

AUTOMATIC BIKE LIGHT

By

Pesandi Gunasekera

Magdalene Noftz

Nathanael Salazar

Final Report for ECE 445, Senior Design, Spring 2026

TA: Chihun Song

27 February 2026

Project No. 56

Abstract

Our senior design project will be an automatic bike light. A light sensor measures the amount of ambient light in the air to determine if it is day or night, and if it determines that it is day, the front light and rear light will be set to a strobing setting, while if it is night, it will be set to a constant light setting. A LIDAR sensor on the back of the bike will alert the user if a car is approaching within 40 meters by turning on an indicator light on the handlebars. Additionally, the biker will provide user input, using two pushbuttons, that controls a rear turning light indicator to let drivers behind them know they are turning. Finally, to conserve energy, there will be a button that turns the whole system on and off. If the bike is not moving for over a minute when the switch is on, determined by an accelerometer sensor, the lights will turn off.

Contents

- 1. Introduction 1
 - 1.1 Problem..... 1
 - 1.2 Solution 1
 - 1.3 Visual Aid..... 2
 - 1.4 High-Level Requirements List 2
- 2 Design..... 2
 - 2.1 Physical Design..... 2
 - 2.2 Block Diagram 6
 - 2.3 Functional Overview and Block Diagram Requirements 7
 - 2.3.1 Control Power Subsystem..... 7
 - 2.3.2 Sensing Subsystem..... 8
 - 2.3.3 Data Processing Subsystem 10
 - 2.3.4 Front Light Power Subsystem 12
 - 2.3.5 Rear Light Power Subsystem..... 13
 - 2.3.6 Front User-Interface Subsystem 13
 - 2.3.7 Rear User-Interface Subsystem..... 14
 - 2.4 Hardware Design..... 15
 - 2.4.1 Light Sensing 15
 - 2.4.2 Motion Sensing 15
 - 2.4.3 Proximity Sensing..... 16
 - 2.4.4 Voltage and Current Regulation..... 17
 - 2.5 Software Design 17
 - 2.6 Commercial Component Selection 18
 - 2.6.1 Microprocessor Selection 18
 - 2.6.2 Sensor Selection..... 18

2.6.2 Sensor Selection.....	19
2.6.3 LED Selection.....	19
2.6.4 Battery Selection.....	19
2.6.5 General Component Selection.....	19
2.7 Tolerance Analysis.....	19
2.8 Cost Analysis.....	21
2.9 Schedule.....	22
3 Engineering Standards, Ethics, and Safety.....	23
3.1 Societal Impact.....	23
3.2 Ethics.....	23
3.3 Engineering Standards.....	24
3.3.1 IEEE Standard 1789-2015.....	24
3.4 Safety.....	24
4 Conclusion.....	25
References.....	26

1. Introduction

1.1 Problem

Bicycles that drive on the road legally must have a light on the front allowing them to be visible for 500 feet and have a rear reflector or rear light in the state of Illinois. It is also recommended that a bike is visible for at least 100 feet for vehicles approaching from behind [1].

The majority of bicycle lights currently in the market do not have systems in place to automatically adjust their brightness, in the same way cars have automatically adjusting headlights. Additionally, they often lack a method for cyclists to indicate when they are turning and which direction they are turning, which can cause confusion if the cyclist doesn't use hand signals to indicate their turns.

Even if cyclists have lights on their bikes, they can forget to turn them on. Similarly, cyclists can forget to turn their lights off, thus draining the battery and making the lights useless. Furthermore, the luminosity of the bike light might not be appropriate for the light level of the environment. Having a light that is too dim for the environment reduces the bicyclist's visibility, while having a light that is too bright wastes the battery of the device.

While there are bicycle lights in the market that address some of these problems, they are often too expensive for the average consumer. The most similar product to our own is the Garmin Varia which is a bike light and radar system which retails for \$200. This price barrier can prevent most cyclists from accessing these features for their bicycles.

1.2 Solution

We propose to create an affordable set of bike lights that automatically adjust based on the amount of ambient light in the environment. This system would include a white light in the front that has two different output settings; a flashing mode for riding in the daylight, a bright constant beam for a nighttime environment. Additionally, a red light would be included on the back that is always on for constant visibility. The system would contain a light sensor that detects the light-level of the environment and each light in the system will switch to the appropriate setting based on the input from the sensor.

The system would also include an accelerometer that detects when the bike is moving. When the accelerometer is activated, it will enable the light sensor to detect the ambient light and set the bicycle lights to the appropriate light level. This system would operate on a timer, automatically shutting off when the sensor hasn't detected any activity for one minute. Additionally, a push button would be in place to manually turn the lights on and off.

In addition to the red LED, the rear light module would contain a LIDAR sensor that detects objects behind the bicycle. If a vehicle were to approach the bicycle from behind within 40 meters, the LIDAR would detect it and send a signal to an indicator light at the front of the bike.

There would also be two turn signals LEDs on the back of the bike that the user can use to indicate their turn. The front of the bike would include a left and right button that corresponds to the respective turn signal light.

1.3 Visual Aid

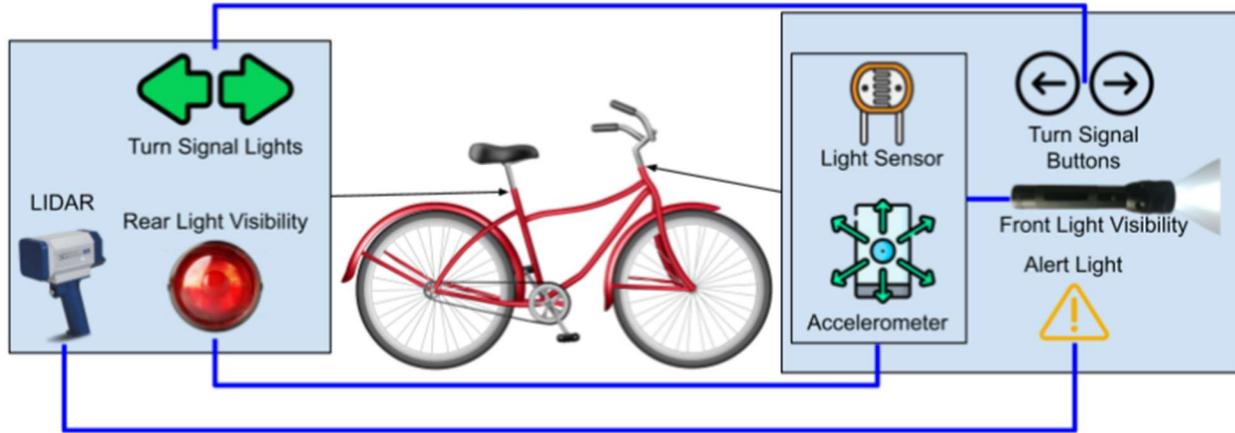


Figure 1: High-Level Diagram of Bike Light System

1.4 High-Level Requirements List

For the automatic bike light project to be considered a working solution to the problem statement described above, the following high-level requirements must be met:

- The light sensor must determine it is daytime when the Analog-to-Digital (ADC) count value is greater than or equal to the threshold value and must determine it is nighttime when the ADC count value is less than the threshold value. The threshold ADC count value will be determined through testing of the sensor's response to different amounts of ambient light.
- The radar sensor must detect an approaching object behind the bike within a range of 40 meters.
- When the bike is not in use, the lights must automatically shut off after a minute of inactivity.

2 Design

2.1 Physical Design

For the front bicycle lights, the LEDs will be placed on a separate PCB that is covered by a transparent plastic lens to allow for visibility of the lights. The rest of the power circuit, including the battery, will be placed in a solid plastic container. Isolating the LEDs from the rest of the power circuit focuses the beam from the headlights, allows the lights to sit at a different angle than the rest of the PCB, and prevents the need for encasing the entire light in a clear plastic container. Both the plastic lenses for the LEDs and the plastic case for the power circuit will be waterproof.

The rear bicycle lights will have a similar structure to the front lights, with the LEDs sitting on a separate board. However, the rear light module will also contain two modules for the amber LEDs which will be used to indicate the turn of the biker. The turn lights will be connected to the same circuit as the rear lights, but sit far enough away from each other that the direction of the turn can be clearly seen, even from a distance.

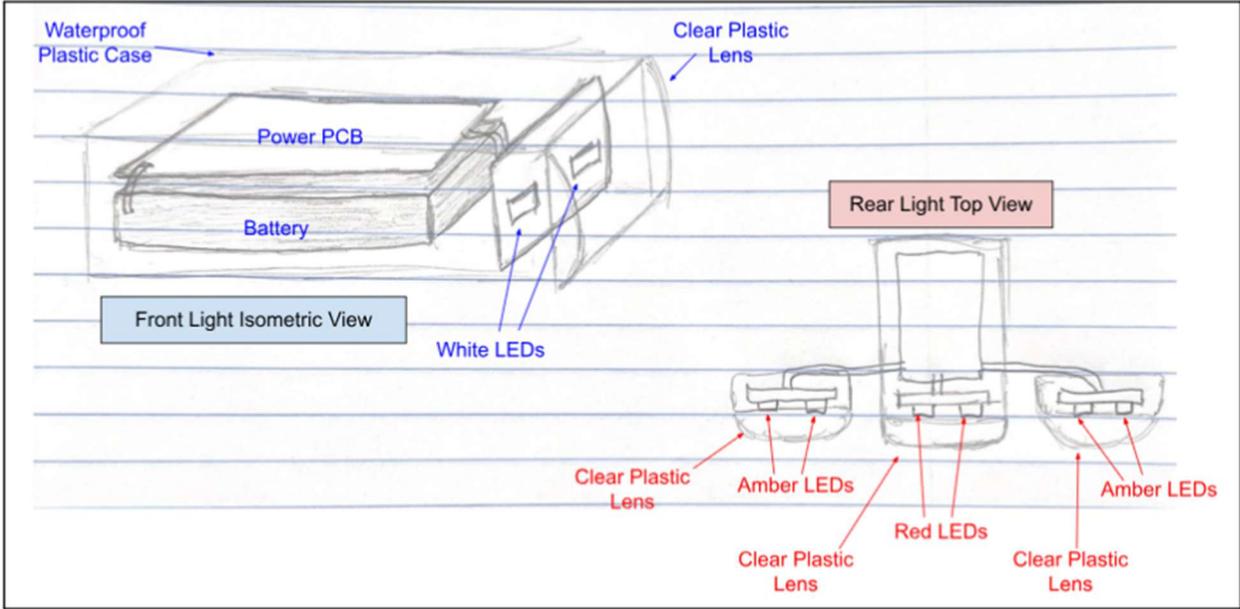


Figure 2: Headlight Housing and PCB Layout

The control PCB and accelerometer are housed in the control box at the center of the bike frame. Cables will run between the control box and the front and rear light modules to transmit information between the sensors, lights, and microcontroller. The control box will be secured to the frame of the bike with screwed connector pieces.

The light sensor will be placed on top of the front light module, with a clear, waterproof cover to protect it from the elements. This sensor was placed on the component with the most exposure to the environmental lighting. The front light will be secured to the bike with a plastic connector.

The LiDAR sensor is located on the rear light module of the bike, along with the turn signal lights. The module will sit high enough to clear the rear wheel, ensuring visibility of the lights and for the LiDAR sensor.

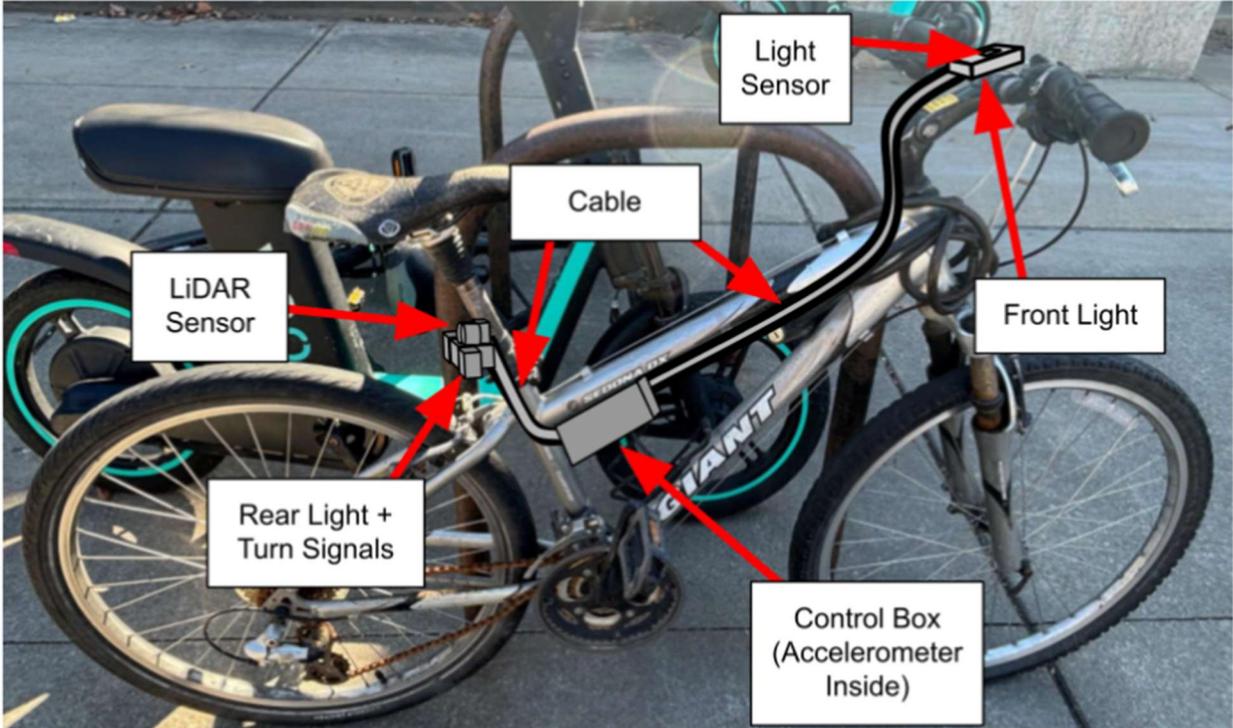


Figure 3: Component and Sensor Placement

In addition to the white LEDs, the front light module will contain the turn signal buttons, the on and off button, the indicator light, and the light sensor. The light sensor will be embedded on the top of the module with a clear waterproof cover protecting it.

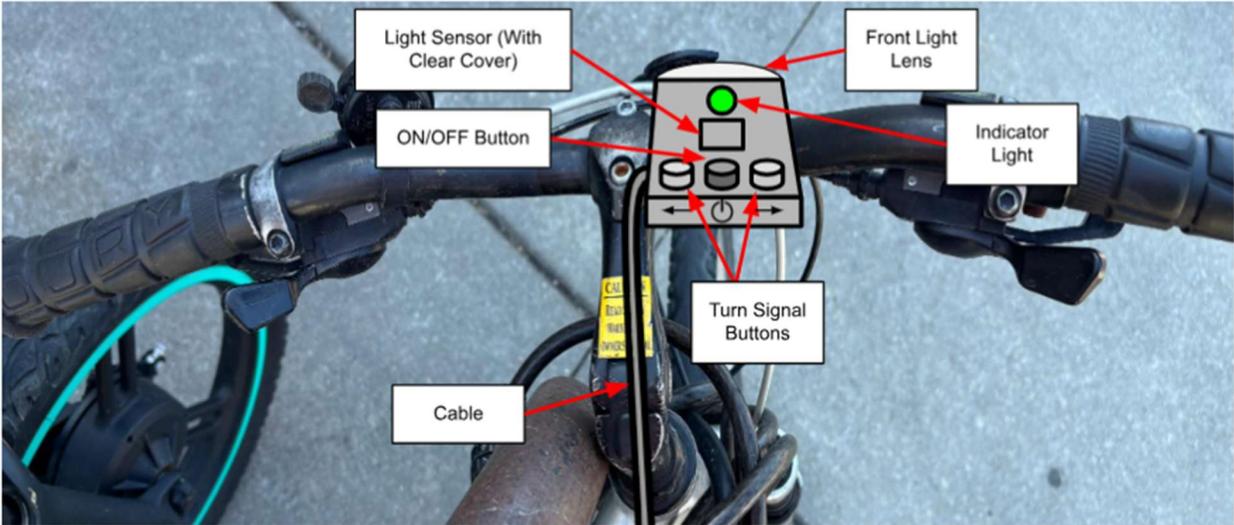


Figure 4: Front Light Module Physical Design

The rear light module will contain the red LEDs, the orange LEDs for the turn signal lights, and the LiDAR sensor. The entire module will fasten to the bike with a standard plastic clip.



Figure 5: Rear Light Module Physical Design

2.2 Block Diagram

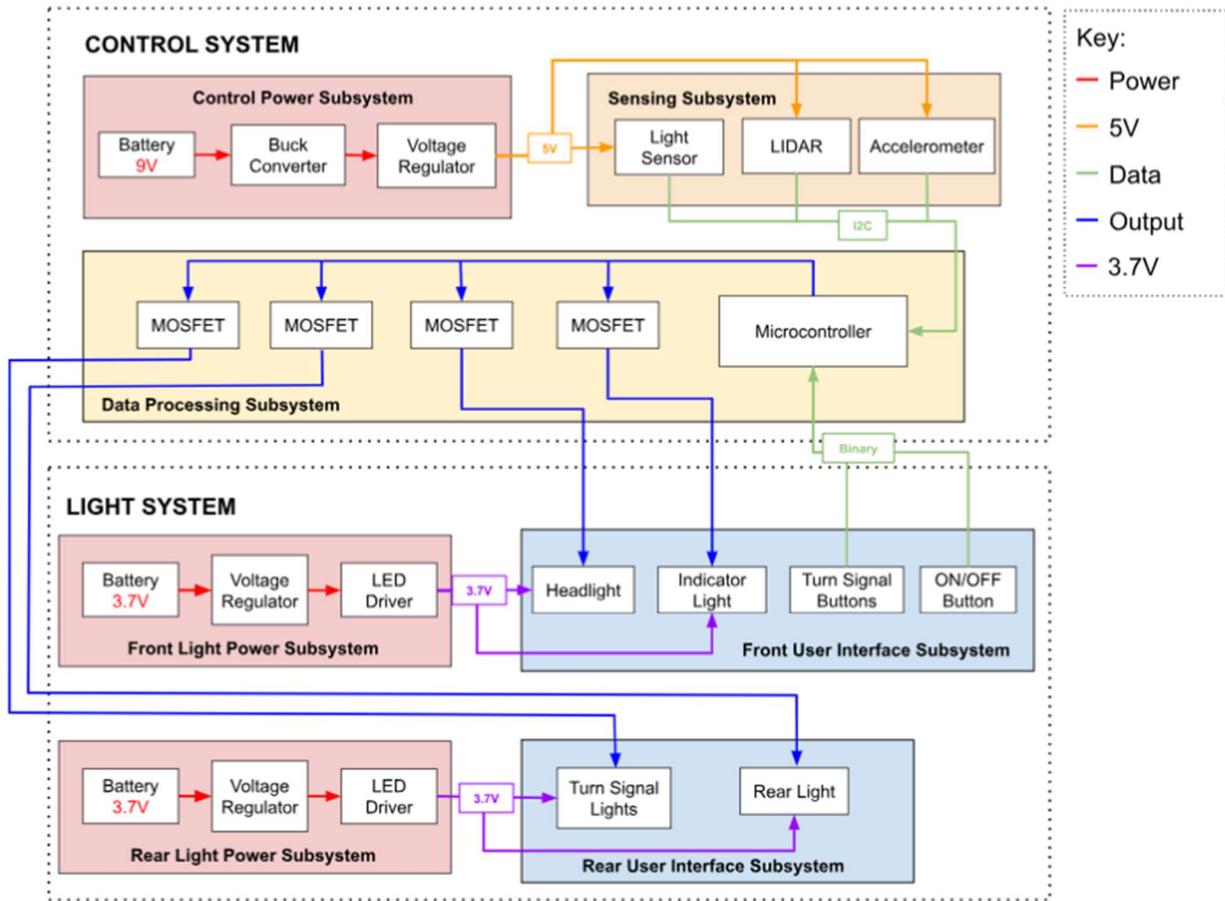


Figure 6: Block Diagram Design

2.3 Functional Overview and Block Diagram Requirements

2.3.1 Control Power Subsystem

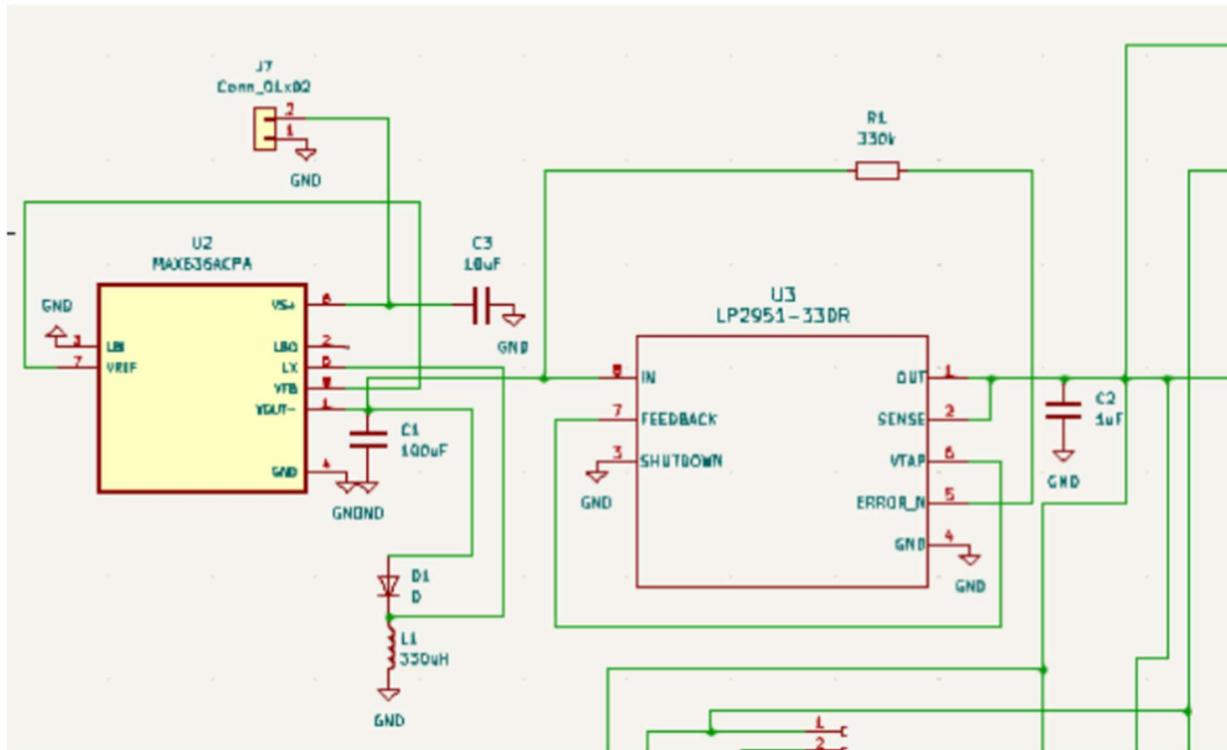


Figure 7: KiCad Schematic of Voltage Regulator and Buck Converter

The control power subsystem is responsible for supplying power to all electronic components on the control system PCB. This subsystem contains a 9V battery which feeds into the buck converter. The buck converter steps down the voltage from 9V to 5V, thus satisfying the voltage requirements for the components in the control subsystem. The output of the buck converter is fed into a voltage regulator which ensures that the output voltage remains within +/- 3% of tolerance and protects the circuit from overvoltage and reverse voltage.

Table 1: Control System Requirements and Verification

Requirements	Verification
The buck converter steps down the 9V battery to 5V	<ul style="list-style-type: none"> First connect a DC power supply to the input of the buck converter, with an output set to 9V. Use a multimeter to measure the output of the buck converter, ensuring it is 5V Replace the DC power supply with a 9V battery. Once again use the multimeter to ensure that the buck converter is still outputting 5V

<p>The voltage regulator maintains a stable output voltage of 5V +/- 3%</p>	<ul style="list-style-type: none"> • Use a multimeter to ensure that the output from the voltage regulator is 5V • Use an oscilloscope to check the voltage ripple of the regulator output. Ensure that the ripple remains within +/- 3% of the intended 5V
---	---

2.3.2 Sensing Subsystem

The front sensing subsystem detects the ambient light using a light sensor and determines whether or not the bike is moving using an accelerometer.

Based on the amount of light detected, the microcontroller communicates what mode the front headlight should be in. If the light sensor detects an ambient light level equivalent to day, the sensor will communicate to the microcontroller that it is daytime outside. We will have to conduct testing to see how sensitive the sensor is and where the Analog-to-Digital (ADC) count level between night and day falls. If the ADC readings are under this value then the sensor will determine it is night. This information will be used in the front and rear interface system.

The subsystem will also contain an accelerometer that communicates with the microcontroller and turns off the front and rear lights if no movement is detected (determined by if the change in acceleration is greater than a threshold value) in the bike when the system is turned on. We will have to conduct testing to see where the change in acceleration threshold between movement and stationary. The sensor is also connected to a timer. If the accelerometer hasn't detected motion in one minute, the sensor will communicate this to the microcontroller causing the lights to turn off.

The LiDAR sensor will be connected to the back of the bike. It will communicate with the microcontroller if a vehicle is approaching the bike within a 40 meter distance. If it senses a vehicle then the control system will turn on an indicator light to the rider.

<p style="text-align: center;">Table 2: Sensing System Requirements and Verification</p>	
<p style="text-align: center;">Requirements</p>	<p style="text-align: center;">Verification</p>
<p>Light sensor correctly determines it is day when: ADC ≥ Threshold</p> <p>And determines it is night when: ADC count < Threshold</p>	<ul style="list-style-type: none"> • On a breadboard, connect the light sensor to the microcontroller. Take the sensor readings outside at different times of the day and at night in areas with differing street lighting. Use these readings to determine the ADC count threshold for day and night. • On the breadboard, connect the light sensor to the rest of the circuit. Observe the sensor ADC readings and the output light setting during the day and night to confirm the light is strobing when ADC ≥ Threshold and the light is set to a constant light setting when ADC count < Threshold.

	<ul style="list-style-type: none"> Repeat testing with the components soldered to the PCB.
<p>Accelerometer determines if bike is moving if: $\Delta \geq \text{Threshold}$</p> <p>And determines the bike is off if: $\Delta < \text{Threshold}$</p>	<ul style="list-style-type: none"> On a breadboard, connect the accelerometer to the microcontroller. Take the sensor readings on the bike while it is in motion (moving slowly and moving quickly) and when the bike is stationary. Calculate for every case to determine the threshold for if the bike is stationary or moving. On the breadboard, connect the accelerometer to the rest of the circuit and mount on the bike. Observe the lights when the bike is moving quickly, moving slowly, and stationary and confirm that the lights are on when $\Delta \geq \text{Threshold}$ and off when $\Delta < \text{Threshold}$ Repeat testing with the components soldered to the PCB.
<p>LiDAR sends a signal to the microcontroller if an object is approaching behind the bike within 40m.</p>	<ul style="list-style-type: none"> Start with breadboarding the LIDAR to the microcontroller and place a wide object one meter behind the LIDAR. Confirm that the LIDAR is reading the correct distance of the object and that the indicator light is on. Move the object incrementally further away up until 40 meters and confirm the correct LIDAR reading and that the indicator light is on. Repeat this process by moving the object further away than 40 meters and confirm that the indicator light is off. Repeat testing with the components soldered to the PCB.

2.3.3 Data Processing Subsystem

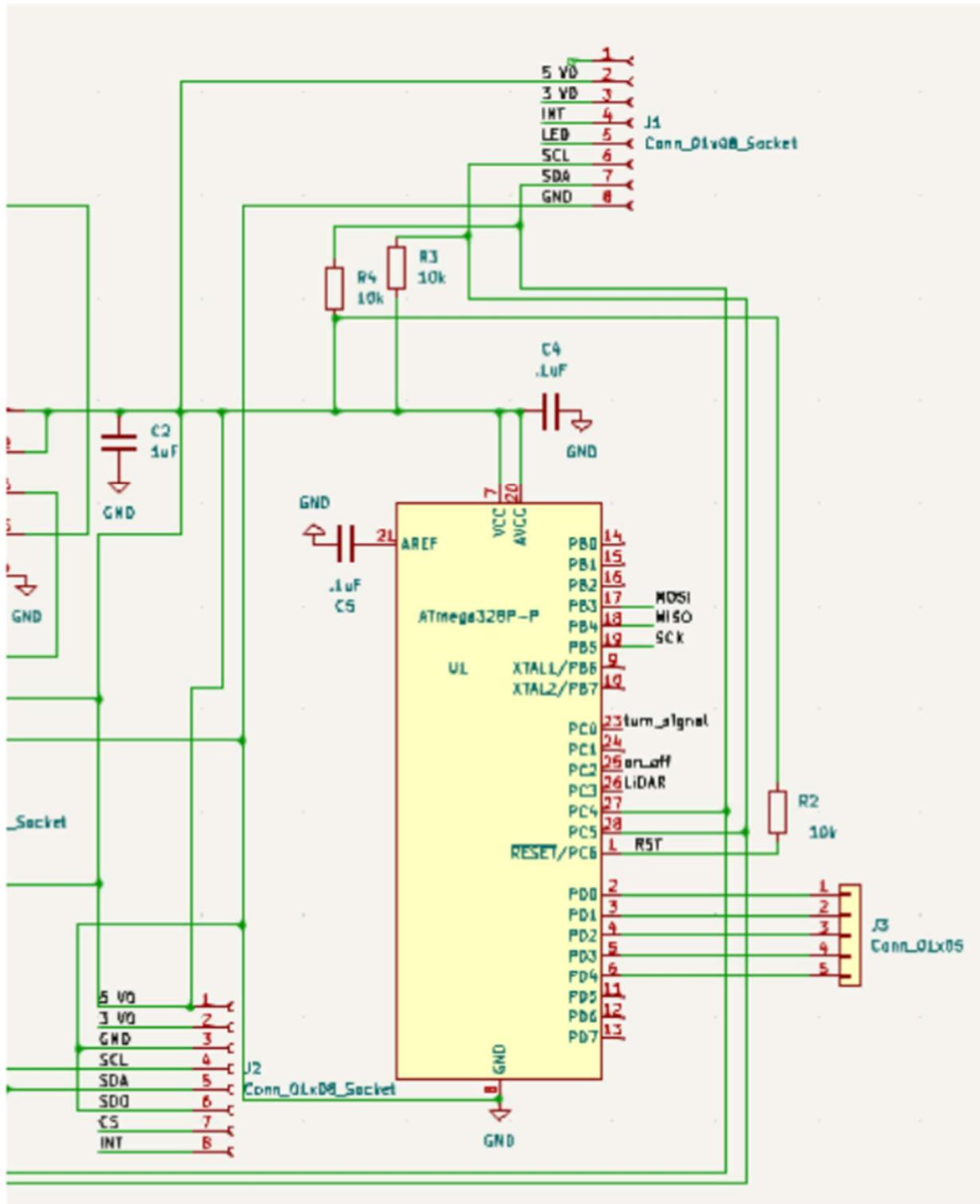


Figure 8: KiCad Schematic of Microcontroller

The control subsystem interfaces with all of the other subsystems, converting the inputs from the sensors and push buttons to outputs for the bike.

The system only turns on if the ON/OFF button is pressed. Once the system is on then the sensor inputs determine the outputs of the bike. If the system is off, then nothing will occur regardless of the ambient light to bike movement.

For the front light system, the control subsystem takes the light sensor and accelerometer inputs and sets the bike lights to the appropriate light setting. The microprocessor determines if it's night or day depending on the light sensor input. If it is day then the front and back light will strobe. If it is night then the front and back light will provide constant illumination.

If the accelerometer is not detecting movement, the microcontroller will set a timer to turn both the front and the rear lights off after one minute, regardless of the light sensor input. For the rear light system, the control subsystem is used to control the rear lights to indicate the direction the cyclist is turning based on the user input buttons on the front of the bike.

The LIDAR sensor on the back of the bike is used to send information to the front or on whether or not an approaching object is within 40 meters behind the bike. If an object is detected the microcontroller will turn on an indicator light on the front of the bike.

Table 3: Data Processing System Requirements and Verification

Requirements	Verification
Turn signal buttons send correct signal to microcontroller	<ul style="list-style-type: none"> Press the turn signal buttons and test the signals sent to the microcontroller with an oscilloscope. There should be current flowing from the button to the microcontroller the first time the button is pressed. The second time the button is pressed, the current should be zero.
On/Off Button sends correct signal to the microcontroller	<ul style="list-style-type: none"> Press the on/off button and test the signal sent to the microcontroller with an oscilloscope. There should be current flowing from the button to the microcontroller the first time the button is pressed. The second time the button is pressed, the current should be zero.

2.3.4 Front Light Power Subsystem

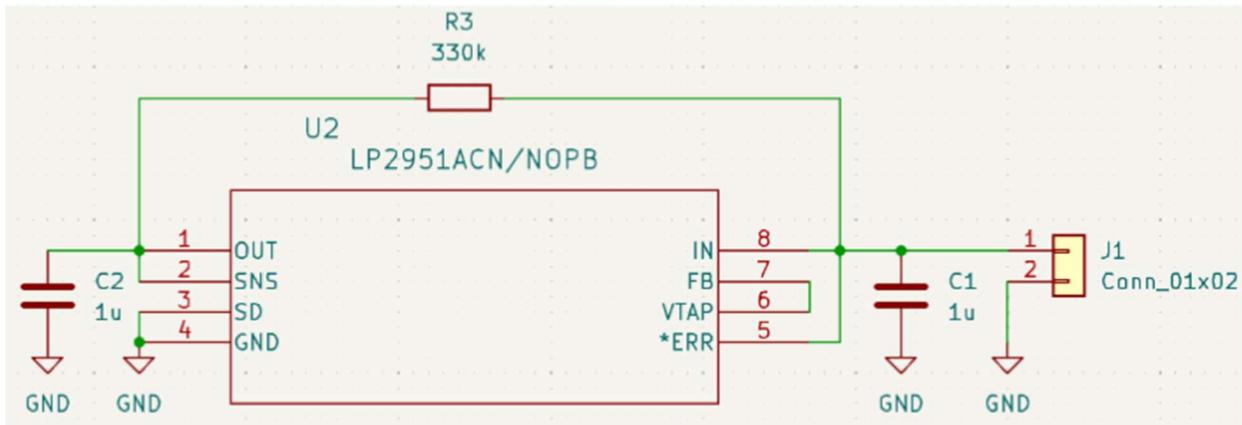


Figure 9: KiCad Schematic for Front Lights

The front light power subsystem provides power to the headlight and indicator light in the front user-interface subsystem. It consists of a 3.7V/3000mAh battery, a voltage regulator, and an LED driver.

The 3.7V from the battery is first passed through the voltage regulator to ensure a stable output voltage of 3.7V +/- 3%. This protects the LEDs from over-voltage. The output signal from the voltage regulator is then passed through an LED driver, which ensures a stable output current. This protects the LEDs from over-current. The resulting output from the subsystem is a stable 3.7V.

The output distributes the power requirements to the loads as follows: 0.15W to each of the two LEDs in the headlight and 0.075W to the indicator light.

Table 4: Front Light Power Requirements and Verification	
Requirements	Verification
Output of subsystem is 3.7V +/- 3%	<ul style="list-style-type: none"> Start with breadboarding components and use a multimeter to ensure 3.7V output Measure 3.7V output signal with the oscilloscope and measure the ripple voltage to ensure it is $0.06(3.7) = 0.222V$ Repeat testing with components soldered onto PCB

2.3.5 Rear Light Power Subsystem

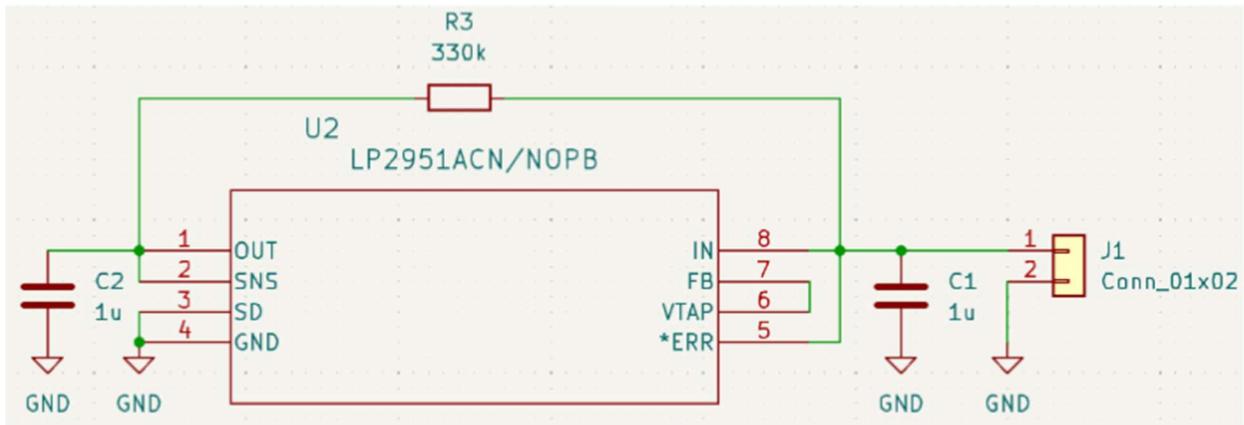


Figure 10: KiCad Schematic for Rear Light Power

The rear light power subsystem provides power to the turn signal lights and rear light in the rear user-interface subsystem. It consists of a 3.7V/3000mAh battery, a voltage regulator, and an LED driver.

In the same way as in the front light power subsystem, the 3.7V from the battery is first passed through the voltage regulator to ensure a stable output voltage of 3.7V +/- 3% to protect the LEDs from over-voltage. The output signal from the voltage regulator is then passed through an LED driver to protect the LEDs from over-current with a stable output current. The resulting output from the subsystem is a stable 3.7V, which is the same as the output of the front light power subsystem.

The output distributes the power requirements to the loads as follows: 0.41W to each of the four LEDs in the turn signal lights and 0.31W to each of the two LEDs in the rear light.

Table 5: Rear Light Power Requirements and Verification	
Requirements	Verification
Output of subsystem is 3.7V +/- 3%	<ul style="list-style-type: none"> Start with breadboarding components and use a multimeter to ensure 3.7V output Measure 3.7V output signal with the oscilloscope and measure the ripple voltage to ensure it is $0.06(3.7) = 0.222V$ Repeat testing with components soldered onto PCB

2.3.6 Front User-Interface Subsystem

In the front user interface subsystem, the front headlight of the bike is set to a strobing setting during the day and a constant light setting at night. A small rear-alert light on the right handlebar is connected to the rear control subsystem and turns on when the rear control subsystem detects an approaching object within 40 meters behind the bike. Turn signal buttons (one button for right and one for left) on the left handlebar are used to indicate the direction the cyclist intends to turn and sends this information to the rear control subsystem.

The front user interface subsystem consists of the front headlight, the rear-alert light, the turn-signal button, and the on/off button. The front headlight is powered by the front power subsystem and is controlled by a MOSFET in the control subsystem, which receives its setting instructions from the microcontroller. To be considered functional, the headlight must be set to a strobing setting during the day and one of two constant light brightness settings at night. The rear-alert light is a small indicator light on the right handlebar that is powered by the front power subsystem and is controlled by a MOSFET in the control subsystem, which receives instructions from the microcontroller on whether the light should be on or off. To be considered functional, the rear-alert light must be on when an approaching object behind the bike is within 40 meters and must be off otherwise. The turn-signal buttons, one for left-turn and one for right-turn, are located on the left handlebar and send information on the direction the cyclist is turning to the microcontroller. To be considered functional, when the left-turn button is pushed, the button must send the direction to the front control subsystem that the cyclist is turning left. When the right-turn button is pushed, it must send the direction that the cyclist is turning left. If the button is pushed a second time, it must send the direction to the front control subsystem that the turn signal should be reset.

Table 6: Front User Interface Requirements and Verification	
Requirements	Verification
Rear approaching alert light lights up when a vehicle is within 40 m of bicycle	<ul style="list-style-type: none"> Put the bike 40 meters in front of a building and confirm that the indicator light turns on. Move the bike to 30 meters in front of the building and confirm that the indicator light is still on. Repeat this process until the bike is right up against the building and confirm that the indicator light stays on. Put the bike more than 40 meters away from the building and confirm that the indicator light turns off.
Front head light correctly chooses between strobing/constant light depending on day/night	<ul style="list-style-type: none"> Based on the signal sent from the microcontroller (measured through oscilloscope/microcontroller) the front head light strobos or has constant light.
Front light turns off after a minute of no acceleration	<ul style="list-style-type: none"> When the system is on, wait for a minute with no motion and confirm that the lights turn off.

2.3.7 Rear User-Interface Subsystem

The rear user interface subsystem includes the turn lights on the back of the bike that indicate if the user is turning as well as a rear light that is always on for visibility purposes.

The rear user interface subsystem is composed of turn signals that operate based on user input transferred from the control subsystem, which indicate if the rider is turning right or left, as well as a

rear visibility light. All lights are powered by the rear power subsystem. To be considered functional, the rear visibility light must be continuously on and the turn signal lights must correctly indicate whether the cyclist is turning left or right, based on the user input.

Table 7: Rear User Interface Requirements and Verification	
Requirements	Verification
Turn-Signal buttons turn the turn signal lights on and off	<ul style="list-style-type: none"> • Press the left indicator button and confirm that the left turn-signal light turns on. Press the button again and determine that the light turns off. • Press the right indicator button and confirm that the right turn-signal button turns on. Press the button again and determine that the light turns off.

2.4 Hardware Design

2.4.1 Light Sensing

The light sensor we are using to detect the amount of ambient light is the Adafruit SI1145 UV index/IR/Visible Sensor. The sensor is compatible with 5V microcontrollers, transmits data using I2C protocol (two-wire, short-distance, and low speed communication between a controller and its peripherals), and can measure both artificial and natural light. Our microcontroller (ATMEGA328P-PU) can process I2C signals and is powered using 5V, so this sensor can be used in our circuit.

The SI1145 sensor does not directly output the amount of ambient light (in units of luxes), but rather operates primarily by calculating the UV index. However, according to the SI1145 [6] datasheet, the sensor does output 16-bit Analog-to-Digital conversion (ADC) counts. The photodiodes on the sensor generate current when exposed to light, and the Analog-to-Digital converter converts it into a number based on the amount of light hitting the photodiodes. So, the higher the ADC count, the more ambient light.

The ADC count from the SI1145 light sensor can be used by the microcontroller to determine whether it is day or night, and what light level the bike headlight should be at. Testing will need to be done on the sensor to determine the threshold values for day and night and what ADC count ranges should result in each bike light setting (strobing, brighter constant light, dimmer constant light).

2.4.2 Motion Sensing

The accelerometer we will be using to detect whether the bike is in use or not is the Adafruit LIS3DH Triple-Axis Accelerometer. The sensor is compatible with 5V microcontrollers and has the option for transmitting data using either I2C or SPI protocol (four-wire, short distance, and high speed communication between a controller and its peripherals). Our microcontroller operates at 5V and can process both I2C and SPI signals, so this sensor can be used in our circuit. We have chosen to use the I2C

protocol due to the lower wire number requirements and for simplicity, so that all the sensors can connect to the same input on the microcontroller.

The LIS3DH accelerometer operates by measuring the acceleration of the object it is mounted on. According to the LIS3DH datasheet [7], The sensor outputs the acceleration in the x, y, and z direction either as raw data or normalized data. The normalized data is only important if calculations must be done with the results, and since for our project we only care about whether there is a change in acceleration and not the actual values, we will use the raw data. To determine whether the bike is stationary or in motion, the data will be sent to the microcontroller, which will calculate the change in acceleration between the current value and the previous value using the following equation:

$$\Delta = |x - x_{prev}| + |y - y_{prev}| + |z - z_{prev}|$$

This value will then be compared to a threshold value, determined through testing of the accelerometer, that represents the lowest change in acceleration possible while in motion. For the testing of the accelerometer, we will mount the LIS3DH on the bike and observe the readings from the sensor when the bike is stationary, slowly moving, and moving quickly. This will help us to determine the minimum change in acceleration when the bike is moving (the threshold value). If the delta value is less than this threshold, then the bike will be determined to be stationary. If the bike is determined to be stationary for at least one minute, then the microcontroller will turn off the system.

2.4.3 Proximity Sensing

The proximity sensor we will be using to detect whether a car is approaching from behind the bike is the Garmin LIDAR Light V3. According to the description of the sensor on the Adafruit website [10], the LIDAR can detect an object within 40 meters with an accuracy of +/- 2.5cm, requires 5V for operation, and transmits data using I2C protocol. Our control circuit, including our microcontroller, operates at 5V and our microcontroller can process I2C signals, so this sensor can be used in our circuit.

According to the datasheet [8], the LIDAR sensor works by transmitting a laser beam, at a wavelength of 905nm so it is not visible to the human eye, and the beam reflects off of objects within range and returns to the sensor. The sensor continuously sends the laser beam and receives return signals, and stores the strength of the return signals in memory. If there is a peak in the return signal strength that emerges from above a threshold (determined by typical noise), then the sensor will determine that there is an object within range and use the information and the speed of light to calculate the distance of the object.

The spread of the transmitted laser beam can be calculated by dividing the distance of the object by 100. So, for an object 10 meters away, the spread of the beam is 10cm, and for an object at the maximum distance of 40 meters away, the spread of the beam is 40cm. This will not be a problem for our project because we only want to detect objects directly behind the bike, so the focus of the laser beam will work for these purposes.

We will mount the LIDAR sensor on the back of the bike facing directly backwards. It will only output a non-zero distance value if it determines there is an object within 40 meters behind the bike. The

microcontroller will receive this information, so if it receives a non-zero distance value, it will turn on the indicator light on the front of the bike.

2.4.4 Voltage and Current Regulation

One important aspect of our circuit design is voltage and current protection. Many of the components we are using, such as the sensors, microcontroller, and LEDs, could be damaged during conditions of over-voltage or over-current. To protect the components from these conditions, we will use voltage regulators and LED drivers.

The voltage regulator we will use is the LP2951 adjustable regulator. Since we have three batteries in the overall system, one regulator will be connected to each battery. In the control power subsystem in the control system, the voltage regulator will take the output from the buck converter and output a steady 5V. This will protect the microcontroller and sensors from sudden voltage spikes. In the front light power subsystem and rear light power subsystem, the voltage regulator will take the 3.7V from the battery and output a steady 3.3V, which will protect the LEDs from sudden voltage spikes. The slight drop in voltage from the battery is not a problem for the overall operation of our circuit since the necessary forward voltage of the LEDs is around 3V.

The LED driver we will use for the LEDs will be resistors. Since the LEDs are all low power (less than one watt), a simple resistor is enough to protect each LED from sudden current spikes. The value of the resistor will depend on each LED, since we are using several different kinds, and we will calculate each of the necessary resistor values using the voltage and current specifications of each LED.

2.5 Software Design

For our software design our key decision making component was our microcontroller. The microcontroller takes information from our three sensors (light, acceleration, and LiDAR), as well as user input to produce a series of outputs. The flow chart of our code can be seen below in figure 11. We have an on/off button to turn the system on and off. When the system is off nothing will occur. When the system turns on the accelerometer will be used to determine if the bike is moving. If the bike is not moving then the front and rear visibility lights will turn off. If the bike is moving these lights will turn on. Based on the ambient light detected by the light sensor the microcontroller will determine if it's night or day. If it is day then the visibility lights will strobe. If it is night the visibility lights will provide constant illumination.

If a car approaches the bike within a 40 m range when the system is on an indicator light will turn on to alert the rider of a car approaching. If there is no car approaching then this light will be off. Lastly the rider can click a turn signal button to activate a turn signal. This button is a latching button, so once pressed the rider must press the button again to release the button from the pressed position to turn the indicator lights off.

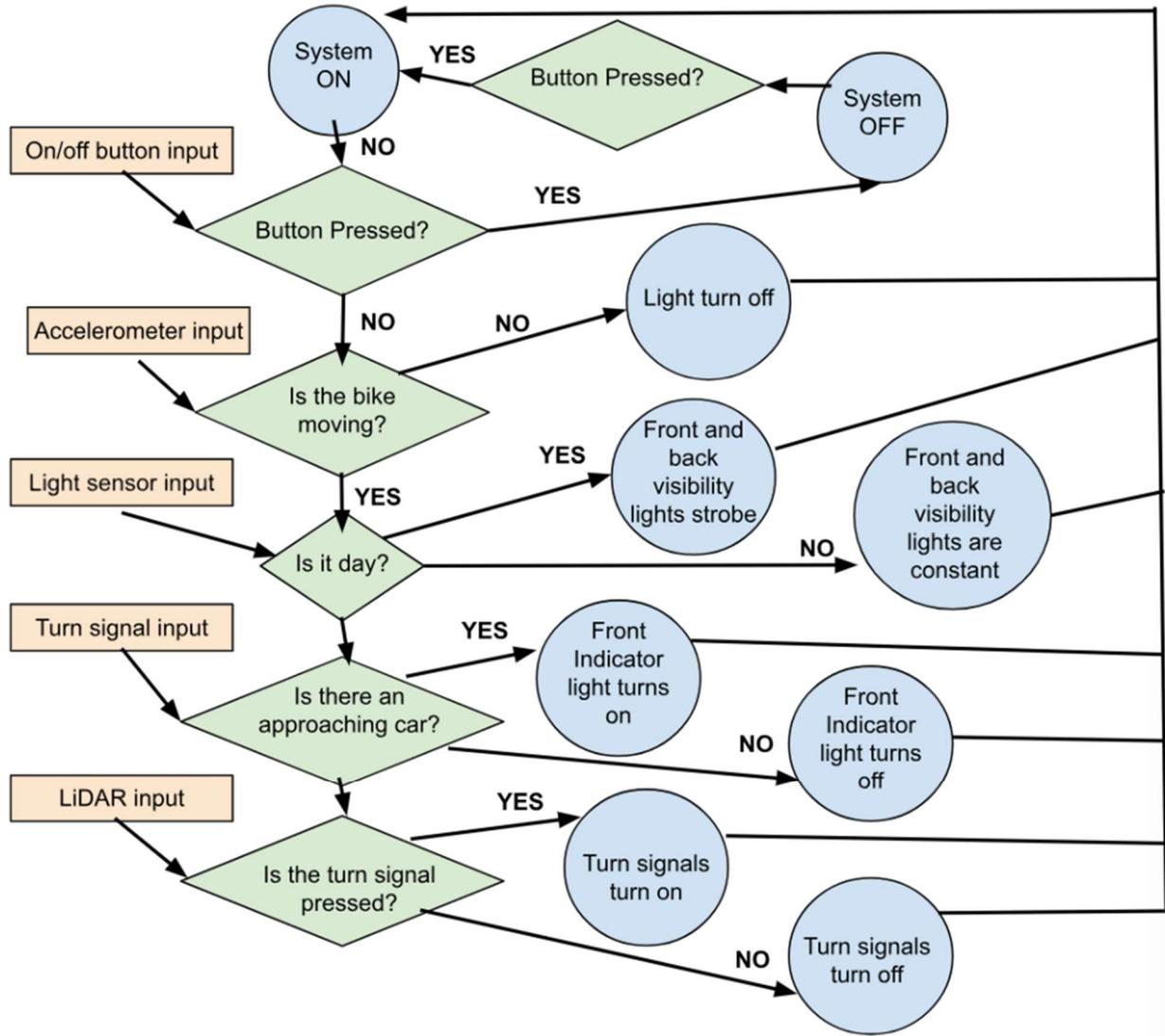


Figure 11: Headlight, Turn Signal, and Indicator Light Flowchart

2.6 Commercial Component Selection

2.6.1 Microprocessor Selection

We chose to use ATMEGA328P-PU because they were free from the ECE self service center and provided the necessary functionality of managing multiple lights and switches with its 2KB of program storage and 2KB of SRAM. Because we do not use RF communication we chose a cheaper chip without RF capabilities. Additionally, because the chip is commonly used in some Arduino models there was a lot of technical documentation online about it.

2.6.2 Sensor Selection

For our light sensor we looked at parts available to us in the ECE 445 closet and found a light sensor that fit our requirements for light sensitivity and functionality. For our LiDAR sensor we initially wanted to use radar but realized that at the distance we were looking into (20-50m) the sensors on the market

were out of our budget. We then found a LiDAR sensor that fit all our requirements in the ECE 445 cabinet from a project from a previous year. We would not have been able to complete our project without this because the sensor was out of our budget. For our acceleration sensor we initially were going to use a vibration sensor, but found difficulties in sensors that fit the specific functionality of our intended use. We then found an accelerometer on DigiKey that was relatively inexpensive and would be able to determine if the bike was moving or not.

2.6.2 Sensor Selection

For our light sensor we looked at parts available to us in the ECE 445 closet and found a light sensor that fit our requirements for light sensitivity and functionality. For our LiDAR sensor we initially wanted to use radar but realized that at the distance we were looking into (20-50m) the sensors on the market were out of our budget. We then found a LiDAR sensor that fit all our requirements in the ECE 445 cabinet from a project from a previous year. We would not have been able to complete our project without this because the sensor was out of our budget. For our acceleration sensor we initially were going to use a vibration sensor, but found difficulties in sensors that fit the specific functionality of our intended use. We then found an accelerometer on DigiKey that was relatively inexpensive and would be able to determine if the bike was moving or not.

2.6.3 LED Selection

We chose our LEDs based on three factors: color, power usage, and brightness. We wanted to follow transportation standards and use white, amber, and red lights for our bicycle. We were also concerned about the brightness of the lights because we wanted the rider to be properly illuminated for safety reasons. We looked at the brightness of popular bike lights to ensure that ours would have a similar level of brightness. Our last major concern was power usage. We wanted to balance having bright enough LEDs, but also batteries that would last for multiple hours. Taking these considerations into our decision we found white, red, and amber LEDs from Cree LED that fit all our requirements.

2.6.4 Battery Selection

We also chose a 3.7 V battery with 3000 mAh of current which allows us 4-5 hours of minimum run time based on the power consumption of our chosen LEDs. This calculation is further discussed in the tolerances section (2.7) below. We also chose to use rechargeable batteries so the rider does not need to buy batteries repeatedly making this choice more cost effective long term and eco-friendly.

2.6.5 General Component Selection

For our other components we chose components (resistors, capacitors, voltage/current regulators, buck converter, MOSFETs, etc.) we chose components that operate at our desired voltage and current usages. We also chose components with ample technical documentation that would be easy for us to solder and work with.

2.7 Tolerance Analysis

An aspect of the design of this project that poses a risk to successful completion is the lifespan of the battery. The battery must be able to power all of the required sensors and circuit components and last for a reasonable amount of time.

Battery 1: 9V battery power rating = 5 Wh

Battery 2 and 3: 9V battery power rating = (3.7V) (3000 mAh) = 11.1 Wh

Battery 1 (9V):

Light sensor power: $(100\text{mA}_{\text{max}})(5\text{V}) = 0.5 \text{ W}$

Accelerometer: $(100\text{mA}_{\text{max}})(5\text{V}) = 0.5 \text{ W}$

Microcontroller: $(11\text{mA})(5\text{V}) = 0.055 \text{ W}$

Battery 1 total power: 1.055 W

Using a 5 Wh battery → minimum of 4.739 hours of power

Battery 2 (3.7V):

Headlight: $(55\text{mA}) (2.67\text{V}) (2 \text{ lights}) = 0.2937 \text{ W}$

Rear-alert light: $(15\text{mA})(5\text{V}) = 0.075 \text{ W}$

Battery 2 total power: 0.3687 W

Using a 11.1 Wh battery → minimum of 30.11 hours of power

Battery 3 (3.7V):

LiDAR sensor power: $(100\text{mA})(5\text{V}) = 0.5 \text{ W}$

Red rear light: $(140\text{mA}) (2.2\text{V}) (4 \text{ lights}) = 0.616 \text{ W}$

Turn signals: $(140\text{mA})(2.96\text{V}) (2 \text{ lights}) = 0.8288$

Battery 3 total power = 2.56 W

Using a 11.1 Wh battery → minimum of 4.33 hours of power

We chose to use a 9 V battery because most of our sensors and our microcontroller needed 5 V for operation. We then chose to use a rechargeable 3.7V battery because it provided more mAh allowing for a longer battery life and because the batteries were rechargeable for cost efficiency.

2.8 Cost Analysis

Table 8: Itemized list of Components and Costs						
Part Name	Part Number	Manufacturer	Unit Cost (\$)	Bulk Purchase Cost (\$)	Quantity	Actual Cost (\$)
LiDAR Connector Housing	GHR-06V-S	JST	\$0.13	N/A	2	\$0.26
LiDAR Connector Terminal	SSHL-002T-P0.2	JST	\$0.07	N/A	12	\$0.86
Accelerometer Sensor	1528-1516-ND	Adafruit	\$4.95	N/A	1	\$4.95
Buck Regulator	MAX636ACPA	Maxim	\$0.01	\$9.24 / 124 chips	3	\$0
Voltage Regulator	LP2951	Texas Instruments	\$0.60	N/A	2	\$0
Microcontroller	ATMEGA328P-PU	Microchip Technology	\$2.89	N/A	2	\$0
ISP Header (Microcontroller)	AVR-ISP-6	Adafruit	\$0.85	N/A	3	\$2.55
USB ISP Programmer	N/A	Unknown	\$0 (borrowing)	N/A	1	\$0
MOSFET	2SK3703-1E	Onsemi	\$0.34	N/A	4	\$0
9V Battery - Alkaline (Control Circuit)	EN22	Energizer	\$3.04	N/A	2	\$0
DC Female Barrel Jack	N/A	Unknown	\$0	N/A	1	\$3.29
9V Battery Clip to Male DC Barrel Jack	233	Keystone Electronics	\$0.84	N/A	2	\$1.68
Cable	9540-001	Various	\$1.57 per foot	N/A	5 ft	\$7.85
Female Banana Socket	N/A	Various	\$0	N/A	4	\$0
Banana Jack to Barrel Plug Adapter	N/A	Various	\$0	N/A	4	\$0
3000mAh Battery	N/A	MakerFocus	\$0	N/A	4	\$29.99
On/off/turn button	IP66	DMWD	\$2.80	\$13.99 / 5 buttons	4	\$13.99
Front Light (White LEDs)	JK2835AWT-P-B57EB0000-N0000001	Cree LED	\$0.10	N/A	20	\$2.00
Rear Light (Red LEDs)	MLESRD-A1-0000-000W01	Cree LED	\$0.88	N/A	10	\$8.80
Turn Signal Lights (Orange LEDs)	JE2835APA-N-0001A0000-N0000001	Cree LED	\$0.21	N/A	20	\$4.20
Indicator Light	5102H5-5	VCC	\$3.89	N/A	2	\$7.78
Programmer reverse current protection	MBR0520	Micro Commercial Components	\$0.23	N/A	5	\$1.15

The total cost of all our components is \$89. Total sales tax in the state of Illinois is 9.25% making our components total cost be \$97.23.

Our labor costs are calculated as follows:

Labor cost: (14 weeks) (10 hours) (3 people) (\$42/hr) (2.5 overhead costs) = \$44,100. Rounding up our total cost for this project is around \$44,200.

2.9 Schedule

Table 9: Project Schedule

Feb 22-28	Finish Control PCB prototype	Everyone
	Finish light PCB prototype	Everyone
	Finish Power breakout board	Everyone
	Order Digikey lights and batteries	Everyone
	Finish Design Document	Everyone
March 1-7	Breadboard components	Everyone
	Test signals on PCBs	Everyone
	Work on feedback given by design review	Everyone
March 8-14	Finalize PCB design	Everyone
March 15-21 (spring break)	No work	Everyone
March 22-28	Final PCB design submission	Everyone
	Solder PCB	Everyone
March 29-April 4	Testing	Everyone
	Finalizing/attaching components to bicycle	Everyone
	Solder PCB	Everyone
April 5-11	Team contact assessment	Everyone
	Progress demo preparation	Everyone

	Solder PCB	Everyone
April 12-18	Mock demo and presentation preparation	Everyone
April 19-25	Mock demo and presentation preparation	Everyone
April 26- May 2	Final demo and presentation preparation	Everyone
	Final paper writing	Everyone
May 3-9	Finish final paper	Everyone

3 Engineering Standards, Ethics, and Safety

3.1 Societal Impact

According to the National Highway Traffic Safety Administration (NHTSA), one of the most common bike accidents is collisions with cars. In 2023 alone, 1,166 bikers were killed in traffic accidents [13]. Our automatic bike light project will help to reduce bicycle traffic accidents by making bikes more visible on the street. The strobing setting of the front headlight and the rear light will make bikes more visible during the day, and the different constant light level settings will make the bike more visible at night. Additionally, the rear turn signal lights will alert surrounding vehicles the direction the cyclist intends to turn, which will help to prevent accidental collisions.

3.2 Ethics

One ethical consideration for our project is the effect of the brightness of the lights on the eyes of bystanders. If the lights are too bright, we risk blinding drivers and pedestrians, which could cause an accident. To mitigate this risk, we will ensure that our lights are under 1,000 lumens, which shouldn't pose an issue for our design, as each LED only abouts about 30 lumens.

Another ethical consideration for our project is the effect of the strobing lights on individuals in society with epilepsy, which is an important consideration in terms of the overall impact of our product. To prevent negatively harming these individuals, we will ensure our lights are of a similar brightness level to existing street and car lights.

Additionally, it is always important to adhere to the IEEE code of ethics and ethics guidelines for the Senior Design course. Part of the code of ethics is to take into account the safety, health, and well-being of the public, so to comply with this guideline, we will make sure to disclose all of our safety and ethical considerations to anyone interested in using our product.

3.3 Engineering Standards

3.3.1 IEEE Standard 1789-2015

IEEE Standard 1789-2015 is titled: modulating current in high-brightness LEDs for mitigating health risks to viewers [12]. This relates to our project because we are working with different light levels and strobing light effects. The recommended practice is to always use a frequency under 90 Hz for AC power, which we will be compliant with, since we are using DC power.

3.4 Safety

Bicycle lights are a crucial feature to ensure safe riding and protect the user from collisions. The underlying purpose driving our project is to save the battery of these devices to maximize the proper operation of these safety features and ensure reliable visibility for the user in any riding environments. Since the state of the lights depends on input from various sensors, there are several potential sources of error which could prevent the lights from turning on when necessary. A lack of lighting in low-light level or high-traffic areas could prevent other vehicles from seeing the bicycle, potentially leading to collisions causing serious bodily harm. To mitigate the risk of injury that could result from a lack of lighting, we will incorporate the following considerations into our design: Should the light sensor fail to properly detect light and turn the light on, the push button input should allow the user to override the input from the sensor and turn on the lights as necessary. Similarly, should the accelerometer fail to detect when the user is on the bike, the push button should allow the user to once again bypass the sensor, so as to prevent the light from prematurely turning off.

The safety features added to this bicycle light system are not a replacement for traditional safe bike riding practices such as wearing a helmet, being aware of your surroundings, and indicating your turn. As such, any technical failures on the proximity sensor or turn signal features of the system should not pose any additional risk to the rider. Should the LiDAR fail to alert the user of a vehicle approaching from behind, the user should continue to exercise caution and awareness of their surroundings by not wearing any headphones to allow them to hear any approaching vehicles.

4 Conclusion

In conclusion, our overall project design is an automatic bike headlight that turns on when the bike is moving, strobos during the day, and is on constantly at night. We also have a constant rear light for visibility. We will determine if the bike is moving using an accelerometer and determine day and night using a light sensor. Additionally, the bike will have a LIDAR sensor on the back of the bike that will warn the rider with a front warning light if a vehicle is approaching them within 40 meters. The rider will also have a turn signal button on the left handlebar to control turn signal lights on the back of the bike.

We must consider important safety concerns about possibly blinding drivers and pedestrians, and possible repercussions of if features like the LIDAR and the turn signals malfunction. For future work beyond the scope of this semester students could work on making this system wireless, for a more compact design.

References

- [1] A. Giannoulas, "Illinois Bicycle Rules of the Road," 2024.
https://www.ilsos.gov/content/dam/publications/pdf_publications/dsd_a143.pdf
- [2] IEEE, "IEEE Code of Ethics." Available:
https://ewh.ieee.org/cmte/substations/posted_documents/ieee_codeofethics.pdf
- [3] IEEE, "IEEE 802.3 ETHERNET," *www.ieee802.org*, Feb. 07, 2026. <https://www.ieee802.org/3/>
- [4] UL, "PREFACE." Available: https://code-authorities.ul.com/wp-content/uploads/2014/04/WC_MG.pdf
- [5] Microchip Technology Inc., "ATmega48A/PA/88A/PA/168A/PA/328/P," 2020.
- [6] Silicon Laboratories, "Si1145/46/47 PROXIMITY/UV/AMBIENT LIGHT SENSOR," 2022. Available:
<https://www.silabs.com/documents/public/data-sheets/Si1145-46-47.pdf>
- [7] Adafruit Industries, "Adafruit LIS3DH Triple-Axis Accelerometer Breakout," 2024. Available:
<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-lis3dh-triple-axis-accelerometer-breakout.pdf>
- [8] Garmin, "LIDAR-LITE V3HP OPERATION MANUAL AND TECHNICAL SPECIFICATIONS Laser Safety," Apr. 2020. Available: https://static.garmin.com/pumac/LIDAR-Lite_v3HP_Instructions_EN.pdf
- [9] Circuit Designer, "How to Use the Adafruit SI1145 Digital UV Index + IR + Visible Light Sensor," *Circuitdesigner.com*, 2025. <https://docs.circuitdesigner.com/component/fbb140ba-0251-a55e-c22c-3e9603f083c9/adafruit-si1145-digital-uv-index-ir-visible-light-sensor> (accessed Feb. 26, 2026).
- [10] Adafruit Industries, "Garmin LIDAR-Lite Optical Distance Sensor - V3," *Adafruit.com*, 2016.
https://www.adafruit.com/product/4058?srltid=AfmBOooqzvxTYkaEQwrGQkYEbq1llMe3r6Yx2o2C52U_M_kEGA2WI-c (accessed Feb. 27, 2026).
- [11] B. Krump, "What Are the Most Common Causes of Bicycle Accidents? | Bicycle Accidents," *Ben Crump*. <https://bencrump.com/faqs/what-are-the-most-common-causes-of-bicycle-accidents/>
- [12] Institute of Electrical and Electronics Engineers, "IEEE Recommended Practices for Modulating Current in High-Brightness LEDs for Mitigating Health Risks to Viewers," *IEEE.org*, 2015.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7118618&tag=1>
- [13] National Highway Traffic Safety Administration, "Bicycle Safety," *NHTSA*, Dec. 18, 2018.
<https://www.nhtsa.gov/road-safety/bicycle-safety>