Rear-end Collision Prevention System

By

Kunhao Li

Xiaohang Yu

Jingchao Zhou

Final Report for ECE 445, Senior Design, Fall 2017

TA: Michael Genovese

8 December 2017

Team 32

Abstract

In this project, a rear-end collision prevention system for bicyclist is designed and implemented, which includes a blind spot detection system and a emergency brake light. The former device could detect the presence of objects in six direction behind the bicycle and warn the biker with alternate color circles on a display positioned in the front. The latter could warn the driver behind with Red LED lights that the bike is decelerating sharply. The system passes a subset of the verifications, despite occasional cases of inaccuracies.

Contents

1 Introduction

1.1 Overview

This project aims to decrease the likelihood of rear-end collision, the most dangerous among all types of bike accidents. In particular, the problem is attacked from perspectives of bicyclists and the drivers behind:

- Bicyclists need the accurate perception of their surrounding environment,
- Vehicle conductors behind bicycles should be advised of an imminent encounter of the bicycle.

For cyclists, this project provides a proximity alert of rear-approaching objects based on input from 6 ultrasonic sensors, to advise cyclists not steer in the way of rear-end obstructions. 6 proximity level indicators are shown on the display, which is installed on the handle of the bicycle. If the program decides the distance of a object behind the bicycle changes, one of the indicator circles will change color accordingly.

For approaching vehicles, the project features a brake light, that will flash rapidly when the cyclist does an emergency stop. The decision of whether the bicycle is braking normally or sharply is made based on inputs from an on-PCB accelerometer. The microcontroller will change the output signals to the brake light's combinational control logic, so the brake light LED matrix will switch among 3 states: OFF, ON and FLASH.

The final product of this project is able to display proximity level indicators with correct proximity levels in most instances, but does present incorrect indicators occasionally. The brake light has the correct control logic, but due to a sudden brake down of the accelerometer chip, the brake light was not shown with full capability at the demonstration.

1.2 Background

The League of American Bicyclist researched upon the 628 fatal bike accidents in 2012. One of their conclusions is that among all the collision types, rear-end collision turns out to be the deadliest[1]. It is, therefore, vital for the bicyclists to pay attention to what is behind them when riding their bikes.

Blind spot is certainly one infamous hazard that causes rear end collision. To avoid such hazard, bicyclists have to move their heads to visually check the blind spot area, but such action is purely voluntary and it has to be done with enough frequency at just the correct instant. An aid that continuously supplies critical environment information to human judgment is therefore necessary to minimize the possibility of sudden change in surroundings during the period of inattention.

Such aid always exists for the motorcycles. Car drivers have enjoyed the benefits of LIDAR-based proximity alert systems for long. However, there is no such device in the market yet designed for cyclists. As the number of bicyclist increases over the years. The circumstance justifies a specially designed system to reduce the likelihood of bike accidents, as a check to the expectation.

2 Design

The system is divided into the following modules (Fig. 1):

- Power Module: Supply stable power to all other modules listed below.
- Control Module: Receive input from sensor module, executes algorithm and decide output (control) signal when necessary.
- Sensor Module: Provides distance and acceleration input to the control module.
- Display Module: Show the proximity outputs from control module to the bicyclist.
- LED Module: Container for the brake light matrix, with the matrix's control logic.

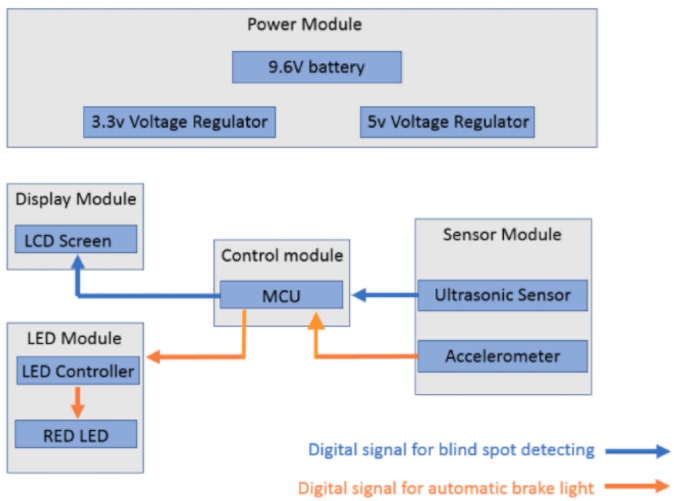

Figure 1: Block Diagram with Digital Signal Flow

For ease of description, sensors are number 0-5, in the order of rear to front, left side to right side (see Fig. 2).

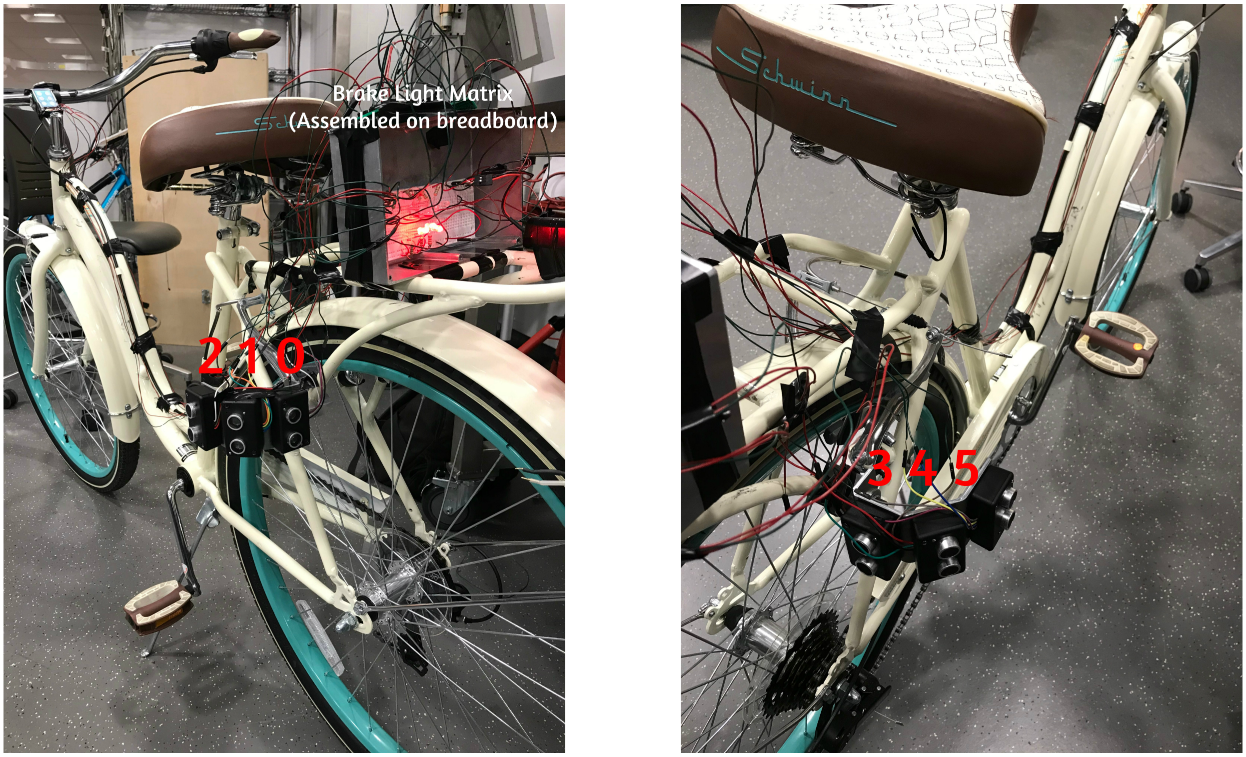

Figure 2: Numbering of Ultrasonic Sensors

2.1 Power Module

2.1.1 Design Procedure

The power module is responsible for providing the correct operating voltages to the entire system. Microcontroller, ultrasonic sensors, logic gates, counters and LED require a 5V input voltage while the requirements of the LED and accelerometer are 3.3V. L7805 is chosen as the voltage regulator for the 5V due to its reliability and price. Since the dropout voltage for the L7805 is 2V, the battery has to provide at least 7V voltage input. From the perspective of power efficiency, a 7V battery is preferred. However, since the output voltage of a battery isn't stable and usually would drop when the battery is low, which would compromise the whole system, a 9.6V battery is chosen over the 7V to ensure reliability. Because of the higher power density and safety, NIMH battery is preferred over the Li-on battery. For the 3.3V, LD1117 is chosen as the regulator due to its low dropout voltage, which would increase the margin of error in case the 5V output voltage from the previous stage is not steady. Originally two unity gain buffer was planned to be implemented at the output of the voltage regulators to achieve better load regulation. However, since all the components used in this project have a high tolerance in supply voltage, the two unity gain buffer are forfeited to trade off the simplicity and cost.

In an effort to enhance the stability and reduce ripple voltage, a 0.33uF capacitor is implemented between input node of the L7805 and ground and a 10uF capacitor is connected between the output node and the ground. The two capacitors form a low pass filter to filter out undesired AC signal. For the same reason, a 0.1uF capacitor and a 10uF capacitor are connected at the input and output of the LD1117 respectively.

2.1.2 Design Detail

Fig. 3 is the schematic for the power module. Both voltage regulators adopt the device model of L7805 since they both use TO-220 package and have similar pin assignment.

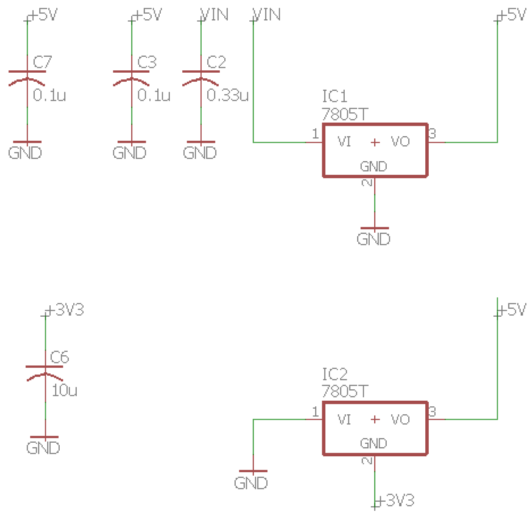

Figure 3: Power module schematic

2.2 Control Module

2.2.1 Design Procedure

The control module is the centerpiece that processes sensor data and decides bike movement state, presence of objects if any, and proximity to these objects. To facilitate easier debugging, a module construct with Arduino platform support is preferred. Two constructs, Arduino Uno and Teensy LC, are therefore selected.

Arduino Uno is a widely used development board and features a ATMega328P (Atmel AVR) microcontroller. Using Arduino's configuration, the chip can provide 14 digital signal pins and 6 analog signal pins[2]. A great advantage of using this construct is that the microcontroller can be detached from an Arduino, integrated with an external peripheral circuit, and become readily usable. However, the microcontroller is limited in terms of speed, running at maximum of 20MHz under 5.5V[3].

Teensy LC is a cheap yet powerful development board, utilizing a MKL26Z64VFT4 (ARM Cortex-M0) microcontroller. Since Teensy LC, Teensy is compatible with most programs that are originally intended for Arduino. Teensy's configuration of the chip provides 27 digital pins and 13 analog pins[4]. Speed rating of this chip, 48MHz at 3.6V[5], is more preferable than that of ATMega328P. A significant drawback of this construct is the need of another bootloader chip, so 2 chips must coexist on the PCB design. Because the microcontroller is only available in QFN and LQFP packages, it cannot be easily detached to re-burn the program, so debug ports must also be built into the PCB design, increasing PCB footprint.

We decided that the ease of debugging and integration is the most important factor in the development phase, so an Arduino Uno was used during development. The development board's microcontroller should be detached and moved to our own PCB for final testing.

2.2.2 Design Detail

The control module PCB is combined with the power module. The microcontroller section (Fig. 12) on schematic is modified from Arduino's original board design[6]. The PCB gets 5V supply from the power module, and features a pinout of 14 digital pins, 6 analog pins, the same as that of Arduino Uno. A 16MHz crystal supplies a clock signal at the same frequency.

The program is implemented as single-threaded, containing the following classes:

- `SensorArray`: This class represents an array of 6 ultrasonic sensors that are installed around the bike. It triggers the TRIG signal each time it is activated, monitors all 6 echo signals and calculate object distance based on time of ECHO's rising and falling edges. It contains an decision algorithm specified in Fig. 16.
- `BrakeLight`: This class receives analog input from accelerometer to decide the bike's current state of movement. Using pre-defined thresholds, it changes operation mode of the LED matrix by varying the two control signals EN and LIT (ref. 2.5.1). Operation of this class follows Figure 17.

The two classes are each activated once per iteration of the main loop.

2.3 Sensor Module

2.3.1 Design Procedure

The sensor module serves as a collection of the sensors used in this project. For the blind spot warning system, the most financially viable sensor is the HC-SR04 ultrasonic sensor, whose cost of \$5.95 per unit[7] is significantly less than \$100 and above[8] for LIDAR sensors.

Using multiple HC-SR04 sensors to detect different locations, the project features 6 ultrasonic sensors in total. Each side of the rear wheel will have 3 sensors covering 3 separate locations. Triggering a sensor to send out ultrasonic wave bursts requires raising the trigger digital signal (TRIG) to HIGH for at least 10μs. When each ultrasonic sensor detects the existence of an object, i.e. ultrasonic waves are reflected on the object and received, it raises its ECHO digital signal. The duration of ECHO being HIGH in milliseconds is proportional to the distance between that sensor and the object. Conversion from signal HIGH duration to object distance is done by

$$D_{object} = \frac{T \times 340}{2} \approx \frac{T}{0.0058} \tag{1}$$

where D_{object} is the real time distance of the approaching object, 340 is the approximated speed of sound at sea level and 5°C and T is the duration of ECHO being high in **seconds**. Division by 2 accounts for the round-tip time.

A timeout is required to mark the lack of objects in the vicinity, after TRIG is raised. With a desired radial distance $D_{radial} = 3$m, the timeout should be

$$t_{timeout} = \frac{2 \times D_{radial}}{340} = \frac{2 \times 3}{340} \approx 0.018s \tag{2}$$

Accounting for potential latency caused by function overheads in the program, the timeout used in controller program is set to 20ms.

With 20ms timeout, the worst-case time necessary to trigger and wait for ultrasonic sensor is $6 \times 20 = 120$ms. If a tailing vehicle moves at speed 30 mph, while the bike moves at 10 mph, within 120ms, the distance in between can reduce by $0.12 * 63360 * \frac{30-10}{3600} = 42.24$ inches, more than half the length of a 52-inch bike. Therefore, we decide to use a unified bus to provide TRIG signals for all 6 sensors, so all sensors would send out ultrasonic wave at the same time.

For the brake light, an accelerometer is used (Fig. 13). This sensor is positioned on the brake light's control logic PCB, which is fixed above the rear wheel of the bike. The z-axis is made parallel to the span of the bike, with positive z-axis pointing to the rear. The axis are interchangeable, but should another axis is selected for use, the direction of the axis must be fixed and changes must be made in microcontroller code.

Based on the datasheet of the accelerometer[9], the conversion relationship between the acceleration and the real time voltage is 360mV/g, where g is the gravitational constant. Therefore

$$Acc = \frac{(Z_{value} - Z_{base}) \times 4.9mV}{360mV} \times 9.8m/s^2 \tag{3}$$

where Acc is the acceleration; Z_{value} is the real time analog read value of the z axis; and Z_{base} is the basic value of z axis when the bike does not accelerate or decelerate. The factor of 4.9mV comes from the 4.9mV/unit characteristic of `analogRead`[10]. The difference between Z_{value} and Z_{base} should be 40 units to make the lights flash.

2.3.2 Design Detail

On each side of the bike, a metal plate (Fig. 4) is affixed to the frame of the bike. The metal plate is bent, forming 3 segments on the surface. Each segment would hold exactly 1 sensor.

Each sensor obtains TRIG signal from a bus but connects to the microcontroller in separate wires. Overall, all ultrasonic sensors would require 7 digital signals connected to the microcontroller (Table 1). In the later phase of this project, irregularities of the digital HIGH on TRIG bus were detected, distorting the supposed rectangular wave pattern. The sensors did not perceive the distorted TRIG signal as a proper trigger, so no ultrasonic wave was sent. Thus, a low-pass filter was added between the TRIG bus and the ground, so that high-frequency signals in the TRIG bus were removed, and sensor functionality was back to normal.

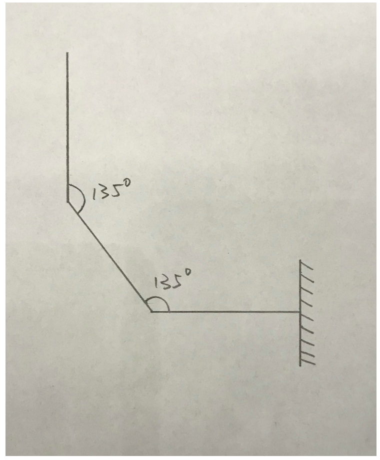

Figure 4: Sketch of Sensor Mount

Table 1: Pin Assignment for Ultrasonic Sensor

Name	Abbreviation	To MCU Pin
Trigger Signal	TRIG	12
Echo Signal 0-5	ECHO	4, 5, 6, 7, A0, A1
Voltage Input 5V	Vcc	N/A
Ground	GND	N/A

The accelerometer PCB is modified based on the SparkFun breakout chip[11], ADXL335. It provides three analog output pins, for x, y and z axis. The voltage supply is 3.3V. We only use the z axis output to calculate acceleration 3.

Table 2: Pin Assignment for Accelerometer

Name	Abbreviation	To MCU Pin
Voltage Input 3.3V	Vcc	N/A
Ground	GND	N/A
Analog Output of X axis	X_OUT	N/A
Analog Output of Y axis	X_OUT	N/A
Analog Output of Z axis	Z_OUT	A5
Self Test	ST	N/A

2.4 Display Module

2.4.1 Design Procedure

The display module is part of the blind spot warning system, and should provide visual indications to the cyclist of objects behind. TFT LCD was chosen instead of typical 16x2 LCD due to the location of this module. As a visual indicator, the display must be in front of the cyclist on the bike's handle. Considering the limited space on any bike's handle, the display should preferably not exceed a dimension of 7x7x0.8 (cm). Typical 16x2 LCD displays do not satisfy this constraint, due to their length.

An 1.8" TFT LCD display from Adafruit is eventually selected. This particular type of display is especially suitable for the space constraint mentioned in previous subsection. Its 3.4x5.6x0.65 (cm) dimension[12] satisfies our 7x7x0.8 (cm) expectation. Also, the TFT display is able to display more intricate shapes than a black-and-white 16x2 LCD display. This TFT display communicates with the microcontroller using *Serial Peripheral Interface Bus (SPI)*, for which Arduino provides an SPI communication library.

The information conveyed by the display should be simple and clear, while able to show updates as soon as new contents are available. More intuitive than just words, a graphical user interface is preferred, but an elaborate interface design would slow down the time taken for microcontroller to refresh all necessary pixels on the screen. The trade-off of user interface indicates that the code written for the display module would have implications to whether the display module would be able to satisfy the necessary constraints.

2.4.2 Design Detail

Although a breakout board is available from Adafruit, we decide that the breakout contains too much extra functionality, such as a SD card reader that can supply bitmap data directly to the display. A standalone TFT display is soldered onto a customized PCB (Fig. 11), which is modified from Adafruit's design of the LCD breakout[13]. The customized board design requires 5 input signals, with 2 voltage supplies and 1 ground pin (Table 3), each connected to a through-hole header. The design also contains a voltage buffer to prevent the LCD's load from interfering with the external circuitry.

Table 3: Pin Assignment for Display PCB

Name	Abbreviation	To MCU Pin
Serial Clock	SCLK	13
Master Output Slave Input	MOSI	11
TFT Chip Select	TFT_CS	10
Data/Command Select	D/C	8
Reset	RESET	9
Backlight Voltage Supply (3.3V - 5V)	LITE	N/A
Control Logic Voltage Supply (3.3V)	Vcc	N/A
Ground	GND	N/A

The user interface is purely graphical, using 6 filled circles to indicate 6 sensor locations behind the bicycle. Different fill colors correspond to different proximity ranges as in Table 4. Each circle has a fixed center, and its fill color is only updated when the proximity level of that particular location has changed. The radius of each circle is set at 18 pixels (Fig. 5), a value that makes the circles visually identifiable.

Table 4: Color Mapping for Proximity Ranges

Color	Proximity Range (m)
Red	< 1
Yellow	1 - 2
Green	2 - 3, or object not present

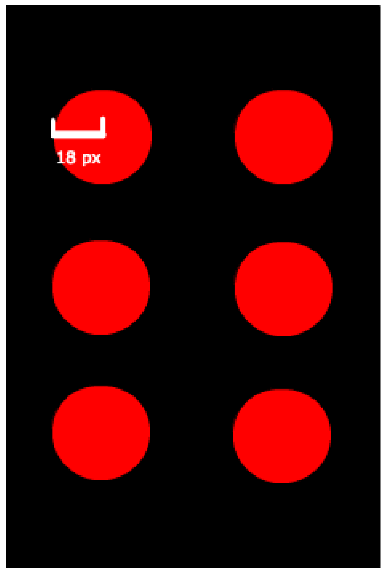

Figure 5: Example of Display User Interface

2.5 LED Module

2.5.1 Design Procedure

The LED module is a 2x4 matrix of red LEDs, functioning as the brake light. This module will be installed on the rear-rack (Fig. 2). The two LED rows are numbered 0 (top) and 1 (bottom), each array defined as a sequence of 4 LEDs.

The LED module should have 3 modes of operation:

- OFF: Bike is not reducing speed. All LEDs off.
- ON: Bike is reducing speed by cyclist gently applying brakes. All LEDs on.
- FLASH: Bike is abruptly reducing speed. Array 0 and array 1 of LEDs alternatingly switches on and off.

To make the LED array flash, a timer chip can be used to emulate a periodic signal. To achieve a more fine-grained control over the frequency of oscillation, the signal can act as the clock for a counter chip, so using different bit outputs of the counter is equivalent to applying a divider to the timer frequency. Then, array 0 of LED matrix is connected directly to one of the bit output, called A_0, while array 1 is connected to the inverse of the bit output, called A_1, so the two rows have only one row of LEDs lit.

Thus, 3 signals in total should participate in the operation of brake light LEDs:

- EN: Enable signal. LEDs are always off when EN is LOW. If EN is HIGH, LEDs is either ON or FLASH.
- LIT: When enabled (HIGH EN), the LEDs should be always on when LIT is HIGH, i.e. ON mode.
- $COUNT$: Counter input used for alternate flashing, under FLASH mode.

These inputs are transformed into 2 outputs fed to 2 rows of LED lights, respectively:

- A_0: Voltage input for row 0 of LED matrix.
- A_1: Voltage input for row 1 of LED matrix.

With desired behavior as in Table 5, the logic equations between these 3 inputs and 2 outputs are:

$$A_0 = EN \cdot (COUNT + LIT) \tag{4}$$

$$A_1 = EN \cdot (\overline{COUNT} + LIT) \tag{5}$$

Table 5: Behavior of Brake Light Control Logic

EN LIT	A_1 A_0	Operation Mode
0x	00	OFF
10	01/10	FLASH
11	11	ON

2.5.2 Design Detail

The PCB design for brake light (Fig. 14) implements the control logic as described by equations 4 and 5, using a quad-input AND gate, individual OR gates and inverters.

The PCB requires 2 digital signals, EN and LIT, from microcontroller to control the operation mode of the brake light matrix (Table 6). Because all parts used in the design are 5V-tolerant, a 5V pin is used to connect to an external source of power to this PCB.

Table 6: Pin Assignment for Brake Light Control PCB

Name	Abbreviation	To MCU Pin
Voltage supply (5V)	Vcc	N/A
ON/FLASH switch	LIT	3
ON/OFF (enable) switch	EN	2
Ground	GND	N/A

LM555 timer chip is selected as clock signal to a 74LS193N 4-bit counter. The output frequency f_{Timer} of LM555 can be tuned by varying R_1 and C_1 (see Fig. 14) under astable mode to obtain clock signals at different frequencies[14], described by[15]:

$$f = \frac{1}{T} = \frac{1.44}{2R_1C} \tag{6}$$

By setting R_1 to 1MΩ and C_1 to 0.01μF, the output frequency of the timer is

$$f_{Timer} = \frac{1.44}{2R_1C} = \frac{1.44}{2 \times 10^6 \times 10^{-8}} = 72\text{Hz}. \tag{7}$$

Output of the second most significant bit of the counter is used, equivalent to dividing f_{Timer} by a factor of 8. Thus, the flash frequency f_{Flash} is

$$f_{Flash} = \frac{f_{Timer}}{8} = \frac{72}{8} = 9\text{Hz} \tag{8}$$

which is a visually acceptable flash frequency for a row of LEDs.

3 Design Verification

Verification failures in this project are mostly related to sensor distance inaccuracies that affect the controller, sensor and display modules. Besides sensor issues, the power module experiences ripple voltages at the output. For the following subsections, italicized sentences are requirements that were not satisfied at the time of demonstration.

3.1 Power Module

The ripple voltage at the output of 5V regulator should be smaller than 0.1V (2 pts).

The ripple voltage at the output of 3.3V regulator should be smaller than 0.1V (2 pts).

Verification of the power module is performed by probing the outputs of both voltage regulators with the oscilloscope.

Fig. 6 shows the measurement taken at the output port of 5V regulator when the micro-controller is off. This is the situation that the power module isn't driving any load. It can be clearly seen from the picture that power module is able to provide a steady 5v voltage with acceptable noise level (4.9V - 5.1V).

Fig. 7 shows the measurement at the same node taken when the micro-controller is on. The voltage ripple deteriorate by around 0.5V. One reason behind this deterioration is that the two power modules are cascaded together and were connected directly into the circuit, so the load regulation is not optimized. When the microcontroller runs programs, the load inevitably changes and the voltage ripple increases. Another reason that might cause this error comes from the layout. During the debug process, it turned out the display impacts the stability of the power module most. On the PCB, the port of the microcontroller that connects to the display lays right beside the analog 5V port, which is connected directly to the 5V voltage regulator. Since the microcontroller periodically sends signal to the display, it is likely that the AC component of the current in the display port induces AC current in the 5V port, which feeds back to the power module, causing the problem. Same rationale applies to the ripples of 3.3V voltage regulator.

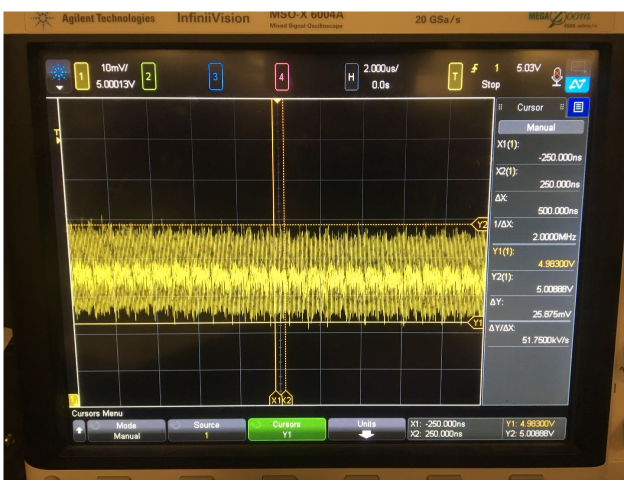

Figure 6: Output 5v voltage without the load

Figure 7: Output 5v voltage with the load

3.2 Control Module

Microcontroller should output correct proximity level to screen at the corresponding location when an object is within the range of 3 meters (9 pts).

Verification of this requirement is done by observing proximity color on the display. During demonstration, sensors 0 and 3 (ref. Figure 2) exhibits intermittent shifts between red color and green color. This behavior indicates that the decision algorithm gets fluctuating distances that fall in the range of different color mappings (described in Table 4), so the requirement is only partially verified. As the centerpiece of the decision algorithm, the microcontroller should possess the ability to filter out abnormal distance values, calculated from ultrasonic sensors' ECHO signal. However, the sensor program does not have this functionality built-in, so wrong proximity levels are sometimes put to the display.

The remaining requirements associated with brake light system also failed to verify at demonstration, due to a broken accelerometer chip (see Subsection 3.4). After the demonstration, the on-PCB accelerometer was bypassed to a breakout board. Then, the requirements related to brake light system could be verified as correct.

3.3 Sensor Module - Ultrasonic

Sensor should detect objects within range of 3 meters (2 pts).

The ultrasonic sensor is able to detect objects within range of 3 meters, but the detection is not always accurate. Per the requirement and its verification, we should get an accurate reading of the distance between the sensor and the object, with an error tolerance of ± 5cm. However, at demonstration, a green indicator was shown if a person standing at approximately 1.8 meters away, while the correct indicator should be yellow. Rapid shifts between yellow and green indicators also occurred, suggesting an error of at least 20cm.

3.4 Sensor Module - Accelerometer

Should output higher/lower voltage as acceleration increases (1 pt).

The accelerometer chip was broken during our demonstration, so the requirement could not be verified. Due to the timing of its breakdown - the weekend before our Monday demonstration, when no suppliers were open, and the lack of spare accelerometers, no fix could be done even after the problem was discovered on Saturday. The cause is believed to be incorrect wiring, as the brake light system was functional when the circuit was rebuilt after demonstration, and the on-PCB accelerometer was bypassed to a new accelerometer on a breakout board. The on-PCB accelerometer remains broken.

3.5 Display Module

Display content should indicate the correct proximity level of approaching objects from different directions (10 pts).

Verification shows that our system only partially satisfies this requirement. Due to inaccuracies in the ultrasonic sensor itself and lack of data filtering in the microcontroller program, the proximity level indicators on the display do not always represent the actual proximity. Intermittent shifts between colors can occur to a subset of indicators.

The requirements rules that new display content should appear on display within 80ms, one that our display module satisfies. By enclosing display update function calls inside timing code, we obtained data points of total time needed to update display content in a 10-second time span (Fig. 8). It can be observed from the plot that none of the data points exceed 80ms in the 10-seconds time span.

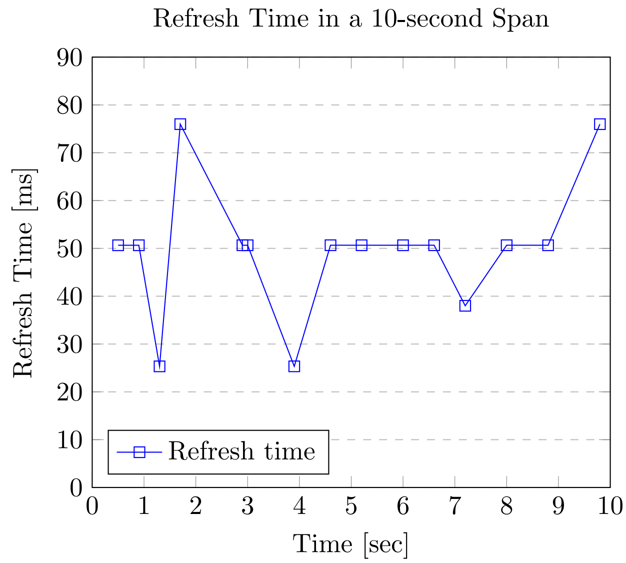

Figure 8: Refresh Time in a 10-second Span

3.6 LED Module

All requirements of the LED module are fully verified, but some characteristics of the final product exhibits deviation from our design calculations. In particular, the flash frequency (output frequency of A1/A0) is 7.8Hz (Fig. 9), less than the expected value 9Hz (see Equation 8), although still satisfies the minimum requirement of 5Hz. The cause is the less-than-expected frequency f_{Timer} of the timer output, having only 62.8MHz (see Fig. 10). The cause of this observation is attributed to 0.01μF capacitor tolerance ($\pm$10%) and 10MΩ resistor tolerance ($\pm$1%), that can affect the actual frequency configuration to the capacitor.

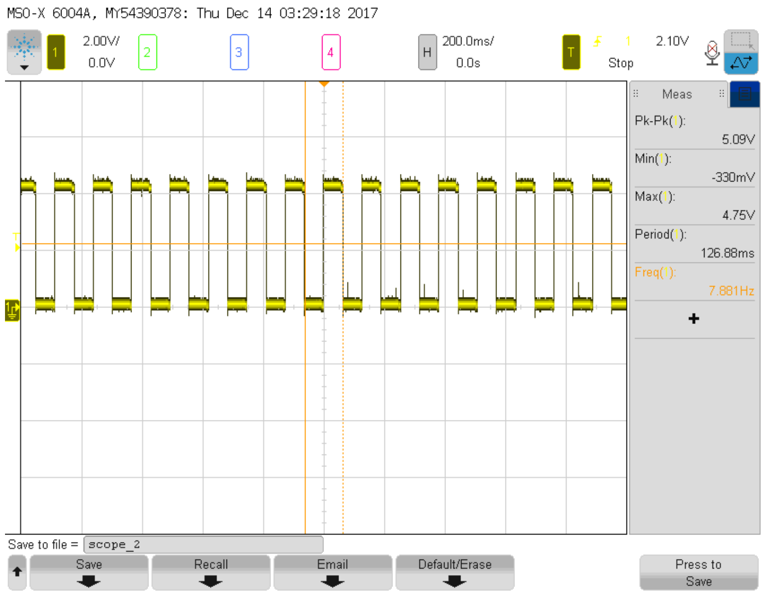

Figure 9: Wave form and Measure of LM555 Timer Output

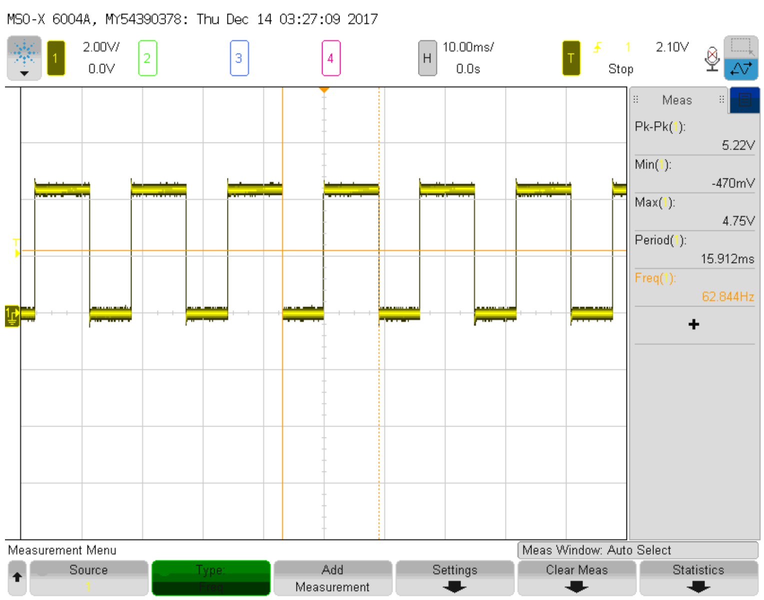

Figure 10: Wave form and Measure of LM555 Timer Output

4 Cost

4.1 Parts

Table 7: Parts Costs

Part	**Manufacturer No.**	**Retail Cost ($)**	**Bulk Purchase Cost ($)**	**Actual Cost ($)**
Microcontroller	ATMEGA328P-PU	2.14	2.14	4.28
Resonators	CSTCE16M0V53-R0	0.46	0.46	2.30
Diodes	CD1206-S01575	0.15	0.128	1.28
Linear Voltage Regulators	NCP7805TG	0.43	0.43	2.15
Ultrasonic Sensor	HC-SR04	5.95	5.95	35.70
Resistors 1K	CR0805-FX-1001ELF	0.10	0.061	1.22
Resistors 1M	CR0805-FX-1004ELF	0.10	0.037	0.74
Capacitors 0.1uF	C0805C104J5RACAUTO	0.22	0.089	3.56
Voltage Regulators 3.3V	LD1117AV33	0.90	0.744	7.44
Capacitors 0.33uF	C0805C334K3RACAUTO	0.27	0.126	2.52
Capacitors 10uF	C1206X106J3RACAUTO	1.24	0.871	17.42
4 bit Counter IC	SN74LS193N	1.10	0.904	9.04
AND Gates Quad 2-Input	SN74HCT08D	0.42	0.31	3.10
Inverters Single 2-input	MC74HC1G04DFT1G	0.34	0.228	2.28
OR Gate Single 2-input	SN74AHC1G32DCKR	0.36	0.271	2.71
Timer	LM555CMX/NOPB	0.99	0.99	4.95
Capacitors 0.01uF	MM051C103KCZ2A	0.99	0.949	18.98
Voltage Buffer	CD74HC4050M	0.54	0.52	2.60
BJT Bipolar Transistors	MMBT 2222A LT1	0.20	0.174	3.48
Resistors 22Ohm	ERJ-P06F22R0V	0.28	0.112	2.24
PCB MCU	PCBway	1	1	5
PCB Accelerometer	PCBway	1	1	5
PCB LCD Screen	PCBway	1	1	5
Total				142.99

4.2 Labor

Table 8: Labor Calculation

Name	Labor
Kunhao Li	\$40/hour $\times$ 2.5 $\times$ 150 hours = \$15000
Jingchao Zhou	\$40/hour $\times$ 2.5 $\times$ 150 hours = \$15000
Xiaohang Yu	\$40/hour $\times$ 2.5 $\times$ 150 hours = \$15000
Machine Shop	\$50/hour $\times$ 2.5 $\times$ 150 hours = \$18750
Total	\$63750

5 Conclusion

5.1 Accomplishments

This project provides a blind spot warning system that can alert the presence of objects to a cyclist. With proximity level indicators, cyclists are able to know the distance to the approaching object, but with inaccuracies due to deficiencies in the design.

With bypasses made as in subsection 3.4, the brake light can provide a early visual cue to the drivers behind the bike when the bike brakes. The LEDs are on the when cyclist brakes normally, and flashes when the cyclist makes an emergency stop.

5.2 Uncertainties

Accuracy is the most significant cause of verification failures, which are usually manifested as incorrect proximity level indicators. Sensor inaccuracies can affect the calculated range between approaching objects and the bike, while the program does not filter out irregular sensor data. Exact cause, such as noise or wave reflection, of out-of-tolerance values needs to be investigated, in order to increase the overall usability of the entire system.

It is unknown if the reading accuracy of ultrasonic sensors can be improved by using a microcontroller that supports more interrupts. Because of the limited number of interrupt pins available in ATMega328P, the program does not use interrupt to react as soon as sensor ECHO signal exhibits an edge. If the polling program model is changed to interrupt model, edge capture of ECHO signals may be more real-time and calculations could be more accurate.

5.3 Ethical Considerations

Physically, the system requires installations on the handle (LEDs) and rear parts of the bicycle frame (controller and sensors). Modules on the rear frame will extend wires through the whole middle frame to connect LEDs on the handle. Such wiring, if not carefully located and adhered, may be tangled in the wheels during cycling activities, and causes danger. The teams choice to install sensors on the seat stay is the result of consulting with ECE Machine Shop, in which both parties agreed that wiring only needs to be adhered to the bicycle frame, without going near the wheels and derailleurs.

The metal cases used to contain the brake light matrix can pose a electricity hazard, most notable of which is static electricity. This method of containment should not go into production, and must changed to insulator materials such as custom-made plastic cases.

We are responsible for making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment through the whole project. This is in accordance with the Section 7 IEEE Code of Ethics[16].

5.4 Future Work

There are several improvements that could be implemented in the future:

- Since the long wires across the middle frame would likely cause accidents when tangled in the wheel, it is desirable that in the future, a Bluetooth module would be added to handle digital communication between display, control and LED modules.
- With the aid of the Bluetooth module, the system could communicate with the user's phone, which would allow control of the system from cell phones, and potentially opens up possibility to synchronize cycling data to cloud.
- Currently the system consists of 2 PCBs and 1 breadboard, in the future, they could be merged as one PCB to make installation easier
- Due to the static electricity hazard of the iron box that holds the PCB, a wooden or plastic box with a more stable mount is a better option.

References

[1] *Bicyclist safety must Be a priority*, http://bikeleague.org/sites/default/files/EBC_report_final.pdf, League of American Bicyclists, 2014.

[2] "Arduino Uno Rev3," https://store.arduino.cc/usa/arduino-uno-rev3, 2017.

[3] *ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH*, Atmel Corporation, San Jose, CA, 2015.

[4] "Teensy LC (Low Cost)," https://www.pjrc.com/teensy/teensyLC.html#specs, PJRC.com, LLC., 2017.

[5] *Kinetis KL26 Sub-Family 48 MHz Cortex-M0+ Based Microcontroller*, Freescale Semiconductor, Inc., Austin, TX, 2014.

[6] *Arduino(TM) Uno Rev3*, Arduino, 2017.

[7] "HC-SR04 Ultrasonic Distance Measuring Transducer Sensor Module for Arduino," https://www.ebay.com/itm/HC-SR04-Ultrasonic-Distance-Measuring-Transducer-Sensor-Module-for-Arduino/191841217528, 2017.

[8] "LIDAR-Lite v3," https://www.sparkfun.com/products/14032, 2017.

[9] *ADXL335 Small, Low Power, 3-Axis 3g Accelerometer*, Analog Devices, Inc, Norwood, MA, 2009.

[10] "Arduino Reference," https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/, Arduino, 2017.

[11] "SparkFun Triple Axis Accelerometer Breakout - ADXL335," https://www.sparkfun.com/products/9269, SparkFun, 2017.

[12] "1.8" SPI TFT DISPLAY," https://www.adafruit.com/product/618, Adafruit Industries, 2017.

[13] "EagleCAD PCB files for TFT breakout," https://github.com/adafruit/1.8-TFT-breakout, Adafruit Industries, 2017.

[14] "How to Build a Clock Circuit with a 555 Timer," http://www.learningaboutelectronics.com/Articles/555-timer-clock-circuit.php, 2017.

[15] *LM555 Timer*, Texas Instruments, Dallas, TX, 2015.

[16] *IEEE Code of Ethics*, https://www.ieee.org/about/corporate/governance/p7-8.html, IEEE, 2017.

Appendix A Requirement and Verification Table

Table 9: System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
The voltage supplied by the 9.6V battery should remain above 7V for more than 7 hours. (1 pt)	Attach multimeter to cathode and anode. Read voltage value from multimeter, and check its above 7V for 7 hours.	Y
The ripple voltage at the output of 5V regulator should be smaller than 0.1V. (2 pts)	Connect the output pin of the 5V regulator to the oscilloscope and check if the output voltage is within 4.9V - 5.1V.	N
The ripple voltage at the output of 3.3V regulator should be smaller than 0.1V. (2 pts)	Connect the output pin of the 3.3V regulator to the oscilloscope and check if the output voltage is within 3.2V - 3.4V.	N
Microcontroller should output correct proximity level to screen at the corresponding location when an object is within the range of 3 meters. (9 pts)	Write a function testAlgorithm for the decision algorithm: • For a sensor 1, set the previous distance measurement to 3.0, and provide a mock new measurement of 2.5. • For the adjacent sensor 2 to the forward direction of the bike: Set previous measurement to 3.2, and provide a mock new measurement of 2.7. • Assert that after one iteration of the loop (described above), microcontroller outputs a green proximity level for sensor 1 on screen.	N
The two brake light arrays should flash alternately when the deceleration of bicycle exceeds the BRAKE_SHARP, a parameter we will find through experiments. (4 pts)	Ride bicycle at a 10 mph, and full brake. Observe whether the brake light flashes.	N
Both the brake light arrays should be on when the cycler brakes under a normal braking circumstance, in which the deceleration exceeds BRAKE_NORMAL but does not exceed BRAKE_SHARP. (5 pts)	Ride bicycle at 10 mph, brake the bike and stop when arriving at the destination, like a normal and non-emergency stop. Observe whether the brake lights are all on and not flashing.	N

Continued on next page

Table 9 – continued from previous page

Requirement	Verification	Verification status (Y or N)
Sensor should detect objects within range of 3 meters. (2 pts)	Connect TRIG and ECHO signals to digital pins of an Arduino. Put a piece of paper in front of the sensor, with initial separation of 4 meters. Gradually move paper closer to the sensor. After actual distance decreases below 3 meters, check if arduino reports calculated distance within actual range 0.05m.	N
For a 3 meters range, a sensor should raise ECHO return signal to HIGH in no more than 20ms after TRIG is set to HIGH (ultrasonic pulse is sent). (1 pt)	Connect TRIG and ECHO signals to digital pins of an Arduino. Put a piece of paper in front of the sensor with separation no more than 3m. Write test code that writes 1 to TRIG and asserts that ECHO is raised to 1 within 20ms.	Y
Should output higher/lower voltage as acceleration increases. (1 pts)	• Connect accelerometer to Arduino • Write a program testAcc that repeatedly print returned x-, y- and z-direction voltages to the serial monitor. • Hold accelerometer in hand and swing arm in accelerometers native positive y-direction. • Assert that the value on serial monitor for y-direction increases.	N

Continued on next page

Table 9 – continued from previous page

Requirement	Verification	Verification status (Y or N)
New display content should appear on display within 80ms.(2 pts)	• Connect display module to a dev board. • Write a function testDrawCircle that uses drawCircle library call to draw a circle on LCD with radius 20 pixels. The fill color of the circle should alternate among red, green and blue for different calls to testDrawCircle. testDrawCircle records timestamp before and after the call to drawCircle, returning the time difference in milliseconds. • Call testDrawCircle 100 times, summing up returned values and divide by 100 to get average time to draw the circle. Assert the average time is within 80ms	Y
Display content should include the correct proximity level of approaching objects from different directions. (10 pts)	• Create function testDisplayProximity. • – Mock distance from sensor X are [2.8, 1.6, 0.8], while distance from sensor X+1 are [3.0, 1.9, 1.0]. – Feed above sequence of values to algorithm in MCU. – Assert that proximity levels at sensor 1s corresponding location on display changes from green to red (see figure 2.3.4.1).	N
The flashing should be noticeable from a distance of more than 5 meters. (2 pts)	Ride bicycle at 10 mph. Brake sharply at a marked location. Another person should stand 5 meters away from the location, and notice whether the flashing is noticeable.	Y
When flashing, the LED light array should have an on-off frequency of at least 5 times per second. (1 pts)	Count the number of flashes in 10 seconds, and calculate the average number of flashes in one second. Check whether the result is greater than 5.	Y

Continued on next page

Table 9 – continued from previous page

Requirement	Verification	Verification status (Y or N)
Correct behaviors are exhibited with inputs specified in table 5 (8 pts)	Assemble control logic and connect with LED arrays. Supply different input signals and assert that behaviors are same as specified in table 5	Y

Appendix B Schematics

This section contains the relevant schematics mentioned in design details. The first schematic starts at next page.

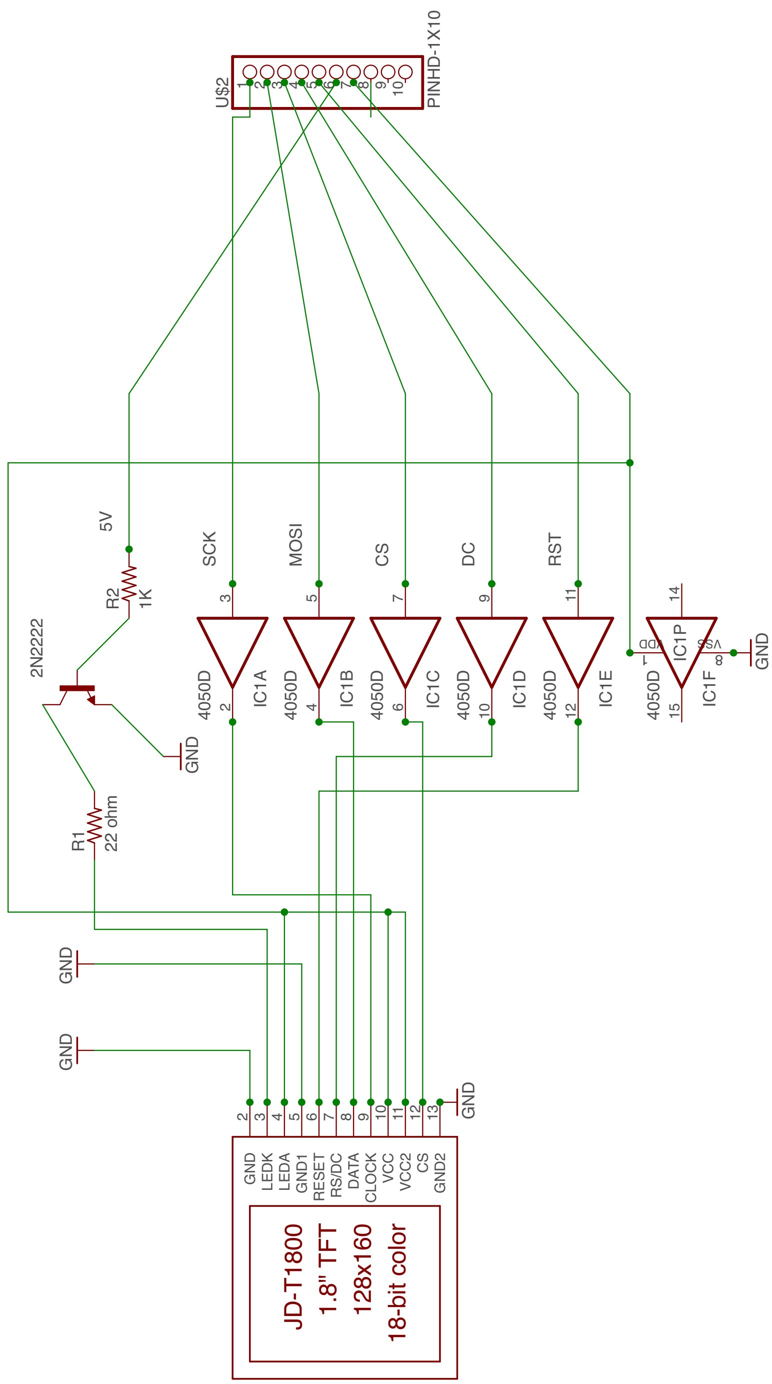

Figure 11: Schematic for Display Module

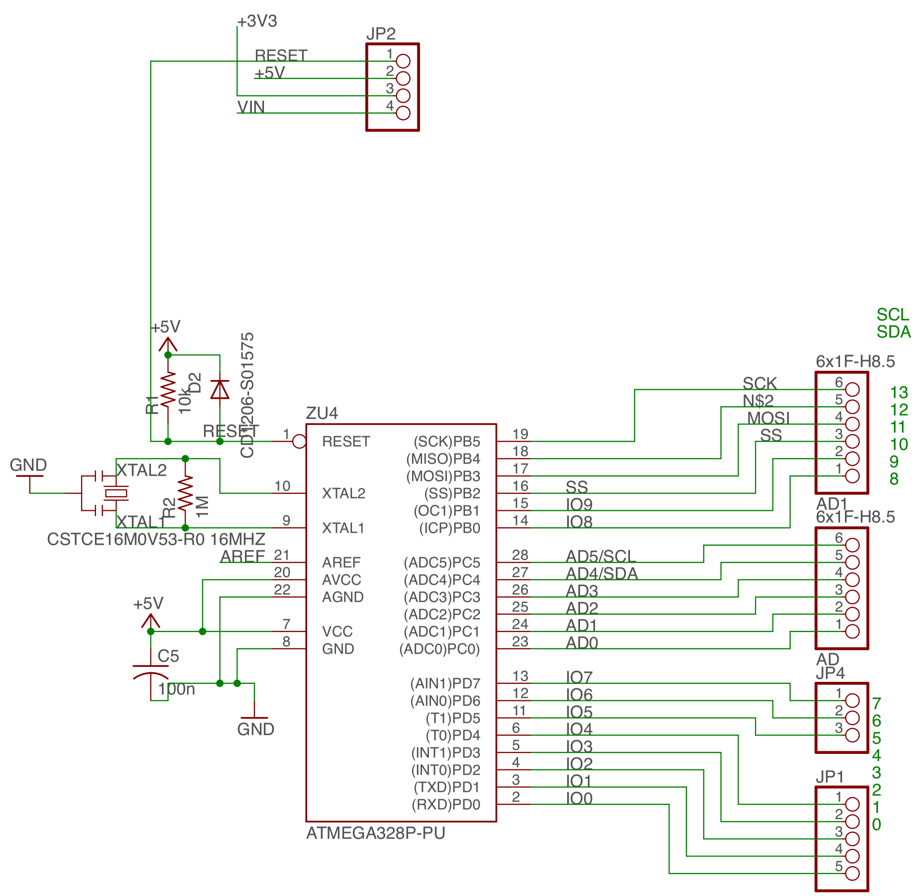

Figure 12: Schematic for Controller Module

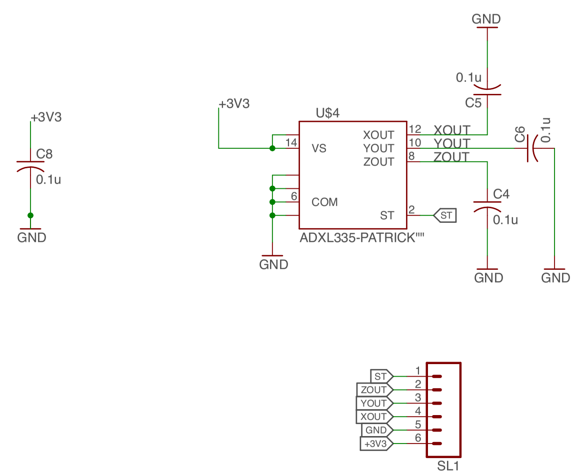

Figure 13: Schematic for Accelerometer (on Brake Light PCB)

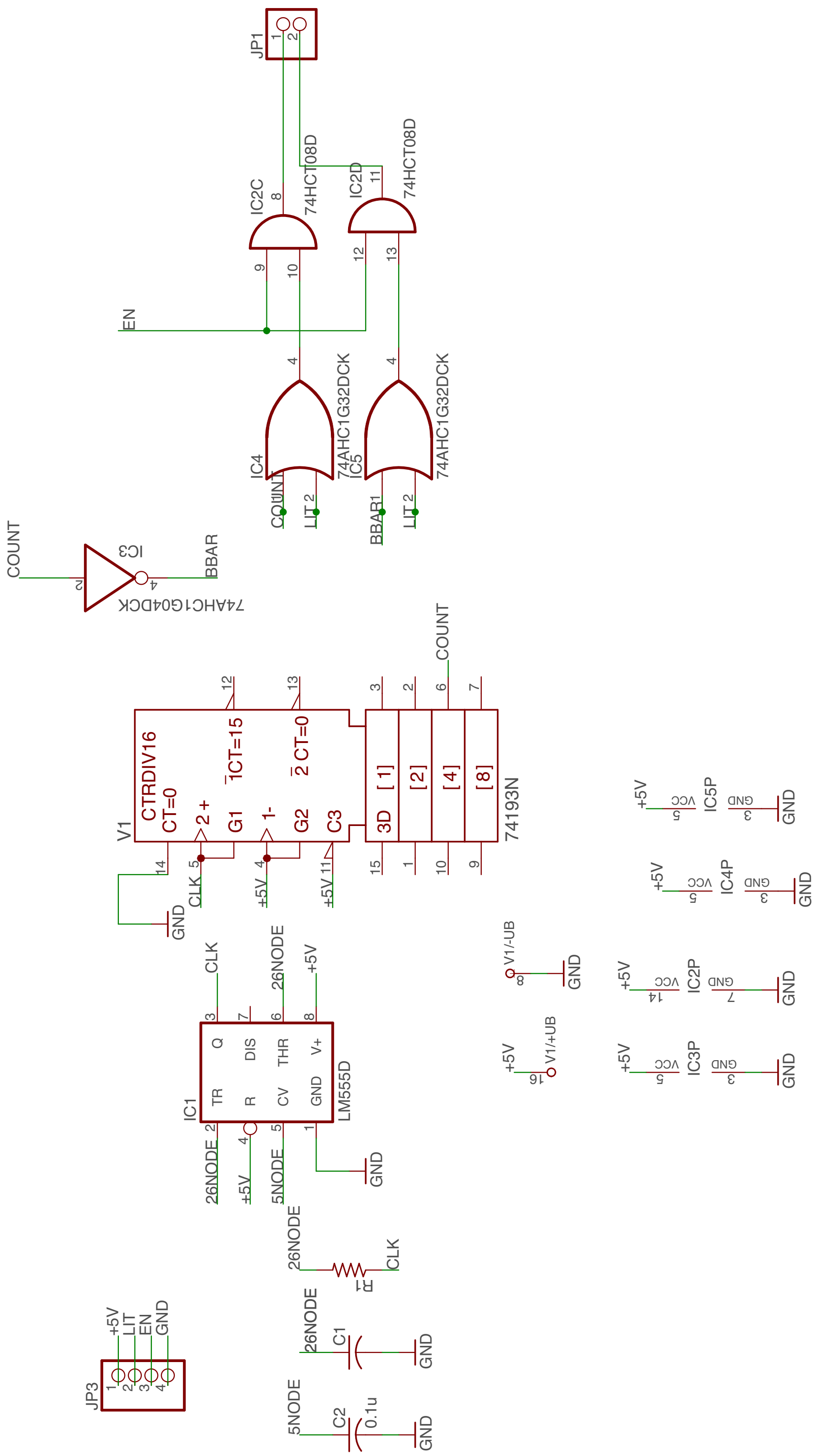

Figure 14: Schematic for Brake Light Control PCB

Appendix C Program Flowcharts

This section presents the algorithm flowcharts for the control module program. All flowcharts use legend as in Figure 15.

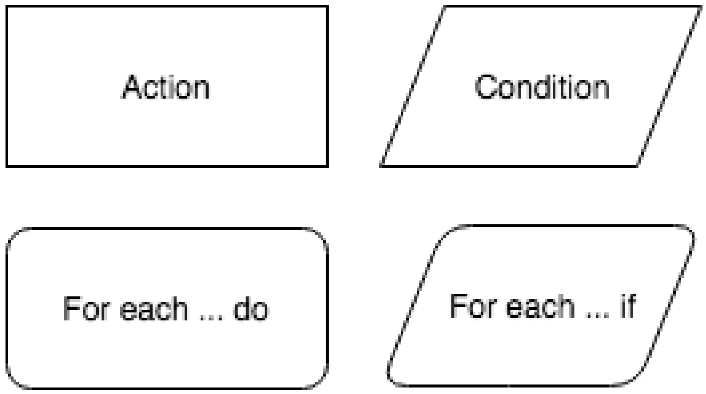

Figure 15: Legends for Controller Program Flowcharts

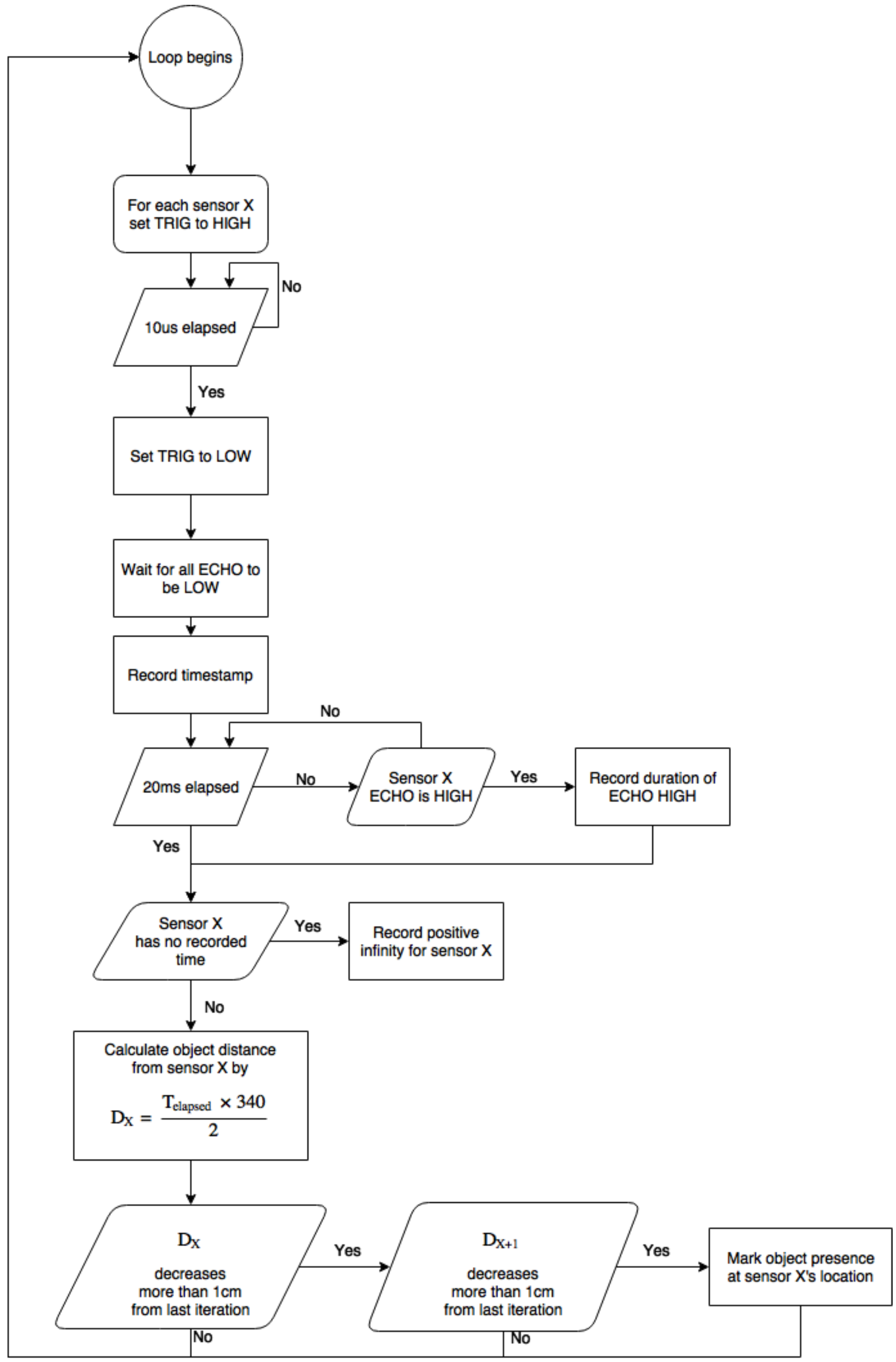

Figure 16: Flowchart for `SensorArray` Class

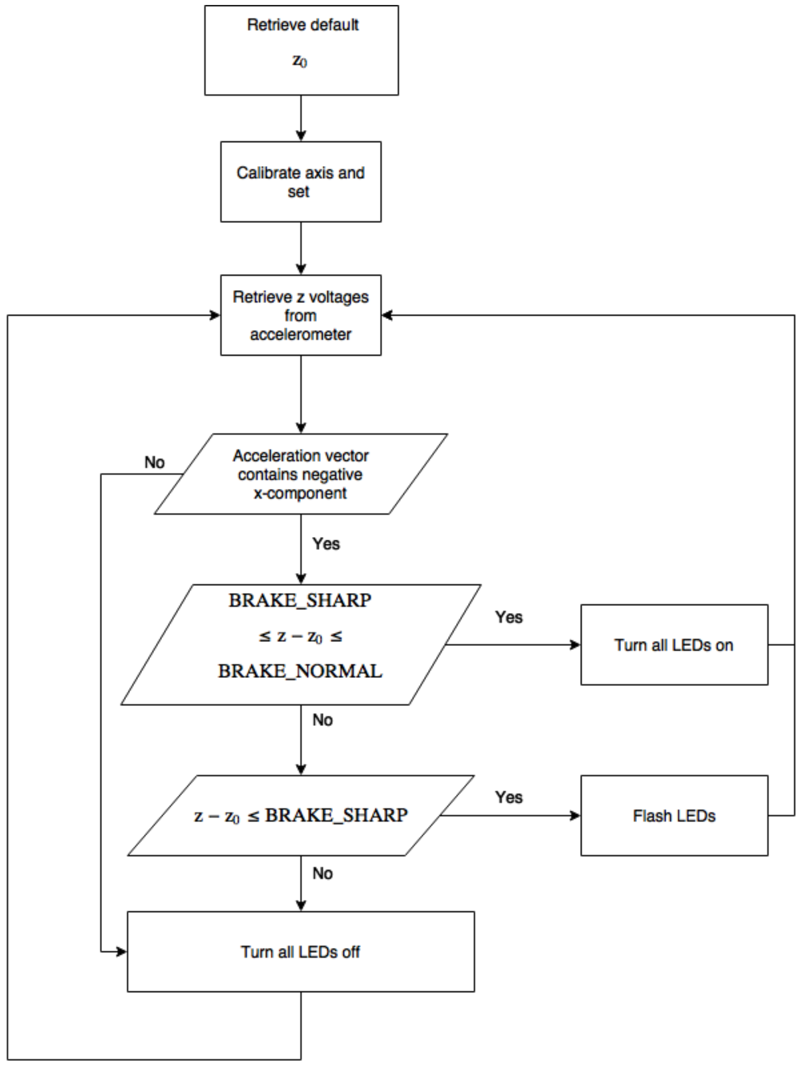

Figure 17: Flowchart for `BrakeLight` Class