

# HIGH EFFICIENT STAND-ALONE STREET LAMP

---

By

Collin Hasken

Xiaolou Huang

Final Report for ECE 445, Senior Design, Fall 2017

TA: Kexin Hui

13 December 2017

Project No. 11

## Abstract

This project was one of three parts involved in making a solar powered LED street lamp with a smartphone app to view circuit and battery statistics. We focused on discharging the battery to increase the luminance of the LED as the ambient light grew dimmer and to provide a user interface to view the battery's state of charge (SoC) and other statistics of the circuit and set a dimming interval. We were unable to combine the ambient light detection and dimming circuit in the finish product and opted for only the ambient light detecting circuit. The device can also be configured to the user's WiFi router, allowing it to update an online database with circuit statistics and the battery's SoC. This data is then displayed in graphs and table on the smartphone app. Before going to market, the dimming circuit and data from other groups needs to be integrated.

## Contents

1. Introduction .....	1
1.1 Objective .....	1
1.2 Functions .....	1
1.3 Blocks .....	1
2 Design .....	2
2.1 Power Supply .....	2
2.1.1 Voltage-regulator 3.3V .....	3
2.1.2 Fuel Gauge .....	4
2.2 Output .....	4
2.2.1 Discharging circuit .....	4
2.2.2 Luminance controller .....	5
2.3 Control Unit .....	5
2.3.1 Microcontroller .....	5
2.3.2 WiFi-module .....	7
2.3.3 App .....	9
3. Design Verifications .....	10
3.1 Fuel Gauge .....	10
3.2 Discharge Circuit .....	10
3.3 WiFi Chip .....	10
4. Costs .....	11
5 Conclusions .....	11
5.1 Accomplishments .....	11
5.2 Challenges .....	11
5.3 Future Work .....	12
5.4 Ethical Considerations .....	12
6. References .....	13
Appendix A Requirement and Verification Table .....	14
Appendix B Circuit Diagram .....	15

# 1. Introduction

## 1.1 Objective

The problem we addressed was discharging a lead acid battery to power an LED when it's dark outside and provides circuit statistics and the battery's state of charge (SoC) to the user. Having the LED turn on due to ambient light allows the lamppost to be autonomous and efficient. Being able to look at the battery's SoC lets the user know if and when the lamp will turn off and see trends in the charging and discharging of the battery. The device, once configured, will automatically upload data to an online database allowing the user to use their smartphone app to check the data anywhere.

The remainder of chapter 1 will go over the project's functions and design blocks. Chapter 2 will go through each of the important design blocks, including a detailed description, equations, simulations, successes, and challenges. Chapter 3 will go through the requirements and verifications for the project. Chapter 4 will cover the cost of the project such as the parts required for the product as well as the calculated salaries for our work. Chapter 5 will conclude with overall accomplishments, future work, and ethical considerations.

## 1.2 Functions

The device can detect the ambient light and adjust the LED luminance to be brighter when there is less ambient light. It can also read the output voltage of the battery and convert it to a SoC percentage from 0 to 100. The microcontroller can send a signal when the current time is in between a user configured dimming period to allow for battery saving during low usage times. A smartphone can connect to the device to configure the time as well as the user's WiFi router's SSID and password to allow the device to continuously upload the battery's SoC and circuit statistics to an online database. The app can also retrieve data from the database to show the current statistics as well as display graphs for trends over time.

## 1.3 Blocks

The main blocks from our design in Figure 1.1 are the fuel gauge, discharging circuit, WiFi, and app. The microcontroller receives the fuel gauge and other group's data and sends it to the WiFi chip. The microcontroller controls the dimming of the light during the dimming interval while the discharge is based off of the ambient light detection. The WiFi chip both sends data to the database and receives data from the smartphone app to retrieve WiFi router configuration and the dimming time interval. The app allows the user to send the configuration data and view the data in an aesthetically pleasing format. The WiFi chip we purchased had a microcontroller as well as a 5V voltage regulator. This left us with only needing to create the 3.3V voltage regulator.

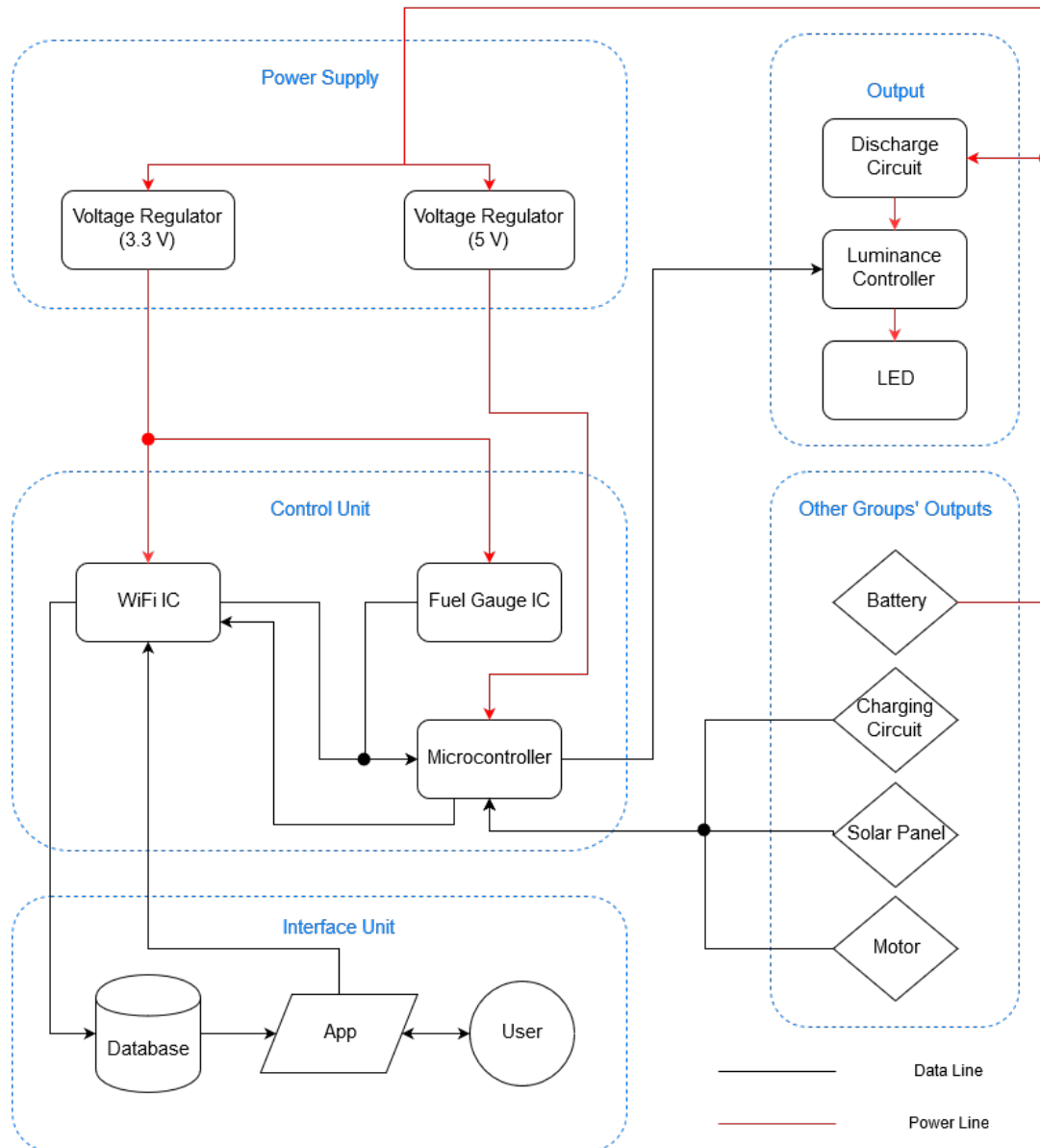


Figure 1.1 Block Diagram

## 2 Design

### 2.1 Power Supply

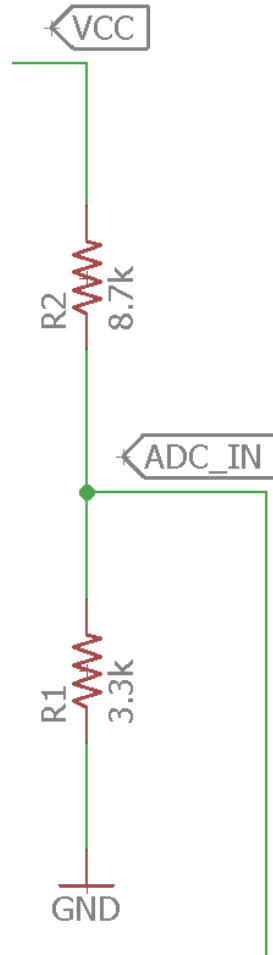


Figure 2.1 Voltage regulator schematic

### 2.1.1 Voltage-regulator 3.3V

The maximum voltage for the WiFi's analog to digital converter (ADC) is 3.3 V. The ADC needs to receive the voltage the battery is outputting to determine the battery's state of charge (SoC), so the voltage must be converted from a 0 through 12 V range to 0 through 3.3 V. We had originally planned on creating a buck converter, however we used a simple voltage divider shown in Fig. 2.1. The two

resistors values of 8.7 k $\Omega$  and 3.3 k $\Omega$  were calculated from  $\frac{V_{out}}{V_{in}} = \frac{R_1}{R_1 + R_2}$  leading to  $\frac{3V}{12V} = \frac{3.3 \text{ k}\Omega}{3.3 \text{ k}\Omega + 8.7 \text{ k}\Omega}$ .

We calculated the power consumption with  $P_2 = \frac{V^2}{R_2} = \frac{144}{8700} = 16. \text{ mW}$  for  $R_2$  and  $P_1 = \frac{10.89}{3300} = 3.3 \text{ mW}$  for the  $R_1$ . These wattages are low enough to not affect the rest of the circuit.

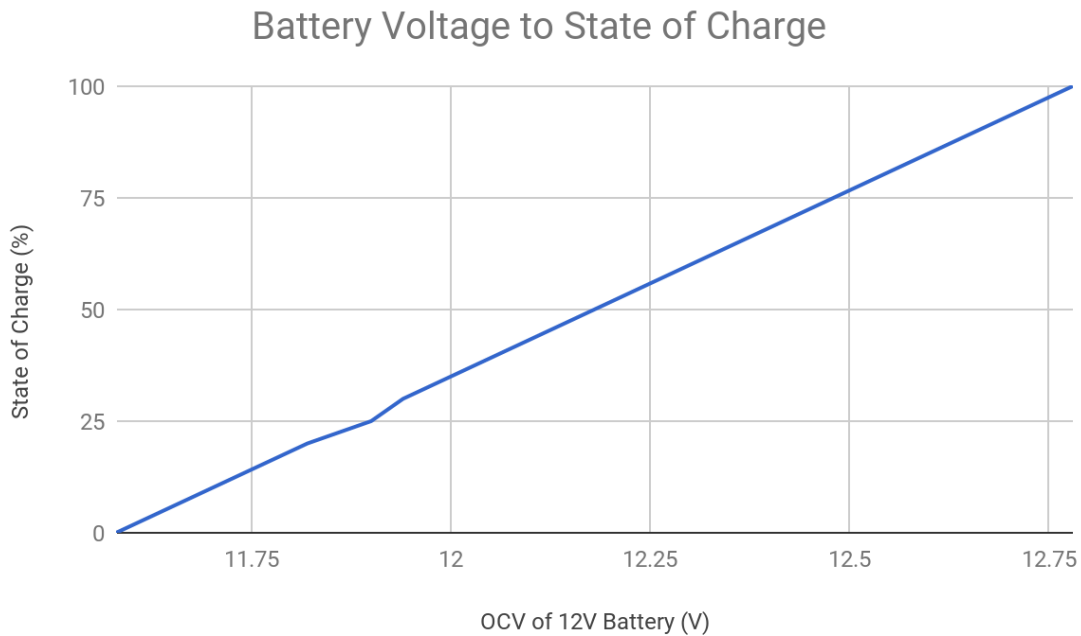


Figure 2.2 Battery voltages to state of charge constructed from battery manual

### 2.1.2 Fuel Gauge

For the fuel gauge, we need to retrieve the voltage output from the battery and convert it to a SoC percentage. The voltage is ran through the voltage regulator shown in Fig. 2.1 to bring it down to 3.3 V for the microcontroller's ADC. The voltage is converted back to 12 V with  $V = \frac{\text{sum}}{\text{samples}} * \frac{1}{1024} * 12V * 1.314368$ . The average input is found between multiple readings and divided by 1024 because the ADC's values range from 0 to 1024. Then, the read voltage is converted back to a 12 V range and multiplied by a tuning constant. Using a table from the battery's manual [1] we constructed a graph, Fig. 2.2, to find an equation that will convert the voltage to a SoC percentage. The equation we found is  $y = 83.6x - 968$ .

## 2.2 Output

### 2.2.1 Discharging circuit

To save on energy, the LED needs to turn on only when the ambient light becomes dark enough to necessitate the light to be on. This is achieved through using a photo resistor to change the current going through the LED. To have the LED turn on and off when needed, we first measured the resistance of the photo resistor when the ambient light was low, resulting in 1.3 MΩ. Next we measured the resistance when bright light was applied and found 14 kΩ. This result means we need to have the photo resistor in parallel along with the use of a BJT to make the current going to the LED higher when the photo resistor receives less resistance from bright light. Through testing different resistor values, we were able to find two resistors of value 1.1 kΩ and 27 kΩ worked well to have the LED be turned off during daylight hours and gradually turn on until brightly shining at total darkness. The full circuit is shown in Fig. 2.3.

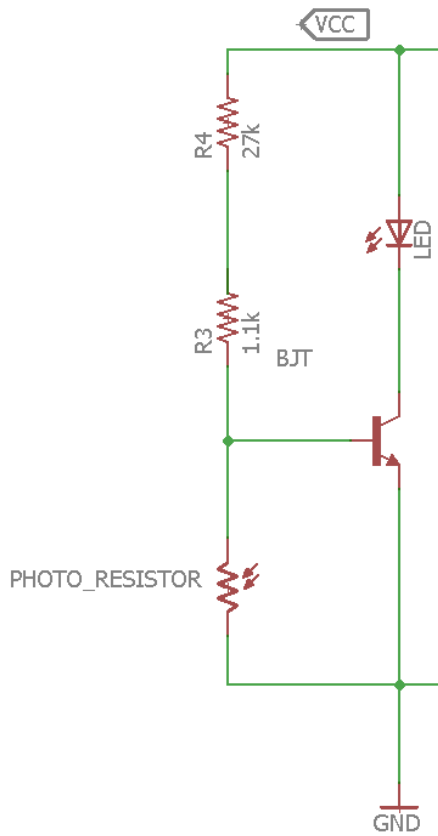


Figure 2.3 Discharging circuit

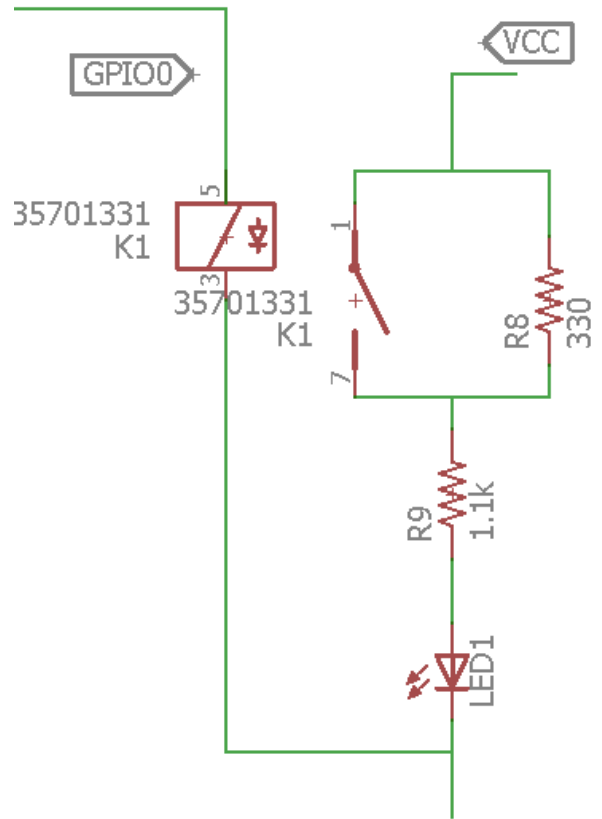


Figure 2.4 Luminance controller circuit

## 2.2.2 Luminance controller

For the luminance control, we needed to be able to allow the user to set a time interval for the LED to be dimmer for foreseen low usage times to save on power. This process requires the smartphone app to have an interface for the user to set a time and to be able to send the message to the WiFi controller. Then the microcontroller needs to store the time interval and query the [nist.time.gov](http://nist.time.gov) website to receive the current time. Then send a signal dependent on the current time falling in between the set interval. This signal would be able to short a resistor in the circuit, resulting in the LED becoming brighter or dimmer.

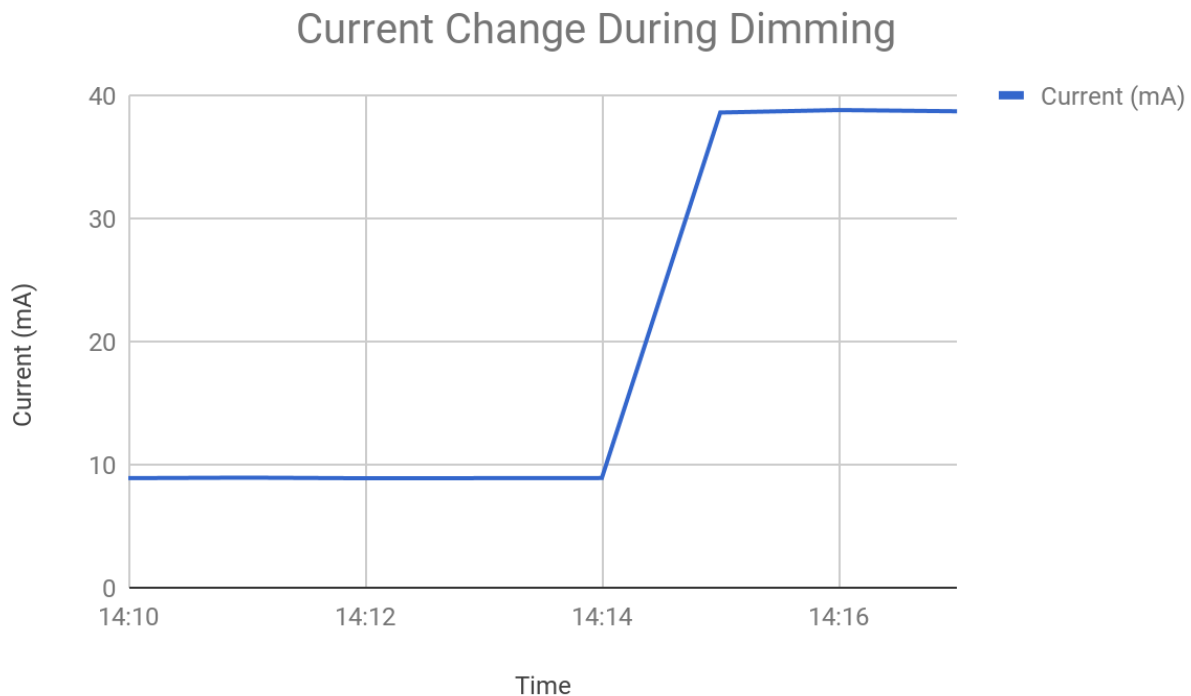
We used the circuit shown in Fig. 2.4 to dim the light. Through multiple testing, we found a 330  $\Omega$  and 1.1 k $\Omega$  resistor gave us a visible difference. The microcontroller can then send a 3.3 V signal to make the relay short out the 1.1 k $\Omega$  resistor, increasing the current from 8.9 mA to 38.6 mA as shown in Fig. 2.5 on the next page.

## 2.3 Control Unit

### 2.3.1 Microcontroller

The microcontroller calculates the battery's SoC, interfaces with connected devices to receive configuration information, send a signal dependent on the current time and dimming interval, and creates a message to send to an online database. The overall control flow can be seen in Fig. 2.6 on the next page. There is a flash memory that the microcontroller can save the WiFi router's SSID and





**Figure 2.5 Measured current change from dim setting to bright setting**

password as well as the dim time interval. Upon turning on, the controller will try to load saved data to allow it to reconnect to the router without the user needing to reconfigure. When a user connects, they can send a message through the phone app to the WiFi chip. This message is then sent to the controller. The controller can then parse the request to retrieve either the user's WiFi router information or the dimming time interval.

The microcontroller receives the voltage from the voltage regulator in Fig. 2.1 and calculates the SoC as discussed in Chapter 2.1.1. If we had data from other groups, we would retrieve and calculate the other information at this time. Without the other groups' information, we created dummy values that fell within expected ranges. These values are then created into a message to be sent to an online database setup with Amazon's Relational Database Service (RDS). There were many other options for databases such as Microsoft SQL Server 2017 SSEE and 000webhosting, but they had many drawbacks such as low storage space, unreliable servers, or possible loss of data. Amazon's RDS uses MySQL, a very easy to use engine, 20GB of storage, multiple connections and 24/7 service. The service is free for a year, giving plenty of time to prototype with. It is also to purchase the service if it fulfills our needs during prototype.

When constructing the database message, the current date and time is needed to correctly store and index the entry. This time is stored to keep the device's time synchronized to the real world time. Using this, we can check to see if the current time falls within the set dim setting interval. If it is, then we send a low signal through one of the controller's general-purpose input/output pins. If it doesn't, then a high signal will be sent.

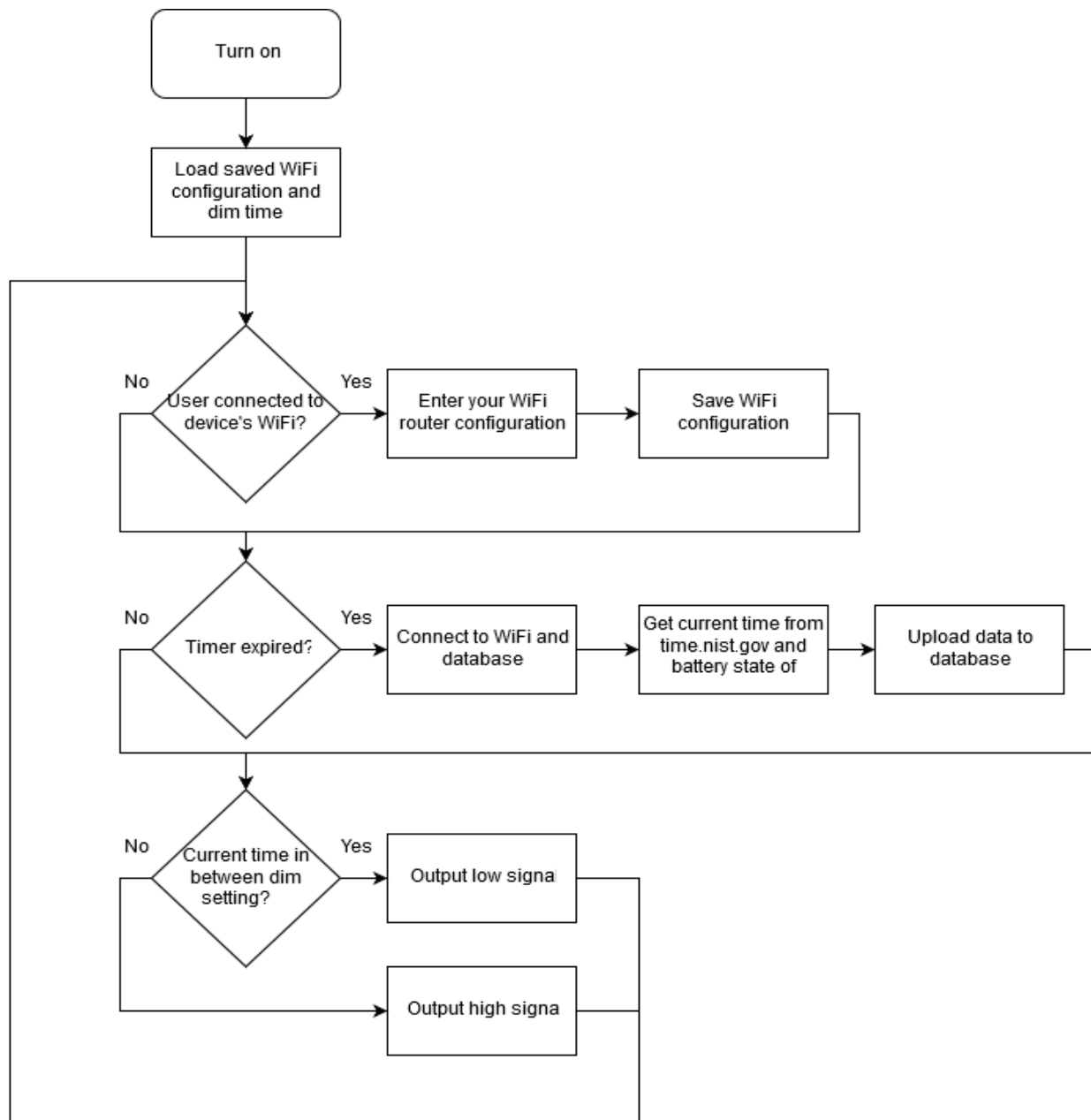


Figure 2.6 Microcontroller general control flow

### 2.3.2 WiFi-module

The WiFi chip receives signals sent from the smartphone app, connects to the user's WiFi router, and sends data to the database. The chip is in an access point and a station at the same time, allowing phones to connect to the device and for the device to send data to the

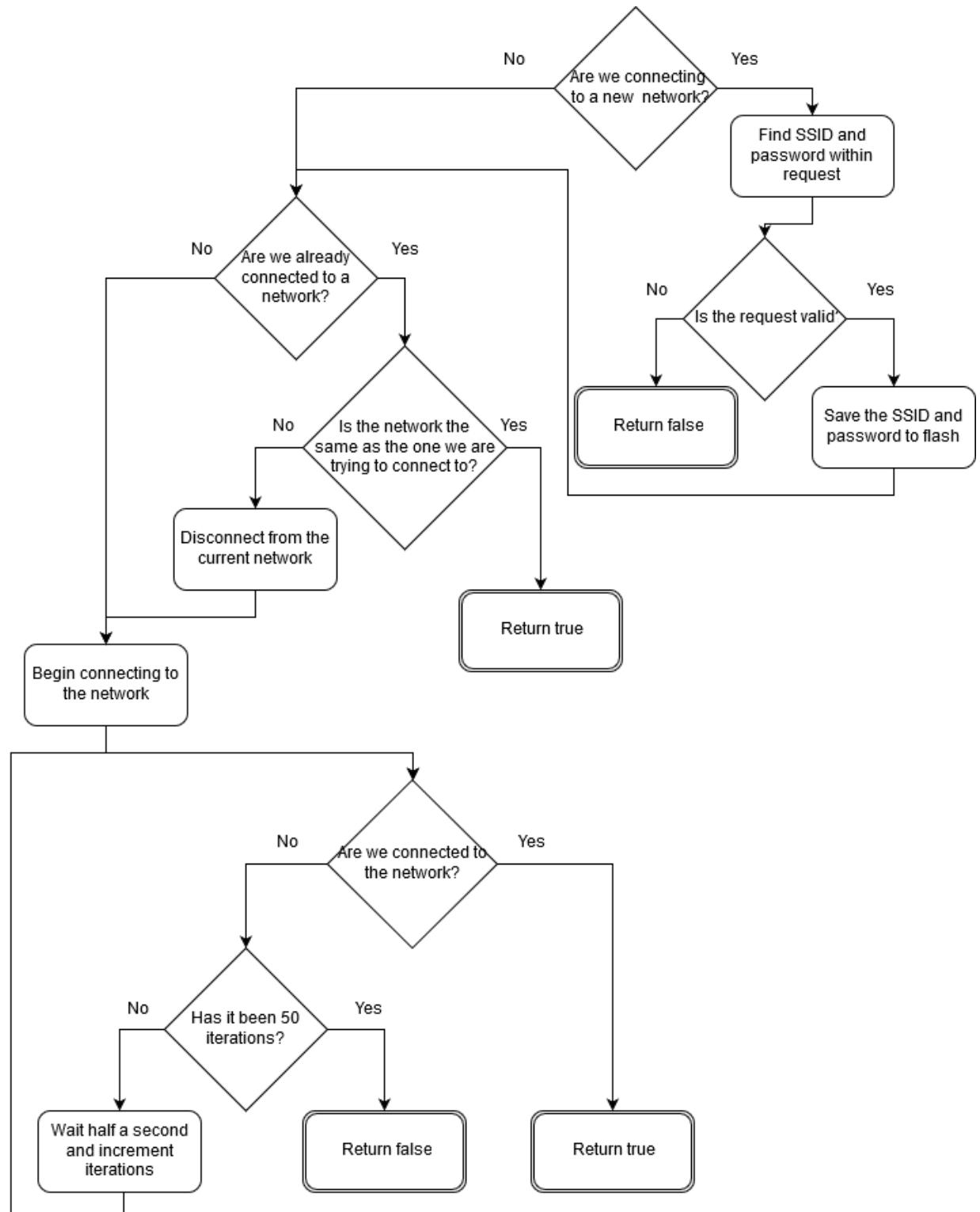


Figure 2.7 WiFi connection flow

Time	State of Charge	Motor Angle	Motor Angle 2	Charging Voltage	Charging Current	Solar Panel Power
2017-10-26 23:17:08	50	47	40	6	2	20
2017-10-26 23:17:23	50	29	27	6	2	6
2017-10-26 23:17:27	50	32	48	8	4	1
2017-10-26 23:17:35	50	69	89	2	4	9
2017-10-26 23:17:39	50	20	88	11	2	19
2017-10-26 23:17:57	50	62	63	2	2	14

Figure 2.8 Uploaded data in the database

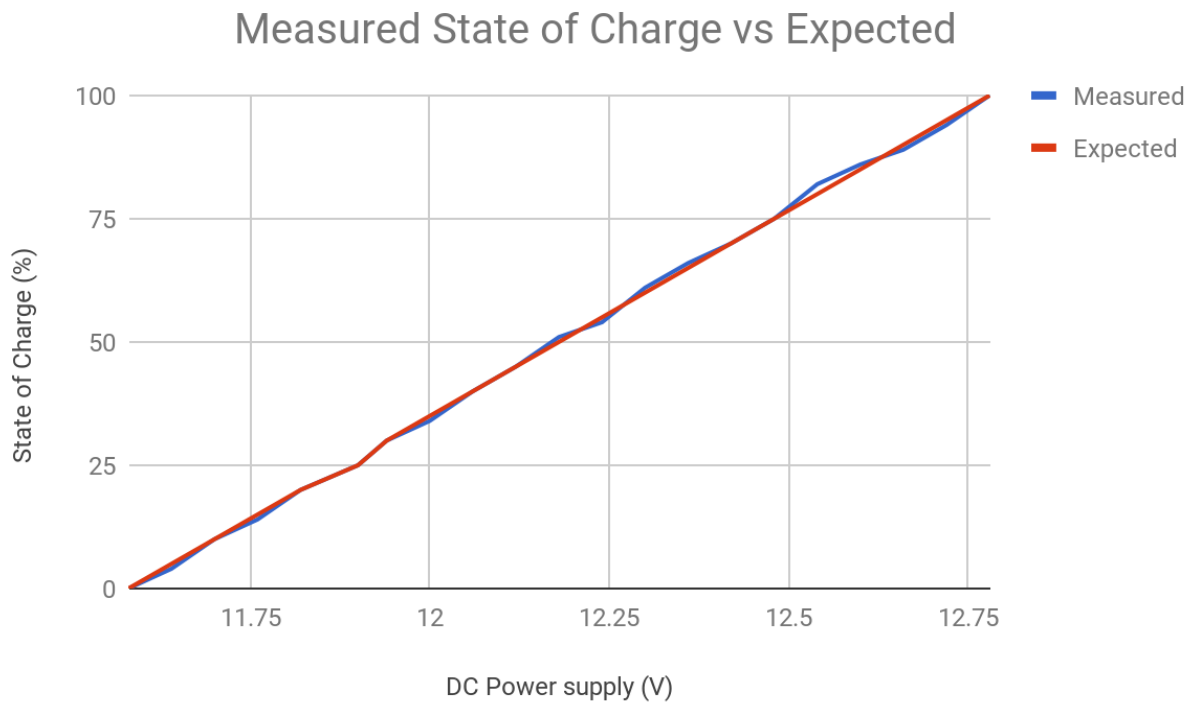


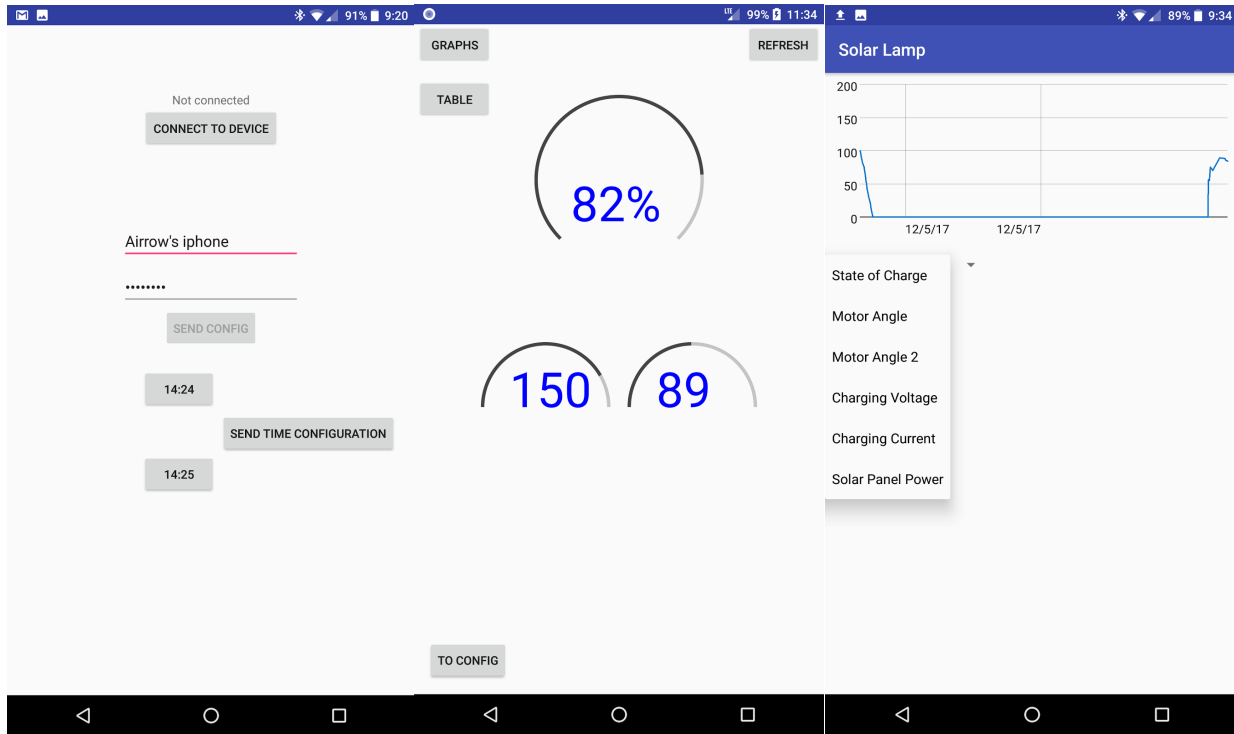
Figure 2.9 Fuel gauge test

router. When a signal is received, it is then sent to the microcontroller for processing. The general procedure for connecting to the router and database is outlined in Fig. 2.7. Figure 2.8 shows dummy data uploaded to the database.

We chose the ESP8266 NodeMcu because it is a cheap, simple, and efficient chip but still provides all of the functionality we need. It operates at 80 Mhz, which is more than enough for what we need, WiFi capabilities, 4 MB of flash, and only costs \$8.79. Another popular option is the Arduino WiFi shield, however this can cost up to three times as much and is a larger device.

### 2.3.3 App

For the smartphone app, we wanted to have an easy to use interface that allowed the user to send their router information and dim time settings, Fig. 2.10, show the current status of the battery, Fig. 2.11, and show trends in the data, Fig 2.12.



Figures 2.10, 2.11 and 2.12 showing the configuration, home screen, and data trends from left to right.

### 3. Design Verifications

#### 3.1 Fuel Gauge

For the fuel gauge, we wanted to minimize the inaccuracy to 1%. To achieve this, we used a DC power supply to output voltages that matched with the percentage marks in the battery's table [1]. Then the calculated SoC is uploaded to the database and shown on the phone to check for accuracy. Our test result is shown in Fig. 2.9, showing our fuel gauge is within the accuracy tolerance.

#### 3.2 Discharge Circuit

For the discharge circuit, we needed to have the LED turn on when the photo resistor detects no light and to be off when there is ambient light. To test this, we adjusted the resistors until the current produced when the photo resistor was covered producing a bright light and to have no light when the resistor is in light.

#### 3.3 WiFi Chip

For the WiFi chip, we wanted to be able to connect to the device from 50 meters away. While connected to the device, I sent a signal to switch the onboard LED on and off every meter. Once I reached 25 meters, I was unable to stay connected to the device. This doesn't reach the original 50 meters, however 25 meters is sufficient for users to reach the lamppost from their house.

*Table 4.1 Parts and costs*

Part	Cost
WiFi: HiLetgo ESP8266 NodeMCU CP2102      ESP-12E ( Amazon )	\$8.79
Battery : 12 V 44Ah SLA Rechargeable Battery by Interstate Batteries (Amazon)	\$180
Resistors	\$12
Photo resistor (Amazon)	\$10
Perfboard	\$7.99
<b>Total</b>	<b>\$218.78</b>

## 4. Costs

Our labor cost is \$30 per hour for two people for the duration of 12 weeks:  $2.5 * \$25 \text{ hour} * (128 \text{ hours} + 48 \text{ hours}) = \$11,000$  and the estimated parts costs is totaled to \$217.76 in Table 4.1. This totals \$11,218.78.

## 5 Conclusions

We were able to create a device that powers an LED based off of the ambient light and connect to a smartphone app and WiFi router. The device can also detect the battery's SoC and upload it to an online database to be later viewed by the user anywhere through the app to see trends and current data. We were also able to create a circuit that accepts input from the microcontroller to control the LED's luminance, however we were unable to integrate this into our final design.

### 5.1 Accomplishments

The final design is able to detect the ambient light and adjust the LED's luminance accordingly. It can also calculate the current battery's SoC utilizing the voltage regulator and equations found using the battery's manual [1] and testing. Our fuel gauge is very accurate and is able to let the user know exactly how much battery they have left and trends in its usage. The device is able to receive WiFi router configuration and dimming time interval from the phone app and save it to the flash memory, allowing it to reconnect and become operable after powering off and on. This is very important because then the device can still work if the battery runs out of charge without the user needing to reconfigure. The device is also able to connect to a WiFi router and create a connection with the online database to upload new data. The phone app has a home screen that shows the current statistics, allowing for quick status checking. It also has the ability to look at the table of data, allowing the user to look at specific entries. Users can also look at data trends through the graphs screen, selecting any of the stats to get a zoomable, scrollable interface to view a graph of the recorded data.

### 5.2 Challenges

One main challenge we faced was combining the luminance controlling circuit and the discharge circuit. We were unable to design and retrieve parts for connecting the two circuits. We choose to

include the discharge circuit instead of the luminance controlling in the final product because being able to detect when the light needs to be on makes the lamp more autonomous and energy efficient. This conforms more to the overall goal of the lamp.

Another challenge was getting the full 50 meters range; however the 25 meters we achieved would be sufficient for normal usage. A final challenge is being able to keep the fuel gauge accuracy under varying conditions. The temperature the battery is in can affect the capacity retention, affecting the equation used to calculate the SoC as well as natural degradation of the battery changing its output. We would need to include a temperature sensor to detect the current temperature and adjust the equation to take this and how long the battery has been operable into account.

### **5.3 Future Work**

For the future, we would need to combine our luminance controlling circuit and the discharging circuit to have all of the functionality. We also need to add an antenna to increase the range if we found 25 meters isn't enough. Finally, we need to work with the two other groups to retrieve data from their circuits to show correct data instead of dummy values. After this, we would have a finished product.

### **5.4 Ethical Considerations**

Consulting the IEEE Code of Ethics [2], we found our project, has few ethical concerns. One is disclosing factors that might endanger the public or the environment. We are working with a lead acid battery, which introduces a potential danger of the battery exploding or expelling acid. For this reason, we will show warnings for any danger and accept responsibility for our choices. We also have to keep operational temperatures in mind. For lead acid batteries, this is between -20 to 50 degrees Celsius for charging and discharging [3] so these warnings would need to be displayed as well. Due to lead acid batteries being volatile and very dangerous, we utilized the material data safety sheet [4] at all times in lab and keep it for future work.

Another concern is that we will be lighting someone's driveway. If we are unable to provide enough light or if the light is inconsistent, we might cause accidents due to poor lighting. For this reason we will not over promise on the output of our lamp and to make our circuit be efficient to make the LED stay on for as long as possible.

## 6. References

- [1] *TECHNICAL MANUAL For Sun Xtender<sup>®</sup> Batteries*. (2017). 6th ed. [ebook] West Covina: Concorde Battery Corporation, p.35. Available at:  
[http://www.sunxtender.com/pdfs/Sun\\_Xtender\\_Battery\\_Technical\\_Manual.pdf](http://www.sunxtender.com/pdfs/Sun_Xtender_Battery_Technical_Manual.pdf) [Accessed 30 Nov. 2017].
  
- [2] *ieee.org, "IEEE IEEE Code of Ethics"*, (2017). [Online]. Available:  
<http://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed: 19- Sept- 2017].
  
- [3] *"Charging at High and Low Temperatures"*, (2017) [Online]. Available:  
[http://batteryuniversity.com/learn/article/charging\\_at\\_high\\_and\\_low\\_temperatures](http://batteryuniversity.com/learn/article/charging_at_high_and_low_temperatures). [Accessed: 19 - Sept - 2017].
  
- [4] *Safe Practice for Lead Acid and Lithium Batteries*. (2016). [ebook] Champaign, pp.1-2. Available at:  
<https://courses.engr.illinois.edu/ece445/documents/GeneralBatterySafety.pdf> [Accessed 2 Oct. 2017].



## Appendix A Requirement and Verification Table

*Table A Requirements and Verifications*

Requirements	Verifications	Verification Status (Y or N)
<ol style="list-style-type: none"> <li>1. Minimize the inaccuracy to 1%</li> <li>2. Should accept an input Voltage of 5V</li> <li>3. Send battery capacity and other information status to the microprocessor through I2C or 1-Wire.</li> </ol>	<p>Obtain the I-V curve under different circumstances. Charge and discharge the battery for one cycle in the lab room, record the time taken, output voltage range, and take the average of the output and consider it as the standard voltage. Then calculate the inaccuracy.</p> <p>Finally, we minimize the inaccuracy by manipulating the circumstance of the battery box, for example, adjusting temperature, correcting coefficient of cycles charged.</p>	Y
<ol style="list-style-type: none"> <li>4. Discharge when light density is low, which is when light intensity is not bigger than 35 LX.</li> </ol>	<p>By putting the photo resistor as closest to the ground as we can, we measure the light intensity and set the photo resistor parameters and make it turn on when the light intensity is low.</p>	Y
<ol style="list-style-type: none"> <li>5. Change the luminance of the LED by changing the current going through the LED.</li> <li>6. Luminance weaker after midnight</li> </ol>	<p>Obtain the voltage range that can be applied across the LED. Connect a resistor and a switch in parallel then connect these circuit in series with the LED. The switch is connected to the microcontroller and will bypass the resistor, which will make the light intensity bigger before midnight. After midnight, the switch will turn off, and there are just a resistor and a LED in series, which will make the luminance weaker.</p>	N
<ol style="list-style-type: none"> <li>7. Send/Receive messages up to 50 +/- 5 meters</li> </ol>	<p>Measure 50 meters with a tape measure, then send a message to get battery charge status and see if a message is sent back with the data.</p>	N

### Figure B Overall circuit schematic

