0

A PROJECT REPORT

IoT Based Energy Meter

GUEE

Under the guidance of

Prof. Saravanan B

BY

Apurv Aggarwal 20BEE0272

Loknath Sai 20BEE0392

IoT based energy Meter

Loknath Sai
20BEE0392

Apurv Aggarwal
20BEE0272

I. ABSTRACT

This project presents the design and implementation of an IoT-based smart energy meter using the ESP8266 microcontroller and ACS712 current sensor, with Blynk IoT as the cloud-based server. The system can measure the real-time energy consumption of a residential or commercial building and transmit the data to the Blynk IoT Cloud Platform for remote monitoring and analysis. The system is also capable of generating alerts when energy usage exceeds a predetermined threshold, enabling users to take timely corrective actions. The Blynk IoT platform provides an easy-to-use interface for data visualization and analysis, allowing users to track their energy consumption patterns, set energy-saving goals, and monitor their progress over time. The proposed system provides an efficient and cost-effective solution for energy management, making it a promising option for both residential and commercial applications.

II. INTRODUCTION

As consumer electronics trends increase, the world of power generation, which satisfies consumer requirements, is expanding significantly. In addition to the tendency, social culture is also driving up the rate of consumption. Government regulations, on the other hand, are causing a change in favour of requiring renewable energy sources to counter the rise in energy usage. The two-way shift will strengthen the technology of sustainable development in the energy sector and conservatories on electricity consumption to achieve network efficiency. It will also enable energy auditing and practical consumer involvement. We all agree with the conclusions. As a result, the need for metering infrastructure is unavoidable, as it is essential for measuring usage and predicting demand characteristics. Common domestic energy metres and industrial metres are the most fundamental metering elements in the AMI. Utilizing energy metres the Government Electricity Board has installed, the Government Electricity Board measures domestic and industrial energy usage. Despite the digitization of energy meters, some features are still lacking. In particular, real-time pricing transparency, network latency in communications, and device-level performance portfolio are among these. However, real-time smart billing is available with the current smart meters, and the price of each individual load bill is not shown on the metre.

III. LITERATURE SURVEY

When we are referring to the journal papers we concluded with the following results, Not only measuring energy there are many features available in different modules that we referred to. Few are in the document; robbery of the electricity can be identified through the meter.

Home automation can be done using energy meters. Smart grid technologies can be developed faults can be detected much energy-based real-time analysis can be done using these intelligent services can be made. Software like Blynk can be used for better interaction and automation. The need for a higher security system via IoT protocols for better service is discussed tempering can also be detectable.

IV. METHODOLGY

We can obtain Instantaneous voltage and current directly from sensors by calibrating their output and instantaneous power can be obtained as product of those two quantities. To caliberate the output and to calculate the derived parameters such as RMS power , RMS voltage , RMS current ,power factor we use Arduino IDE. We will calculate total energy consumed by using Arduino code and write it on blink library using blink virtual Write function

V. ALGORITHM

ACS712 current to Arduino.
Formula to read current from acs712 to Arduino
To measure current using the ACS712 current sensor and an Arduino, you can use the following formula:
CURRENT=(3.3/2-OPV*3.3/1024+3)/0.185
This formula assumes that you are using the ACS712-05A sensor, which has a sensitivity of 185mV/A. If you are using a different model of the ACS712, you will need to adjust the sensitivity value accordingly.
In the formula, "analogRead(A0)" reads the voltage output from the ACS712 sensor, which is proportional to the current being measured. The value 512 is subtracted to account for the fact that the ACS712 outputs a voltage centered around 2.5V, so subtract From that current, tell me all the expressions as algorithm in the formula to find energy for energy meter with fixed voltage of 230v and power factor of 0.8

To calculate the energy consumption using the current reading from the ACS712 sensor and assuming a fixed voltage of 230V and a power factor of 0.8, you can use the following algorithm:

Calculate the instantaneous power:
instantaneous_power = current * voltage * power_factor

where current is the current reading from the ACS712 sensor, voltage is the fixed voltage of 230V, and power_factor is the assumed power factor of 0.8.

Calculate the time elapsed since the last power calculation. You can use the Arduino's millis() function to get the time in milliseconds:

makefile
Copy code
current_time = millis();
elapsed_time = current_time - last_time;
Convert the elapsed time to hours:
elapsed_hours = elapsed_time / 3600000.0

where 3600000.0 is the number of milliseconds in an hour.

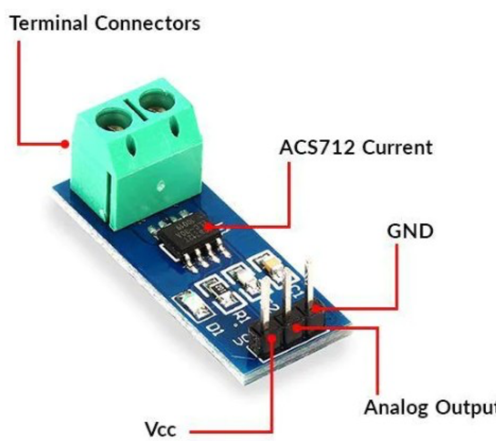
Calculate the energy consumed during the elapsed time:
energy = instantaneous_power * elapsed_hours

Add the energy consumed during the elapsed time to a running total:
total_energy += energy

Store the current time as the last time a power calculation was made:
last_time = current_time
Repeat steps 1-6 for each power calculation, which should be made at a regular interval (e.g. every second).
At the end of the measurement period, the total energy consumed (in watt-hours) will be stored in the total_energy variable.

A. ACS712 Current Sensor:

The main advantage of using the ACS712 current sensor is the ability to measure both AC and DC currents and the isolation between the load (AC/DC load) and the measuring unit (microcontroller part). Each module has three pins as shown in the picture: Vcc, Vout and Ground. The 2-pin terminal block is where live wires pass. Since the module operates at +5V, Vcc should be supplied with 5V and ground should be connected to system ground. The Vout pin has an offset voltage of 2500mV. That is, when there is no current in the line, the output voltage is 2500mV, and when the current is positive, the voltage is greater than 2500mV and the current is negative. , the voltage is less than 2500mV. Read the module’s output voltage (Vout) using the analog pins of the Arduino. This is 512 (2500 mV) when there is no current in the wire. This value decreases when current flows in the negative direction and increases when current flows in the positive

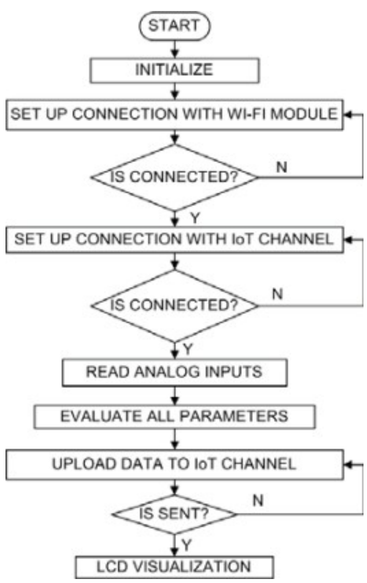


ACS712 Current Sensor.

direction. The table below will help you understand how the output voltage and ADC value change based on the current flowing through the wire.

B. Arduino IDE

Using the Arduino integrated development environment or simply the Arduino software (IDE), you created a program and uploaded it to the Arduino module. A program written in it is called a sketch and is saved as a .ino file extension . You can later upload these programs to your Arduino or Genuino device. The IDE also features a text console that allows you to easily monitor the ’s output. You can also edit the and upload it if you don’t get the results you want. Software and hardware integration is the backbone of such "intelligent" devices. Flowchartshown in FIG. 5 illustrates the complete process by which device connects to the Internet, collects data, evaluates parameters, and displays results. It also shows how and when to establish a connection to an IoT channel.



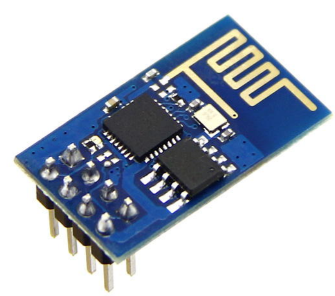
Flow Chart of IDE.

C. Blynk IoT Platform

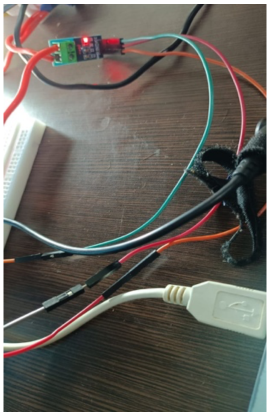
The IoT platform that we have used in this project is Blynk Iot. Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet. This application is used to create a graphical interface or human machine interface (HMI) by compiling and providing the appropriate address on the available widgets.

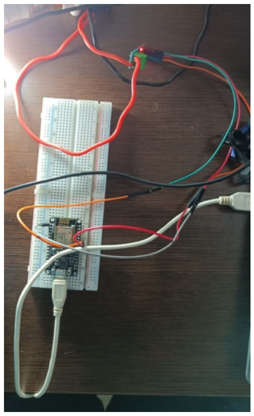
D. ESP 8266 Wifi Module

ESP8266 is Wi-Fi enabled system on chip (SoC) module developed by Espressif system. It is mostly used for development of IoT (Internet of Things) embedded applications. It employs a 32-bit RISC CPU based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed through SPI. ESP8266 module is low cost standalone wireless transceiver that can be used for end-point IoT developments. To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate.

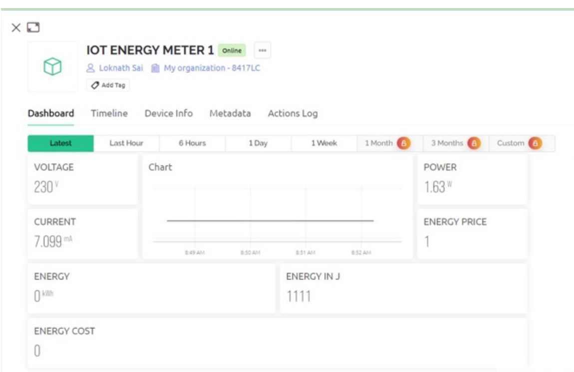


VI. PROJECT PICS





VII. RESULT




```
float getcurrent()
{
  unsigned int x=0;

float AcsValue=0.0,Samples=0.0,AvgAcs=0.0,AcsValueF=0.0;

  for (int x = 0; x < 150; x++){ //Get 150 samples
  AcsValue = analogRead(A0);     //Read current sensor values
  Samples = Samples + AcsValue;  //Add samples together
  delay (3); // let ADC settle before next sample 3ms
}
AvgAcs=Samples/150.0;//Taking Average of Samples
AcsValueF = (3.3/2 - (AvgAcs * (3.3/2/ 1024.0)) +3*0.185)/0.185;
return(AcsValueF/1000);
}
float current=getcurrent();
void setup()
{
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}
float power=0,amount=0;
void loop()
{
  Blynk.run();
  //timer.run();
//  current=sensor.getCurrentAC();
  power=current*voltage;
  //energy+=power/1000/3600;
  amount=energy*cost;
  Blynk.virtualWrite(V5, cost);
  Blynk.virtualWrite(V0, voltage);
  //float v1=-current;
  Blynk.virtualWrite(V1, current*1000);

  Blynk.virtualWrite(V7, current*1000);
  Blynk.virtualWrite(V2, power);
  energy=voltage*current+energy;
  units=energy/3600/1000;
  Blynk.virtualWrite(V4, energy);
  Blynk.virtualWrite(V3, units);
  Blynk.virtualWrite(V6, units*cost);
```

CURRENT=(3.3/2-OPV*3.3/1024+3)/0.185

VIII. COST

NodeMCU	350 Rs
ACS712	300 Rs
JUMPER WIRES	10 Rs
ZMPT101	400 Rs
Blynk subscription	600 Rs
TOTAL COST	1660 Rs

IX. LITERATURE REVIEW

Development of PDAP (Probabilistic Distributed Arithmetic progression) algorithm Energy utility of each component can also be calculated. They are proposing power line mode of communications for security purpose. New form of communication protocol is used i.e., LoRa designed for checking utility. The monthly bill with unit consumption and user Id will be sent to the service provider. They concentrate more on component selection. Once the robbery connection is identified with the help of that the system, then this will be informed to the authorized consumers through SMS. The threshold value can be set on a webpage with the help of Wi-Fi, as per the consumer's requirement. Here thereby are creating a web interface with a user id the user will get to know the
right power consumption and recharge it very well.

.

Conclusion

This article introduced an IoT-based smart energy man- agement system. An Electronic Meter Automation Device (EMAD) has been developed that can be installed on older meters. Installed in older meters, this device allows the to act as a smart meter. website and intelligent app developed. Gauges can be easily accessed through 's customized web pages and smart apps, and notifications can be sent via SMS. The counter can be turned on or off automatically by the user when the prepayment threshold is exhausted or via a smart app .

References

[1] https://sci-hub.se/https://ieeexplore.ieee.org/document/8748696
[2] https://sci-hub.se/https://ieeexplore.ieee.org/document/8521650
[3] https://sci-hub.se/https://ieeexplore.ieee.org/document/8658501
[4] https://how2electronics.com/iot-based-electricity-energy-meter-using-esp32-blynk/
[5] https://iotdesignpro.com/projects/iot-based-smart-energy-meter