//Arduino code for Calculating reactive power and apparent power with different type of loads

#include <PZEM004Tv30.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#define SWITCH\_PIN 4 // Switch connected to digital pin 4

float tariff = 0;

float units = 0;

int displayMode = 0; // 0: Voltage & Current, 1: Tariff & Units, 2: Energy & Real Power, 3: Reactive and Apparent Power, 4: Frequency & Power Factor

LiquidCrystal\_I2C lcd(0x27, 16, 2);

PZEM004Tv30 pzem(8, 9); // Software Serial pin 8 (RX) & 9 (TX), connected to TX and RX of PZEM004T respectively

void setup() {

Serial.begin(9600);//baud rate 9600

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

pinMode(SWITCH\_PIN, INPUT\_PULLUP); // Set switch pin as input with pull-up resistor

}

void loop() {

// Read switch state and cycle display mode when pressed

if (digitalRead(SWITCH\_PIN) == LOW) {

displayMode = (displayMode + 1) % 5; // Cycle through 5 modes for lcd

delay(200); // Debounce delay

}

float voltage = pzem.voltage();

float current = pzem.current();

float power = pzem.power();

float energy = pzem.energy();

float frequency = pzem.frequency();

float pf = pzem.pf();

float reactive = voltage\*current\*sqrt(1-sq(pf));

float apparent = sqrt(sq(power) + sq(reactive));

// Accumulate energy and calculate tariff

if (!isnan(power)) {

float energyPerSecond = power / (1000.0 \* 3600.0);

float unitsPerSecond = energyPerSecond / 0.001; //for example, taking 1 unit as 0.001kwh

units += unitsPerSecond;

if (units < 5) {

tariff += 2 \* unitsPerSecond;

} else if (units < 10) {

tariff += 3.5 \* unitsPerSecond;

} else if (units < 15) {

tariff += 5 \* unitsPerSecond;

} else {

tariff += 8 \* unitsPerSecond;

}

}

// Clear LCD before updating

lcd.clear();

// LCD Display Mode Switching

switch (displayMode) {

case 0:

lcd.setCursor(0, 0);

lcd.print("V:");

lcd.print(voltage);

lcd.setCursor(0, 1);

lcd.print("I:");

lcd.print(current);

break;

case 1:

lcd.setCursor(0, 0);

lcd.print("Units:");

lcd.print(units, 2);

lcd.setCursor(0, 1);

lcd.print("Tariff: ");

lcd.print(tariff, 2);

break;

case 2:

lcd.setCursor(0, 0);

lcd.print("Energy:");

lcd.print(energy, 3);

lcd.setCursor(0, 1);

lcd.print("Power:");

lcd.print(power);

break;

case 3:

lcd.setCursor(0, 0);

lcd.print("Reactive:");

lcd.print(reactive, 3);

lcd.setCursor(0, 1);

lcd.print("Apparent:");

lcd.print(apparent);

break;

case 4:

lcd.setCursor(0, 0);

lcd.print("Freq:");

lcd.print(frequency);

lcd.setCursor(0, 1);

lcd.print("PF:");

lcd.print(pf);

break;

}

// Serial Monitor Output

Serial.println("===== PZEM Readings =====");

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

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

Serial.print("Real Power: "); Serial.print(power); Serial.println(" W");

Serial.print("Reactive Power: "); Serial.print(reactive); Serial.println(" VAR");

Serial.print("Apparent Power: "); Serial.print(power); Serial.print(" + j");Serial.print(reactive);Serial.println(" VA");

Serial.print("Energy: "); Serial.print(energy, 3); Serial.println(" kWh");

Serial.print("Frequency: "); Serial.print(frequency); Serial.println(" Hz");

Serial.print("Power Factor: "); Serial.println(pf);

Serial.println("===== Tariff & Units =====");

Serial.print("Units: "); Serial.println(units, 3);

Serial.print("Tariff: "); Serial.println(tariff, 3);

Serial.println("==========================\n");

delay(1000);

}