



A Project book of

**Microcontroller and GSM Based Power Measurement
and Monitoring System**

Submitted By

Utpal Debnath

Rahul Saha

Sadia Sultana Shithil

ID: 2012-3-55-012

ID: 2014-1-55-011

ID: 2014-1-55-019

Supervised By

Prof. Dr. Md. Habibur Rahman

Dept. of Electronics and Electrical Engineering

University of Dhaka

Approval

This project has been prepared and submitted by **Utpal Debnath, ID: 2012-3-55-012, Rahul Saha, ID: 2014-1-55-011 and Sadia Sultana Shithil, ID: 2014-1-55-019**. This project report is presented to the Department of Electronics and Communications Engineering, East West University for the partial fulfillment of the requirement for the degree of **B.Sc.in Electronics and Telecommunications Engineering**, under complete supervision of the undersigned.

Dr. M. Mofazzal Hossain

Professor & Chairperson
Department of ECE
East West University

Dr. Md. Habibur Rahman

Project Supervisor
Professor
Department of EEE
University of Dhaka

Rasel Ahmmed

Project Co-Supervisor
Lecturer
Department of ECE
East West University

Declaration

We, students of ECE department, hereby declare that the project entitled “**Microcontroller and GSM based power measurement and monitoring system**” is performed by ourselves. We have submitted the project to Electronics and Communication Engineering Department under the supervision of **Dr. Md. Habibur Rahman**, Professor, Department of Electronics and Electrical Engineering, University of Dhaka.

.....

Prof. Dr. Md. Habibur Rahman

Project Supervisor

Professor

Department of EEE

University of Dhaka

.....

Utpal Debnath

2012-3-55-012

.....

Rahul Saha

2014-1-55-011

.....

Sadia Sultana Shithil

2014-1-55-019

Acknowledgement

We are highly indebted to our adviser **Dr. Md. Habibur Rahman**, Professor, Department of Electronics and Electrical Engineering, University of Dhaka for his guidance and constant supervision as well as for providing necessary information regarding to the project & also for his support in completing the project.

We also thank to our project co-supervisor **Md. Rasel Ahmed** sir for his participation and maintaining discipline and performing regular tasks responsively.

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of this project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

Finally, we would like to thank our chairperson sir and department for giving us the chance to complete the project with the facilities needed for it.

ABSTRACT

This development of “**Microcontroller and GSM based power measurement and monitoring system**” is for measuring the voltage and current at a time for the user. The aim of the project is to show the voltage and current based on a peak value by microcontroller system (Arduino). The project constitutes a microcontroller, a current sensor, a transformer, bulb as load and a plug to connect with ac load. The load is connected with the transformer and the current sensor. Current sensor converted the current into 0 to 5 volts that directly flow towards into the Arduino and measured the current. Transformer converts the voltage into 12 volt that goes on the preset and connected to the Arduino analog channel and the voltage is measured. The basic code of our project is to get the ac current and voltage to measure the power by the microcontroller system.

TABLE OF CONTENTS

APPROVAL.....	01
DECLARATION.....	02
ACKNOWLEDGEMENT.....	03
ABSTRACT.....	04
<u>CHAPTER 1</u>	
1.1 Introduction.....	07
1.2 Aim of the Project.....	07
<u>CHAPTER 2</u>	
Related theory:	
2.1 Microcontrollers.....	08
2.1.1 Types of Microcontrollers.....	09
2.2 ARDUINO.....	10
2.2.1 Power pins of ARDUINO.....	11
2.3 Pin Description.....	12
2.4 Potentiometer.....	12
2.5 Types of Potentiometer.....	13
2.6 Current Sensor.....	14
2.7 Transformer.....	15
2.8 GSM Module.....	16

CHAPTER 3**3.1 HARDWARE DESIGN:**

3.1.1 Block Diagram.....	17
3.1.2 Control Circuit.....	18
3.1.3 System Circuit Physically Presentation.....	22

3.2 SOFTWARE DESIGN:

Design of Different Unit.....	23
3.3 Code Discussion.....	23
3.4 Verifying the Hardware.....	27
3.5 ARDUINO Language.....	27

CHAPTER 4

4.1 Performance Analysis.....	28
4.2 Working process of power meter system.....	28
4.3 Data.....	30
4.4 Monitoring Output result.....	31

CHAPTER 5**DISCUSSION**

5.1 CONCLUSION.....	32
5.2 FUTUREWORK.....	33
APPENDIX.....	34
REFERENCE.....	39

CHAPTER 1

INTRODUCTION

1.1 Introduction

An electricity meter, electric meter, electrical meter, or energy meter is a device that measures the amount of electric energy consumed by a residence, a business, or an electrically powered device.

Electric utilities use electric meters installed at customers' premises for billing purposes. They are typically calibrated in billing units, the most common one being the kilowatt hour (kwh). They are usually read once each billing period.

1.2 Aim of the Project

With the advancement of technology, objectives of energy meters are not merely to measure energy units. Objectives went far behind from just giving energy readings of some specific consumer. Also, these objectives are not constant; they are increasing day by day with the increase in advancement in energy sector.

By measuring the power energy by microcontroller based system which not only shows the current or voltage value but also we can implement the new features as usable belongings to comfort and make easy our lifestyle.

Chapter 2

Related theories

2.1 Microcontrollers:

A microcontroller is a small computer on a single integrated circuit (IC). A microcontroller is self-controlled system with peripherals, memory and processor that can be used as an embedded system. It is also known as embedded controller.

.

Typically this includes a CPU, RAM, some form of ROM, I/O ports, and timers. Unlike a general-purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task -- to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production costs.

Microcontrollers are sometimes called an embedded microcontroller, which just means that they are part of an embedded system-- that is, one part of a larger device or system.

2.1.1 Types of Microcontrollers

a) Classification According to Number of Bits

The bits in microcontroller are 8-bits, 16-bits and 32-bits microcontroller.

In **8-bit** microcontroller, the point when the internal bus is 8-bit then the ALU is performs the arithmetic and logic operations. The examples of 8-bit microcontrollers are Intel 8031/8051, PIC1x and Motorola MC68HC11 families.

The **16-bit** microcontroller performs greater precision and performance as compared to 8-bit. For example 8 bit microcontrollers can only use 8 bits, resulting in a final range of 0x00 – 0xFF (0-255) for every cycle. In contrast, 16 bit microcontrollers with its 16 bit data width has a range of 0x0000 – 0xFFFF (0-65535) for every cycle. A longer timer most extreme worth can likely prove to be useful in certain applications and circuits. It can automatically operate on two 16 bit numbers. Some examples of 16-bit microcontroller are 16-bit MCUs are extended 8051XA, PIC2x, Intel 8096 and Motorola MC68HC12 families.

The **32-bit** microcontroller uses the 32-bit instructions to perform the arithmetic and logic operations. These are used in automatically controlled devices including implantable medical devices, engine control systems, office machines, appliances and other types of embedded systems. Some examples are Intel/Atmel 251 family, PIC3x

b) Classification According to Memory Devices

The memory devices are divided into two types, they are

1. Embedded memory microcontroller
2. External memory microcontroller

2.2 Arduino:

Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

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. The Uno differs from all preceding boards in that it does not use the FTDI USB-to -serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to -serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode [6].

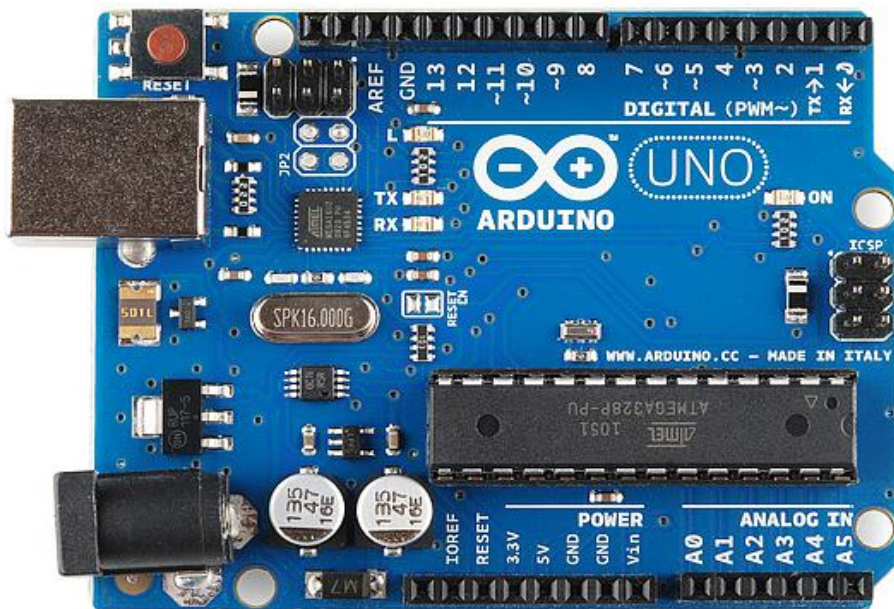


Figure 2.2: Arduino UNO Board

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. The Uno differs from all preceding boards in that it does not use the FTDI USB-to -serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to -serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

2.2.1 Power pins of Arduino:

- 1. VIN:** The input voltage to the Arduino board when it's using an external power source. You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges.
- 2. 5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3. 3.3V:** A 3.3 volt supply generated by the on-board FTDI chip.
- 4. GND:** Ground pins

2.3 Pin Descriptions:

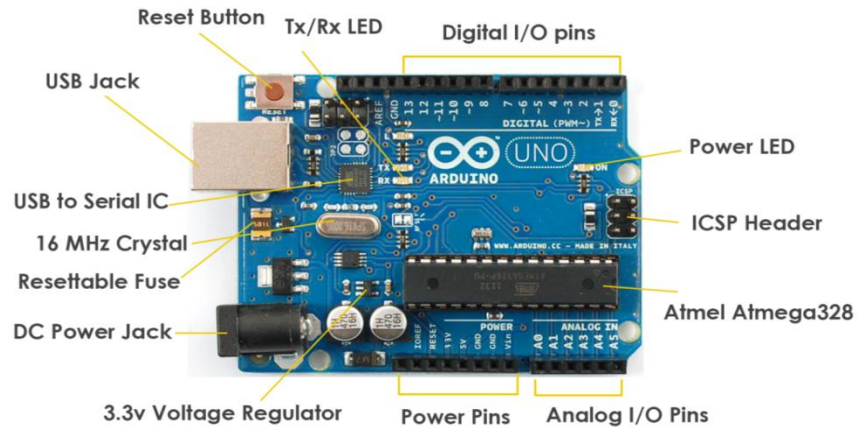


Figure2.2: Power pins of Arduino

2.4 Potentiometer:

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor [1].



Figure 2.6: Potentiomer

2.5 Types of Potentiometer:

Type	Description	Applications
Single-turn pot	Single rotation of approximately 270 degrees.	Most common pot ,used in applications where single turn provides enough control resolution
Multi-turn pot	Multiple rotation for increased precision.	Used where high precision and resolution is required. The worm-gear multi turn pots are often used as trim pots on PCB.
Dual-gang pot	Two potentiometer combined on the same shaft, enabling the parallel settings of two channels.	Used in stereo audio volume control.
Concentric pot	Duel potentiometer, where the two potentiometers are individually adjusted by means of concentric shafts. Enables the use of two controls on one unit.	Often encountered in car radios, where the volume and tone controls are combined.
Servo pot	A motorized potentiometer automatically adjusted by a servo meter.	Used where manual and Automatic adjustment is required. Often seen in audio equipment, where the remote - control can turn the volume control knob.

Table2.5: Types of Potentiometer [2]

2.6 Current Sensor:

A **current sensor** is a device that detects electric current in a wire, and generates a signal proportional to that current. The generated signal could be analog voltage or current or even a digital output. The generated signal can be then used to display the measured current in an ammeter, or can be stored for further analysis in a data acquisition system, or can be used for the purpose of control [3].

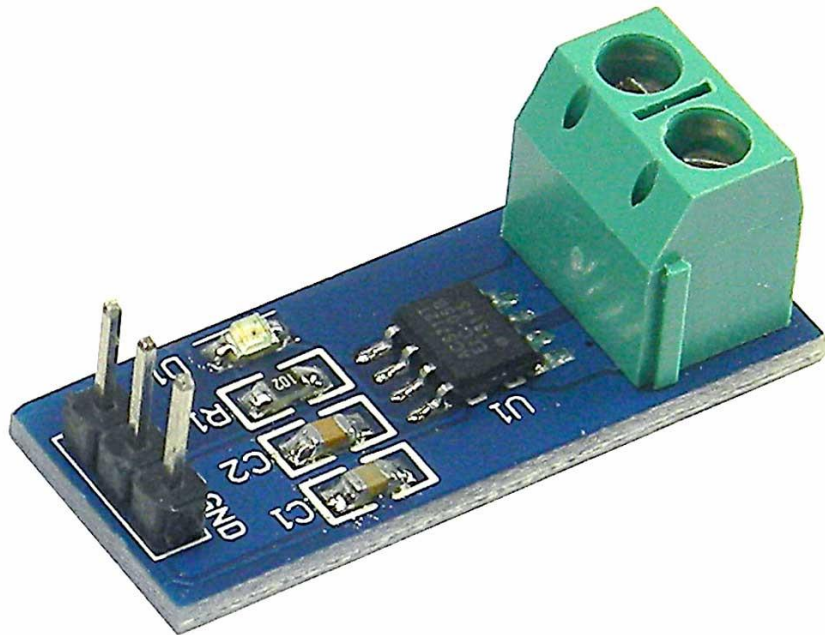


Fig: Current Sensor

2.7 Transformer:

A **transformer** is a static electrical device (sometimes called an electrical machine) that transfers electrical energy between two or more circuits. A varying current in one coil of the transformer produces a varying magnetic flux, which, in turn, induces a varying electromotive force (emf) or voltage across a second coil wound around the same core. Electrical energy can be transferred between the two coils, without a metallic connection between the two circuits. Faraday's law of induction discovered in 1831 described the induced voltage effect in any coil due to changing magnetic flux encircled by the coil [4].

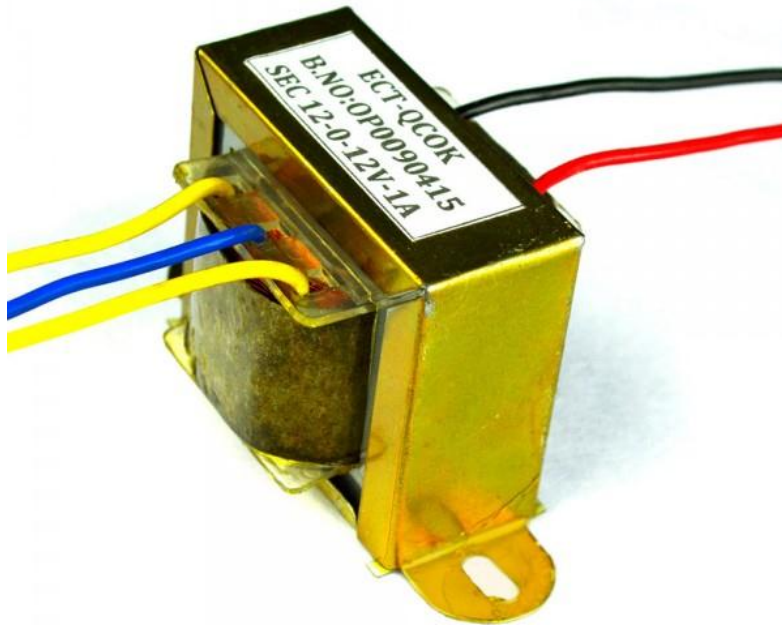


Fig: 12V Stepdown Transformer

2.8 GSM Module:

A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. The modem (modulator-demodulator) is a critical part here. These modules consists of a GSM module or GPRS modem powered by a power supply circuit and communication interfaces (like RS-232, USB 2.0, and others) for computer. A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, or it can be a mobile phone that provides GSM modem capabilities [5].



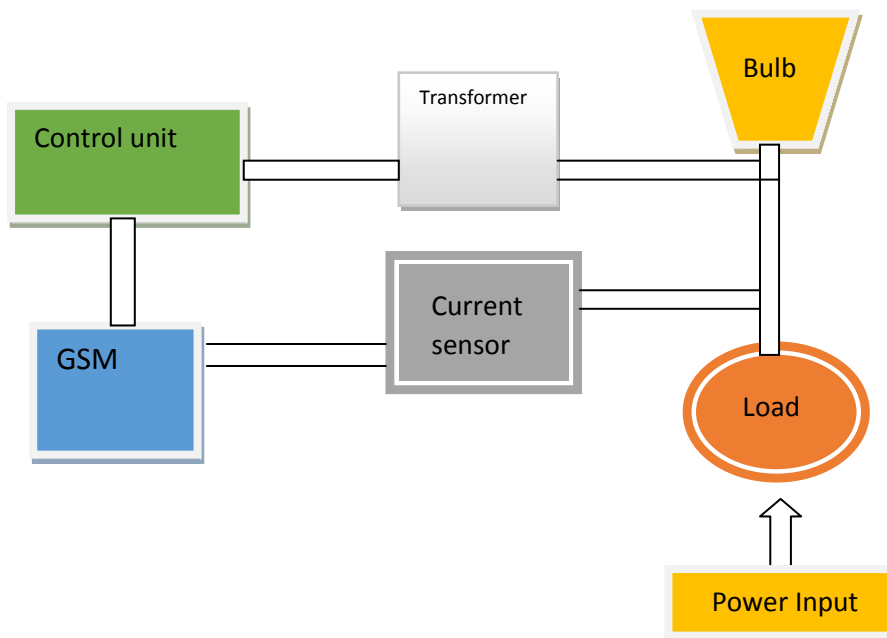
Fig: GSM Module

CHAPTER 3

HARDWARE DESIGN

3.1.1 Block Diagram:

This block diagram represents the whole connection of power metering of current and voltage with the gsm that can make the connection with wireless or internet connection.



Block diagram of power measuring circuit

3.1.2 Control Circuit

Block diagram of current sensor control circuit:

Here, the current sensor connects with the Arduino output A1 and the current sensor connects with the load to measure the current.

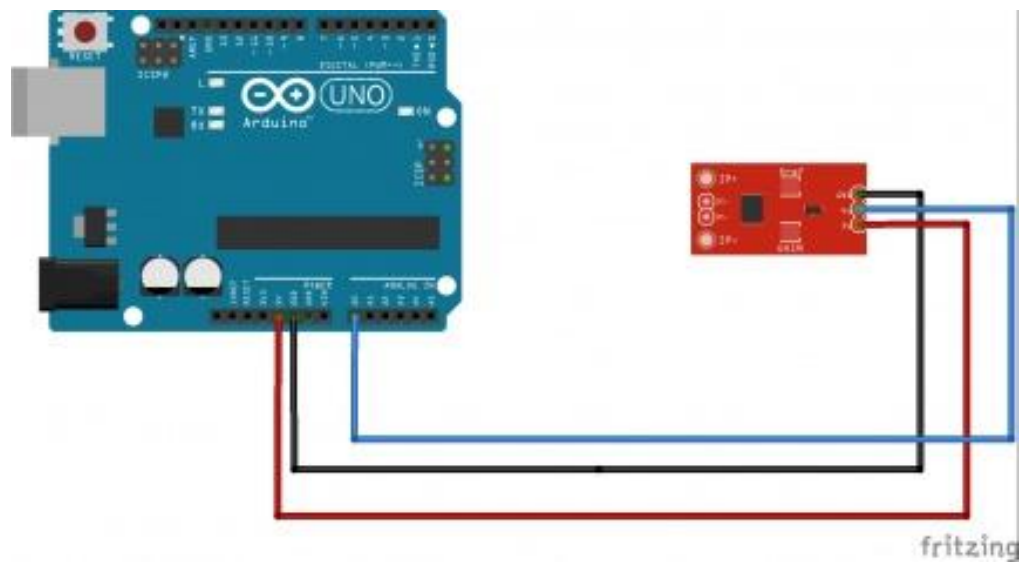


Fig: Connection of current sensor [8].

Block diagram of transformer control circuit:

Here, the transformer connects with the Arduino output A0 via pot and the transformer connects with the load to measure the voltage.

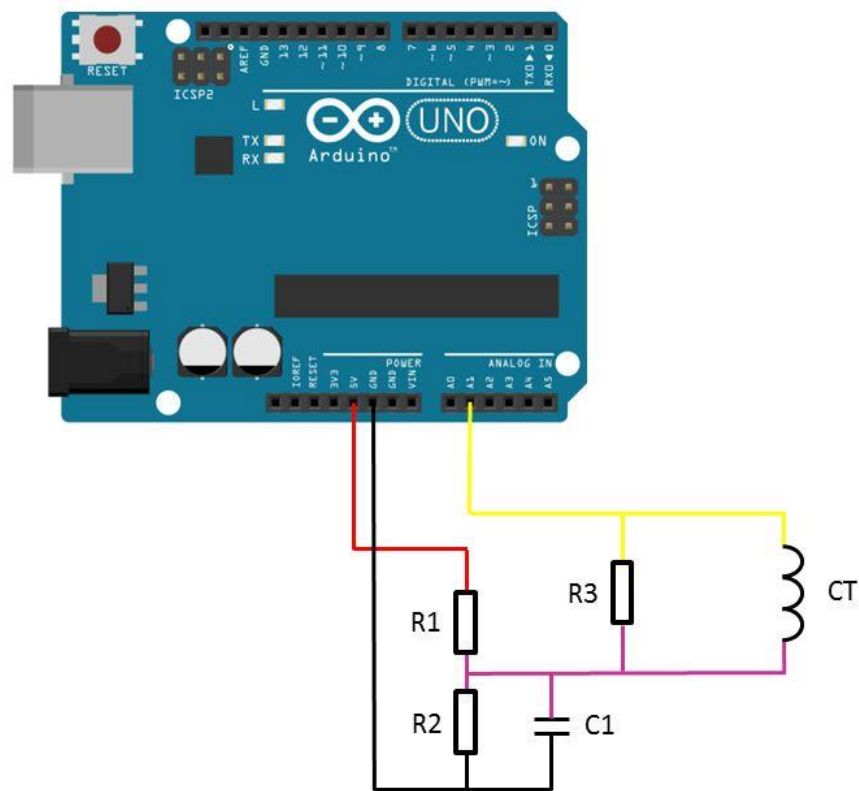


Fig: Connection of Transformer [7].

Potentiometer pin connection with arduino:

Potentiometers are variable resistors and they function to alter their resistance via a knob or dial. Potentiometers have a range of resistance. They can be attuned from zero ohms to whatever maximum resistance that is specific to it. For example, a potentiometer of 10 k Ω can be adjusted from 0 Ω to its maximum of 10 k Ω . All potentiometers have three pins. The outer pins are used for connecting power source (Vref and gnd). The middle pin (output) give us the variable of resistance value. Connect battery to outer pins of potentiometer and the positive end of led (larger pin) to middle pin. Now turn the knob (or dial) left and right.

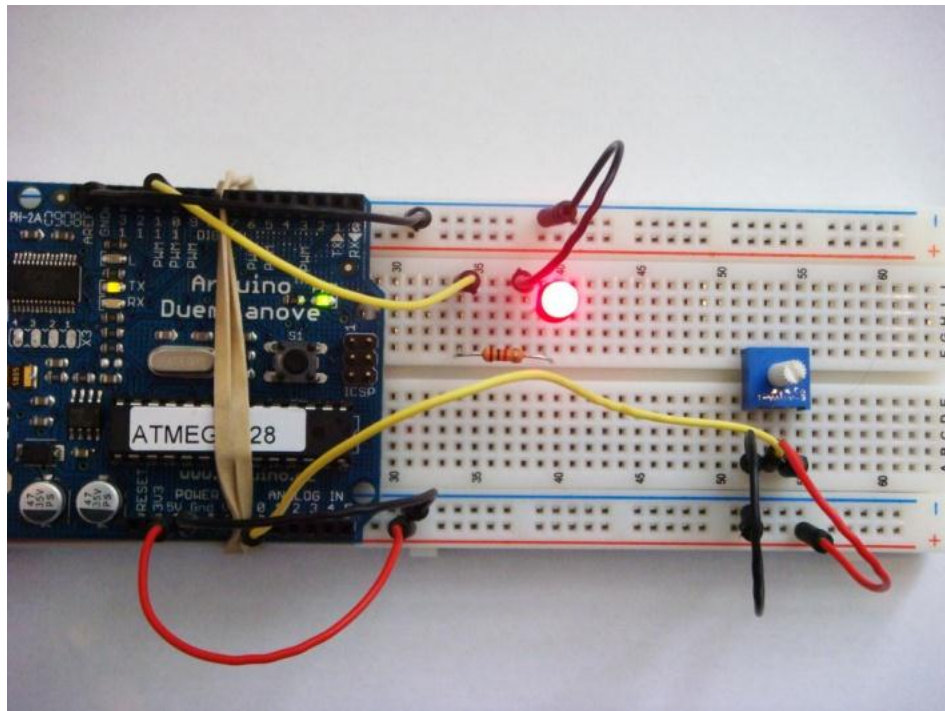


Fig: Connection with the Potention meter

GSM connection with Arduino

GSM is an international standard for mobile telephones. It is an acronym that stands for Global System for Mobile Communications. It is also sometimes referred to as 2G, as it is a second-generation cellular network.

To use GPRS for internet access, and for the Arduino to request or serve web pages, you need to obtain the Access Point Name (APN) and a username/password from the network operator. See the information in Connecting to the Internet for more information about using the data capabilities of the shield. Among other things, GSM supports outgoing and incoming voice calls, Simple Message System (SMS or text messaging), and data communication (via GPRS).

The Arduino GSM shield is a a GSM modem. From the mobile operator perspective, the Arduino GSM shield looks just like a mobile phone. From the Arduino perspective, the Arduino GSM shield looks just like a modem [6].

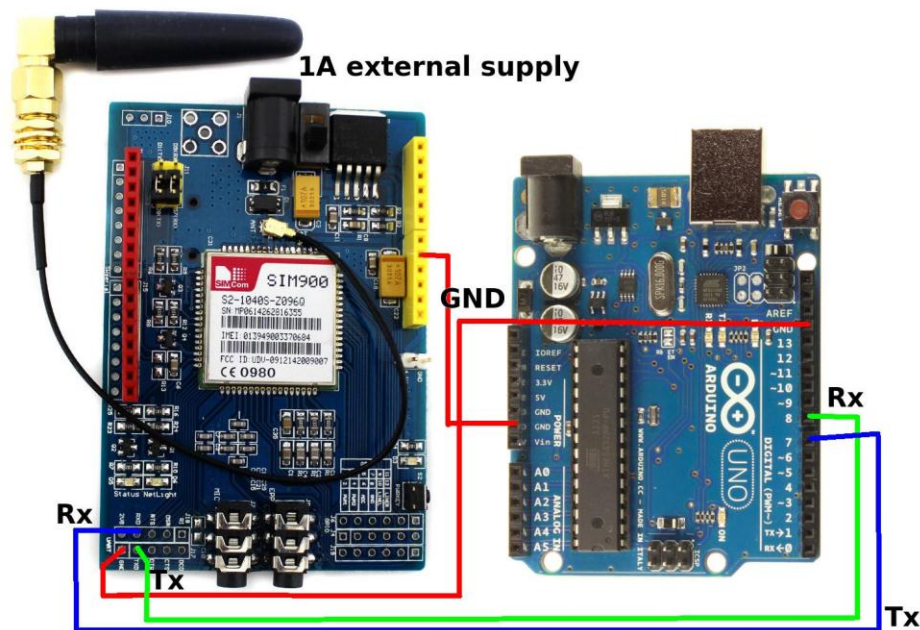


Fig: GSM connection with Arduino

3.1.3 System Circuit Physically Presentation:

The working process of our power metering measurement system that looks not so professional but the connection of wire circuit and the measuring the power from the load which goes through the AC load.

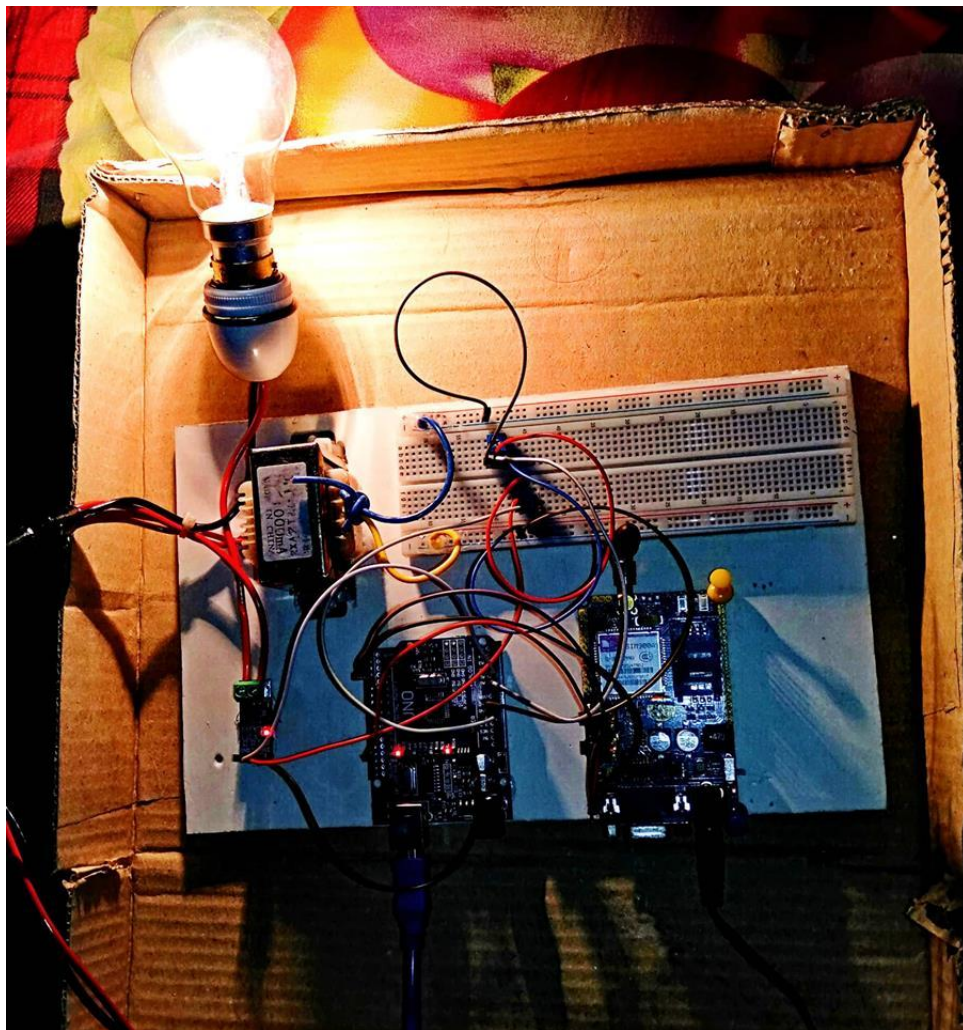


Fig: whole connection circuit and power measuring board

3.2 SOFTWARE DESIGN:

The working process of the project is also helps to complete by code which we programmed gradually to show the result of measuring power current and voltage on the mobile screen by coding on Arduino.

Design of Different Unit

3.3 Code Discussion:

There is the code of whole project work done by the arduino software coding system and the calculation and measurement of putting values were accurately correct as shown.

```

File Edit Sketch Tools Help
final_code
#include <SoftwareSerial.h>
const byte rxPin = 2;
const byte txPin = 3;

SoftwareSerial SIM900(rxPin, txPin);
char msg;

float voltageB = 0;

//float amp = 0;
float rmsc,power, rmsc_volt, power_volt;
boolean sendata = false;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  SIM900.begin(9600);
  SIM900.print("AT+CMGF=1\r");
  delay(100);
  SIM900.print("AT+CNMI=2,2,0,0,0\r");
  delay(100);
  // SendMessage();
}

void loop() {

  // put your main code here, to run repeatedly:
  if (SIM900.available() > 0)
  {
File Edit Sketch Tools Help
final_code
void loop() {

  // put your main code here, to run repeatedly:
  if (SIM900.available() > 0)
  {
    Serial.println("in");
    if (msg == 'F') {
      Serial.println("F");

      String s = "Voltage = ";
      String d = String(voltageB);
      String ss = " : Power = ";
      String dd = String(power_volt);
      String e = " : Current = ";
      String f = String(rmsc);
      String g = " : Power = ";
      String h = String(power);
      String t = s + d + ss + dd + e + f + g + h;
      Serial.println(t);
      SendMessage(t);
      msg = "";
    }
  }
  // int a = 2;
  // String msgs = String(a);
  // SendMessage(msgs);
  msg = SIM900.read();
  delay(1000);
  getCurrent();
}
}

```


File Edit Sketch Tools Help



final_code

```

delay(1000);
getCurrent();
}

void getCurrent() {
  int current[200];
  int volt[200];
  int i,maxc,minc;
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  voltageB= voltage*(220/5);

  Serial.println("Voltage=");
  Serial.println(voltageB);
  /*int sensorValue1 = analogRead(A1);
  float voltageA = sensorValue1 * (5000.0 / 1023.0);
  float amps=((voltageA-2500)/66);
  amp=abs(amps);*/

  for(i=0;i<200;i++)
  current[i] = analogRead(A1);

  maxc=0;
  minc=32000;
  for(i=0;i<200;i++)
  {

```

File Edit Sketch Tools Help



final_code

```

for(i=0;i<200;i++)
{
  //Serial.println(current[i]); //raw current display on serial monitor
  if(current[i]<minc) minc=current[i];
  if(current[i]>maxc) maxc=current[i];
  rmisc=(0.079*(maxc-minc)/(2*1.4142)) - 0.08;
}
Serial.println("Current=");
Serial.println(rmisc);
power=voltageB*rmisc;
Serial.println("Power=");
Serial.println(power);

// for voltage
for(i=0;i<200;i++)
volt[i] = analogRead(A0);

maxc=0;
minc=32000;
for(i=0;i<200;i++)
{
  //Serial.println(current[i]); //raw current display on serial monitor
  if(volt[i]<minc) minc=volt[i];
  if(volt[i]>maxc) maxc=volt[i];
  rmisc_volt=(0.079*(maxc-minc)/(2*1.4142)) - 0.08;
}
Serial.println("Voltage=");

```

File Edit Sketch Tools Help



final_code

```

{
  //Serial.println(current[i]); //raw current display on serial monitor
  if(volt[i]<minc) minc=volt[i];
  if(volt[i]>maxc) maxc=volt[i];
  rmisc_volt=(0.079*(maxc-minc)/(2*1.4142)) - 0.08;
}
Serial.println("Voltage=");
Serial.println(rmisc_volt);
power_volt=voltageB*rmisc_volt;
Serial.println("Power=");
Serial.println(power_volt);

//gsm will sent the value of voltageB [voltage] and voltageA[current]
}

void SendMessage(String msg)
{
  SIM900.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
  delay(1000); // Delay of 1000 milli seconds or 1 second
  SIM900.println("AT+CMGS=\"+8801770722247\\\""); // Replace x with mobile number

  delay(1000);
  SIM900.println(msg);// The SMS text you want to send
  delay(100);
  SIM900.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
}

```

Coding procedures:

1. Connect the pushbutton to the breadboard.
2. Connect one side of the pushbutton to the 5-volt pin on the Arduino board using a jumper wire.
3. Connect one side of the 10K resistor to the other side of the pushbutton.
4. Connect the other side of the resistor to the ground pin on the Arduino. You may have to use a jumper wire to make it reach.
5. On the same side, the resistor is connected to the pushbutton, connect a jumper wire and run it to pin 2 on the Arduino board.
6. Connect the to the computer with the USB cable.
7. Open the sketch for this section.
8. Click the Upload Button (located to the immediate right of the Verify button). This button will also turn orange and then back to blue once the sketch is uploaded to the Arduino board.
9. Now go to the menu bar at the top and select Tools > Serial Monitor. Or you could use the shortcut key, Shift + Control + M.
10. The serial monitor window will open and will be spouting off numbers. It should be a bunch of zeros.
11. Press and hold the pushbutton – watch the serial monitor window, the numbers should now be ones.
12. If the numbers are not scrolling, click Auto-scroll at the bottom left of the serial monitor window.

3.4 Verifying the Hardware:

When the Arduino IDE software installation is done, let's connect the computer to the mc board, load a small program, and verify that all components are working together. First, need to connect the USB cable to our mc board and then plug the other end of the USB cable into our computer.

3.5 ARDUINO Language:

The ARDUINO language is implemented in C/C++. When we write an sketch we are implicitly making use of the Wiring library, which is included with the ARDUINO IDE. This allows us to make run able programs by using only two functions: setup () and loop (). As mentioned, the Wiring language is inspired by Processing, and the ARDUINO language structure is inherited from the Processing language, where the equivalent functions are called setup (). We need to include both functions in every ARDUINO program, even if we don't need one of them. Let's analyze the structure of a simple ARDUINO sketch using again the Blink example.

CHAPTER 4

4.1 PERFORMANCE ANALYSIS:

All the apparatus of the system are connected according to the circuit diagram. The system has been tested several times and it is found working perfectly. The whole process of connection by wire and cable to the load is absolutely safe and tested. Program running devices repaired several times for better performance and the code established in very carefully to measure the power as close to accurate.

4.2 Working Process of Power meter system:

The whole project was done by our project partners and the most importantly our honorable sir that turns the project work so fluent and interesting.

First of all, we connected the 100W bulb with the direct AC load to get the AC current. Then we took two important materials **Current sensor** and **Transformer** to measure the current and voltage value.

Two AC line connections gone through the **current sensor** that built in with three output pin. One is indicating the ground, another one is indicating the Vcc(5v) and the last one is indicating the output pin connection on Arduino output A1.

Again, two AC line connections gone through the **Transformer** where two lines separately connected with the one side of **Potentiometer** and the other line straight through the ground of the bread board.

One important thing, the potentiometer has got three pin connections which are connecting with Arduino, Transformer and ground. We put there a resistor to deduce the negative current and voltage.

The basic connections of getting the output of voltage and current, when the output pin connections are connected with the Arduino.

We also measured the **Power** of the current and voltage by calculating the **rms** value.

GSM module used only to connect with the wireless mobile connection to get the output digitally in anywhere. But I have planned something extra theoretical experiments which could not possible to represent due to lack of financial support and time. The theories are like **Billing system, Control the power connection, measuring the cost of smart metering system.**

The main work done actually to get **current, voltage and power** where the **units** are **Ampere, Volt and Watt in hour.**

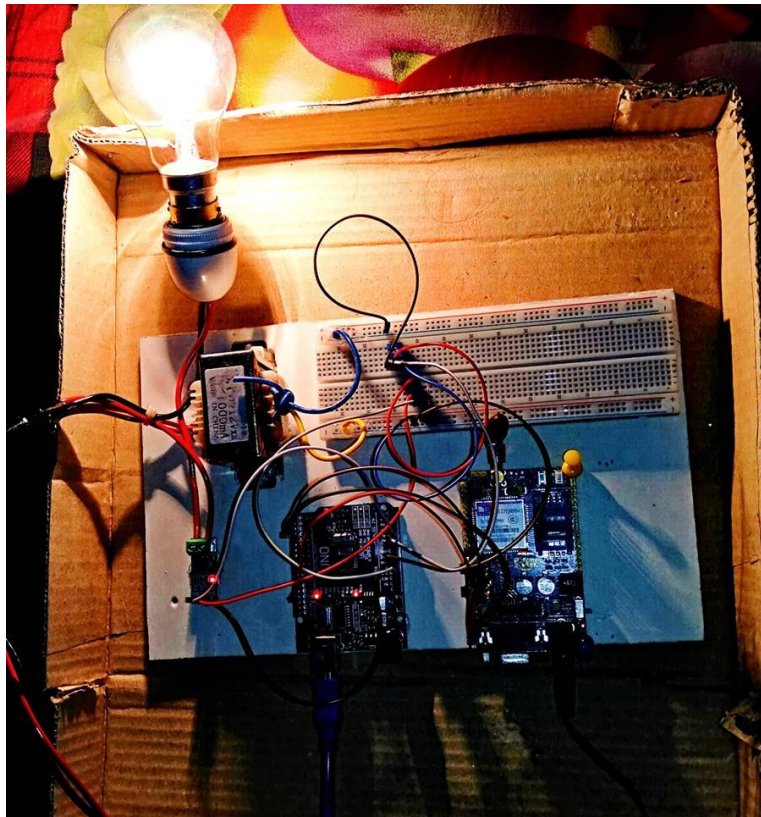


Fig: The complete work of the power meter project

4.3 Data:

We initially took the raw current to measure the variation of current flow that flows in sinusudially. We can see here the maximum and minimum values that represent the variations of peak to peak values which were monitored by arduino.

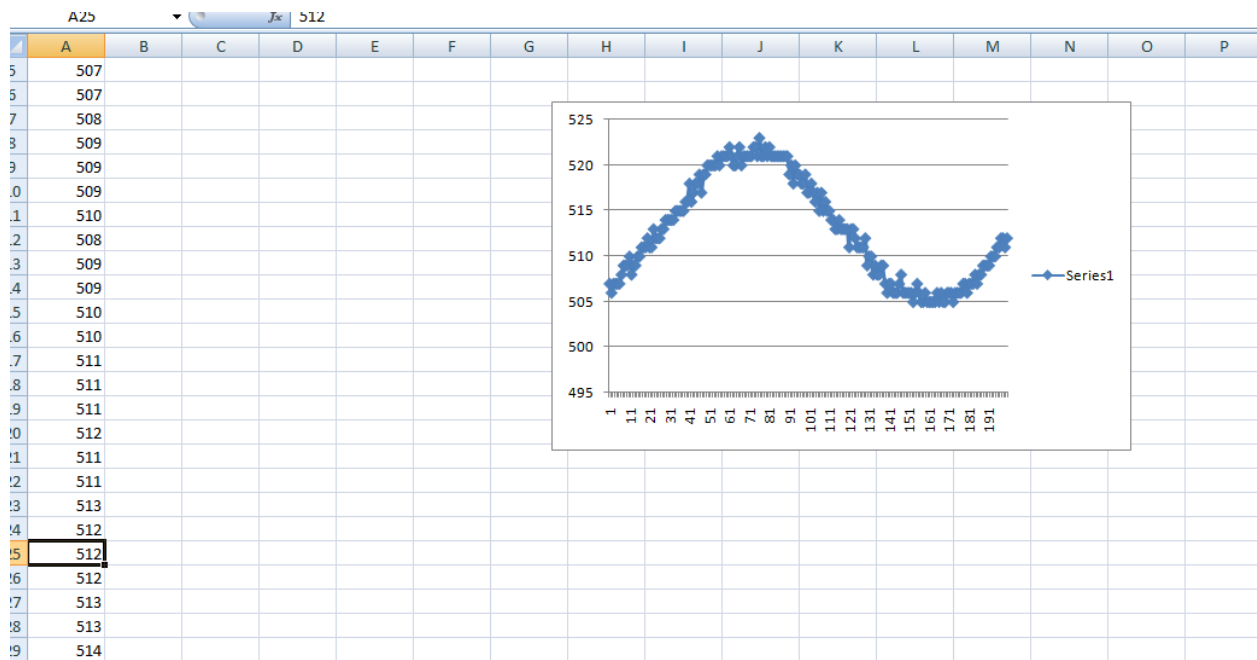


Fig: Variations of RAW current peak to peak values

4.4 Monitoring Output Result:

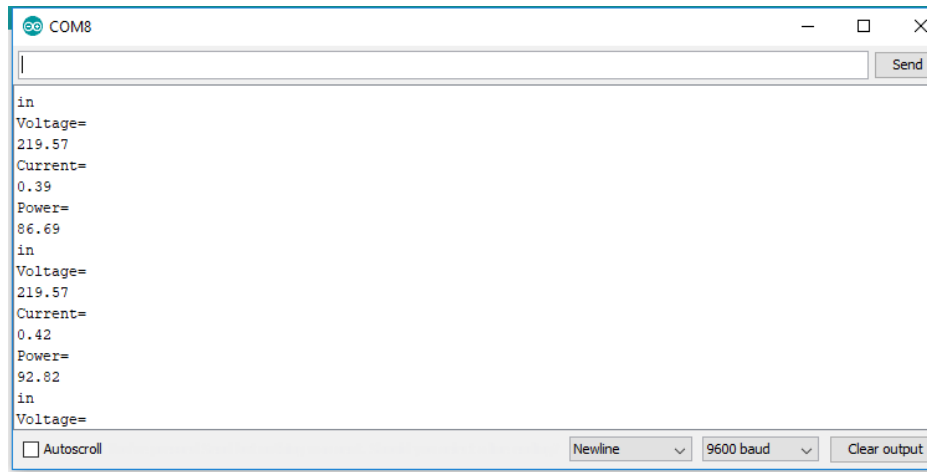


Fig: Final result of serial monitoring

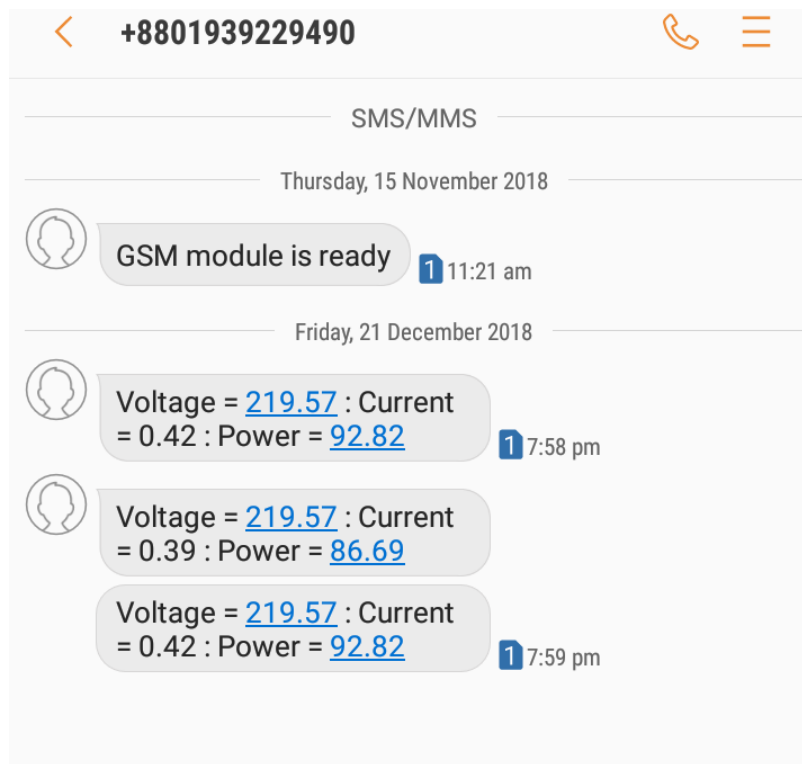


Fig: Final result of mobile messaging

CHAPTER 5

DISCUSSION

5.1 CONCLUSION:

As mentioned above, our setup of circuit connection with the load which is connected with a transformer and a current sensor. There were different voltage and different current output shown in the monitoring screen because the scaling factor of low quality equipments and also the variation of electron flow to the load.

To calibrate our power meter, that was quiet sure the current and voltage of the meter was drawn accurately what we expected to be completed the work. We used 100W bulb to measure the current that works perfectly as we expected. The voltage from the power converted to 5V by potentiometer as we used the arduino machine to get the output digitally and to reduce the power short circuit of the board.

Finally, the outcome of the project is successful by measuring the power, current and voltage from the load by micro controlling coding system.

5.2 FUTURE WORK:

The project of power measuring the power, voltage and current system works only for the user who wants to know the instant power value of current and voltage from their usable meter. Now a days, peoples are so much curious to get a better opportunity to easy their lifestyle.

This power meter system is actually a theory of smart metering system and it's future concept for better upgrading the life work and consuming the time on metering system. We measured here only current and voltage to show only the theory of improvements of the metering system. Here we can measure the phase, velocity of current flow, voltage signaling and the total control of electricity of a home or corporate office.

By using GSM module system, we can operate the power meter from outside as well. Now a days, few lights and fans are already made in wireless connection system. By adding this system the whole project could stand with the side of govt. to reduce the power loss and prevent the electrical accident.

Appendix

```
#include <SoftwareSerial.h>

const byte rxPin = 2;

const byte txPin = 3;

SoftwareSerial SIM900(rxPin, txPin);

char msg;

float voltageB = 0;

//float amp = 0;

float rMSC,power, rMSC_volt, power_volt;

boolean sendata = false;

void setup() {

  // put your setup code here, to run once:

  Serial.begin(9600);

  SIM900.begin(9600);

  SIM900.print("AT+CMGF=1\r");

  delay(100);

  SIM900.print("AT+CNMI=2,2,0,0,0\r");

  delay(100);
```

```
// SendMessage();  
}  
  
void loop() {  
  
// put your main code here, to run repeatedly:  
if (SIM900.available() > 0)  
{  
  
Serial.println("in");  
  
if (msg == 'F') {  
  
Serial.println("F");  
  
  
String s = "Voltage = ";  
  
String d = String(voltageB);  
  
String ss = " : Power = ";  
  
String dd = String(power_volt);  
  
String e = " : Current = ";  
  
String f = String(rmsc);  
  
String g = " : Power = ";  
  
String h = String(power);  
  
String t = s + d + ss + dd + e + f + g + h;  
  
Serial.println(t);  
  
SendMessage(t);
```

```
    msg = "";  
  }  
}  
  
// int a = 2;  
  
// String msgs = String(a);  
  
// SendMessage(msgs);  
  
msg = SIM900.read();  
  
delay(1000);  
  
getCurrent();  
  
}  
  
void getCurrent() {  
  int current[200];  
  int volt[200];  
  int i,maxc,minc;  
  
  // read the input on analog pin 0:  
  int sensorValue = analogRead(A0);  
  
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):  
  float voltage = sensorValue * (5.0 / 1023.0);  
  
  // print out the value you read:  
  
  voltageB= voltage*(220/5);
```

```
Serial.println("Voltage=");  
  
Serial.println(voltageB);  
  
/*int sensorValue1 = analogRead(A1);  
  
float voltageA = sensorValue1 * (5000.0 / 1023.0);  
  
float amps=((voltageA-2500)/66);  
  
amp=abs(amps);*/  
  
for(i=0;i<200;i++)  
  
current[i] = analogRead(A1);  
  
  
maxc=0;  
  
minc=32000;  
  
for(i=0;i<200;i++)  
  
{  
  
//Serial.println(current[i]); //raw current display on serial monitor  
  
if(current[i]<minc) minc=current[i];  
  
if(current[i]>maxc) maxc=current[i];  
  
rmsc=(0.079*(maxc-minc)/(2*1.4142)) - 0.08;  
  
}  
  
Serial.println("Current=");  
  
Serial.println(rmsc);  
  
power=voltageB*rmsc;  
  
Serial.println("Power=");  
  
Serial.println(power);
```

```
// for voltage
for(i=0;i<200;i++)
volt[i] = analogRead(A0);

maxc=0;
minc=32000;
for(i=0;i<200;i++)
{
//Serial.println(current[i]); //raw current display on serial monitor
if(volt[i]<minc) minc=volt[i];
if(volt[i]>maxc) maxc=volt[i];
rmsc_volt=(0.079*(maxc-minc)/(2*1.4142)) - 0.08;
}
Serial.println("Voltage=");
Serial.println(rmsc_volt);
power_volt=voltageB*rmsc_volt;
Serial.println("Power=");
Serial.println(power_volt);

//gsm will sent the value of voltageB [voltage] and voltageA[current]
}
```

```
void SendMessage(String msg)
{
    SIM900.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

    delay(1000); // Delay of 1000 milli seconds or 1 second

    SIM900.println("AT+CMGS=\"+8801770722247\"\\r"); // Replace x with mobile number

    delay(1000);

    SIM900.println(msg); // The SMS text you want to send

    delay(100);

    SIM900.println((char)26); // ASCII code of CTRL+Z

    delay(1000);
}
```

References

1. <https://www.arduino.cc/en/tutorial/potentiometer>
2. <https://www.brainy-bits.com/potentiometer-values/>
3. https://en.wikipedia.org/wiki/Current_sensor
4. <https://en.wikipedia.org/wiki/Transformer>
5. <https://electronicsforu.com/resources/gsm-module>
6. <https://www.arduino.cc/en/Guide/ArduinoGSMShield>
7. <https://electronics.stackexchange.com/questions/53122/how-to-use-a-transformer-to-power-a-5v-chip-an-atmega328>
8. <https://electronics.stackexchange.com/questions/181513/reading-current-transformer-using-arduino>
9. <http://www.smartmeters.vic.gov.au/about-smart-meters/reports-and-consultations/advanced-metering-infrastructure-customer-impacts-study-volume-1/3-objective-and-methodology>
10. <http://engineerexperiences.com/objectives-energy-meter.html>
11. https://en.wikipedia.org/wiki/Electricity_meter