Final Report for ECE 445
Spring 2019

Project Title : Safe-Walk Hat for people with visually impaired

Team # 72
Woo Young Choi (wchoi19) / Yong Jun Lee (ylee168) / Tae Hwa Kim(tkim142)
Assigned TA : Dongwei shi
Assigned Professor : Michael L Oelze
Submission date : 05/01/2019

Table of Contents

1. Introduction

There are several enhancement methods for visual impaired people such as a cane and a guided dog. However, we wanted to design and build an effective device that could alert to the user about the potential risk and help them to avoid any risks. Our team designed and built an enhancement device called, “Safe Walk Hat” and it has 6 different modules and an Android application. There are Doppler Sensor module, microcontroller module, bluetooth module, Audio with SD card reader module, 5V Buck boost module, and Battery Management Module. After building all the individual modules, we combined them all together and we were able to get successful result which we expected.

1.1 Objective

There is a high chance that people with vision impairment or blindness would get involved with traffic accidents including collision accidents with bicycles. According to a research, Normally Sighted, Visually Impaired, and Blind Pedestrians Accurate and Reliable at Making Street Crossing Decisions written by Shirin E. Hassan, there are two decision variables to consider in order to cross a street safely; the time that it will take them to cross the street; the time available before the next vehicle reaches them[1]. Compared to people with normal vision, visually impaired people or people with blind disability recognized 12% fewer crossable gaps, and made 23% more errors by estimating a gap as crossable[2].

The chance that people with blind disability get involved with accidents has been increased since electric automotives, which does not make any noise even on the road, started to draw people’s attention because global warming is one of biggest problems needed to be solved around the world. People with vision trouble will have more difficult time to safely cross a street as the number of electric automotive keeps increasing. For example, the number of electric automotive increased from 160,000 in 2016 to 280,000 in 2017 in the united states, and is estimated to keep increasing in the future[3]. Furthermore, the number of people using bicycles which poses a danger to people with blind disability in the U.S has been also increasing from 36 million to approximately 48 million since 2006[4].

1.2 Background

There are several ways to support people with vision trouble walking; First, a cane; Second, a guide dog and a cane. Of course, those two methods are also effective to walk around the outside, but the problem is that both of methods are just to detect unknown objects near a person who uses them. Both methods do not have an ability to prevent accidents from happening. There are many products of comfortable canes or of training a service dog, but there is no such products on the market that protect people with vision trouble from accidents and allow visually impaired people to cross a street safely and to walk around without exposing themselves to a danger. The device of this project is designed and implemented in order to protect visually impaired people from car accidents and pedestrian injury. In terms of hardware, the design of the device is a hat integrated with a microcontroller Atmega 2560, six doppler radar sensors HB-100, a SD card reader, a Bluetooth module of HC-05, two bone conduction transducers, and a SD card. And the device is divided into 5 different modules: Power supply module, Sensor module, Speaker module, Bluetooth module, and Microcontroller module. The power supply module includes battery management device, Li-po Battery, and 5V buck-boost converter. This module provides enough power to turn the microcontroller on more than 6 hours. The next is the sensor module. The sensor module has six different doppler radar sensors and six high gain amplifiers. The six different doppler radar sensors facing six different direction in order to cover 360 degree detects any moving object in range of 20 meters, and notifies the user if the object is approaching toward the user over certain speed limit by sending a signal to the microcontroller. The amplifiers are to amplify the output of each doppler radar sensor since reflected microwaves are very weak to analyze. The third module is the speaker module, which includes a audio amplifier, the SD card reader, the SD card and the bone conduction speaker. The purpose of the SD card reader and the SD card is to store WAV files of voices of 8 different directions such as east, west, north, south, northeast, northwest, southeast, and southwest. Whenever the microwave sensors send a signal to notify the user, the microcontroller sends a signal, that indicates the direction of an approaching object, to the SD card reader; the SD card reader sends a corresponding recorded direction file to the speaker; the speaker plays the recorded direction sound file. Next one is the Bluetooth module. The reason that the Bluetooth module is integrated into the device is one more functionality. This functionality is for the safety of the user. If the user is in any emergency situation, the user pushes a button located back of the hat. When the button is pushed, the microcontroller receives a signal to send a signal to the Bluetooth module.

The Bluetooth module send a signal to the user's android phone so that the phone sends GPS location of the user to a designated emergency center number. Lastly, there is the microcontroller module. As it is mentioned above, the microcontroller is Atmega 2560, programmed with our arduino code. The microcontroller receives the data from the sensor module, analyzes the data, and calculates the speed and the direction of an approaching object. And If the calculated speed of an approaching object is above certain speed, the microcontroller also reads the .wav files from the SD card, and sends audio files to speaker, so that the speaker module plays the sound indicating the direction of an approaching object. Furthermore, when the emergency button located back of the hat is pushed, the microcontroller receives a signal, and sends that signal to the Bluetooth module.

1.3 High-Level Requirement List

- The range of detection component, including doppler radar sensors will be approximately 20 meters
- Six of the doppler radar sensors must be able to detect an object within 360 degree
- The doppler radar sensors must be able to detect a speed of an object from 1.4 km/s to at least 60 km/h
- The bluetooth module can send data to cell phone by pressing a button.
- The battery must be able to provide power for more six hours.
- The device can read sound data from a SD card and play them through two bone conduction speaker.

2. Design

2.1 Block Diagram

The previous design included not only six doppler radar sensors but also six ultrasonic sensors. The idea behind six ultrasonic sensors is that the device detects any moving object near the device, and decides whether the object is approaching toward the device using the ultrasonic sensors. And then the microwave sensors measure the speed of the object, and determines to notify the user if the object is approaching toward the user over certain speed. The problem with this design is the range of detection. The ultrasonic sensor, HC-SR04, can only detect up to 4 meters. The distance of 4 meters does not provide enough time for the user to dodge an approaching object. Therefore, after thorough research and consideration, the revised design of the device has only six doppler radar sensors. Using the frequency of reflected microwave, the device decides whether an object is moving toward the device, and notifies the user if the object is moving over certain speed. Without ultrasonic sensors, the device is able to detect up to 20 meters [3]. In other words, compared to the device with the previous design, the device alerts the user ahead of time so that the user has enough time to avoid an approaching object such as a car and a bicycle.

There is another change made in the design. The first design had a module to check remaining capacity of the battery. If a user push a button, then the module sends a signal to the microcontroller module to notify remaining capacity of the battery. Instead of that module, the revised design has a different functionality. The newly added functionality is to connect the device to any android phones using the Bluetooth module through serial port. If the user presses a button located back of the hat in case of emergency, the device sends a signal to a connected android phone, and the phone sends a GPS location of the user to someone registered. In order to add this functionality, the button has been added to the revised design. Moreover, we decided to keep the battery management module to alert the user if the remain capacity is lower than 20% through the speaker.

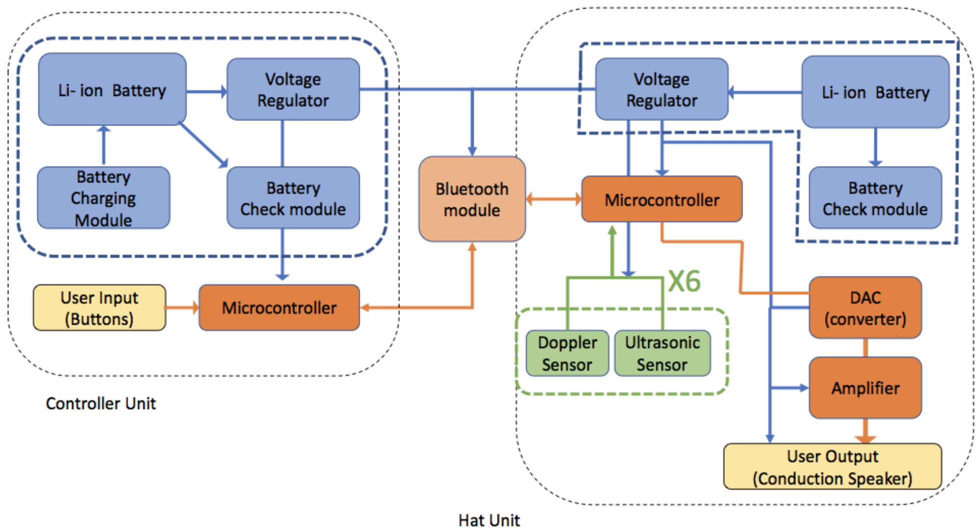


Figure 1. First version of Block diagram

The last change that was made on the design is the microcontroller. At first, ESP 32 was chosen for a microcontroller of the device. However, later on, ESP 32 was replaced with Atmega 2560 due to the number of Pulse-width modulation(PWM) pins.. Atmega 2560 has more PWM pins than ESP 32. Also, the device needs to support not only six sensors but also the SD card reader, and the user input. The Atmega2560 has enough GPIOs for our project.

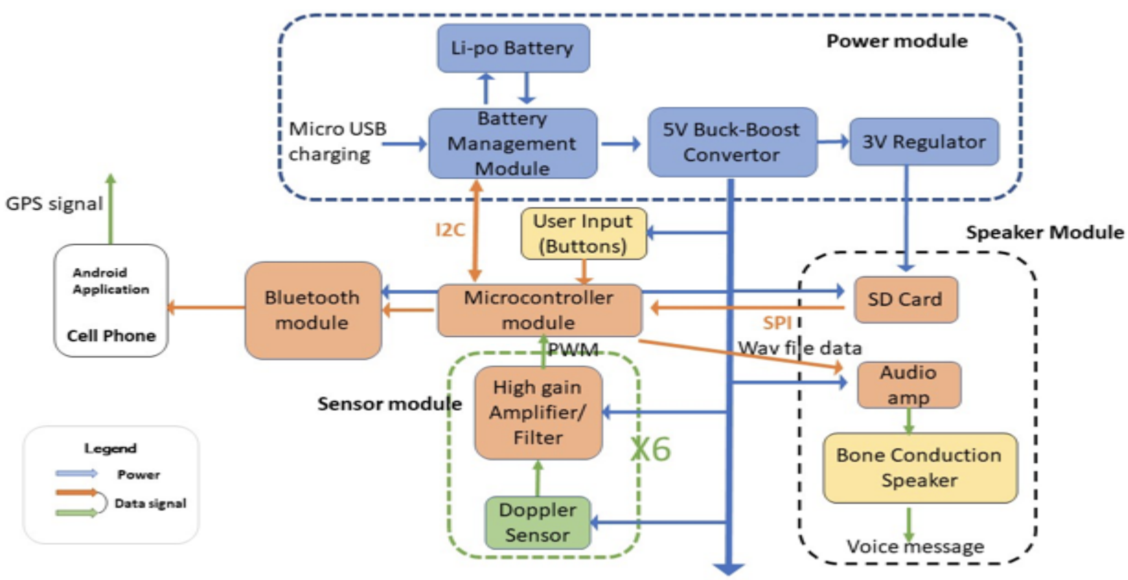


Figure 2. Final version of Block diagram of the product

2.2 Physical Design

As can be seen from the Figure 3, 6 doppler sensors are aligned around the hat to cover 360° detection. The button and the battery pack which includes Li-po battery and the battery management PCB are attached to back of the hat. Other PCBs, main PCB and sensor amplifier PCB, are attached inside of the hat.

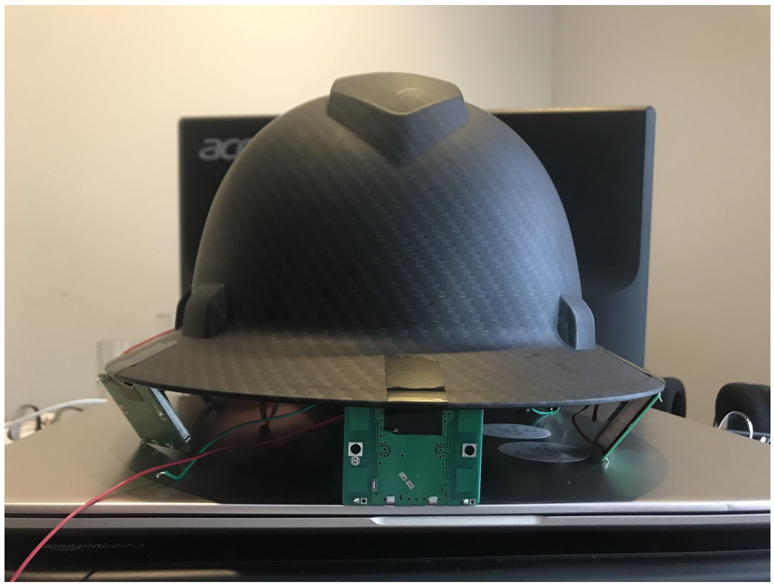


Figure 3. Front View of the Device

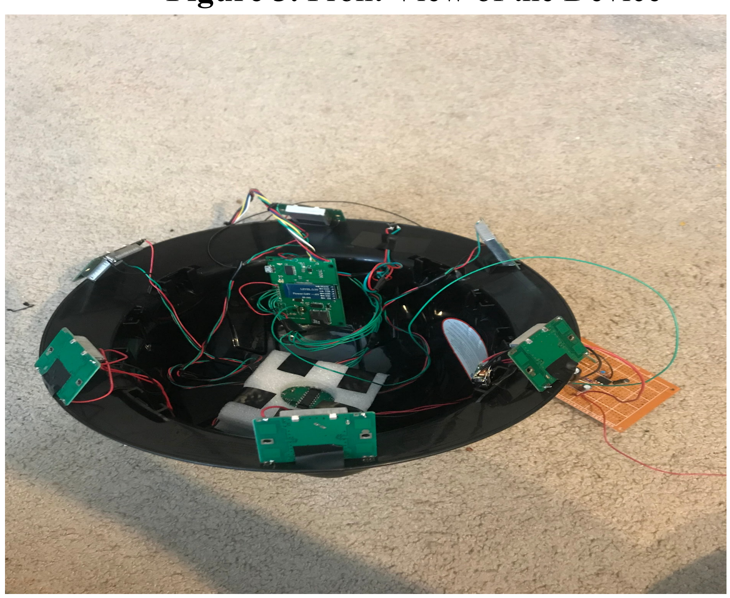


Figure 4. Inside View of the Device

2.3 Design Details

- Microcontroller

  - There are several reasons of choosing Atmega2560 for the microcontroller of this project . Firstly, this microcontroller has multiple number of GPIOs. For our project, the microcontroller must be able to support 8 PWM pins for six doppler radar sensors and two speakers, 1 input pin for the button and 1 output pin for the Bluetooth module. Furthermore, Atmega 2560 also supports digital-to-analog conversion. DAC which is required to play an audio file stored in SD card on a bone conduction speaker.

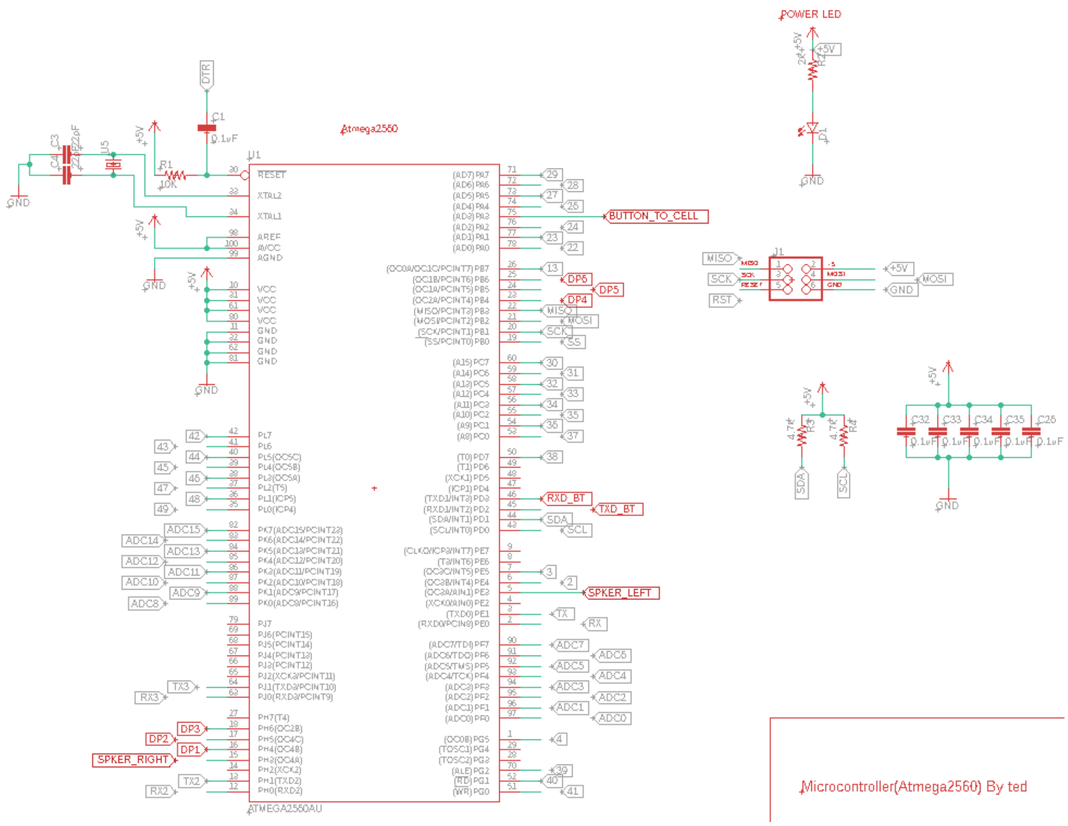


Figure 5. Schematic of Atmega 2560

FT232R is used to program the mcu with micro usb UART. Also, there is in circuit serial programming port on the main PCB to boot load the microcontroller. After soldered, we were able to program the microcontroller with micro USB successfully.

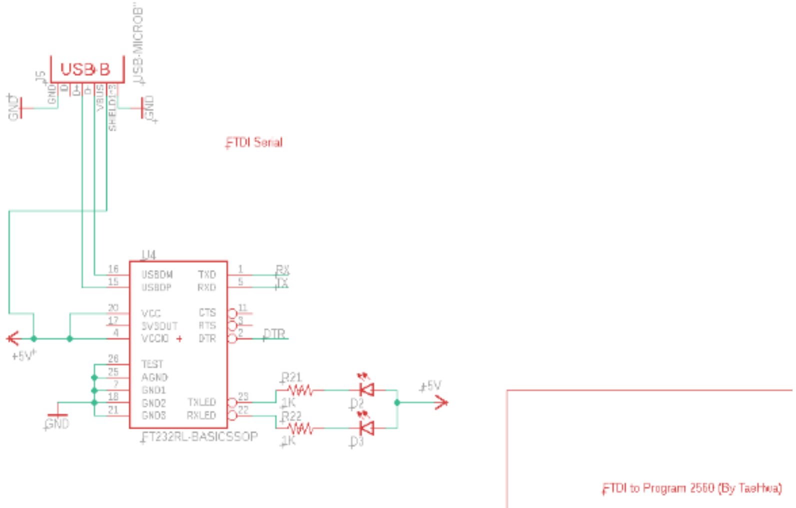


Figure 6. FTDI and Micro USB UART

- Doppler Radar Sensor Module

    - The sensor module is composed of two main parts: doppler radar sensors, and amplifiers. In other words, there are two main parts that needs to be considered for the design. First important design consideration for the sensor module is doppler radar sensor. Because the main functionality of our project is to detect an approaching object and to alert a user before any collision accident happens, the device must be integrated with a sensor with long detection range. That is why HB-100 is chosen, since the detection range of HB-100 is approximately 20 meters. Secondly, because the output of the sensor is couple of microvolts which is extremely small to analyze, the sensor module needs high gain amplifier. Therefore, we built an amplifier with gain around 12100 using two gain stages amplifier; the gain from the first stage is 100, and the gain from the second stage is 120.
    - As can be seen from the Figure 8. The actual gain was around 12100 and the pulse forms square wave as we wanted.

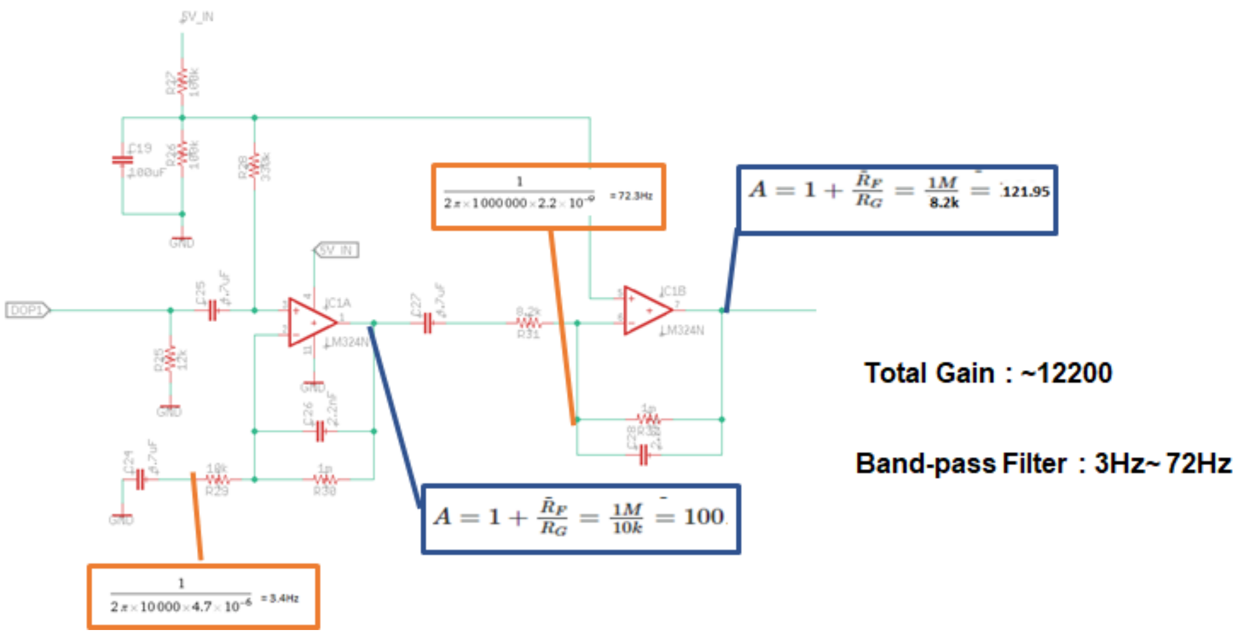


Figure 7. Schematic of doppler radar sensor amplifier

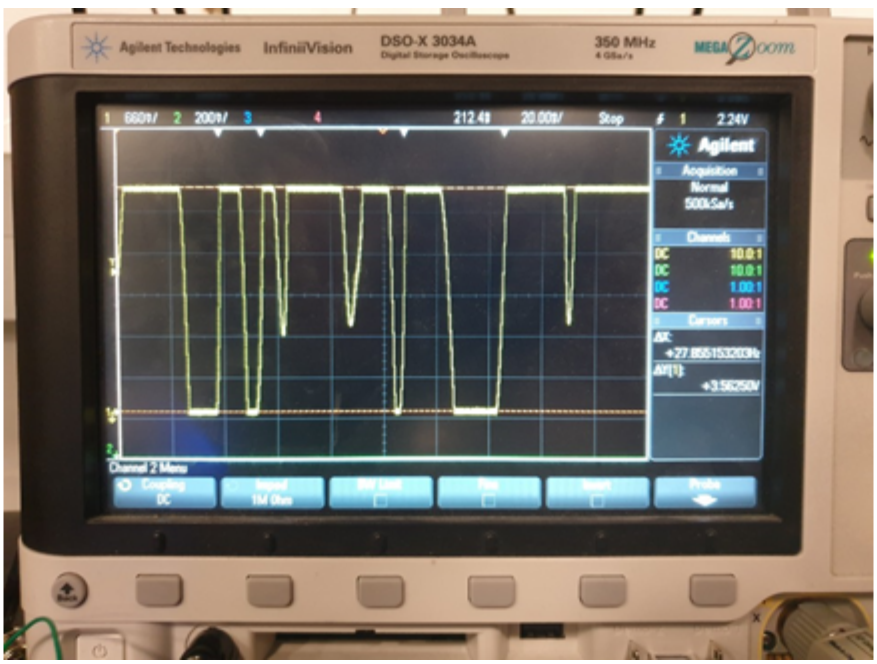


Figure 8. Ouput from the Sensor Amplifier(Moving)

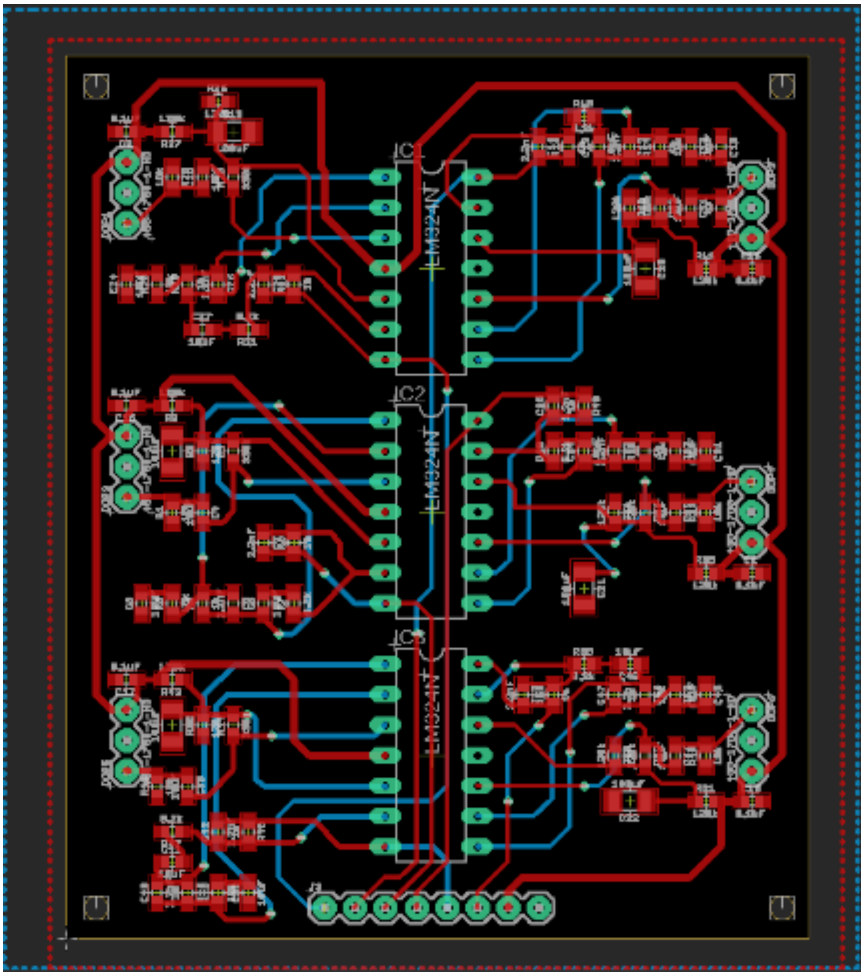


Figure 9. Sensor Amplifier PCB

- Power Management Module

  - The power management module is implemented with 3.7 V 2500mAh Li-po battery, BQ27441, BQ24705, and other peripherals such as capacitors and resistors. Firstly, there are two reasons that our group chose Li-po battery for the system. First reason is because of operating time of the project. We decided that the project needs to operate at least 6 hours. And Li-po battery, which has relatively large capacity compared to other types of battery such as Li-ion battery, can support the our project system approximately 6 hours. Second reason is that Li-po battery is rechargeable. Furthermore, there are two ICs for the power management module: BQ27441 and BQ24705. BQ27441 and BQ24705 are required in order to charge the battery in certain current and estimate the remaining capacity of the battery. Both ICs have wide voltage operating range, and both of them are programmable.

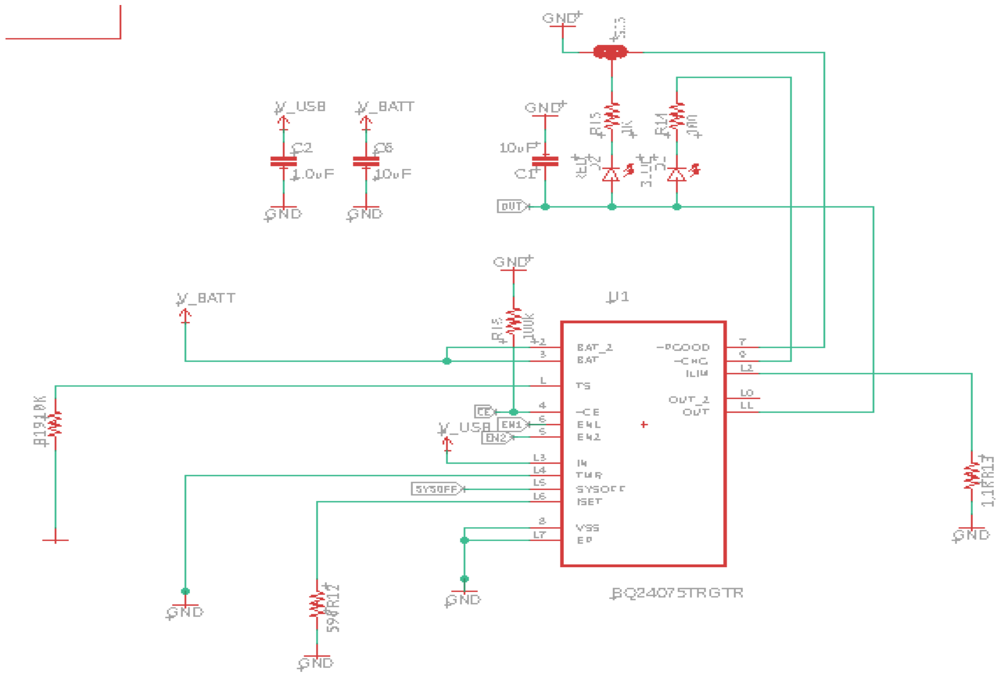


Figure 10. Schematic of BQ24075

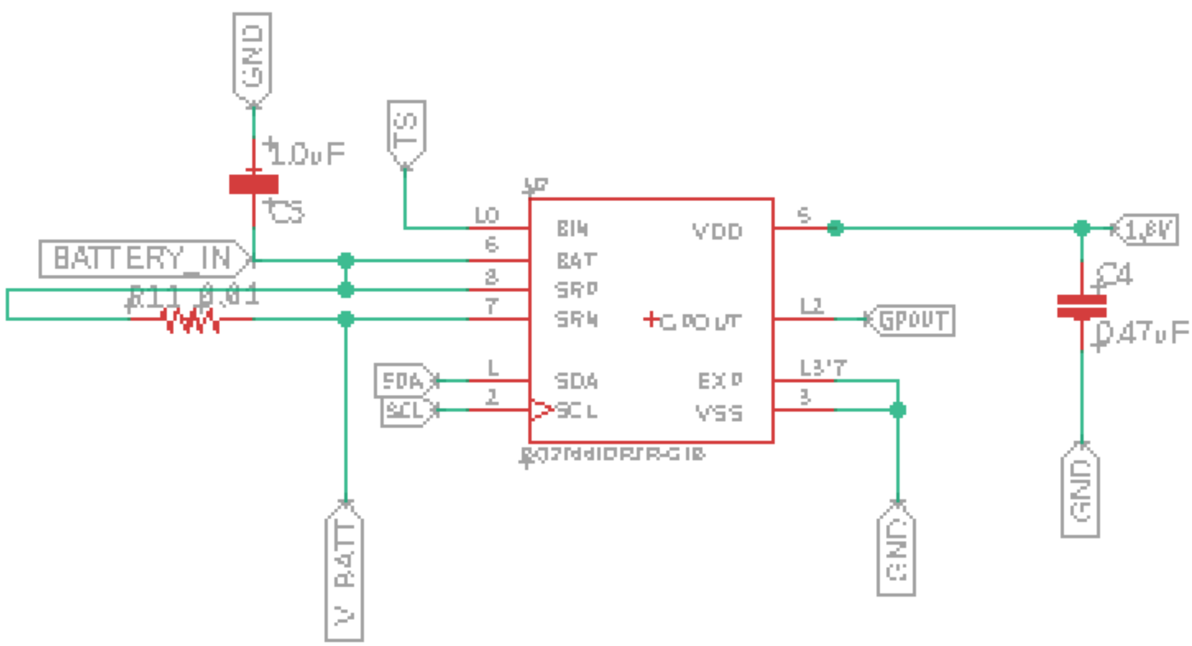
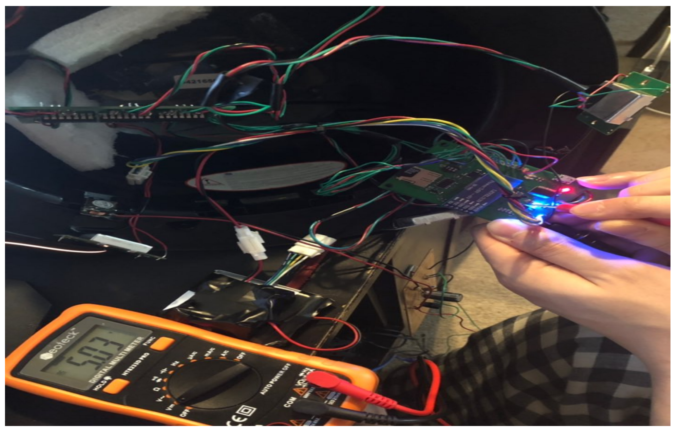
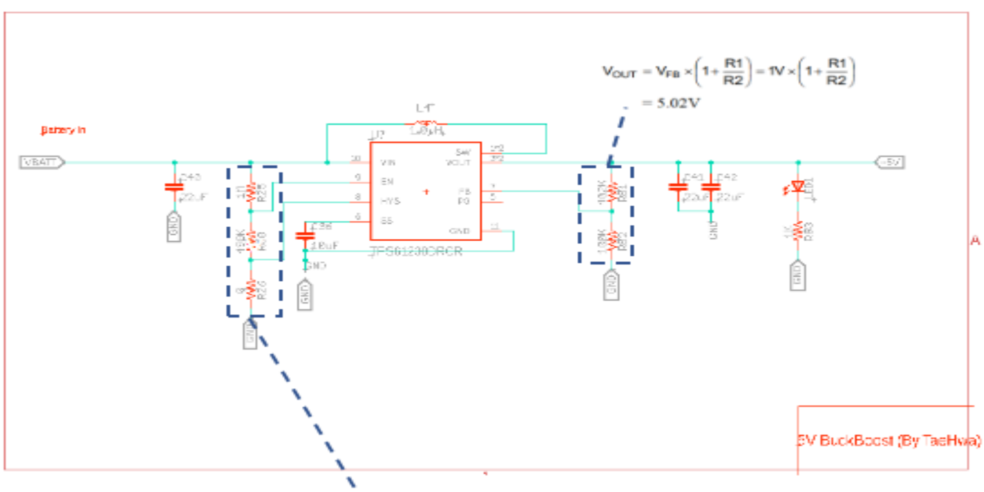


Figure 11. Schematic of BQ27441

• 5V Buck Boost Voltage Convertor Module

- We designed this 5V Buck Boost convertor module, because we want to boost 3.7V from the Li-po Battery to 5V which is our system voltage. The value of the resistors are chosen to make it operate from 2.4 V which is reasonable for our 3.7V battery, and the output range to be 5V which is our system voltage. TPS61230 is chosen because it is small so that we can save much space, and allowed output current is 2.1A which is good for our system. And it provides constant 5V with 96% high efficiency.





$$V_{IN\_OFF} = V_{TH\_EN\_OFF} \times \left(1 + \frac{R25}{R80 + R26}\right) = 1.14V \times \left(1 + \frac{1M}{499k + 402k}\right)$$

$$= 2.41V$$

Figure 12. Output Test Picture and Schematic of Buck-Boost converter

- Bluetooth Module
  - For the Bluetooth module, our group used a bluetooth module, HC-05, on the market. This module operates at DC 5V which is our system voltage, and consumes low current. Most important thing is it can operate as both slave and master mode so that we can send data to connected cell phone via serial communication with the microcontroller. Moreover, this module is cheap and easy to use. From the reasons, we decided to use HC-05.
  - On the Main PCB, there is 6-hole connection port for the bluetooth module. The RXD pin of the bluetooth module is connected to TXD pin of the microcontroller, and the TXD pin of the bluetooth module is connected to RXD pin of the microcontroller so that they can communicate bidirectionally.

- Audio with SD card Reader Module
  - The device detects any approaching object with speed higher than a limit that we programmed, and notifies a direction of the approaching object through a speaker. In order to play an audio file indicating the direction of the object, the device needs to store 6 different audio files corresponding to 6 different direction. Therefore, the SD card reader and SD card are included in the design, because we wanted to alert an user with human voice sound. The TXB0104 in the SD card reader module is the 4-bit Bidirectional translator which makes SPI communication between the microcontroller and the SD card. The AP2112K is a simple 3.3V regulator. Since the translator needs 3.3V on the SD card side and 5V on the MCU side, we decided to have AP2112K. Also to play the audio file on the speaker, audio files converted from digital to analog need to be amplified so that those audio files are audible to the user. After researching and measuring the output of converted audio file, we decided to amplify the audio file with gain 8. That is why we chose MAX98306. This IC operates from 2.6V to 5.5V, and draws low current. And most importantly, the maximum gain of this is around 8, which is

18 db.

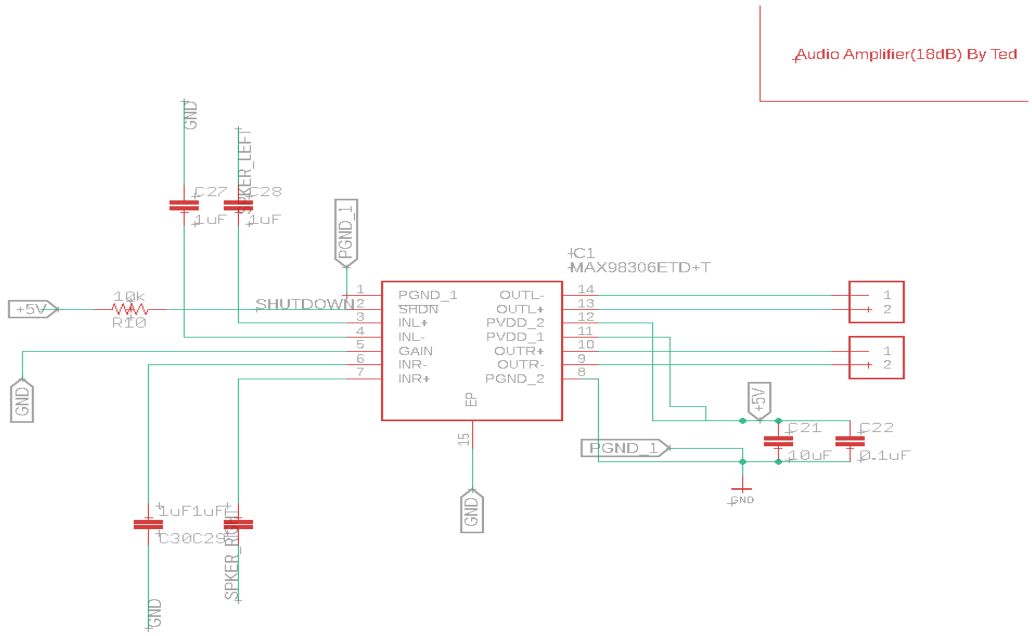


Figure 13. Schematic of MAX98306

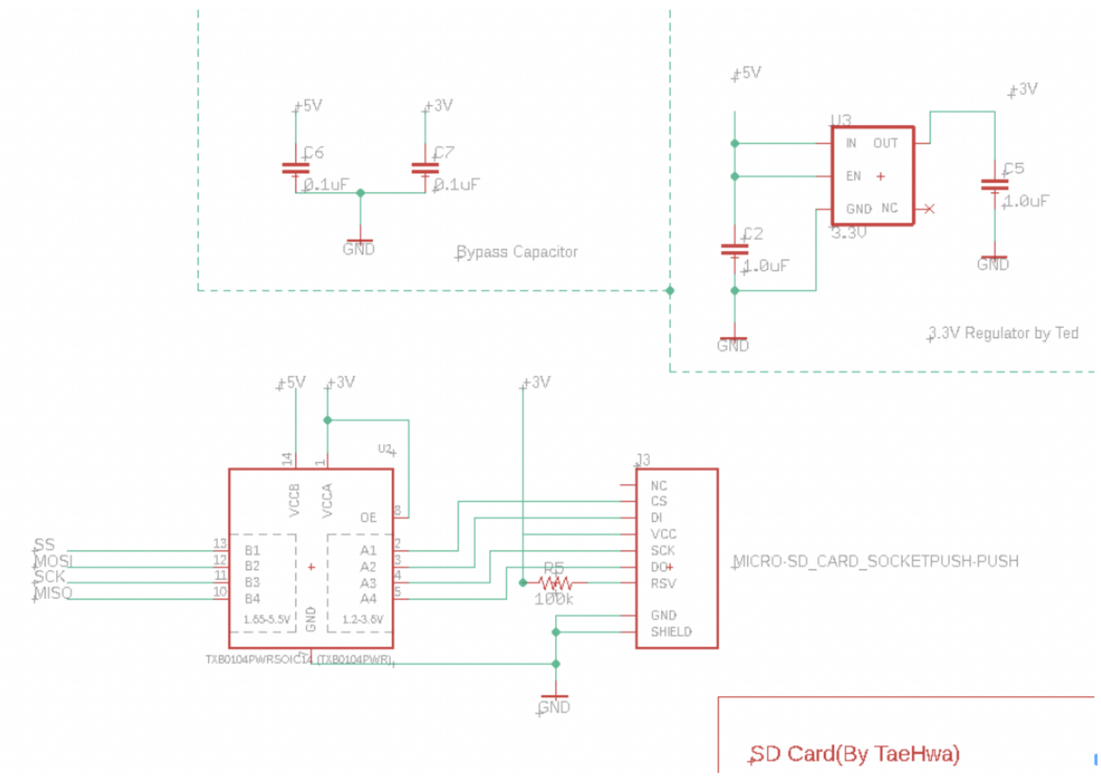


Figure 14. Schematic of SD card reader

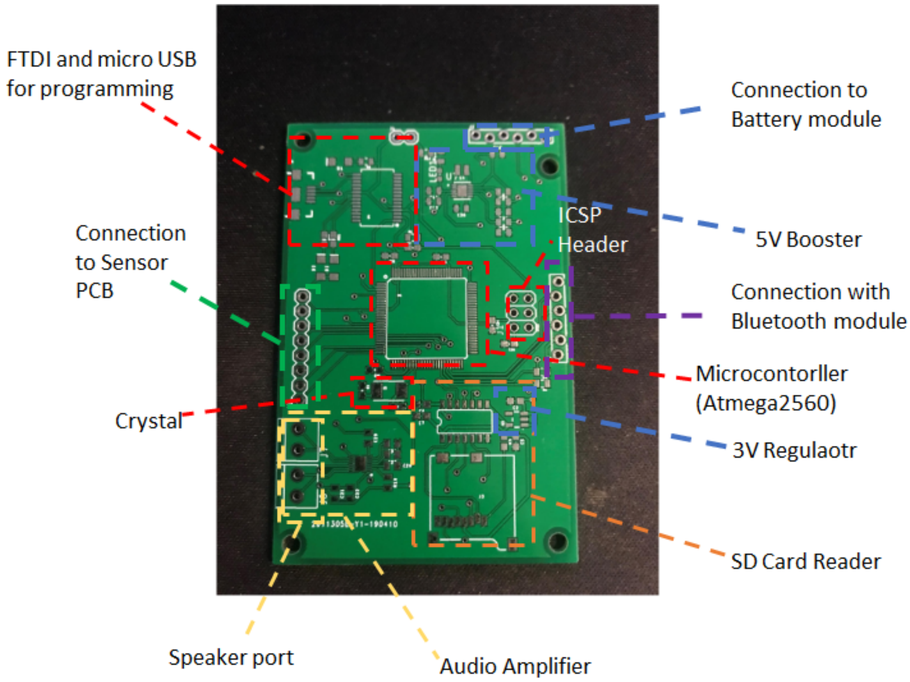


Figure 15. Picture of Main PCB

- Software (System & Android Application)

  - The Arduino codes are implemented to enable Bluetooth module, Application module, Doppler Radar Sensor module, SD card module, speaker module, and the battery module. It also prints out the frequency of the pulse that the radars are reading and the velocity in km/hr of the object that the radars are reading to the serial monitor. We implemented three different arduino codes for using only one Doppler radar sensor, two Doppler radar sensors, and 6 radar sensors. We first planned to use 6 Doppler radar sensors to cover 360 degrees around the user. However, we found out that there are some bugs and problem when using multiple sensors with Atmega 2560. Atmega 2560 operates 16Mhz and this is not enough for multiple sensors. We noticed that using STM32 which operates 25Mhz would solve this problem and also using multiple micro controllers would also solve this problem. Thus, to show the difference depends on how many Doppler radar sensors are being used, we implemented three different codes. They are almost the same but have different numbers of sensors enabled in the code. We set 10 km/hr as the "dangerous speed" to alert for the demo and we decided to output the fastest speed object's direction. To cover all the cases, there are 2^6 (6 sensors) which is 64 cases and we decided not to cover all the extreme cases where three objects are approaching at the same time. When multiple objects are approaching at the same time, we output the most dangerous object by comparing their velocities.

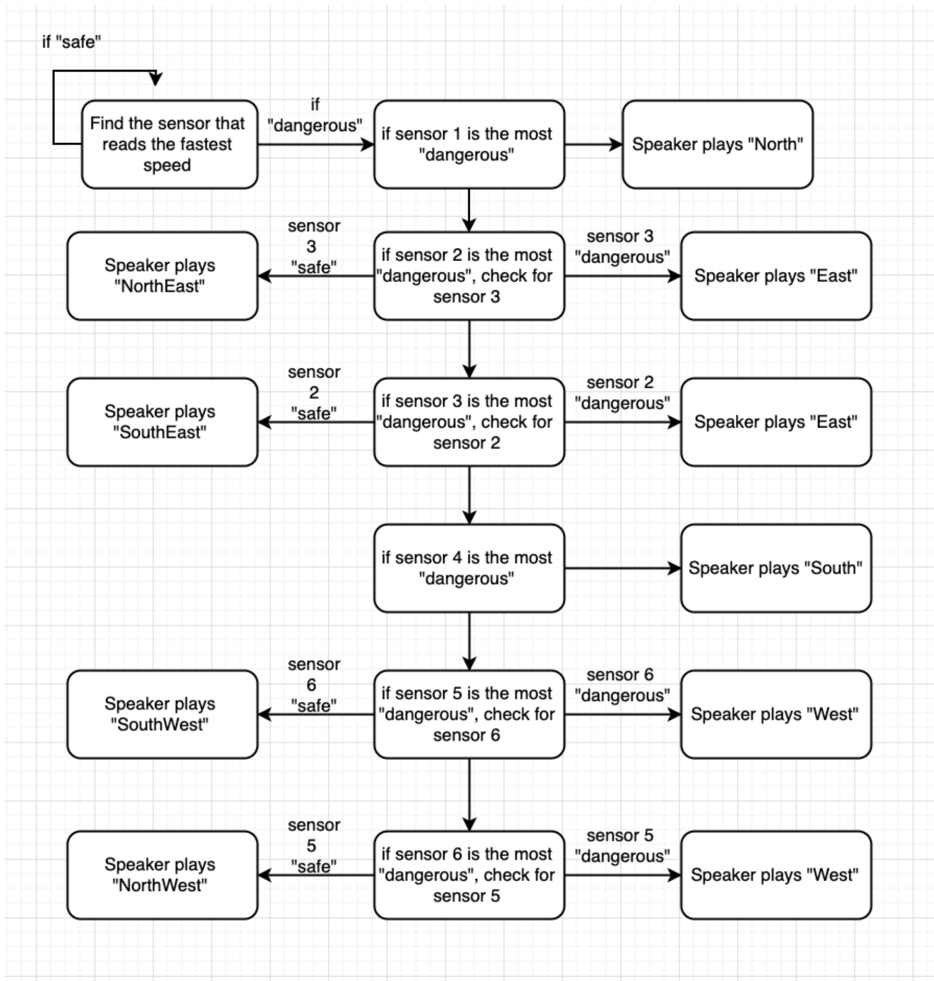


Figure 16. Logic for Doppler Radar sensors

- Doppler radar sensor

- We declared the struct for Doppler radar sensor (Figure. 17). It contains the pin number, array of the samples, Boolean expression of samples, nbpulsetime, ttime, frequency, and pulse length. They are important data to keep track for Doppler radar sensors. The Figure.

2 shows that the Doppler radar sensor receives data using interrupt and stores the data at the array of samples.

```
typedef struct {

  int pin_number;
  unsigned int samples[AVERAGE];
  bool samples_ok;
  unsigned int nbPulsesTime;
  unsigned int Ttime;
  unsigned int Freq;
  unsigned int pulse_length;

}sensor;
```

Figure 17. Struct of the Sensor

```
noInterrupts();
  // 1st sensor
    pulseIn(PIN_NUMBER, HIGH);
    //unsigned int pulse_length = 0;
    for (x = 0; x < AVERAGE; x++)
    {
      Sensor1.pulse_length = pulseIn(PIN_NUMBER, HIGH);
      Sensor1.pulse_length += pulseIn(PIN_NUMBER, LOW);
      Sensor1.samples[x] = Sensor1.pulse_length;
    }
interrupts();
```

Figure 18. Sensor getting pulse

```
//// Check for consistency
//    Sensor1.samples_ok = true;
//    Sensor1.nbPulsesTime = Sensor1.samples[0];
//    for (x = 1; x < AVERAGE; x++)
//    {
//    Sensor1.nbPulsesTime += Sensor1.samples[x];
//    if ((Sensor1.samples[x] > Sensor1.samples[0] * 2) || (Sensor1.samples[x] <
Sensor1.samples[0] / 2))
//    {
//        Sensor1.samples_ok = false;
//    }
//    }
//
//    if (Sensor1.samples_ok)
//    {
//      Sensor1.Ttime = Sensor1.nbPulsesTime / AVERAGE;
//      Sensor1.Freq = 1000000 / Sensor1.Ttime;
//    if(Sensor1.Freq/doppler_div <=80){
//    #ifdef PYTHON_OUTPUT
//    Serial.write(Sensor1.Freq/doppler_div);
//    #else
//    Serial.print("\r\n");
//    Serial.print(Sensor1.Freq);
//    Serial.print("Hz : ");
//    Serial.print(Sensor1.Freq/doppler_div);
//    Serial.print("km/h\r\n");
//    #endif
//
//        if(Sensor1.Freq/doppler_div>=5){
//           tmrpcm.play("west.wav");
//           delay(500);
//           tmrpcm.play("west.wav");
//           delay(500);
//        }
//
//
//    }
//    }
```

Figure 19. Printing out Frequency and velocity

• Speaker and SD card

- We used “SD.h” and “TMRpcm.h” libraries for speaker and SD card modules. The libraries were already written and the functions were written. We studied about the libraries function and decided which function to use to enable a speaker and a SD card reader. We saved total 7 wav files in SD card : test.wav, west.wav, north.wav, east.wav, south.wav, northeast.wav, northwest.wav, southeast.wav, southwest.wav. We implemented an algorithm to determine the most dangerous direction and outputs the corresponding wav file to the speaker. There is a library function which is “tmrpcm.play(“wav file name”) and we used this library function to output the audio files. We also initialized the volume of the speaker to 5 by using “tmrpcm.setVolume(5)”

• Android App

- We implemented Android Application that sends out the location information of the device user to the saved phone contact when the button at the device is pressed. We used MIT App inventor, which is an open source web application managed by the Massachusetts Institute of Technology. It allows to create software applications for the Android operating system easily without any detailed knowledge with application and do not require app inventing experience. We thought sending current location information to the saved phone number is essential function of our device because it is the device for visual impaired people and sending current location function would be beneficial in many ways. It could be useful when they want to notify the users' location for pick ups and also it could be used when they are in emergency situation. We didn't have any experience building Android application and we didn't have enough time to educate ourselves for DB, MySQL , etc. However, we found a useful online source to build up Android application. As shown in Figure 20, the user can connect Bluetooth with the main PCB board by clicking the Bluetooth button located in the middle of the application main page.

Figure 20. The main display of the application

- After the Bluetooth is connected with the main PCB, the location sensor is enabled and ready to scan the current location of the user. In a clock cycle, the application keeps check whether it receives data from the Bluetooth client, which is the main PCB in this case. As shown in Figure 21, we implemented in Arduino that sends out a character “p” to the Bluetooth server when the button is pressed. The Sms text message contains the information of latitude and longitude information of the user which is like the Figure 22. (e.g. “Help! I am at Help! I am at latitude is : 40.11682, longitude is : -88.23028 current address is :608 E University Ave, Champaign, IL 61820 ,USA)

```
//button check
if (digitalRead(button) == HIGH) {
      mySerial.write("p");
}
```

Figure 21. Arduino code that sends out a character “p” to Bluetooth server

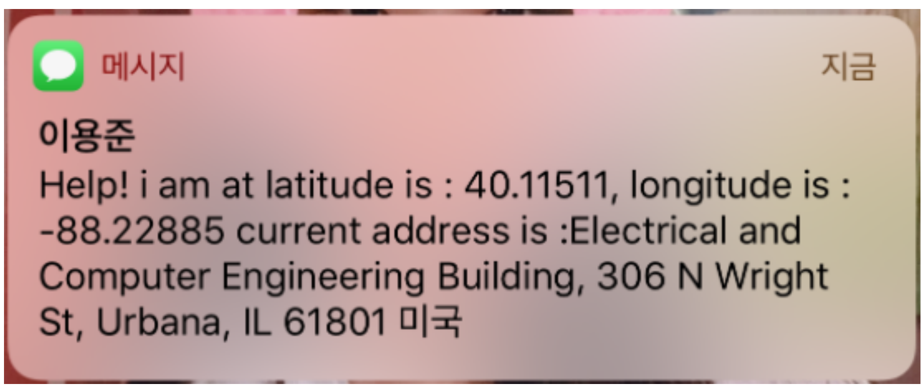


Figure 22. Text message sent from the app

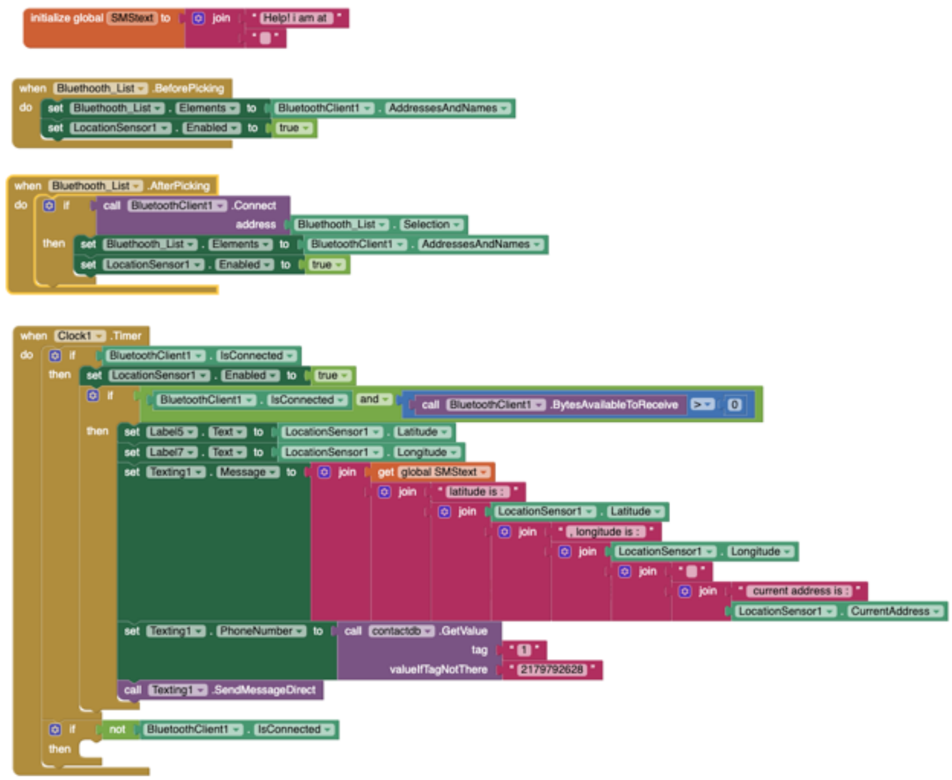


Figure 23. The block diagram and the logic of the application

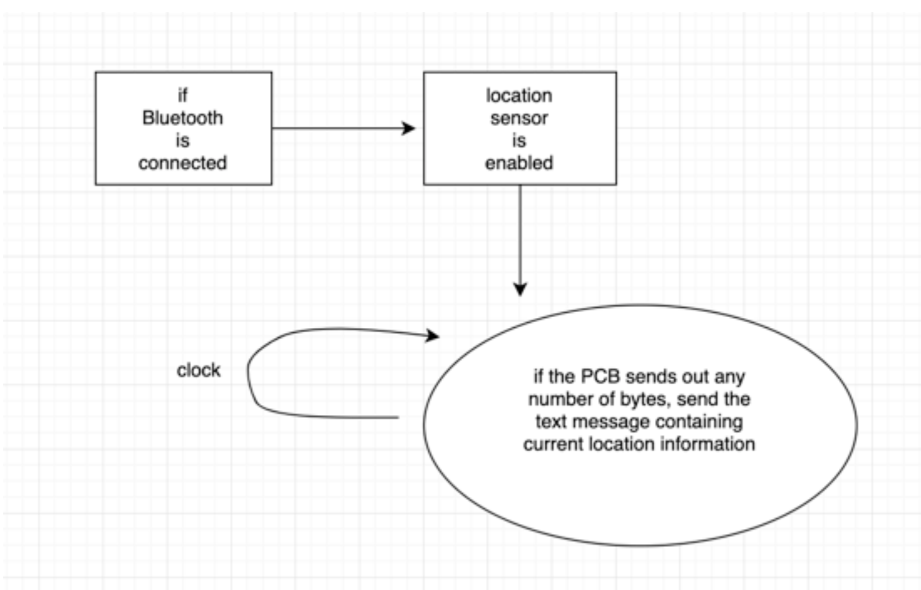


Figure 24. Simple logic of the App

2.4 Requirement and Verification

- Microcontroller

Requirement	Verification
-The microcontroller must be able to be programmed via the FTDI micro USB serial convertor. - The microcontroller, Atmega 2560, must be able to communicate with Bluetooth module, Doppler Radar Sensor, SD card module and Speaker module. - The microcontroller should process six different doppler radar sensors in same time.	- The microcontroller reads amplified PWM from the doppler radar sensor. Then it determines direction of an object and calculates the speed of the object that is approaching toward a user. And the microcontroller reads a .WAV files corresponding to the direction of the object coming, and converts the digital data to analog data. Then the microcontroller sends a data to the speaker module. -When the button is pressed, the microcontroller sends trigger to cell phone via the Bluetooth serial so that the cell phone can send GPS data to the person who registered through an application.

- Doppler Radar Sensor

Requirement	Verification
- The gain of sensor amplifier has to be at least two hundreds. (approximately 100 from the first op-amp, and approximately 120 from the second.) - Six doppler radar sensors covers 360 degree. (60 degree for each and aligned them to cover 360)	- The doppler radar sensors must detect an object that is approaching toward the sensor. Or check the gain of the amplifier using oscilloscope.

• Bluetooth Module

Requirement	Verification
- The Bluetooth module must be able to send a signal from a button located back of the helmet to android cell phone.	- Were the Bluetooth module successfully able to send a signal to a cell phone connected to the module, then the cell phone sends longitude and latitude of a user, and help message to designated number.

• Buck Boost Converter

Requirement	Verification
- The 5V boost converter must be able to convert battery voltage to 5 V(System). - The output current from the booster should meet the required amount for our project.	- The output voltage from the booster is constant 5V which could power up our system. - The output current is enough to run all the components in the device.

• Speaker Module

Requirement	Verification
- The gain of audio amplifier has to be at least eight(18db). - Sound coming from the speaker has to be equivalent to 60 ~ 65 decibel.	- The audio amplifier successfully amplifies analog input so that the speaker outputs sound that is audible to users. Or Check the gain of the amplifier using oscilloscope. - Average loudness of human talking voices is around 60 decibel, and loudness of the speaker is close to average loudness of human conversation

• SD Card Module

Requirement	Verification
- The SD Card module must be able to sends recorded .WAV files' data to the microcontroller. - The 3.3V regulator of the SD Card module must be able to convert from 5 V to 3.3 V in order to operate the SD card module.	- The speaker play the recorded human voice saved on the SD card successfully. - The 3.3V regulator provide constant 3.3V to the SD card module so that the module works perfectly.

• Power Management Module

Requirement	Verification
- The power management module should be able to constantly supply 3.7V to main PCB. - The module must be able to charge Li-po battery. -The power management module must be able to estimate the remaining capacity of the battery.	- By connecting the battery management module, LEDs on the main PCB should blink. - Connect the power management module to arduino Atmega 2560 and program the Atmega 2560 using arduino to serial print the the remaining capacity and state of charge.

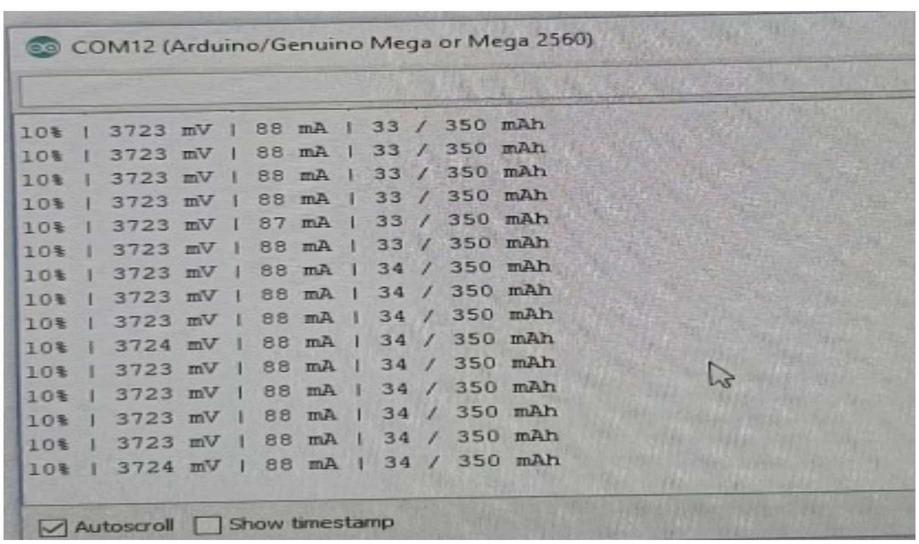


Figure 25. Arduino serial monitor printing remaining capacity and state of charge

2.5 Cost and Schedule

2.5.1 Cost

Cost of Labor

We consider the University of Illinois at Urbana Champaign ECE graduate average starting salary is $45/hour. We have three group members in total and we estimated 10 hours/week as a weekly time effort.

$$3 * 2.5 * \$45/hour * 10\ hour/\ week * 12\ weeks = \$\ 40,500$$

Cost of Parts

Part	Cost
Doppler radar sensor x 6 : HB100 (1.2mA~4mA with 3 to 10% duty cycle pulse at 5v, ST Electronics)	$5.83 * 6 = $34.98
Li-ion Battery x 4 : (<5V and 100mA, last over 6 hours, rechargeable, LiFePO4 3.2V 1500mAh, AA Portable Power Corp)	$3.75 * 6 = $22.5
Speaker (Bone Conductor Transducer with Wires - 8 Ohm 1 Watt, adafruit)	$8.95
Pyramex Safety Products HP54117 Ridgeline Full Brim Hard Hat	$25.89
Microcrontroller x 2 : Atmega 2560 (3.3 V, 500mA, Espressif Systems)	$3.96 * 2 = $7.92
Amplifier circuit *6	$2.01*6 = $12.06
Total	$112.3

2.5.2 Schedule

Week	TaeHwa Kim	YongJun Lee	WooYoung Choi
2/11	Research on doppler radar sensor	Research on doppler radar sensor	Get used to Arduino
2/18	Built amplifier for the doppler radar sensor	Built amplifier for the doppler radar sensor	Working on bluetooth part on software
2/25	Tested doppler radar sensor with amplifier and Working on its PCB	Tested doppler radar sensor with amplifier	Working on doppler radar sensors part on software
3/4	Design Main PCB Order Amp PCB	Research on power management module	Working on doppler radar sensors part on software
3/11	Research and test audio module	Order components for Amp PCB	Research on MIT App inventor
3/18	Research and test 5V booster module and bluetooth module	Design Power management PCB	Working on android application
3/25	Finalizing circuit and sensors and soldering	Order Power management PCB	Working on android application
4/1	Order Main PCB	Solder PCB	Finished android application
4/8	Solder PCBs and test	Tested with other PCB	Working on SD card module on software
4/15	Finalizing all the hardware parts	Finalizing all the hardware parts	Finalizing all the software parts
4/22	Preparing for the demo		
4/29	Preparing for the presentation, writing final report, and preparing for the poster session.		

3. Conclusion

Overall, our project was successful that every individual sub modules worked as we expected. The entire project worked also successfully when we combined all the modules together. We used 3.7V Li-Po battery and since other modules need 5 V, we built a buck boost converter. After doing a test, we could check that the converter successfully converts the voltage. We also checked that the battery is rechargeable and could check the remaining amount by running software codes.The sensor module, which is consisted with 6 amplifiers and 6 HB-100 doppler radar sensors, worked as we expected. We checked with a oscilloscope and we also checked with Arduino program. For our voltage regulator, we gave an input as 3.7 V and were able to check the 5 V output. Microcontroller module worked successfully because we were able to program in it. Speaker module was also successful that we were able to read wav file from SD card and get the output through the attached speaker. We used HC-05 bluetooth module and it worked as we expected that we could connect with the phone with bluetooth. Lastly, the Android application that we built was also successful that it worked as we expected that it sends the location information to the saved phone number when the button was pressed.

However, there were some difficulties we faced during the project and there were some things we need to improve. Almost at the end of the project, we found out that using multiple sensors with one Atmega 2560 microcontroller could occur error and the performance would not be as smooth as just using one sensor. We were not able to find this error earlier because when we tested with protoboard, we used two microcontrollers so there were no errors. We couldn't guess there might be an error with using multiple sensors using one Atmega 2560. We spend a lot of time debugging and soldering amplifier circuit and we finally were able to test the whole 6 doppler radar sensors at the last minute before the demo. Therefore, we didn't have enough time to fix this error and improve the performance. However, we found the reason for the issue. After doing research and visiting office hours, we found out that Atmega 2560 uses 16Mhz and it is slow for using multiple sensors at the same time. Using STM32 microcontroller, which uses 25Mhz, would be a better option. Also using multiple Atmega 2560 microcontrollers could be another option that each microcontroller is in charge of 2-3 doppler radar sensors.

Another issue we had and the thing we want to improve is that we used LM324 for op-amp in our amplifier circuit. However, the quality of LM324 isn't good enough that there were a lot of noises that we couldn't get perfectly smooth outputs. We chose to use LM324 because the datasheet recommended using it and also, it was easy to get. If we replaced LM324 with another op-amp which has better performance, we would have better results. Beside that, every submodules worked as we expected and the overall flow was successful. We were able to manage the time schedule and were able to finish building a device on time.

If we had more time, we would have worked with more microcontrollers or use STM32 microcontroller to improve the performance of the device overall. Also, we were not able to meet perfectly for IEEE safety requirement as we stated in the proposal. We didn't have enough time to make it waterproof and perfectly safe from dust. However, if we had more time, we would have worked more details to meet all the IEEE safety requirements and ethical requirements. In the project proposal and the design review, we planned to use thin copper plate to set the degrees for doppler radar sensors. However, we tested with copper plate and it didn't actually work. We assumed it didn't work because we used the cheapest work, which has low quality. Better quality copper plate was too expensive that we were not able to try in this project. We decided to fix this problem with implementing software codes. If we had more time, we will find better ways to reflect and block microwave, which doppler radar sensor emits and receives. This will allow use to have better result and more accurate data without using software algorithm.

Throughout the course, we were able to design and schedule everything ourselves for the project. For other courses, we just followed the syllabus, attend the lectures, and study for the midterms. However, we get to choose what to do for one semester and plan the entire thing ourselves for ECE 445 which was a great experience taking it. We could experience and learn self time management skills and self working skills. We were also able to learn and experience team working environment and able to practice self managing skills which was significantly helpful and meaningful. ECE 445 has been very valuable experience and we were able to learn and experience so many things through this course. We all hope this valuable experience would be helpful after the graduation if we work in companies or if we go to the graduate schools.

Reference

1. Shirin E. Hassan, “Are Normally Sighted, Visually Impaired, and Blind Pedestrians Accurate and Reliable at Making Street Crossing Decisions?,” 2012. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3358127/#i1552-5783-53-6-2593-b04

2. Wilmer Eye Institute, “Traffic gap judgment in people with significant peripheral field loss,” 2008. https://www.ncbi.nlm.nih.gov/pubmed/18174838

3. Matt Pressman, “Over 3 Million Electric Cars On The Road — You Driving One Yet?,” 2018.https://cleantechnica.com/2018/07/03/over-3-million-electric-cars-on-the-road-you-drivingone-yet/

4. Statista, “Number of participants in bicycling in the United States from 2006 to 2017 (in millions),” 2017. https://www.statista.com/statistics/191204/participants-in-bicycling-in-the-us-since-2006/

5. theoryCIRCUIT - Do It Yourself Electronics Projects. (2019). *HB100 Microwave Motion sensor Interfacing Arduino*. [online] Available at: http://www.theorycircuit.com/hb100-microwave-motion-sensor-interfacing-arduino/ [Accessed 27 Feburary 2019].

6. YouTube. (2019). *RCWL-0516 Microwave Radar Sensor Arduino Distance Test and Review*. [online] Available at: https://www.youtube.com/watch?v=H9TdEP94LeI [Accessed 1 May 2019].

7. ST Electronics, “X-abnd Microwave Motion Sensor Module,” https://www.limpkin.fr/public/HB100/HB100_Microwave_Sensor_Application_Note.pdf

8. Amazon.com. (2019). [online] Available at: https://www.amazon.com/Generic-Surface-Conductor-Transducer-Exciter/dp/B00Y8ISYHG [Accessed 3 March 2019].

9. Espressif Systems, 2019, https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

10. Espressif Systems, 2019,
https://www.espressif.com/sites/default/files/documentation/esp32_bluetooth_networking_user_guide_en.pdf

11. Healthline. (2019). *Average Shoulder Width and How to Measure Yours*. [online] Available at: https://www.healthline.com/health/average-shoulder-width#the-takeaway [Accessed 4 March 2019].