International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

DESIGN AND IMPLEMENTATION OF REAL TIME PRICING SYSTEM IN SMART GRIDS

Taiwo Samuel Aina[1], Oluwaseun.O. Akinte[2]

[1]School of Computer Science and Technology, University of Bedfordshire, UK.
[2]Department of Electrical and Electronics Engineering, Coventry University, U.K

Abstract - This paper focuses on incorporating a real time pricing in smart grids system. This scheme of real time pricing makes it possible to manage the load from the demand side or consumer side. This would provide several benefits to the domestic consumers too, apart from industrial consumers. In addition to conservation, real time pricing also comes along way in protecting and preserving the quality of equipment used by the power utility company. Real time distribution aims towards a uniform distribution of loads throughout the day. Hence, no need to produce very large power for short duration of spikes.

Keywords: Smart grid System, Billing, Pricing, IoT, GSM, Wi-Fi, Real time

1.0 Introduction

The fundamental need of our life is energy because of mechanical growth and urbanization. One of the solutions to energy catastrophic is to control, analyze and reduction of electrical power consumption in households. For real time pricing in smart grid, communication is being done by the use of GSM and Ad-Hoc wireless routing protocol with connection of each home using a radio frequency system, all readings from energy meters may be accessed via a wireless connection to the central/utility office [1]. Electric vitality is assessed by individuals in the traditional electro-mechanical and computerized metering framework; they frequently arrange the bill through suspicion based on its history of power usage [2]. It is possible that the buyer have not used the same amount of electricity in the current month alongside the previous months for a variety of reasons, such as going on vacation or being in the hospital. This charging technique is also inappropriate for a power supply company since it provides an inaccurate picture of overall power usage in the buyer's area which could lead to errors in future planning by the company [2]. Currently, Bluetooth, GPRS and the Global System for Mobile communication (GSM) are being used to develop a variety of Automatic Meter Reading (AMR) systems [1]. In study, an energy meter is defined as a device that uses GSM technology to share information about energy consumption over a long distance. However, if tampered, SMS notifications may go unnoticed, lowering the system's reliability and efficiency. For long-distance two-way communication between utilities and consumers, GPRS is challenging to implement.

Smart energy meters make use of embedded system features such as a combination of hardware and software to implementation required functionality [3].

In this paper we compared Arduino and other controllers, as well as the use of GSM and Wi-Fi modems to establish the concept of real time pricing in smart grids system which enables consumers and service providers to receive the spent energy reading with the corresponding amount by GSM modem. Consumers will also receive notification in

the form of text messages via GSM when they are about to hit their threshold value, indicating that they have reached it.

2.0 Methodology

In this section, presentation of the method and steps used in designing and implementing the project is being applied. It includes the description of the design architecture and detailed explanation of the software development as well as hardware development of the device**.**

2.1 Hardware Construction

The step by step approach taken in the construction of this project started with the following hardware components.

- Arduino-Uno
- ESP12/NodeMCU
- ACS712-30Amp Current sensor
- 1.0mm Male-Female jumper wires
- 1.0mm Male-Male jumper wires
- Double 13amp sub-circuit outlet
- 13amp plug
- 1.5mm copper conductor
- Single channel relay
- 100watt lighting bulb

Construction of this project can be sub-divided into three categories namely: -

- Housing/Casing construction
- Electronic construction
- Software construction

Two double pattress made with a plastic material ware use for the casing in which it is join together with hot glue gum to ensure firmness of both material, One part of the pattress houses the microcontroller, relay, electric sensor, and A.C-D.C converter while the second part of the pattress houses the double 13amp socket outlet to ensure portability this project.

Figure 1: The housing

The construction of electronic system always depends on the transfer function of the component in which the input and the output current signal for every component must be precisely known to avoid electronic hazard. Firstly the use of 220volts to 9volts A.C to D.C converter as an input to power the project, the 220volts was step down to 9volts by the converter which is been connect in series IC7805 D.C to D.C converter to step the voltage down to 5volt D.C to power the other component in this project work.

The first connection after the power layout is the connection of the microcontroller to the output of the current sensor which has three pins; the 5volt, Ground, and out pin. The microcontroller is being connected to the 5volt terminal of the current sensor to power and process the given instruction according to the code.

The other component such as relay are connected to the microcontroller, the relay has five pins which are: normally open, normally closed, common, coil, and the control pin, the common will be connected to 220volts while the common will be connected to the 13Amps socket outlet,

once the relay is activated from the microcontroller it will switch from normally open to normally close thereby allowing the flow of current to the 13Amps socket outlet, the control pin of the relay is been connected to the microcontroller so as to be controlled, once the billing in the meter has attained the threshold as programmed, the microcontroller will activated the relay to normally open to stop the flowing of current to the socket outlet which will automatically switch off the connected appliances

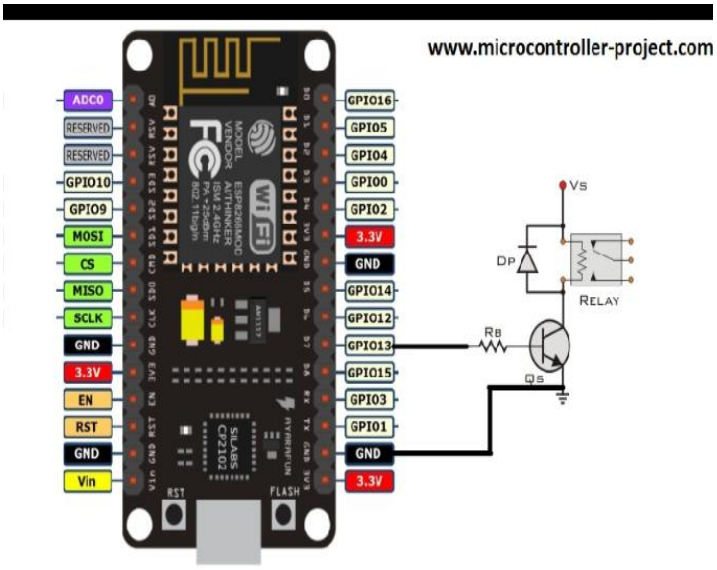

Figure 2: Connect of microcontroller to the Relay

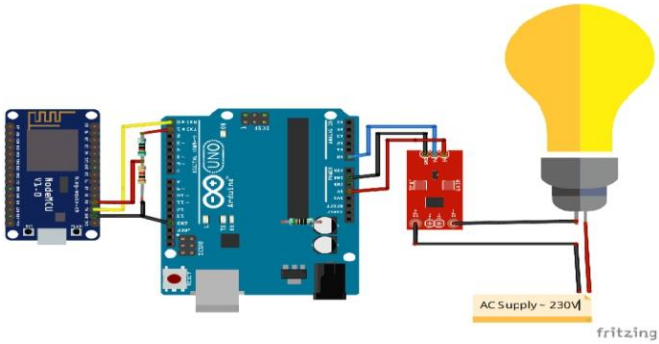

Figure 3: Total circuit connection

2.2 Software Construction

An online IoT (Internet of Things) platform call Blynk was used on the blynk there is access of creating many widget which was done, so credit is being purchased in form of energy on the platform to access the widget, the widget is now what was used to create an interface to control the automated energy meter on the blynk application, two gauges, two value display and four sliders was used.

Figure 4: Blynk Application interface

3.0 The Programming Language

Arduino programming language was used to code the automated smart energy meter, the programming language is the combination on two different programming languages, which are C and C++, this was adopted because of the microcontroller been used for the project work to ensure compatibility of the system for proper functionality.

Figure 6: Arduino Programming interface

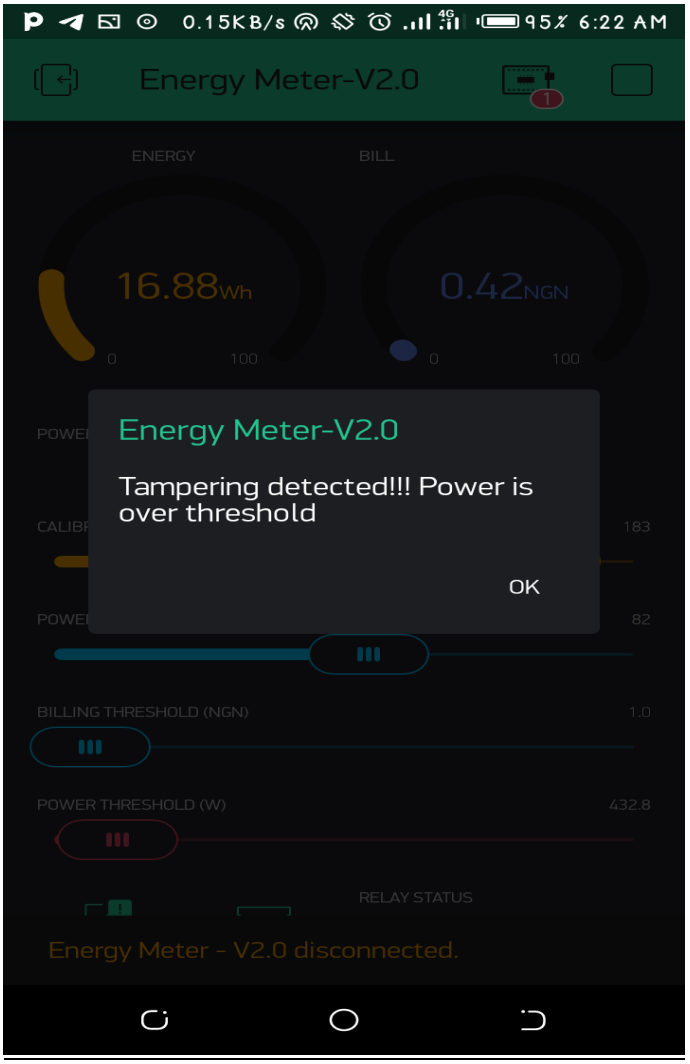

Figure 5: Notification Display

3.1 The Programming Code

```
//#define BLYNK_DEBUG
#define BLYNK_PRINT Serial
#include <ArduinoOTA.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define CLOUD  // comment out for local server

#include <SPI.h>              // include SPI library
#include <Adafruit_GFX.h>     // include adafruit graphics library
#include <Adafruit_PCD8544.h>  // include adafruit PCD8544 (Nokia 5110) library

// Nokia 5110 LCD module connections (CLK, DIN, D/C, CS, RST)
Adafruit_PCD8544 display = Adafruit_PCD8544(D4, D3, D2, D1, D0);

BlynkTimer timer;

charauth[]      = "xxxx";
charssid[]      = "xxxx";
char pass[]     = "xxxx";
char server[]   = "blynk-cloud.com";   // ip or domain
charmyhostname[] = "Energy-Meter-V2.0";  // for OTA and router identification


constintSensor_Pin = A0;
unsignedint Sensitivity = 185;   // 185mV/A for 5A, 100 mV/A for 20A and 66mV/A for 30A Module
floatVpp = 0; // peak-peak voltage
```

```
floatVrms = 0; // rms voltage
floatIrms = 0; // rms current
floatSupply_Voltage = 233.0;          // reading from DMM
floatVcc = 5.0;        // ADC reference voltage // voltage at 5V pin
float power = 0;        // power in watt
floatWh =0 ;            // Energy in kWh
unsigned long last_time =0;
unsigned long current_time =0;
unsigned long interval = 100;
unsignedint calibration = 100;  // V2 slider calibrates this
unsignedint pF = 85;          // Power Factor default 95
floatbill_amount = 0;   // 30 day cost as present energy usage inclapprox PF
unsignedintenergyTariff = 8.0; // Energy cost in INR per unit (kWh)

void getACS712() {  // for AC
Vpp = getVPP();
Vrms = (Vpp/2.0) *0.707;
Vrms = Vrms - (calibration / 10000.0);     // calibtrate to zero with slider
Irms = (Vrms * 1000)/Sensitivity ;
if((Irms> -0.015) && (Irms< 0.008)){  // remove low end chatter
Irms = 0.0;
  }
power= (Supply_Voltage * Irms) * (pF / 100.0);
last_time = current_time;
current_time = millis();
Wh = Wh+   power *(( current_time -last_time) /3600000.0) ; // calculating
energy in Watt-Hour
bill_amount = Wh * (energyTariff/1000);
Serial.print("Irms:  ");
Serial.print(String(Irms, 3));
Serial.println(" A");
Serial.print("Power: ");
Serial.print(String(power, 3));
Serial.println(" W");
Serial.print("  Bill Amount: INR");
Serial.println(String(bill_amount, 2));
Blynk.virtualWrite(V0, String (Wh));  // gauge
Blynk.virtualWrite(V1, String(bill_amount, 2));
Blynk.virtualWrite(V2, String(power,2));
Blynk.virtualWrite(V3, String(Irms, 3));
}

floatgetVPP()
{
float result;
intreadValue;
intmaxValue = 0;
intminValue = 1024;
  uint32_t start_time = millis();
while((millis()-start_time) < 950) //read every 0.95 Sec
  {
Read Value = analog Read(Sensor Pin);
if (read Value>max Value)
    {
Max Value = read Value;
    }
if (read Value<min Value)
    {
Min Value = read Value;
    }
  }
result = ((max Value – min Value) * Vcc) / 1024.0;
return result;
 }

 BLYNK_WRITE(V4) {  // calibration slider 50 to 200
calibration = param. as Int();
   }
BLYNK_WRITE(V5) {  // set supply voltage slider 70 to 260
Supply Voltage = param.asInt();
   }
BLYNK_WRITE(V6) {  // PF slider 60 to 100 i.e 0.60 to 1.00, default 85
pF = param.asInt();
   }
BLYNK_WRITE(V7) {  // Energy tariff slider 1 to 20, default 8 (Rs.8.0 / kWh)
Energy Tariff = param.asInt();
}
BLYNK_WRITE(V8) {  // set 5, 20 or 30A ACS712 sensor with menu
switch (param.asInt())
   {
case 1: {      // 5A Sensitivity = 185;break;}
case 2: {      // 20A Sensitivity  = 100; break; }
case 3: {      // 30A Sensitivity  = 66; break; }

default: {     // 5A Sensitivity  = 185; }
}

Void display data() {
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(1);
display.setCursor(10, 3);
display.println("Pow  : ");
display.setCursor(50, 3);
display.println(power);
display.setCursor(85, 3);
display.println("W");

display.setCursor(10, 13);
display.println("Ener : ");
display.setCursor(50, 13);
display.println(Wh);
display.setCursor(85, 13);
display.println("Wh");

display.setCursor(10, 23);
display.println("Bill : ");
display.setCursor(50, 23);
display.println(bill_amount);
display.setCursor(85, 23);
display.println("INR");
display.display();
}

void setup() {
display.begin();
WiFi.hostname(myhostname);
Serial.begin(115200);
Serial.println("\n Rebooted");
WiFi.mode(WIFI_STA);
  #ifdef CLOUD
Blynk.begin(auth, ssid, pass);
  #else
Blynk.begin(auth, ssid, pass, server);
  #endif
while (Blynk.connect() == false) {}
ArduinoOTA.setHostname(myhostname);
ArduinoOTA.begin();
timer.setInterval(2000L, getACS712); // get data every 2s
}

BLYNK_CONNECTED(){
Blynk.syncAll();
}
void loop() {
displaydata();
Blynk.run();
ArduinoOTA.handle();
timer.run();
}
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:

}
```

3.2 Result of the Programming Language

Real Time pricing Smart Grids system has been achieved in this project by simulating results on a PC with windows operating system, Intel Core i3 processor, 500Gbyte hard disk, and 4Gbyte RAM. The followings tests are done to prove the proposed implementation the project.

At No load: -

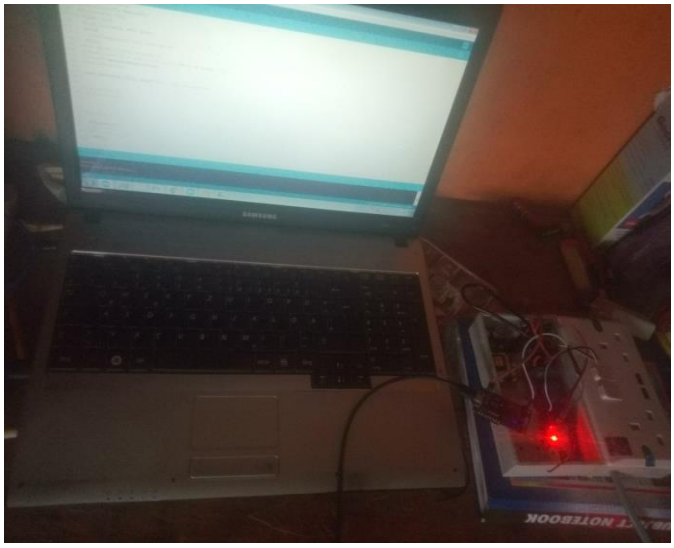

Figure 7: Testing ground

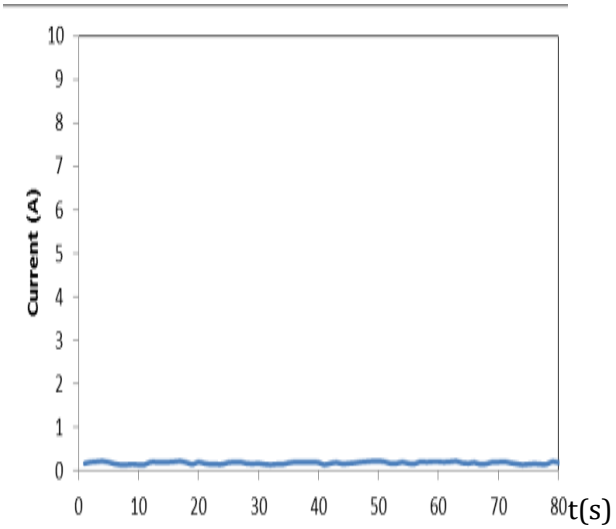

Figure 9: Graphical representation of the current with time at no load

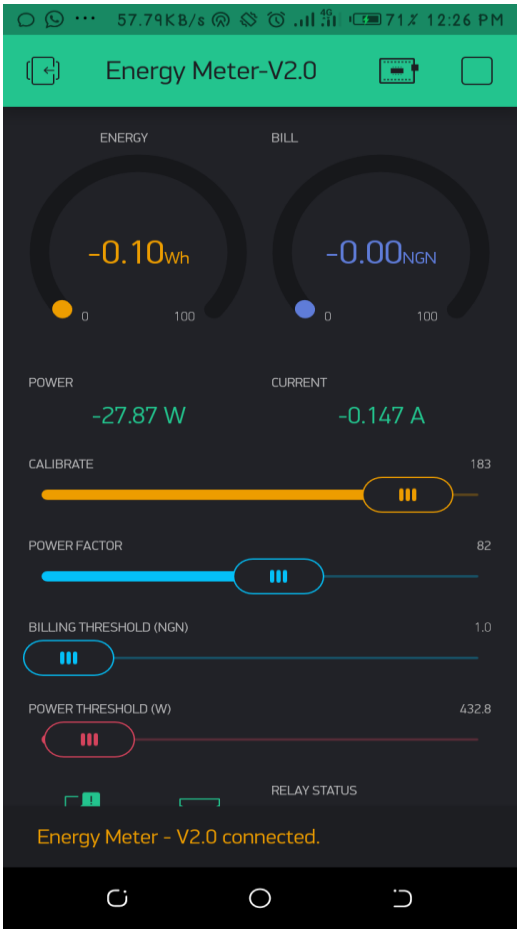

Figure 8: The readings at no load

The result show that the instrument is valid and reliable at no load

3.3 At load: Laptop and fan

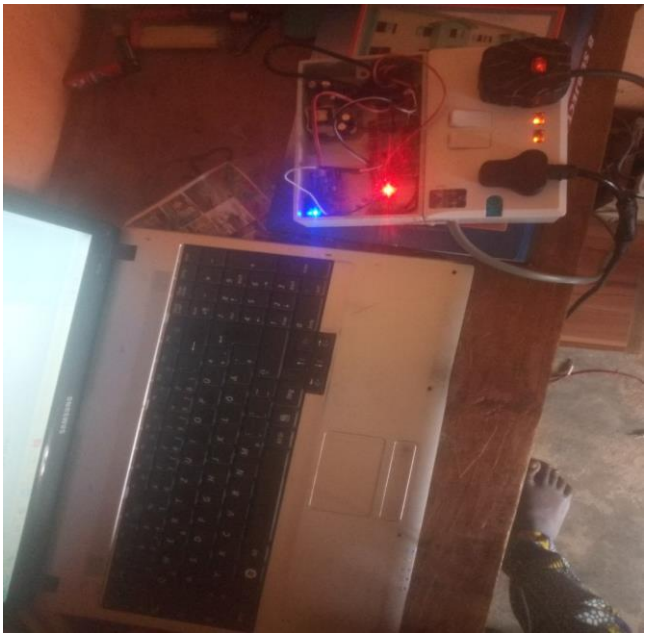

Figure 10: Testing ground

The reading show at the interface is as follow:-

Figure 11: Display Readings at load

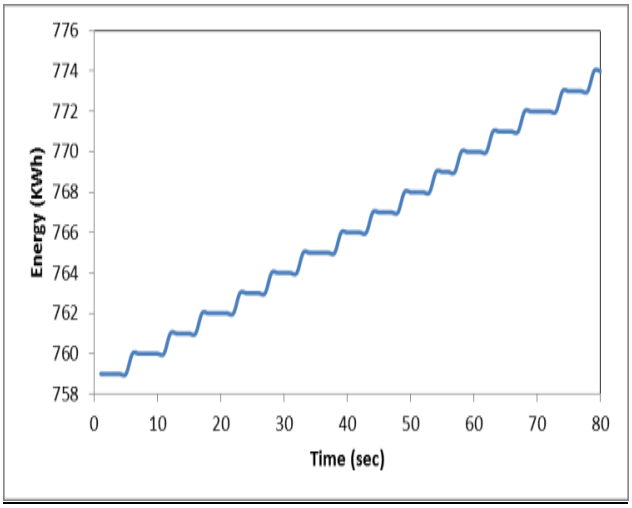

Figure 12: Graphical representation of current at load

Therefore from the observation of the reading it is shown that the power is increasing with respect to time which lead to increase in billing of the energy consumed by the end user.

3.4 At load above threshold level: laptop and fan

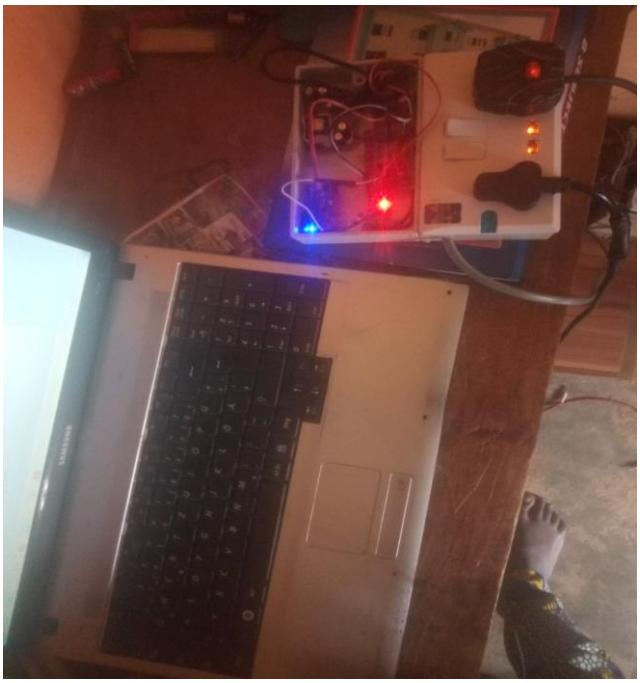

Figure 13: Testing ground

At the level beyond the threshold frequency the following notification was seen on the dashboard of the meter:-

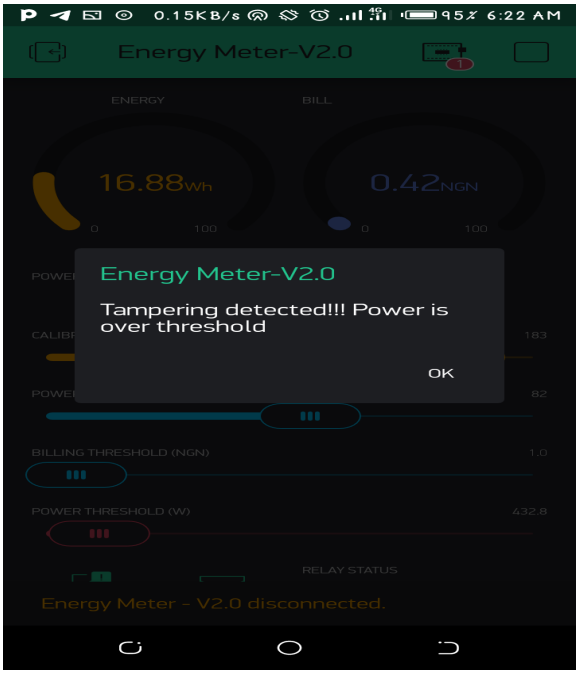

Figure 14: Display at overload

Figure 15: Display at billing threshold

From the above reading it shows that the automated energy is valid and a reliable instrument to detect theft and secure communication.

4.0 Introducing Smart Grid to the Future Electrical Power Grid Infrastructure

Applying smart grid technology to the future electric supply will ensure the reliability of the grid to levels never thought possible, Allow for the advancement and efficiencies yet to be envisioned, exert downward pressure on electricity prices, maintain the affordability for energy consumers, Provide consumers with greater information and choice of supply, accommodate renewable and traditional energy resources, enable higher penetration of intermittent power generation sources, revolutionize not only the utility sector but the transportation sector through the integration of electrical vehicles as generation and storage devices, the Smart Grid will promote environmental quality by allowing customers to purchase cleaner, lower-carbon-emitting generation, promote a more even deployment of renewable energy sources and allow access to more environmentally friendly central station generation. Furthermore, the Smart Grid will allow for more efficient consumer response to prices, which will reduce the need for additional fossil fuel-fired generation capacity, thereby reducing the emission of CO2 and other pollutants.

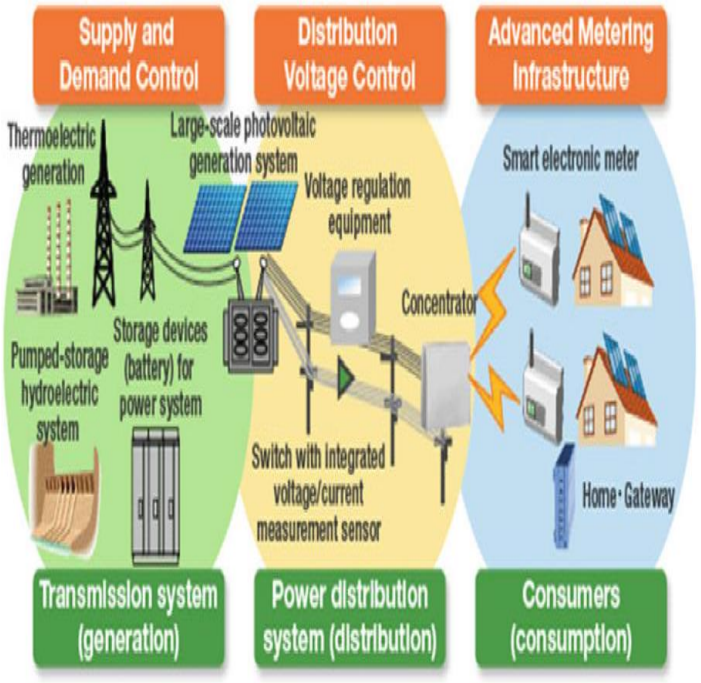

Fig 16: Smart Grid [11]

5.0 CONCLUSION

This research work is based on the difficulties and the limitations that exist in the manual system for monitoring these systems and we therefore embarked on this work to solve the problems noticed. This research work has demonstrated how to get a fully functional embedded product for monitoring the status of an electrical power usage in households and communicate the information to the user through software with minimal error conditions. This is implemented through the utilization of a dedicated electronic circuit, microcontroller and its assembly, and a graphical user interface installed at the consumer

premises.

References

1) Milanpreet Kaur,et al(2018). Implementation of Smart Metering based on Internet of Things

2) A.Subba Rao, and Sri VidyaGarige (2019). IOT Based Smart Energy Meter Billing Monitoring and Controlling the Loads

3) Birendrakumar Sahani1,et al (2017). IoT Based Smart Energy Meter

4) International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 04 Issue: 04 | Apr -2017 www.irjet.net

5) N M Yoeseph, M A Safi'ie, and F A Purnomo(2019). Smart Energy Meter based on Arduino and Internet of Things

6) Dr. R. Kalaivani, and A. Kaaviya sri(2017). Simulation of Smart Meter Using Proteus software for Smart Grid International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 04 www.irjet.net

7) Kofi Fofie (2020). analysis of electrical power usage in houses using smart electrical distribution switch.

8) N M Yoeseph, M A Safi'ie, and F A Purnomo(2019). Smart Energy Meter based on Arduino and Internet of Things

9) Dr. R. Kalaivani , and A. Kaaviya sri(2017). Simulation of Smart Meter Using Proteus software for Smart Grid International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 04 www.irjet.net

10) Oyubu A. O. Nwabueze C.A(2015). Design and testing of a smart energy metering system based on GSM model

11) Alonzo Sierra et al(2019). Simulation-Supported Testing of Smart Energy Product Prototypes Smart Grid (Mitsubishielectric.com 2016)