Smart Energy Meter

Semester 4 – Individual Project

P.B. Guruge 160191N EE3203

Smart Energy Meter and Grid Monitoring System

Semester 4 – Individual Project

EE3203

Pasan Bhanu Guruge

160191N

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

This report describes about a smart energy meter and grid monitoring system which is capable of monitor energy consumption of households using CT Sensors, Arduino and Online Server System. A smart energy meter is an electronic device that records consumption of electric energy and communicates the information to the electricity supplier for monitoring and billing. The monitoring device contains a current transformer, AC-AC adapter, Arduino, WiFi Router and ESP8266 WiFi module. The device measure energy consumption in watts second (Ws) and upload to the server in each minute time. This system has 2 main sensors. YHDC SCT-013-000 current transformer as the current sensor which is capable of measuring current up to 100 amps. 230-9 V AC-AC adapter used to measure voltage. EMON Library is the main power calculation library used in the Arduino, which is developed by Open Energy Monitor Systems. This library automatically calculates the voltage, current and power based on the analogue inputs to Arduino.

The spatiality of this device is it can be plugged in to any existing energy monitoring device to enable the remote monitoring capabilities. Due to the use of current transformers, current can be measured without touching the high current wires. This increase the safety of the device as well as users.

ACKNOWLEDGEMENT

Apart from the efforts of myself, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project. I would like to show my greatest appreciation to Dr. Udayanga Hemapala. I can't say thank you enough for his tremendous support and help. I feel motivated and encouraged every time I attend his lectures. Without his encouragement and guidance this project would not have materialized. The guidance and support received from all the friends and colleagues, was vital for the success of the project. I am grateful for their constant support and help.

Chapter 1

INTRODUCTION

Although electricity was considered a luxury in the past, today it has become an essential necessity with the enhanced living standards of people and growth of the technology and industrial sector. Therefore, the electricity demand in most of countries of the world is growing day by day. To meet this demand most of the developing countries have added more fossil fuel generation to their systems as they are the low risk proven technologies in large scale despite the environmental impact caused by them. Current trends in power generation and use are patently unsustainable economically, environmentally and socially. Without decisive action, increased fossil fuel demand will heighten concerns over the security of supplies and energy related emissions of carbon dioxide (CO_2). In today's world most of the developing countries are already swamped with power crisis due to inadequacy of generation to meet the demand.

Therefore, there is a pressing need to implement mechanisms to manage the growing demand and improve the efficiency while reducing the energy losses, accelerating the development of low-carbon energy technologies such as renewable power generation, in order to address the global challenges of energy security, climate change and economic growth.

Traditional Energy Meters, which can only record the energy consumption over the time gives no data to the suppliers to generate or maintaining the power generation effectively. Therefore, the necessity of Advanced Metering Infrastructure (AMI) has been emerged to address the above matters. So, Smart Energy Meters are now start using in many countries to upgrade the monitoring capabilities of electricity. (Lanka, 2013)

Smart Energy Meter has no exact definition, but it basically refers to using Advanced Metering Infrastructure in conjunction with faster automated communication systems to allow customers to monitor their energy consumption in real time. (Lanka, 2013) A smart meter electronically measures how much energy is being used and how much it costs, and then communicates it to the energy supplier and the customer via internal network. This enables the remote monitoring capabilities, live reports, precise estimations and valuable information to the electricity generation companies. Also reduce the labor cost for meter reading is an added advantage of using these systems.

MOTIVATIONS

In Sri Lanka, Smart Energy Meters introduced in 2011 only for industrial users. Those meters have facility to measure, store and remote monitoring. But remote monitoring only functioning in Colombo City Area for industrial users. For an example dialog uses their own remote energy monitoring system to track the usage of power consumption in Signal Towers. But these meters are not used for home users. Also, Ceylon Electricity Board (CEB) spent millions of rupees for meter readers and maintenance staff. This system can address the above problems.

OBJECTIVES AND SCOPE

Objective of this system is to develop an addon device for existing analog energy monitors to enable remote monitoring capabilities.

This system has features of,

- Easy to install
- Low design and maintenance cost
- Lite server architecture used
- Inbuilt Wi-Fi connection

ORGANISATION

This report contains details about Current Transformer Sensors and monitoring methods, voltage monitoring methods, ESP8266 Wi-Fi Module and emon library for power calculation using Arduino.

Chapter 2

LITERATURE REVIEW

Introduction

A smart meter is an electronic device that records consumption of electric energy and communicates the information to the electricity supplier for monitoring and billing. Smart meters typically record energy hourly or more frequently, and report at least daily. Smart meters enable two-way communication between the meter and the central system. (Open Energy Monitor, 2019)

Communications from the meter to the network may be wireless, or via fixed wired connections such as power line carrier (PLC). Wireless communication options in common use include cellular communications, Wi-Fi, wireless ad hoc networks over Wi-Fi, wireless mesh networks, low power long range wireless, ZigBee, and Wi-SUN.

History

In 1972, Theodore Paraskevakos, worker of Boeing developed a sensor monitoring system that used digital transmission for security, fire, and medical alarm systems as well as meter reading capabilities. This technology was a spin-off from the automatic telephone line identification system, now known as Caller ID.

In 1974, Paraskevakos was awarded a U.S. patent for this technology. In 1977, he launched Metretek, Inc., which developed and produced the first fully automated, commercially available remote meter reading and load management system.

Current Sensor

Figure 1: CT Sensor

Current transformers (CTs) are sensors that measure alternating current (AC). They are particularly useful for measuring whole building electricity consumption or generation.

The split core type, such as the CT in the picture above can be clipped onto either the live or neutral wire coming into the building, without the need to do any high voltage electrical work.

As a usual transformer, a current transformer has a primary winding, a magnetic core, and a secondary winding.

AC Voltage Monitoring

An AC voltage measurement is needed to calculate real power, apparent power and power factor. This measurement can be made safely (requiring no high voltage work) by using an AC to AC power adaptor. The transformer in the adapter provides isolation from the high voltage mains.

Figure 2: AC-AC Voltage Adapter

AC-AC power adapter scale down the waveform and the output fed to analog to digital converter. The analog to digital converter convert sin wave to a digital signal and fed to microcontroller. Microcontroller use several methods to calculate the voltage. This project peek to peek value measuring used for voltage calculation.

ESP8266 WiFi Module

The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by Shanghai-based Chinese manufacturer Espressif Systems.

Application Programming Interface

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Arduino and web application communicate with each other using API. API allows devices to send data securely without exposing it’s internal processes.

Chapter 3

DESIGN PRINCIPLES

Measure Current with CT Sensors

To connect a CT sensor to an Arduino, the output signal from the CT sensor needs to be conditioned so it meets the input requirements of the Arduino analog inputs, i.e. a positive voltage between 0V and the ADC reference voltage.

This can be achieved with the following circuit which consists of two main parts:

- The CT sensor and burden resistor
- The biasing voltage divider (R1 & R2)

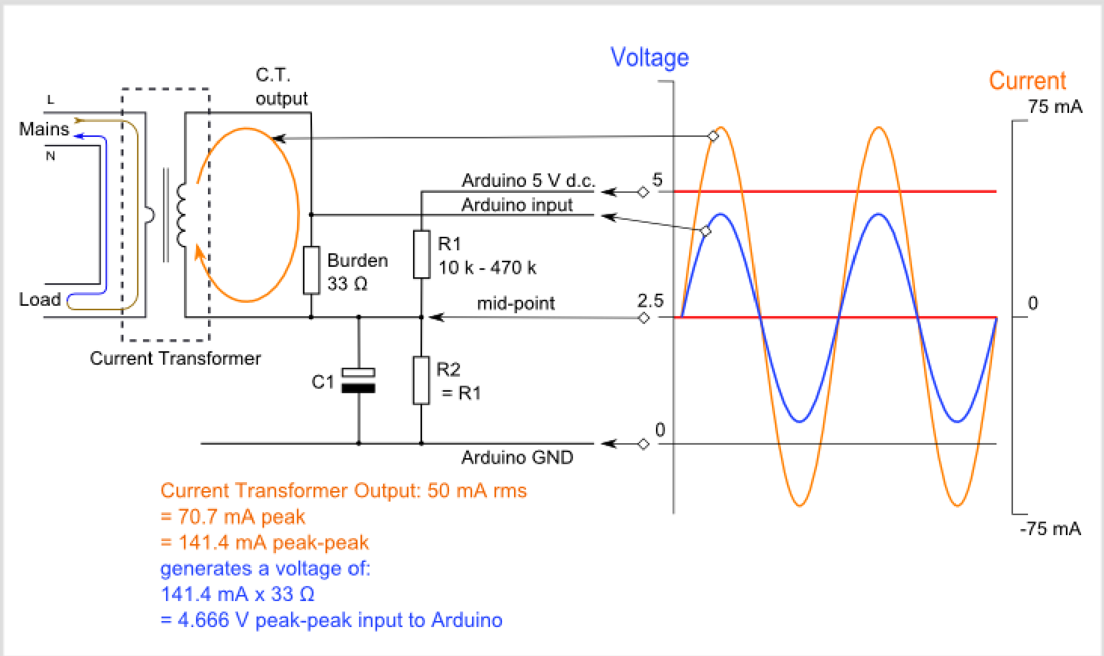

Figure 3: CT Sensor Output

Calculating the Burden Resistor

If the CT sensor is a "current output" type such as the YHDC SCT-013-000, the current signal needs to be converted to a voltage signal with a burden resistor.

a) Choose the current range you want to measure

The **YHDC SCT-013-000 CT** has a current range of 0 to 100 A. Let's choose 100 A as our maximum current.

Model	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Input current	0-100A	0-5A	0-10A	0-15A	0-20A
Output mode	Current/33m A	Voltage/1V	Voltage/1V	Voltage/1V	Voltage/1V
Model	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-070
Input current	0-25A	0-30A	0-50A	0-60A	
Output mode	Voltage/1V	Voltage/1V	Voltage/1V	Voltage/1V	

Figure 4: SCT Sensor Models

b) Convert maximum RMS current to peak-current by multiplying by $\sqrt{2}$.

Primary peak-current = RMS current $\times$ $\sqrt{2}$ = 100 A $\times$ 1.414 = 141.4A

c) Divide the peak-current by the number of turns in the CT to give the peak-current in the secondary coil.

The YHDC SCT-013-000 CT has 2000 turns, so the secondary peak current will be:

Secondary peak-current = Primary peak-current / no. of turns = 141.4 A / 2000 = 0.0707A

d) To maximize measurement resolution, the voltage across the burden resistor at peak-current should be equal to one-half of the Arduino analog reference voltage. (AREF / 2)

If you're using an Arduino running at 5V: AREF / 2 will be 2.5 Volts. So the ideal burden resistance will be:

Ideal burden resistance = (AREF/2) / Secondary peak-current = 2.5 V / 0.0707 A = 35.4 Ω

35 Ω is not a common resistor value. The nearest values either side of 35 Ω are 39 and 33 Ω. Always choose the smaller value, or the maximum load current will create a voltage higher than AREF. Recommend burden resistor is 33 Ω ±1. In some cases, using 2 resistors in series will be closer to the ideal burden value. The further from ideal the value is, the lower the accuracy will be.

Here are the same calculations as above in a more compact form:

Burden Resistor (ohms) = (AREF * CT TURNS) / (2$\sqrt{2}$ * max primary current)

Measure Voltage

The output signal from the AC voltage adapter is a near-sinusoidal waveform. If you have a 9V (RMS) power adapter the positive voltage peak be 12.7V, the negative peak -12.7V. However, due to the poor voltage regulation with this type of adapter, when the adapter is un-loaded (as in this case), the output is often 10-12V (RMS) giving a peak voltage of 14-17V. The voltage output of the transformer is proportional to the AC input voltage.

The signal conditioning electronics needs to convert the output of the adapter to a waveform that has a positive peak that's less than 5V and a negative peak that is more than 0V.

So we need to

- Scale down the waveform and
- Add an offset to omit negative component.

The waveform can be scaled down using a voltage divider connected across the adapter's terminals, and the offset (bias) can be added using a voltage source created by another voltage divider connected across the Arduino's power supply .

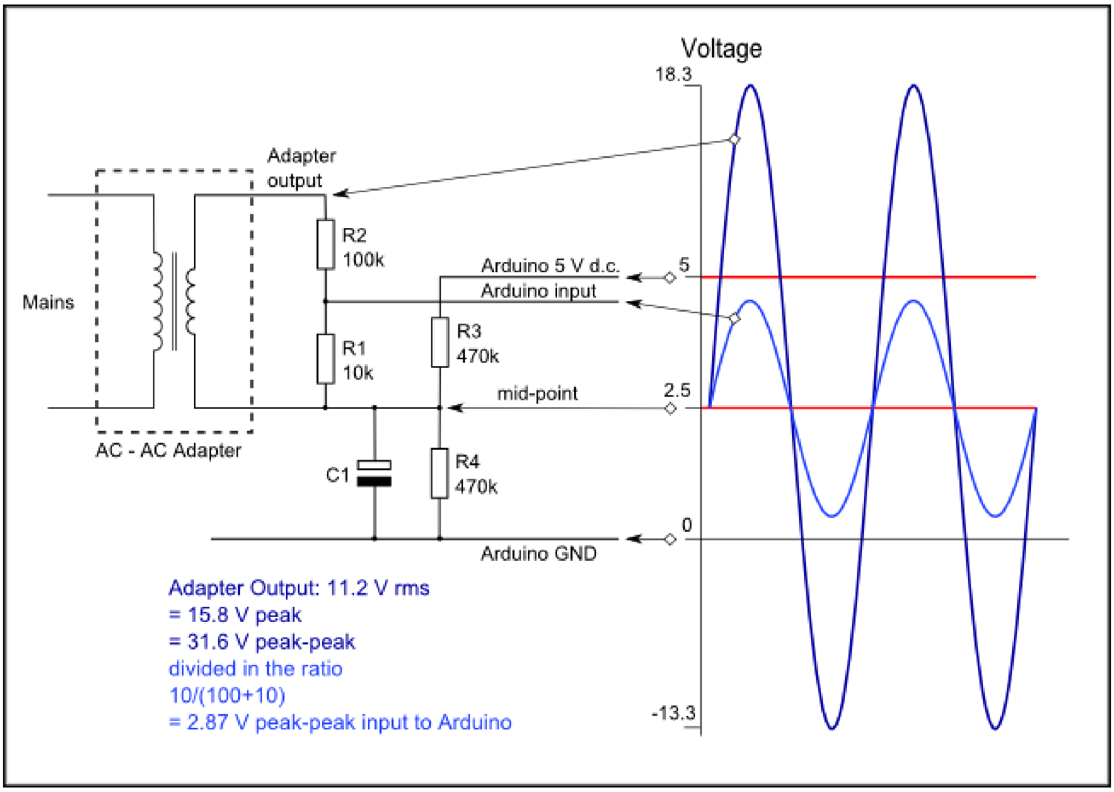

Figure 5: Circuit Diagram and the Voltage Waveforms

Resistors R1 and R2 form a voltage divider that scales down the power adapter AC voltage. Resistors R3 and R4 provide the voltage bias. Capacitor C1 provides a low impedance path to ground for the AC signal. The value is not critical, between 1 µF and 10 µF will be satisfactory.

R1 and R2 need to be chosen to give a peak-voltage-output of ~1V. For an AC-AC adapter with an 9V RMS output, a resistor combination of 10k for R1 and 100k for R2 would be suitable.

Peak Voltage Output = R1 / (R1 + R2) x Peak Voltage Input

= 10k / (10k + 100k) x 12.7V = 1.15V

The voltage bias provided by R3 and R4 should be half of the Arduino supply voltage. As such, R3 and R4 need to be of equal resistance. Higher resistance lowers energy consumption.

Calculate Power and Send to Server

The device calculate power per one minute and send it to the server through as API. In build wifi adapter used for the communication. ESP8266 module used for the Arduino to establish the connection.

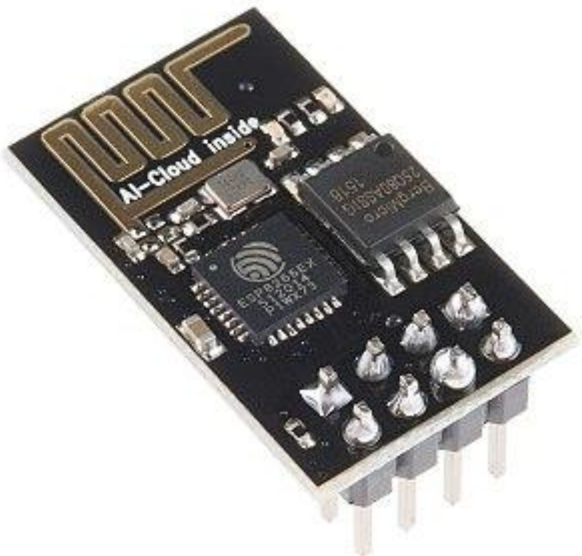

Figure 6: ESP8266 Module

Power calculated in watts second (Ws) by voltage x current equation. The server displays the information in standard units. Realtime monitor display voltage, current and wattage of the system.

REALTIME MONITORING

Device ID	Voltage (V)	Current (A)	Power (Ws)	Update Time
0001	219.84	0.05	10.992	08:11:39

API was developed using the Slim Framework, a specialized framework for HTTP REST API development. This framework can be used to build lightweight applications which even can run on a free server.

The backend was developed using PHP (Hypertext Pre-Processor) language with advanced AJAX technologies. The backend and API are in commercial standards which can be implemented in and server infrastructure.

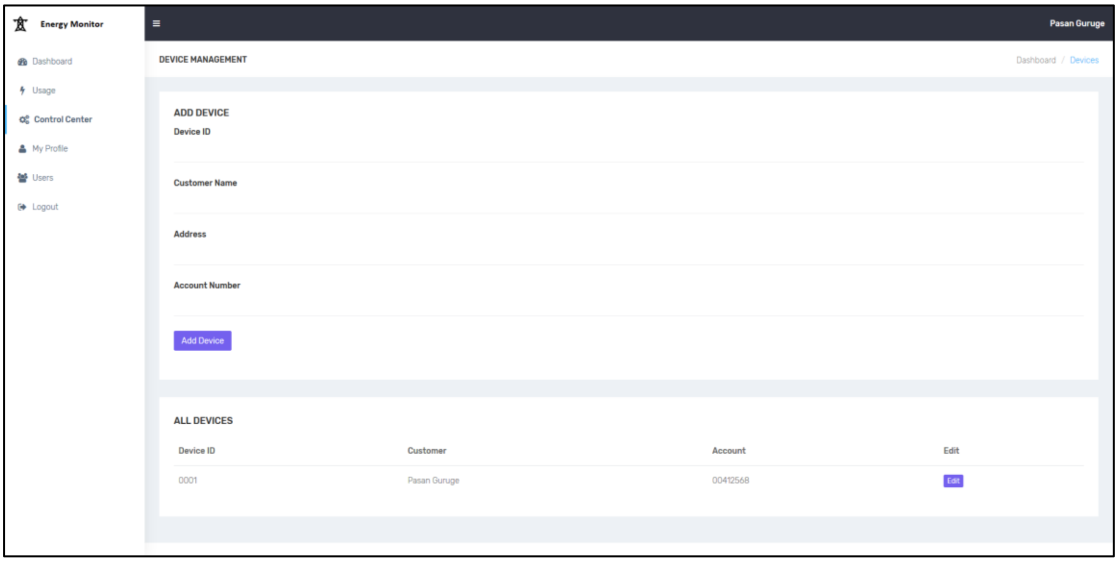

Figure 7: Screenshot of Online Monitoring System

Specialties of the Online Systems are,

- Multi user support
- Unlimited devices can be added
- Daily logging
- Reports

Chapter 4

CONCLUSIONS AND FUTURE WORK

Conclusions

This project primary target is to build a pluggable device for existing energy monitoring systems to enable remote monitoring feature. Current Transformers used to measure current and AC-AC power adapter used for voltage measurement. Arduino interfacing with ESP8266 build the calculation and communication platform. Communication with the main servers is happens using the inbuild wifi connection and API used to securely communicate Arduino and online monitoring system.

This system can be used to monitor the energy usage of the households and send automated electricity bills. This avoid the need of meter readers and this system upgrade the traditional billing systems. Online Energy Monitoring systems are not very common in Sri Lanka due to it's high initial cost but this system minimize the initial cost as well as the backbone network cost.

Recommendation in Future Work

We used a free server for the web application which is not recommended for IoT applications. We can use optimized web server for the web application.

Current sensor has considerable error in low currents. This can be minimized by using 2 current sensors for 2 ranges.

This device, Wi-Fi adapter in not included. In commercial production, it should be in the device.

REFERENCES

(2019, 01 11). Retrieved from Open Energy Monitor: https://openenergymonitor.org/

CT sensors - Interfacing with an Arduino. (2012, 02 20). Retrieved from Open Energy Monitor: https://openenergymonitor.org/forum-archive/node/156.html

Feynman. (2016, 11 07). *Non-Invasive Current sensor SCT-013-000*. Retrieved from Scidle: https://scidle.com/non-invasive-current-sensor-sct-013-000/

Feynman. (2018, 10 30). *How to use Non-invasive AC Current Sensors with Arduino*. Retrieved from Scidle: https://scidle.com/how-to-use-non-invasive-ac-current-sensors-with-arduino/

Grant, C. (2018, 04 20). *Using the ESP8266 WiFi Module with Arduino Uno publishing to ThingSpeak*. Retrieved from Medium: https://medium.com/@cgrant/using-the-esp8266-wifi-module-with-arduino-uno-publishing-to-thingspeak-99fc77122e82

Lanka, P. U. (2013). *Applicability of Smart Metering Technology in Sri Lanka.* Colombo: Public Utilities Commission of Sri Lanka.

LBO. (2016, 01 08). Sri Lanka introduces Smart Metering to reduce electricity generation costs. *Lanka Business Online*.

Santos, R. (2015, 10 08). *Sending Data From an Arduino to the ESP8266 via Serial*. Retrieved from Randomnerd Tutorials: https://randomnerdtutorials.com/sending-data-from-an-arduino-to-the-esp8266-via-serial/

Tech, J. L. (2019, 01 07). *JLanka launches Sri Lanka's first ever Electricity Monitoring ePRO1000 Smart System*. Retrieved from J Lanka Tech: https://www.jlankatech.com/jlanka-launches-sri-lankas-first-ever-electricity-monitoring-epro1000-smart-system/

Unknown. (2015, 11 07). Smart Electric Meters to be introduced to Sri Lanka's electricity field. *Daily News*.

What is an API? (Application Programming Interface). (2019, 01 11). Retrieved from Mulesoft: https://www.mulesoft.com/resources/api/what-is-an-api

APPENDIX I – EQUIPMENT LIST

1.	Arduino Uno	1
2.	ESP8266 Wifi Module	1
3.	YHDC SCT-013-000 Current Transformer	1
4.	230 – 9 V AC-AC Power Adapter	1
5.	230 – 5 V AC-DC Power Adapter	1
6.	Mini Wifi Router	1
7.	Resistors	
	o 33 Ohm	1
	o 10 kOhm	5
	o 100 kOhm	1
8.	Capasitors	
	o 10 uF	2
9.	Jumper Wires	
10.	LEDs	

APPENDIX II – ARDUINO CODE

```
#include "EmonLib.h"
#include <SoftwareSerial.h>

#define RX 10
#define TX 11

// Energy Moniter
EnergyMonitor emon1;

// Energy Calculation Variables
unsigned long startTime;
unsigned long interval;
float watts = 0;
float wattsec = 0;
float totalTime = 0;

// Wifi Details
String AP = "Test";
String PASS = "12345678";

// Server Details
String HOST = "project.softinklab.com";
String PORT = "80";
String getData = "";

// API Call Variables
int countTrueCommand;
int countTimeCommand;
boolean found = false;

SoftwareSerial esp8266(RX,TX);

void setup()
{
  Serial.begin(9600);

  emon1.voltage(A1, 263, 2.5);
  emon1.current(A0, 30);
```

```
    pinMode(7, OUTPUT);

    // Wifi Module Start
    esp8266.begin(115200);

    // Wifi Module Initiate AT Commands
    sendCommand(1,"AT",5,"OK");
    delay(1000);
    sendCommand(2,"AT+CWMODE=1",5,"OK");
    delay(1000);
    sendCommand(3,"AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
    delay(1000);
    sendCommand(4,"AT+CIPMUX=1",5,"OK");
    delay(1000);

    // Indicators
    digitalWrite(7, HIGH);
    delay(2000);
    digitalWrite(7, LOW);
}

void loop()
{

    startTime = millis();
    emon1.calcVI(100,2000);

    float currentDraw       = emon1.Irms;
    float supplyVoltage     = emon1.Vrms;

    // Current Correction
    currentDraw = currentDraw - 0.1;
    if (currentDraw < 0){
        currentDraw = 0.0;
    }

    // Voltage Correction
    if (supplyVoltage < 200){
        supplyVoltage = 0.0;
    }

    // Calculation
```

```
    interval = millis() - startTime;
    totalTime = totalTime + interval;
    watts = currentDraw * supplyVoltage;
    wattsec = wattsec + (watts * (interval / 1000) );

    if (totalTime >= 10000){

        digitalWrite(7, HIGH);
        // Update Server
        getData = "GET /1/" + String(currentDraw) + "/" + String(supplyVoltage) + "/" +
String(wattsec) + " HTTP/1.1\r\nHost: " + HOST + "\r\ncache-control: no-cache\r\n";
        sendCommand(5,"AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
        sendCommand(6,"AT+CIPSEND=0," +String(getData.length()+4),4,">");
        esp8266.println(getData);delay(1500);countTrueCommand++;
        sendCommand(7,"AT+CIPCLOSE=0",5,"OK");
        digitalWrite(7,LOW);
    }
}

// Send Request (AT Command Manager)
void sendCommand(int id, String command, int maxTime, char readReplay[]) {
    while(countTimeCommand < (maxTime*1))
    {
        esp8266.println(command); //AT+cipsend
        if(esp8266.find(readReplay)) //OK
        {
            found = true;
            break;
        }

        countTimeCommand++;
    }

    if(found == true)
    {
        Serial.println(command + "OK");
        if (id==1 or id==2 or id==4){
            blinkLed(2);
        }else if(id==3){
            blinkLed(3);
        }else if(id==6){
            digitalWrite(7,LOW);
```

```
        delay(500);
        blinkLed(1);
        totalTime = 0;
        wattsec = 0;
      }
      countTrueCommand++;
      countTimeCommand = 0;
  }

  if(found == false)
  {
      Serial.println(command + "FAILED");
      if(id==6){
        totalTime = 0;
      }
      countTrueCommand = 0;
      countTimeCommand = 0;
  }

  found = false;
}

void blinkLed(int count){
      for (int i=1; i <= count; i++) {
        digitalWrite(7, HIGH);
        delay(100);
        digitalWrite(7,LOW);
        delay(200);
      }
}
```