**Connecting Smart Energy Meter with ZMPT101B and ESP32**

**1. Components Required**

* ESP32 (for processing and communication)
* ZMPT101B (Voltage Sensor)
* Current Sensor (ACS712 or SCT-013)
* Resistors & Capacitors (for calibration)
* MicroSD Card Module (for tracking data)
* Jumper Wires
* Breadboard or PCB

**2. Wiring the ZMPT101B to ESP32**

| **ZMPT101B Pin** | **ESP32 Pin** |
| --- | --- |
| **VCC** | **3.3V** |
| **GND** | **GND** |
| **OUT** | **ADC Pin (e.g., GPIO 34)** |

* Use a **voltage divider** to scale the output within ESP32’s ADC range.
* Adjust the **potentiometer** on the ZMPT101B for calibration.

**3. Connecting Current Sensor (ACS712 or SCT-013)**

| **Current Sensor Pin** | **ESP32 Pin** |
| --- | --- |
| **VCC** | **3.3V/5V** |
| **GND** | **GND** |
| **OUT** | **ADC Pin (e.g., GPIO 35)** |

* Convert **analog values to current (A)** using sensor-specific formulas.

**4. Storing Data in a File on SD Card**

**Connect MicroSD Module to ESP32**

| **SD Card Module Pin** | **ESP32 Pin** |
| --- | --- |
| **VCC** | **3.3V** |
| **GND** | **GND** |
| **CS** | **GPIO 5** |
| **MOSI** | **GPIO 23** |
| **MISO** | **GPIO 19** |
| **SCK** | **GPIO 18** |

**Arduino Code to Read and Store Data**

#include <SD.h>

#include <SPI.h>

#define VOLTAGE\_SENSOR\_PIN 34 // ZMPT101B

#define CURRENT\_SENSOR\_PIN 35 // ACS712

#define SD\_CS 5 // Chip Select for SD

File myFile;

void setup() {

Serial.begin(115200);

pinMode(VOLTAGE\_SENSOR\_PIN, INPUT);

pinMode(CURRENT\_SENSOR\_PIN, INPUT);

// Initialize SD card

if (!SD.begin(SD\_CS)) {

Serial.println("SD Card failed or not present!");

return;

}

Serial.println("SD Card Initialized.");

}

void loop() {

float voltage = analogRead(VOLTAGE\_SENSOR\_PIN) \* (3.3 / 4095.0) \* 100; // Adjust calibration

float current = analogRead(CURRENT\_SENSOR\_PIN) \* (3.3 / 4095.0) \* 30; // Adjust based on sensor

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

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

// Store in SD card

myFile = SD.open("/energy\_data.txt", FILE\_APPEND);

if (myFile) {

myFile.print("Voltage: "); myFile.print(voltage); myFile.print(" V, ");

myFile.print("Current: "); myFile.print(current); myFile.println(" A");

myFile.close();

}

delay(2000); // 2-second interval

}

**5. Data Tracking & Retrieval**

* The file energy\_data.txt will store continuous readings.
* Use SD.open("/energy\_data.txt") to **read past values** for tracking.

**Next Steps**

* Integrate **WiFi (ESP32) for remote monitoring** via an app or web dashboard.
* Send data to a **PostgreSQL database** for advanced analysis.
* Add **MQTT for IoT-based energy monitoring**.

**Need assistance with database integration or mobile app display?**