Smart Energy Meter with IoT for Real-Time Power Monitoring

#include <WiFi.h>

#include <ThingSpeak.h> // Install the ThingSpeak library via Arduino IDE

#include "EmonLib.h" // Energy Monitoring library

// WiFi credentials

const char\* ssid = "YOUR\_WIFI\_SSID";

const char\* password = "YOUR\_WIFI\_PASSWORD";

// ThingSpeak details

unsigned long channelNumber = YOUR\_CHANNEL\_NUMBER; // Replace with your ThingSpeak channel number

const char\* writeAPIKey = "YOUR\_API\_KEY"; // Replace with your ThingSpeak API key

// Create an instance of the EnergyMonitor class

EnergyMonitor emon1;

// WiFi client for ThingSpeak

WiFiClient client;

void setup() {

Serial.begin(115200);

// Connect to WiFi

WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi");

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("Connected to WiFi");

// Initialize ThingSpeak

ThingSpeak.begin(client);

// Initialize the current sensor (calibration factor may need adjustment)

emon1.current(34, 111.1); // Pin 34 and calibration factor for SCT-013

}

void loop() {

// Measure current and calculate power (P = V \* I)

double current = emon1.calcIrms(1480); // Sample size of 1480

double voltage = 230.0; // Assuming a constant voltage of 230V

double power = voltage \* current; // Power in watts

// Print readings to Serial Monitor

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

Serial.println(current);

Serial.print("Power (W): ");

Serial.println(power);

// Send data to ThingSpeak

ThingSpeak.setField(1, current); // Field 1 for current

ThingSpeak.setField(2, power); // Field 2 for power

// Write data to ThingSpeak

int statusCode = ThingSpeak.writeFields(channelNumber, writeAPIKey);

if (statusCode == 200) {

Serial.println("Data sent successfully");

} else {

Serial.print("Failed to send data, error code: ");

Serial.println(statusCode);

}

// Delay before next reading (e.g., 15 seconds)

delay(15000);

}