VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

“J” – COMPONENT REPORT

COURSE CODE / NAME	ECM4002– IoT SYSTEM DESIGN
PROGRAM / YEAR / SEM	B.Tech (ECM)/III Year /Fall 2018-19
DATE OF SUBMISSION	29-10-2018
PROJECT TITLE	SMART ENERGY MONITOR

TEAM MEMBERS DETAILS	**REGISTER NO.**	**NAME**
	16BLC1021	RAJA HARIKESH N V
	16BLC1092	PRANAV N
	16BLC1135	PRAVEEN KUMAR R

EVALUATION ITEMS	**Yes (√) / No (x)**
The project has achieved the objective?	
Level of Knowledge Gained While Completing the Project was satisfactory?	
Are the students having clear idea on their proposedidea and have they acquired to carry forward it?	
Are the contribution made by the individuals towards attaining objective of the project was satisfactory?	
Are the submitted report and presentation made by each team member was satisfactory?	

COURSE INCHARGE NAME	**Dr. VYDEKI D**	**MARKS**	
REVIEWER’S NAME & SIGN			

Objective of the Project:

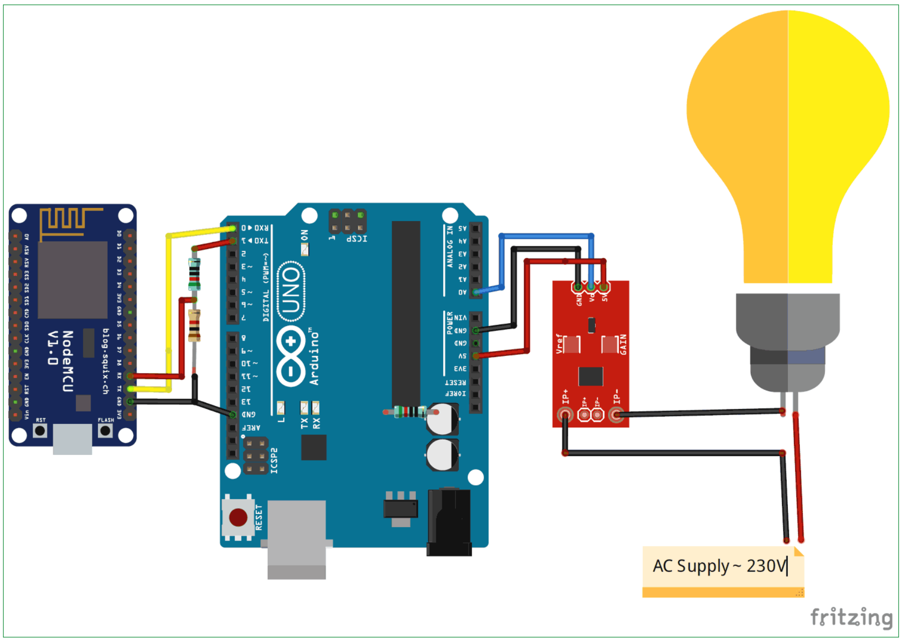
- At last of every month, many of us get worried about the high electricity bill and we have to look at the energy meter once in a while.
- What if we can monitor our electricity uses from anywhere in the world and get an SMS/E-mail when your energy consumption reaches to a threshold value.

Components Required:

S. No	Component Name	Specification	Quantity	Cost (in Rs.)
1	Arduino Uno		1	350
2	ACS712	30A rated	1	275
3	NodeMCU		1	350
4	Jumper Wires		10	25

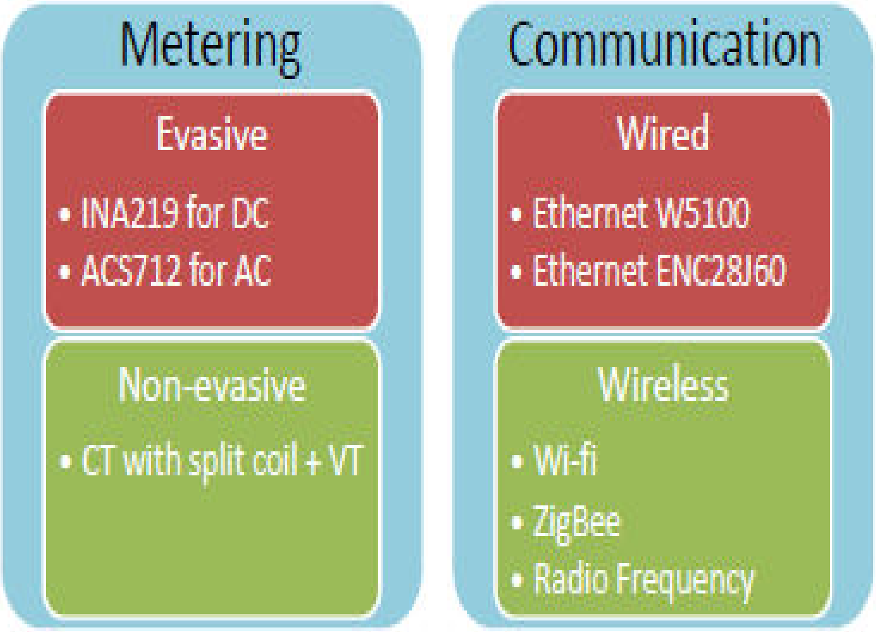
Overall cost of the Project: 1000 (in Rupees)

Block Diagram:



Project Description:

- Most energy monitors enable us to view our real-time electricity usage in units of energy used (kWh), cost or carbon emissions.
- Some have additional features, such as allowing us to set daily electricity usage targets or alarms to alert you when we have used a set amount of electricity.
- Most energy monitors are made up of three parts: a display part, a transmitter and a sensor.
- The sensor clips (evasive) or is connected in series with (evasive) a power cable connected to our electricity meter box. This monitors the magnetic field around the power cable to measure the electrical current passing through it (in amps).
- Once we've attached the transmitter to the sensor, it will send the information wirelessly to the cloud. Then the data will be re-calculated and displayed as real-time power usage (in kWh), cost.



Metering

Evasive

- INA219 for DC
- ACS712 for AC

Non-evasive

- CT with split coil + VT

Communication

Wired

- Ethernet W5100
- Ethernet ENC28J60

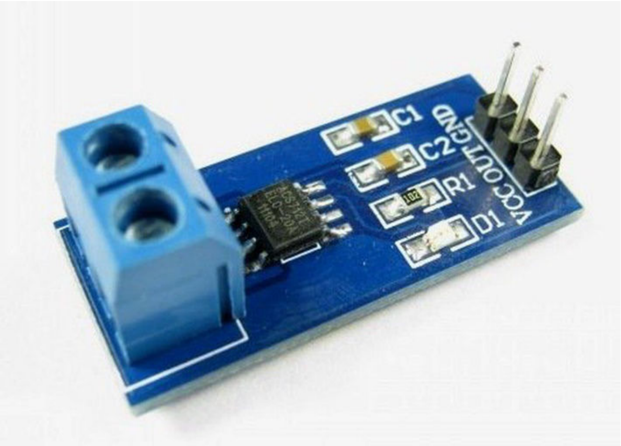
Wireless

- Wi-fi
- ZigBee
- Radio Frequency

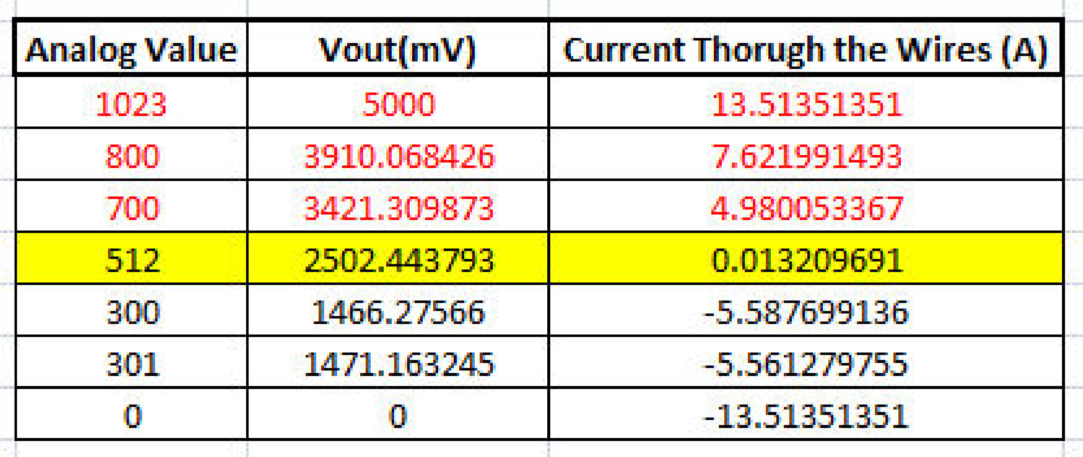
Working of ACS712 Current Sensor:

- Measuring current especially AC current is always a tough task due to the noise coupled with it improper isolation problem etc. But, with the help of this ACS712 module which was engineered by Allegro things have become a lot easier.

- This module works on the principle of **Hall-effect**, which was discovered by Dr. Edwin Hall.
- According his principle, when a current carrying conductor is placed into a magnetic field, a voltage is generated across its edges perpendicular to the directions of both the current and the magnetic field. This measurement will be in terms of millivolts which we called as the hall-voltage. The hall-voltage measured is proportional to the current that was flowing through the conductor.



- The major advantage of using **ACS712 Current Sensor** is that is can measure both AC and DC current and it also provides isolation between the Load (AC/DC load) and Measuring Unit (Microcontroller part).
- The below table will help us understand how the output voltage and ADC value varies based on the current flowing through the wire. These values were calculated based on the information given in the Datasheet of ACS712.



Analog Value	Vout(mV)	Current Thorugh the Wires (A)
1023	5000	13.51351351
800	3910.068426	7.621991493
700	3421.309873	4.980053367
512	2502.443793	0.013209691
300	1466.27566	-5.587699136
301	1471.163245	-5.561279755
0	0	-13.51351351

Program:

Code For Arduino :

```
#include "ACS712.h"
char watt[5];
ACS712 sensor(ACS712_30A, A0);
unsigned long last_time =0;
unsigned long current_time =0;
float Wh =0 ;
void setup() {
  Serial.begin(115200);
  sensor.calibrate();
}
void loop() {
  float V = 230;
  float I = sensor.getCurrentAC();
// Serial.println(I);
  float P = V * I;
  last_time = current_time;
  current_time = millis();
  Wh = Wh+  P *(( current_time -last_time) /3600000.0) ;
   dtostrf(Wh, 4, 2, watt);
Serial.write(watt);
  delay(10000);
}
```

Code For NodeMCU:

```
#define WLAN_SSID       "Hulkspot"
#define WLAN_PASS       "LukaModric"
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                // use 8883 for SSL
#define AIO_USERNAME    "HulkInMidgard"
#define AIO_KEY         "8f61a25a2ea14918a161d90adbe73029"WiFiClient client;
int bill_amount = 0;
unsigned int energyTariff = 8.0;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME,
AIO_KEY);
Adafruit_MQTT_Publish Power = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/Power");
Adafruit_MQTT_Publish bill = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/bill");
void MQTT_connect();
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println(F("Adafruit MQTT demo"));
  // Connect to WiFi access point.
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);
  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
}
void loop() {
  // Ensure the connection to the MQTT server is alive (this will make the first
  // connection and automatically reconnect when disconnected).  See the MQTT_connect
  // function definition further below.
  MQTT_connect();
 int i=0;
 float watt1;
 if(Serial.available() > 0 ){
   delay(100); //allows all serial sent to be received together
    while(Serial.available() && i<5) {
     watt[i++] = Serial.read();
    }
    watt[i++]='\0';
  }
 watt1 = atof(watt);
  bill_amount = watt1 * (energyTariff/1000);      // 1unit = 1kwH
  Serial.print(F("\nSending Power val "));
  Serial.println(watt1);
  Serial.print("...");
  if (! Power.publish(watt1)) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("OK!"));
  }
   if (! bill.publish(bill_amount)) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("OK!"));
  }
if (bill_amount==4){
for (int i =0; i<=2; i++)
{
  bill.publish(bill_amount);
delay(5000);
}
bill_amount =6;
}

delay(5000);
}
// Function to connect and reconnect as necessary to the MQTT server.
// Should be called in the loop function and it will take care if connecting.
void MQTT_connect() {
  int8_t ret;
  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }
  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
       Serial.println(mqtt.connectErrorString(ret));
       Serial.println("Retrying MQTT connection in 5 seconds...");
       mqtt.disconnect();
       delay(5000);  // wait 5 seconds
       retries--;
       if (retries == 0) {
         // basically die and wait for WDT to reset me
         while (1);
       }
  }
  Serial.println("MQTT Connected!");
}
```

Setting up an AdaFruit account for communication:

Step 1:- Login to Adafruit IO with the credentials

Step 2:-Click on *My account -> Dashboard*

Step 3:-Click on *Actions* and *Create a New Dashboard*.

Step 4: Give name and description to the feed and click on *Create*.

Step 5: Click on *Key* button and note down the AIO Keys, to use this key in our code.

Step 6: Click on '+' button to create a new block and click on *Gauge* to display Energy uses level.

Step 7: Now, Enter Name of Feed and click on *Create*. Then Select the feed and click on Next step.

Step 8: In block settings, fill the min. and max values as 0 and 100 respectively or we can modify as we want.

Step 9: Our Power feed is successfully created. Now, create feed to display Bill by clicking on "+" sign.

Create Applet in IFTTT for Triggering SMS/Email for Energy Meter:

Step 1: Login to IFTTT with the credentials.

Step 2: On My Applets, Click on **New Applet**

Step 3: Click on +**this**

Step 4: Search *AdaFruit* and click on it.

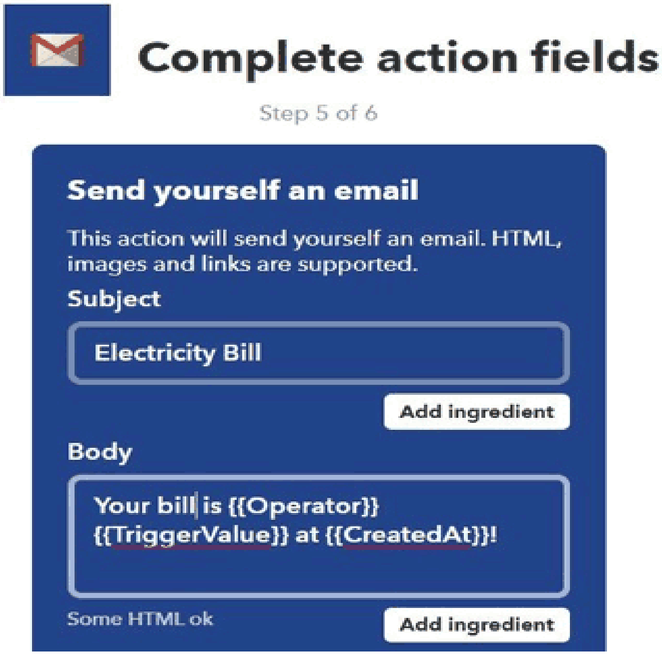
Step 5: Click on *Monitor a feed* on AdaFruit IO.

Step 6: Choose *Feed* as bill, *Relationship* as '*equal to*' and the threshold *value* at which we want an E-mail. Click on *Create action*.

Step 7: Click on +*that*. Search for G-mail and click on it and Login with g-mail credentials.

Step 8: Click on **send yourself an email**.

Step 9: Write the subject and body as shown and click to create.



Complete action fields

Step 5 of 6

Send yourself an email

This action will send yourself an email. HTML, images and links are supported.

Subject

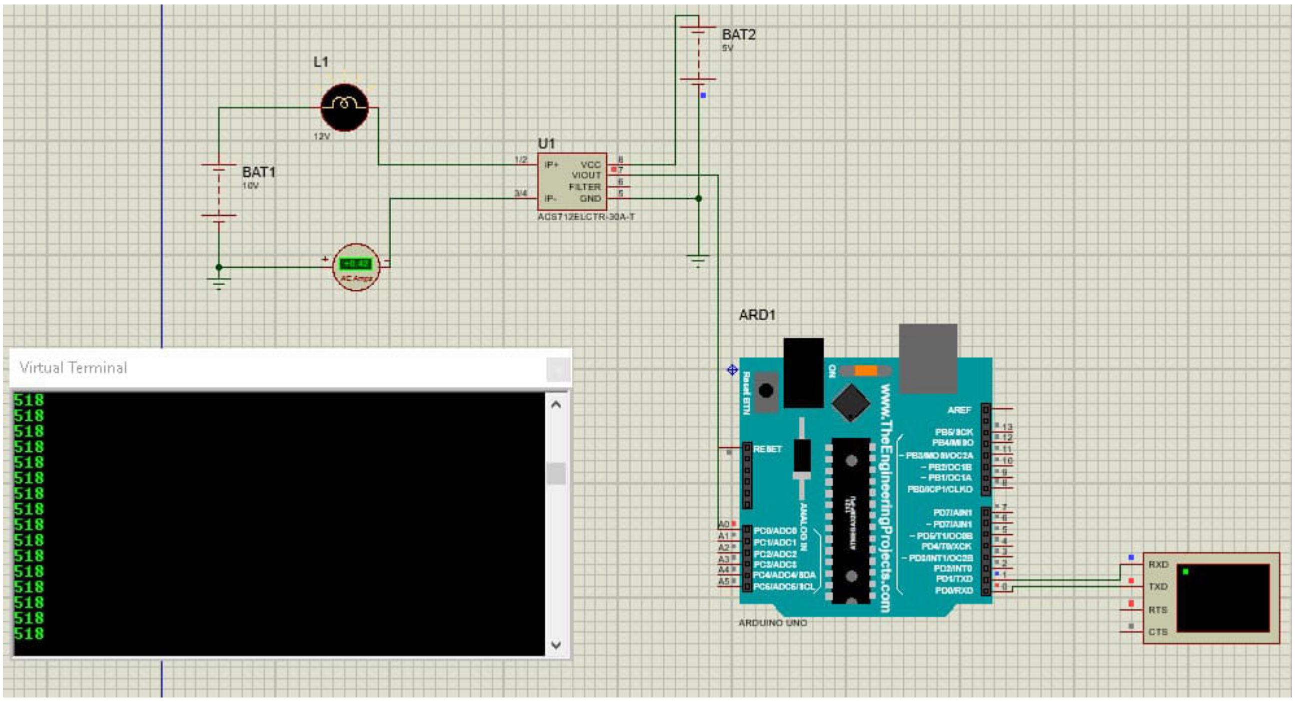
Electricity Bill

Body

Your bill is {{Operator}} {{TriggerValue}} at {{CreatedAt}}!

Step 10: Review it and click on finish.

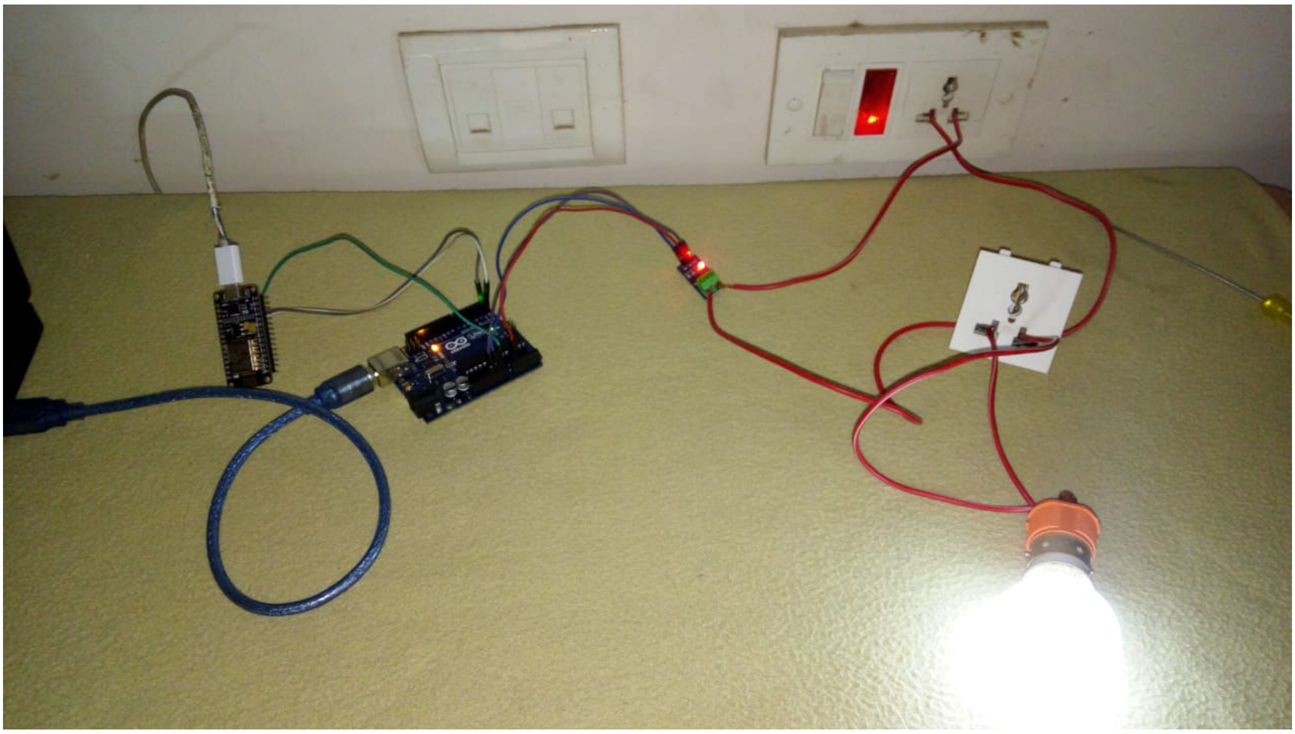
Simulation Output:

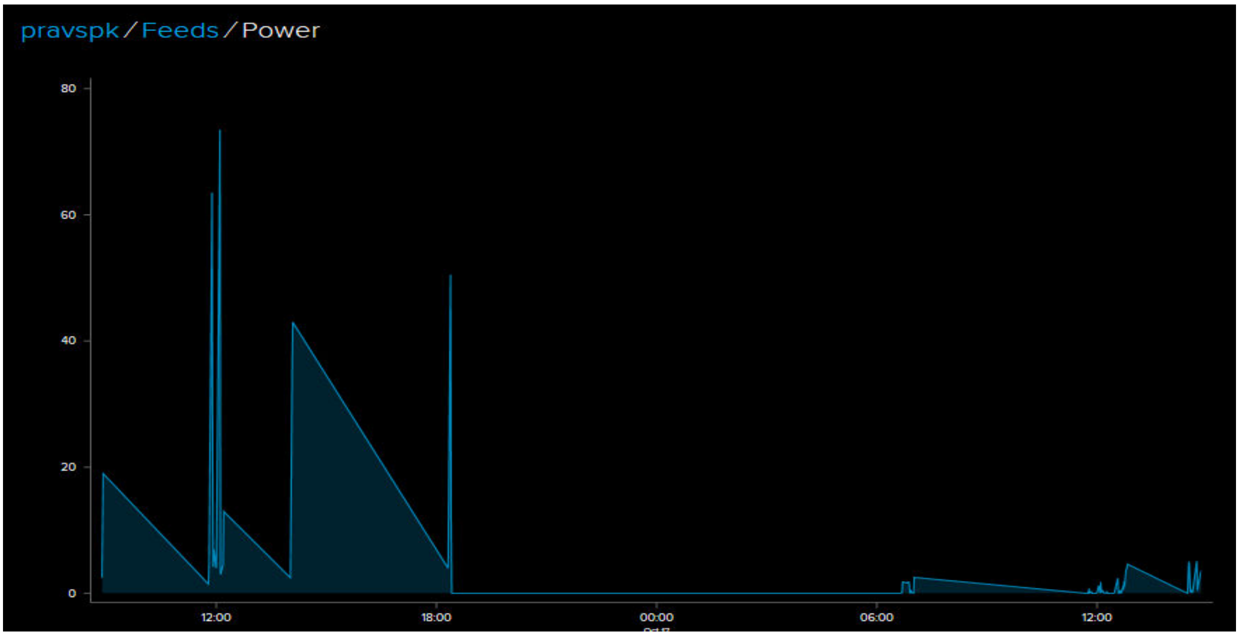


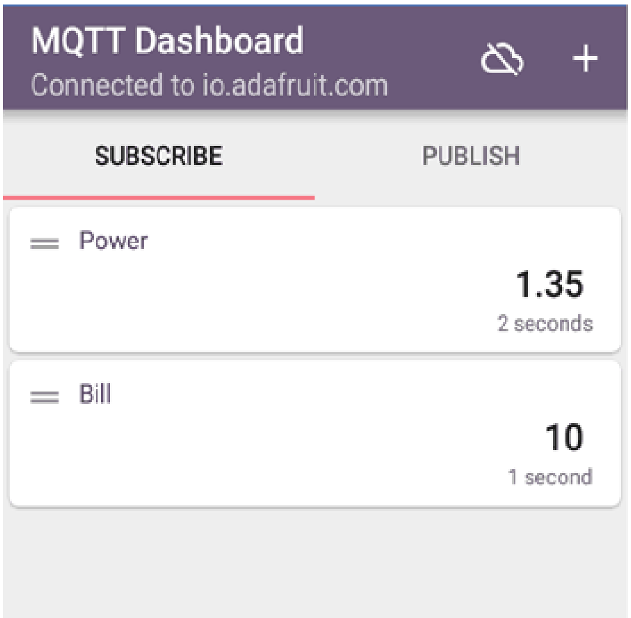
```
518
518
518
518
518
518
518
518
518
518
518
518
518
518
518
518
```

Virtual Terminal displaying the analog value read by current sensor.

Implementation Output (Photo):







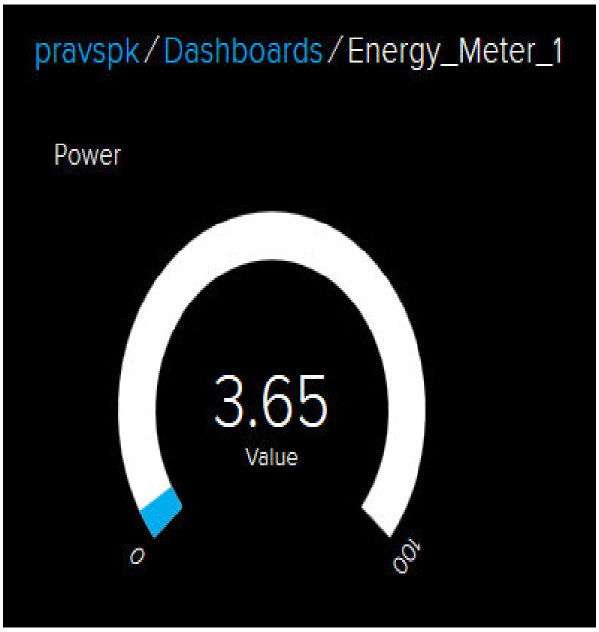
MQTT Dashboard

Connected to io.adafruit.com

SUBSCRIBE PUBLISH

Power 1.35 2 seconds

Bill 10 1 second



Inference:

- A means to reduce household energy consumption is to provide real-time feedback to homeowners so they can change their energy use. So we made a successful attempt to make a real-time home energy monitor.
- Real-time data on how much energy is being consumed in the home was sent directly to the Adafruit feeds and alerts appeared when the energy crossed the limit specified in the setup.

Concepts Learned:

- DC Current measurement
- Sensor calibration
- AC Current measurement
- Serial communication
- WiFi access (NodeMCU)
- Data Analytics (Adafruit IO)

Difficulties faced:

- One of the major difficulty faced is proper wiring since the method being evasive.
- It was also not possible to measure directly the current flowing through the circuit using measurement devices since ammeter works only in series connection. So it was difficult to validate current reading unless the RPS was used to manually set and check the current value.