

```

#include <Wire.h>
#include <Adafruit_ADS1X15.h>

Adafruit_ADS1115 ads;

// Parámetros de tu sistema
const float V_RMS = 110.0; // Voltaje de la red (110V en tu caso)
const float BURDEN_RESISTOR = 50.0; // Burden en ohmios
const int CT_RATIO = 2000; // Para un CT tipo 100A:50mA → 2000
const int SAMPLES = 1000; // Número de muestras para calcular RMS

void setup() {
  Serial.begin(115200);
  Wire.begin(21, 22); // Pines I2C ESP32: SDA=21, SCL=22

  if (!ads.begin()) {
    Serial.println("Error: no se detecta el ADS1115");
    while (1);
  }

  // Ganancia: ajusta según tu señal
  // GAIN_ONE = ±4.096V → 0.125mV por LSB
  // GAIN_TWO = ±2.048V → 0.0625mV por LSB
  // GAIN_EIGHT = ±0.512V → 0.015625mV por LSB (mejor para señales pequeñas)
  ads.setGain(GAIN_EIGHT);

  Serial.println("Medidor de energía iniciado...");
}

void loop() {
  float sumSq = 0;

  for (int i = 0; i < SAMPLES; i++) {
    int16_t raw = ads.readADC_SingleEnded(0); // Canal A0
    float volts = raw * 0.015625e-3; // GAIN_EIGHT → 0.015625 mV/LSB

    // Restar bias (~1.65V en tu divisor 10k-10k)

```

```
float centered = volts - 1.65;

sumSq += centered * centered;
}

// Voltaje RMS en burden
float V_RMS_CT = sqrt(sumSq / SAMPLES);

// Corriente en el secundario del CT
float I_RMS_CT = V_RMS_CT / BURDEN_RESISTOR;

// Corriente real en la línea
float I_RMS = I_RMS_CT * CT_RATIO;

// Potencia (asumiendo  $\cos\phi \approx 1$ , ventilador resistivo-inductivo ligero)
float P = V_RMS * I_RMS;

Serial.print("Corriente RMS: ");
Serial.print(I_RMS, 3);
Serial.print(" A | Potencia: ");
Serial.print(P, 1);
Serial.println(" W");

delay(1000);
}
```