# LoRa BASED SMART METER FOR RURAL AREAS WITHOUT INTERNET

# TABLE OF CONTENTS

**TITLE** **PAGE NO.**

# ABSTRACT

This project is all about creating a simple and effective way to transmit the meter reading to the electricity board automatically by using IOT without any intervention of humans. The project uses LoRa transceiver to transmit the data from each house to the LoRa gateway from which the data is transmitted through the internet to the electricity board server. This project is more cost effective than any of the competitive projects on the same field. By automation of these energy meters man power is saved in very large scale. By this project false reading can be avoided. The LoRa transceivers used in this project has a very long range and designed for transmitting small but essential data through long distance. The smart meter consist of its own voltage and current sensors which will provide essential data to calculate the power consumed by the consumer. This power data is transmitted in real time to a LoRa gateway from which data is transmitted to electricity board through internet. The main advantage of this project is the data is transmitted in real time with lowest cost. The infrastructure developed for this smart meter will open up opportunities for much wider industries to start. The project can provide the infrastructure needed for government to monitor electricity theft in future updates. In future updates using the same infrastructure, location systems can be built as alternative for GPS in cities where these smart meters are installed. This project is sponsored by IEDC

# LIST OF FIGURES

# LIST OF SYMBOLS

| SYMBOLS | EXPLANATION |
|---|---|
| V | Voltage |
| I | Ampere |
| W | power |

# CHAPTER 1

# INTRODUCTION

## 1.1 CHAPTER INTRODUCTION

This chapter gives a brief introduction about the LoRa technology used in this project. This chapter explains about the main aim of the project and also discuses about use of smart energy meters. Though this chapter a brief explanation about the working of the LoRa based smart energy meter is explained. In this chapter [1.3] literature survey displays few IEEE reference papers. These reference papers provided required knowledge to build this project "LoRa based smart meter for rural areas without internet"

## 1.2 INTRODUCTION TO LoRa BASED SMART METER FOR RURAL AREAS WITHOUT INTERNET

The LoRa based smart meter is all about creating a simple and effective way to transmit the meter reading from household energy meters to the electricity board automatically by using IOT without any intervention of humans. The project uses LoRa transceiver to transmit the data from each house to a LoRa gateway from which the data is transmitted through internet to the electricity board server. This project is more cost effective than any of the competitive projects on the same field. By automation of these energy meters man power is saved in a very large scale. By this project false readings can be avoided. The LoRa transceivers used in this project has a very long range and designed for transmitting small but essential data through long distance.

The smart meter consists of its own voltage and current sensors which will provide essential data to calculate the power consumed by the consumer. This power data is transmitted in real time to a LoRa gateway from which data is transmitted to electricity board server through internet. The main advantage of this project is that the data is transmitted in real time with lowest cost. The project can provide the infrastructure needed for government to monitor electricity theft in future updates. In future, updates using the same infrastructure, location systems can be built as alternative for GPS in cities where these smart meters are installed.

## 1.3 LITERATURE SURVEY

Visalatchi Senthilnathan , Kamal Sandeep Karreddula (2017) declare in their research that Energy theft is a very common problem in countries like India where consumers of energy are increasing consistently as the population increases. Utilities in electricity system are destroying the amounts of revenue each year due to energy theft. The newly designed AMR used for energy measurements reveal the concept and working of new automated power metering system but this increased the Electricity theft forms administrative losses because of not regular interval checkout at the consumer's residence. It is quite impossible to check and solve out theft by going every customer's door to door. In this paper, a new procedure is followed based on MICROCONTROLLER Atmega328P to detect and control the energy meter from power theft and solve it by remotely disconnect and reconnecting the service (line) of a particular consumer. An SMS will be sent automatically to the utility central server through GSM module whenever unauthorized activities detected and a separate message will send back to the microcontroller in order to disconnect the unauthorized supply. A unique method is implemented by interspersed the GSM feature into smart meters with Solid state relay to deal with the non-technical losses, billing difficulties, and voltage fluctuation complication.

Gunawan Wibisono, Gilang Permata Saktiaji, Ihsan Ibrahim (2017) researched Traditional electricity meters read manually, large workloads, major errors and distributed at different installation locations, need to spend a lot of time and labor costs. In real-time, the accuracy and extension of the application cannot be fulfilled. With technological developments, electric meters now can be used both ways, for real-time remote reading, and can be used as prepaid and postpaid, making it smart meter. Currently the smart meter has become an important part of the smart grid system. Therefore, the national power utility company (PLN) Bali region decided to replace the existing meter with smart meter to support Bali Eco Smart Grid. Long Range (LoRa) wide area network (WAN) is the access technology that used for implementing smart meter because it has benefits including can reach long distance, low power, and lower cost. This study aims to analyze the techno economic smart meter 2 ways implementation at Bali region by using techno-economic method and cost-benefit analysis. In addition to analyze technological and economic aspects, the business model and the regulatory aspect analysis are also included. The results show that LoRa WAN is one of the technology options that can be used today, the business model that can be used is build own or implementation is done by itself, build operate transfer (BOT) or outsourcing to third party, and rental model. According the ministry of communication and information regulation number 35/2015, LoRa WAN that operate at frequencies 923-925 MHz can be implemented to support smart meter implementation by Bali PLN.

Tian Xia , Cencen Liu , Xin Zheng , Ming Lei , Shangpeng Wang , Yingchun Wang,(2018) developed The principle of field testing for digital electric energy meters is analyzed, according to which a digital reference energy meter with high accuracy is developed based on high-precision frequency division pulse output technology, algorithm of dot product sum of interpolation re-sampling and high performance DSP. The test method based on the digital reference meter is applied field testing comparing with a set of calibration system. The results of the test prove high accuracy, efficiency and stability.

Win Hlaing ; Somchai Thepphaeng ; Varunyou Nontaboot ; Natthanan Tangsunantham ; Tanayoot Sangsuwan ; Chaiyod Pira (2017) propose that the advent of Internet and computational era, not only opportunity to send and receive data between humans, but also among the devices without human control over it. This is known as Internet of Things (IoTs) concept which can be applied for solving the growing issue of power/energy management. A solution is a cheap and easy to implement and manage energy monitoring system for our daily usage of electric power. In order to overcome the human errors, manual labor and cost reducing in energy consumption with more efficiency for the power management system, in this paper, we focus mainly on IoT's energy monitoring. The proposed design is to implement a very low cost wireless sensor network and protocol for smart energy and web application capable of automatically reading the unit and sending the data automatically for the power users to view their current energy meter reading. By using this system, the users will be aware of the electricity usage in his/her home to reduce the power wastage and cost of consumption. The system consists of a digital energy meter, ESP8266 WiFi module and web applications for management system. The ESP8266 WiFi module will be embedded into the meter and implement the TCP/IP protocol for the communications between the meter and web application. The experimental results show that the proposed system works very well with efficiency, and it is feasible to implement in practical applications for very low cost-build automatic energy meter reading.

## 1.4 SUMMARY

This chapter gives a introduction to LoRa based smart meter for rural areas without internet. It gives a brief knowledge about the project and it helps in understanding the aim of the project. It also provides IEEE paper of reference in 1.3 literature survey. These reference paper helps to improve the understanding about the project and also helps to relate the technologies used in this project to that technologies used in the reference papers.

# CHAPTER 2

## 2. INTEGRATION OF LoRa BASED SMART METER FOR REMOTE VILLAGES

### 2.1 CHAPTER INTRODUCTION

This chapter discusses about integration of LoRa based smart meters in rural areas i.e. remote villages. In this chapter we have discussed about the need for LoRa based smart meters in remote villages in 2.1 need for lora based smart meters in remote villages. The explanation given about the need of this project in this chapter is based on the real time day to day conditions. It speaks about the reality conditions and also proposes solutions to handle that situation. In 2.2 implementation of lora based smart meters in remote villages the ways to implement the lora based smart meter is discussed. The block diagrams were used to explain about the data flow starting from household smart meter to a EB server. In this chapter 2.3 implementation of lora based smart meters in remote villages explains about the innovations made in LoRa gateway technologies in this project. It clearly states the major advantages of the innovation in LoRa gateway technology. Through this chapter the needs and ways to accomplish though needs is discussed. This chapter gives clear knowledge about the implementation and advantages of such implementation of this project.

### 2.2 NEED FOR LORA BASED SMART METERS IN REMOTE VILLAGES

Technologies are for everyone. For people in urban cities and also for people living in rural areas. But as of today, conditions changed as if smart technologies are only applicable for people living in cities and people living in rural areas are meant to live old life style. The main reason behind this, is that urban areas have updated infrastructures like high speed internet connections and lot more infrastructure to make their life easier but in remote villages these infrastructures are not available. This makes developers to concentrate more on projects based on city life style.

One among those projects is smart meter working with internet connections. But these smart meters cannot be installed in remote villages as these villages doesn't have internet connection for every house. LoRa based smart meters take the advantage in this point. These LoRa based smart meters doesn't require internet connection in every house. As these LoRa transceivers have a maximum range up to 10Km the energy meter readings can be directly transmitted from a LoRa node i.e. LoRa based smart energy meter and it can be received by a LoRa receiver in the LoRa gateway.

## 2.3 IMPLEMENTATION OF LORA BASED SMART METERS IN REMOTE VILLAGES.

Each energy meter is provided with a LoRa transceivers. This energy meter setup acts as a LoRa node. Each LoRa node transmits energy meter reading with the unique address of that particular node encrypted inside a string. This string is received by the LoRa gateway and the data is uploaded to a server using internet. In this method its enough to have internet connection only for the gateway alone and there is no need to provide internet connection for each and every houses.

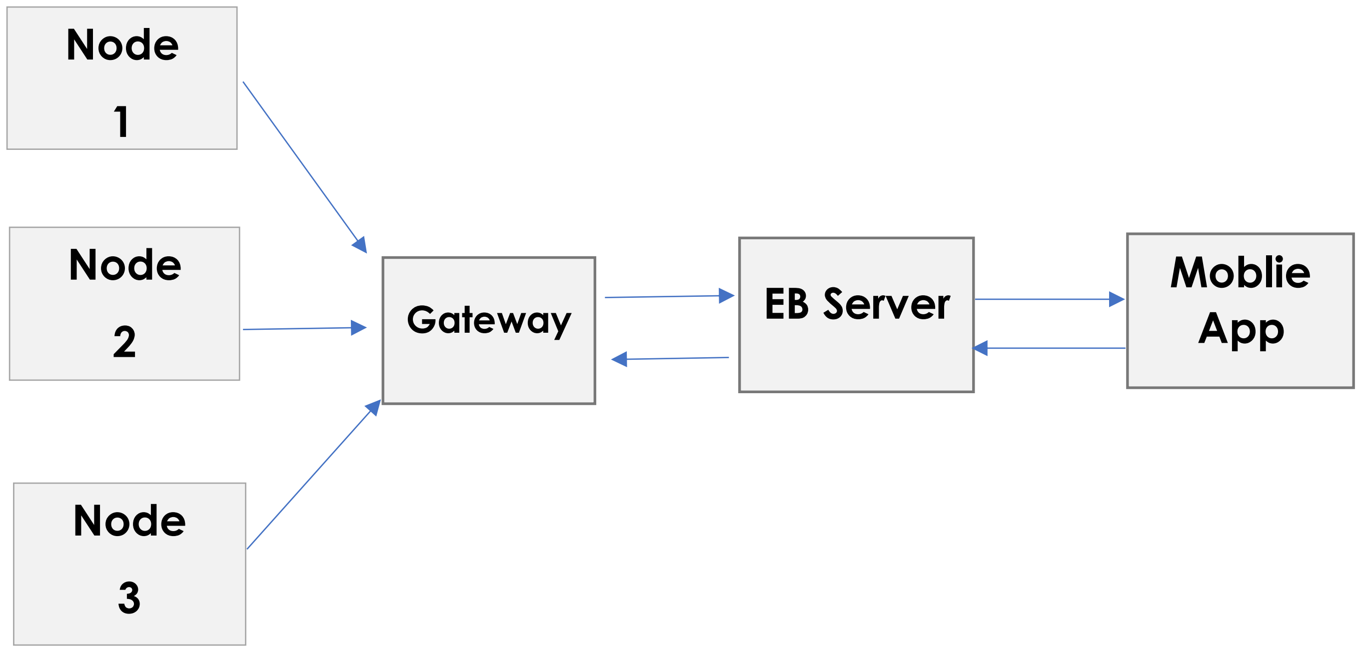


**Fig 2.1 Block diagram of data flow**

Node represents the smart energy meter fixed in every house. These nodes are connected with LoRa gateway, and this gateway is connected to internet. The gateways are programmed to receive energy meter readings from the node and transmit to a EB server. This data can be retrieved from the server and viewed using a mobile application. The mobile application receives the data directly from the server.

## 2.4 LoRa GATEWAY ENHACEMENTS FOR REMOTE VILLAGES

In general LoRa gateways are connected to internet using ethernet cable. But when considering remote villages, even ethernet connections may not be available in worst case scenarios. In this project we have come up with a solution for this. The LoRa gateway that is designed for this project is compatible with both ethernet cables as well as WiFi internet connections. The LoRa gateway is connected to a WiFi module nodeMCU ESP8266 this enables LoRa gateway to get connected with internet just with mobile hotspot connection. In real time usage even a portable WiFi modem can be used to connect the LoRa gateway to internet.

## 2.5 ADVANTAGES OF LoRa BASED SAMRT METERS FOR REMOTE VILLAGES.

Illegal power tapping widely happen in remote villages. These illegal power tapping can be easily identified using this LoRa based smart meter. Every individual house is made accountable for their power usage as the whole process is documented and digitalized. Form technical side, since villages doesn't have tall concrete walls and buildings to weaken the LoRa signal, in these villages lesser number of gateways are enough to establish service when compared to dense urban cities.

## 2.6 SUMMARY

This chapter discussed about the need for LoRa based smart meters in remote villages. It also explained the implementation of LoRa based smart meters in remote villages. The advantages of these smart meters specifically in remote villages is detailly addressed here in this above chapter. This chapter also discuss about the innovation made in LoRa gateway technology. It detailly informs about the two major ways of implementation of LoRa gateway.

# CHAPTER 3

## HARDWARE IMPLEMENTATION OF SMART METER

### 3.1 CHAPTER INTRODUCTION

This chapter discuss about the hardware implementation of smart meter. In this chapter the circuit diagrams used in this project is detailly explained. These circuit diagrams plays a major role in proper execution of this project. This chapter is added with schematic diagrams with its explanations. These schematic diagrams show cast the connections between the microcontrollers and other sensors and transceivers. This would give a clear image about how the components are connected with one another. These connections and explanations also discus about the interface using which the components are connected with one another. In this chapter each and every component used are explained in 3.5 hardware description. These details consist of key features of the components with their specifications. By specifying the specification and key features of the components knowledge about the nature of the components and their uses are explained in this chapter.

## 3.1 CONNECTION OF CT AND VT ACROSS THE LOAD

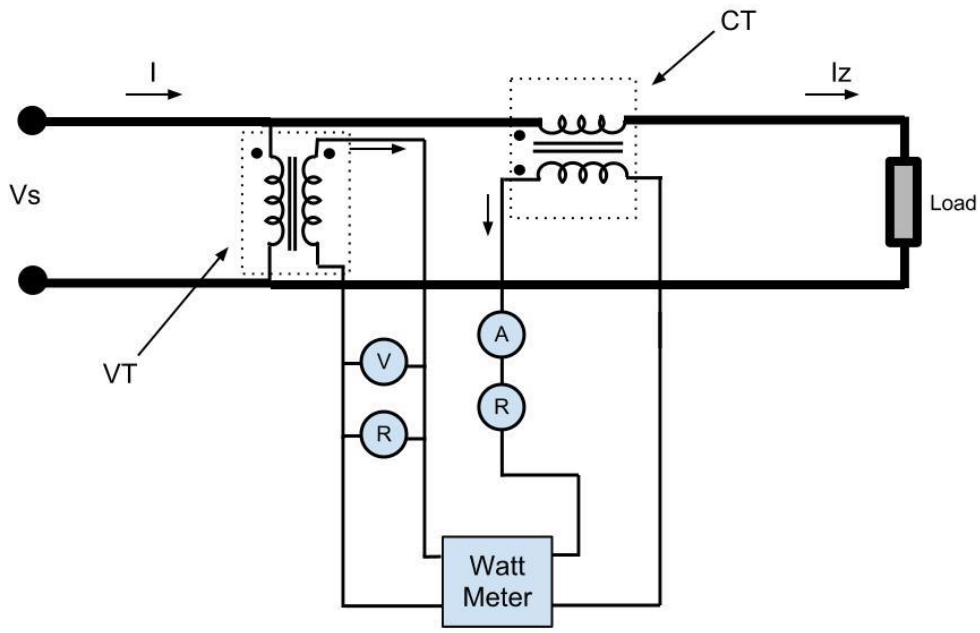


**Fig 3.1 Connection of CT and VT**

For transferring the power data value from each node there is a need to measure the power data initially. For that purpose voltage and current are needed to be measured. It is achieved by connecting the current sensor in series with load and voltage sensor in parallel with load as described in circuit diagram. Here the Arduino Uno acts as the wattmeter, which receives the sensor data from voltage and current sensor and calculate watt and watt per hour data values.

## 3.2 SCHEMATIC DIAGRAM OF SMART ENERGY METER NODE

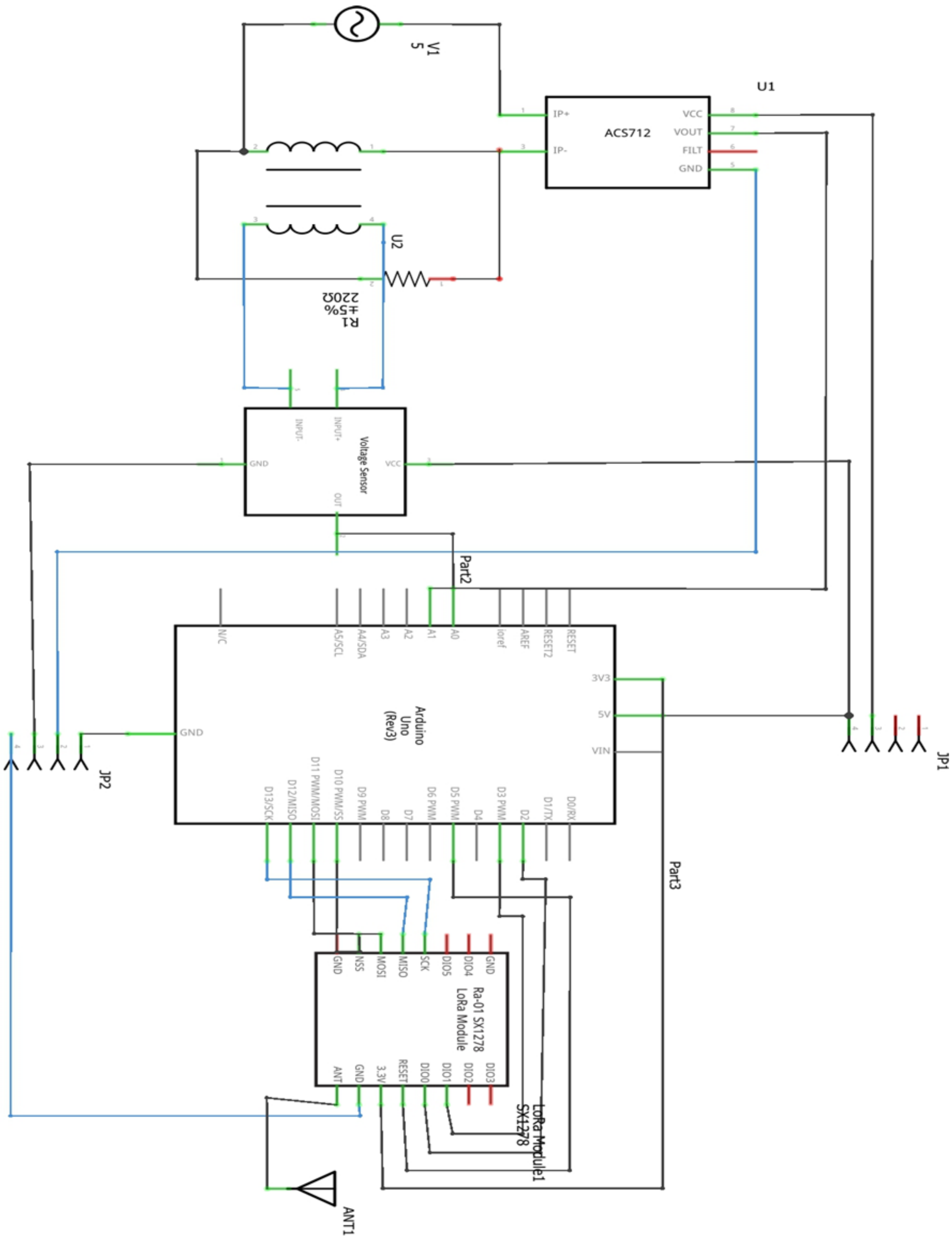

**Fig 3.2 Smart energy meter node**

## EXPLANATION

In this phase energy meter is built and the microcontroller is interfaced with the LoRa module SX1278. The schematic explains the way to connect the current sensor ACS712 and voltage sensor across the load. The voltage and current sensors are powered using +5 volt and Gnd of the Arduino microcontroller. Two junctions are used to utilize the 5V and ground of the Arduino to power multiple sensors connected to it. The voltage sensor is connected to Analog pin A0 of the Arduino microcontroller. Current sensor is connected to analog pin A1 of the Arduino microcontroller. The LoRa module SX1278 is interfaced with Arduino microcontroller using Serial Peripheral Interface (SPI). LoRa module is connected to an antenna of 5dB gain.

## 3.3 SCHEMATIC DIAGRAM OF GATEWAY USING ETHERNET

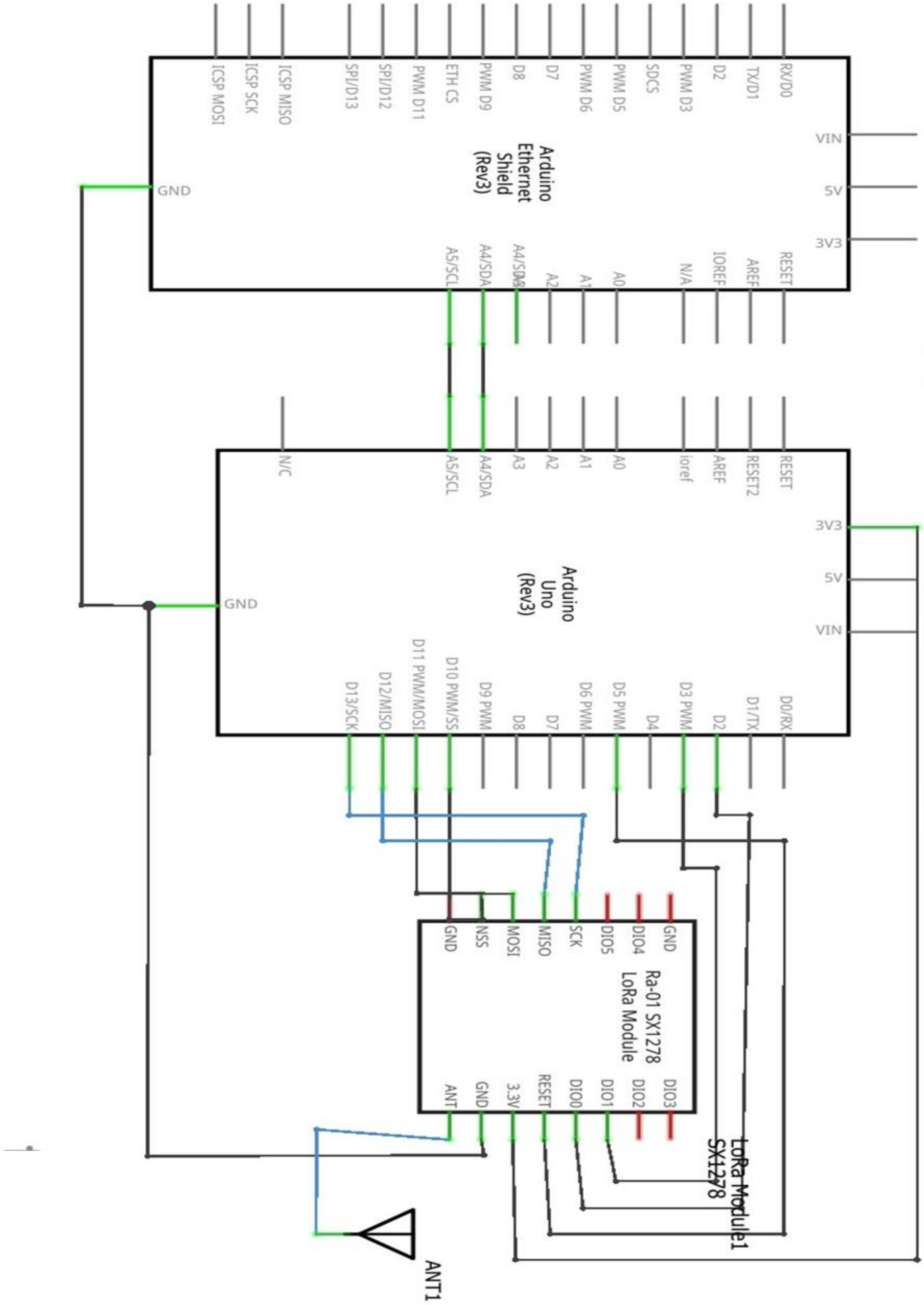

**Fig 3.3 Gateway using ethernet**

## EXPLANATION

The schematic explains how the ethernet shield is connected to LoRa receiver. The LoRa receiver by itself is connected to an Arduino Uno and this Arduino is connected to an Ethernet shield using I2C interface. The LoRa module SX1278 is interfaced with Arduino microcontroller using Serial Peripheral Interface(SPI). LoRa module is connected to an antenna of 5dB gain. The ethernet shield is connected to internet via fibre optic cables. The ethernet shield is grounded through Arduino Uno. The LoRa module is powered by the Arduino microcontroller.

## 3.5 SCHEMATIC DIAGRAM OF GATEWAY USING Wi-Fi

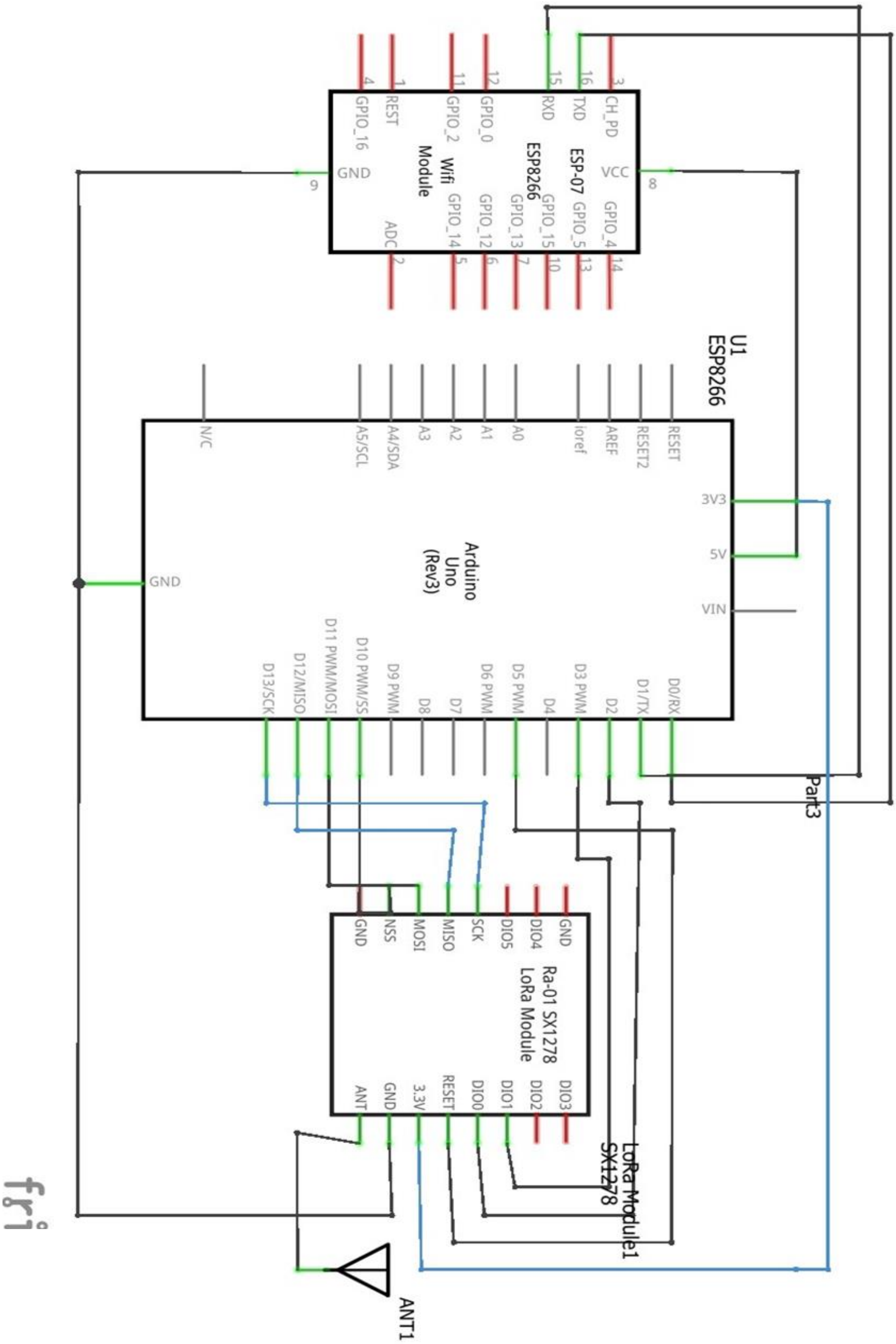


**Fig 3.4 Gateway using wi-fi**

## EXPLANATION

The schematic explains how the node MCU ESP8266 is connected to LoRa receiver. The LoRa receiver by itself is connected to an Arduino Uno and this Arduino is connected to an node MCU ESP8266 using serial Tx and Rx ports. The LoRa module SX1278 is interfaced with Arduino microcontroller using Serial Peripheral Interface (SPI). LoRa module is connected to an antenna of 5dB gain. The node MCU ESP8266 is connected to internet via Wi-Fi. The node MCU ESP8266 is grounded through Arduino Uno. The LoRa module is powered by the Arduino microcontroller.

## 3.6 HARDWARE DESCRIPTION

### 3.6.1 CURRENT SENSOR – ACS712

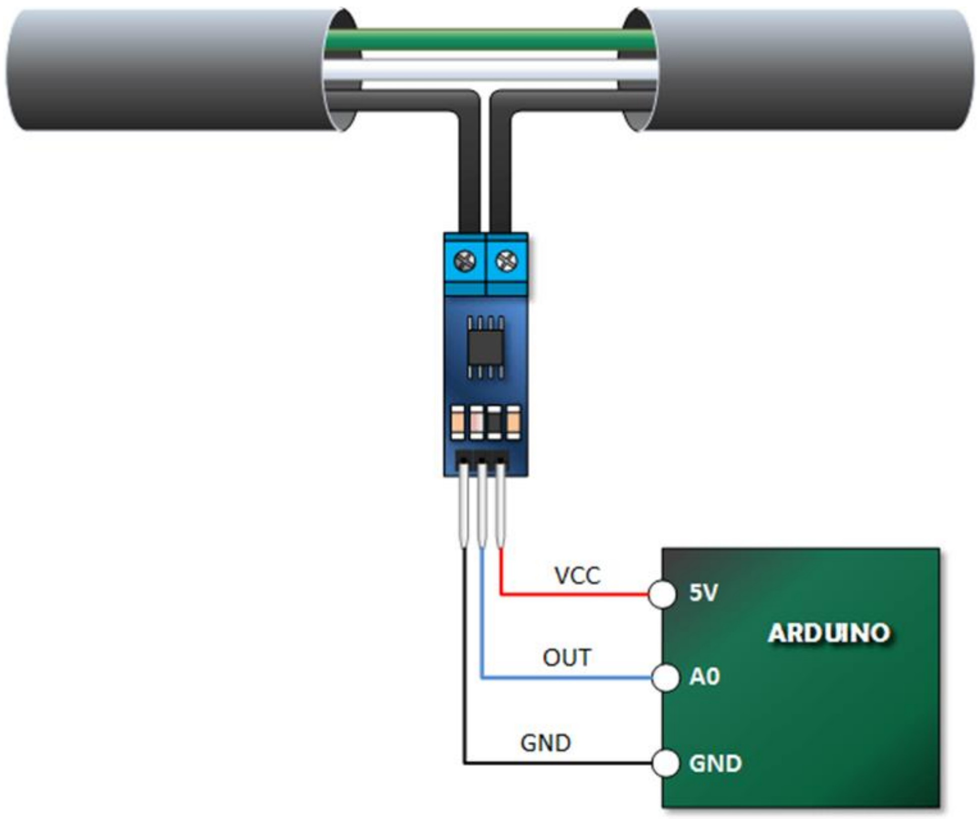


**Fig 3.5 Current sensor**

The Allegro® ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, automotive, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection.

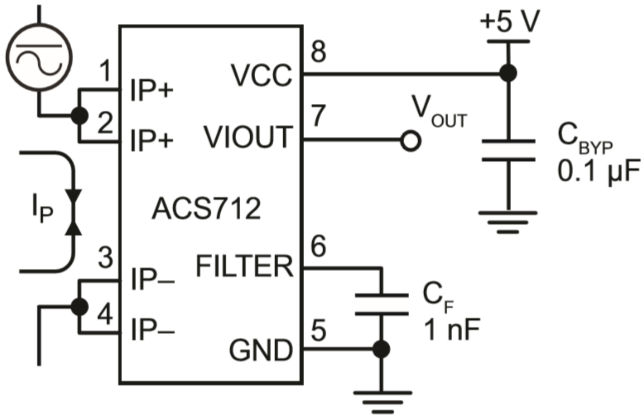


**Fig 3.6 Pin diagram**

## KEY FEATURES

- Low-noise analog signal path
- Total output error 1.5% at TA = 25°C, and 4% at –40°C to 85°C
- 1.2 mΩ internal conductor resistance
- Output voltage proportional to AC or DC currents
  Extremely stable output offset voltage

## 3.6.2 VOLTAGE SENSOR – ZMPT101B

**Fig 3.7 voltage sensor**

ZMPT101B voltage sensor module is a voltage sensor made from the ZMPT101B voltage transformer. It has high accuracy, good consistency for voltage and power measurement and it can measure up to 250V AC. It is simple to use and comes with a multi turn trim potentiometer for adjusting the ADC output. The analysis in this paper tends to find more accurate relationship between the input voltage and the ADC output by regression analysis. The ADC output is adjusted using the trimpot to an appropriate value against a reference input. Figure-1 is the ZMPT101B voltage sensor module.

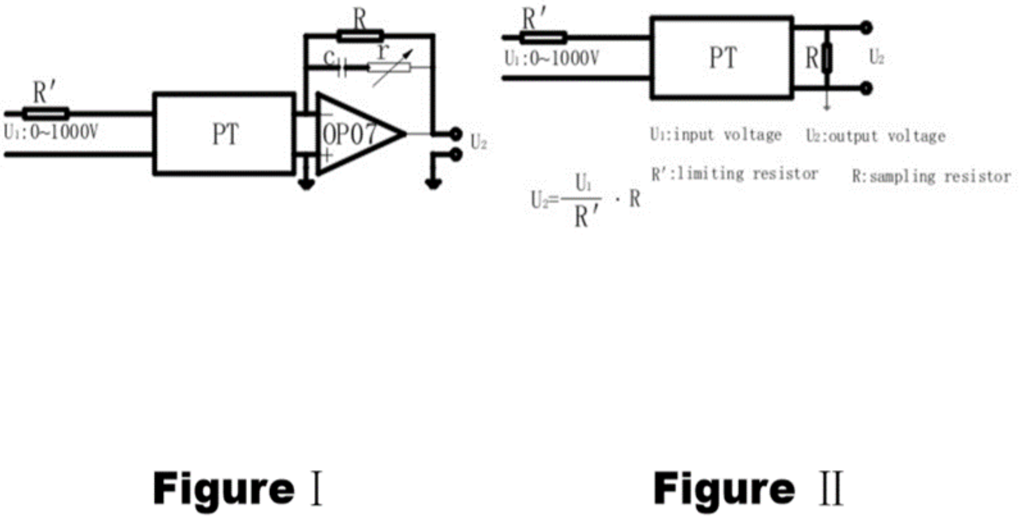


**Fig 3.8 Pin diagram**

**KEY FEATURES**

- Voltage upto 250 volts can be measured
- Light weight with on-board micro-precision voltage transformer
- High precision on-board op-amp circuit
- Operating temperature: 40ºC ~ + 70ºC
- Good consistency, for voltage and power measurement
- Very efficient and accuracy

### 3.6.3 ARDUINO UNO

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

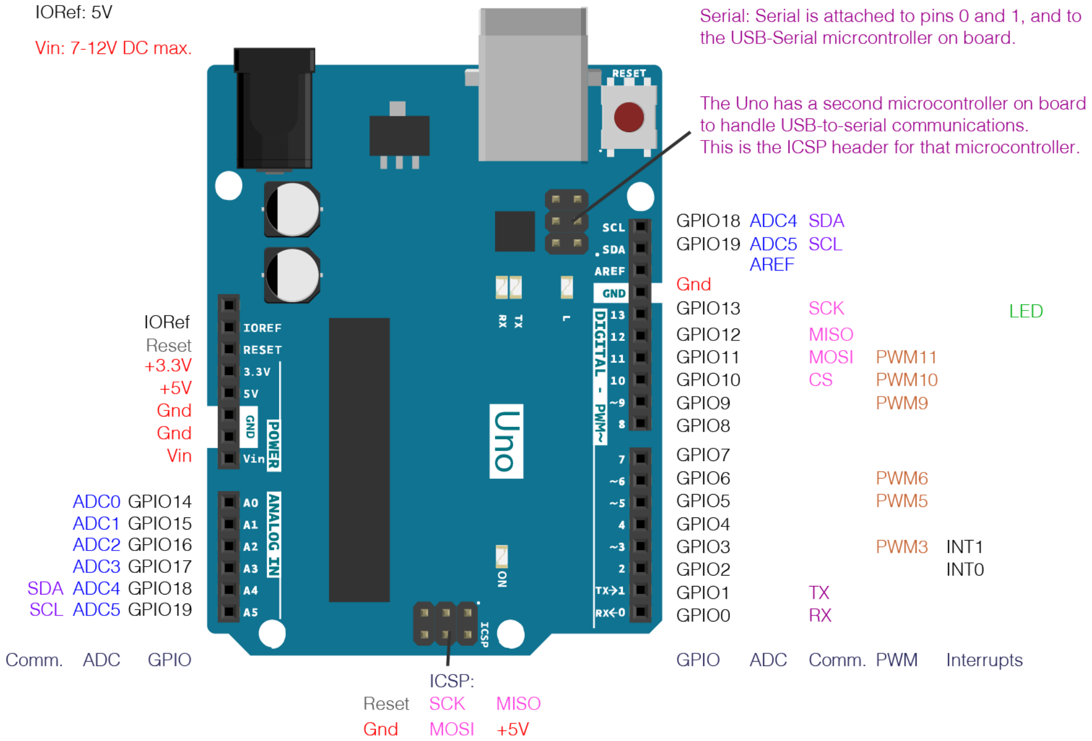


**Fig 3.9 Arduino Uno**

## KEY FEATURES

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6

- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

The **Arduino Uno** is a microcontroller board based on the ATmega328. Arduino is an open-source, prototyping platform and its simplicity makes it ideal for hobbyists to use as well as professionals. The Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

### 3.6.4 ETHERNET SHIELD

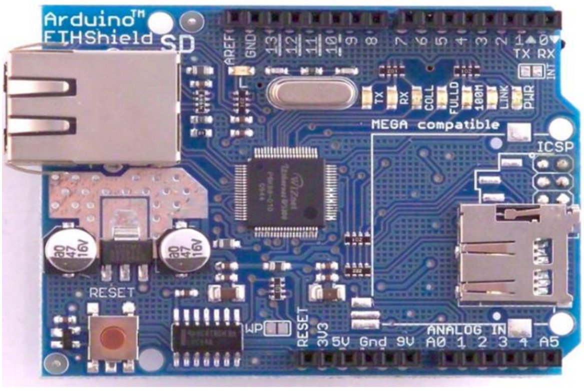

**Fig 3.10 Ethernet shield**

The ethernet shield is used to connect the Arduino Uno with internet using fibre optic cables. It acts as the interface between Arduino Uno and fibre optic cables.

### 3.6.5 NODE MCU ESP8266

The ESP8266 is a low cost MCU with built in Wi-Fi. It can be paired with another host microcontroller, like an Arduino, to provide WiFi networking capability for a basic IoT development platform. Additionally, the ESP8266 can be used as a stand-alone MCU, as it includes a 32-bit 80Mhz processor, 16 GPIO pins (4 PWM enabled) and a built in Analog-to-Digital converter, SPI and I2C interfaces and more… The MCU has a n operating voltage of 2.5V – 3.6V and average operating current of 80 mA. Here is the official ESP8266 MCU datasheet from the Shangai -based manufacturer of the chip Express. The most popular current version is the ESP-12E MCU pictured below. On its own this is not very friendly for prototyping, as the spacing of the pins is not compatible with standard headers and breadboards

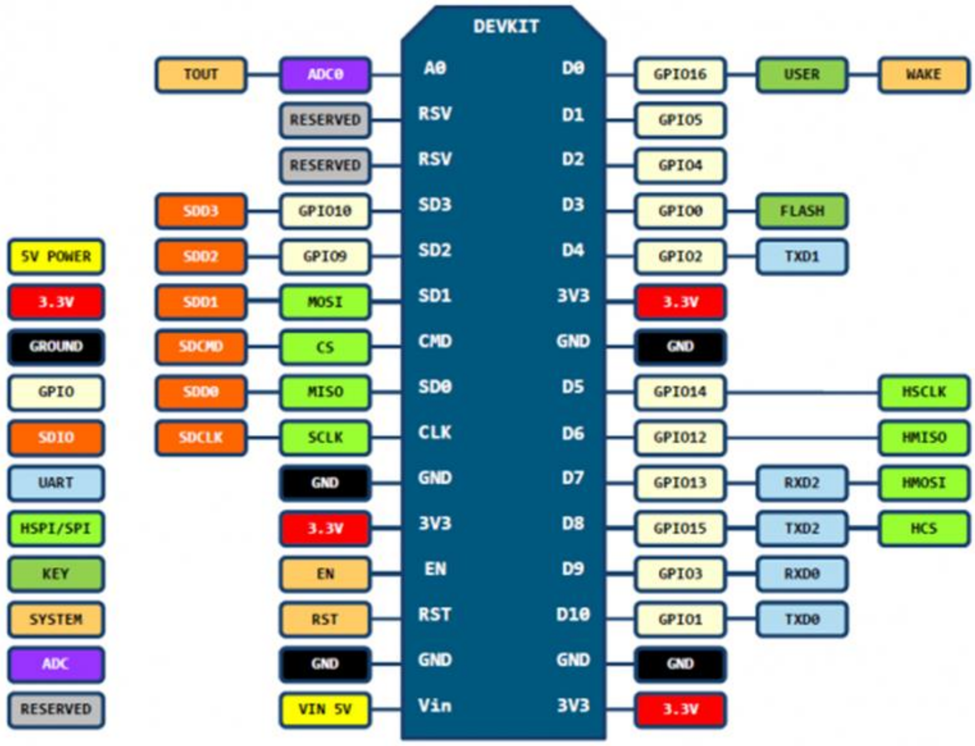


**Fig 3.11 Node MCU**

## 3.7 SUMMARY

This chapter clearly explained about the hardware implementation and circuits used in this project. This chapter consists of schematic diagrams and their explanations. The schematic diagrams show casted in this chapter clearly explained the pin to pin connection between the Arduino microcontroller and other sensors and transceivers. In this chapter the hardware implementations and executions are discussed. This chapter discussed about the hardware specifications of each and every components used in this project. This specifications included the use of these components. It also discussed about the key feature of each and every components. Though this chapter a clear image of hardware implementation and wider awareness about the use of the components and their specifications.

# CHAPTER 4

## SOFTWARE IMPLEMENTATION

### 4.1 CHAPTER INTRODUCTION

When the hardware components and their connections between them describes physical portion of the project, the softwares used and code written in the microcontroller plays the digital part of the project. This software portion acts as important part of the project. The software codeing written in microcontroller is what makes the functions to be performed. To transmit or receive data using LoRa as well as calculating power all of this is carried out only by programming the microcontroller respectively.

The software used to code the microcontroller is Arduino IDE. It uses 'C++' programming language to program the microcontroller. This chapter consists flowcharts, algorithms and as well as programs. The flowchart is the diagrammatic representation of the flow of the program. This chapter consists of 4 flowcharts which represents 4 different programs used in the project. Algorithm are included in this chapter. These algorithms are used to explain the flowcharts. Thus for the four flowcharts included in the project four respective algorithms are included to explain these flowcharts.

The 'C++' programs are included in this chapter. As the programs play a very important role in the working of the project it is included. The programs are written with comment by the side of every important lines. These comment lines are used for understanding the program

## 4.2 SMART ENERGY METER NODE

### 4.2.1 FLOWCHART

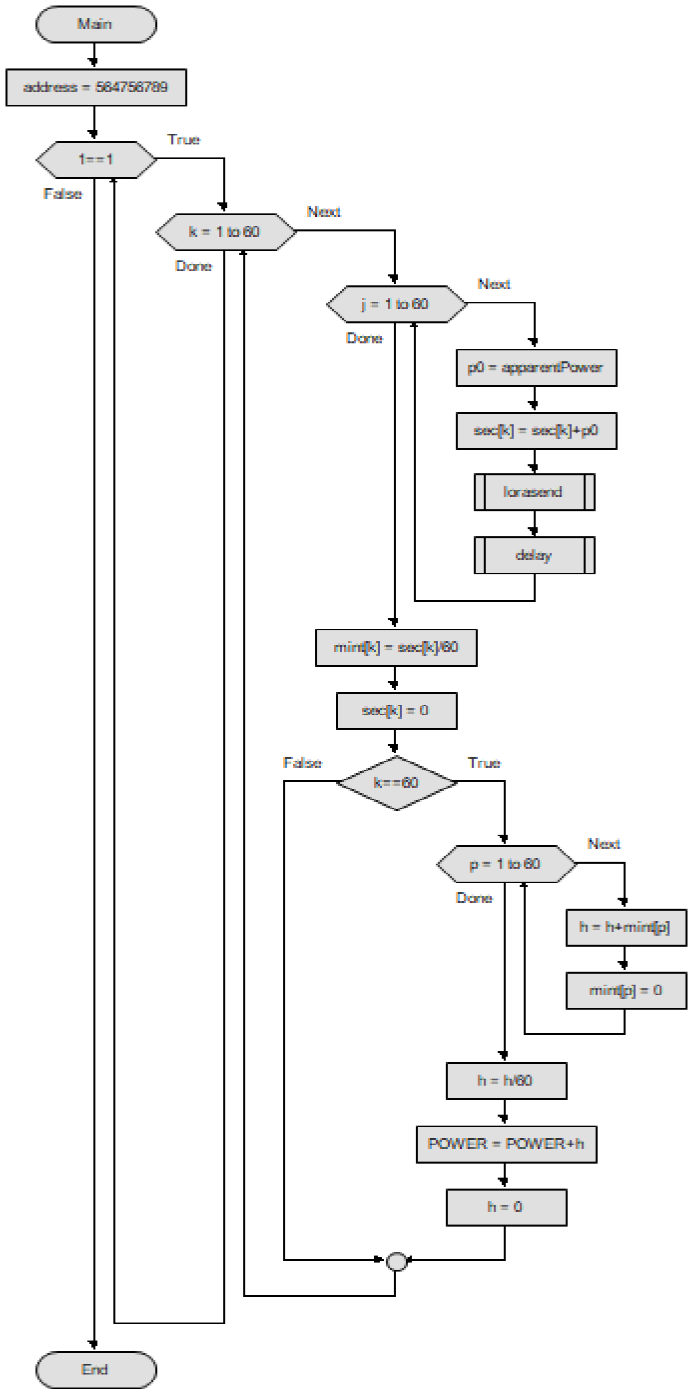


**Fig 4.1 Flowchart of smart energy meter node**

### 4.2.2 ALGORITHM

Step 1: Start.

Step 2: Create variables.

Step 3: Start of while loop.

Step 4: Start a for loop ‘k’ that increment every one minute for one hour and Restart every hour.

step 5: Start a for loop ‘j’ that increment every one second for one minute and restarts every minute.

Step 6: Calculates power value for that particular instance.

Step 7: Sum up the power values for one minute and store in a variable.

Step 8: Transmit the power value wirelessly through LoRa transceiver.

Step 9: Make delay for few milliseconds.

Step 10: Find the power value average and store it in a variable.

Step 11: If the for loop ‘k’ value is equal to ‘60’ continue else jump to step 4.

Step 12: Calculate watt/ hour value once an hour.

Step 13: Go to step 4.

Step 15: When for loop k terminates the command goes to while loop which is infinite loop and program starts to run from step 4 again.

## 4.2.3 PROGRAM

```
#include <SPI.h> // 34rduino spi library

#include <LoRa.h> // 34rduino libraries ra02 lora

 #include “EmonLib.h”            // Include Emon Library

EnergyMonitor emon1;            // Create an instance

float v=0;

float i=0;
```

```
float POWER=0,p0,h,sec[100],mint[100];
int counter=0,f=0;
char address[10]=”564756789”;
long start,esp;
void setup() {
   Serial.begin(9600);
 emon1.voltage(0, 380, 1.7);  // Voltage: input pin, calibration, phase_shift
  emon1.current(1, 10.8);       // Current: input pin, calibration.
  While (!Serial);
  Serial.println(“LoRa Sender”);
  if (!LoRa.begin(433E6)) {
    Serial.println(“Starting LoRa failed!”);
   while (1);
  }
 LoRa.setSpreadingFactor(10);
 LoRa.setSignalBandwidth(62.5E3);
 LoRa.crc();// put your setup code here, to run once
}
void loop() {
 for(int k=1;k<=60;k++) // This loop increments every one minute
  {
  for(int j=1;j<=60;j++)  // This loop increments every second
  {
    emon1.calcVI(20,2000);   //POWER CALCULATION
  float apparentPower   = emon1.apparentPower;  // deriving power value
  p0=apparentPower;
```

```
 if(p0<8)
 p0=0;
  sec[k]=sec[k]+p0; //sum up power value of every second for one minute

Serial.print("Sending packet:");// used for displaying the values
 Serial.println(counter);
 Serial.println("power:");
 Serial.println(p0);
  LoRa.beginPacket();  //LORA  transnmission start
  LoRa.print(POWER);
 LoRa.print('$');
 LoRa.print(address);
 LoRa.print('=');
 LoRa.print(p0);
  LoRa.endPacket(); //LORA transmission end
  counter++;
  if(f==0)
  {
 delay(601); //delay for 601 millisecond
  }
  else
  {
    delay(592); // delay for 592 millisecond

    f=0;
  }
```

```
 }
 mint[k]=sec[k]/60; //takes average of power of one minute,stores in mint[k]
sec[k]=0; // set "sec[k]" to zero to make it empty for next loop cycle
 if(k==60) // "k =60" for every one hour
  {
   for(int p=1;p<=60;p++) // this loop runs once an hour
   {
     h=h+mint[p]; //sum up the all minute values in "h"
     mint[p]=0; // "mint[p]" is emptied for next loop cycle
   }
   h=h/60; // "h" is stroed with power average for one hour i.e. watt/hour
   POWER=POWER+h; //unit value
   Serial.println(POWER); //displaying power
   Serial.println("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx");
  h=0;// assigning "h" as zero
  }
  }
}
```

## 4.3 LoRa RECEIVER

### 4.3.1 FLOWCHART

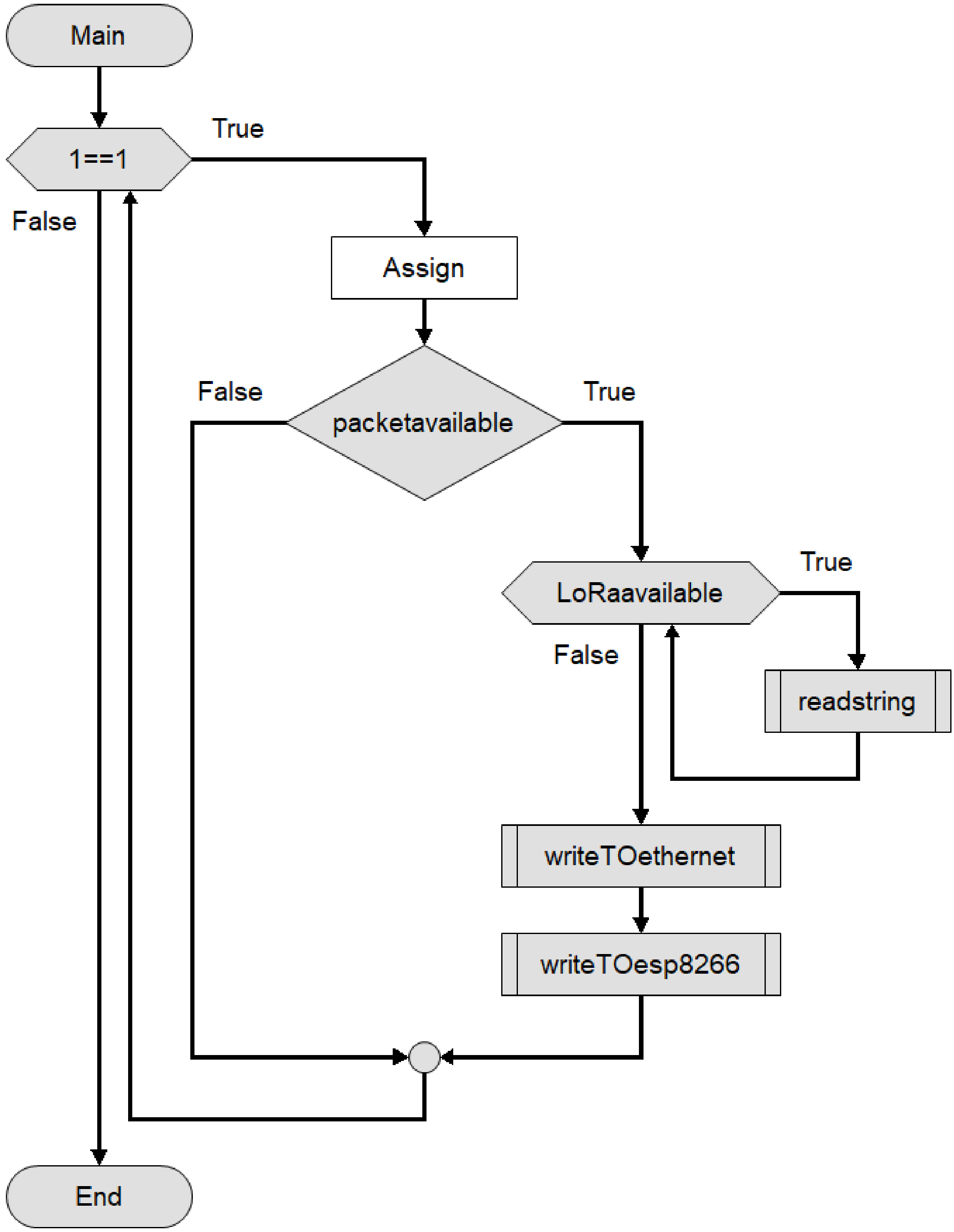


**Fig 4.2 Flowchart of LoRa receiver**

### 4.3.2 ALGORITHM

Step 1: Start.

Step 2: Start a while loop.

Step 3: Sariables are assigned.

Step 4: If LoRa signal is available the program continue else the program goes to step 2.

Step 5: If a LoRa string in received the program continues else goes to step 2.

Step 6: Read the received string and save it in a variable.

Step 7: Write the saved variable to ethernet shield using I2C interface.

Step 8: Write the saved variable to ESP8266 using Tx & Rx serial ports.

Step 9: Goes to step 2 since the while loop used is a infinite loop.

## 4.3.3 PROGRAM

```
#include <SPI.h> // 39rduino spi library
 #include <Wire.h>
#include <LoRa.h> // 39rduino libraries for ra02 lora
 char  val,l,ma,mystr,energy[20],address[20],watt[20];
char fullstr[30]=”123954x486y184u”;
 int j1=0,j2=0,j3=0,i=0,x,t=0,f1=0,f2=0,p=0;


void setup() {
      Serial.begin(9600);
      Wire.begin(8);              // join i2c bus with address #8
      while (!Serial);
     Serial.println(“Lora Receiver”);
```

```
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed ");
  }
 LoRa.setSpreadingFactor(10);
 LoRa.setSignalBandwidth(62.5E3);
 LoRa.crc();
}

void loop()
{
int packetSize = LoRa.parsePacket();
if (packetSize) {          // read packet
int i=0;
t=0;
j1=0;
j2=0;
j3=0;
f1=0;
f2=0;
while (LoRa.available()) //finds signal available or not
 {
mystr=(char)LoRa.read();// read data character by character
fullstr[i++]=mystr; //store the character as string in "fullstr"
if(mystr=='$')
{
f1=1;
continue;
}
```

```
else if(mystr=='=')
{
f2=1;
continue;
}
if(f1==0&&f2==0) //splitting "energy"
energy[j1++]=mystr;
else if(f2==1&&f1==1) //splitting "watt"
watt[j2++]=mystr;
else //splitting "address"
address[j3++]=mystr;
}
x=strlen(energy);
Serial.println("");
Serial.print("ADDRESS:");
Serial.println(address);
Serial.print("ENERGY:");
Serial.println(energy);
Serial.print("WATT/HR:");
Serial.println(watt);
Wire.beginTransmission(8); // transmit to device #8 to ethernet
Wire.write(fullstr);
Wire.write(" ");
Wire.endTransmission();    // stop transmitting to ethernet
Serial.write(fullstr,30); // transmitting  to ESP8266
}}
```

## 4.4 GATEWAY USING ETHERNET

### 4.4.1 FLOWCHART

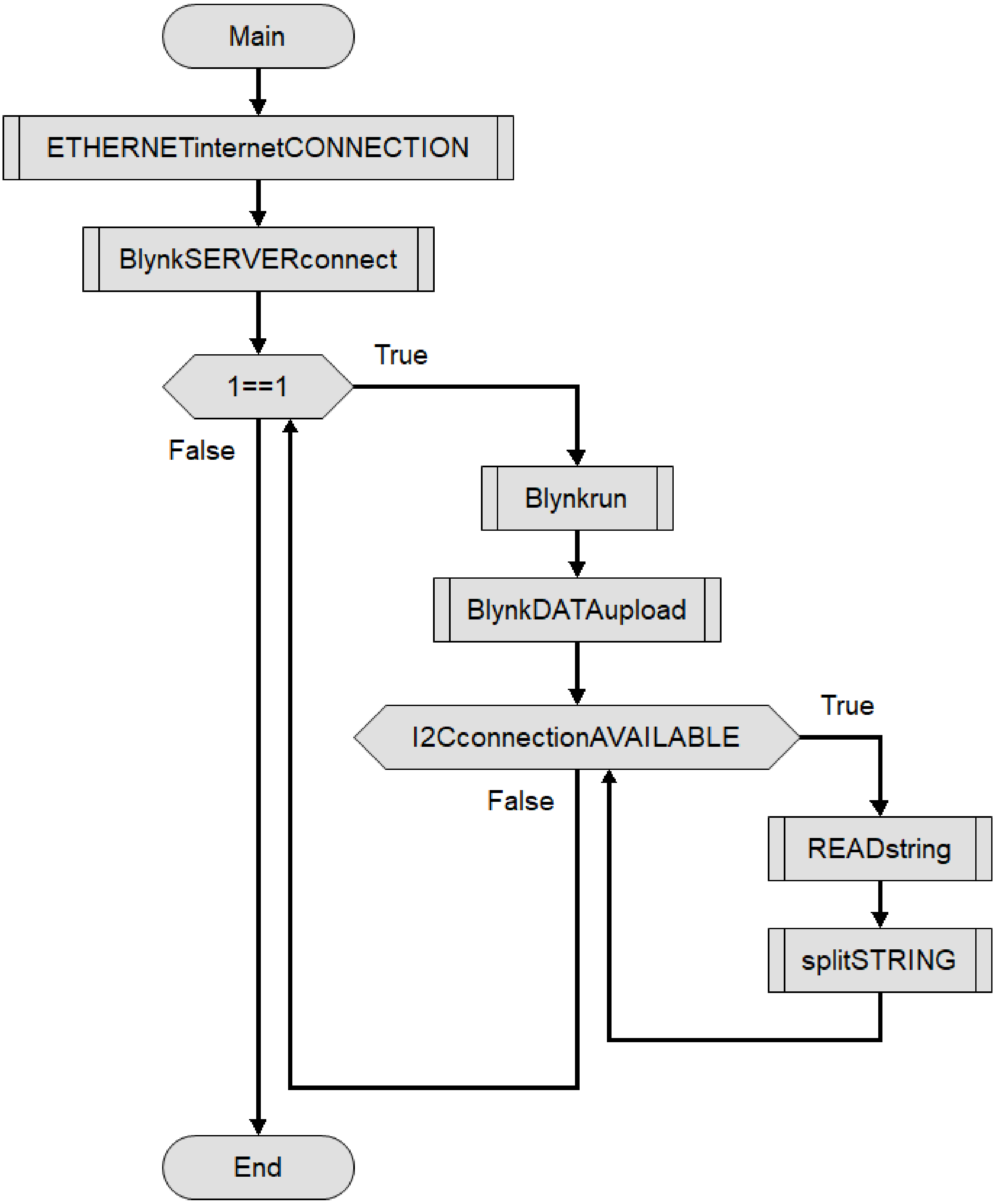


**Fig 4.3 Flowchart of gateway using ethernet**

### 4.4.2 ALGORITHM

Step 1: Start.

Step 2: Connects to internet through ethernet shield.

Step 3: Connect to Blynk server.

Step 4: Start a while loop.

Step 5: Run the blynk library.

Step 6: Upload the variable to blynk server.

Step 7: If connection available I2C interface program continues

else go to step 4

Step 8: Read the variable from I2C connection and save it in a variable.

Step 9: Split the received string.

Step 10: Goes to step 4 and the while loop run again and again since it is a

Infinite loop.

### 4.4.3 PROGRAM

```
#define BLYNK_PRINT Serial
#include <Wire.h>
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
Char auth[] = “6ec4b4c46e77464f8e0c2a9908e95993”;

#define W5100_CS  10
```

```
#define SDCARD_CS 4
    // DHT 11
//#define DHTTYPE DHT22   // DHT 22, AM2302, AM2321
//#define DHTTYPE DHT21   // DHT 21, AM2301

BlynkTimer timer; // creating object for blynk
char mystr[20],energy[20],address[20],watt[20];

 int j1=0,j2=0,j3=0,i=0,x,t=0,f1=0,f2=0,p=0;
// This function sends Arduino’s up time every second to Virtual Pin
(5).
// In the app, Widget’s reading frequency should be set to PUSH.
This means
// that you define how often to send data to Blynk App.
Void sendSensor()
{
 char val[20]=””;
 for(int j=0;j<i;j++)
 {
 val[j] =mystr[j];
 }
// Serial.println(i);
Serial.println(val);
Blynk.virtualWrite(V1, watt); //uploading to server
 Blynk.virtualWrite(V0, energy); //uploading to server
  Blynk.virtualWrite(V2, address); //uploading to server
 strcpy(mystr,””);//empty “mystr”
```

```
}
void setup()
{
  // Debug console
  Wire.begin(8);                // join i2c bus with address #8
  Wire.onReceive(receiveEvent); // register event
  Serial.begin(9600);           // start serial for output

  pinMode(SDCARD_CS, OUTPUT);
  digitalWrite(SDCARD_CS, HIGH); // Deselect the SD card

  Blynk.begin(auth);
  // You can also specify server:
  //Blynk.begin(auth, "blynk-cloud.com", 80);
  //Blynk.begin(auth, IPAddress(192,168,1,100), 8080);
  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}
void loop()
{
  Blynk.run();
  timer.run();
 i=0;
 t=0;
 j1=0;
 j2=0;
    j3=0;
    f1=0;
    f2=0;
```

```
 while (1 < Wire.available()) { // loop through all but the last
   char mystr = Wire.read(); // receive byte as a character
    if(mystr=='$')
{
f1=1;
continue;
}
else if(mystr=='=')
{
 f2=1;
 continue; }
//Serial.print(mystr);
if(f1==0&&f2==0) // split "energy" from "mystr"
energy[j1++]=mystr;
else if(f2==1&&f1==1) //split "watt" from "mystr"
watt[j2++]=mystr;
else // split "address" from "mystr"
address[j3++]=mystr;
 }
 int x;    // receive byte as an integer
 x = Wire.read();
  Serial.println(address);
 Serial.println(watt);
 Serial.println(energy);
 delay(500);
}
void receiveEvent(int howMany)
{        }
```

## 4.5 GATEWAY USING NODE MCU ESP8266

### 4.5.1 FLOWCHART

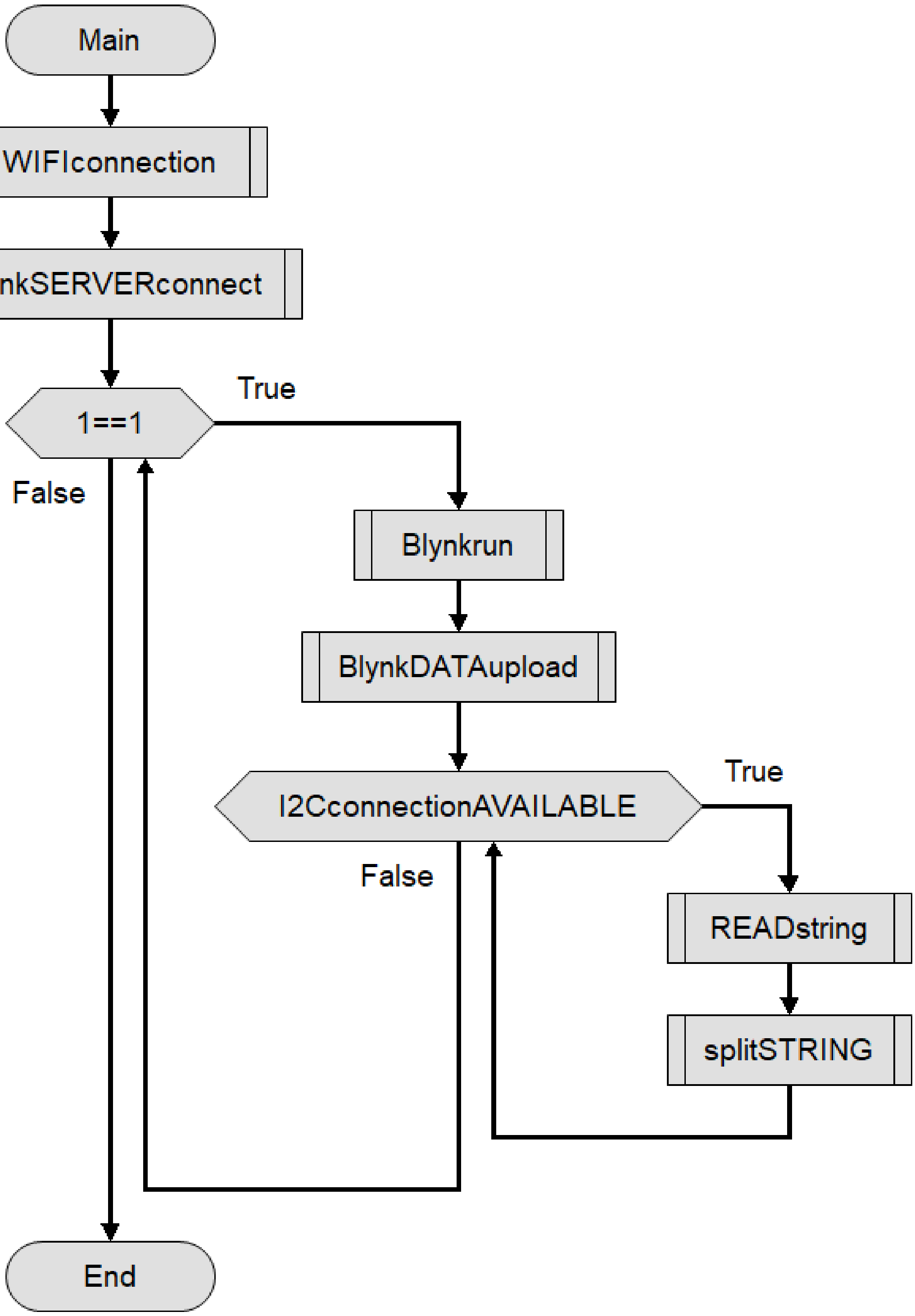


**Fig 4.4 Flowchart of gateway using Node MCU**

### 4.5.2 ALGORITHM

Step 1: Start.

Step 2: Connects to internet through Wi-Fi.

Step 3: Connect to Blynk server.

Step 4: Start a while loop.

Step 5: Run the blynk library.

Step 6: Upload the variable to blynk server.

Step 7: If connection available I2C interface program continues
else go to step 4.

Step 8: Read the variable from I2C connection and save it in a variable.

Step 9: Split the received string.

Step 10: Goes to step 4 and the while loop run again and again since it is a
Infinite loop.

## 4.5.3 PROGRAM

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "9c0f15fc9db146f19808881e9fdc4611";

// Your WiFi credentials.

// Set password to "" for open networks.
```

```
char ssid[] = "lone";

char pass[] = "12345678";

char mystr[20];

BlynkTimer timer;

char energy[20],address[20],watt[20];

 int far,h,po=0,f3=0,j2=0,j3=0,i=0,x,t=0,f1=0,f2=0,p=0;

void sendSensor()

{

 char val[20]="";

 for(int j=0;j<i;j++)

 {

 val[j] =mystr[j];

 }

//Serial.println(i);

 Blynk.virtualWrite(V0, address); //uploading to server

  Blynk.virtualWrite(V1, watt); //uploading to server

    Blynk.virtualWrite(V2,energy ); //uploading to server

 strcpy(mystr,"");//empty mystr

}

void setup()

{

  // Debug console
```

```
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

  timer.setInterval(1000L, sendSensor);

  // You can also specify server:

  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);

  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

}


void loop()

{

  Blynk.run();

  timer.run();

  Serial.readBytes(mystr,30); //Read the serial data and store in “mystr”

 // Serial.println(mystr); //Print data on Serial Monitor

  far=strlen(mystr);

  // Serial.println(far);

    i=0;

    t=0;

 po=0;

 j2=0;

     j3=0;

     f1=0;
```

```
    f2=0;
    f3=0;
  for(int h=0;h<far;h++)
  {
    if(mystr[h]=='$')
 {
 f1=1;
 continue;
 }
 else if(mystr[h]=='=')
 {
  f2=1;
  continue;
 }
if(mystr=="%")
f3=1;
 if(f1==0&&f2==0)
 {
address[po++]=mystr[h];
 }
else if(f1==1&&f2==0)
{
```

```
watt[j3++]=mystr[h];
  }
  else
{
energy[j2++]=mystr[h];
}
  }
Serial.print("ADDRESS:");
  Serial.println(address);
  Serial.print("WATT:");
 Serial.println(watt);
 Serial.print("ENERGY:");
 Serial.println(energy);
}
```

## 4.6 BLYNK MOBILE APP

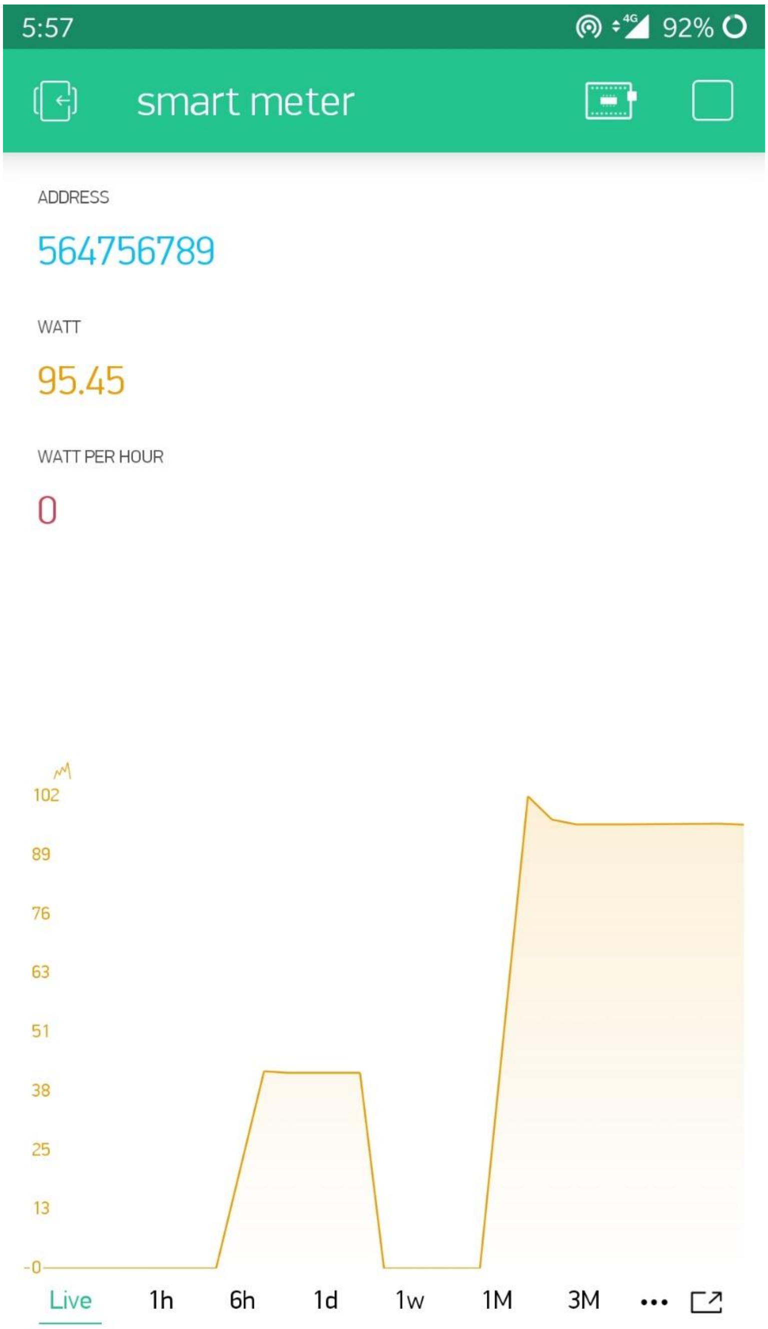


**Fig 4.5 Mobile app screenshot**

This is a mobile application developed for this project. Through this application the power flowing the load can be view in real time. In this application the mac address of the device can be viewed and as well as the power consumed in by the consumer can also be viewed in real time.

### 4.7 SUMMERY

This chapter explains the software process carried out in this project. It provides a view about how the programs function and it also helps to picturize the working of the project. It explains the use and funcsioning of each components in the project. In this chapter flowcharts are used to explain the functioning of the program. In these flowcharts working of the loops used in the program is represented diagrammatically. This chapter consist of algorithms for understanding of the flowcharts. This chapter is also provided with the programs used in the project. These programs are provided with comments by its side. The comments are included in important lines. These comment line explains the function of that particular line. In this chapter the screenshots of the app developed for this project is show casted.

# CHAPTER 5

## CONCLUSION

## 5.1 CHAPTERS OVERVIEW

### 5.1.1 CHAPTER 1 OVERVIEW

This chapter gives a brief introduction about the LoRa technology used in this project. This chapter explains about the main aim of the project and also discuses about use of smart energy meters. Though this chapter a brief explanation about the working of the LoRa based smart energy meter is explained. In this chapter [1.3] literature survey displays few IEEE reference papers. These reference papers provided required knowledge to build this project "LoRa based smart meter for rural areas without internet"

### 5.1.2 CHAPTER 2 OVERVIEW

This chapter discusses about integration of LoRa based smart meters in rural areas i.e. remote villages. In this chapter we have discussed about the need for LoRa based smart meters in remote villages in 2.1 need for lora based smart meters in remote villages. The explanation given about the need of this project in this chapter is based on the real time day to day conditions. It speaks about the reality conditions and also proposes solutions to handle that situation. In 2.2 implementation of lora based smart meters in remote villages the ways to implement the lora based smart meter is discussed. The block diagrams were used to explain about the data flow starting from household smart meter to a EB server. In this chapter 2.3 implementation of lora based smart meters in remote villages explains about the innovations made in LoRa gateway technologies in this project. It clearly states the major advantages of the innovation in LoRa gateway technology. Through this chapter the needs and ways to accomplish though needs is discussed. This chapter gives clear knowledge about the implementation and advantages of such implementation of this project.

### 5.1.3 CHAPTER 3 OVERVIEW

This chapter discuss about the hardware implementation of smart meter. In this chapter the circuit diagrams used in this project is detailly explained. These circuit diagrams plays a major role in proper execution of this project. This chapter is added with schematic diagrams with its explanations. These schematic diagrams show cast the connections between the microcontrollers and other sensors and transceivers. This would give a clear image about how the components are connected with one another. These connections and explanations also discus about the interface using which the components are connected with one another. In this chapter each and every component used are explained in 3.5 hardware description. These details consist of key features of the components with their specifications. By specifying the specification and key features of the components knowledge about the nature of the components and their uses are explained in this chapter.

### 5.1.4 CHAPTER 4 OVERVIEW

This chapter explains the software process carried out in this project. It provides a view about how the programs function and it also helps to picturize the working of the project. It explains the use and functioning of each components in the project. In this chapter flowcharts are used to explain the functioning of the program. In these flowcharts working of the loops used in the program is represented diagrammatically. This chapter consist of algorithms for understanding of the flowcharts. This chapter is also provided with the programs used in the project. These programs are provided with comments by its side. The comments are included in important lines. These comment line explains the function of that particular line. In this chapter the screenshots of the app developed for this project is show casted.

## 5.2 ENERGY METER MODULE

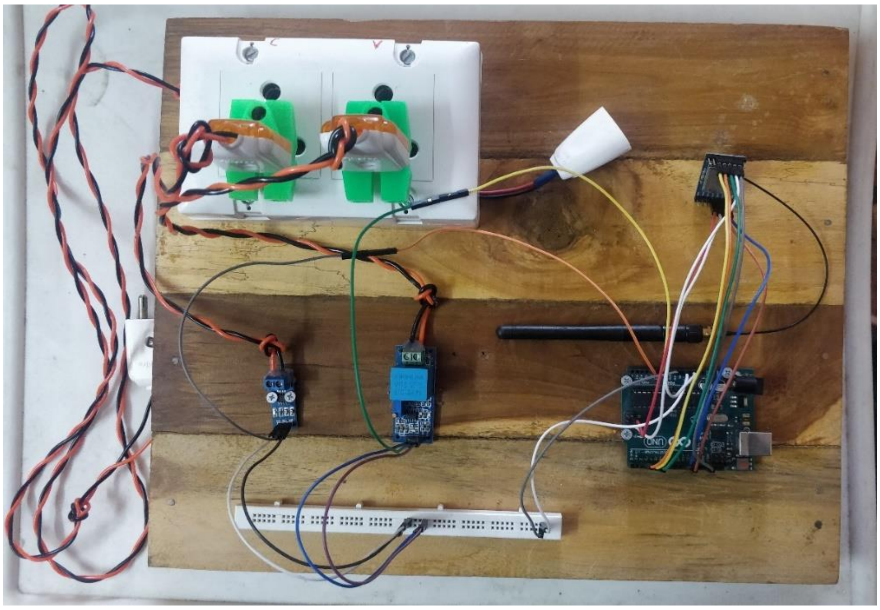

Fig 5.1 Energy meter module

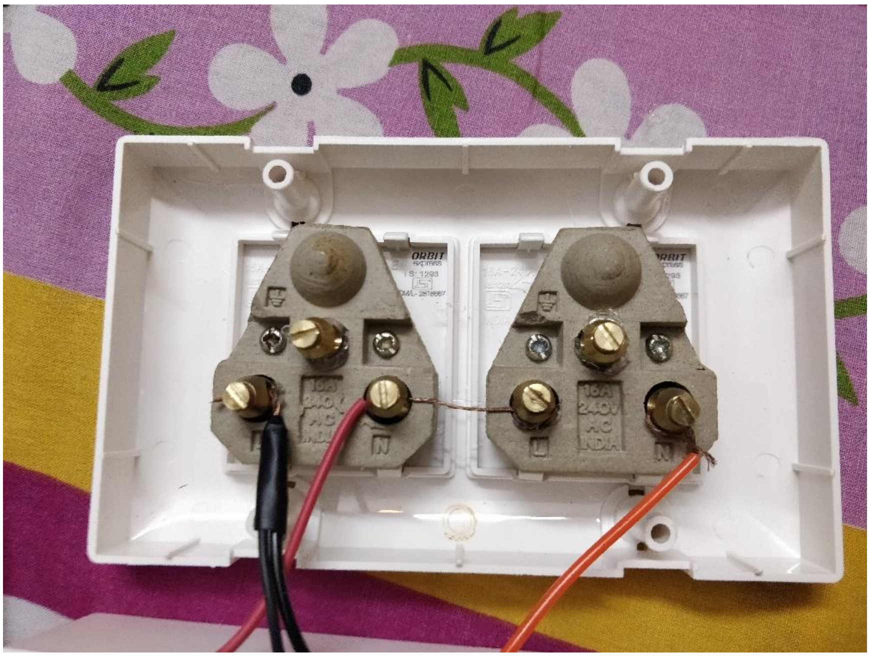

Fig 5.2 Circuit box

The components are mounted on a solid wooden slab. The switch box connection is given as specified in the fig.5.2.2. This connection provides current transformer (CT) and voltage transformer (VT) to get connected across the load serially and parallelly. The Arduino Uno is screwed up with the wooden slab. Bread board is stuck to the wooden slab which acts as the junction for the sensors and Arduino Uno’s 5V and Gnd. LoRa is connected to the Arduino Uno using jumper wires with SPI interface.

## 5.3 LoRa GATEWAY MODULE

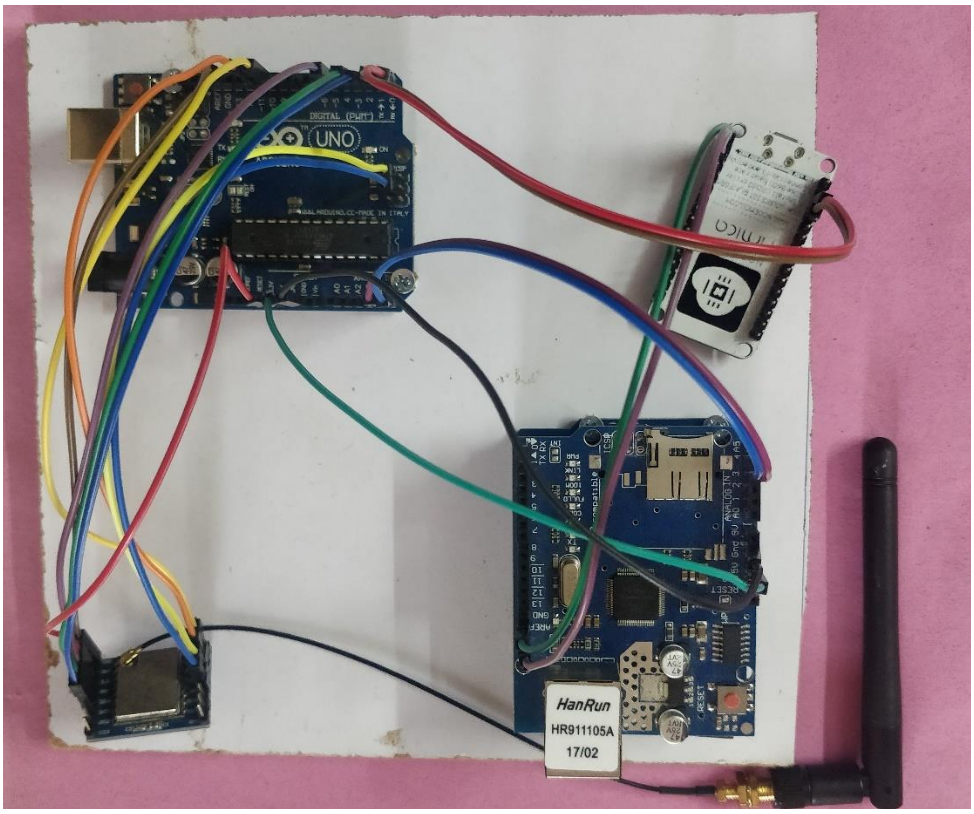


**Fig 5.3 LoRa gateway module**

In this setup two Arduino Uno are screwed up on cardboard. One of which is mounted with ethernet shield, while the other is connected to LoRa. Node MCU ESP8266 is also mounted on the cardboard. The node MCU ESP8266 is connected to the LoRa receiver Arduino and ethernet shield is also connected to the Arduino.

## 5.4 FINAL CONCLUSION

The project can be implemented in smart cities as well as it can be used in remote rural areas. Since it doesn't use any telecom services the data transfer cost is minimised. Even though building up LoRa gateway architecture is expensive it allows data flow to happen at minimal cost. This project will help to reduce man power by atomizing energy meter. The project has further more applications in future, as an alternative for GPS. This project is more cost effective than GSM based smart meter. Through this project the illegal tapping can be detected at a particular location where it happens. The wattmeter reading is sent to the EB server directly which makes documentation process more flexible and transparent. An mobile application is developed for the project which helps the consumer to view the wattmeter reading in their mobile app in real time. Since the documentation is digitalized the big data analysis can be run over the consumer power usage in various circumstances. This helps the Electricity Board to predict the power requirements in future and plan accordingly. Since the complete process is so flexible and transparent it avoids bribery to happen.

# CHAPTER 6

## REFERENCE

## 6.1 Reference


[1].Visalatchi Senthilnathan , Kamal Sandeep Karreddula for Smart energy metering and power theft control using arduino & GSM, 2017 2nd International Conference for Convergence in Technology (I2CT),. Electronic ISBN: 978-1-5090-4307-1, Print on Demand(PoD) ISBN: 978-1-5090-4308-8

[2] .Gunawan Wibisono, Gilang Permata Saktiaji, Ihsan Ibrahim for Techno economic analysis of smart meter reading implementation in PLN Bali using LoRa technology,. 2017 International Conference on Broadband Communication, Wireless Sensors and Powering (BCWSP), Electronic ISBN: 978-1-5386-2833-1, Print on Demand(PoD) ISBN: 978-1-5386-2834-8

[3]. Tian Xia ; Cencen Liu ; Xin Zheng ; Ming Lei ; Shangpeng Wang ; Yingchun Wang,. Research on Field Testing Method of Digital Energy Meters Based on Digital Reference Meter with High Accuracy, 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Electronic ISBN: 978-1-5386-8549-5

[4]. Win Hlaing ; Somchai Thepphaeng ; Varunyou Nontaboot ; Natthanan Tangsunantham ; Tanayoot Sangsuwan ; Chaiyod Pira,. Implementation of WiFi-based single phase smart meter for Internet of Things, 2017 International Electrical Engineering Congress (iEECON), Electronic ISBN: 978-1-5090-4666-9