Wireless Bicycle Signaling System

Kevin Tian

Larry Liu

Suriya Kodeswaran

Final Report for ECE 445, Senior Design

TA: Xinrui Zhu

Fall 2017

December 12, 2017

Project No. 38

Contents

1. Introduction

1.1 Objective and Background

In today's society, bicyclists continue to utilize hand signals to communicate their intended direction of turning to other cyclists and vehicles. We aim to eliminate the need for hand gestures as they sometimes pose a lack of physical clarity (especially at night) as well as provide more convenience for riders in keeping their hands on handlebars. "In 2015 in the United States, over 1,000 bicyclists died and there were almost 467,000 bicycle-related injuries" [3]. Numerous of these deaths result from vehicle drivers having low visibility of other road bikers either at night or miscommunication from unclear bicycle signaling. Reflector lights, blinkers, and hand signals are just not sufficient for providing road safety to cyclists. Our project aims to resolve these real-world issues through small modules fitted to bicycles.

On the front of the bicycle, two pressure sensing handlebar pads will placed onto the inner sides of both handlebars, which we will refer to as pressure sleeves. The system will work as follows: When a cyclist is ready to signal turning, he/she would squeeze the pressure sleeve corresponding to the right or left turn. The MCU will detect the press from the circuit logic, and forward output signals that are wirelessly transmitted to the helmet receiver connected to the LED lights. The appropriate LEDs will light up to display the cyclist's intended turn. After the cyclist completes the turn, a second press of the corresponding sensor will shut off the LEDs. If the cyclist forgets to turn the blinking lights off, a 12 second timeout will force them off. Additionally, another module we'll attach to the back of the bicycle includes an encasing of ultrasonic sensors and another set of LED lights. When these sensors detect another object approaching the bicycle too closely, a signal would trigger these LEDs to start flashing. The goal is to prevent collisions when other vehicle drivers become unaware of or do not see the cyclist.

Evidently, our approach provides more than simply convenience for cyclists, but also promotes better road safety as it's shared by countless other cyclists and road vehicles. The rest of this paper will outline our approaches, design decisions, verifications, costs, and references for this project.

2. Design

2.1.1 Block Diagram

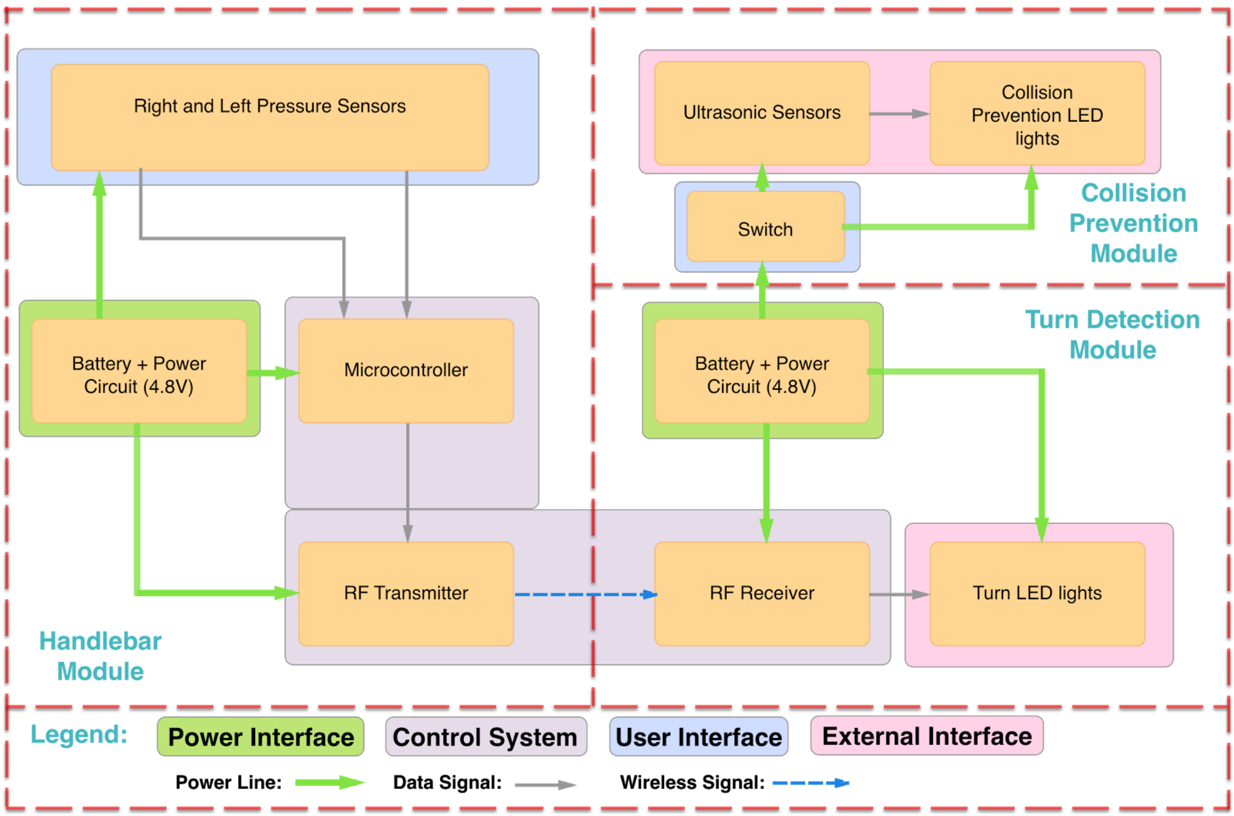

Figure 1. Block Diagram of Wireless Signaling System.

2.1.2 Physical Design

Figure 2. Physical Design of our project. Photo Credit: Professor Can Bayram.

Shown above is a picture of our working project during demo day of the course. The picture clearly depicts the two separate components as described earlier in the introduction as well as our block diagram. The high-level overview of our system's functionalities and demo links are as follows:

Basic Functionalities:

1) When a pressure sensor is pressed, the corresponding helmet's turn LEDs will start blinking to signal the desired turn.
2) Ultrasonic sensor flashes rear LEDs on detection of an incoming object

Demo Links:

1) Wireless transmission of signals from pressure sensors of front circuit
2) Ultrasonic sensor detecting approaching objects behind it

2.2 Front Module

2.2.1 Pressure Sensor: Force Sensitive Resistor 0.5"

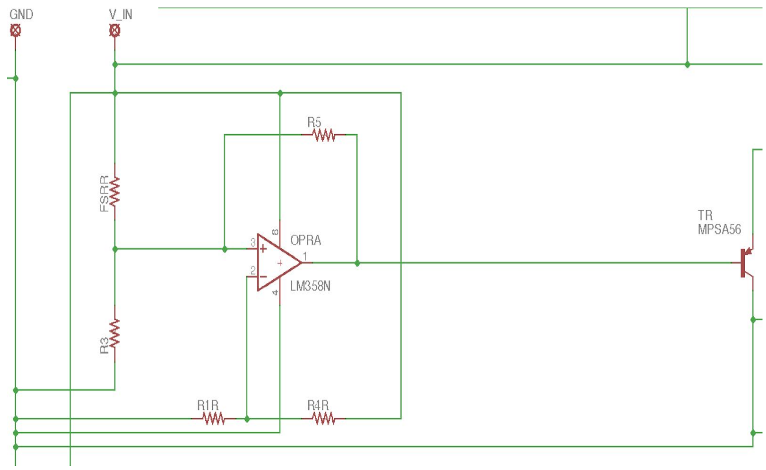

Figure 3a. FSR Threshold Switch Circuit Schematic [4].

The circuit schematic in Figure 3a. shows the force sensitive resistor threshold switch. The two force sensitive resistors are the main input that the user provides to the circuit. One FSR will be located on each handlebar and be easy to squeeze for the biker. When the FSR is squeezed, its resistance decreases, causing the voltage drop across it to decrease as well. We use this in conjunction with an Op-Amp functioning as a comparator. When the positive node of the Op-Amp has a higher voltage than the negative node, the Op-Amp emits high voltage, else low voltage. To set our threshold, we characterized the FSR and discovered its resistance vs force relationship, shown below in Figure 3b.

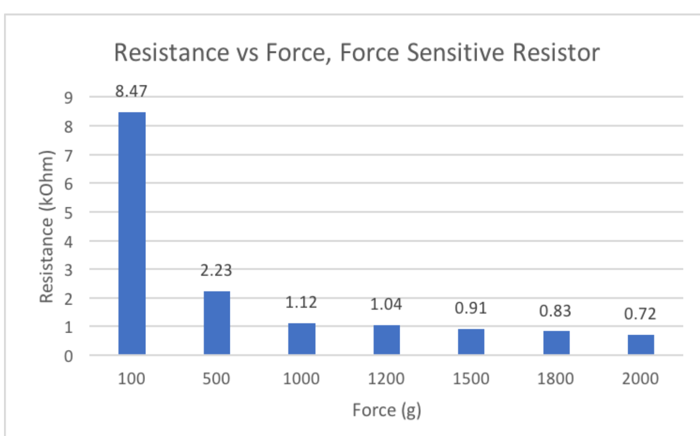

Figure 3b. Data from Force Sensitive Resistor

The data follows a logarithmic trend. The FSR will have 1MΩ resistance when no force is applied to it, and only responds to a minimum force of 100g [4]. The target force of 4lbs is approximately 1800g, at which force the FSR will offer 830Ω of resistance. Using a simple voltage divider circuit, we are able to calculate the exact voltages that the positive and negative nodes of the the Op-Amp should receive to properly process the input signals and determine if the biker has issued a push. The output of the Op-Amp is then directed as to the base node of a transistor. Because the input pins of our MCU are active high, we want a grounded signal to indicate that the pressure sensor had been pressed. The transistors, when receiving a sufficient voltage from the force threshold switch, will route ground to the corresponding input pin on the MCU, thus passing knowledge of if a button was pressed to the MCU.

2.2.2 Microcontroller: ATMega328p

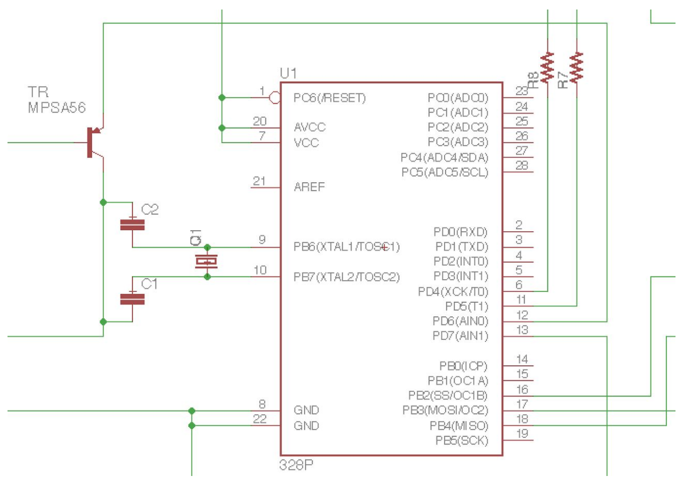

Figure 4. ATmega328 Microcontroller Circuit Schematic [1].

We chose to use the ATmega328p microcontroller, shown above in Figure 4. It has 32 KB Flash Memory, 32 pins with an input voltage of 4.8 V. Active high inputs were set on digital pins 6 and 3. This is due to the high sensitivity of the digital pins if there were to be used as inputs. For example, simply placing a wire into the the pin would be interpreted as high. Thus, we made the design decision to use active high inputs that were switched to low when receiving a logical high input. We accomplished this using the internal pull-up resistor in the ATmega328p. MCU logical outputs were delivered out on digital pins 5 and 4. Our MCU operates standalone using external clock at a frequency of 16 MHz and two capacitors. We programmed the microcontroller via AVRDUDE, which is a utility to download, upload and manipulate ROM and EEPROM contents of AVR microcontrollers via the ISP technique [2]. The figure shown below outlines a high-level state diagram of our MCU logic for processing user inputs, simulating blinking and managing outputs.

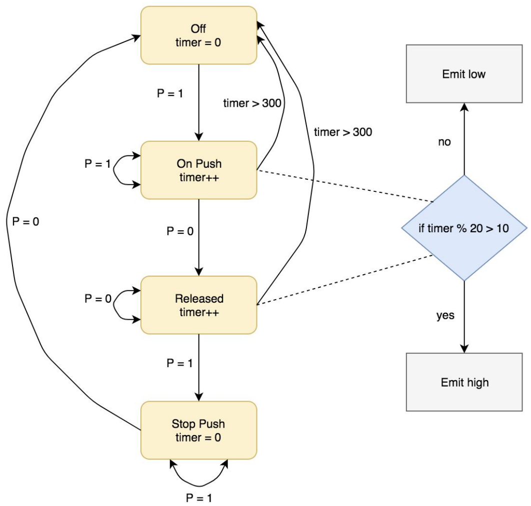

Figure 5. State Diagram of for ATmega328p input/output logic.

2.2.3 RF Transmitter: WRL-10534 434 Mhz

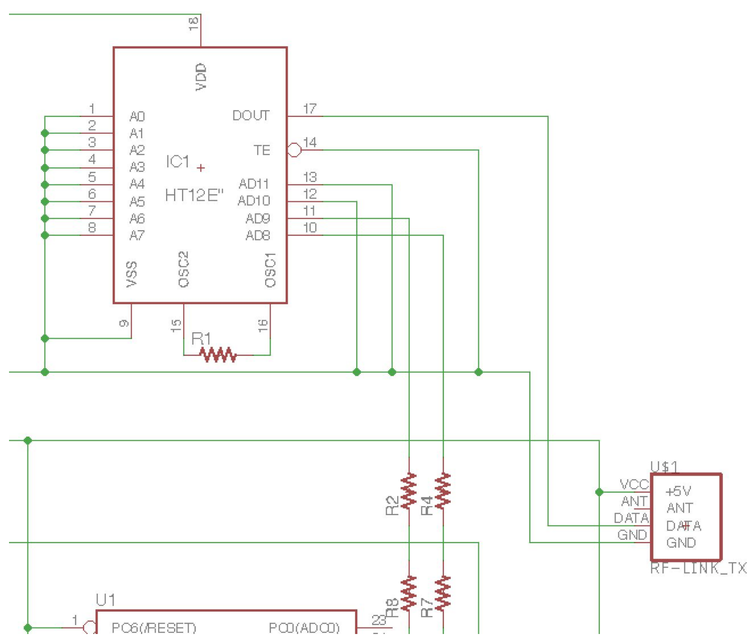

Figure 6. HT12E and RF Link Transmitter Circuit Schematic [6].

In our design of wirelessly transmitting data from our front printed circuit board to the rear circuits, we utilized a 4-bit encoder to translate individual bit information received from digital pin outputs of MCU. The encoder relayed information to the data in pin of our transmitter as shown in the above Figure. With the correct input power of 4.8 V, the RF Link Transmitter wirelessly transmitted data to the receiver at 434 MHz [10]. We chose to utilize this method for encoding information instead of another implementation such as Bluetooth because our design is low-powered and minimalistic considering we are not transferring an enormous amount of data. We are constantly transferring information bits to the rear PCB to determine the state of the helmet LEDs; thus, it becomes a pivotal requirement for our design to be low-powered.

<u>2.3 Rear Module</u>

2.3.1 RF Receiver: *WRL-10532 434 MHz*

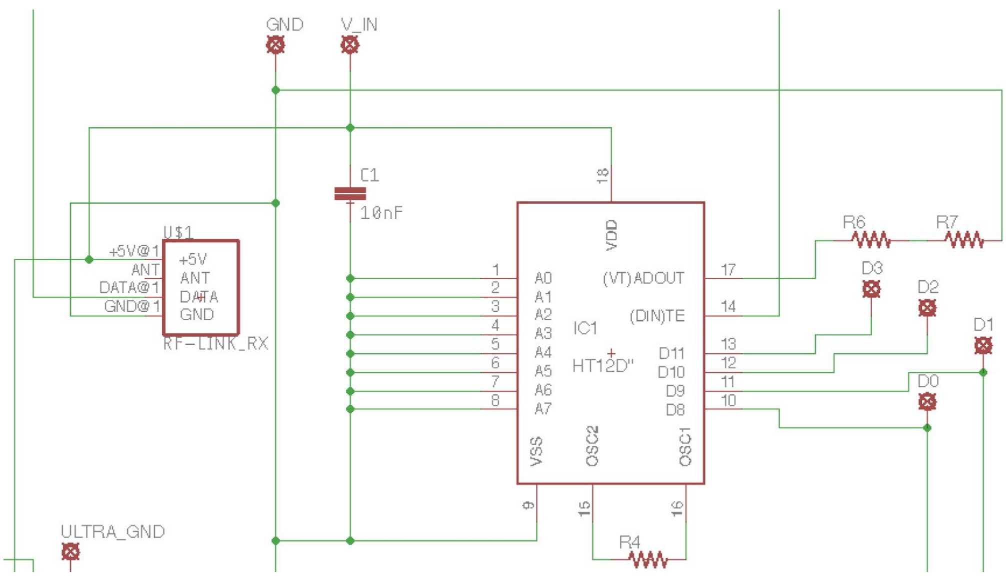

Figure 7. HT12D and RF Link Receiver Circuit Schematic [5].

Data transferred at 434 MHz and eventually arriving at RF Link receiver [9] is routed to a decoder. Decoded output bits are combined with Ultrasonic Sensor output to determine the state of the helmet LEDs. Evidently, we implemented a similar design as how we encoded and transmitted the bits wirelessly. Again, this is to maintain a low-powered circuit. We discuss how we verified the wireless transmission of data in our verifications section of this paper for this component of the project.

2.3.2 Ultrasonic Sensor: LV-MaxSonar-EZ2, SEN-08503

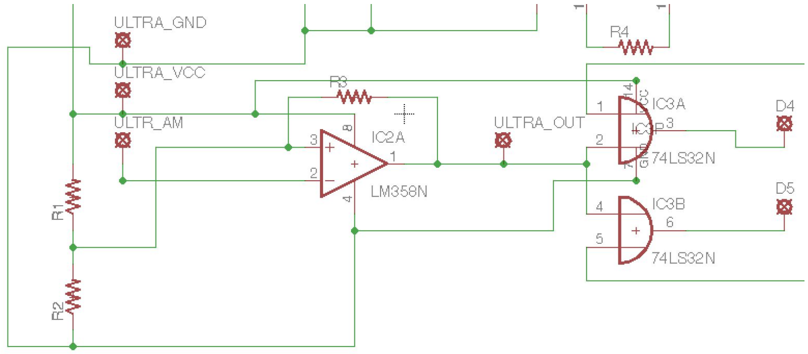

Figure 8a. Ultrasonic Sensor Switch Circuit Schematic [8].

The ultrasonic sensor is the system's source of input data, along with the FSRs. The ultrasonic sensor circuit behave in many ways similarly to that of the FSR. It also utilizes an Op-Amp as a comparator to set a voltage threshold. In this case, because the ultrasonic sensor output voltage decreases as something approaches it, it is hooked up to the negative node while the positive node is fixed using the voltage divider rule after characterizing the ultrasonic sensor. In this way, when the negative end drops below the positive end, indicating that something is within the desired detection range, the Op-Amp will output high voltage, else low voltage. The status of the ultrasonic sensor is then OR'd with both the light signals that have been received by the RF receiver. The means that whenever an object is detected by the ultrasonic sensor, both lights will light up, giving it a higher priority that the turn signals. This is as intended, as it is safer for a car to driving behind a biker to know that a biker is present rather than to know if they are turning. The data found when characterizing the ultrasonic sensor is shown below in Figure 8b.

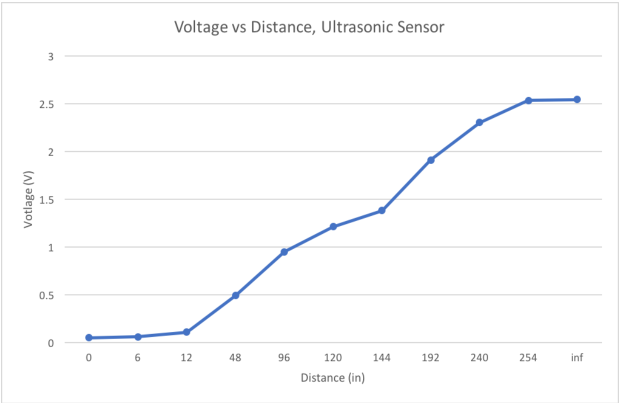

Figure 8b. Data Plot of Ultrasonic Sensor voltage vs distance tests.

The ultrasonic sensor operate with a detection range between 256 and 6 inches. When nothing is detected, the output is Vcc/2. From there, the output voltage decreases linearly with distance, with a ratio of Vout = (Vcc/512) * Distance [8]. Our desired detection distance of 10 ft. results in an ultrasonic sensor output voltage of 1.27V, as shown in Figure 8b, which we will assign as the voltage threshold. The ultrasonic sensor results are always consistent and provide accuracy up to 1 inch. However, the cone of detection for the sensor is very wide, requiring precision surrounding the angle it operate at.

2.3.3 LEDs

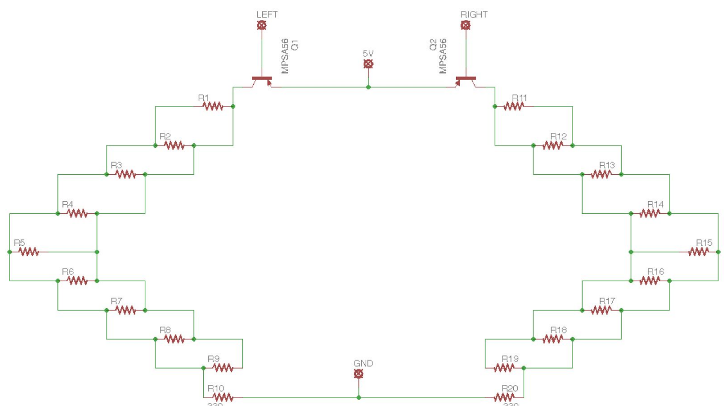

Figure 9. LED Turn Signaling Circuit Schematic.

We needed to create a design that does not use too many LEDs but is easily visible to people around the user and makes it clear to the intent of the user. We decided to create a left arrow and right arrow each being made up of 9 LEDs for a total of 18 LEDs. Each set of 9 LEDs are in parallel to each other so even if one LED fails then the rest of the LEDs will still function. Whenever a high input is received by the left or the right transistor, it allows the battery pack to power and light up the LEDs.

2.4 Power Consumption

Front Component:

Battery Pack: 4.8V - Three 1.6V AA batteries (11.25 Wh)

MCU (0.96 mW), RF Transmitter (38.4 mW)

11.25 Wh / (0.96 mW + 38.4 mW) = 285.8 hours

Rear Component:

Battery Pack: 4.8V - Three 1.6V AA batteries (11.25 Wh)

18 LEDs (18*80 mW), Ultrasonic Sensor (9.6 mW), RF Receiver (38.4 mW)

11.25 Wh / (18*80 mW + 9.6mW + 38.4 mW) = 192.1 hours

3. Design Verification

(Refer to the Appendix for the original R&V table)

3.1 Front Module

3.1.1 Force Sensitive Resistor

To test the FSR module, we needed to verify that a force of at least 4 lbs on the pressure sensor would result in the corresponding pin to the MCU to be grounded. This verifies the voltage threshold we set, the Op-Amp comparator, and the transistor as all functional. A sufficient force, tested using weights (Figure 10.), proved that the input to the MCU was indeed grounded at the correct times, where is can then be read and processed.

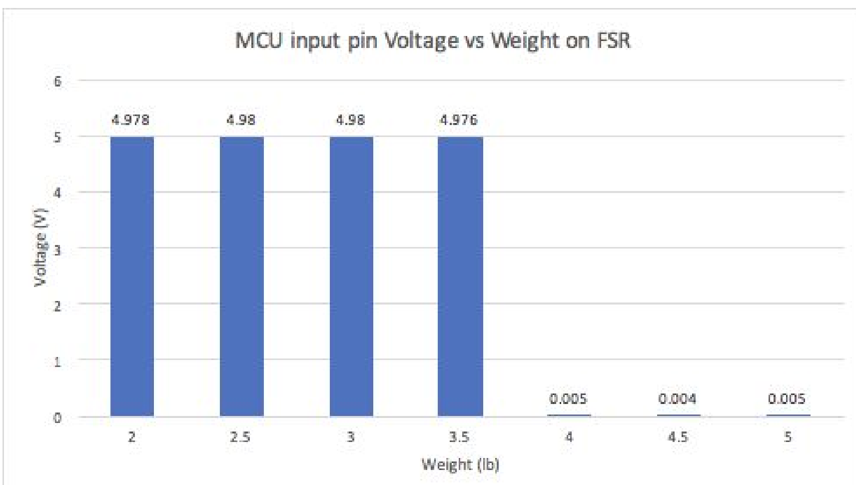

Figure 10. MCU input pin Voltage vs Weight on FSR

3.1.2 Microcontroller

After we verified that the MCU was receiving the input signals from the FSR, we needed to ensure that is processed and outputted them properly. This was done using an oscilloscope and checking the input and output pins of the MCU simultaneously. We verified using the state diagram (Figure 5) that the output pin was changing in accordance to the input signal and all of the corresponding software states and logic, including blinking and timer functionality, are present.

3.1.3 Encoder/Decoder

Encoder	Encoder Input Value	Decoder LED
Data Pin 0	High	On
Data Pin 1	Low	Off
Data Pin 2	High	On
Data Pin 3	Low	Off

Figure 11. Encoder/Decoder Verification Table.

We verified the encoder and decoder together by taking the output from the decoder and wiring it directly into the input of the decoder. We grounded the address bits of the both chips so they matched and manually turned the data pins on the encoder high and low. We then connected the output data pins on the decoder to LEDs and whenever a data bit on the encoder is high the respective LED would turn on.

3.2 Back Module

3.2.1 Transmitter/Receiver

In order to test the transmitter and receiver we connected the encoder to the transmitter and connected the decoder to the receiver. Refer back to Figure 10 for the table we used to verify the transmitter and receiver. We used it to verify that the serialized bits coming from the encoder was properly sent from the transmitter to the receiver and the decoder would deserialize the bits.

3.2.2 Ultrasonic Sensor

The ultrasonic sensor is required to detect an object within a distance of 10 feet. This would be discernible by checking the output of the op-amp the ultrasonic sensor is attached to. Similar to the verification for the FSR, we made sure that the threshold for the ultrasonic sensor is properly set and that it catches anything within the desired detection distance. The result of this verification is show below in Figure 12.

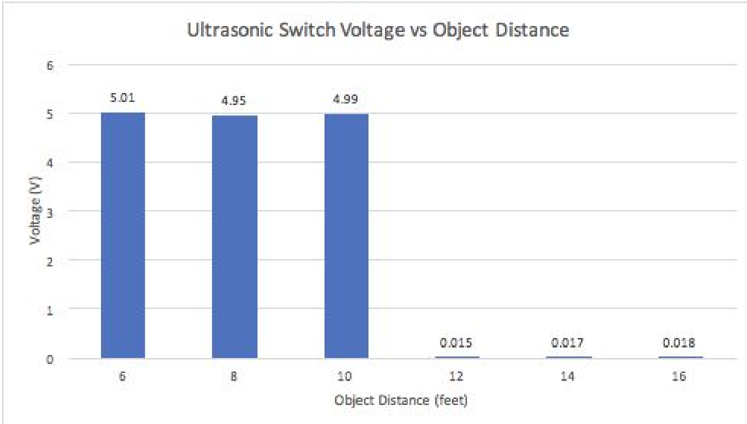

Figure 12. Ultrasonic Switch Voltage vs Object Distance

3.2.3 LEDs

When a high input is received at the base of the left transistor then all of the LEDs that make up the left arrow should light up and same with the right transistor and right LEDs. We verified this by simply powering the LEDs and then setting the base on the left or right transistor to high and seeing if the LEDs light up.

4. Costs

4.1 Parts

Part	Model	Unit Cost	Quantity	Total
Microcontroller	ATmega328p (bought at ECE store within Arduino Dev board)	$20	1	$20
Ultrasonic Range Finder	LV-MaxSonar-EZ2, SEN-08503	$27.95	1	$27.95
Force Sensitive Resistor 0.5”	SEN-09375	$4.95	2	$9.90
RF Link Transmitter	WRL-10534 0 434 MHz	$3.95	1	$3.95
RF Link Receiver	WRL-10532 - 434 MHz	$4.95	1	$4.95
Resistors	Varying - see schematics	$0.10	20	$2
Alkaline Batteries (1.6 V)	AA batteries	$2.50	2	$5
Capacitors	Varying - see schematics	$0.5	4	$2

Transistors	MPSA56	$0.5	6	$3
PCBs	Front PCB, Rear PCB, LED PCB	$1	3	$3
LEDs		$.25	20	$5
Total Costs for Parts				**$80.75**

4.2 Labor

We have a fixed salary rate of $30/hr for each employee. Because there are 3 people working, we multiply this by 3, by 2.5 on the fixed rate and finally by the time of development. This calculation is based on the references for labor wage provided by the ECE 445 Course Website.

3 * $30/hr * 2.5 * 12 hrs/week * 14 weeks = **$37,800**

5. Conclusion

5.1 Accomplishments

We were able to design and construct a modular implementation of a wireless signalling system. Our approach is easily extensible, allowing the option for a variety of different features to be added. As of now, we can offer a low power, low cost system that will immensely affect biker safety. This consists of a set of force sensitive resistors reporting input from the biker to the MCU. The MCU will communicate wirelessly to the LEDs, in conjunction with a ultrasonic sensor used for object detection. The lack of wires also makes the product incredibly easy to use and install.

5.2 Uncertainties

There a couple of simple problems surrounding our product. One for instance, is the hard-stop operational voltage of 5.1 V on our RF Receiver. This is not a good complement to our LEDs, which require a higher voltage to appear brighter. With more time, we could develop a voltage regulator circuit to power the receiver independently.

We were also not able to integrate an accelerometer into our product, as we initially intended. The accelerometer was intended to sense when the biker has completed a turn and force the MCU to turn the corresponding LEDs off. The accelerometer we chose through Sparkfun had a very low-level driver of C code. When using the code, we found that it was written assuming a different clock speed and voltage, thus making certain registers and timing constraints slightly off, rendering the output of the accelerometer incorrect. This would have required a significant amount of low-level software debugging to fix. We chose to prioritize our time on more hardware related aspects of our project, as this is a purely software challenge, as the accelerometer connects directly to the MCU. With more time, we believe we could have solved this too, but for the time being chose to implement double-press and timeout functionalities to the pressure sensor to mimic the desired effect of the accelerometer.

5.3 Ethical considerations

The greatest ethical concern for our project is the potential of placing the cyclists and the people around them in more danger than if they have never used our system in the first place. For example, if the user gets used to routinely utilizing our product and assumes the turn lights will always turn on when the switch is flipped, there may be unexpected consequences. These negative consequences may result from broken or faulty lights; thus, putting the cyclist at more risk than he or she would normally be. Thus, this is our primary focus in ensure Conduct 1 of the IEEE Code of Ethics is not violated [7]. The rest of the conducts (Conduct 2-10) will not be concerns for us for the following reasons: there is one clear objective for this project, there will be an organized structure for how are components are verified and constructed, there is no sponsorship, there will be emphasis on following safety procedures and guidelines, we will hold the responsibility in the safety of potential users, and the agreement that we must work together in an organized, professional environment.

The first safety concern we need to address is making sure that our device does not change how the user would operate the bike normally and making sure that our sensors will not interfere with the functionality and safety of the bike. Additionally if the cyclist were to fall off the bike we want to make sure none of the components in our design could potentially shatter and inadvertently hurt the cyclist. Additionally, we used alkaline batteries as we were told that lithium-ion batteries pose many hazards. It became our full focus to ensure that our project is both functional and safe; otherwise, it would defeat the purpose of the attached components mentioned in this design.

Another safety feature we've considered throughout the construction of our components for the final demo is ensuring our electronics and circuitry are well-covered. However, we did not have the manufacturing resources to complete this. If our project ever reaches production, the minimized circuits will be much easier to place in a waterproof casing. For the purposes of showing we've thought about this safety concern, we simply encased our components in large metal boxes that were attached the bicycle and helmet.

5.4 Future work

There is a variety of great improvements we can implement on our current design of the system. Shown below is list several ideas we believe would help transform our course project into a marketable product:

1) Charge helmet LEDs via micro-USB or solar cells
2) Integrate 2nd MCU into helmet circuit to perform more advanced signal processing
3) Upgrade to low-cost LIDAR to detect surrounding objects more accurately
4) Streamlined LEDs, utilize strip for better clarity and compactness
5) Generate subtle helmet vibrations when detecting vehicle approaching at rapid speed
6) Yellow hazard lights
7) Develop a smart-app that connects to helmet via Bluetooth
    a) Opens another broad class of connected features
    b) Integrate “Hey Siri” + “Ok Google” features
    c) Measure bike time/routes, take phone calls, music, etc.
    d) Cloud updates to helmet software: Server → Mobile phone → Helmet circuit

6. References

[1] Atmel ATmega32p Datasheet. http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf. Accessed on 10/20/2017.

[2] AVRDUDE - AVR Downloader/UploaDEr. “http://www.nongnu.org/avrdude/. Accessed 10/10/2017.

[3] Centers for Disease Control and Prevention (CDC). Web-based Injury Statistics Query and Reporting System (WISQARS). Atlanta, GA: Centers for Disease Control and Prevention, National Center for Injury Prevention and Control. Available at www.cdc.gov/injury/wisqars. Accessed on 09/19/2017.

[4] Force Sensing Resistor Integration Guide and Evaluation Parts Catalog. Interlink Electronics. https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf. Accessed on 10/1/2017.

[5] HT12D Datasheet. http://www.farnell.com/datasheets/57850.pdf. Accessed on 11/1/2017.

[6] HT12E Datasheet. http://www.farnell.com/datasheets/1899539.pdf. Accessed on 11/1/2017.

[7] “IEEE IEEE Code of Ethics.” IEEE - IEEE Code of Ethics www.ieee.org/about/corporate/governance/p7-8.html. Accessed 20 Sept. 2017.

[8] LV-MaxSonar -EZ Series Datasheet. High Performance Sonar Range Finder. https://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf. Accessed on 09/25/2017.

[9] RF Receiver. Wireless Hi Sensitivity Receiver Module (RF ASK). https://cdn.sparkfun.com/datasheets/Wireless/General/RWS-371-6_433.92MHz_ASK_RF_Receiver_Module_Data_Sheet.pdf. Accessed on 09/20/2017.

[10] RF Transmitter. Wireless Hi Power Transmitter Module (RF ASK). https://cdn.sparkfun.com/datasheets/Wireless/General/TWS-BS-3_433.92MHz_ASK_RF_Transmitter_Module_Data_Sheet.pdf

7. Appendix - Original R&V Tables

7.1 User Interface

Requirements	Verification
Variable Force Threshold Switch: 1. When a 3 pound force is exerted on the pressure sensor we should see 5V output out otherwise it is 0V with a tolerance of +/- 25% Figure 10 **15 points**	1a. Connect the resistor to a 5V power source and an ammeter. Pressing the sensor should increase the current that flows through it (No specific information provided on data sheet, must test and set threshold ourselves) 1b. When there is no force applied there should be 0V outputted measured by multimeter 1c. With 3 pounds of force we should see around 5V measured by multimeter
Accelerometer [5]: 1. When turning left the accelerometer on the left handlebar should have a value of -4 g while the accelerometer on the right handlebar should have a value of 4 g and the reverse when the bike is turning right. **10 points**	1a. Use our MCU to read in data from the accelerometer to make sure we our values work out while simulating the a turn caused by the handlebar. 1b. The LED's on the seat should turn off when the turn is almost complete.
Ultrasonic sensor [7]: When there is a car or another bike within 120 inches behind the biker a LED should turn on. We will give this a tolerance of +/- 10 inches. **15 points**	1a. Whenever we place an object within 120 inches of the sensor, with a power source of 5V, pin 3 (analog output) of the Ultrasonic Sensor should output at least a minimum voltage of ~1.2V (from datasheet) 1b. When object is within 120 inches then trigger a set of LED's to turn on.
Switch: 1. Controls power to Collision Prevention LEDs **5 points**	1. Supply power to the collision prevention LEDs (consult figure 10, V_IN) with a 5V source. Ensure flipping the switch turns off the lights, and flipping it back restores them again.

7.2 Control System

Requirements	Verification
Microcontroller [4]: 1. 3.6V at 8mA with a tolerance of +/- 10% 2. Send data across bluetooth at a rate of 4800 bps **25 points**	1. Use a multimeter to test there is a 3.6V with 8mA of current going into the MCU. 2. Send a message from the microcontroller to the RF receiver and use an oscilloscope to read the message from the MCU from the digital out pin. 3. Turn LED on when data from accelerometer indicates a turn has been completed
RF Receiver [6]: 1. Operates at 5V 2. Reads in data at 4800 bps and different messages should trigger different LED's to	1. Use a multimeter to measure there are 5V going into the receiver 2a. Send a message from the microcontroller to the RF receiver

turn on **10 points**	and use an oscilloscope to read the message from the MCU from the digital out pin. 2b. Given these different messages see which LED's turn on.

7.3 Power Interface

Requirements	Verification
1. The power supply will give us 3.7 volts of power that can supply us around 2000mAh with a tolerance threshold of +/- 10%. **10 points**	1. Measure the power supply with a multimeter to see if we are getting 3.7 V with +/- 10%.

7.4 External Interface

Requirements	Verification
1. Specific LED lights should turn on when signal is received. **10 points**	1a. When the right pressure sensor is squeezed then the LED's that correspond to the right arrow and left pressure sensor should trigger the left arrow 1b. When an item is detected by the Ultrasonic Sensor, the Collision prevention lights should light up.