REMOTELY MONITORED IoT SERVICE

A Report on Major Project Analysis and Design
Semester VII

SUBMITTED BY

DURGESH KOLTE	**[B206003]**
YASH PAMNANI	**[B206011]**

GUIDED BY

Prof. Vaishali Katkar

SCHOOL OF ELECTRICAL ENGINEERING

MIT | Academy of Engineering
[An Autonomous Institute affiliated to Savitribai Phule Pune University]

ALANDI (D), PUNE-412105, MAHARASHTRA, INDIA

DECEMBER 2020

REMOTELY MONITORED IoT SERVICE

A Report on Major Project Analysis and Design
Semester-VII

Submitted in partial fulfilment of the requirements of Semester VII for the award of the degree

of

Bachelor of Technology

in

ELECTRONICS ENGINEERING

BY

DURGESH KOLTE	**[B206003]**
YASH PAMNANI	**[B206011]**

SCHOOL OF ELECTRICAL ENGINEERING

MIT Academy of Engineering

[An Autonomous Institute affiliated to Savitribai Phule Pune University]

ALANDI(D), PUNE-412105, MAHARASHTRA (INDIA)

DECEMBER 2020

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

CERTIFICATE

It is hereby certified that the work which is being presented in the B.Tech. Major Project Analysis and Design [ET432] Report entitled **"Remotely monitored IoT Service",** in partial fulfillment of the requirements of Semester VII for the award of the **Bachelor of Technology in Electronics Engineering** and submitted to the **SCHOOL OF ELECTRICAL ENGINEERING of MIT Academy of Engineering, Alandi (D), Pune** is an authentic record of work carried out during a period from August 2020 to December 2020 under the supervision of **Prof. Vaishali Katkar, School of Electrical Engineering**.

DURGESH KOLTE	**PRN No. 0120170036**	**Exam Seat No. B206003**
YASH PAMNANI	**PRN No. 0120170159**	**Exam Seat No. B206011**

Date: 19/11/20

Dean,
School of Electrical Engineering,
MIT Academy of Engineering, Alandi (D), Pune

Seal of School

(STAMP)

Date:

Signature of Internal examiner/s

Signature of External examiner/s

*Name:*________________________________

Name: ________________________________

CERTIFICATE

The matter presented in this major project report has not been submitted by us for the award of any other degree elsewhere.

Date: 19/11/20

Signature of Candidates

(Name: Durgesh Kolte)

(Name: Yash Pamnani)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: 19/11/20

Signature of project Advisor

(Name: Prof. Vaishali Katkar)

INDEX

CHAPTER-1

1. INTRODUCTION

a. MOTIVATION:

The demand for these circuits is increasing day by day with increasing Internet of Things' (IoT) products, Nuclear Reactors and Medical devices which demand low power consumption for better battery life and high-speed operations. Nowadays, technology is transforming the system in becoming compact throughout, this is possible due to the use of Complementary Metal Oxide Semiconductor Field-Effect Transistor. And hence the below diagram depicts perfectly how we intend to create a network efficient enough to serve the need.

b. PROBLEM STATEMENT:

To deploy IoT Service in Remote location to monitor, analyze & share data wirelessly over the cloud web service to end-user.

c. OBJECTIVES:

- **i.** To design a solar-powered system that works independently without power grid
- **ii.** To build a network that can transfer data wirelessly from/to sensors remotely over miles of distances without internet
- **iii.** To interface the received data from a remote location through cloud
- **iv.** To monitor all the data from the source site to a remote location
- **v.** To create conditional events

CHAPTER-2

2. LITERATURE SURVEY

The service industry is growing at a much faster rate. Almost all the big giants are now moving their primary source of income from product to service as it is more reliable. And parallel to that IoT services is getting implemented at a larger and faster pace. This was possible due to Industry 4.0 majorly which was backed up by the wider reach of 4G network bandwidth at lower costs and higher speeds. This not only helped by digitalizing small industries but also made the availability of internet connectivity over on-the-go platforms like vehicles. Thus, the automation industry emerged broadly covering all aspects of life and making it easier, reliable, efficient and cheaper as compared to conventional manual methods.

The author focuses on calculating the amount of water used in the field along with moisture and temperature numbers. These numbers are then getting uploaded to their cloud web service with the help of ESP8266 and it's Wi-Fi module. This module is an ideal choice for any IoT application and is a low-cost device. The main drawback was the limitation of the size of string while sending to cloud by ESP8266 was found to be 64 characters. But since the ESP and sensors connections are wired, this limits the placement of sensors. Using a charge controller is must to regulate the flow of current and voltage to maintain the PV graph if powering the system through solar as shown in these papers [2] [6].

The main crux is regarding setting conditional events. The demonstration has shown there is setting a threshold value to the temperature sensor. When the sensor crosses that safe operating limit then it triggers a real-time alert sent on users' mobile phone through SMS as well as email id. They also show proper development of dashboard to represent graphical data of reading that are on their cloud hosting service. The cloud platform used by them is Ubidots which is fairly simple to set-up and operate for free. This can help the data center taking immediate actions to a specific cause [5].

Remaining papers are a summarization of technical use case of NRF module for wireless data transfer over a range of 1 km without the need for internet connectivity as per [4]. We can also see the good implication of specialized energy-saving feature of ESP8266 which is a sleep mode that ensures power saving by 100 times according to [3].

The main conclusion from all the reviewed paper can be as follows

- **a.** ThinkSpeak & Ubidots are great Cloud Hosting open source platforms with user friendly dashboard.
- **b.** NRF module wireless System can be implemented over a large area (1 km per node) for a comparatively minor investment.
- **c.** ESP8266 is an ideal controller to deal with IoT operations because of its wireless and power saving functionalities.
- **d.** Ubidots can be used to set conditional events and trigger alerts on SMS and email id.

CHAPTER-3

3. THEORY

a. CHARGE CONTROLLER:

TP4056 is the module in our project to be used as a charge controller. It ensures a constant flow of current/voltage from the solar panel while charging the Li-ion battery. This is necessary because solar panels generate energy according to the photosensitivity. Hence, less sunlight would imply low power generation that in turn results in less current and voltage output from the solar panel. Thus, throughout the day, the panel would generate an uneven amount of power at unpredicted battery voltage and current. To regulate this flow TP4056 is fitted which is also cheap in nature considering its cost. It also helps to disconnect the circuit in case of overcurrent or if the battery is charged. It is a single-cell Li-ion battery charge controller used to regulate and monitor current with other features including under-voltage lockout, automatic recharge and status pin indicators for charge termination or presence of voltage respectively. With the help of below charge cycle graph, one can estimate the charging times for a battery.

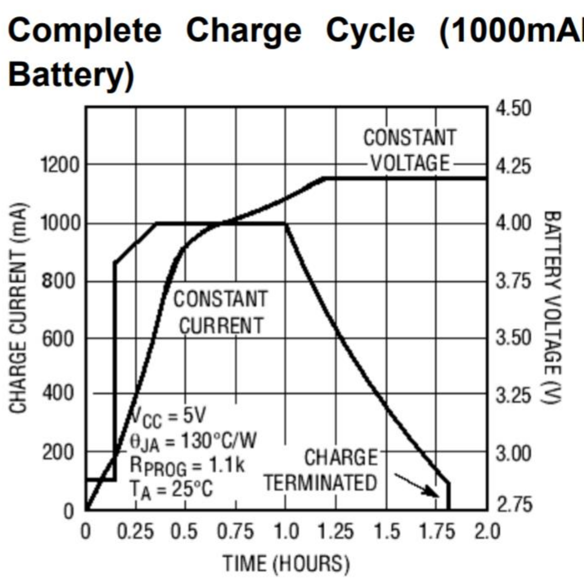


Fig. 1 TP4056 charge cycle

b. ESP8266 (SLEEP-MODE):

ESP8266 is a simple microchip produced by Expressif systems. It is a low-cost microcontroller with embedded Wi-Fi module in it. This module connects directly through TCP/IP protocol. It is referred to as NodeMCU generally consumes 75mA with Wi-Fi ON and uses the Arduino IDE environment. ESP8266 has a very peculiar feature of sleep-mode apart from the active mode. It has three levels of sleep-mode namely modem, light and deep sleep-modes. The main purpose of them is disabling particular finalities like Wi-Fi or clock according to the level of sleep-mode to reduce current consumption to as low as 250 uA.

Item	Modem-sleep	Light-sleep	Deep-sleep
Wi-Fi	OFF	OFF	OFF
System clock	ON	OFF	OFF
RTC	ON	ON	ON
CPU	ON	Pending	OFF
Substrate current	15 mA	0.4 mA	~20 uA
Average current (DTIM = 1)	16.2 mA	1.8 mA	–
Average current (DTIM = 3)	15.4 mA	0.9 mA	–
Average current (DTIM = 10)	15.2 mA	0.55 mA	–

Table 1 Sleep-mode details

c. SERIAL PERIPHERAL INTERFACE (SPI):

The wireless communication of the nRF module uses the SPI communication protocol. The nRF + transceiver communicates over 4-pin SPI with maximum 10 Mbps data rate. The 125 frequency channels on the wireless module can be configured through SPI at 250 Kbps, 1 Mbps and 2 Mbps at pre-defined output power. This system works on the master-slave concept, microcontroller like ESP8266/Arduino being masters and nRF/transceiver being slaves. This signal range can be boosted further using PA LNA setup discussed further. It is a synchronous parallel communication protocol using data bus as shown below. The oscillating clock ensures the sampling cycle on both sides and separate data lines are used for data and clock.

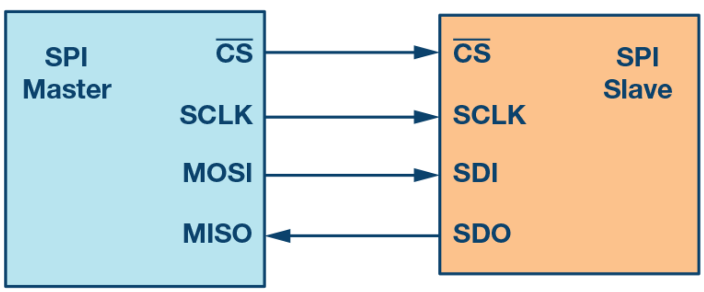


Fig. 2 SPI communication protocol

d. APPLICATION PROGRAMMING INTERFACE (API):

Ubidots REST API is used for the creation of dashboard and tweaking variables. It is a software tool that helps user and programmer to communicate with each other through a user-friendly interface. It is flexible enough to work with any type of external hardware device making it agnostic. This API will call or request data as long as the user can communicate through at least one of HTTP, MQTT and TCP/UDP protocols. So, Ubidots receives data from the device through these protocols every time a sensor data is updated. This updated data point is stored into variables at their respective timestamps.

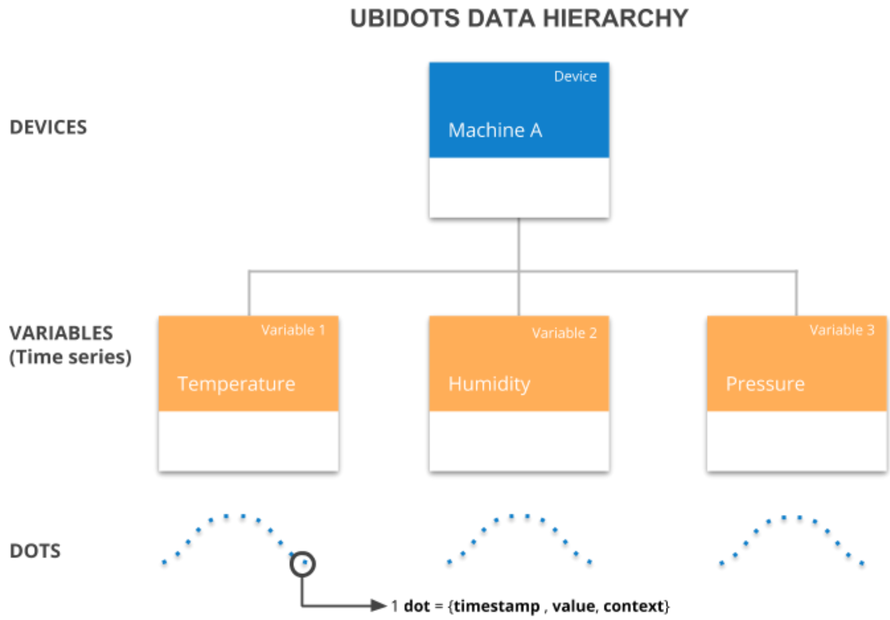


Fig. 3 Ubidots Data Hierarchy model

i. HYPERTEXT TRANSFER PROTOCOL (HTTP):

HTTP is based on request-response protocol. It transmits hypertext messages to access web resources through hypermedia HTML files. It uses Transmission Control Protocol (TCP) to establish a connection between client and servers. The communication in this layer refers to the application layer. It can be used with or without internet by setting the access pint to the server node ESP8266 and the client end as the station point. This way both the devices can access the data through the request-response method.

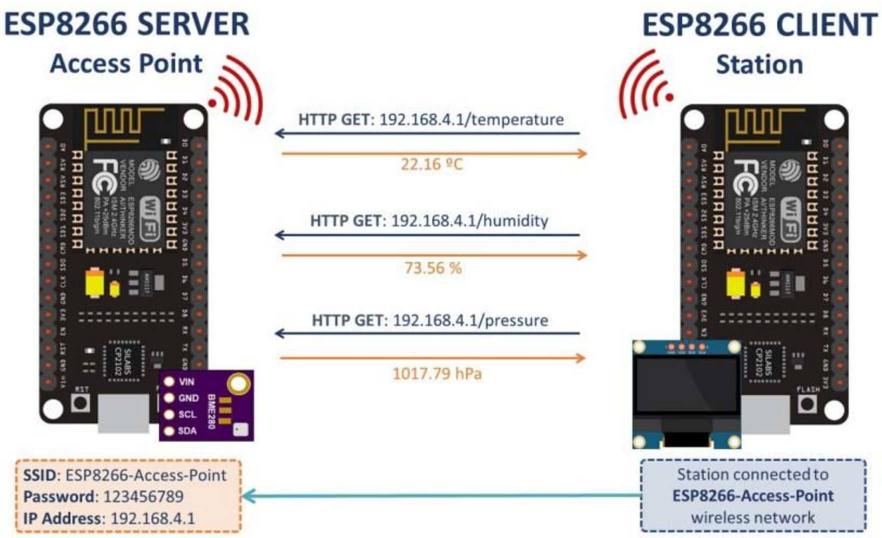


Fig. 4 HTTP client-server communication

ii. Message Queuing Telemetry Transport (MQTT):

MQTT is the most widely used communication protocol in the field of IoT. It is a publish-subscribe based messaging protocol which has low power consumption and efficient-bandwidth as compared to others. It is an OASIS and ISO standard network protocol which runs over TCP/IP but not limited to it. Any network which is bi-directional, ordered and lossless connection can support MQTT. It allows edge-of-network devices to publish to a broker. The client connects to the broker and as per the subscription plan, the connection is mitigated.

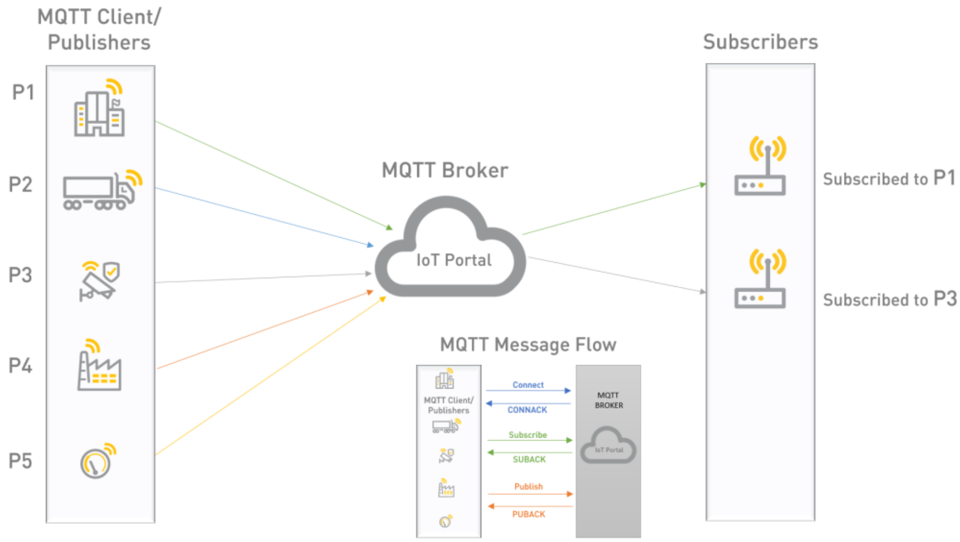


Fig. 5 MQTT publish-subscribe model

iii. TCP / UDP:

Transmission Control Protocol (TCP) is connection-oriented protocol which means it establishes connection between sender and receiver like before sending data like emails. But on the other hand, User Datagram Protocol (UDP) is connectionless and doesn’t establish connection before sending like call.

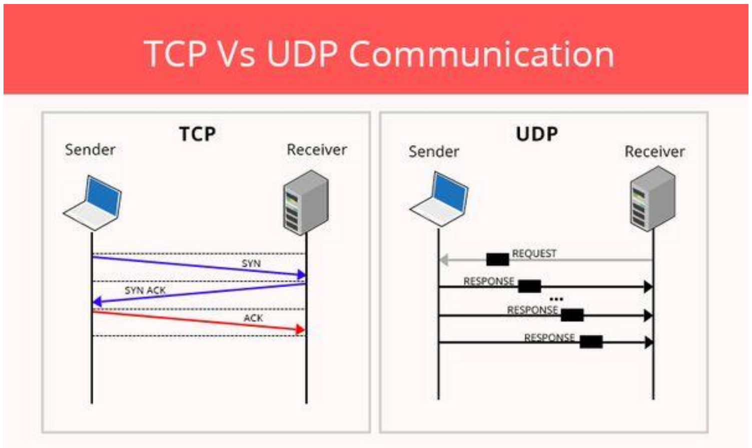


Fig. 6 TCP vs UDP communication

CHAPTER-4

4. PROJECT ANALYSIS

a. POWER SOURCE:

Considering we can get solar energy from 9:00 AM to 5:00 PM, we can easily say that the Power Efficiency we expect from Solar Panel would be 75% of whatever calculated as we won't have 100% energy throughout the day.
Location: Pune

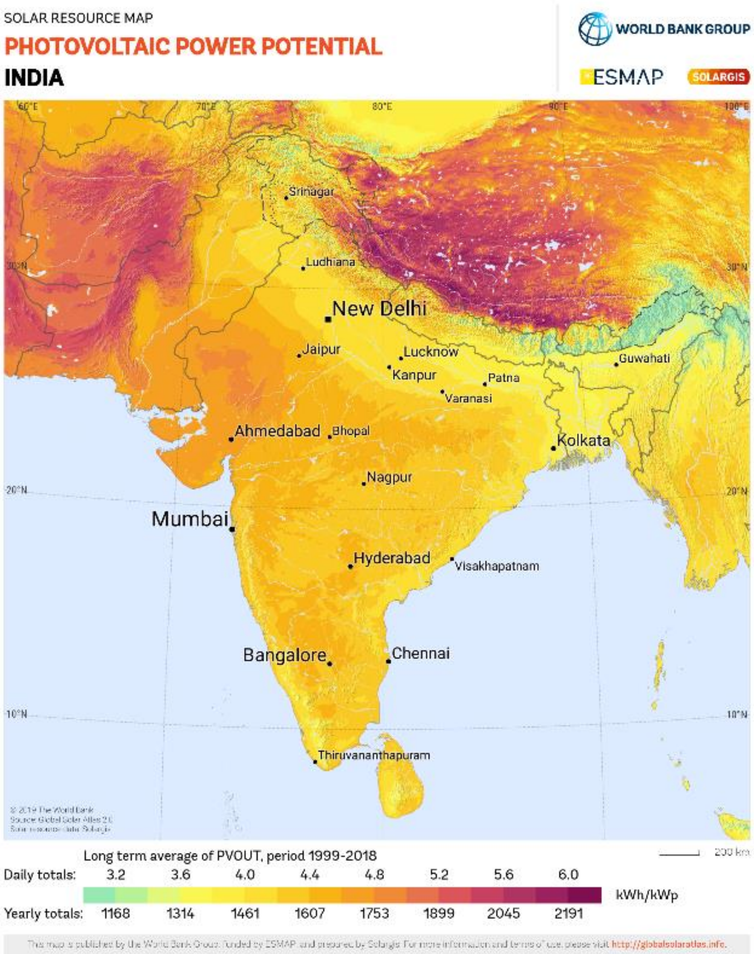


Fig. 7 Solar Power Potential of India heat signatures

Solar Power (per sq. meter):

YEARLY:	1607	KW
MONTHLY:	4.4	KW
DAILY:	183	W

b. BATTERY SELECTION:

Solar Panel Efficiency (avg)	: 10% of Sunlight
Solar Panel Rating (max)	: 18 Watt
Actual Solar Panel Rating (6 volt)	: 10 Watt
Current Capacity (max)	: 1.67 amp
Power Efficiency (avg for 8 hrs)	: 75%
Battery Voltage (max)	: **4.5 V**
Battery Current (max)	: **1.25 amp**

c. CURRENT DISCHARGE (ACTIVE MODE):

Current Drawn:

ESP8266 + Wi-Fi	= 75 mA
Sensors	= 25 mA
Total Current	= **100 mA**

Battery Runtime (1500 mAh):

Capacity / Current Drawn	= **15 hrs**

But since, 15 hrs won't be sufficient to charge a battery fully as we might have days where we don't receive sunlight at all and then this system would fail within a day. And hence we have to take into account a longer period. For this main reason along with the addition of Wi-Fi/RF on ESP8266, it is the perfect choice for our use. Hence, we would be using the Deep-Sleep mode feature of the ESP8266 to extend the runtime of battery.

CHAPTER-5

5. METHODOLOGY

a. BATTERY RUNTIME:

In this mode, the CPU and all peripherals are paused. Any wake-up such as external interrupts will wake up the chip. Without data transmission, the Wi-Fi Modem circuit can be turned off and CPU suspended to save power consumption.

Current Drawn:

Active Mode	= 100 mA
Sleep Mode	= 0.25 mA
Average Current (every 5 min)	= [100*10 + 0.25*290] / 300 = **3.575 mA**

Battery Runtime:

Capacity / Current Drawn	= 420 hrs = **17.5 Days**

Therefore, with the help of below command, we can trigger the controller into deep-sleep Mode where it consumes 250 uA rather than 100mA which extends the Runtime up to 17 days which is much more than sufficient to charge a 1500mAh battery. But the maximum, sleep time is 71 minutes in a flow for ESP8266.

```
ESP.deepSleep(sleepTimeInSeconds);
```

b. VOLTAGE REGULATION:

After a certain voltage level, the battery won't discharge. This is due to the insufficient current. This requires regulation of voltage of the system to avoid shutting down of the system. This can be done by limiting it to a certain value. So, whenever the battery voltage crosses that value, we switch it into deep-sleep mode into which it can run for a longer period. And also collect data on an hourly basis rather than every 5 minutes.

This is where things get interesting. Since on ESP8266 boards, we cannot read the battery voltage that is Vin without external components. But indirectly we can detect the discharged battery through measuring input voltage of ESP8266 which is V3.3. There is a method to detect a discharged battery indirectly, by measuring the input voltage of ESP8266 that would be V3.3. Following graph shows the corresponding values of battery for detected voltage on V3.3.

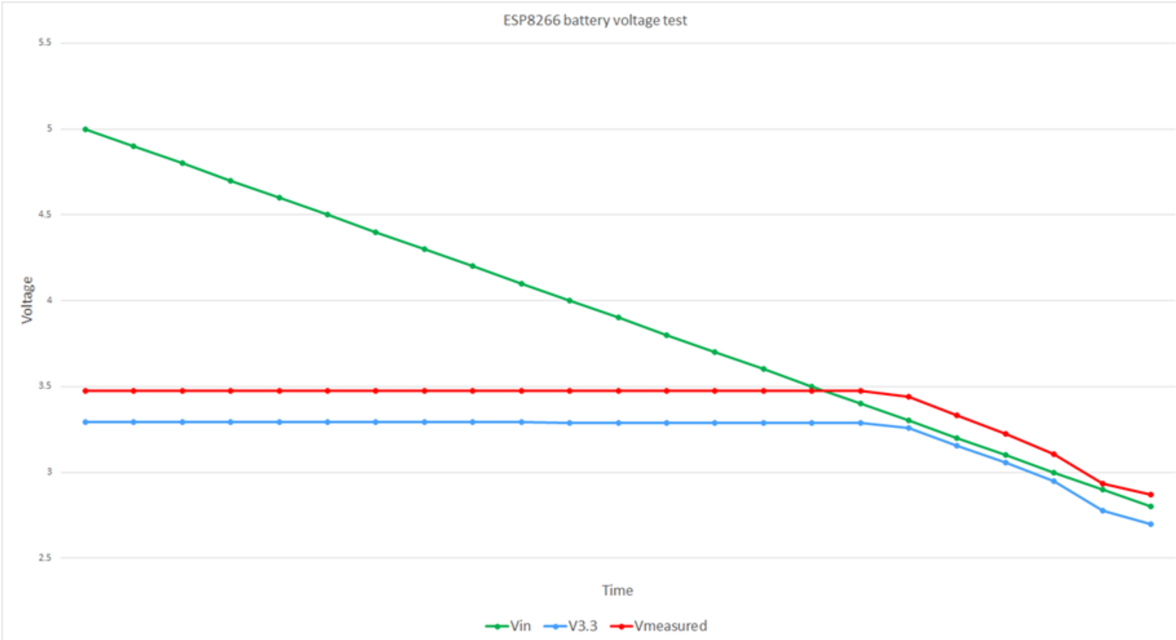
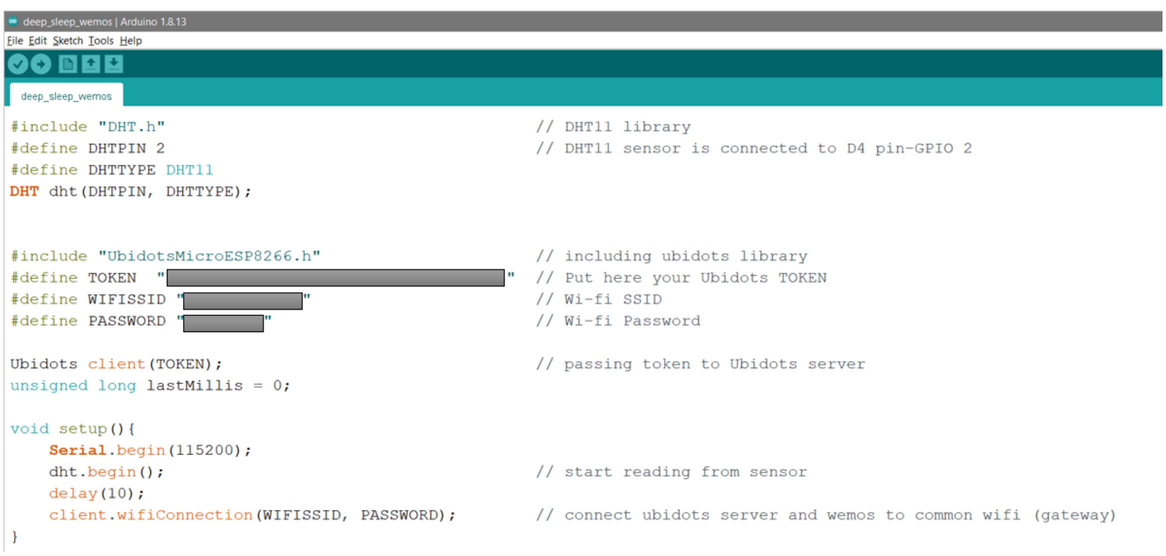


Fig. 8 Actual battery voltage vs measured voltage

This voltage at V3.3 can be generated using following code

```
Batt = ESP.getVcc();
  // If the battery is discharged don't go any further!!!
  if (Batt < 3100) {
    // Deep sleep for as long as you can
    ESP.deepSleep(ESP.deepSleepMax());
  }
```

c. UBIDOTS ACCESS-POINT:



deep_sleep_wemos | Arduino 1.8.13

File Edit Sketch Tools Help

deep_sleep_wemos

```
#include "DHT.h"                              // DHT11 library
#define DHTPIN 2                              // DHT11 sensor is connected to D4 pin-GPIO 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);


#include "UbidotsMicroESP8266.h"              // including ubidots library
#define TOKEN  "[redacted]"                   // Put here your Ubidots TOKEN
#define WIFISSID "[redacted]"                 // Wi-fi SSID
#define PASSWORD "[redacted]"                 // Wi-fi Password

Ubidots client(TOKEN);                        // passing token to Ubidots server
unsigned long lastMillis = 0;

void setup(){
    Serial.begin(115200);
    dht.begin();                              // start reading from sensor
    delay(10);
    client.wifiConnection(WIFISSID, PASSWORD);      // connect ubidots server and wemos to common wifi (gateway)
}
```

Fig. 9 Connect ESP8266 to Ubidots APN Server

CHAPTER-6

6. MODULE DESCRIPTION

a. nRF24L01 WIRELESS MODULE:

The inexpensive nRF24L01 module is a device used for 2-way communication between two microcontrollers reliably up to 1 Km of rated distance. This operates on 2.4 GHz frequency which is the basic frequency used in many modules like NFC and Bluetooth for communication. This single-chip radio transceiver is certified to be used in most of the countries due to its Industrial, Scientific and Medical (ISM) band compatibility. The data transfer rates that this device supports is 250 Kbps, 1 Mbps and 2 Mbps. The operating voltage can be between 1.9 to 3.6 volts for its logic pins but can be used up to 5 volts. With one module we can connect 125 distinct channels for communication spaced at 1 MHz bandwidth.

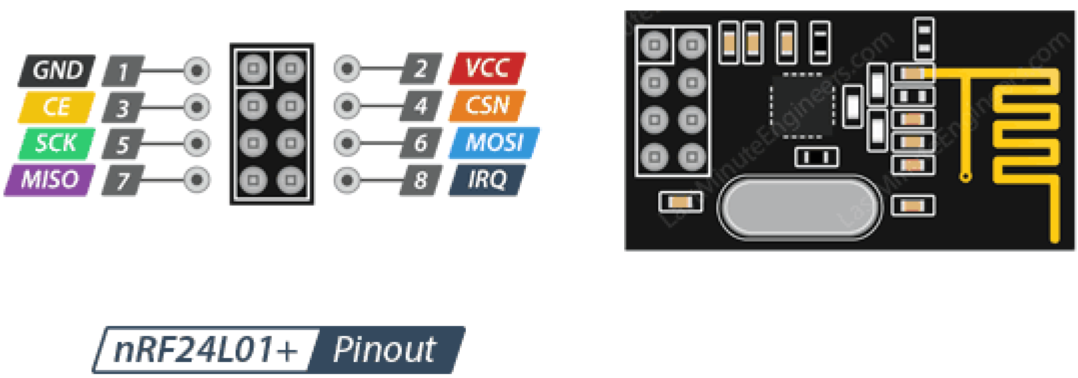
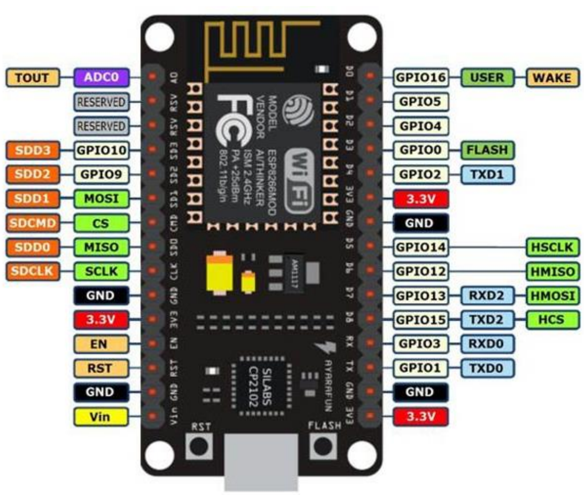


Fig. 10 nRF24L01+ Transceiver Module Pinout

b. DHT22 SENSOR MODULE:



DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND

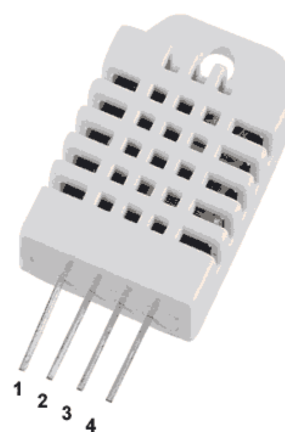


Fig. 11 ESP8266 pinout diagram

Fig. 12 DHT22 sensor module pinout

DHT22 is a temperature and humidity sensing module. It is later version of its predecessor DHT11 which is slightly cheaper but has its disadvantages. DHT22 is more reliable in terms of humidity measuring range from 0-100% accounting 2-5% accuracy. But at the same time, DHT11 has 20-80% range with 5% accuracy. The major advantage of DHT22 is power consumption. DHT11 consumes 50 mA of current whereas DHT22 consumes only 5 mA for the same thing but more accurate. Hence, for power conserving projects like ours, DHT22 is the ideal choice for any IoT standalone system.

c. AMPLIFIER MODULE:

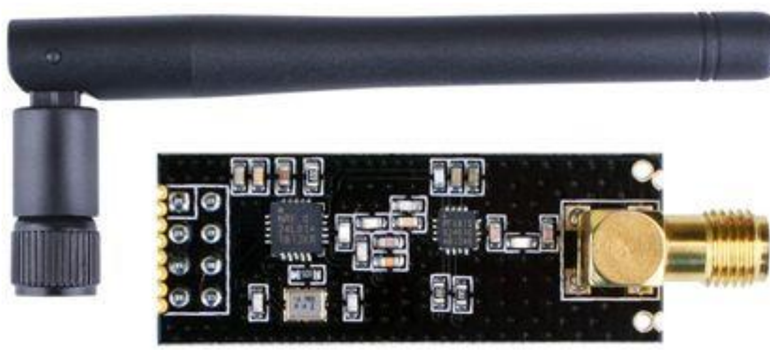


Fig. 13 nRF24L01 + PA LNA wireless module + antenna

PA LNA transceiver module is attached to the nRF module to boost its communication range. The regular nRF has a range of about 100 m but after adding this external system, it boosts up to 1 Km. This greatly increases the usability case of the system. This difference is observed due to the RFX2101C chip implanted in the module. It also has a duck-antenna to support the transmission. This chip consists of a transmit-receive switching circuit along with Power Amplifier (PA) to boost the transmitting signal from the chip. But the extremely weak and uncertain signals from the antenna are captured by the Low-Noise Amplifier (LNA) transforming microvolt signal up to 1 volt.

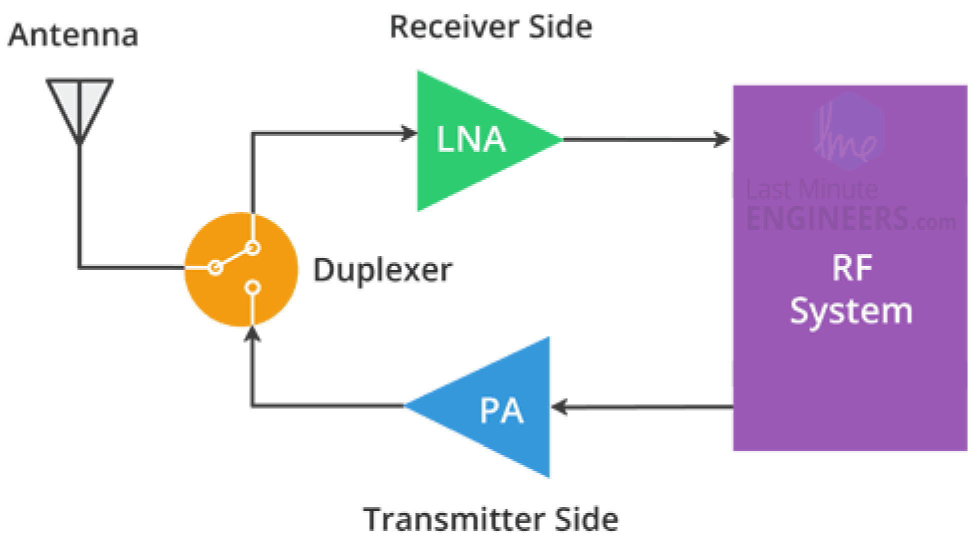


Fig. 14 PA LNA block diagram

CHAPTER-7

7. SPECIFICATION

a. SOLAR PANEL:

- **Efficiency 10.03%**
- Low - Light Performance
- High Load Resistant
- Dimension-290x190x17 mm

b. TP4056 CHARGE CONTROLLER:

• Protection IC	DW01A
• Charge/Discharge Control MOSFET	FS8205A
• **Input Supply Voltage**	**4.5~6.0 V**
• **Constant Charge Current**	**1 A**
Overcharge Protection	
• Overcharge Detection Voltage	4.3 V ±50 mV
• Overcharge Release Voltage	4.1 V ±50 mV
Over-Discharge Protection	
• Over-Discharge Detection Voltage	2.4 V ±100 mV
• Over-Discharge Release Voltage	3.0 V ±100 mV
Overcurrent Protection	
• Overcurrent Protection Threshold	3 A
• Overcurrent Cutout Delay	10~20 ms
• Short-Circuit Cutout Delay	5~50 µs
Trickle Charge (Battery Reconditioning)	
• Trickle Charge Threshold Voltage	2.9 V ±0.1 V
• Trickle Charge Current	130 mA ±10 mA

c. **Li-ion BATTERY:**

- **Li-ion 2200mAh**
- Higher Discharge Current
- Weight 41 grams
- Maximum safe discharge current 4400mA (2C)
- **Maximum charging voltage 4.2V**
- Maximum charging current 4200mA

d. **nRF24L01 TRANSCEIVER:**

- **2.4GHz RF transceiver Module**
- **Operating Voltage: 3.3V**
- Nominal current: 50mA
- Range: 50 – 200 feet
- Operating current: 250mA (maximum)
- **Protocol: SPI**
- Baud Rate: 250 kbps - 2 Mbps
- Channel Range: 125 (1 MHz each)

e. **DHT22 SENSOR:**

- **Operating Voltage: 3.5V to 5.5V.**
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data.
- Temperature Range: -40°C to 80°C.
- Humidity Range: 0% to 100%
- Resolution: Temperature and Humidity both are 16-bit.
- Accuracy: ±0.5°C and ±1%

CHAPTER-8

8. PROJECT DESIGN

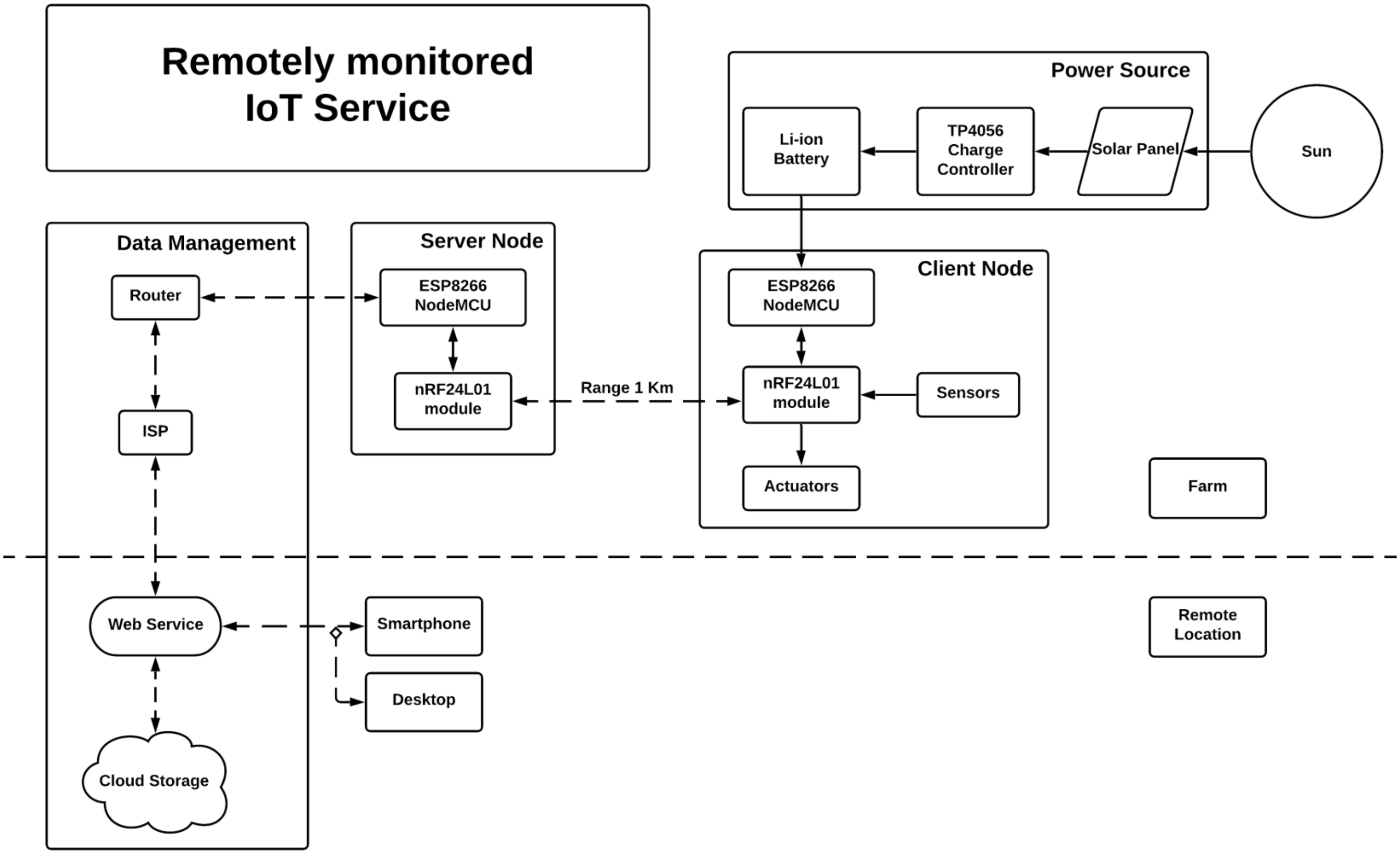


Fig. 15 Remotely monitored IoT service block diagram

This block diagram perfectly describes how the individual modules of the system come together to form a working sophisticated system. The Power Source module ensures continuous power monitoring and recharging of the Li-ion battery throughout the year with the help of TP4056 charge controller. This charges the battery at constant voltage/current to make sure the ESP module at the client end doesn’t stop working. The sensors attached at this client node publishes data server at every pre-defined amount of time as shown in Fig. 16. This data is being published at the MQTT broker which uploads it to its cloud service. This data can be accessed by the user through his smartphone/desktop for referring.

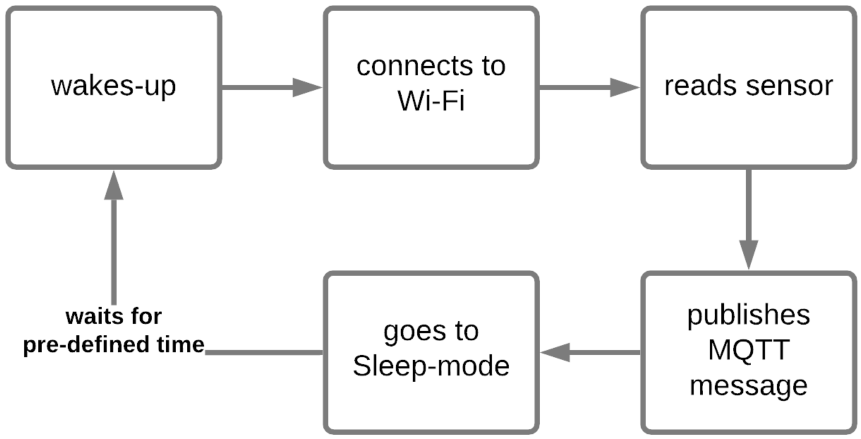


Fig. 16 ESP8266 deep-sleep sketch

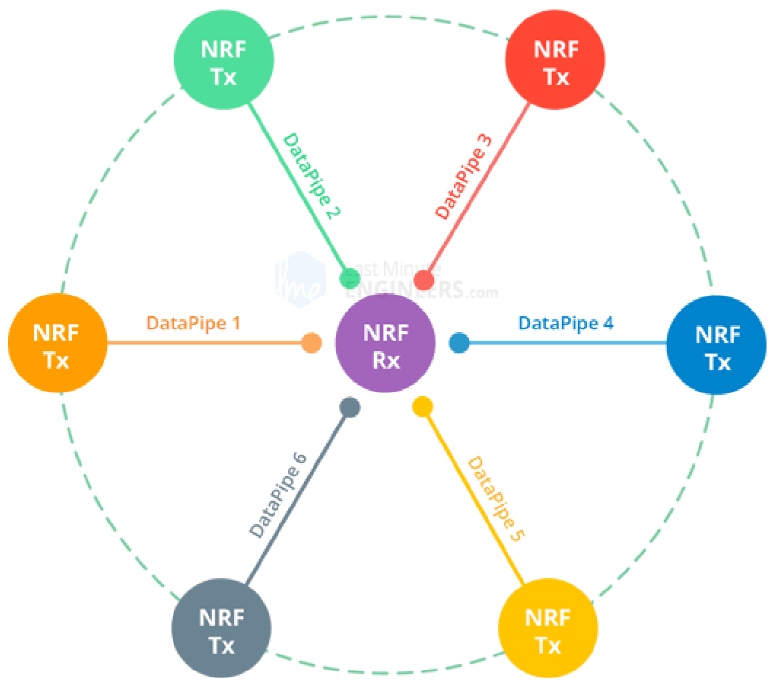


Fig. 17 Project Client-Server network distribution

The nRF2401L provides a feature of Multiple Transmitters Single Receiver. In which each RF channel is logically divided into parallel data channels called Data Pipes. A data pipe is a logical channel in the physical RF Channel. Each data pipe has its own physical address (Data Pipe Address). These can be configured individually into data rates, bandwidths and frequency. The primary receiver collects information from different transmitter nodes simultaneously.

CHAPTER-9

9. PROJECT PLANNING

a. SEMESTER-7:

Sr. No.	TOPIC	DUE DATE (2020)
1	Energy Calculations	18/09
2	Battery Runtime	18/09
3	Voltage Regulation	25/09
4	Cloud Interface	02/10
5	Wireless Data Transfer	09/10
6	Ubidots Dashboard	16/10
7	Graphical Data	23/10
8	Conditional Event	30/10

b. SEMESTER-8:

Sr. No.	TOPIC	DUE DATE (2021)
1	Power Module Testing	29/01
2	Network Simulation	05/02
3	Data Transfer (nRF - Cloud)	12/02
4	Data Hopping	26/02
5	Distance Algorithm	12/03
6	Data Manipulation	26/03
7	Final Testing (Power + Client + Server)	09/04
8	Debugging	23/04

CHAPTER-10

10. EXPECTED OUTCOME

- **a.** To run the system without solar panel for continuous 1-week
- **b.** To be able to build standalone Client Node for indefinite period of time
- **c.** To transmit bi-directional data up to 1 Km of range from one point to another
- **d.** To hop data from different data points using shortest distance algorithm
- **e.** To set-up conditional events as per need and push alerts
- **f.** To control system environment from indefinite distance
- **g.** Reduce system power consumption using deep-sleep mode

CHAPTER-11

11. DELIVERABLES

- **a.** Controlling distant environment from smartphone at different locations
- **b.** Publish data to ubidots from temperature sensor
- **c.** To develop graphical representation of that data
- **d.** Set-up conditional events and trigger email alerts
- **e.** Power budgeting calculations
- **f.** Transfer data between nRF modules up to 700 m apart
- **g.** Switching ESP8266 from-and-to deep-sleep mode

CHAPTER-12

12. CONCLUSION

This system has a very wide future scope where it can be deployed in any IoT system. One such application could be affordably deploying this system into Farm/Barren Lands for Weather monitoring/sensing along with sharing wireless information through Wi-Fi/RF in a mesh network and uploading it to the cloud so that any human can visualize data or get notified of system faults if needed over his/her Smartphone/Website. And if deployed on large scale it can be affordable for multiple users.

Off-grid Power System is very crucial in areas of hard to reach or where frequent maintenance isn't possible. This system is carefully crafted to suit months of runtime without any human interference. The flow of system designing started with the Thermal layout of the country or region where the system would be laid. This helped in generalizing the Solar Panel Size and its specifications. Further to which the battery selection process continued which matches up to the system requirements. And considering all the offsets and weather conditions, we can call this as one of the most robust Off-grip powered or Solar powered system for any type of work.

CHAPTER-13

13. REFERENCES

[1] Martínez, Alejandro, Esteban Cañibano, and Javier Romo. "Analysis of Low Cost Communication Technologies for V2I Applications." Applied Sciences 10, no. 4 (2020): 1249.

[2] Rout, Kshirod Kumar, Samuchita Mallick, and Sivkumar Mishra. "Solar powered smart irrigation system using Internet of Things." In 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), pp. 144-149. IEEE, 2018.

[3] Mahalakshmi, M., S. Priyanka, S. P. Rajaram, and R. Rajapriya. "Distant Monitoring and Controlling of Solar Driven Irrigation System through IoT." In 2018 National power Engineering conference (NPEC), pp. 1-5. IEEE, 2018.

[4] Barai, Suvankar, Debajyoti Biswas, and Buddhadeb Sau. "Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique." In 2017 IEEE Conference on Antenna Measurements & Applications (CAMA), pp. 170-173. IEEE, 2017.

[5] Saha, Saraswati, and Anupam Majumdar. "Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system." In 2017 Devices for Integrated Circuit (DevIC), pp. 307-310. IEEE, 2017.

[6] Singh, Pushkar, and Sanghamitra Saikia. "Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module." In 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), pp. 1-4. IEEE, 2016.

[7] Schwartz, Marco. Internet of Things with ESP8266. Packt Publishing Ltd, 2016.

[8] Wang, Yonghui, Chao Hu, Zhongqing Feng, and Yupeng Ren. "Wireless transmission module comparison." In 2014 IEEE International Conference on Information and Automation (ICIA), pp. 902-907. IEEE, 2014.

[9] Piyare, Rajeev, and Seong Ro Lee. "Towards internet of things (iots): Integration of wireless sensor network to cloud services for data collection and sharing." arXiv preprint arXiv:1310.2095 (2013).

[10] Johnson, David B., David A. Maltz, and Josh Broch. "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks." Ad hoc networking 5, no. 1 (2001): 139-172.