ENEL 300- Electrical and Computer Engineering Design 2

Department of Electrical and Electronic Engineering
University of Canterbury

Design and Build Project

Implementing Beam Steering for Bicycle Lights

GROUP MEMBERS

Hassan Alhujhoj (35352623)

Saranya Ramani (27179399)

Sasiru Goonatillake (51643980)

Stuff needs that needs to be finished or revised

- Background (revision sasi)
- Requirements (revision sasi)
- Specifications
    - Parts (revision Sasi)
    - Code (Hassan)
    - Circuitry (sasi)
    - Something regarding the case (sasi)
- Costing & marketing (Saranya &or Hassan)
- Conclusion (Saranya)

Executive Summary

The document is based on the project called 'Automatic Beam Handle Bike lights. The document describes the development of the prototype programmed on Arduino Nano. The project is a prototype for mainly for mountain bikers, triathletes and cycle commuters. Mountain bikers require high-resolution bike lights while biking at night. They generally use two separate lights, one placed on the handlebar and the other is a headlight. The regular bike lights fail to illuminate the area of turn. Therefore, imposing the risks of misjudging the narrow area of turn. It also becomes difficult for cycle commuters to commute through steep areas with poor lighting.

The regular bike lights can be redesigned by optimising the electrical and mechanical features. The circuitry of the bike lights can include a microcontroller, controlling the sensors of the circuit. For detecting the direction of turning and the effectively calculate the angle, a motion-tracking sensor can be used. In order to signal the LED's (Light Emitting Diodes) to illuminate the path, a circuitry is required. The mechanical consideration of the project would to design a case for holding the various electrical components. The designed case must withstand rough weather and temperature conditions.

The angle measurement and motion tracking are done by using the gyroscopic sensor. The LED's are connected to the front surface of the case. The case is rectangular but has curved edges at the front. One LED is connected to the centre and the other two LED's are connected to either side of the curved edges. The gyroscopic sensor is placed inside the case. When the handlebar turns, the gyroscope sensor measures the angle and signals the Arduino Nano. The Arduino Nano computes the direction and sends PWM signals to the base of the Mosfet. The Mosfet acts as a switch and controls the intensity of the LED's. If the biker turns right, the Brightness of the LED's descends from the right to the left and vice versa.

In this project, Arduino Nano will be used to process the angle rate of change that is produced by the gyroscope sensor. Since the Arduino module cannot power the circuit that controls the LEDs, it is necessary to find a solution to the power that the LED control circuit in another way. The simplest method is to power the control circuit by an external DC battery while controlling this circuit with Arduino. This control circuit consists of a MOSFET that would receive the control signal, i.e. PWM signal, from the Arduino. It will also consist of the super-bright LEDs and a resistor to limit the current flowing directly from the DC battery.The software mainly uses mathematical equations to map the handlebar angle to a beam steering angle which in turn is used to produce three PWM wave outputs which controls the brightness of each LED.

The project can be improved by installing Surface Mount LED (SMD). They are cheap, provide high lumens and are very durable as compared to the LED's used in the project. Second improvement would be to install external portable batteries. They have a longer shelf life and can be recharged after use. The third addition would be to have an Anti-glare design. The high intensity of beams can blind motorists; therefore, anti-glare systems encompass half-cut conical shapes. The beam of the LED's points downwards. The fourth addition would be to install an LDR to control the

intensity of the LED's. The LDR can regulate the overall intensity of the LED's. The LED's can radiate more if the biker is travelling through a poorly lit street or radiate low beams when travelling through a well-lit street.

Table of Contents

1 Background

Biking is not only seen as a mode of transportation but also as a hobby or for sports. Biking, like any other activity can present some dangers to bike riders. Therefore, cyclists need to consider safety measures. Bike lights, helmets and reflective clothing are essential safety considerations. Bike riders are generally concerned while riding bikes at night. During the night hours, pedestrians and motorists find it difficult to spot bikes on roads and streets. Therefore, bike lights are an important piece of safety equipment for cyclists.

Mountain bikers and triathletes need bright bike lights with wide beams to illuminate the sharp turns and other features of a trail at night. It is common practices to use two separate lights with a powerful wide beam flood light fixed to the handlebar to light up what's ahead of the bike and a more focused beam light on the helmet to light up where the rider is looking. The requirement for multiple lights for biking increases cost and reduces power efficiency due to the need for multiple batteries. Two bike lights are used because a light fixed to the handle bar lights up the area perpendicular to the handle bar as seen in Figure 1, which is generally not where the trail is headed hence the need for a helmet light. Another factor which affects this lighting problem is when the rider leans to a side while cornering at higher speeds. This reduces the angle at which the handlebar is turned further amplifying the lack of illumination of the trail by the handle light.

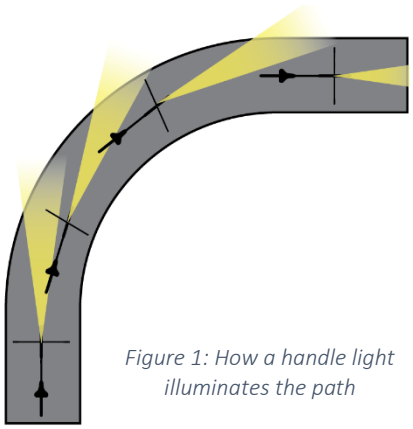


Figure 1: How a handle light illuminates the path

The current solution to this is using two lights of which one is a power-hungry flood light to light up a wide area in front of the bike. This same phenomenon plagues cycle commuters. Though using floodlights would illuminate a wide area and the turn, they are less energy efficient. Since regular bike lights fail to intensify the turn, cycle commuters sometimes find it difficult to commute through narrow turns of a poorly lit street at night. Ill equipped cyclist could misjudge turns in the road or the trail, leading to accidents or falls.

2 Requirements

In order to solve the issues concerning bike lights, the design needs to be revamped. The problems such as incomplete coverage of the turn and multiple use of lights for safety can be solved by improving the circuitry of bike lights. The revamped circuitry would eliminate the need for multiple lights (headlights and bike lights) and automatically illuminate the area of turn. To overcome the issues as mentioned above, a few electrical and mechanical changes need to be executed. The mechanical changes such as installing the circuitry in a case need to be considered. The case needs to be built considering the application, providing efficient illumination of the turn.

The circuitry is an important aspect of the system design. It needs comprise of the correct electrical components to fulfil the requirements. In order to detect the turn area (left or right), the circuitry

must include a motion-tracking sensor. The circuitry must also include a microcontroller/processor to control and compute the action of each sensor. It acts as a brain of the system and responds to the requirements of the code. In addition, a circuitry for controlling the intensity of the LED's, needs to be implemented.

The circuitry needs to have a powering device such as a battery for activating the system. For complete control of the bike lights, the user must be allowed to operate a switch to turn the system ON/OFF. The switch would determine the operating state of the system. The circuit must be designed to be stable and firm to face harsh circumstances. The material of the case needs to be made considering changes in temperature, seasons and durability of material.

3 Specifications

The regular bike lights fail to perceive and illuminate the area of turn, thus, imposing risks of injuries to mountain bikers and cycle commuters at night. The original design of the bike lights can be modified by implementing motion-tracking sensors. The motion-tracking sensor would enable automatic angle calculation and would illuminate the area ahead of the biker, precisely following the turn. The sensor placed in the case would control the intensity of the LED's based on the movement of the handlebar. The redesigned bike lights would also have high lumens LED's to increase the intensity of the bike lights and act as a replacement to headlights. The goal of the system design to use a sensor to accurately compute the angle of the handlebar and signal the LED's.

3.1 Main components

3.1.1 MPU6050:

It is a Motion Tracking device. It consists of a combination of six axis – a three-axis gyroscope and a three-axis accelerometer. The device can be used for measuring rotation velocity, orientation and displacement. The Yaw, Pitch and Roll can be calculated using the MPU6050

3.1.2 Arduino Nano:

Arduino Nano is a small and compatible version of Arduino Uno. The Arduino Nano is based on the ATMega328p microcontroller. The operating voltage of the Nano is 5V and the input voltage can be varied from 7V to 12V. It consists of 14 digital pins, 2 reset pins, 8 analog pins and 6 power pins. The Arduino Nano consists of a crystal oscillator and operates on a frequency of 16MHz. For the project, the team utilized pins A3, A5 and A7 for producing the PWM signals from the Arduino.

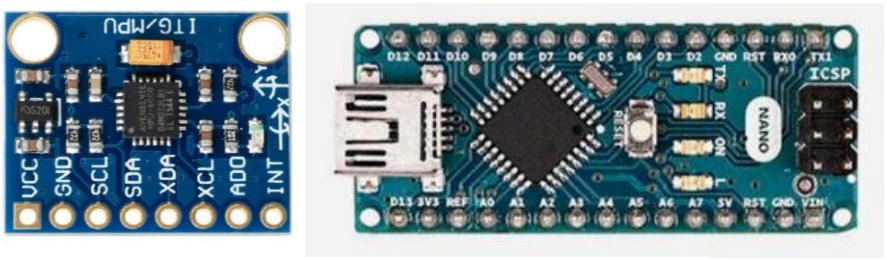


Figure 2: MPU6050 modle and the Arduino Nano
https://store.arduino.cc/usa/arduino-nano

3.2 LED Control Circuit

Since super bright LEDs require a significant amount of current to operate at full brightness, they cannot be directly connected to the Arduino Nano. The Arduino cannot simply supply all that current to the LEDs and will be limited to 40mA at each PWM pin, and all the pins will provide a combined current of 200mA [1]. Thus, the current at each PWM pin must be limited to say 30mA to avoid damage to the Arduino module. AA alkaline batteries can be used instead to power up the LEDs and the rest of circuitry since they can provide constant 50mA and 1.5V [2].

Thus, since the Arduino is unsuitable for the task power, the LEDs it would be used only to control the LEDs, as seen in Figure 1. This circuit is based on the fact that MPU6050 gyroscope will provide the Arduino chip with rate of change of the handlebar of the bicycle and based on the Arduino code written the Arduino will change the duty cycle of the PWM signals that would be used in each pin. In this project, only 3 LEDs will be used, and thus only 3 PWM signals would be needed from the Arduino. The circuit in Figure 1 is what would be most suitable to perform the task of changing the intensity of lighting on the bike. That circuit only requires four components: a battery, an LED, a resistor, and an N channel MOSFET.

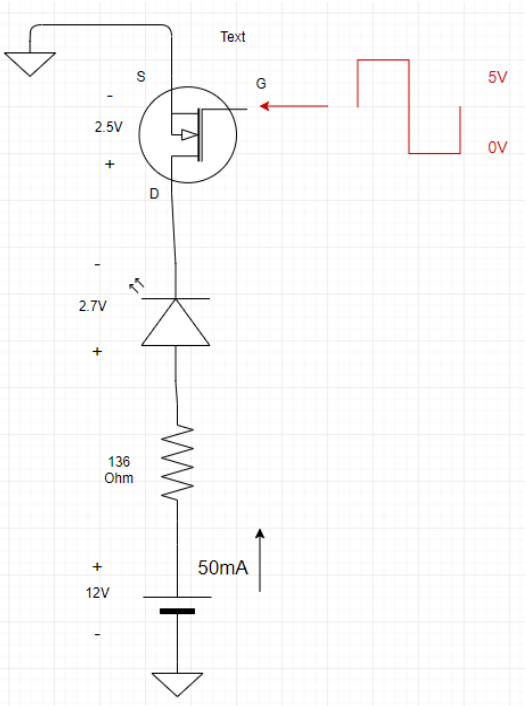


Figure 3: LED control circuit.

A MOSFET is used because it is voltage controlled, and it is easier to control. A BJT can as well used, but the base current must be controlled to be βI_c or$100I_c$, which is usually the current gain for many BJTs. With MOSFETs, as long as the control voltage Vg is higher than the threshold voltage Vgs(th), the MOSFET will be on conducting or it will be off if the PWM signal is lower than the threshold voltage. For 2N7000, which is the N-channel MOSFET used in this design, the datasheet specifies that the threshold voltage is 0.8V low enough and the transistor is excellent to use in the project [3].

The MOSFETs, resistors and LEDs are all readily available in the Electrical Wing shop. Since the 2N7000 MOSFET and the super bright 2.7V LEDs are available in the shop, the only selection left is the battery and the resistance needed to limit the current in the circuit. To select the battery size, the battery should be suitable to be applied to the Arduino Vin pin. Since the Arduino Nano can accept from 7-12V input voltage, a 12V battery is most suitable. The battery has to be big enough to power the Arduino module, and three LED control circuits as can be seen in Figure 2 and
Figure 3. Once the battery is selected, the resistor value should be calculated based on the supply battery current which is 50mA. This value can be calculated by Ohm's law, i.e. $R = \frac{12V-5.2V}{50mA} =$

136Ω. The 2.5V seen across the transistor is the difference in voltage between the drain and source when the transistor is on, refer to datasheet [3].

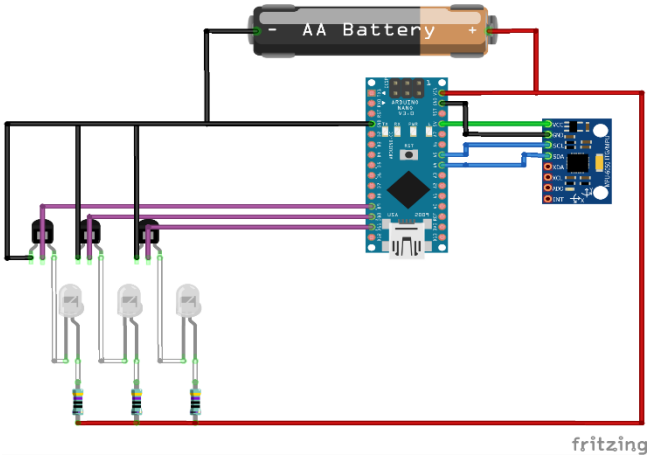


Figure 5: Automatic Beam Bike Light Schematic.

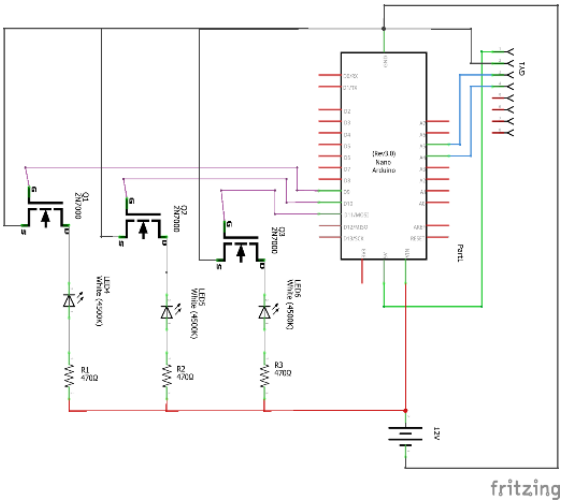


Figure 4: Schematic of bike light.

3.3 Software

The software for the nano was written and edited in Arduino IDE. The primary tasks of the software are tackled in the main loop of the code as shown in Figure 6. The software implements and open loop control system which takes the angle of the handlebar, processes it and controls the intensity of 3 LEDs to provide beam steering capabilities to a bike light. The complete software used by the nano is provided in appendix 1.

The angle of the handlebar is measured using a 6-axis motion tracking module (mpu6050) consisting of a gyroscope and an accelerometer. The software utilizes an Arduino library MPU6050_tockn (tockn, 2019) to communicate with the mpu6050 module. The setup function sets the serial communication speed of the nano to 9600 bps and calibrates the mpu6050 module. The function *readAngle()* reads raw the value of the raw angle Z which is calculated by integrating the rate of change of the angle measured using the gyroscope.

```
void loop() {
  handleAngle = readAngle();
  readLDR();
  beamAngle = amplifyHandleTurn(handleAngle);
  beamSteering(beamAngle);
  PWMgen();
  dataOut();
  delay(50);
}
```

Figure 6: Main loop

The angle for the beam steering is decided by the function *amplifyHandleTurn()* which uses the equation $y = \frac{40}{1+e^{-0.1*x}} - 20$ which outputs graph shown in Figure 8. The *beamAngle* is then used to determine the intensity of each LED using the function *beamSteering().* This function uses three equations as shown below in Figure 7, the red graph shows the intensity on Y-axis and the beamAngle on the X-axis for the Middle LED. The other two graphs show the intensity levels for the side LEDS.

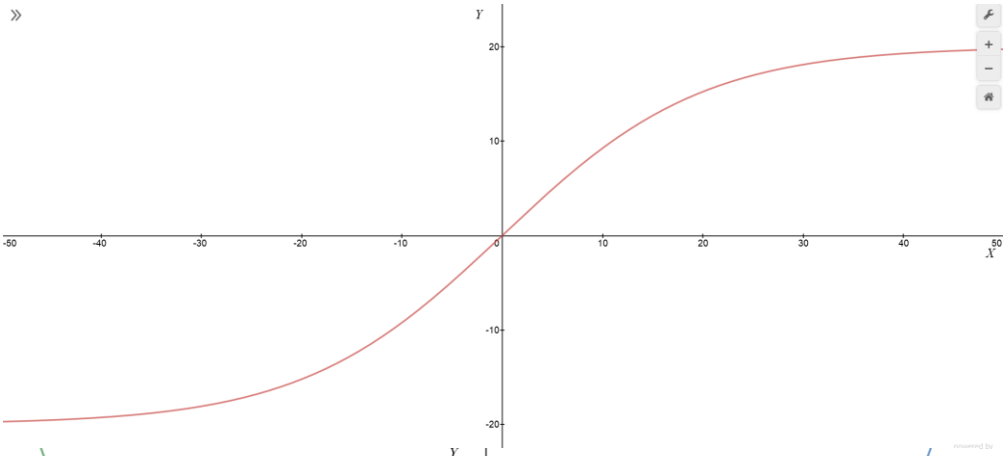


Figure 8: Graph of beamAngle agaist handleAngle.

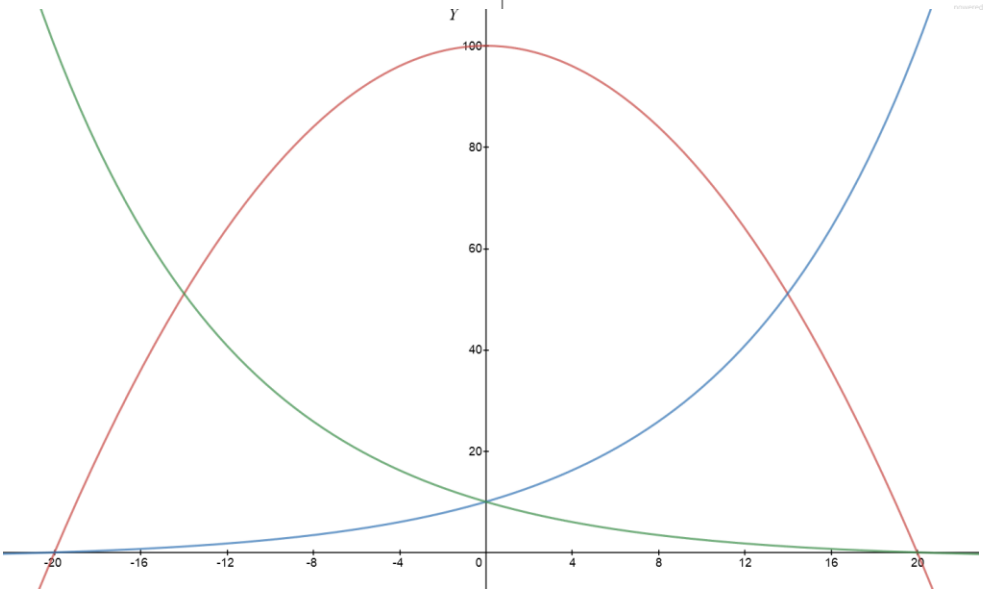


Figure 7: Equations for the intesnity of each LED

The functions *pwmGen()* then generates 3 PWM wave outputs which correspond to the intensity graphs which controls the brightness of the LEDs. The PWM waves uses the digital pins 3, 5 & 6 as output pins.

3.4 Case

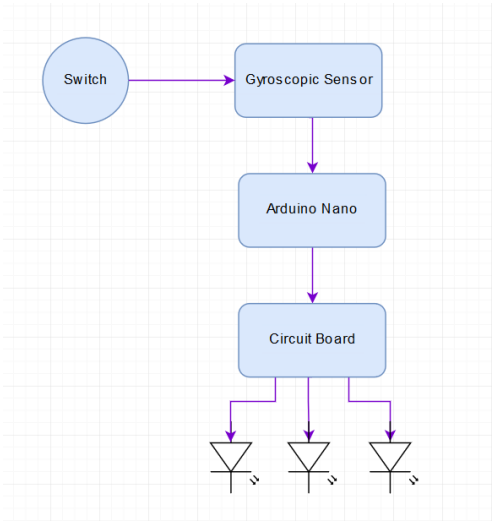
The mechanical design of the system is equally important as the electrical design. The mechanical design includes modelling and material considerations. The electrical circuitry of the system would be enclosed in a case. The prototype of the case is modelled on SolidWorks. The front surface of the case is curved at the edges and there are three LED's popping out of the surface of the case. One LED is centred and the other two are angled on each of the edges of the case. The holes are parabolic in shape to provide optimum dispersion of LED photons. The case incorporates the circuitry, switch and the battery pack. The material that can be used for developing the case would be made from hard plastic. This type of plastic would help the case thrive through rough conditions such as temperature, rains and rough handle. The issues and inaccuracies can be solved by

redesigning bike lights. The handlebar would comprise of three LED's placed on the outer surface of the case. The problem of detecting and illuminating turns can be solved by installing a motion-tracking sensor. The sensor would be installed on the handlebar detecting the angle of movement. It would signal the LED's connected to the circuitry to highlight the area of turn. Therefore, mitigating the probability of misjudging the road.

The issue of battery efficiency can also be solved by installing the redesigned bike lights. They eliminate the need for headlights, making it more energy efficient.

3.5 Working of the Product

The Gyroscopic sensor is placed on the handlebar. It calculates the rotation angle of the handlebar. For this project, the Gyroscope measures the angle along the Z-Axis. The Arduino Uno acts as an interface between the Gyroscope and the LED's. The output from the gyroscopic sensor is fed into the Arduino Nano which activates the PWM signals available on the pins A3, A5, A7. These signals are applied as an input to the base of the 2N7000 Mosfet, which acts as a switch. The input to the Mosfet activates the LED's (Light Emitting Diodes) and controls the intensity of light. For example, if the handlebar is turned towards the right, then the intensity of the three LED's descends from the left to the right and vice versa. The Mosfet is connected to a 100kΩ resistor. The circuitry is enclosed in a case, with the LED's placed on the front side of the case. The 12V battery is shared between the Vin pin of the Arduino Nano and the Circuitry which includes powering the Mosfet and the LED's. The switch is used to turn the system ON or OFF.



4 Discussion

4.1 Costing:

This section describes the costing of the components used for the 'Automatic Beam Handle Bike Lights'. The various components, their individual cost and the total cost of the project are listed below in Table 1.

Table 1: Table of Costs

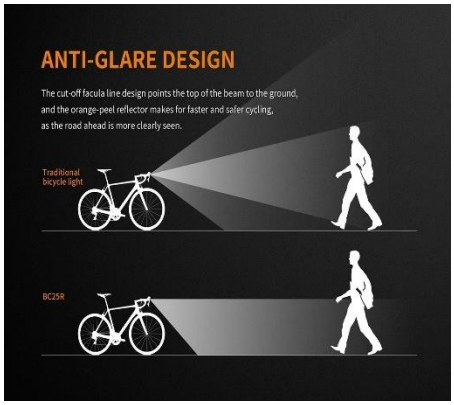
Components	Cost for each component	Total Cost
Arduino Nano	$0.00	$0.00
Gyroscopic Sensor	$4.90	$4.90
Transistors	$0.75	$2.25
LED	$6.00	$18.00

Resistors	$0.00	$0.00

4.2 Improvements

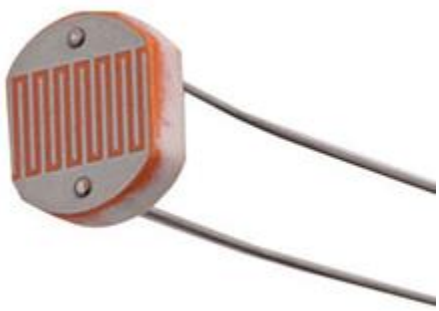
Surface Mount LED (SMD): SMD's are compact and can perfectly be soldered onto a PCB. They are durable and provide high lumens approximately 1080 lumens per meter. They are cheap and consume less power, therefore can act as a replacement to High lumens LED's. Considering the durability of regular LED's used in bike lights these LED's can serve for a longer duration providing more lumens output.

External Batteries: The designed system, currently comprises of an internal battery, which has a short shelf life and would dissipate energy at a very high level. Thus the biker would have to replace his batteries regularly. External rechargeable batteries are energy efficient and are suitable for regular use. They are portable and can be recharged after use. They charge quickly and have a longer shelf life as compared to the alkaline batteries currently used in the project.



Anti-Glare Design: If the bike lights comprise of LED's of high lumens, they tend to reflect a high intensity of light. If there are incoming motor vehicles, the intensity of the bike lights can blind the motor vehicles. Therefore, to eliminate this issue, bike lights can be designed in a half-cut conical shape to point the beam towards the ground as shown in Fig XXXXX. It provides safety to the cycle commuter and clearly illuminates the road. Therefore, anti-glare design would not cause visibility issues to incoming motorists.

LDR to turn the system ON/OFF:



LDR (Light Dependent Resistor) varies its resistance depending upon the intensity of the light. For future improvements we would use the LDR to regulate the overall intensity of the LED's. If the biker is travelling through a poorly lit street, the intensity of the lights would be more and if he's travelling through a well-lit area then the intensity of the LED's would be low.

5 Conclusion

6 References

[1] Farnell, "Arduino Nano," http://www.farnell.com/datasheets/1682238.pdf, n.d., p. Arduino Nano datasheet.

[2] A. Kumar, Can we achieve 1A current from 1.5v battery cell, if yes for how long?, https://www.quora.com/Can-we-achieve-1A-current-from-1-5v-battery-cell-if-yes-for-how-long#targetText=Normal%20AA%2FAAA%20batteries%20that,(Source%3A%20Wikipedia)., Mar 7, 2016.

[3] ONSemi, "Small Signal MOSFET 200 mAmps, 60 Volts," https://www.onsemi.com/pub/Collateral/2N7000-D.PDF, 2011, p. 2N7000 datasheet.

[4] Farnell, "Arduino Nano," http://www.farnell.com/datasheets/1682238.pdf, n.d., p. Arduino Nano datasheet.

[5] A. Kumar, "Can we achiece 1A current from 1.5V battery cell, if yes for how long?," https://www.quora.com/Can-we-achieve-1A-current-from-1-5v-battery-cell-if-yes-for-how-long#targetText=Normal%20AA%2FAAA%20batteries%20that,(Source%3A%20Wikipedia)., Mar 7, 2016.

[6] OnSemi, "Small Signal MOSFET 200mAmps, 60 Volts," https://www.onsemi.com/pub/Collateral/2N7000-D.PDF, 2011, p. 2N7000 datasheet.

7 Appendix

Code for project

```
/*
  Beam steering for a bike light


  This code uses input from an accelerometer and a gyroscope (mpu6050) to steer
  the light beam of a bike light towards which way the bike is intending to travel.
  Amplifies the turn of the handlebar to turn the light to see where you are going.
  Written for the Design & build project of ENEL300 course of University of
```

```
  Canterbury, Electrical & Computer Engineering Department.

  Author:      Sasiru Goonatillake
  Date:        Aug 29th 2019
  Last Updated: Sep  6th 2019
*/

#include <MPU6050_tockn.h>
#include <Wire.h>

#define LDR_PIN A3
#define ANALOG_RANGE 1023
#define LUX_CONTROL 70    // a value between 0 and 1 to decrease the overall intensity as a percentage
#define LED1 3
#define LED2 5
#define LED3 6
MPU6050 mpu6050(Wire);

//Variables
double gyroOut, handleAngle, beamAngle, amp, lux, invLux, holder;
int LDRValue, PWMs[2], LED_DC[2];

//_____________________________________________________________________________
____

void setup() {
  Serial.begin(9600);
```

```
  //  TCCR0B = TCCR0B & B11111000 | B00000100;    // set timer 0 divisor to    256 for PWM frequency of   244.14 Hz of pin 5 & 6
  //  TCCR2B = TCCR2B & B11111000 | B00000101;    // set timer 2 divisor to    128 for PWM frequency of   245.10 Hz of pin 3
  gyroSetup();
}

void loop() {
  handleAngle = readAngle();
  readLDR();
  beamAngle = amplifyHandleTurn(handleAngle);
  beamSteering(beamAngle);
  PWMgen();
  dataOut();
  delay(50);
}

void dataOut() {
  Serial.print("\tangleZ : ");
  Serial.print(handleAngle);
  Serial.print("\t   amp : ");
  Serial.print(amp);
  Serial.print("\t   LDR Value : ");
  Serial.print(LDRValue);
  Serial.print("\t   invLUX : ");
  Serial.println(invLux);
  Serial.print("\tDC0 : ");
  Serial.println(LED_DC[0]);
```

```
  Serial.print("\tDC1 : ");
  Serial.println(LED_DC[1]);
  Serial.print("\tDC2 : ");
  Serial.println(LED_DC[2]);
}

/*
   Sets up I2C communcicaitons with mpu6050 and calibrates the gyro for each reset
*/
void gyroSetup() {
  Wire.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);
}

/*
   Read the Z axis angle change
*/
double readAngle() {
  mpu6050.update();
  gyroOut = mpu6050.getAngleZ();
  return gyroOut;
}

/*
   Read the resistance value given by the Light Dependent Resistor
*/
```

```
void readLDR() {
  LDRValue = analogRead(LDR_PIN);
  lux = LDRValue / ANALOG_RANGE;
  holder = 1 - lux;
  invLux = (map(holder, 0, 1, LUX_CONTROL, 1)) / 100;
  invLux = 1;
}

/*
   Uses the predetermined 4th order polynomial to estimate the amplification needed
*/
double amplifyHandleTurn(double x) {
  //  amp =  -0.0001*pow(x,3) + (2*pow(10,-20))*pow(x,2) + 0.7407*x + 4*pow(10,-12);
  amp = (40 / (1 + (exp(-0.1 * x)))) - 20;
  return amp;
}

/*
   Creates 3 pwm waves for the LEDs based on handle bar position
*/
double beamSteering(double angle) {
  PWMs[0] = 10 - (1.235939 / 0.1098612) * (1 - exp(0.1098612 * angle));
  PWMs[1] = 100 - ((pow(angle, 2)) / 8);
  PWMs[2] = 10 - (1.235939 / 0.1098612) * (1 - exp(-0.1098612 * angle));

}
```

```
/*
  Creates 3 PWM waves for the 3 LEDs to control the intestiy of each according to PWMs[] array
*/
void PWMgen() {
 LED_DC[0] = map(PWMs[0], 0, 100, 5, 255);
 LED_DC[1] = map(PWMs[1], 0, 100, 5, 255);
 LED_DC[2] = map(PWMs[2], 0, 100, 5, 255);

 analogWrite(LED1, LED_DC[0]);
 analogWrite(LED2, LED_DC[1]);
 analogWrite(LED3, LED_DC[2]);
}
```

Circuit schematic