

Peacefair PZEM-004T AC Multi-function Electric Energy Metering Power Monitor



In many electrical projects, engineer directly deals with measurements with few basic requirements like.

- High galvanic isolation
- Parameter display
- Direct communication with computer
- Data acquisition and storage with subsequent viewing or copying to the computer.

PZEM-004T V3.0 or Version 3.0 is the upgraded version to replace the old **PZEM004T** V1.0. The old version has been sold out in most of

the online store and no produce anymore. The updated version of PZEM004T is best for the DIY project, where we need to measure the voltage, current, power, energy, frequency, Power factor (frequency and PF is extra added in the new version) using Arduino/ESP8266/Raspberry Pi like open-source platform.



This module comes with 3 different current measurement options

- 10A range with a *built-in Shunt resistor*
- 100A external *closed Current Transformer*
- 100A external split current Transformer

I recommend the shell protected split current transformer module for a project like **Portable AC energy meter**, Energy meter for electrical Lab, Energy auditing and measuring equipment etc. as split

current Transformer is more flexible than closed one. Closed CT is a low-cost option suitable for fixed type projects like Residential Energy management system, Load control system, **IoT based smart Energy meter, etc.**

1. 10A range with a built-in Shunt resistor



2. 100A external closed Current Transformer



100A Module+Closed CT

3. 100A external split current Transformer



The physical dimensions of the **PZEM-004T v3** board is 3.01×7.3 cm, The pzem-004t V3 module is bundled with 33mm diameter 100A current transformer coil.

The new version has higher precision, faster refresh speed, and more stability communication than the old version.

Voltage

- Measuring range:80~260V
- Resolution: 0.1V
- Measurement accuracy: 0.5%

Power factor

- Measuring range: 0.00~1.00
- Resolution: 0.01
- Measurement accuracy: 1%

Frequency

- Measuring range: 45Hz~65Hz
- Resolution: 0.1Hz
- Measurement accuracy: 0.5%

Current

- Measuring range: 0~10A(PZEM-004T-10A); 0~100A(PZEM-004T-100A)
- Starting measure current: 0.01A(PZEM-004T-10A); 0.02A(PZEM-004T-100A)
- Resolution: 0.001A
- Measurement accuracy: 0.5%

Active power

- Measuring range: 0~2.3kW(PZEM-004T-10A); 0~23kW(PZEM-004T-100A)
- Starting measure power: 0.4W
- Resolution: 0.1W
- Display format:

<1000W, it displays one decimal, such as: 999.9W

≥1000W, it display only integer, such as: 1000W

- Measurement accuracy: 0.5%

Active Energy

- Measuring range: 0~9999.99kWh
- Resolution: 1Wh
- Measurement accuracy: 0.5%
- Display format:

<10kWh, the display unit is Wh(1kWh=1000Wh), such as: 9999Wh

≥10kWh, the display unit is kWh, such as: 9999.99kWh

Function Description in short

Function	Measuring range		Starting measure current/power		Resolution	Measure-ment accuracy	Display format
	10A	100A	10A	100A			
Voltage	80~260V				0.1V	0.5%	
Current	0~10A	0~100A	0.01A	0.02A	0.001A	0.5%	
Active power	0~2.3kW	0~23kW	0.4W		0.1W	0.5%	<1000W, it display one decimal, such as: 999.9W; ≥1000W, it display only integer, such as: 1000W
Power factor	0.00~1.00				0.01	1%	
Frequency	45Hz~65Hz				0.1Hz	0.5%	
Active energy (Reset energy: use software to reset)	0~9999.99kWh				1Wh	0.5%	<110kWh, the display unit is Wh(1kWh=1000Wh), such as: 9999Wh; ≥10kWh, the display unit is kWh, such as: 9999.99kWh
Over power alarm	Active power threshold can be set, when the measured active power exceeds the threshold, it can alarm						
Communication interface	RS485 interface						
size	Length * width * height=73.7*30*14.3mm (Bare pager)						
Power Supply	The power supply of single-phase power-frequency network supplies power to the main circuit through resistance-capacitance step-down, TTL output communication interface and Main circuit optocoupler isolation, for passive output, communication needs to provide external 5V power supply						
working temperature	-20°C~+60°C						

Download links

DataSheet PZEM-004T

Download the Datasheet of PZEM-004T Modules. In datasheet detailed specifications and interfacing details are available.

[Download Now](#)



PZEM-004T Arduino Library

PZEM-004T Library for Arduino IDE

[Get from github.com](https://github.com)



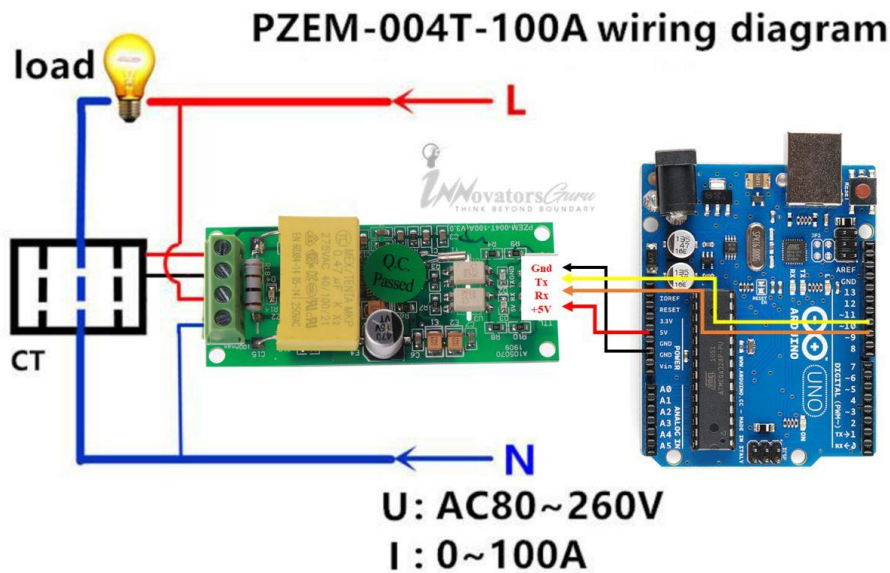
PZEM-004T Windows Software

Windows Software for PZEM004T Module Testing

[Download Now](#)

Arduino PZEM004T V3 Interfacing

PZEM-004T Arduino Wiring Diagram



PZEM-004T V3 module has passive serial communication to connect Arduino Uno. Its RX and TX pin needs an external power supply. Connect the 5V pin to Arduino 5V pin, Rx pin to D11 of Arduino and Tx pin to D12 pin of Arduino, GND to GND. In the below diagram, PZEM004T V3 is connected to Arduino using Software serial. Use the following Program to test the module. Before compile download and install the [PZEM-004T V3 library](#).

PZEM004T Arduino Code

This pzem004t [Arduino](#) code enable you to measure AC voltage, AC current, Power, Energy, Frequency and Power Factor on serial interface with arduino microcontroller. The [library of PZEM-004t V30](#) works on hardware serial as well as software serial. The result of all parameters are printed on serial monitor of [Arduino IDE](#). I tested this library code with arduino uno, arduino mega, and nano board.

Copy to Clipboard



```
1
2
3
4
5
6
7
8
9
10
11
12
13
14

#include <PZEM004Tv30.h>

PZEM004Tv30 pzem(11, 12);

void setup() {

Serial.begin(115200);

}

void loop() {

float voltage = pzem.voltage();

if(voltage != NAN){

Serial.print("Voltage: "); Serial.print(voltage); Serial.println("V");

} else {

Serial.println("Error reading voltage");

}

float current = pzem.current();
```

```
if(current != NAN){
    Serial.print("Current: "); Serial.print(current); Serial.println("A");
} else {
    Serial.println("Error reading current");
}

float power = pzem.power();

if(current != NAN){
    Serial.print("Power: "); Serial.print(power); Serial.println("W");
} else {
    Serial.println("Error reading power");
}

float energy = pzem.energy();

if(current != NAN){
    Serial.print("Energy: "); Serial.print(energy,3); Serial.println("kWh");
} else {
    Serial.println("Error reading energy");
}

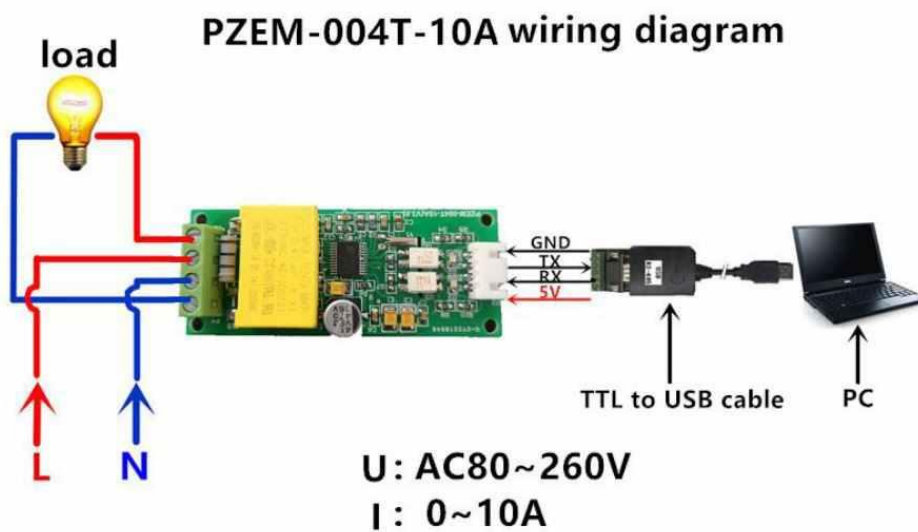
float frequency = pzem.frequency();

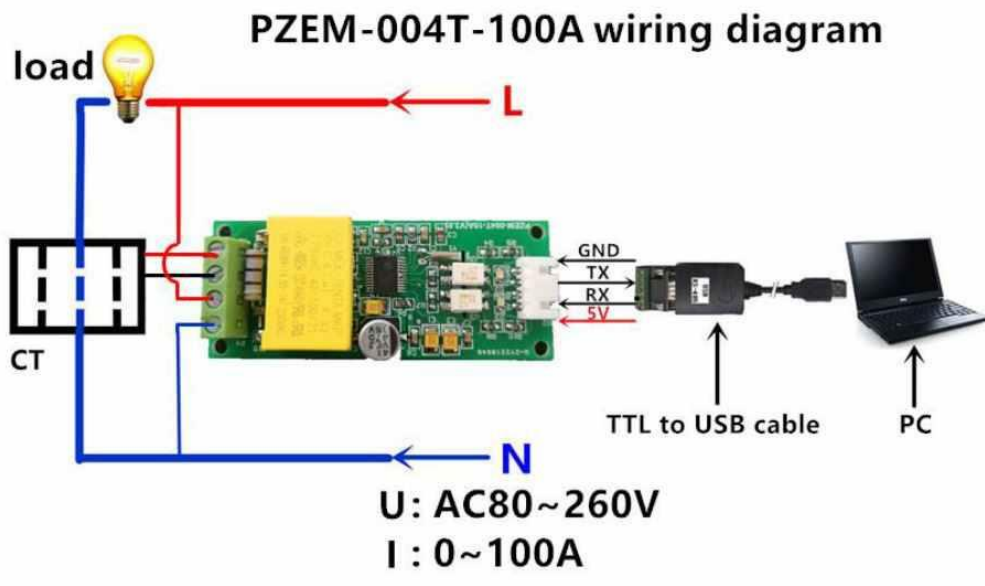
if(current != NAN){
    Serial.print("Frequency: "); Serial.print(frequency, 1); Serial.println("Hz");
} else {
    Serial.println("Error reading frequency");
```

```

}
36
float pf = pzem.pf();
37
if(current != NAN){
38
Serial.print("PF: "); Serial.println(pf);
39
} else {
40
Serial.println("Error reading power factor");
41
}
42
Serial.println();
43
delay(2000);
44
}
45

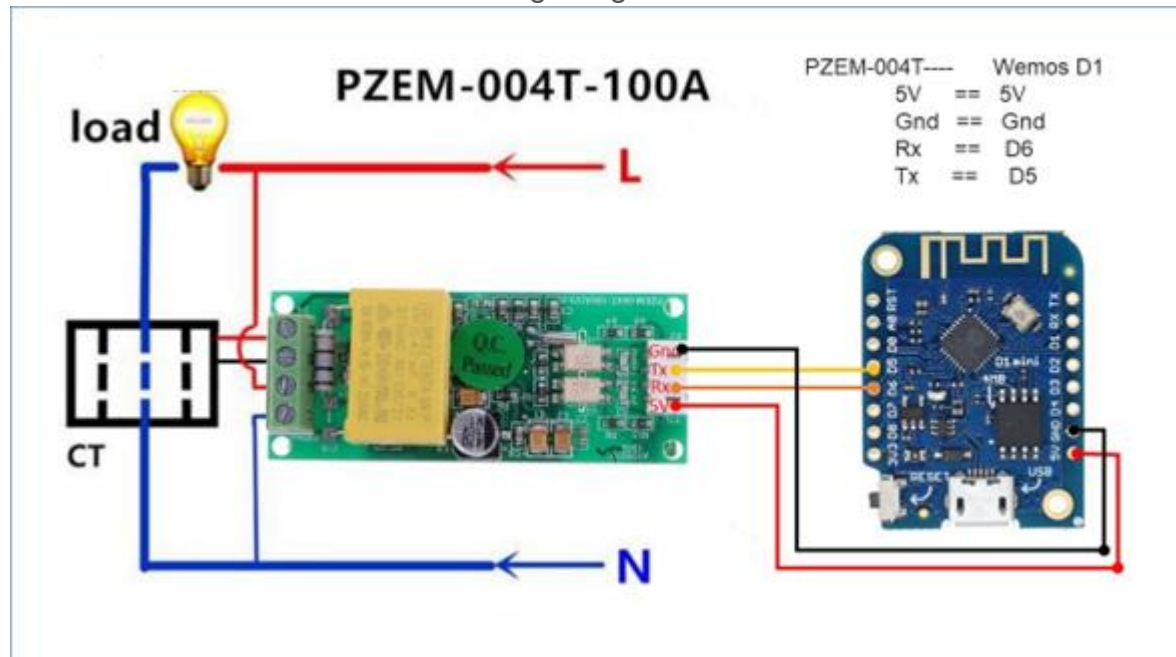
```



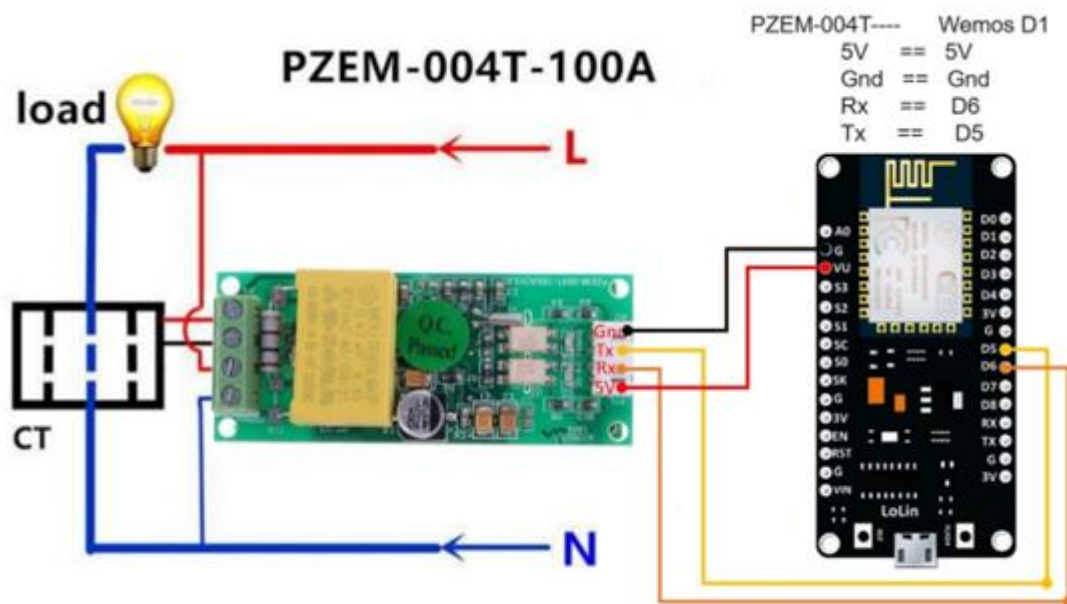


NodeMCU/Wemos D1 PZEM004T V3 Connection Diagram

PZEM-004T Wemos d1 mini Wiring Diagram



PZEM-004T V3 NodeMCU Wiring Diagram



PZEM-004T V3 module serial communication to connect NodeMCU or Wemos. (Connection is same for ESP8266 NodeMCU) Its RX and TX communication pin needs external 5V power. Connect the 5V pin of Wemos D1 mini to PZEM004T 5V pin, Rx pin to D6 of arduino and Tx pin to D5 pin of Wemos, gnd to gnd. In the below diagram, we connect PZEM004T V3 to Wemos D1 Mini using Software serial. Use the program to test the module. Before compile, download and install the [library](#).

Note: PZEM-004T V3 and ESP8266 NodeMCU are communication works on different voltage level theoretically but in practical life 3.3V nodemcu pins have tolerance of 5V. VU pin is USB 5V pin (only works when NodeMCU powered through USB)

PZEM-004T ESP8266 (NodeMCU) Code

This pzem-004t NodeMCU (ESP8266 based development board) code allows you to measure voltage, current, Kw, kwh, Frequency and PF display data on the [Blynk app](#). The purpose of this code is to learn to connect PZEM-004T v3.0 to [NodeMCU](#) or [Wemos Mini D1](#).

For complete code and explanations

- Visit [community.blynk](#)
- [GitHub Resources](#)

Copy to Clipboard



```
1
2 #include <ArduinoOTA.h>
3
4 #include <BlynkSimpleEsp8266.h>
5
6 #include <SimpleTimer.h>
7
8 #include <ModbusMaster.h>
9
10 #include <ESP8266WiFi.h>
11
12 #include "settingsPZEM.h"
13
14 #include <SoftwareSerial.h> // ( NODEMCU ESP8266 )
15
16 SoftwareSerial pzem(D5,D6); // (RX,TX) connect to TX,RX of PZEM for NodeMCU
17
18 //SoftwareSerial pzem(D7,D8); // (RX,TX) connect to TX,RX of PZEM
19
20 #include <ModbusMaster.h>
21
22 ModbusMaster node;
23
24 SimpleTimer timer;
25
26 //WiFi data
27
28 char ssid[] = "Put your WiFi SSID here"; //WiFi Credential
29
30 char pass[] = "Put your WiFi password here"; //WiFi Password
31
32 char server[] = "Put your Blynk local server IP address here"; //Blynk local server IP address
33
34 int port = 8080; //Blynk local port
35
36 #define USE_LOCAL_SERVER //Use local Blynk Server - comment-out if use Blynk hosted cloud
37 service
38
39
```

```

#define AUTH "put your Blynk App Authorization code here" //PZEM-004v3 Auth code for
Blynk Local Server
20

int timerTask1;
21

double U_PR, I_PR, P_PR, PPR, PR_F, PR_PF, PR_alarm;
22

uint8_t result; uint16_t data[6];
23

void setup(){
24

Serial.begin(115200); Serial.println("Start serial"); pzem.begin(9600); Serial.println("Start
PZEM serial");
25

node.begin(1, pzem); Serial.println("Start PZEM"); // 1 = ID MODBUS
26

WiFi.mode(WIFI_STA);
27

#ifdef USE_LOCAL_SERVER
28

WiFi.begin(ssid, pass);
29

Blynk.config(AUTH, server, port);
30

#else
31

Blynk.begin(AUTH, ssid, pass);
32

#endif
33

while (Blynk.connect() == false) {}
34

ArduinoOTA.setHostname(OTA_HOSTNAME);
35

ArduinoOTA.begin();
36

// timerTask1 = timer.setInterval(1000, updateBlynk);
37

}
38

void updateBlynk() {
39

Blynk.virtualWrite(vPIN_VOLTAGE, U_PR);

```

```
40
Blynk.virtualWrite(vPIN_CURRENT_USAGE, I_PR);
41
Blynk.virtualWrite(vPIN_ACTIVE_POWER, P_PR);
42
Blynk.virtualWrite(vPIN_ACTIVE_ENERGY, PPR);
43
Blynk.virtualWrite(vPIN_FREQUENCY, PR_F);
44
Blynk.virtualWrite(vPIN_POWER_FACTOR, PR_PF);
45
Blynk.virtualWrite(vPIN_OVER_POWER_ALARM, PR_alarm);
46
}
47
void loop(){
48
Blynk.run();
49
//ArduinoOTA.handle();
50
//timer.run();
51
result = node.readInputRegisters(0x0000, 10);
52
if (result == node.ku8MBSuccess) {
53
U_PR = (node.getResponseBuffer(0x00)/10.0f);
54
I_PR = (node.getResponseBuffer(0x01)/1000.000f);
55
P_PR = (node.getResponseBuffer(0x03)/10.0f);
56
PPR = (node.getResponseBuffer(0x05)/1000.0f);
57
PR_F = (node.getResponseBuffer(0x07)/10.0f);
58
PR_PF = (node.getResponseBuffer(0x08)/100.0f);
59
PR_alarm = (node.getResponseBuffer(0x09));
60
}
61
```

```

Serial.print("U_PR: "); Serial.println(U_PR); // V           62
Serial.print("I_PR: "); Serial.println(I_PR,3); // A       63
Serial.print("P_PR: "); Serial.println(P_PR); // W         64
Serial.print("PPR: "); Serial.println(PPR,3); // kWh       65
Serial.print("PR_F: "); Serial.println(PR_F); // Hz        66
Serial.print("PR_PF: "); Serial.println(PR_PF);           67
Serial.print("PR_alarm: "); Serial.println(PR_alarm,0);   68
Serial.println("=====");                               69
updateBlynk();                                           70
delay(1000);                                             71
}

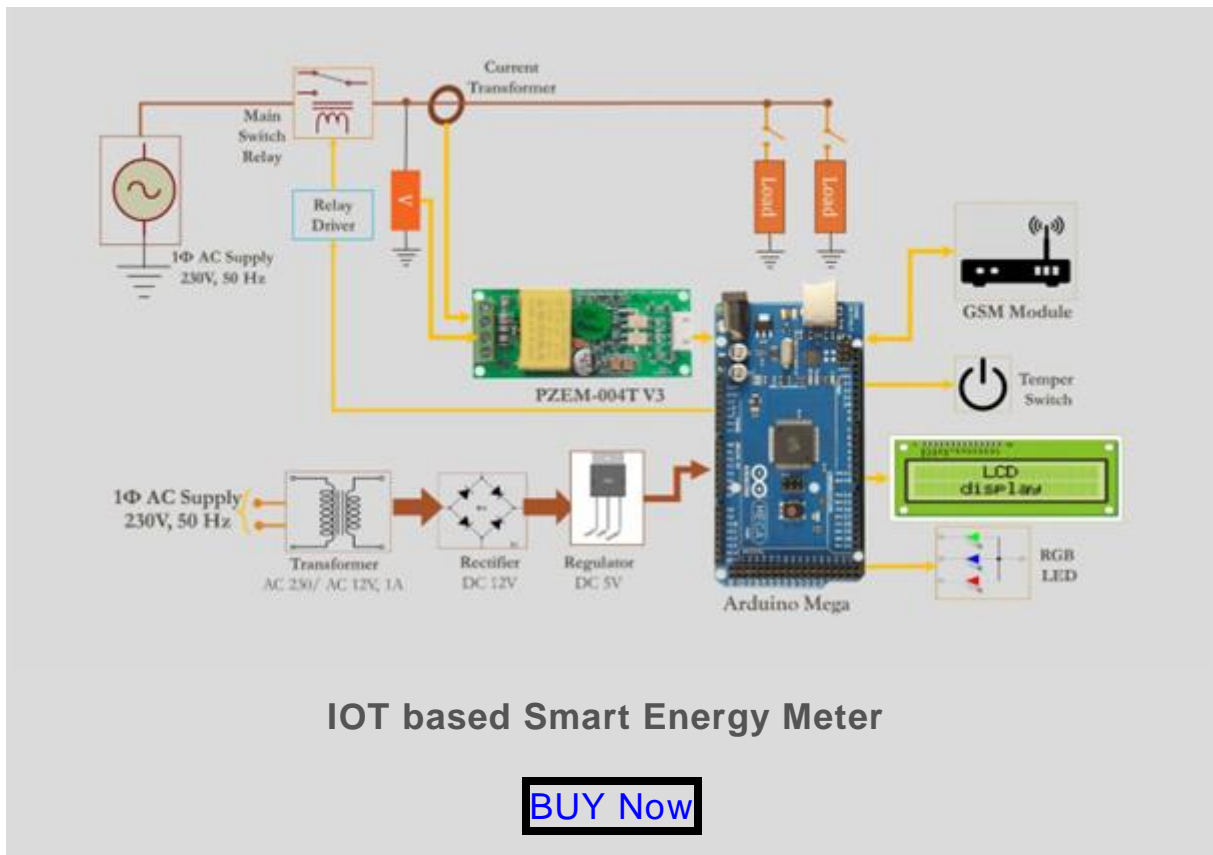
```

Conclusion

As you can see, having a such low-cost module you can design best DIY Projects for your academics, Hobby or personal use etc. Remember to be safe while working on it as it comes with high voltage AC supply. As given in this guide, different projects will determine how you will use this energy meter module.

I hope you have found some useful tips in this article to carry out your electronics projects. I would love and appreciate your suggestions on how you feel we can improve this article upon. If you have gained a few tips from this article and it has helped you in any way, be generous to share it with others.

Project Applications



More Great Project Using PZEM-004T V3

- AC Digital Multi function [Smart Meter using Arduino and PZEM-004T](#)
- Smart Energy meter for Automatic [Meter Reading \(AMR\) and theft Detection](#)
- PZEM-004T Alternative DIY [AC Power Measurement using Arduino](#)
- IOT Based Energy Meter Reading Using Arduino
- [fcgdam/PowerMeter: ESP8266 and PZEM-004T based Power Meter \(github.com\)](#)
- [ESP8266-PZEM004T](#) : A WiFi enabled PowerMeter for AC energy monitoring. Simple and elegant Web-Interface implementations, a RESTful API, OpenHAB 1 & 2 Callbacks and a MQTT client.