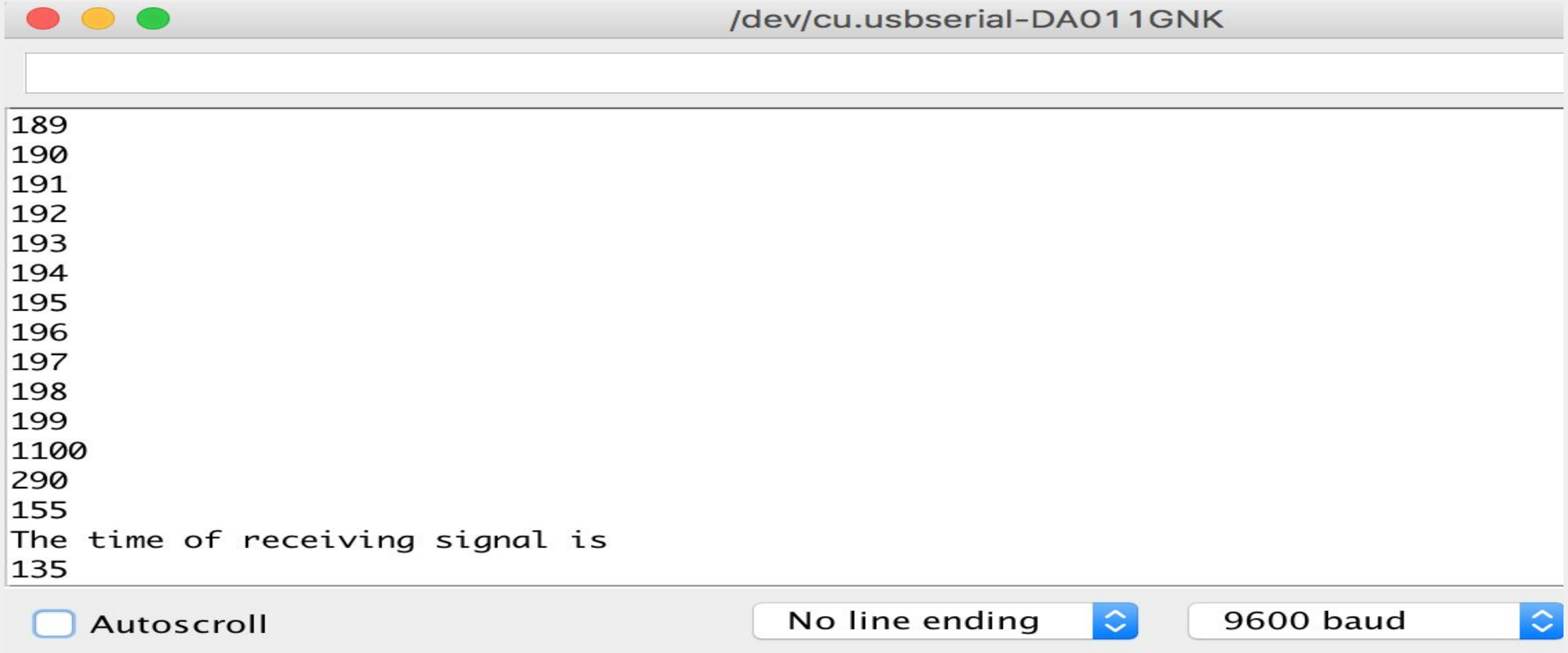
Slave side sending signal



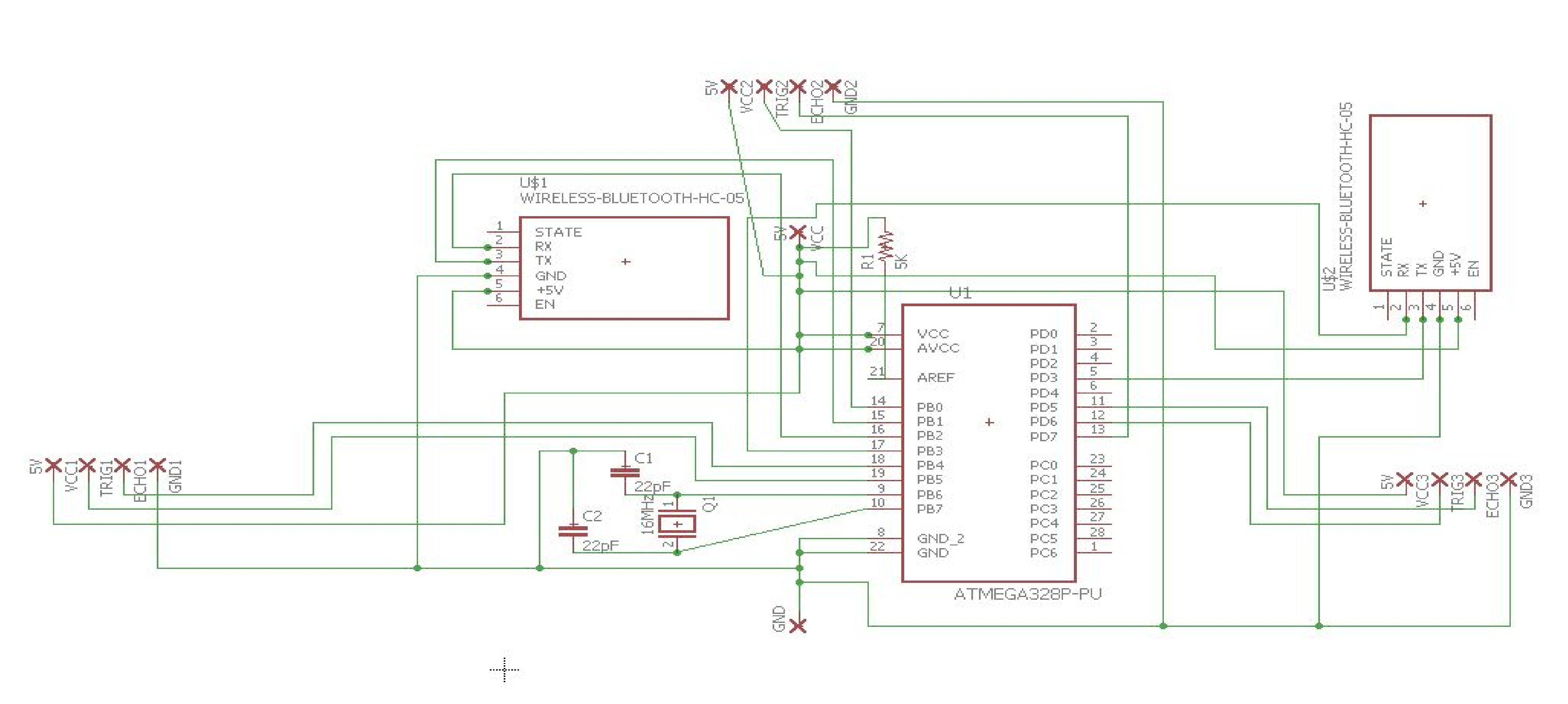
Master side receiving signal

Verification of Bluetooth

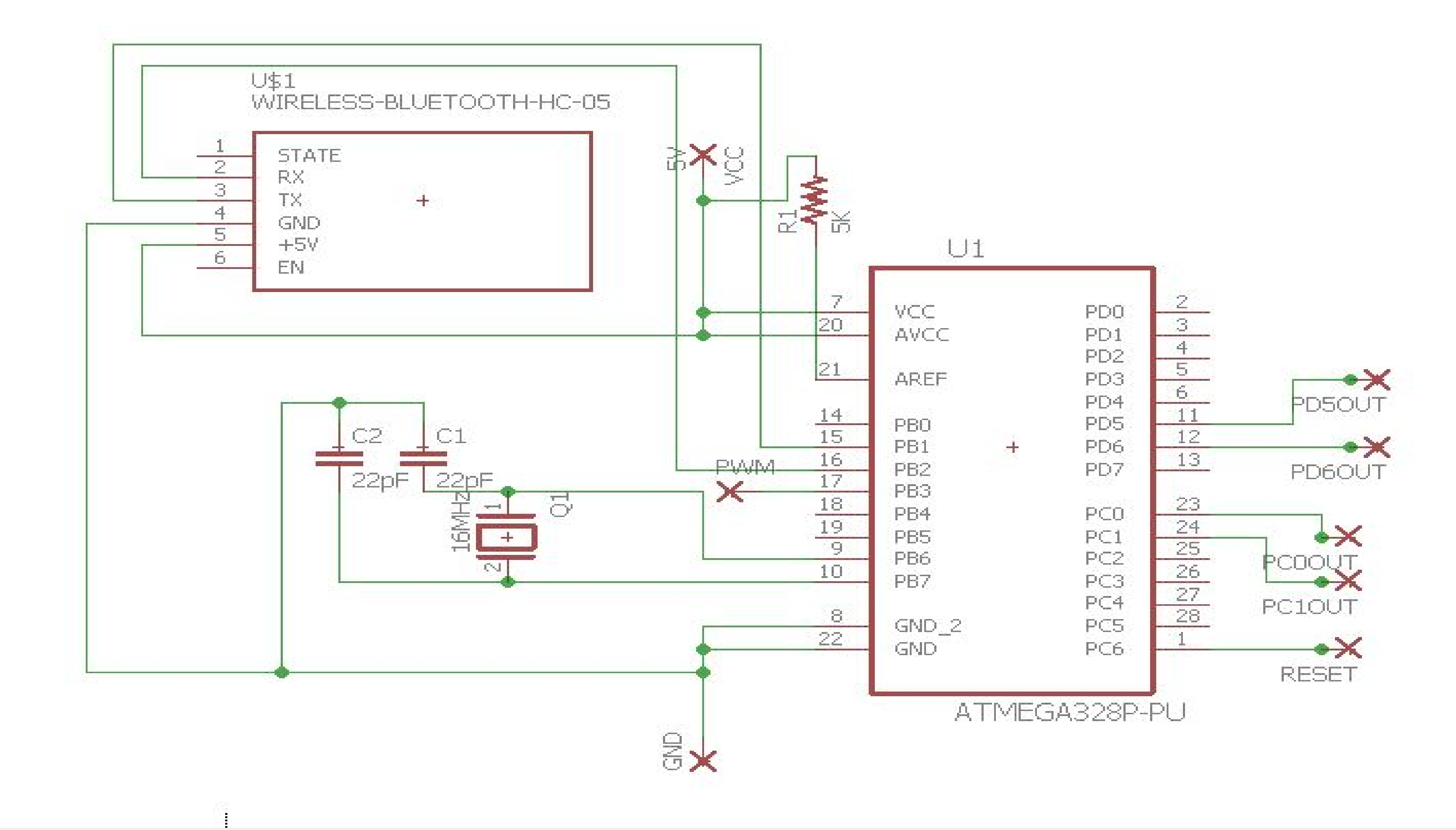
2. The delay should be within 0.3 seconds since we need time for the cyclists to react to the hazard.



MCU schematic for detector

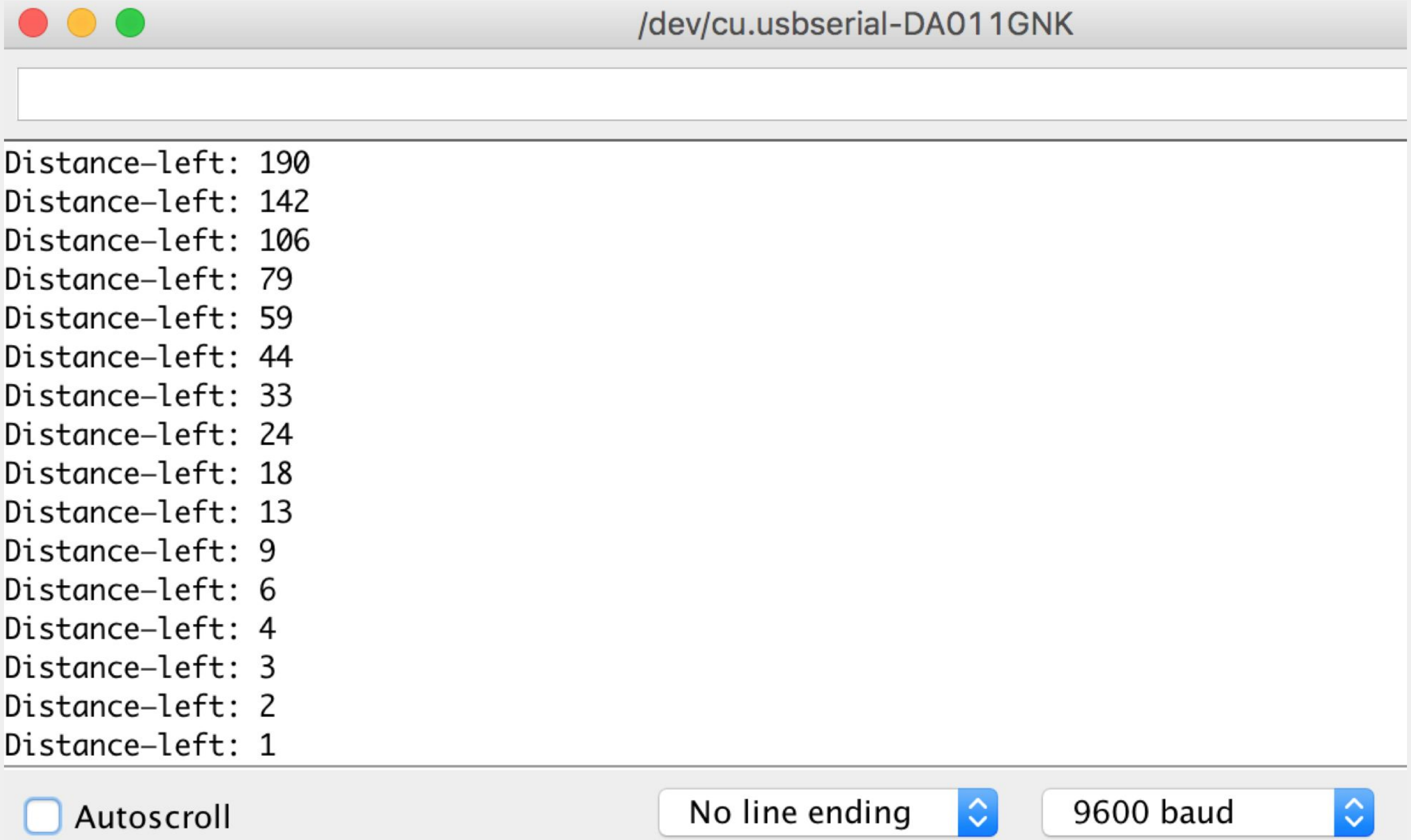


MCU schematic for the AHF



Ultrasonic Sensors - HC04

- The ultrasonic sensors use DC 5V
- Max range of 4m
- Measuring angle 15 degrees each side
- Sensor emits signal from one side and receives signal on the other side, distance will be calculated by measuring time taken to receive the signal.



Testing result on the ultrasonic sensor

Power Distribution Board

Power Distribution Board

Responsible for :

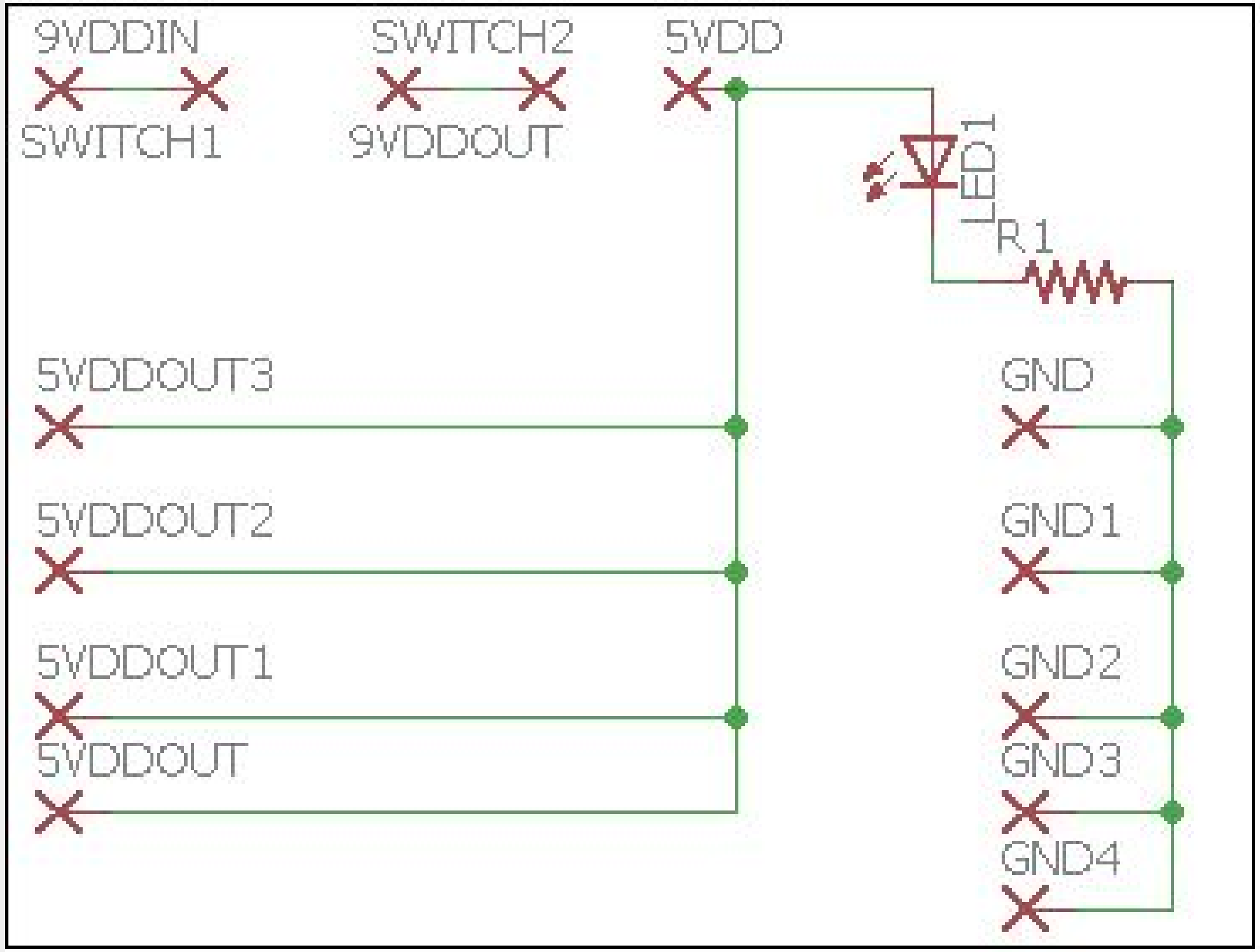
- Controlling the On and Off of the AHF or Detector Module
- Indicating working status using LED
- Distributes power to other PCB boards and provides common GND

Requirements:

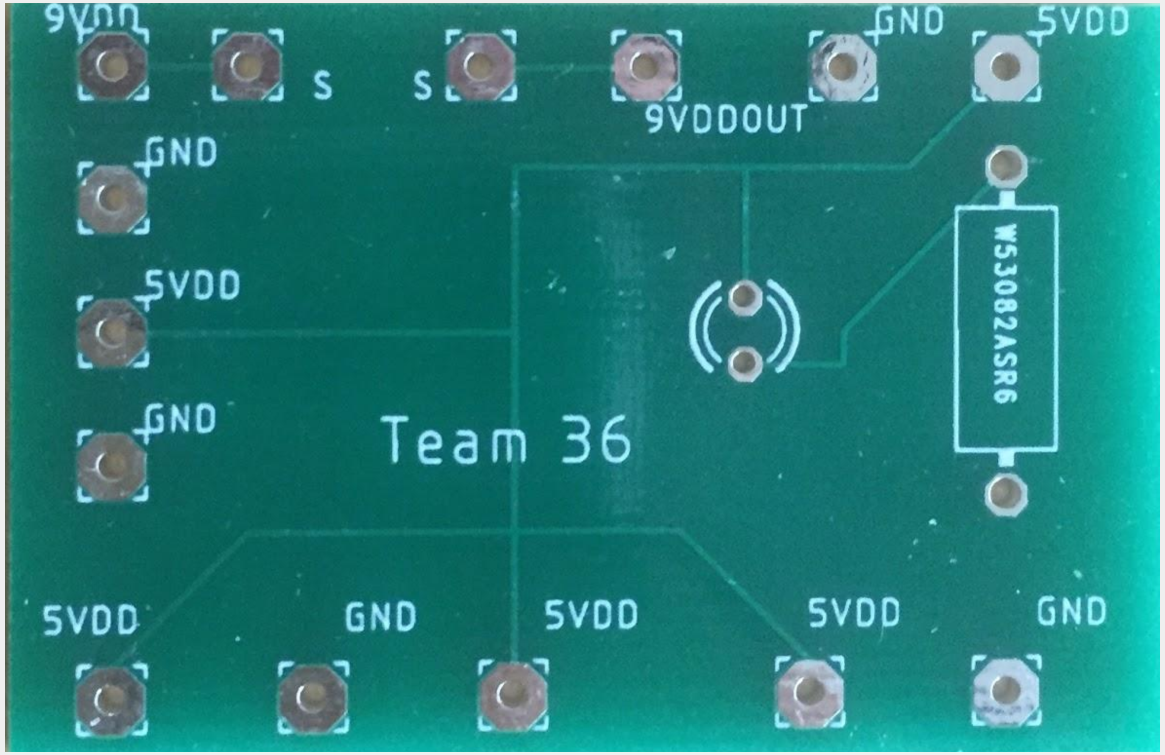
- The switch should be able to turn all other component on and off
- When turned on the LED should be lighten up
- Provides relative same voltage to other PCB

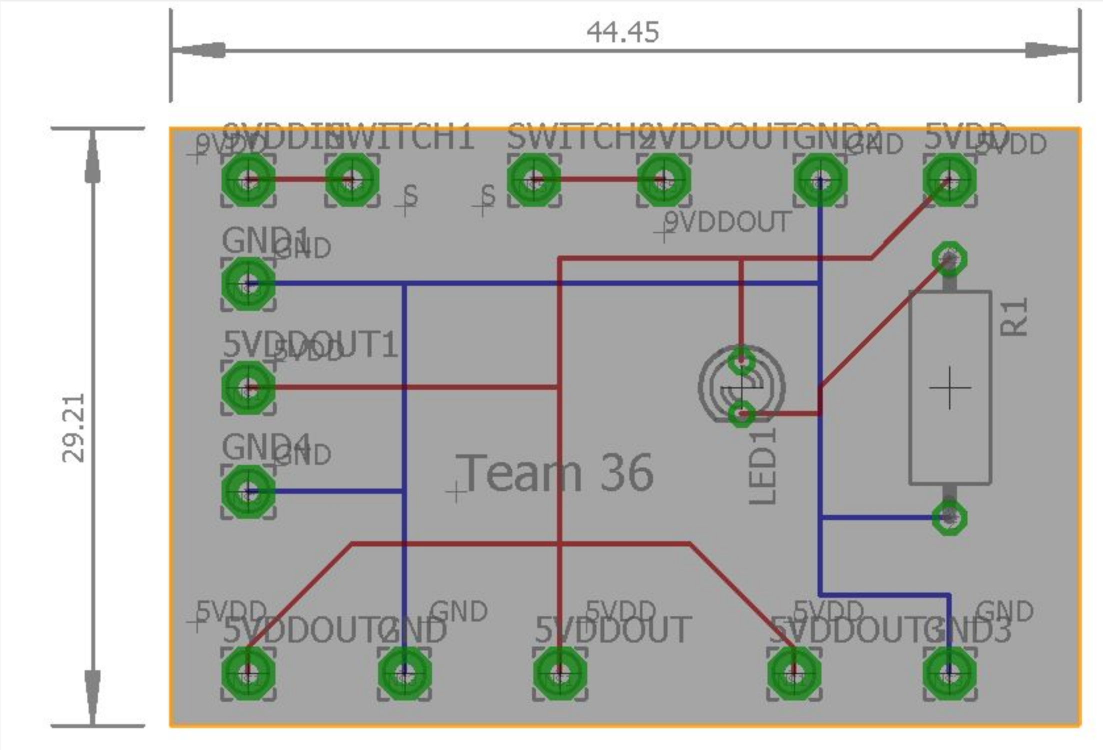
Power Distribution Board

- 9V Power supply coming from battery
- Switch connected to battery power input to control operation
- Other port of the switch connected to voltage regulator input and 5V port connected to voltage regulator output
- Multiple 5VDD and GND for multiple PCBs



Power Distribution Board

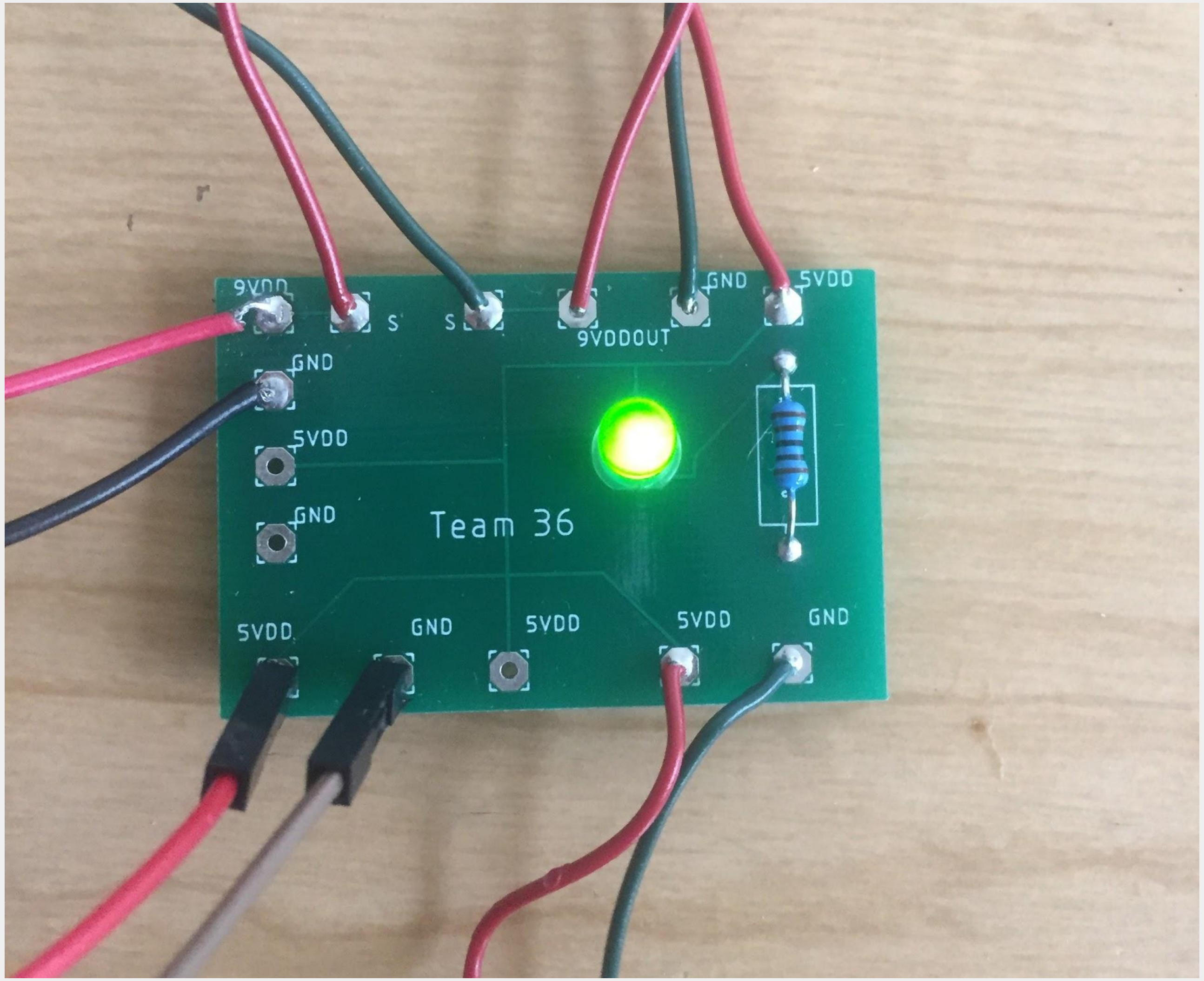




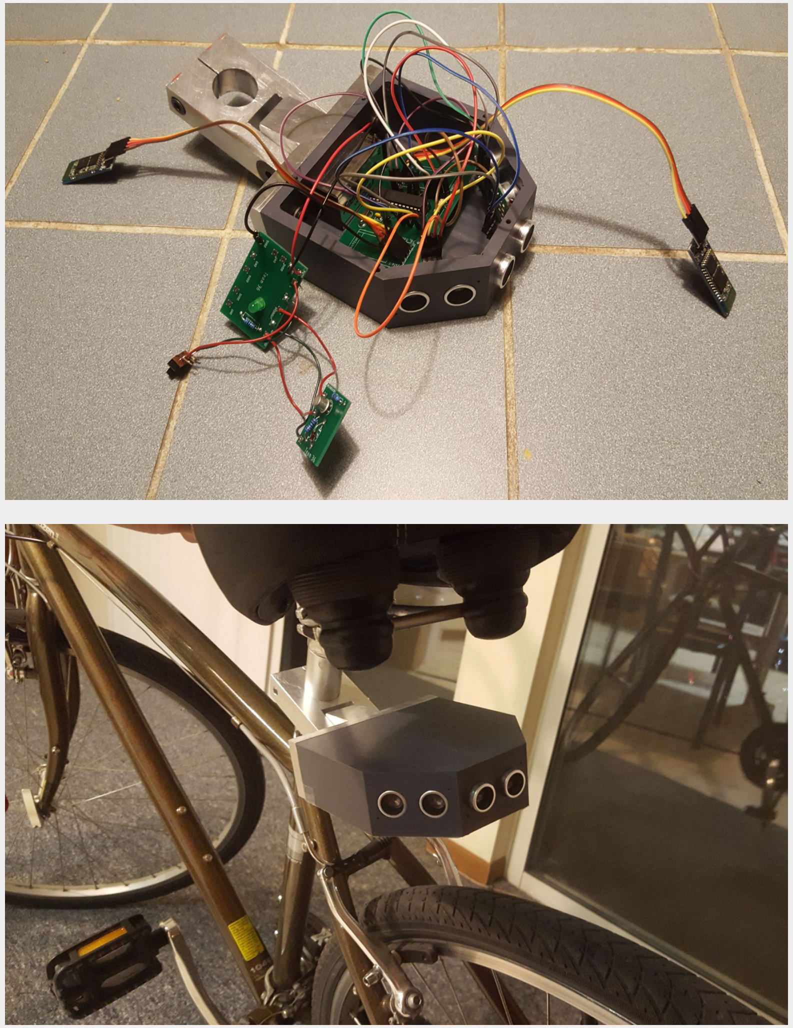
Power Distribution Board

Fully connected power distribution board:

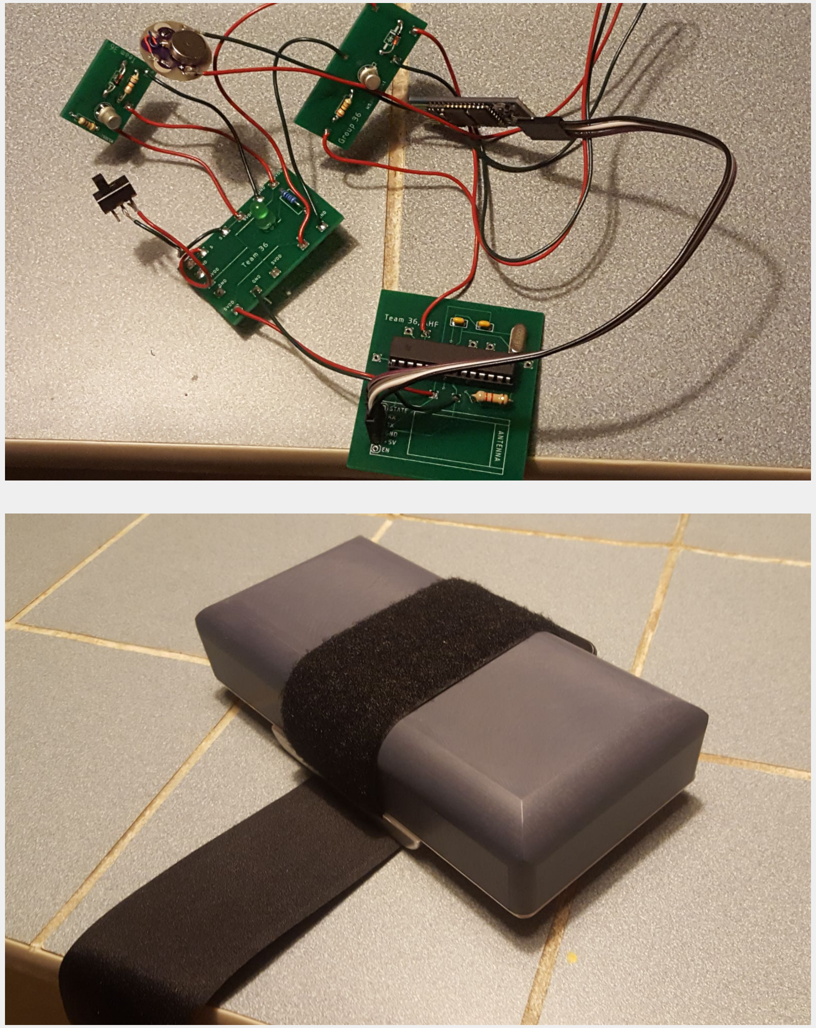
- Switch connected using wires to allow more flexibility on final board arrangement
- VDD and GND grouped together to allow easier connection
- $V_{regulator}$=4.93V and VDD = 4.91V +/- 0.4% at 5VDD ports



The final product



Detector Module



Attachable Haptic Feedback Module

Conclusion

- Overall, all the individual components of our project worked fine, up to the point of integration where there were issues with the ultrasonic sensors and the bluetooth communication
- Possible improvements could be to use two microcontrollers for the detector module as the module uses two bluetooth devices. This will place less strain on a single microcontroller as it would reduce the computational cost per MCU once the workload is divided.

Future work

- Possible future projects could be to use high end ultrasound sensors to improve the resolution of the distance detection and increase the range.
- Refine power distribution and PCB layout to improve driven signal
- Reshape the casing for AHF module so the user can wear the device more comfortably
- Integrate all individual modules into one PCB for a cleaner appearance

Questions