Environmental Sensing For Cyclists

Zhaoxin Gu
Yanda Hu
Jiayi Ke

Final Paper for ECE445
Fall 2017
TA: Yuchen He

Team: 36

Abstract

This report documents the process of the ideation, development and testing of the environmental sensing device for cyclists. Looking behind you while on a bike increases the risk of an accident because even a few seconds of being distracted is all it takes for an accident to happen, which can cause cyclists long term injuries with crippling medical bills. Using the product we have developed, our group aims to reduce the risk of riding a bicycle so that cyclists will not need to constantly look behind them, yet, they will still know when and which direction a vehicle is approaching behind them, therefore, they will always be able to focus on the road ahead of them. The device utilizes haptic feedback from a vibrating motor on both arms to warn the user where the hazard is detected, and a sensory case is attached onto the bike below the seat.

1. Introduction

1.1 Objective:

Bicycles are one of the best ways to use as a method of transportation or just for recreational use, however, when you're on a bicycle, you are more susceptible to dangers on the road due to poor visibility and less protection than that of a car.

Looking behind you while on a bike increases the risk of an accident because even a few seconds of not looking in front of you is all it takes for an accident to happen, which can cause the cyclist to sustain long term injuries and heavy medical bills. There are currently countermeasures for preventing these problems right now such as the "Garmin Varia Rear-view Radar Rear Bike Light" which you can get on the market for 200$, however this product will alert you of cars behind you via warning lights flashing on the GPS fitted onto the bike, but it requires the user to look down and spend precious few moments of not paying attention to the road ahead, which is still dangerous.

Our group aims to reduce the risk of riding a bicycle so that cyclists will not need to constantly look behind them, yet, they will still know when and which direction a vehicle is approaching behind them, and they will always be able to focus on the road ahead of them.

This device will have two components: one part will detect hazards from behind using sensors fitted onto the bike itself, the other component will be an Attachable Haptic Feedback that the cyclist will wear which will have an array of motors that vibrates with varying frequency depending on the direction of the hazard and the distance of the hazard so the cyclist will know whether to swerve right or left to avoid the approaching vehicle.

1.2 High level Requirement

We decided that the vest idea might be too difficult to implement and uncomfortable for the user, especially with the electronics digging into the user's body, thus we will test different ways to integrate the electronics and the motors for the user to wear. The options we have are the belt or the wearable wrist watch that will hold all the electronics and motors. This will be generalized as an attachable haptic feedback(AHF). We called it the AHF because the product will be attached to the user's body in a way we find to be most aesthetically pleasing and ergonomic. Haptic means that the user will receive vibrations to receive information, and feedback is because the user will be given information based on sensors receiving information that is useful to the user.

- The device must be able to detect vehicles up to 4.m straight behind the cyclist and up to 4m on either side in order to warn the cyclist about potential object(s) behind.
- The ultrasonic sensors[2] will need to be able to communicate with the MCU to send signals to the motors via bluetooth with a few bytes of data within the Attachable Haptic Feedback. Must have short or no time delay in order to alert the bicyclist about vehicles from behind instantaneously.
- The vibrations in the Attachable Haptic Feedback that are caused by motors will need to vibrate at varying frequencies depending on how far away vehicles are behind the bicyclist, if the car is

further away, i.e 3-4m, the motors will have a low vibration frequency such as one vibration every second, from 1-3m, the motors will vibrate with more frequency at 2 vibrations every second, which warns the cyclist to maneuver away from that direction. The Bicyclist who will be wearing the Attachable Haptic Feedback will feel these vibrations for an extended period of time, indicating that there is a vehicle behind the user.

2. Design

2.1 Block Diagram

We plan to separate the system into two modules, the sensing module in hooked on a bicycle and has one bluetooth module, three-direction ultrasonic sensor pointing to three different directions, a power supply and a microcontroller. The Attachable Haptic Feedback part has vibration motors which indicate hazard from different directions to the cyclist. It also has a receiver to receive the bluetooth signal from the sensing module and a microcontroller to control the frequency of the vibrations of the motors, which indicate the direction and distance of the hazard from behind.

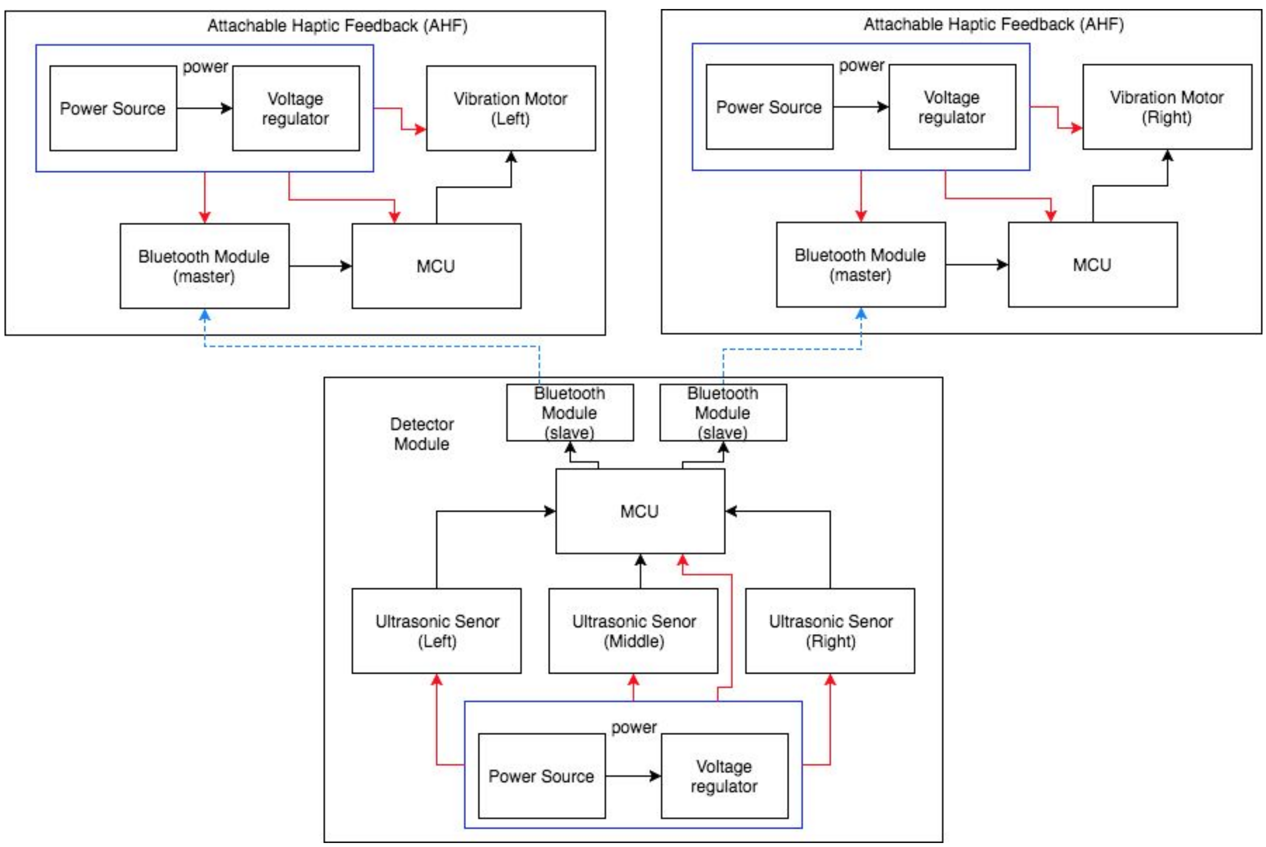

Fig.1 Block diagram

2.2 Physical Design

The physical design of the detector module is shown in Appendix B, Fig. 22 and Fig. 23. It can hold all the PCBs that allow the detector module to operate.Appendix B Fig. 22 shows how the module can be affixed onto the bike, the angle of the detector module can also be adjusted depending on the angle of the bike's seat pole. Appendix B Figures 24, and 25 show the AHF (Attachable Haptic Feedback) at different angles and how the case will be able to affix itself onto the user's forearm.

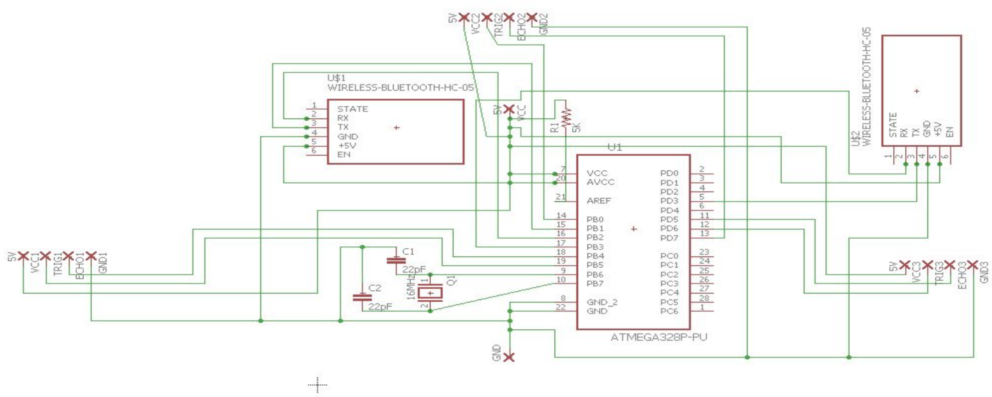

Fig. 2 Circuit schematic of MCU and bluetooth connection for detector module

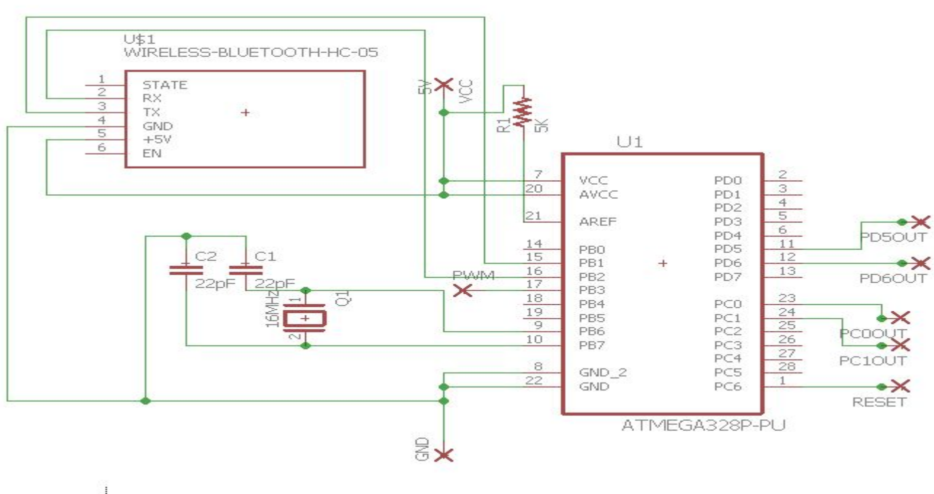

Fig. 3 Circuit schematic of MCU and bluetooth connection for AHF module

2.3 Lithium Ion Battery

The battery that we are using will be a lithium ion 9V battery, which will be the power supply for each of our integrated modules in the block diagram. Our requirements for the batteries are as follows:

- Batteries will need to work for more than 2 hours while constantly giving power to the motor subcircuit as the motors will use the most power.

- New batteries will need to be within 5% of 9V.

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

Fig. 4 Table for battery voltage while constantly supplying power to motor

Even after 2 hours of powering the motors, the battery voltage level did not drop significantly, thus, the battery will be able to operate the product for a long period of time efficiently. As seen from the table, the initial battery voltage was 8.956V which is within the 5% error allowed for the battery requirement.

2.4 Voltage regulator

Fig. 7 as seen below shows the circuit schematic of the zener diode voltage regulator. This voltage regulator relies on the zener diode breakdown voltage because it works with the 2N2222 BJT in order to create a voltage ceiling at the level we want, in our project, we needed ~5V voltage supply for all of our individual components, so we used a 9V battery as the power source, which had to go through the voltage regulator, resulting in a ~5V Vout that will be used by the other electrical components in the project.

The way that this voltage regulator works is by using a zener diode that is 0.7V higher than the Vout we need, which is 5.7V due to the fact that the BJT has a Vbe of 0.7V.

$$\text{Maximum } V_{OUT} = V_{zener} - V_{be} \qquad \textbf{\{2.41\}}$$

Therefore, with this formula, we can obtain Vout to be ~5V if we can get a zener diode with breakdown voltage of 5.7V, however, there were no 5.7V zener diodes available on the market and we were able to only obtain zener diodes with breakdown voltages of 5.6V, which resulted in a Vout of 4.9V, which is very close to our ideal voltage of 5V.

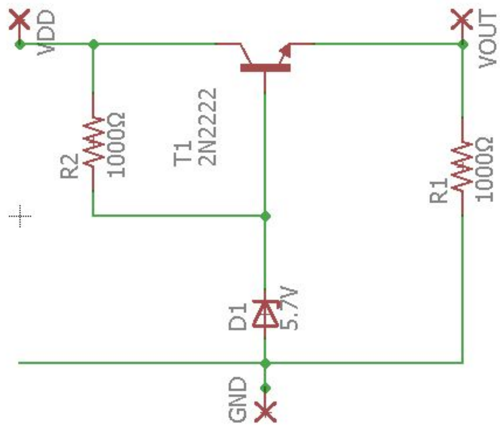

Fig.5 Ideal Voltage regulator circuit schematic

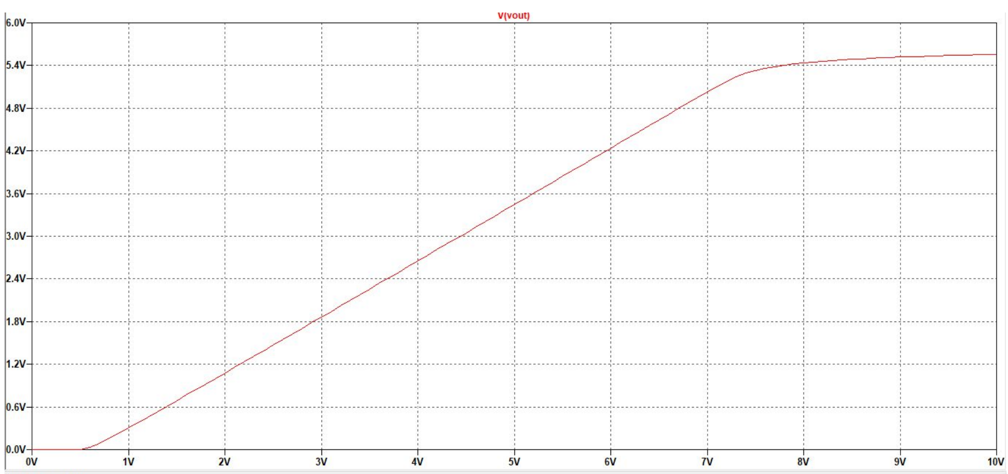

Fig. 6 Plot of Vout(vertical) vs Vin(horizontal)

The graph Fig. 8 shows an LTSpice simulation of a DC voltage sweep for Vin from 0 to 10V and Vout on the vertical axis. The voltage output plateaus at the maximum allowed voltage output of 5.5V which is the intended result for this circuit when the zener voltage breakdown is 6.2V. We used the model for a 6.2V zener diode because there were no 5.7V zener diode models in the LTSpice directory, thus a zener diode breakdown voltage of 6.2V will result in Vout max of 5.5V due to the equation:

$$\text{Maximum Vout} = \text{Vzener} - \text{Vbe} \qquad \textbf{\{2.41\}}$$

Where Vbe = 0.7V for the 2N2222 BJT.

2.5 Motor Sub-Circuit

The motor sub-circuit is responsible for controlling the motor's operation correctly and safely. According to the specification of the motor[3] being used in our design, the maximum current through the motor needs to be no larger than 80ma with 5V power supply. The circuit will also need to be able to correctly turn the motor on and off base on the signal coming out from the microcontroller. Due to the constraints mentioned above, we chose the 2N2222 BJT[1] as our key component. The circuit schematic was set up as the Fig. 7. A diode is placed in parallel with the motor to ensure that the current is flowing the correct way and make sure that the voltage across the motor is less than 5V. A resistor was placed at the Base terminal of the BJT to limit the current through the motor.

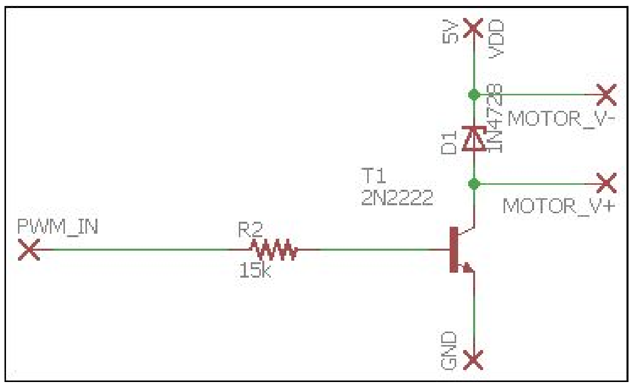

Fig. 7 Motor Sub-Circuit Schematic

As shown in Fig. 7, the circuit takes a PWM input signal from the microcontroller and a 5V voltage supply from external source. The diode and motor are connected in parallel between the power supply and the Collector terminal of the BJT. A BJT was chose due to the fact that the Collector current I_C is directly proportional to the Base current I_B and the relationship is given by the equation:

$$I_C = \beta I_B \quad \textbf{\{2.51\}}$$

By using the equation above a required current of 80mA and a β value of 200[1] we have to make make sure that the I_B is no larger than 0.4mA.

Since the highest voltage that the microcontroller can output is 5V, we assume a worst case scenario which is constant 5V for the signal V_{PWN_IN}. By using R_{BE} = 1kΩ and the following equation:

$$I_B = V_{PWM\_IN}/(R_2 + R_{BE}) \quad \textbf{\{2.52\}}$$

We have R_2 = 11.5kΩ for the Base resistance of our circuit.

After designing the circuit, we built the circuit on breadboard to get measurements that allow us to better modeling the components and verify our design. The measurement and final parameters are shown in Fig. 8. And we were able to make sure that I_C was less than 80mA.

B Value	R_{BE} Value	I_B	I_C	R_2
216	1.13kΩ	0.31mA	67mA	15kΩ

Fig. 8 Actual parameters of Motor Sub-Circuit

Fig. 8 illustrates our test results for the circuit controlling the motor to provide different vibration pattern and strength. Our circuit was able to successfully control the motor to provide different vibration pattern and strength based on different PWM frequencies and Duty Cycles.

Frequency	Strength(20% Duty Cycle)	Strength(50% Duty Cycle)	Strength(80% Duty Cycle)	Comment
1Hz	Low	Medium	High	ON/OFF of the motor become noticeable
10Hz	Low	Medium	High	Four distinct levels of vibration strength
1000Hz	Too weak	Less than Medium	High	levels of vibration strength become less distinct

Fig. 9 result and comment on different PWM settings

Due to our testing results shown in Fig. 9, we decided to control the duration of the motor being turned on instead of controlling the duty cycle of the PWM signal.

2.6 Ultrasonic sensors

Fig. 10 shows the ultrasonic sensors we are using (shown in Appendix B, Fig. 16), which has a Vcc voltage requirement of 5V, a max range of 4m and works by emitting a sound pulse from one lobe while receiving the sound signal from the other lobe. This will process will have a time delay which the sensor will use to calculate how far away the object is.

Fig. 10 HC-05 ultrasonic sensor

To test it, we connect it to arduino. This module should be able to get distance feedback from object(s) within the range specified and thus obtain the distance of the object from the sensor.

The Ultrasonic works well. When pointing the Ultrasonic sensor in one direction, and approaching it with constant speed, the distance can be correctly displayed on the serial output screen. The resolution is about 20cm, which is acceptable. Results from the Ultrasonic sensor when approaching:

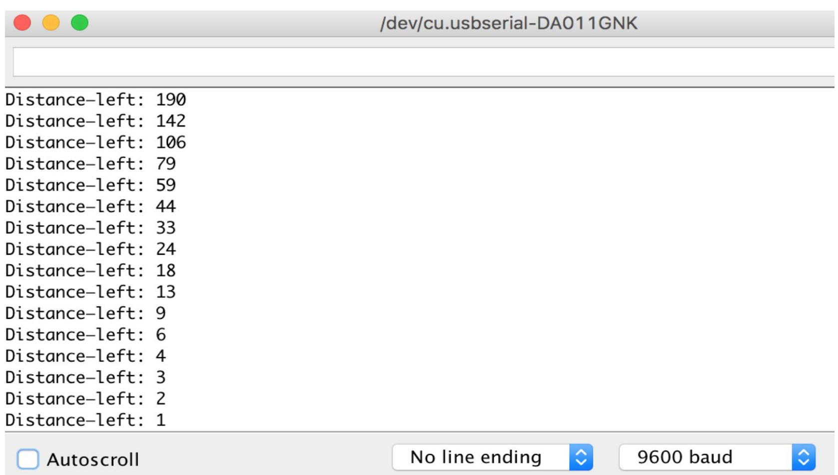

Fig. 11 The result of ultrasonic sensor when approaching it

2.7 Pseudocode with flowcharts

The program should always be running when the power is on. Therefore, we modified program logic and make it be an infinite loop. And it sends signal from slave to the master every 1500ms. After testing the software, we found that the best time to send the bitwise signal is 1500ms, otherwise the channel may be stuck and the receiving bits will be messed up.

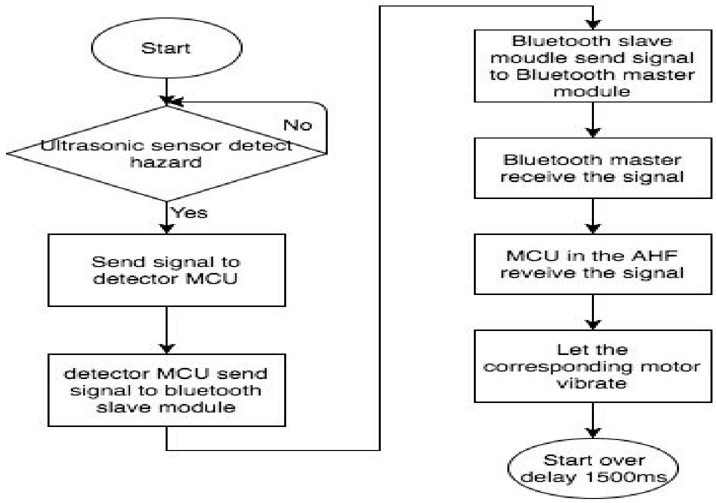

Fig. 12 Flowchart for control logic

2.8 Bluetooth

We are using HC-05 bluetooth modules. The bluetooth is responsible for communicating between the Attachable Haptic Feedback module and the detector module. There are two different configurations on bluetooth: master and slave. The bluetooth slave is used to send data collected from the ultrasonic sensor while the bluetooth master is used to receive data from slave.

In the detector, there will be two slave bluetooth modules which send signal to the two bluetooth master modules in the two Attachable Haptic Feedback modules. Therefore, there are two pairs of master-slave bluetooth pair. To test the pairing, we send a couple of signal generated from slave to master in Fig. 13, and the receiving data bits are shown in Fig.14

```
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);
```

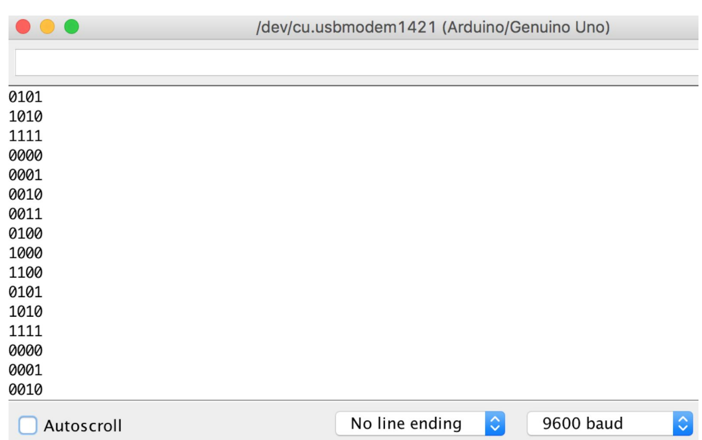

Fig. 13 The data bits sent from the slave

Fig. 14 The data bits received by the master

Moreover, we want to make sure that the latency in the bluetooth transmission is small and the data loss rate in our sending and receiving is low. To test the latency, we send 200 data packages and recode sending time as T1, receiving time as T2. Calculating the sending time as:

$$\text{Sending time} = T2 - T1 \qquad \textbf{\{2.81\}}$$

After getting the total sending time of 200 packages, we calculate the average sending time is 135ms and print it out on Fig. 15

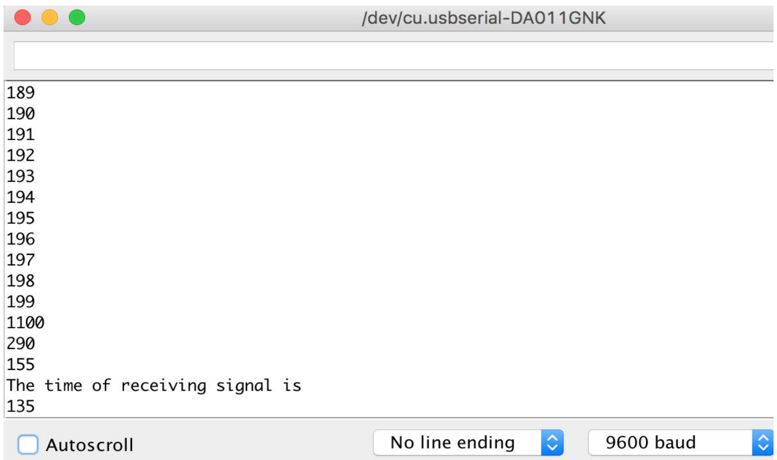

Fig. 15 Average time for master to receive data

To find the data loss rate, we send 1000 groups of data from the master and the slave module and calculate the loss as:

$$\text{Loss rate} = \text{Data packages not received} / \text{Total packages} * 100\% \qquad \textbf{\{2.82\}}$$

And the result of the data loss rate is shown in Fig. 16:

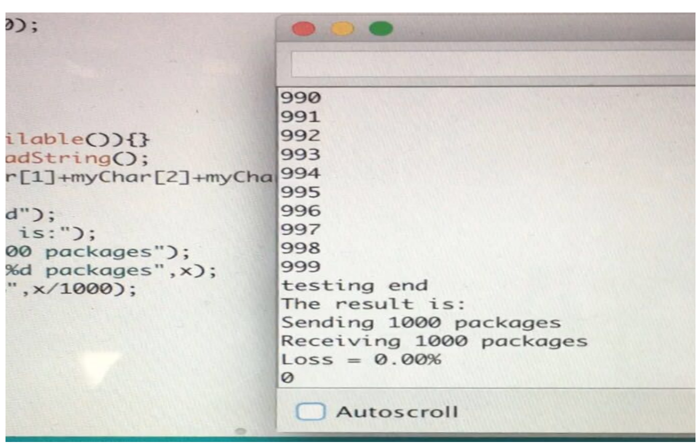

Fig. 16 Data loss rate in sending and receiving

2.9 Microcontroller

The microcontroller we are using is ATMega328P, which is compatible with Arduino. The advantage of using it is that we can easily modify our code by insert the microcontroller on Arduino board and upload the code. Fig. 17 is ATMega328P.

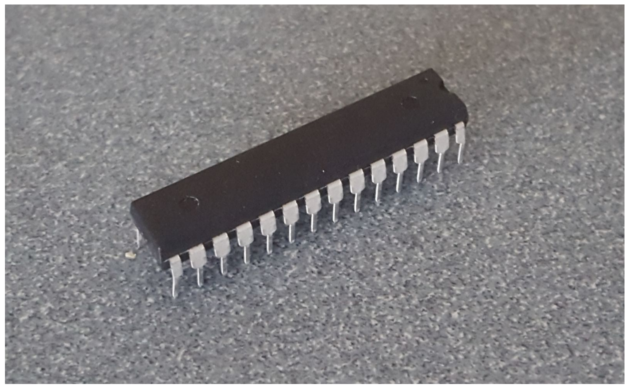

Fig. 17 ATMega328P

To make the digital pins output Pulse Width Modulation (PWM), we use two different methods: First method is use the built-in analogWrite() function. Since analogWrite has a scale of 0-255, where analogWrite(255) means a duty cycle of 100%, we test different values of PWM in Fig. 18 and Fig. 19:

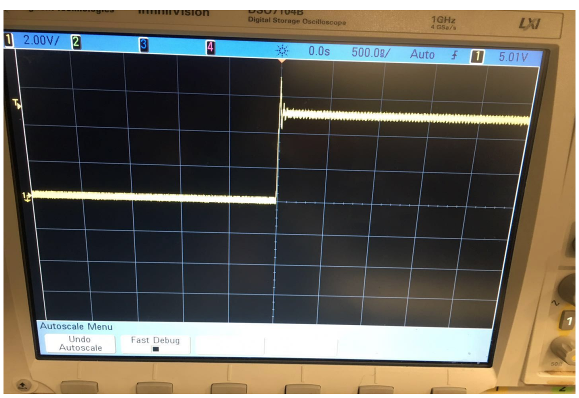

Fig. 18 AnalogWrite(128) 50% duty cycle

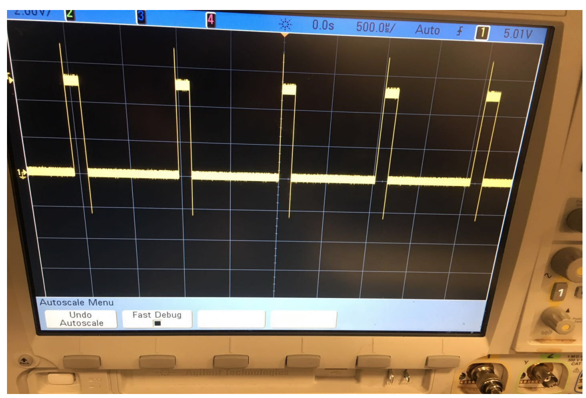

Fig. 19 analogWrite(32) 12.5% duty cycle

The second method is to use delay() and digitalWrite() function to create a delay when pull the voltage HIGH and LOW. We test this method by delaying HIGH for 3ms, LOW for 9ms to create an duty cycle of 25% in Fig. 20:

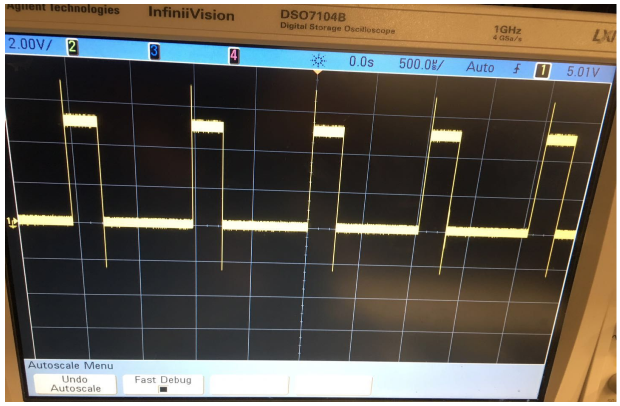

Fig. 20 25% duty cycle using delay method

3. Costs

3.1 Parts

Description	Quantity	Cost/Unit	Total Cost
ATmega328P Microcontroller	3	$7.78	$23.34
HC- SR04 Ultrasonic Ranging module (4m)	3	$3.95	$11.85
HC-05 Bluetooth Module	4	$8.59	$34.36
VPM2 Motor	2	$3.95	$7.90
Lithium Battery 9V	3	$6.64	$19.92
5.7V Zener Diode	5	$0.32	$1.60
PCB	11	$0.42	$4.58
2N2222	5	$0.53	$2.65
Total			$106.20

3.2 Labor

Name	Hourly Rate	Hours	Total	Total with expense multiplier (x2.5)
Hu Yanda	$30	300	$9000	$22,500

Gu Zhaoxin	$30	300	$9000	$22,500
Jiayi Ke	$30	300	$9000	$22,500
Total			$27,000	$67,500

Total Cost

Total cost comes out to be $67,606.2.

4. Conclusion

4.1 Accomplishments

In conclusion, all the individual components in our project worked out fine because they met our requirements, up to the point of integration where there were issues with the ultrasonic sensor range detection when they connected to the microcontroller PCBs as shown in Fig. 2. Furthermore, the bluetooth modules had a recurring problem of glitching while attempting to pair the slave and master bluetooth modules which ultimately prevented the final product from functioning.

4.2 Future Work

- Use high end ultrasonic sensors to improve the resolution of the distance detection and also 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

References:

[1] 2N2222A BJT, datasheet, On Semiconductor. Semiconductor Components Industries, LLC, 2013. Available at:
http://web.mit.edu/6.101/www/reference/2N2222A.pdf

[2] HC- SR04 Ultrasonic sensors, datasheet, ElecFreaks. Available at:
http://www.mouser.com/ds/2/813/HCSR04-1022824.pdf [Accessed October 2017- December 2017]

[3] vibration motor, datasheet, SolarBotics. Available at:

http://www.robotshop.com/media/files/pdf/datasheet-vpm2.pdf [Accessed October 2017 - December 2017]

[4] Time Required for a third degree burn to occur. Available at:

http://chfs.ky.gov/NR/rdonlyres/2CFE26ED-59A5-4F60-91A7-E2B5C0E9D000/0/BurnsintheElderly091213.pdf

[5] Robotics society of Southern California, US sensor pros and cons. Available at:

http://web.csulb.edu/~wmartinz/rssc/content/pros-and-cons-range-sensors.html

[6} HC-05 bluetooth datasheet. Available at:

https://cdn.makezine.com/uploads/2014/03/hc_hc-05-user-instructions-bluetooth.pdf
[Accessed October 2017- December2017]

Appendix A.

Power source[6 points]:

Module	Requirements	Verification - **success**
Power source-sensor case	1. Battery supply +9V +/-5% power 2. Lifetime for battery should last for at least 2 hours if used non-stop, has shelf life of 10-20years if not used according to product's information on package.	1. Multimeter will be used to check if the voltage and current outputs are equal to specified values. Use an oscilloscope to check if the voltage signal is steady. 2. Test battery life, battery life will be calculated based on the current rating in milliAmpere per Hour. The battery life or capacity can be calculated from the input current rating of the battery and the load current of the circuit. Battery Life = Battery Capacity in Milliamps per hour / Load Current in Milliamps * 0.70. *The factor of 0.7 makes allowances for external factors which can affect battery life.
Power source-AHF	1. Battery supply +9V +/-5% power 2. Lifetime for battery should last for at least 2 hours if used non-stop, has shelf life of 10-20 years if not used according to product's information on package.	1. Multimeter will be used to check if the voltage and current outputs are equal to specified values. Use an oscilloscope to check if the voltage signal is steady. 2. Test battery life, battery life will be calculated based on the current rating in milliAmpere per Hour. The battery life or capacity can be calculated from the input current rating of the battery and the load current of the circuit. Battery Life = Battery Capacity in Milliamps per hour / Load Current in Milliamps * 0.70. *The factor of 0.7 makes allowances for external factors which can affect battery life.

Power Regulator[6 points]:

Module	Requirements	Verification - **success**

Voltage regulator	1. Must regulate the input voltage so that the output voltage is at 5V to supply power to other components.	1. Use a multimeter and oscilloscope across the voltage regulator to measure potential difference across Vout, this will check if Vout is within acceptable range.

Vibration motors[6 points]:

Module	Requirements	Verification - **success**
Vibration Motors	1. Vibration motors' power supply is: 3.3V+/- 5%, no larger than 80 mA current, with 10% error**.	1. Connect to oscilloscope to verify voltage drawn by motor is constant and at required voltage of 3.3V. Test motors with the PWM to check compatibility.
Vibration Motors Sub-Circuit	1. Have different operating duration depending on the PWM, so when the PWM has high output, the motor vibrates, and the motor stops when the PWM outputs a low signal. 2. Current at Collector side of BJT no larger than 80mA.	1. Turn PWM on and off using a power supply and the motor should turn on and off accordingly. 2. Connect circuit to multimeter to verify current amplitude at the collector side of the BJT.

Bluetooth Modules[10 points]:

Module	Requirements	Verification - **success**
AHF Bluetooth receiver (master module)	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.	1. Receive 1000 groups of data from the master, and count if the data group received is larger than 990. 2. Move the master module away and to check whether at 100 cm, the slave can still get the signal from master. 3. Use the timing function in software and once the slave module sends out a signal, save time as T1. Once the slave receives the signal, save the time as T2. Use T2-T1 to see if the delay is in the range.
Sensor Case Bluetooth Emitter (slave module)	1. Can always send coherent signal (>99% accuracy, which is <1% loss) to the bluetooth slave module 2. Distinguish the out-going signal of which in direction the hazard is coming from the response of ultrasonic sensor 3. Transmit signals within 100cm.	1. Send 1000 groups of data to slave, and count if the data group received is larger than 990. 2. Have different kinds of signals to separate the hazard from left, right and close, far. Print out different conditions by different states. Since each sensor can have 4 states, we have 3 sensors, total is 12 different signal possibilities. Then we send the out-going signal by according to theses states. 3. Move the slave module further to check if at 100 cm, the master can still get the signal sent from the slave module.

Ultrasonic Sensors [10 points]:

Module	Requirements	Verification - **successs**
Ultrasonic Sensors	1. Can detect when an object enters its field of sonar sensing and produce the distance of object(s) from 0.5-2.5m with a resolution of at least 20cm onto a display to show that the sensor works. 2. Voltage requirement:5V 3. Sensor must send and receive signals every 100 millisecond. 4. Sensors will have a 30 degrees of detection, sensors will be more accurate when detecting objects directly to the front.	1. Prototype: Connect to arduino. This module(s) should be able to get distance feedback from object(s) within the range specified and thus obtain the distance of the object from the sensor. 2. Use multimeter to gauge the voltage needed to operate the sensors, then do final Stage: Connect to MCU and redo verification in the prototype stage. 3. Use serialPrint() function to print out the different data get from the sensor every 100 millisecond. 4. Place objects at different angles in front of the sensors while also changing the distance from the sensor and use results to construct a table.

Microcontroller [12 points]:

Module	Requirements	Verification - **success**
MCU in AHF	1. When connected to PCB, microcontroller should control Bluetooth module (slave) to receive 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.	1. Set up a MCU-master and MCU-slave pair and try to send data between them to see if the master can receive the data. 2. Program the output of those pins and plug the oscilloscope into those pins to check if the duty cycle will change. 3. Program the output of those pins and plug the oscilloscope into those pins to check if the duty cycle will change.
MCU in Sensor case	1. When connected to PCB, microcontroller should control Bluetooth module (master) to transmit data. 2. For the software RX, TX pin, the data(mentioned in section 2.33) presenting how far the object is can interchange with MCU. The ultrasonic sensor is going to automatically give a HIGH pulse whose duration is the time	1. Set up a MCU-master and MCU-slave pair and try to send data between them to see if the master can receive the data. 2. Program the MCU to test the RX, TX with one ultrasonic sensor, connecting the PWM output with a motor circuit and to see if something comes closer to the sensor, motor circuit will vibrate more intensely. And to check if the distance formula is correct, make an object stays

	(in microseconds) from the sending of the ping to the reception of its echo off of an object. Then we are going to convert the time into a distance cm = (duration/2) / 29.1. We are using it through a current library, not implementing by ourselves.	still and determine if the distance is equal to the actual distance.

Fig. 21 Requirements and Verification table

Appendix B

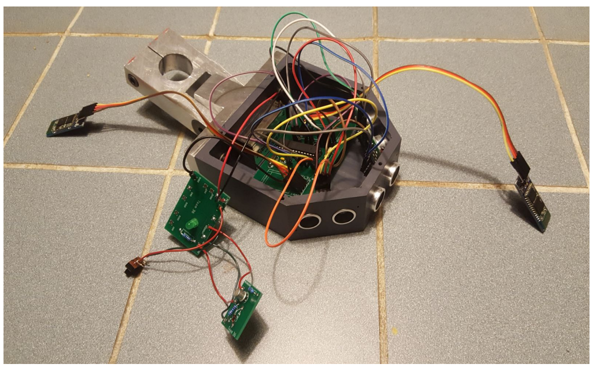

Fig. 22 Detector module without cover

Fig. 23 Detector module installed onto Jiayi Ke's bike

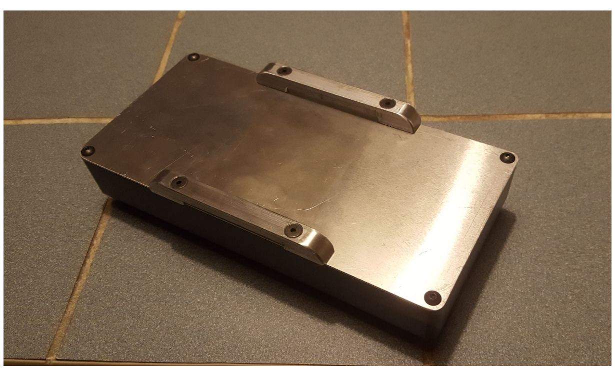

Fig.24 AHF case underside

Fig.25 AHF case with velcro straps