

IoT Based Smart Energy Meter Project

The **IoT Based Smart Energy Meter** is an **innovative project for monitoring electricity usage in real-time**. It combines **digital energy measurement, IoT connectivity, and automation** to help **users, industries, and students** track and manage power consumption. With **live data, mobile app dashboards, and smart alerts**, this project is ideal for **learning IoT, embedded systems, and smart grid concepts**.

“The IoT Smart Energy Meter empowers users to track electricity consumption in real-time, reducing wastage and enabling smart billing.”

Why Choose IoT Smart Energy Meter?

- Real-time monitoring of power consumption
- Wi-Fi enabled with IoT dashboard support
- Automatic alerts for over-usage or faults
- Data logging for monthly energy analysis
- Customizable for homes, industries, and research

Core Features & Technology

- Energy measurement using current/voltage sensors (e.g., ACS712, ZMPT101B)
- Microcontroller: ESP32/Arduino with Wi-Fi support
- IoT Cloud integration for live monitoring (Blynk, Thingspeak, Firebase, etc.)
- LCD/OLED display for local readings
- Mobile notifications & usage history

Key Components

- Current Sensor (ACS712 / SCT-013)
- Voltage Sensor (ZMPT101B)
- ESP32 / Arduino + Wi-Fi module
- 16x2 LCD / OLED Display
- IoT Cloud (Blynk, Thingspeak, Firebase)

Quick Technical Specs

Operating Voltage	3.3V (ESP32) / 5V (Arduino)
Communication	Wi-Fi, MQTT, REST API
Current Measurement Range	Up to 30A (ACS712)
Cloud Support	Blynk, Thingspeak, Firebase
Extras	Mobile app alerts & web dashboard

First Project: Energy Monitoring (Code Example)

```
// IoT Smart Energy Meter Example
#include <WiFi.h>
```

```

#include "ThingSpeak.h"

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASS";
WiFiClient client;

unsigned long channelID = YOUR_CHANNEL_ID;
const char* apiKey = "YOUR_API_KEY";

int sensorPin = 34; // ACS712 output pin
float calibration = 0.185; // 185mV per Amp for ACS712-5A

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi Connected");
  ThingSpeak.begin(client);
}

void loop() {
  int adcValue = analogRead(sensorPin);
  float voltage = (adcValue / 4095.0) * 3.3;
  float current = (voltage - 2.5) / calibration;

  Serial.print("Current: ");
  Serial.println(current);

  ThingSpeak.writeField(channelID, 1, current, apiKey);
  delay(15000); // Upload every 15 sec
}

```

Learning Outcomes

- Understand sensor interfacing with ESP32
- Learn how to measure current and voltage
- Experience cloud data upload using IoT platforms
- Build your first IoT-based Smart Energy Meter

Next Steps & Applications

After energy monitoring, students can expand to **smart billing**, **load control (relay-based ON/OFF)**, and **AI-powered energy optimization**. This project prepares learners for **smart grid**, **IoT automation**, and **renewable energy integration**.

Get Started with Smart Energy Meter at iTechBuilders

iTechBuilders provides **Smart Energy Meter kits**, **tutorials**, and **project mentorship** for **students**, **startups**, and **industries**. From **basic energy monitoring** to **advanced IoT billing systems**, we offer **DIY kits**, **training**, and **custom development**. **Contact us** to bring Smart Energy Meter innovation into your projects today!