Dissertation on

"Smart Lamp Post Platform Development"

Submitted by

DEEPAK M K

(PES1201702295)

August. 2018 - April. 2019

Internal Guide's

Dr. Venkatarangan M.J, Mrs. Susmita Deb

Associate Professor's

Department of Electrical and Electronics Engineering

PES University

Bengaluru -560085

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

CERTIFICATE

This is to certify that the Dissertation entitled

"Smart Lamp Post Platform Development"

Is a bonafide work carried out by

DEEPAK M K

(PES1201702295)

In fulfilment for the completion of final year course work in the Program of Study **M.Tech (EMBEDDED SYSTEMS) in Electrical and Electronics Engineering** under rules and regulations of PES University, Bengaluru during the period **August. 2018 – April. 2019**. It is certified that all correction/suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the academic requirements in respect of project work.

M.S.V. R.

Signature with date
Internal guide

22/07/19

Signature with date
Internal guide

23/7/19

Signature with date & Seal
Chairperson of EEE

Dr. B.K. KESHAVAN
Chairperson
Dept. of Electrical & Electronics
PES UNIVERSITY
100 Feet Ring Road, BSK 3rd Stage
Bangalore - 560 085

Signature with date & Seal
Dean –Faculty & Technology

Dean
Faculty of Engineering Technology
PES University
100 Feet Ring Road, BSK III Stage
Bengaluru-560085

External Examiner
Name of Examiner:

Signature with date

1.

2.

3.

DECLARATION

I, **DEEPAK M K** hereby declare that the dissertation entitled, **"Smart Lamp Post Platform Development"**, is an original work done by me under the guidance of **Dr. Venkatarangan M.J.**, Associate Professor and Co-guide **Mrs. Susmita Deb**, Associate Professor in **EEE Department**, and is being submitted in fulfilment of the requirements for completion of final year course work in the Program of Study **M.Tech (Embedded Systems) in Electrical and Electronics Engineering**.

Deepak. M.K

Name of the student with Signature

PLACE: BENGALURU

Deepak M K (PES1201702295)

Date: 22/07/2019

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere thanks to the people without whom this Project would not have been possible and whose firm belief in my capabilities and morale boosting support to bring out full potential towards the forefront has played a major role in each and every step of my endeavour to success.

I express my sincere gratitude to **Dr. M. R. Doreswamy**, Chancellor of PES University, **Prof. Dr. Jawahar** Pro Chancellor of PES University, **Dr. K.N.B Murthy**, Vice-Chancellor**, Dr. K.S. Sridhar** Principal, PESIT Bengaluru and **Dr. V. Krishnamurthy** Registrar of PES University for providing me a very good opportunity to study in your institution.

I express my sincere gratitude to **Dr. B K Keshavan**, Chairperson of Electrical and Electronics Engineering Department, PES University Bengaluru for providing me with adequate facilities, ways and means by which I am able to complete this project.

I deeply indebted to my project guide’s **Dr. Venkatarangan M.J** and **Mrs. Susmita Deb** Associate Professor’s in Electrical and Electronics Engineering Department for providing me with valuable advice and guidance during the course of study.

I would like to extend my gratitude to the **teaching and non-teaching staff members** of Electrical and Electronics Engineering Department for their support and cooperation during the course of study.

Finally, I would like to extend my gratitude to the **library staff members** and **all my friends**, who have directly or indirectly supported me during my project work.

With regards

Deepak M K (PES1201702295)

Abstract

As the year passes the technologies are improving at an increased pace and has played a prominent role towards smart cities. The smart cities make use of collecting data and manage the resources efficiently using the data. Nevertheless to state that the cities will have many installations of street lights and they consume enormous electric energy. The project is an attempt to work on solutions around street lights to reduce the electric energy consumption and to make use of technologies to collect relevant data towards managing the resources. It is very obvious to go for renewable solution and depend on solar energy that is available in abundance. Thus the project is to advance the existing street lamps to include more sensors and controls and make it a smart lamp post. The project work is an experimental step towards the deployment in PES University. The solution comprised of solar based charging using solar controller, LED light output for street lighting and communication network currently based on GSM.

The following features are incorporated on smart lamp post:

i. System running solely on solar energy
ii. Automatic light control (PWM control) for day and night.
iii. Real time data of temperature, humidity and various other sensors which are available can be uploaded to the server that can help data analytics and any further actions based on it.
iv. Humidity and temperature is displayed on Red LED display.

The solution is incorporated using Arduino as microcontroller, SIM800C as GSM module to send the data to ThingSpeak server. The data from various sensors are uploaded to server once in every 15 minutes. The theoretical postulation of power consumption for a day was calculated and compared with the actual measurements in the system. The uploaded data was analysed for the charging and discharging of battery. In this project we have tested with two batteries by initially charging it. Battery-1 discharged 535Wh while running the entire setup whole night and Battery-2 discharged 441Wh whole night. The location where solar panels where placed did not get enough light during first half of the day and hence was not good enough to self-sustain the system. It is to be experimented at a different location. However the experimental setup is a lead towards the deployment in the campus.

Table of Contents

List of Figures

List of Tables

Abbreviations

1. **GSM- Global System for Mobile communications**
2. **TCP- Transfer control protocol**
3. **SPI-** ***Serial Peripheral Interface***
4. **RTC- Real time clock**
5. **AT-Attention command**

Chapter 1: Introduction

1.1 Motivation

The traditional lamp post was only to illuminate light during night time using different lamps like sodium vapour lamp, incandescent lamp etc. With the advent of new technologies and with countries moving towards smarter cities, the lamp posts trends state that they become smarter and a hub for monitoring different parameters in the city. With countries like India that has abundant sunlight it is quite natural to make best use of solar energy…. .

Since lamp post are usually tall, we make use of the space available and upgrade with IoT devices to make it into a Smart lamppost. With the IoT integrated with the lamppost we can achieve some of the following things:

1) **Smart energy:**
   - **i.** LED lights are more efficient and consume less energy and save money.
   - **ii.** Light output management can be brought out based on solar and switching on and off only during day/night.
   - **iii.** The energy which used up by LED can be further reduced by dimming the LED after few hours of Switching on.
2) **Data collection and analytics:** Large amounts of data are collected and they can be analysed to take meaningful actions for an efficient use of resources. The data is also made available to public and can be used for data analytics.

1.2 Outcomes and Objectives

The aim of the project is to develop an initial platform and demonstrate the use cases for smart lamp:

1) LED Light output management – switching on and off according to day/night.

2) Measurement of parameters – humidity, temperature

3) Monitoring solar and battery voltage and current

4) Upload of the data to Cloud via GSM for analysis purposes.

Following are the Learning Objectives:

1. Learning Embedded Programming for Arduino
2. Understand and develop interfaces for GSM.

3. Test all Software and Hardware like GSM, Voltage and Current Sensor.
4. Creating data acquisition system with the battery and Solar panels
5. Interfacing Display.
6. PWM Control of light based on voltage.
7. Integrating all the components.
8. Testing of Lamp post in real time with the battery and solar panel connected.
9. Analyse data available
10. Sending the real time data to the server where it can be remotely accessed and analysed.

1.3 Scope of work

As the smart lamp post can be designed with multiple features, the scope of the project was limited to the following:

Sensors (Inputs): humidity, temperature, voltage and current sensors.

Display: RED Display, LED output

Connectivity: GSM

Light controls based on movement detection are not yet included as the detection has to happen on a separate node and not on lamp post.

Chapter 2: Literature Survey

2.1 Literature review

[1] “S. Siregar and D. Soegiarto, Solar panel and battery street light monitoring system using GSM wireless communication system, in *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, 28-30 May 2014 2014, pp. 272-275, doi: 10.1109/ICoICT.2014.6914078”

This paper talks about monitoring of solar and battery using various sensors like volatge and current sensors which we have included in our project and made a data aqusition system and remotely monitoring the data.

[2] “N. Yoshiura, Y. Fujii, and N. Ohta, Smart street light system looking like usual street lights based on sensor networks, presented at the 2013 13th International Symposium on Communications and Information Technologies (ISCIT), 2013.”

This paper talks about smart lighting system, whenever a person is passed by lamp the light is turned on or off based on short ranged motion detection sensors. We haven’t yet used motion detection sensor but automatic turn on or off is achieved. The idea is still considered.

[3] “M. Revathy, S. Ramya, R. Sathiyavathi, B. Bharathi, and V. M. Anu, Automation of street light for smart city, in *2017 International Conference on Communication and Signal Processing (ICCSP)*, 6-8 April 2017 2017, pp. 0918-0922, doi: 10.1109/ICCSP.2017.8286503”

This paper talks about light energy management of a system using LDR, and whenver fault has occurred with in the system, the information is communicated to the main station via GSM. We have taken communication via GSM in our project. Similar to LDR concept we have used voltage sensor to measure volatge and as soon as it goes below certain voltage level light is going to turn on.

2.2 Survey of Existing System

The conventional sources of light are still existing such as sodium vapour lamp, incandescent lamp with are not efficient for economic growth.

Some of the conventional lamps have being replaced with LED lamps. Non-Conventional sources light are such as solar based lamps are now every popular shown in **Figure 2.1** .Wind based lamppost are setup in Urmia, Iran shown in **Figure 2.2**.In Asia trials are going on by installing camera on the lamppost to monitor people, cyclist and vehicle as shown in **Figure 2.3**.

Now in Kolkata they upgraded lamppost with some of the sensors like rain sensor, light sensor and are remotely monitored as shown in **Figure 2.4** .

Europe was the first country to start lamppost which detect motion of object or people based on that the lamp would turn on or off as shown in **Figure 2.5** . It’s so popular that the research based on motion detection is still being continued all over world which would help save energy as well as develop one’s country

Figure 2.1: Solar based lamppost

Figure 2.2: Wind based lamppost

Figure 2.3: Camera based lamppost

Figure 2.4: Sensors installed in lamppost, Kolkata

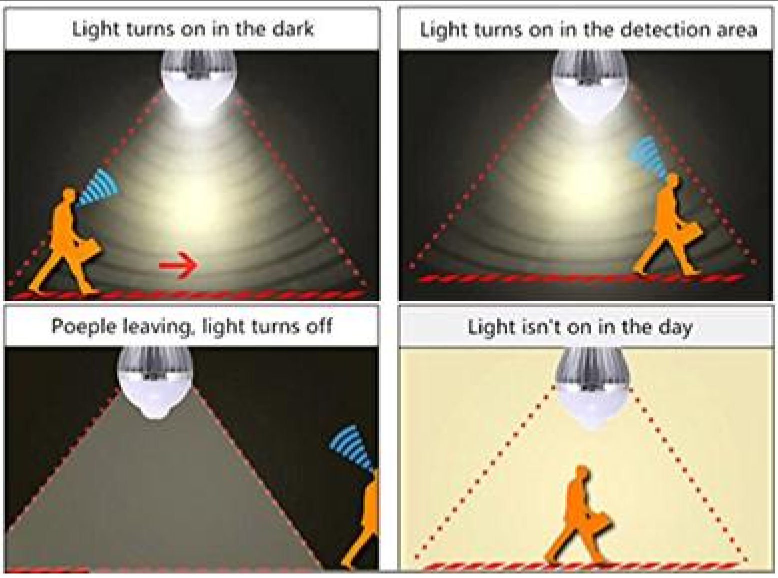

Figure 2.5: Motion Detection in lamppost

2.2.1 Take away from the above system to our project

1. Solar and wind energy can be installed together to the lamppost to use them as renewable energy.
2. Use of LED Lamp is much better that sodium vapour lamp because of energy savings [4].
3. Including rain sensor and various other pollution detection sensor enables users to know the environmental parameters in the area.
4. Remotely controlling all lights.
5. Installing camera for safety [5]and be used for traffic control [6].
6. Motion detection helps in more conservation of energy s compared to the traditional methods.

Chapter 3: Hardware Architecture

3.1 Smart Lamppost Electrical Architecture

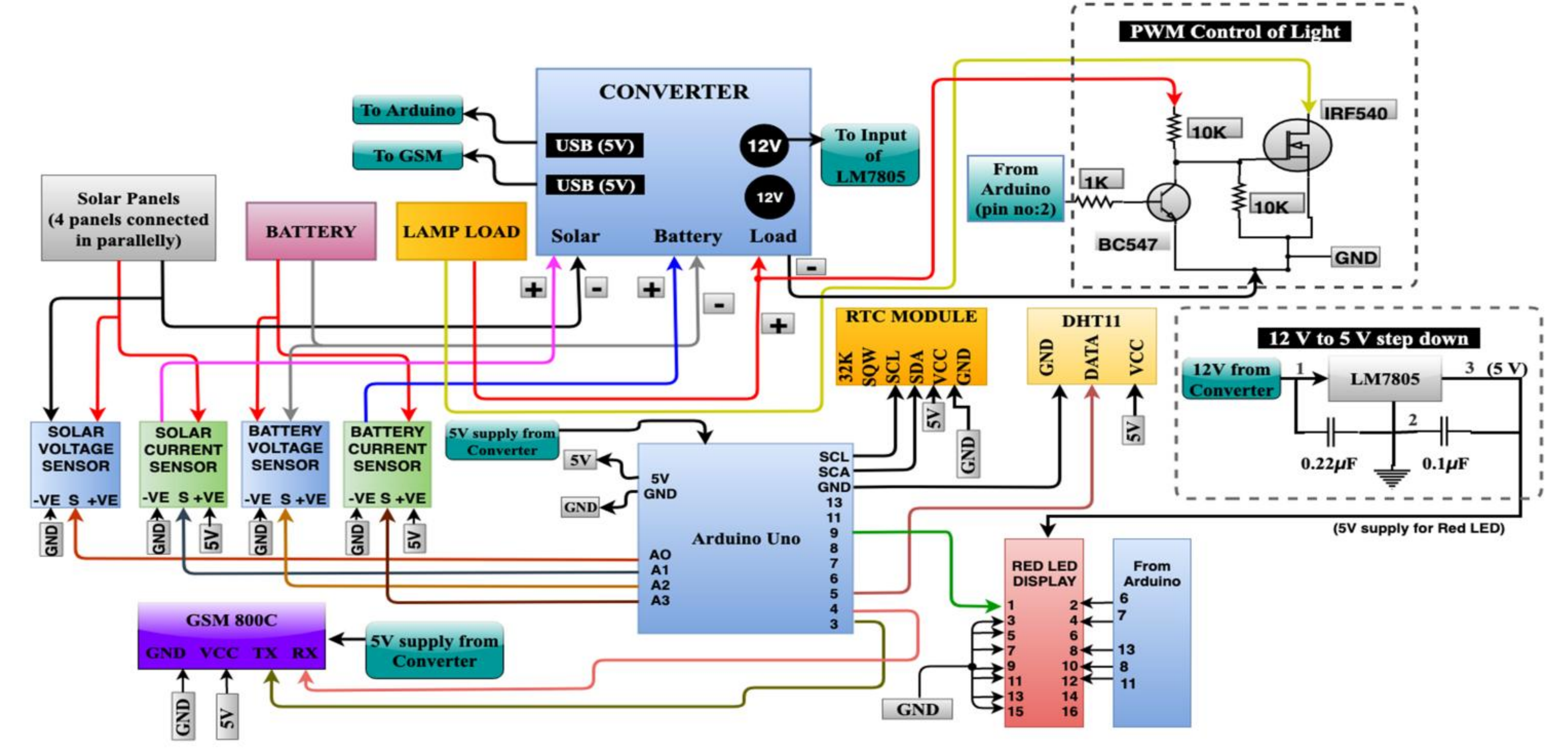

Figure 3.1: Electrical architecture of smart lamppost

3.1.1 Detail Description of overall hardware connection

Figure 3.2: Solar panels kept in balcony of EEE dept.

In our project we are using 4 panels each rating of 50W (12V).

All the panels are connected in parallel to increase the required current of 5A to charge the battery this is because all the panels are placed in balcony of EEE department, PES University as shown in **Figure 3.2**. Where sunlight falls only after 1:30 or 2pm onwards. Voltage of 12V will be available from morning 6am but current will be low because of less irradiation falling on the panels and starts to increase in afternoon.

Figure 3.3: Battery 12V, 150Ah

The battery shown in **Figure 3.3** is 150Ah (12V)

The power stored in the battery is

P=12(V) *150(Ah) =1800Wh

1Ah means it is capable of delivering 1A of current in One hour.

Figure 3.4: 30 watt lamp used in smart lamppost

The **Figure 3.4** shows the 30watt (12V, 2.5A) lamp load which is used for Smart lamp post [7, 8].

Originally this lamp had internal converter doing automatic charging of battery by taking the power from solar panels. Automatic turn ON or OFF of the light based on voltage levels like below 4.5V it use to turn ON the light (**Manually Tested**). Now we have taken over that control from the converter and given it to the Arduino by using PWM control.

Figure 3.5: Two P10 Red LED Display connected together

The **Figure 3.5** shows Two P10 Red LED joined together and displaying the real time data obtained from Arduino Uno. We have used 3m long Flat cable to connect Arduino and right side of the display. Arduino uses SPI protocol to transmit the data to the red display.

Figure 3.6: Solar charge controller

The **Figure 3.6** shows the solar charge controller at the bottom we see solar, battery and lamp load connections are given.

This controller provides Two USD ports which gives 5V supply which is given it the Arduino and GSM

Two 12V jacks are given where we have taken one of this input and given it to stepdown circuit to give 5V, which is used to power the Red LED Display.

The **main disadvantage** of this controller is that when it is switched ON it is going to turn ON the lamp load it doesn't have the automatic turning ON or OFF of the Lamp Load thus we have given the control of light the Arduino.

3.2 Microcontroller and sensors used

We are using Arduino as the controller right now, GSM is used for communication. Temperature, Humidity, current and voltage sensors are for measurement. We have created a Real time Data Acquisition system between Solar and battery.

3.2.1 Arduino Uno

It is also one most popular boards available in the market. In this we use C++ programming for coding. It requires 12V supply to the jack. We are going to utilize all the pins for different sensor since Arduino has more number of pins are available compared raspberry pi.

All the sensors connected to Arduino we use ADC or DAC because it only provide only 5V and supply more than 5V to Arduino will damage the board.

The real time data which available is sent to the server via GSM module. Some the sensor’s data is also sent to the raspberry pi to display or displayed on P10 Red LED display.

3.2.2 DHT11 sensor

Figure 3.7: DHT11 sensor

DHT11 are compact, low in price, consumes low power which shows accurate temperature and humidity value

3.2.3 Current sensor (ACS712-5A)

Figure 3.8: Current sensor 5A

ACS712 are most affordable current which are available in market, we get 5A, 20A, 30 A in range. Technically speaking this sensor measures voltage and it is divided by sensitivity value to get current value. Arduino uses analog to digital conversion otherwise board will get damaged

Current = (Voltage)/ (sensitivity)

Sensitivity value of this sensor varies from 180mm to 190mm but nominal sensitivity is 185mm.

3.2.4 Voltage sensor

Voltage sensor are most affordable module which are available in market. Technically speaking this sensor uses voltage divider circuit and whatever the input voltage is supplied to module is reduced to the factor by 5. Suppose 25 volt is fed to the module and it is reduce by a factor 5 meaning we are going to get 5volt as the output which helps all the development to read otherwise board will get damaged because of over-voltage.

Figure 3.9: Voltage sensor

3.2.5 RTC (real time clock)

Figure 3.10: RTC module

Arduino doesn’t have RTC in built that is why use RTC module to keep track of time.

Specifications of RTC module

1. Supply 5V DC.
2. Keep Time track with the help of coin cell.
3. I2C Serial Interface.
4. “Consumes Less than 500nA in Battery-Backup”.
5. “Automatic Power-Fail Detect and Switch Circuitry”.

3.2.6 GSM (SIM800C)

Figure 3.11: GSM SIM800C

“GSM stands for Global System for Mobile communications” for cellular communications. It use 5 volt, 1.35 amps supply.

In this project we are making use of AT commands to make communication with the GSM [9, 10]. We use some of the feature for sending, receiving messages and sending the data to the server.

Some of the tests are done on GSM using AT commands with the help of TERMITE software. We have used TTL to USB converter to establish communication between these two Systems.

3.3 Theoretical postulation of power consumption in a day

The below **Table 3.1** gives the postulation of power consumption in a day. Based on the calculations and other rationale the solution needed the following for 30W LED light output with system up and running all the time:

1) 4 panels of 12V (50W) to meet the required power consumption in a day.
2) Battery capacity of about 45Ah is what is needed for system to be up and running for full day. The decision was to use 150Ah Lead Acid battery as it was already available.

Table 3.1: Theoretical power consumption in a day

Sl no	Components	V	I (A)	Power (W)	Time (hours)	Power Wh	Remarks
1	LED light	12	2.5	30	12	360	Considering 1 LED for now and smart power management
2	Arduino	5	0.05	0.25	24	6	Scope to reduce by Software Power management
3	GSM SIM Card	5	0.13	0.65	22	14.3	Idle current for GSM
		5	0.416	2.08	2	4.16	Considering current at data transmission
4	LED Display	5	0.3	1.5	24	36	Presuming this to be used for displaying humidity, temp, etc. at different times
					Total	**420.46**	**Watts for 12 hours**
B	**Needed Battery capacity with 12V**					35.04	**Ah**
	Additional losses not being full efficient		30%			**45.552**	Ah

Chapter 4: Software Architecture

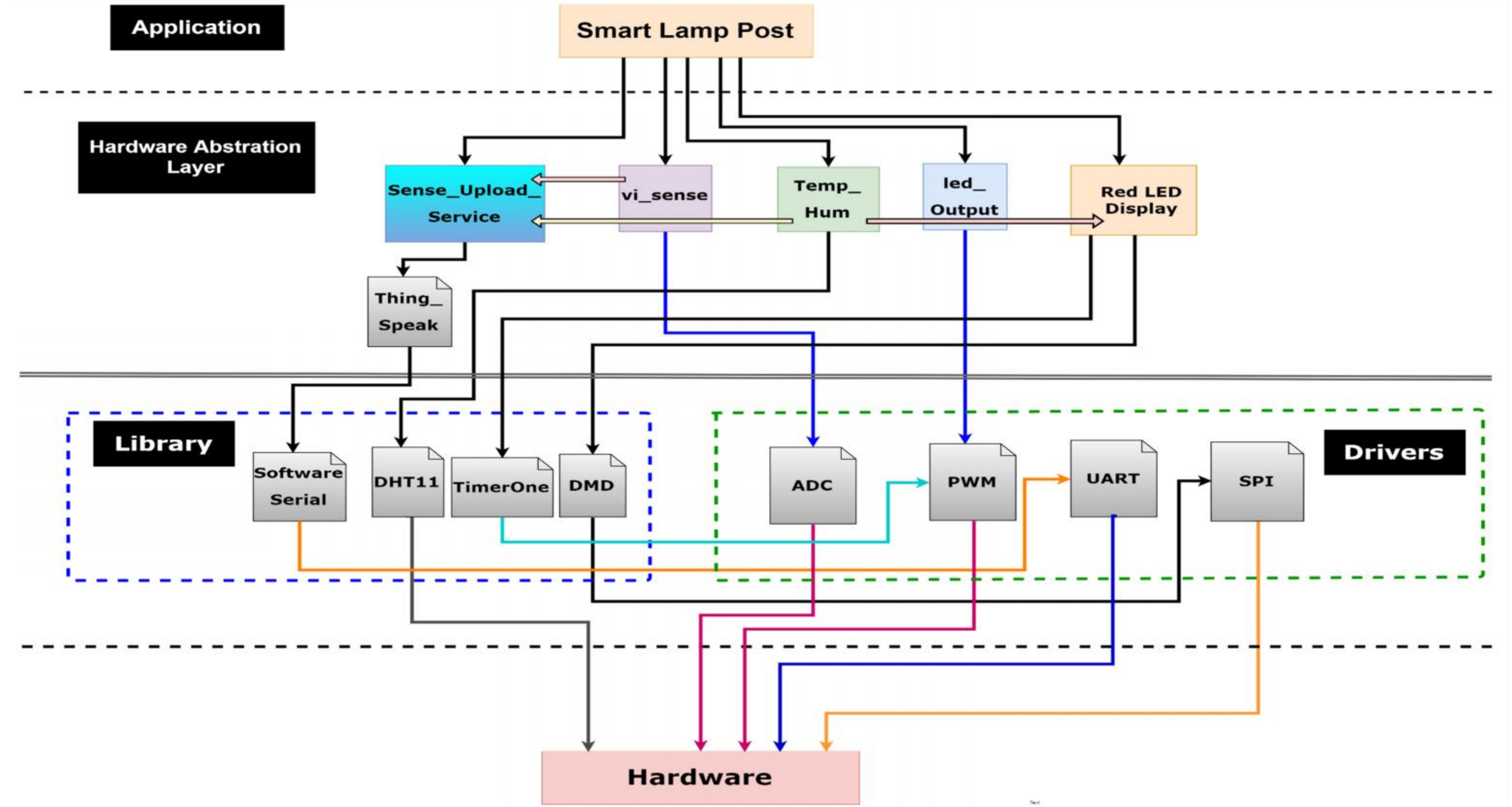

Figure 4.1: Software Architectural Diagram

In the **Figure 4.1** Sense_Upload_service is taking data from vi_sense and Tem_hum files. Temperature and humidity data are taken and displayed on red display.

4.1 Red LED Display

It uses SPI Protocol for communication

4.1.1 red_display_setup ()

Function name	red_display_setup
Parameters	None
Return	None
Description	This function sets up the needed hardware setup for making use of RED display

4.1.2 red_display_loop ()

Function name	red_display_loop
Parameters	None
Return	None
Description	This function will invoke the calls to get the humidity and temperature values using *temp_hum* module and passes the values to RED display using library calls for *DMD.*
External modules	temp_hum Library: DMD, Timer One

4.2 led_output

4.2.1 led_setup ()

Function name	led_setup ()
Parameters	None
Return	None
Description	This function sets up the needed hardware pin for controlling LED Light. The pin will be configured as output and to be used for PWM output. Note: As there is an issue in using PWM output and Red Display (SPI interface) simultaneously, it makes use of digital output for now.
External modules	Library: Arduino (PWM)

4.2.2 led_toggle ()

Function name	led_toggle ()
Parameters	None
Return	None
Description	This function will toggle the LED light output. If it is on, it will be switched off and if off, it will be switched on.

External modules	Library: Arduino (PWM)

4.2.3 led_turn_off ()

Function name	led_turn_off ()
Parameters	None
Return	None
Description	This function will switch off the light
External modules	Library: Arduino (PWM)

4.2.4 led_turn_on ()

Function name	led_turn_on ()
Parameters	None
Return	None
Description	This function will switch on the light
External modules	Library: Arduino (PWM)

4.3 temp_hum (Temperture and Humidity)

4.3.1 get_temperature ()

Function name	get_temperture ()
Parameters	None
Return	Float: Returns the temperature value
Description	This function will make use of *DHT* library and returns the temperature values as receied from DHT11.
External modules	Library: DHT

4.3.2 get_humidity ()

Function name	get_humidity ()
Parameters	None
Return	Float: Returns the humidity value
Description	This function will make us of *DHT* library and gives the humidity values.
External modules	Library: DHT

4.4 vi_sense

4.4.1 Battery_Ampere ()

Function name	Battery_Ampere ()
Parameters	None
Return	Float: Returns the current drained from or charged to battery. -ve: Charging +ve: Discharging
Description	This function will give us battery current making use of ADC.
External modules	Drivers: ADC

4.4.2 Battery_Voltage ()

Function name	Battery_Voltage()
Parameters	None
Return	Float: Returns the voltage from battery
Description	This function will give us battery voltage making use of ADC
External modules	Driver: ADC

4.4.3 Solar_Ampere ()

Function name	Solar_Ampere ()
Parameters	None
Return	Float: Returns the current from solar panels +ve: charging the battery during day Zero: during night
Description	This function will gives us solar current making use of ADC.
External modules	Driver: ADC

4.4.4 Solar_Voltage ()

Function name	red_display_loop
Parameters	None
Return	Float: Returns the voltage from solar panels
Description	This function will gives us solar voltage making use of ADC.
External modules	Drivers: ADC

4.5 Sense_upload_service

It uses UART for communication

4.5.1 Sense_upload_setup ()

Function name	Sense_upload_setup ()
Parameters	None
Return	None
Description	This function will invoke the calls to take care of hardware and software setup for communication with GSM, sets the private key for communication with Thingspeak server thingsSpeakGsmSetup (); thingsSpeakSetPrivateSendKey (“xxxxx”); and setup necessary software requirement.

4.5.2 Sense_upload_loop ()

Function name	Sense_upload_loop ()
Parameters	None
Return	None
Description	This function will invoke the calls to get the humidity, temperature, Battery voltage, Battery Ampere, Solar Voltage and Solar Ampere values using *temp_hum* and *vi_sense* modules and uploads the parameters to ThingsSpeak server.
External modules	temp_hum vi_sense thingSpeak

4.5.3 thingSpeak ()

4.5.3.1 thingspeakGSMsetup ()

It uses Software serial to communicate with GSM and arduino

Function name	thingSpeakGSMsetup ()
Parameters	None
Return	None
Description	This function will setup necessary hardware requirement for GSM (SIM800C) module using UART communication via Software serial

4.5.3.2 thingsSpeakSetPrivateSendKey (String privateKey_in)

Function name	thingsSpeakSetPrivateSendKey (String privateKey_in)
Parameters	privateKey_in
Return	String: Returns the API key from obtained from ThingSpeak server in string format.
Description	This function will set private key for thingSpeak web site when function is called. The private key is a key that can be found on ThingSpeak web site corresponding Write API.

4.5.3.3 thingsSpeakSend(float*parFields_in, unsigned char noOfParameters_in)

Function name	thingSpeakSend (float* parFields_in, unsigned char noOfParameters_in)
Parameters	parFields_in,: This is a pointer to the storage list of parameters noOfParameters_in: This signifies the number of parameters passed to the function.
Return	Float: Returns all the parameters value in float unsigned char: Returns number of characters used in “noOfParameters_in”
Description	This function will upload the parameters to ThingsSpeak server.

4.5.3.4 Send2thingspeak (String parUploadCmdString_in)

Function name	Send2thingspeak(String parUploadCmdString_in)
Parameters	parUploadCmdString_in: respective field numbers and data are kept in string format
Return	String: Returns in String format for the data that needs to be sent to server.
Description	This function will use the At commands, establishes communication and sends data to thingSpeak server using TCP.

Chapter 5: Results

5.1 Harware connections

The below **Figure 5.1** is the overall hardware connections setup in one single board. The Solar panels, Battery, Lamp Load (30) and P10 Red LED Connections are also included in this board.

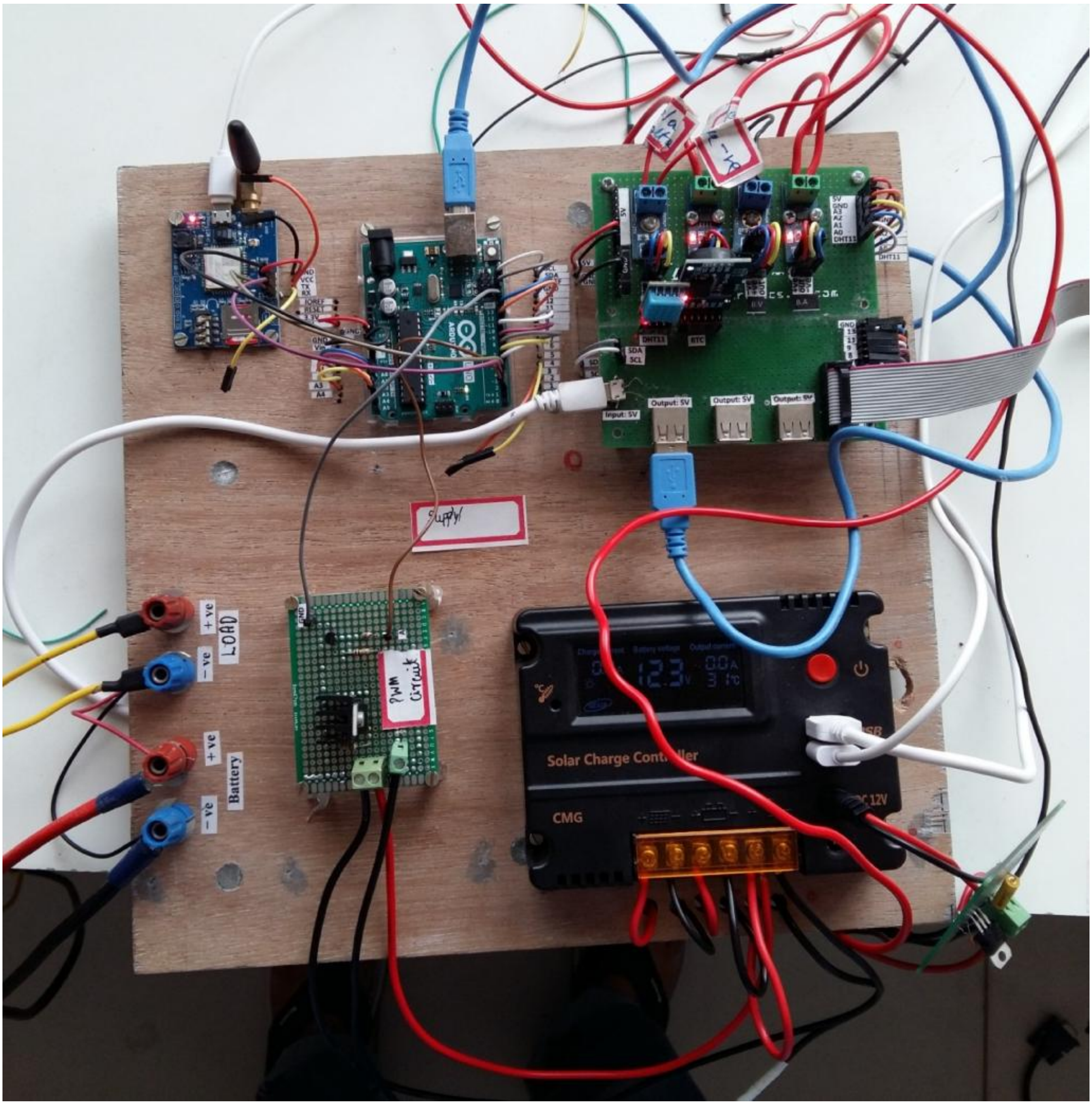

Figure 5.1: Overall Hardware Connections

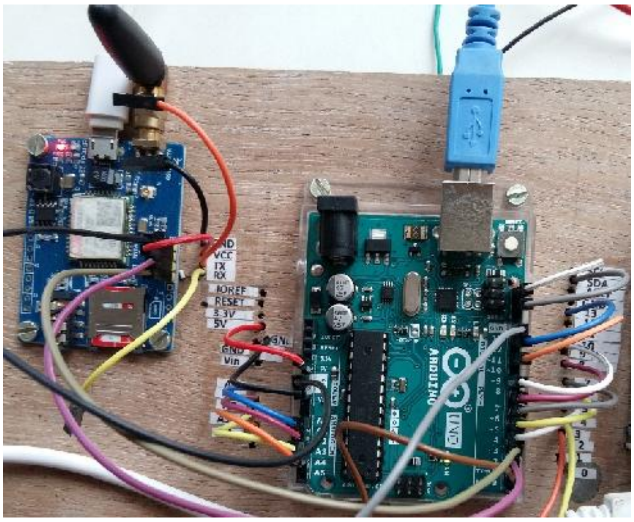

Figure 5.2: GSM and Arduino Connected to board

The **Figure 5.2** shows GSM and Arduino Uno.

The power for GSM and Arduino are given from the converter each of them take 5V.

The Transmit and Receive pins of GSM are connected to Receive (pin no 3) and Transmit (pin no 4) of Arduino respectively.

Various sensors like voltage and current sensors of both Solar and Battery, DHT11 sensor, P10 red LED display, RTC module and PWM connections are given to Arduino. It acts as the controller for entire circuit.

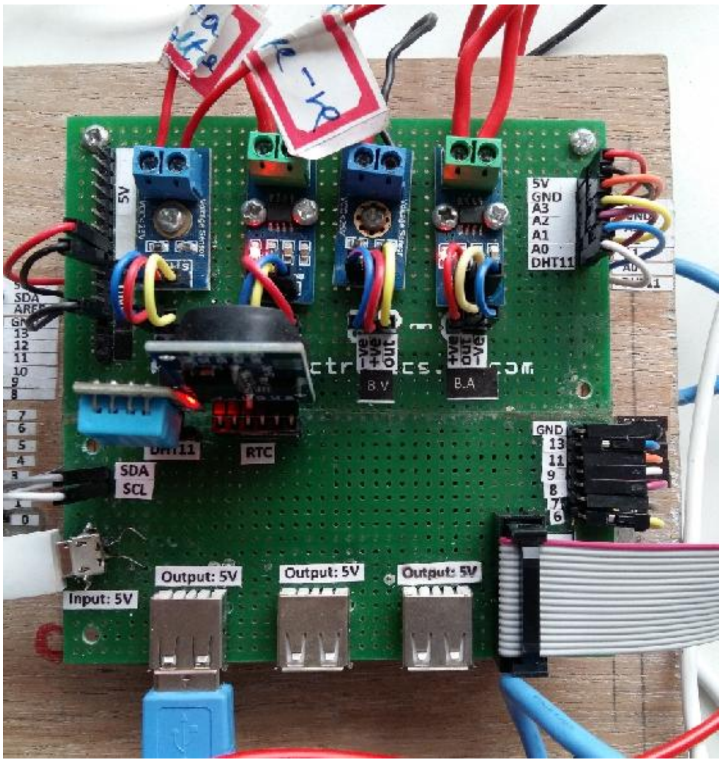

Figure 5.3: Various Sensors placed on General Purpose PCB

The **Figure 5.3** various sensors are placed on Single PCB.

1. **From top left to right :**

"Solar-Voltage sensor", "Solar-Current sensor", "Battery-Voltage sensor", "Battery-Current sensor". Jumper wire coming from Arduino.

2. **Middle left to right :**

DHT11 sensor and RTC module are placed and connections are given to Arduino**.**

3. **Bottom left to right :**

5V of supply coming from converter via battery is given to micro-USB port and split into three Type-A USB port where Arduino, GSM are powered up from this ports.

Towards the right side jumper wires coming from Arduino is connected to the Flat Ribbon cable connector internally. The data sent through the flat cable to the Red LED display as shown in **Figure 5.4**.

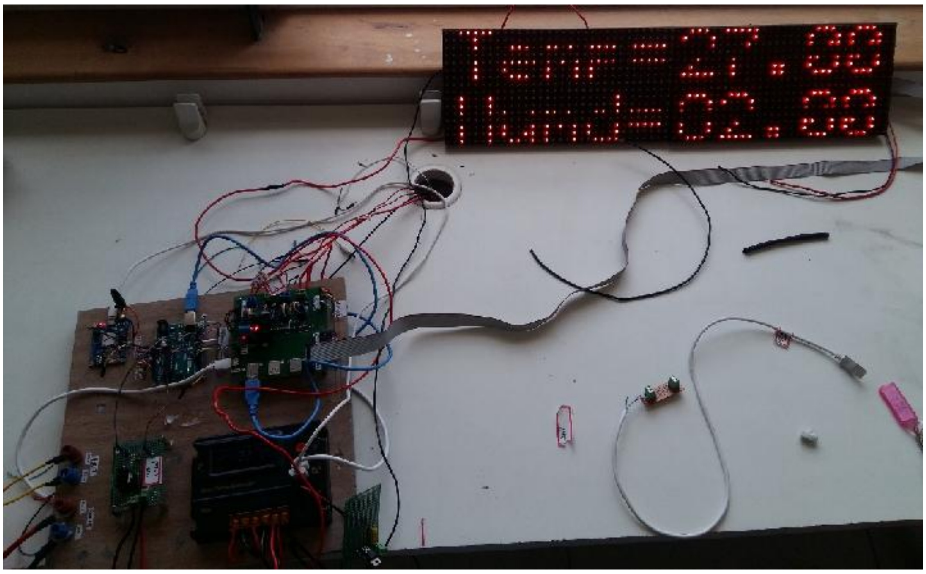

Figure 5.4: Red LED display is connected via flat cable from PCB

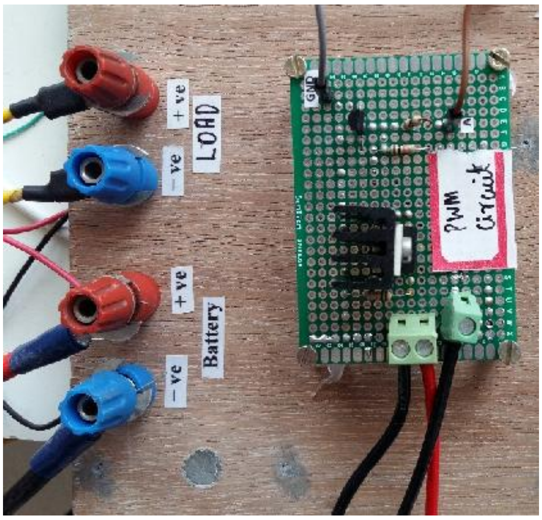

Figure 5.5: Battery, Light load, PWM connections

The **Figure 5.5** shows the battery and lamp load connections which are the left most side. This connection can be removed or connected back freely.

Towards the right side is the PWM circuit. From the bottom of this circuit negative and positive 12V supply are connected to the screw connectors. Positive of lamp is given to 12V and negative is given to 3rd screw connector. For more details of circuit you can refer **Figure 3.1**, in that PWM Control of Light circuit.

Where MOFET (IRF540) and Transistor (BC547), two 10KΩ and 1KΩ resistors are used. MOSFET is used because Arduino can’t provide more than 5V of supply for Lamp load to control PWM that is why we are using MOSFET.

Transistor is used because MOSFET is going draw more current from the Arduino which can only provide 500mA if it draws more current it might damage the Arduino. That is why we use transistor as protective circuit for Arduino and to control the PWM. It uses an inversion logic when logic 1 is given from Arduino the lamp is turned OFF and vice versa. Pin number 2 from Arduino is given to this circuit.

We have programmed it in such a way that whenever the solar voltage goes below 4 volts it is going to turn ON the lamp, if the solar voltage goes above 4.5V it going to turn OFF the lamp.

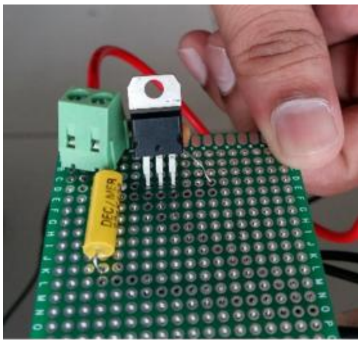

Figure 5.6: Stepdown from 12V to 5V for Red LED Display

The **Figure 5.6** shows the step down of 12volt to 5volt for Red LED display.

For circuit uses LM7805 IC, 0.22 micro F and 0.1 micro F capacitors for detail circuit refer **Figure 3.1** Step down of 12V to 5V dotted section for more details.

5.2 Current and Voltage measurements

The below **Table 5.1** shows the voltage and current measured by each devices

Table 5.1: Voltage and Current Measurement of devices

Sl no	Name of the Device	Voltage	Current
1	Arduino Uno	5 V	fluctuating from 27.7-72.2mA
2	GSM module	5V	fluctuating from 71.1-72.5mA
3	Red LED Display (2) when displaying Temperature and humidity	5V	Started from 0.4A and stabilised to 0.2 A
4	Red LED Display (2) when all led's are turned	5V	2.11A is consumed by both the led's.
5	Lamp load, 30W	12V	2.5A

5.3 Communication with GSM and ThingSpeak

For communication between GSM and the server we are using Transmission Control Protocol (TCP) or Internet Protocol, which uses networking protocols to communicate. The AT commands listed below is the sub-set from the whole AT command list which is been used in realisation.

Table 5.2: AT commands used for TCP IP communication

AT command for TCP	Description
AT+CGATT?	“GPRS service status”
AT+CSTT="internet"	“Start task and set APN”
AT+CIICR	“Bring up the wireless connection”
AT+CIFSR	“Get local IP address”
AT+CIPSTART="TCP","184.106.153.149","80"	“Starting the connection, where 184.106.153.149 is IP address of Thing Speak & port no 80”
AT+CIPSEND >GET/update?key=HZGMTC53I4KG2FZP&field3=16	“Send data to remote server”
AT+CIPCLOSE	“Closes TCP IP connection”
AT+CIPSHUT	“Closes the GPRS and PDP connections”

5.4 Data collected on thingSpeak website

5.4.1 Preconditions and setup

- ✓ The Solar panels are placed in balcony of Micro-controller lab EEE dept. PESU facing in west direction, only after 1pm or 2pm around that time enough irradiation is going to fall on the panels.
- ✓ Data is sent to the server every 15 minutes and Data is taken on cloudy days.
- ✓ Battery-1 was charged for about 16-17 hours by ac charger and data is taken from 22-06-2019 at 18:42:32 on wards till it was changed to battery-2.
- ✓ Battery-2 was charged for 2 whole days by ac charger and data is taken from 01-07-2019 at 18:42:00 on wards.
- ✓ In **Figure 5.11** and **Figure 5.13** for Battery Current Curve above zero is discharging curve and below zero is charging curve.

We have set 6 fields to upload the data to the ThingSpeak server as shown in the below figures.

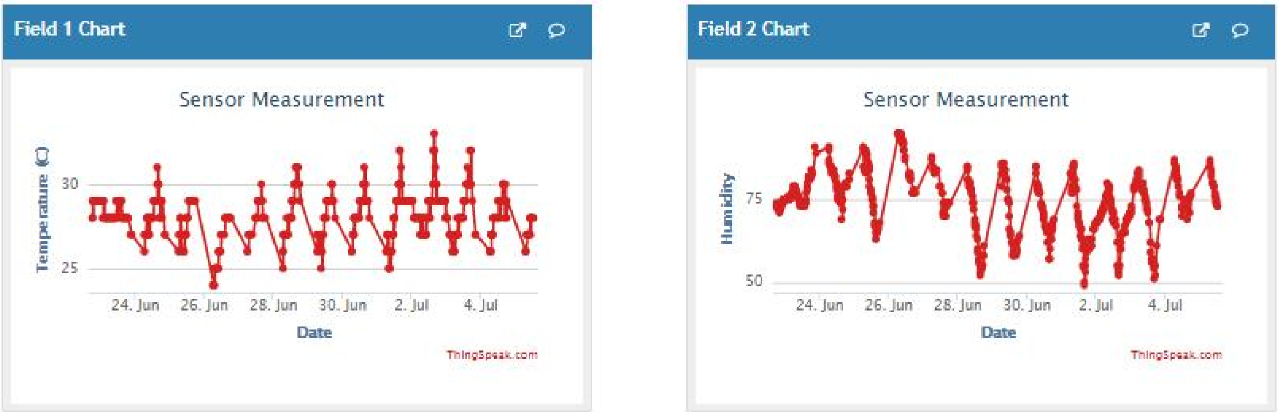

Figure 5.7: Temperature and Humidity data for 2 weeks

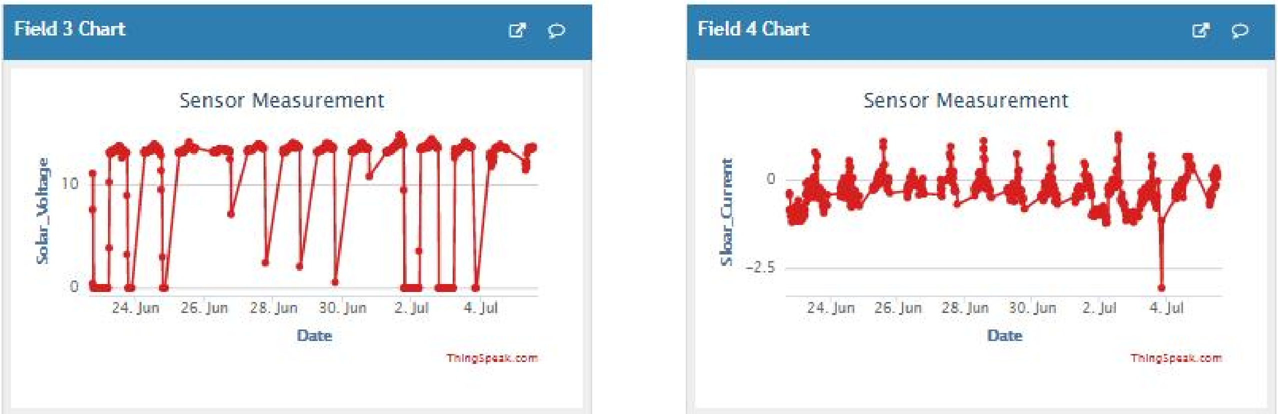

Figure 5.8: Solar Voltage and Current data for 2 weeks

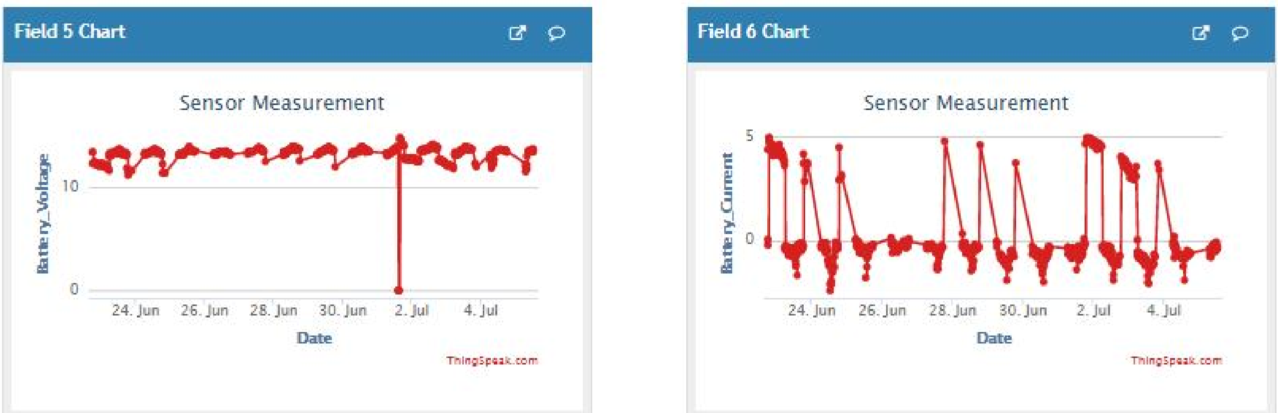

Figure 5.9: Battery Voltage and Current data for 2 weeks

5.4.2 Plotting of Data for Battery-1

The data is plotted for first three days. The below table gives the data obtained from the ThingSpeak Server in CSV file. Where **Table 5.3** gives discharging of data and **Table 5.4** gives charging of battery.

The below data is taken on 22-06-2019 at 18:42:32, the **BATTERY-1** was drained out before taking this reading and therefore we had to charge the battery using battery charger for 16-17 hours.

The data shows how much voltage and current is drawn from LED lamp, Arduino, GSM, Red LED Display and various other sensors on that particular night.

- In the below **Table 5.3** first few data are in for 5 minutes, later on it is taken for 15 minutes.

Table 5.3: Sensor data for discharging of battery-1

created_at	Time in 24hrs	field3= Solar Voltage	field4= Solar Current	field5= Battery Voltage	field6= Battery Current	Solar Power= (Solar Voltage)* (Solar Current)	Battery Power= (Battery Voltage)*(Battery Current)
22-06-2019	18:42:32	11.11	-0.43	13.37	0.05	-4.7773	0.6685
22-06-2019	18:47:22	7.6	-0.38	13.3	-0.22	-2.888	-2.926
22-06-2019	19:03:17	0.42	-0.86	12.23	4.36	-0.3612	53.3228
22-06-2019	19:19:12	0	-0.86	12.2	4.85	0	59.17
22-06-2019	19:35:08	0	-0.86	12.33	4.71	0	58.0743
22-06-2019	19:51:03	0	-1.02	12.43	4.95	0	61.5285
22-06-2019	20:06:58	0	-0.89	12.2	4.42	0	53.924
22-06-2019	20:22:53	0	-0.83	12.35	4.52	0	55.822
22-06-2019	20:38:49	0	-1.21	12.28	4.52	0	55.5056
22-06-2019	20:54:44	0	-1.16	12.33	4.5	0	55.485
22-06-2019	21:10:39	0	-0.94	12.15	4.58	0	55.647
22-06-2019	21:26:34	0	-0.92	12.33	4.63	0	57.0879
22-06-2019	21:42:29	0	-1.18	12.18	4.58	0	55.7844
22-06-2019	21:58:24	0	-0.97	12.28	4.44	0	54.5232
22-06-2019	22:14:18	0	-1.05	12.25	4.33	0	53.0425
22-06-2019	22:30:14	0	-0.86	12.08	4.07	0	49.1656

The battery discharging is plot in **Figure 5.11**

- ✓ The battery voltage is indicated in blue line discharges during night below 12V.
- ✓ The battery current is indicated in orange line discharges during night which is plotted above zero and charging is indicated below zero value.

Table 5.4: Sensors data for charging battery-1

Date & Time	Field3= Solar voltage	Field4= Solar current	Field5= Battery voltage	Field6= Battery Current	Solar Power= (Solar Voltage)* (Solar Current)	Battery Power= (Battery Voltage)* (Battery Current)
23-06-2019 06:11	10.28	-0.65	12.9	-0.35	-6.682	-4.515
23-06-2019 06:27	13.15	-0.4	13.17	-0.35	-5.26	-4.6095
23-06-2019 06:43	12.97	-0.11	13.02	-0.24	-1.4267	-3.1248
23-06-2019 06:59	13.07	-0.43	12.97	-0.48	-5.6201	-6.2256
23-06-2019 07:15	13.25	-0.38	13.3	-0.35	-5.035	-4.655
23-06-2019 07:31	13.17	-0.32	13.15	-0.4	-4.2144	-5.26
23-06-2019 07:47	13.22	-0.38	13.12	-0.51	-5.0236	-6.6912
23-06-2019 08:03	13.1	-0.54	13.05	-0.32	-7.074	-4.176
23-06-2019 08:19	13.32	-0.05	13.15	-0.73	-0.666	-9.5995
23-06-2019 08:35	13.37	-0.4	13.32	-0.4	-5.348	-5.328

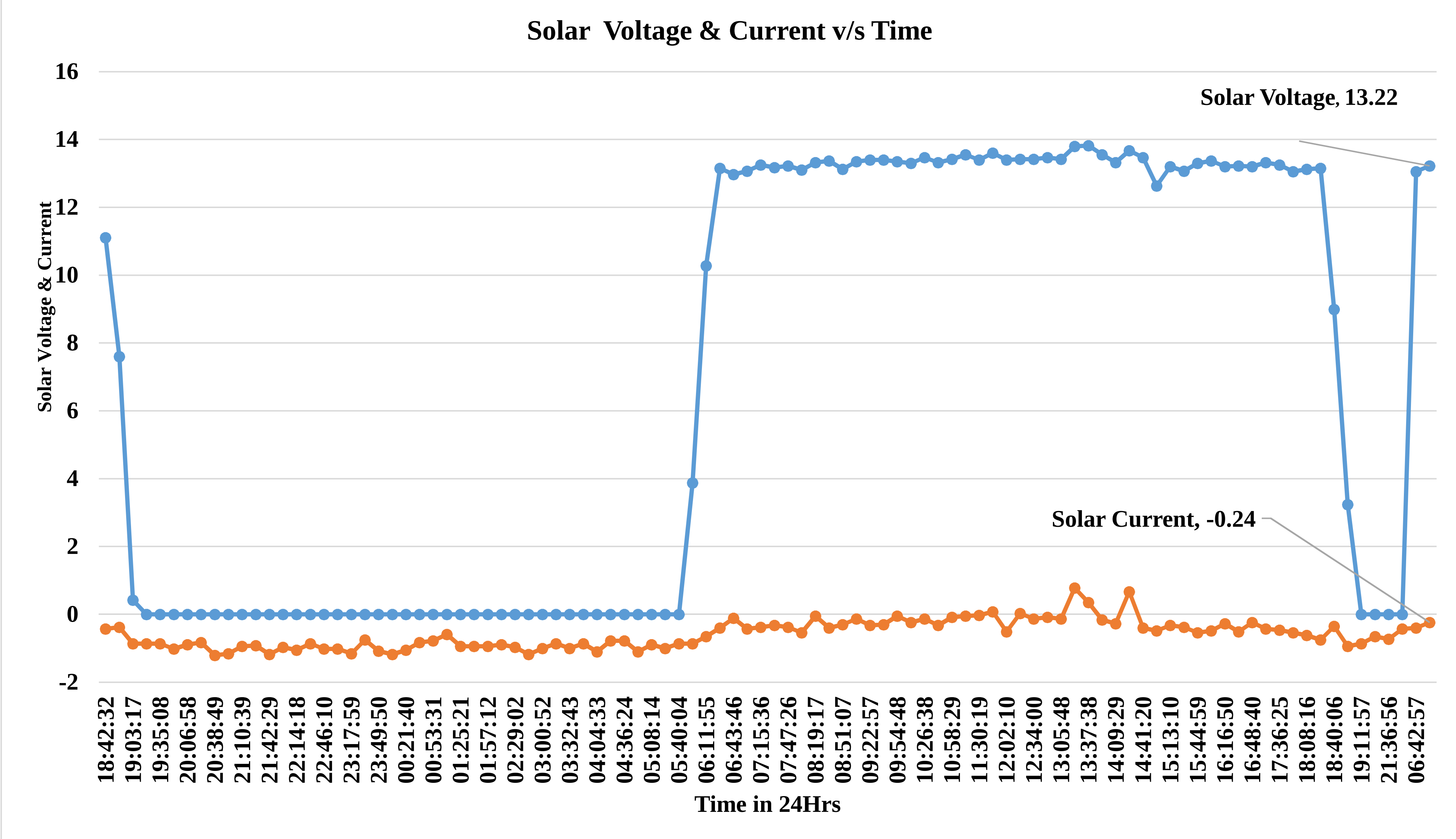

Figure 5.10: Solar Voltage and Current v/s time data from 22nd June to 24th June 2019

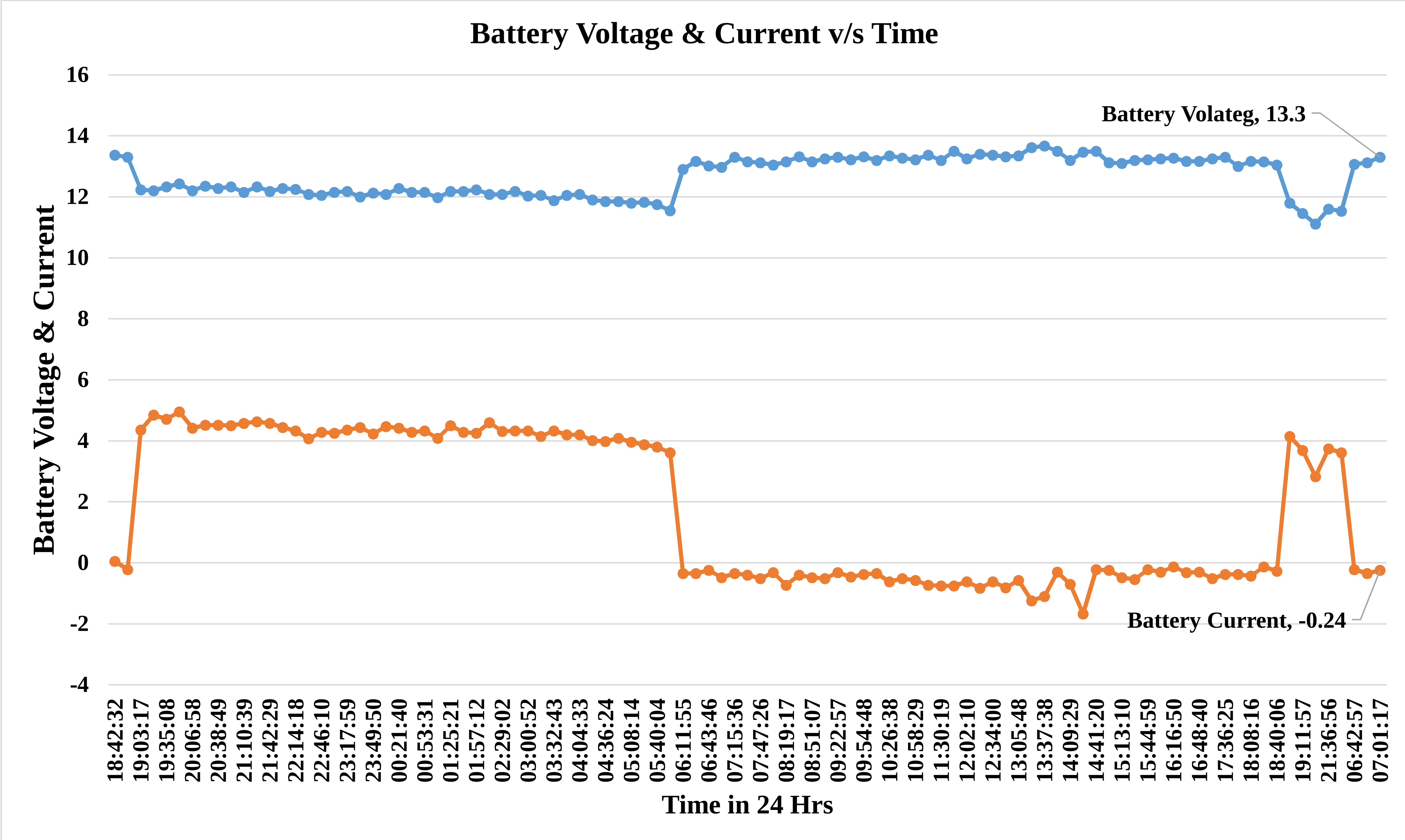

Figure 5.11: Battery Voltage & Current v/s Time data from 22nd to 24th June 2019

Table 5.5: Charging & Discharging Results of Battery-1 from 22nd -23rd June 2019

Day **PM (Evening)** **AM (Day Time)**	**Power (Wh)**	**Remarks**
22nd June – PM Discharged	535.61	Compared to theoretical postulation, Power consumed is more because the sensors give higher value than actual value.
23rd June – AM Charged	79.335	Enough sunlight was not there and hence power added is very low
23rd June – PM Discharged	51.92	As there was not enough power in the battery for the system to run the whole night, the battery got discharged at some point of time and the system went down.

5.4.3 Plotting data for battery-2

The below table gives the data obtained from the ThingSpeak Server in CSV file

The below data is taken on 01-07-2019 at 18:42:00, the **BATTERY-2** was drained out before taking this reading and therefore we had to charge the battery using battery charger for about 2 whole days

The data shows how much voltage and current is drawn from LED lamp, Arduino, GSM, Red LED Display and various other sensors on that particular night.

Table 5.6: Sensors data for discharging of Battery-2

created at	**Time in 24hr**	**Field3= Solar Voltage**	**Field4= Solar Current**	**Field5= Battery Voltage**	**Field6= Battery Current**	**Solar Power=(Solar Voltage)* (Solar Current**	**Battery Power= (Battery Voltage)* (Battery Current**
01-07-2019	**19:19:04**	**0**	**-1.02**	**12.85**	**4.63**	**0**	**59.4955**
01-07-2019	**19:34:59**	**0**	**-0.92**	**12.7**	**4.82**	**0**	**61.214**
01-07-2019	**19:50:54**	**0**	**-1.02**	**12.63**	**4.93**	**0**	**62.2659**
01-07-2019	**20:06:49**	**0**	**-1**	**12.8**	**4.9**	**0**	**62.72**
01-07-2019	**20:54:35**	**0**	**-1.05**	**12.7**	**4.79**	**0**	**60.833**
01-07-2019	**21:10:30**	**0**	**-0.94**	**12.77**	**4.87**	**0**	**62.1899**
01-07-2019	**21:42:21**	**0**	**-1.02**	**12.73**	**4.93**	**0**	**62.7589**

01-07-2019	22:46:01	0	-0.86	12.55	4.85	0	60.8675
01-07-2019	23:01:57	0	-0.83	12.65	4.77	0	60.3405
01-07-2019	23:17:52	0	-0.94	12.6	4.87	0	61.362
02-07-2019	00:05:37	0	-1	12.68	4.52	0	57.3136

Table 5.7: Sensor data for charging Battery-2

Date	Time in 24 hrs	Field3= Solar Voltage	Field4= Solar Current	Field5= Battery Voltage	Field6= Battery Current	Solar Power= (Solar voltage)* (Solar current)	Battery power= (Battery voltage)* (Battery current)
02-07-2019	06:27:40	13.42	-0.43	13.4	-0.3	-5.7706	1.73118
02-07-2019	06:43:35	13.52	-0.27	13.47	-0.38	-3.6504	1.38715
02-07-2019	06:59:30	13.67	-0.22	13.62	-0.48	-3.0074	1.44355
02-07-2019	07:15:25	13.62	-0.27	13.5	-0.24	-3.6774	0.88258
02-07-2019	07:47:15	13.52	-0.22	13.52	-0.38	-2.9744	1.13027
02-07-2019	08:03:11	13.72	-0.16	13.62	-0.24	-2.1952	0.52685
02-07-2019	08:19:06	13.5	-0.3	13.4	-0.24	-4.05	0.972
02-07-2019	08:35:01	13.57	-0.19	13.52	-0.35	-2.5783	0.90241
02-07-2019	08:50:56	13.72	-0.38	13.6	-0.46	-5.2136	2.39826
02-07-2019	09:06:51	13.57	0	13.4	-0.65	0	0

The battery discharging is plot in **Figure 5.13**

- ✓ The battery voltage is indicated in blue line discharges during night below 12V.
- ✓ The battery current is indicated in orange line discharges during night which is plotted above zero and charging is indicated below zero value.

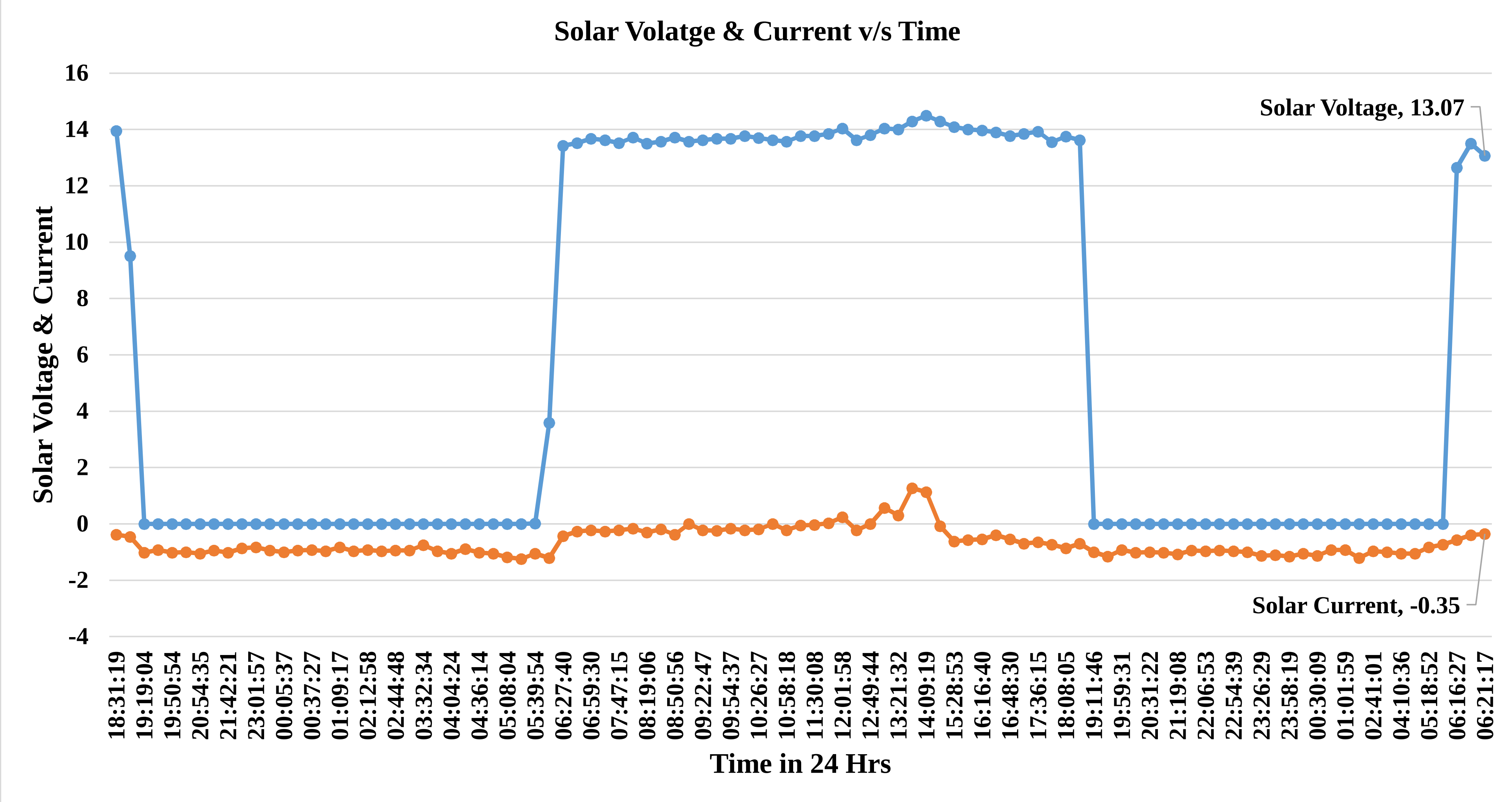

Figure 5.12: Solar Voltage & Current data for Battery-2 from 1st-3rd July 2019

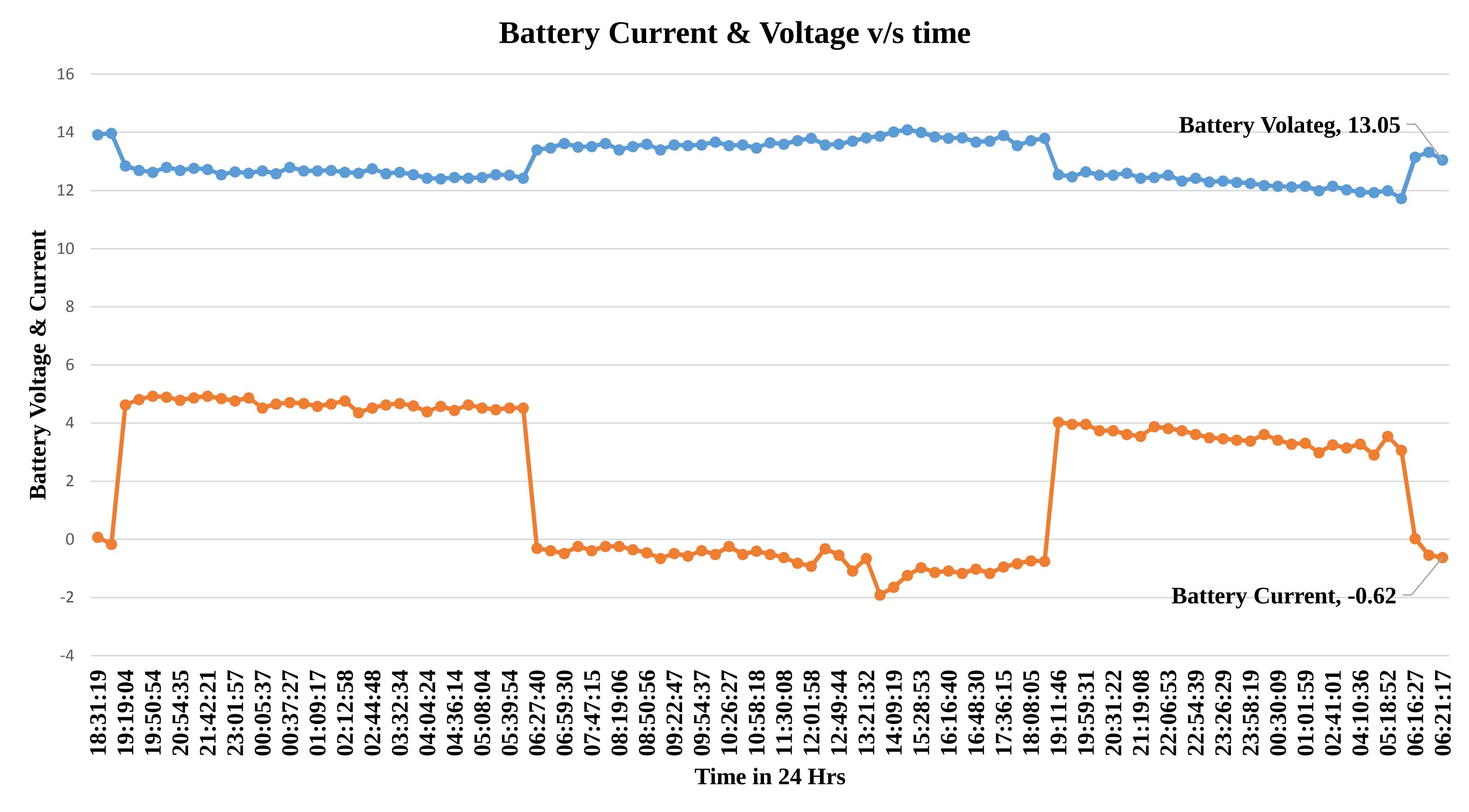

Figure 5.13: Battery-2 data fom 1st-3rd July 2019

Day PM (Evening) AM (Day Time)	Power (Wh)	Remarks
Start: 1st July – PM Discharged	441.04	Compared to theoretical postulation, Power consumed is more because the sensors give higher value than actual value.
2nd July – AM Charged	92.12	Enough sunlight was not there
2nd July – PM Discharged	280.38	Battery Discharged at some point of time.

5.5 Problem faced during the project

Sl no	Problem	Root cause	How was it fixed?
1	Initial connections with patch cords near battery and lamp for measuring purpose was giving us problems like: ❖ Battery used to get drained quickly..	➢ Impedance of path cord was more. ➢ Proper contact with the patch cord was not there	Then we changed the wire gauge to: ✓ Single stranded wires for measuring voltage. ✓ Multi-stranded wires for measuring current.
2	Intermittent issues with system working	❖ Repeated connection and disconnection without flexibility in connections ❖ Most of the time rats used to eat away the wires in the balcony.	✓ Routing the wires properly to the walls. ✓ Rewiring the connections for battery and panels many times
3	Complex wiring connections on bread board	❖ As circuit gets bigger more wires going above the circuit. ❖ It was consuming too much of space & figuring out which wire goes where was difficult.	✓ Then we shifted everything to General purpose PCB.
6	Intermittent issues with RED display	❖ Broken Flat cables used for communication between Arduino and red display.	✓ Crimped with 2x8 female connector by using crimping tool or by hammer

9	RED display goes off when LED output is switched ON via PWM	❖ Simultaneous usage of PWM and SPI causes issues with ATMEGA328	✓ Temporary fix is to use digital outputs for ON and OFF only (No intensity control) ✓ …….
10	Compiler issues when Adafruit library is integrated	❖ Too many dependencies.	✓ Independent library is used for DHT11.

5.6 Conclusion

1. From **Table 3.1** we see that postulated calculation for system consumption for is 420Wh. Battery-1 power consumption at 22nd June night was 541Wh and Battery-2 power consumption at 1st July was night is 441Wh. The difference between the postulated and actual reading mainly due to sensors give higher value than actual value.
2. The above results are taken when Red LED display is continuously turned for 24hr.
3. Power Optimization needs to be considered for LED display and Lamp Light to make the system self-sustainable based on solar energy.

Appendix

The smart lamp post related codes, datasheets and GSM tests are uploaded to GitHub account, below is the link

https://github.com/DeepakMK2613/Smart-Lamp-Post

The final code used in the project is saved in “SmartLampPost_V1” file in GitHub.

References.

[1] "S. Siregar and D. Soegiarto, Solar panel and battery street light monitoring system using GSM wireless communication system, in *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, 28-30 May 2014 2014, pp. 272-275, doi: 10.1109/ICoICT.2014.6914078".

[2] "N. Yoshiura, Y. Fujii, and N. Ohta, Smart street light system looking like usual street lights based on sensor networks, presented at the 2013 13th International Symposium on Communications and Information Technologies (ISCIT), 2013".

[3] "M. Revathy, S. Ramya, R. Sathiyavathi, B. Bharathi, and V. M. Anu, Automation of street light for smart city, in *2017 International Conference on Communication and Signal Processing (ICCSP)*, 6-8 April 2017 2017, pp. 0918-0922, doi: 10.1109/ICCSP.2017.8286503".

[4] "B. Roy, A. Acharya, T. K. Roy, S. Kuila, and J. Datta, A smart street-light intensity optimizer, in *2018 Emerging Trends in Electronic Devices and Computational Techniques (EDCT)*, 8-9 March 2018 2018, pp. 1-4, doi: 10.1109/EDCT.2018.8405098".

[5] "G. Gagliardi *et al.*, A smart city adaptive lighting system, in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, 23-26 April 2018 2018, pp. 258-263, doi: 10.1109/FMEC.2018.8364076".

[6] "M. Saifuzzaman, N. N. Moon, and F. N. Nur, IoT based street lighting and traffic management system, in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 21-23 Dec. 2017 2017, pp. 121-124, doi: 10.1109/R10-HTC.2017.8288921".

[7] "A. Jha and M. Kumar, A wide range constant current LED driver with improved power quality and zero standby, in *2018 IEEMA Engineer Infinite Conference (eTechNxT)*, 13-14 March 2018 2018, pp. 1-6, doi: 10.1109/ETECHNXT.2018.8385327".

[8] "D. K. Srivatsa, B. Preethi, R. Parinitha, G. Sumana, and A. Kumar, Smart Street Lights, in *2013 Texas Instruments India Educators' Conference*, 4-6 April 2013 2013, pp. 103-106, doi: 10.1109/TIIEC.2013.25".

[9] "B. Kul, IoT-GSM-based high-efficiency LED street light control system (IoT-SLCS), in *2017 XXVI International Scientific Conference Electronics (ET)*, 13-15 Sept. 2017 2017, pp. 1-5, doi: 10.1109/ET.2017.8124361".

[10] "A. Szalai, T. Szabó, P. Horváth, A. Timár, and A. Poppe, Smart SSL: Application of IoT/CPS design platforms in LED-based street-lighting luminaires, in *2016 IEEE Lighting Conference of the Visegrad Countries (Lumen V4)*, 13-16 Sept. 2016 2016, pp. 1-6, doi: 10.1109/LUMENV.2016.7745518".

Smart Lamp Post Platform Development

by Deepak M K

Submission date: 20-Jul-2019 03:13PM (UTC+0530)
Submission ID: 1153403845
File name: SLPPD_V3.3.pdf (1.78M)
Word count: 7692
Character count: 35550

Smart Lamp Post Platform Development

ORIGINALITY REPORT

1%	0%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Technological Institute of the Philippines Student Paper	<1%
2	Submitted to Institute of Technology, Nirma University Student Paper	<1%
3	Submitted to University of Birmingham Student Paper	<1%
4	Submitted to University of Houston System Student Paper	<1%
5	Submitted to University of East London Student Paper	<1%
6	Submitted to University of Southampton Student Paper	<1%
7	www.cheers4all.com Internet Source	<1%
8	Submitted to University of Newcastle upon Tyne Student Paper	<1%