```
#include "ZMPT101B.h"
#include "ACS712.h"
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>


ZMPT101B voltageSensor(34);
ACS712 currentSensor(ACS712_5A, 36); // Use ACS712_30A for a 30A sensor


#define SCREEN_WIDTH 128 // OLED display width, in pixels
```

```
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

#define SCREEN_ADDRESS 0x3C // See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32



Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);



float P = 0;

float U = 0;

float I = 0;

long dt = 0;

float CulmPwh = 0;
float units = 0;
long changeScreen = 0;
float lastSample = 0;
unsigned long lasttime = 0;
long ScreenSelect = 0;

void setup() {
Serial.begin(9600);
delay(100);

if (!voltageSensor.begin()) {
  Serial.println("Failed to initialize ZMPT101B");
  while (1);
}

if (!currentSensor.begin()) {
  Serial.println("Failed to initialize ACS712");
  while (1);
}

voltageSensor.setCalibration(2621, 0.0025);
currentSensor.setCalibration(2943, 0.15);
```

```
if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
  Serial.println(F("SSD1306 allocation failed"));
  for (;;);
}

display.clearDisplay();
display.display();
}

void loop() {
```

```
U = voltageSensor.getVoltageRMS();
if (U < 55) {
  U = 0;
  CulmPwh = 0;
}

I = currentSensor.getCurrentDC();
if (I < 0.15) {
  I = 0;
  CulmPwh = 0;
}
```

```
P = U * I;
CulmPwh = CulmPwh + P * (dt / 3600); // uWh
units = CulmPwh / 1000;

if (millis() - changeScreen > 5000) {
  ScreenSelect += 1;
  changeScreen = millis();
}

if (millis() - lasttime > 500) {
  if ((ScreenSelect % 4) == 0) {
    displayVoltCurrent();
  } else if ((ScreenSelect % 4) == 1) {
    displayInstPower();
  } else if ((ScreenSelect % 4) == 2) {
    displayEnergy();
  } else if ((ScreenSelect % 4) == 3) {
    displayUnits();
  }
}
```

```
lastSample = micros();

}



void displayVoltCurrent() {

display.clearDisplay();

display.setTextColor(WHITE);

display.setTextSize(3);

displayCenter(String(U) + "V", 3);

display.setTextSize(3);

displayCenter(String(I) + "A", 33);

display.display();
```

```
lasttime = millis();
}

void displayInstPower() {
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(0, 0);
displayCenter("Power", 3);
display.setTextSize(3);
if (P > 1000) {
  displayCenter(String(P / 1000) + "kW", 30);

} else {

  displayCenter(String(P) + "W", 30);

}

display.display();

lasttime = millis();

}



void displayEnergy() {

display.clearDisplay();

display.setTextColor(WHITE);
```

```
if (CulmPwh > 1000000000) {
  display.setTextSize(2);
  displayCenter("Energy kWh", 3);
  display.setTextSize(3);
  displayCenter(String(CulmPwh / 1000000000), 30);
} else if (CulmPwh < 1000000000 && CulmPwh > 1000000) {
  display.setTextSize(2);
  displayCenter("Energy Wh", 3);
  display.setTextSize(3);
  displayCenter(String(CulmPwh / 1000000), 30);
} else if (CulmPwh < 1000000 && CulmPwh > 1000) {
  display.setTextSize(2);

  displayCenter("Energy mWh", 3);

  display.setTextSize(3);

  displayCenter(String(CulmPwh / 1000), 30);

} else {

  display.setTextSize(2);

  displayCenter("Energy uWh", 3);

  display.setTextSize(3);

  displayCenter(String(CulmPwh), 30);

}

display.display();
```

```
lasttime = millis();

}



void displayUnits() {

display.clearDisplay();

display.setTextColor(WHITE);

if (units > 1000000) {

  display.setTextSize(2);

  displayCenter("Units", 3);

  display.setTextSize(3);

  displayCenter(String(units / 1000000), 30);
} else if (units < 1000000 && units > 1000) {
  display.setTextSize(2);
  displayCenter("MilliUnits", 3);
  display.setTextSize(3);
  displayCenter(String(units / 1000), 30);
} else {
  display.setTextSize(2);
  displayCenter("MicroUnits", 3);
  display.setTextSize(3);
  displayCenter(String(units), 30);
}
```

```
display.display();
lasttime = millis();
}

void displayCenter(String text, int line) {
int16_t x1;
int16_t y1;
uint16_t width;
uint16_t height;
display.getTextBounds(text, 0, 0, &x1, &y1, &width, &height);
display.setCursor((SCREEN_WIDTH - width) / 2, line);
display.println(text);

display.display();

}
```