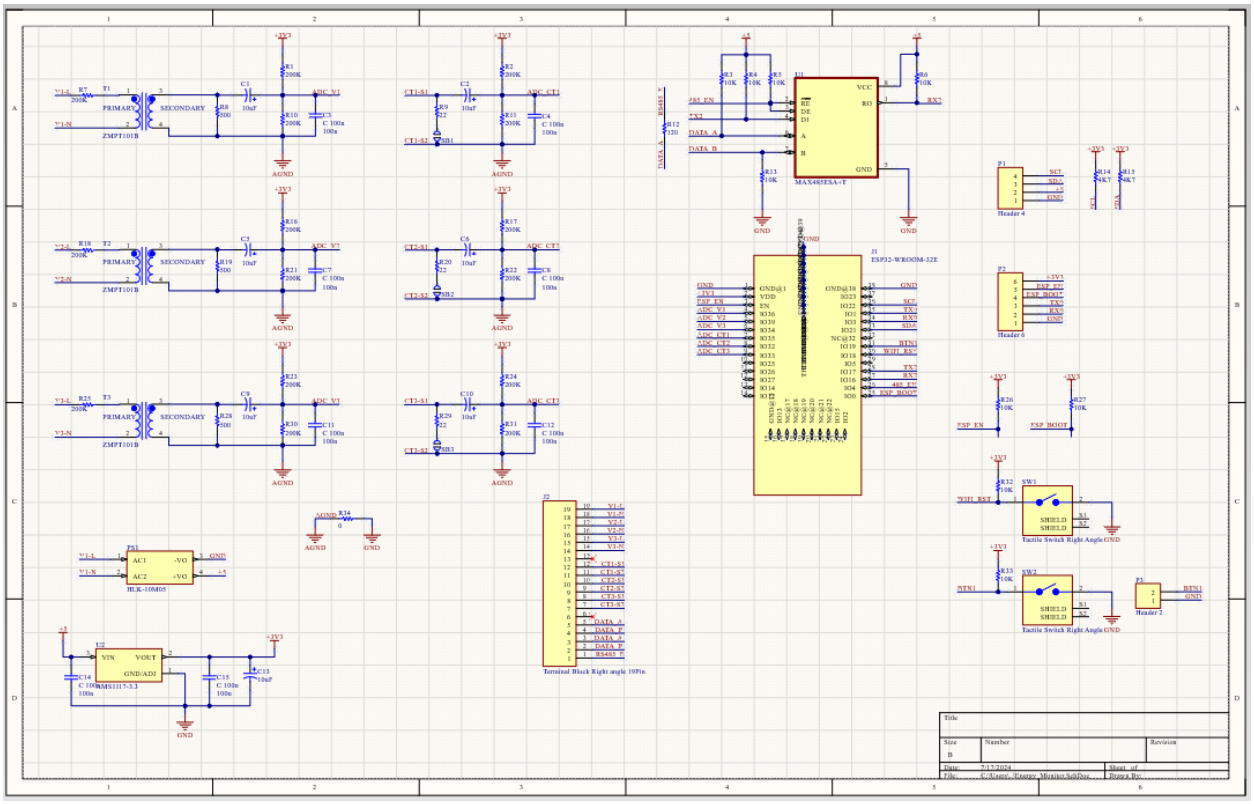
Energy Monitor Project

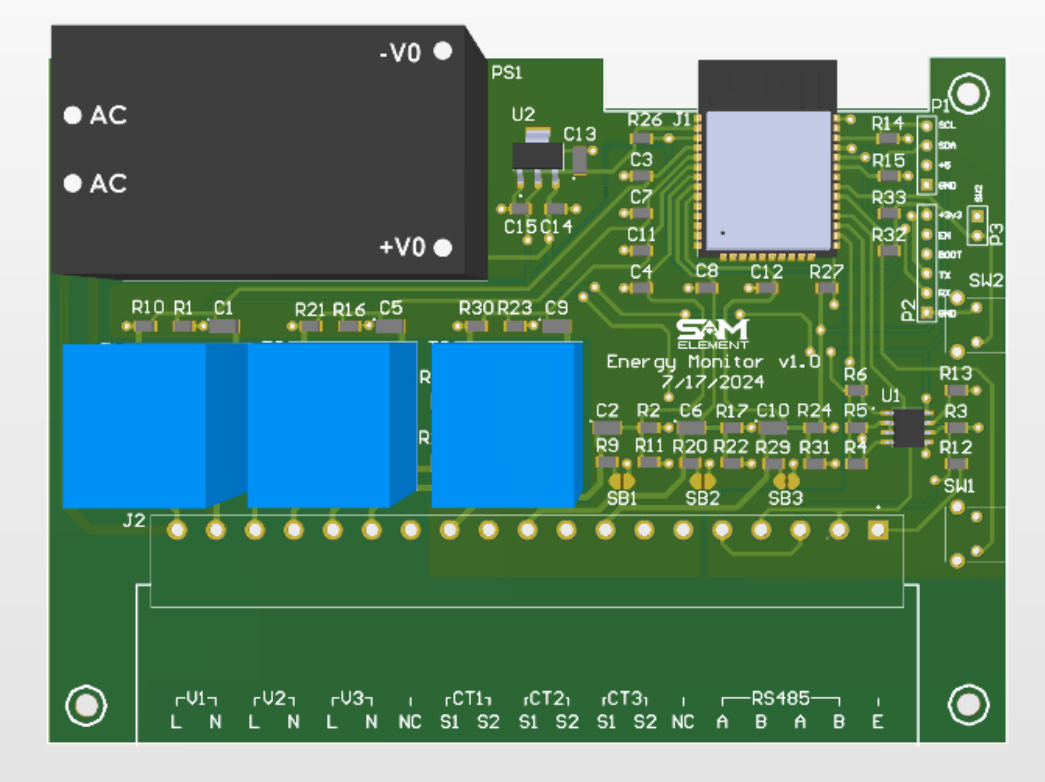
Fitur

- 3 channel pengukuran (3 pasang sensor tegangan dan arus).
- IoT.
- LCD Display 20x4.
- Komunikasi RS485 (Modbus RTU).

Hardware

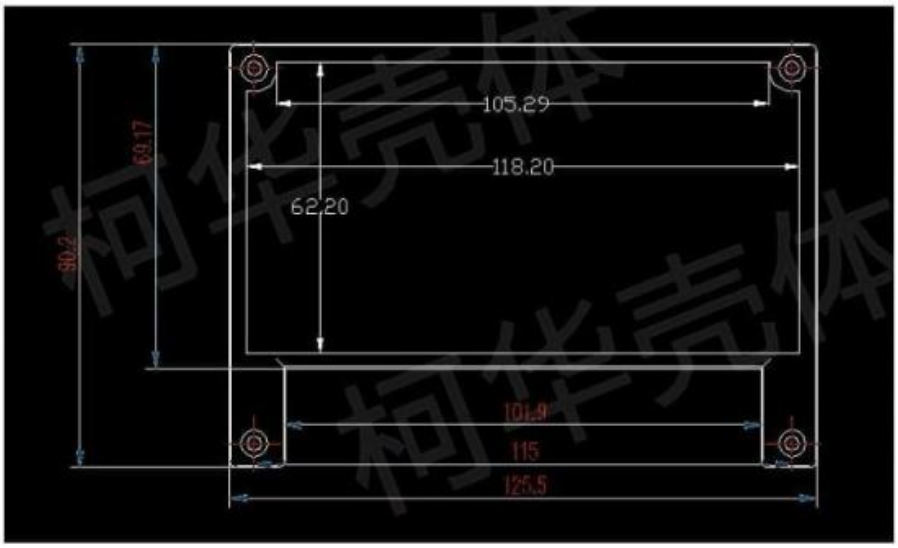


skema keseluruhan.



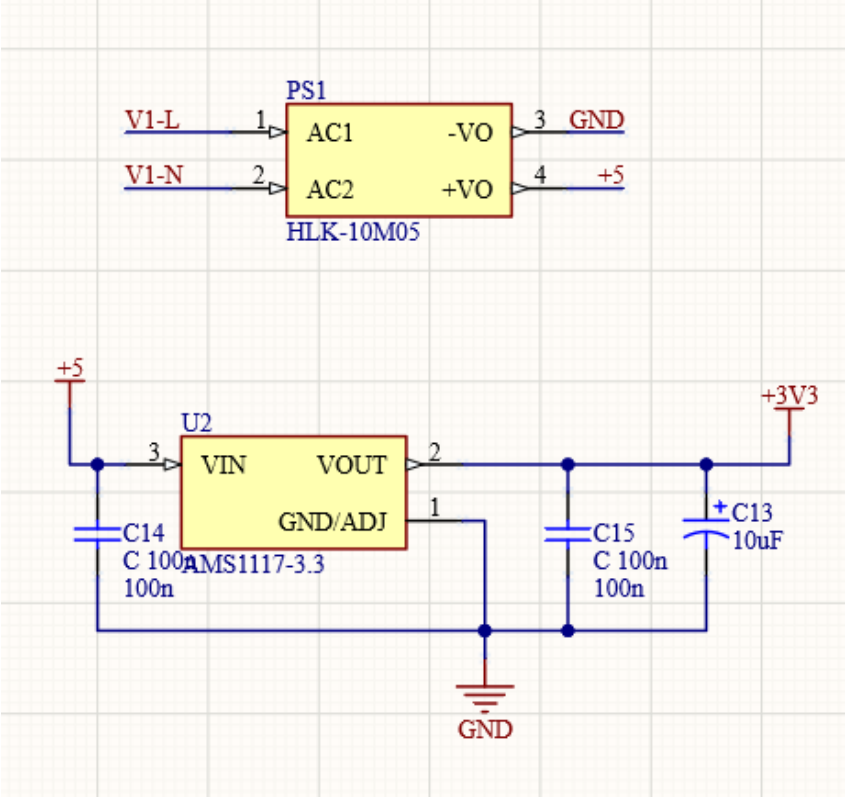
Layout PCB

Untuk dimensi board dan posisi lubang, disesuaikan dengan box berikut:
https://www.tokopedia.com/jogjarobotika/plastic-industrial-box-plc-2-02c-145x90x41mm?extParam=src%3Dshop%26whid%3D3000837



Power Supply.

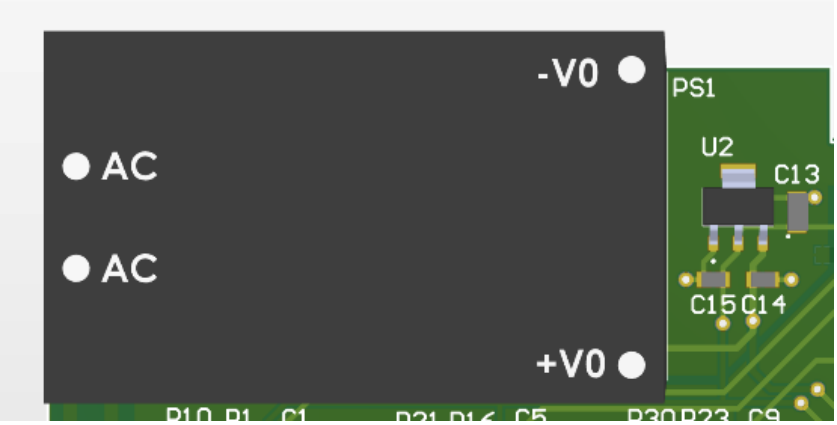
Power supply menggunakan modul HLK10M05 (5V 2A). dan untuk tegangan 3v3 menggunakan ams1117 3v3.



Skema power Supply.

Input HLK10M05 didapat langsung dari AC220V. dengan jalur yang sama dengan input sensor tegangan channel 1.

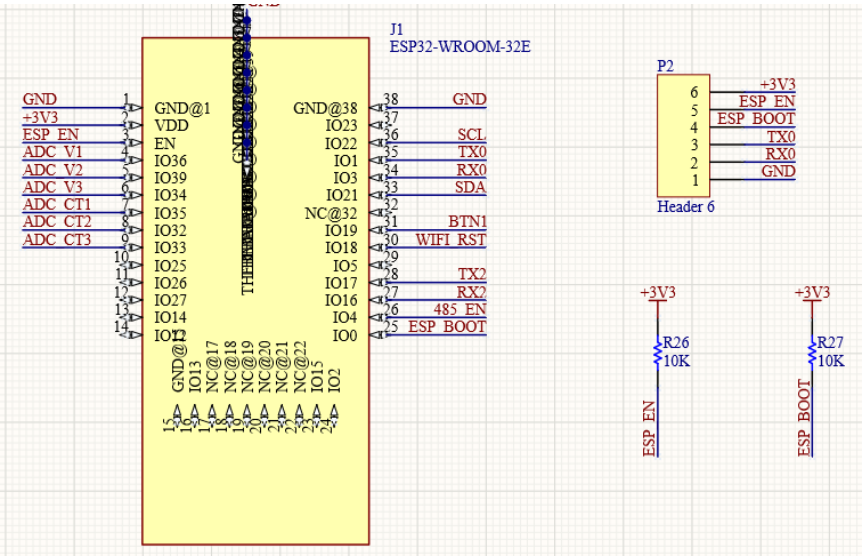
https://www.tokopedia.com/iotstore/hi-link-hlk-10m05-ac-to-dc-isolated-5v-2a-10-watt?extParam=ivf%3Dfalse%26keyword%3Dhlk+5v%26search_id%3D20240708093206F299CAA2C12B660BAYAD%26src%3Dsearch



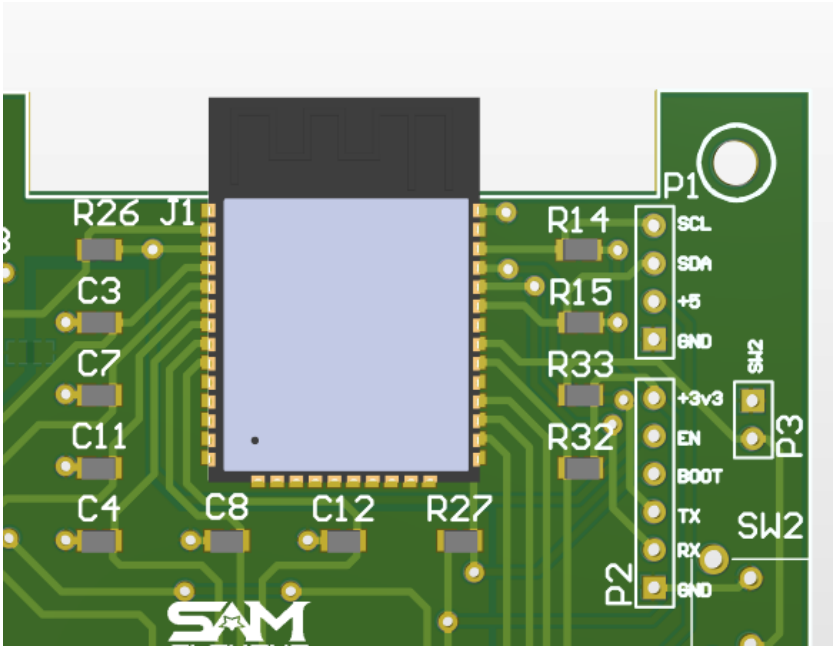
Posisi power Supply pada PCB di kanan atas.

Controller

Controller menggunakan esp32.



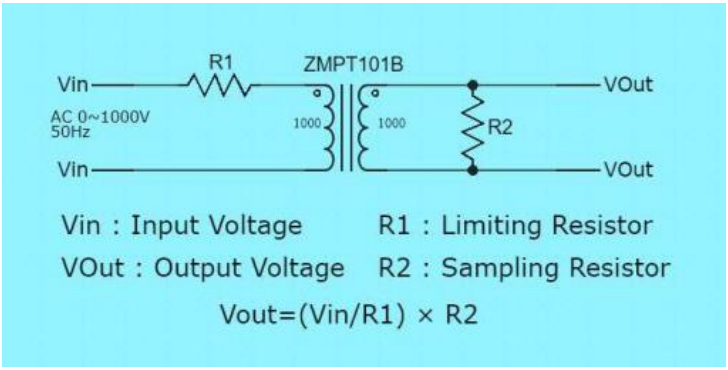
ESP32 beserta pin mapping



ESP32 diletakkan pada kanan atas PCB.

Sensor Tegangan

Komponen utama dalam sensor tegangan adalah trafo arus ZMPT101B. R7 merupakan current limiting resistor (membatasi arus pada trafo) dan R8 merupakan sampling resistor (mengonversi lagi menjadi tegangan).



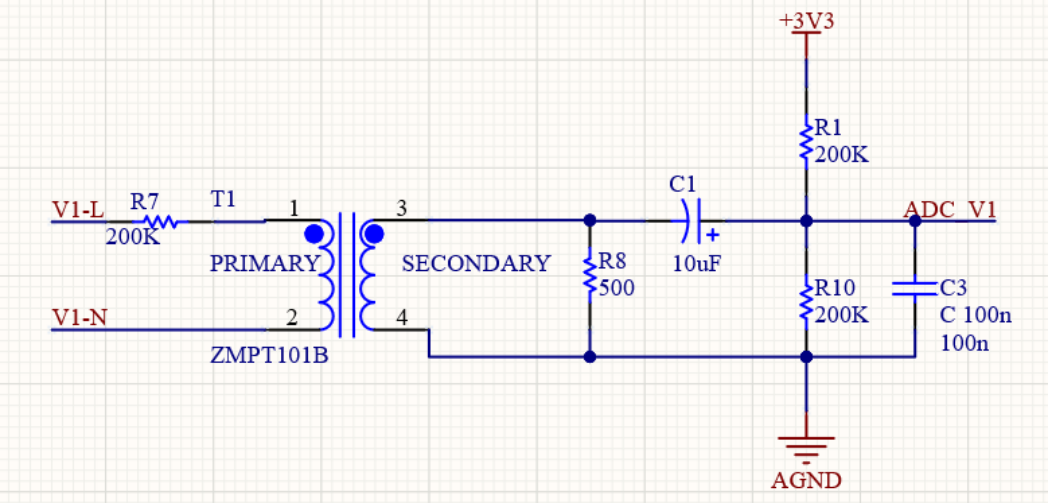
Karena tegangan AC memiliki nilai negative. Maka sebelum masuk adc microcontroller dilakukan offset tegangan (menggunakan R1 dan R10) agar titik nol vac menjadi setengah dari nilai max adc (3.3/2 = 1.65V).

C1 digunakan untuk coupling dan c3 agar ADC yang terbaca lebih stabil.

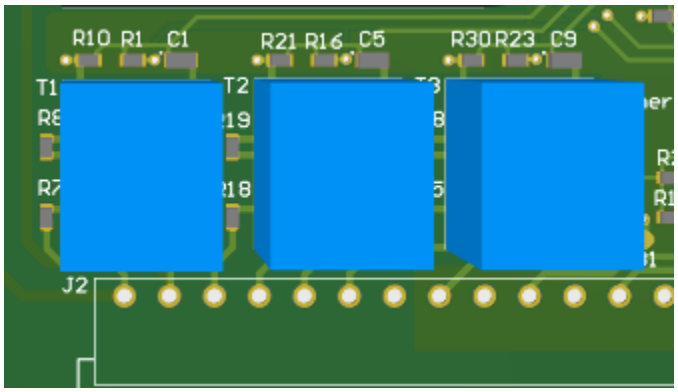
referensi dapat dilihat melalui link:

https://innovatorsguru.com/wp-content/uploads/2019/02/ZMPT101B.pdf

https://docs.openenergymonitor.org/electricity-monitoring/voltage-sensing/measuring-voltage-with-an-acac-power-adapter.html



Rangkaian sensor tegangan dengan ZMPT101B.



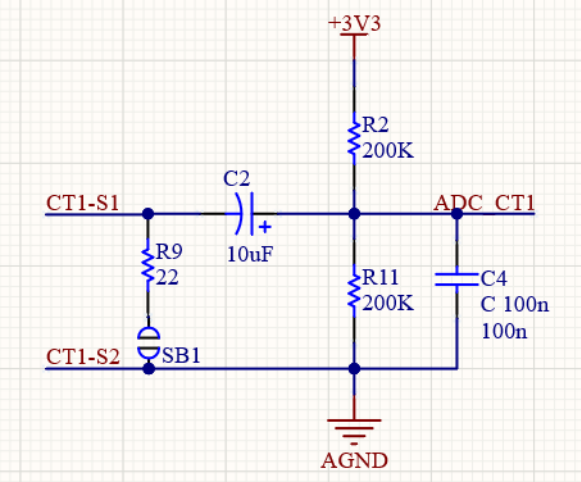
Rangkaian sensor tegangan terdapat di sisi bawah kiri.

Sensor Arus

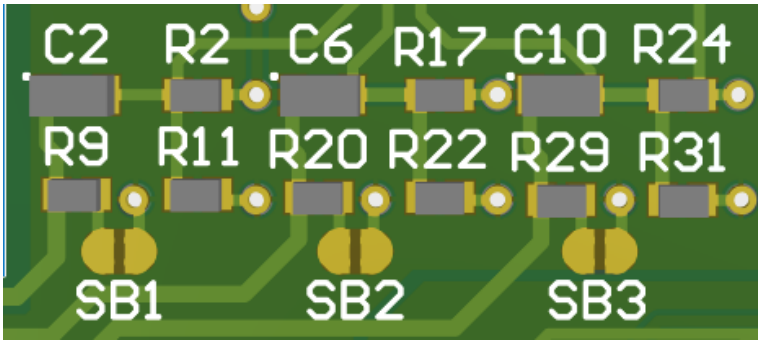
Untuk sensor arus, menggunakan CT clamp SCT-013-xxx. output dari CT tersebut dapat berupa arus atau tegangan tergantung serinya.

Skema rangkaian sama dengan sensor arus. Terdapat R9 sebagai burden resistor, diperlukan jika CT memilliki keluaran arus. Terdatap solder bridge do bawahnya. Dihubungkan jika CT yang digunakan memiliki keluaran arus. Dibiarkan open jika keluaran CT sudah berupa tegangan.

R2 dan R11 merupakan pembagi tegangan untuk offset sinyal. C2 coupling dan c4 untuk menstabilkan pembacaan adc.



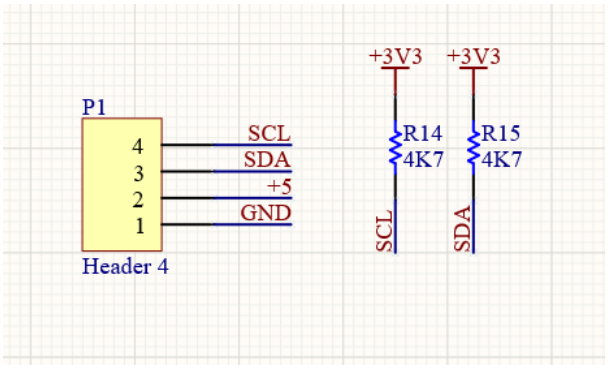
Skema untuk sensor arus.



Layout sensor arus

Display

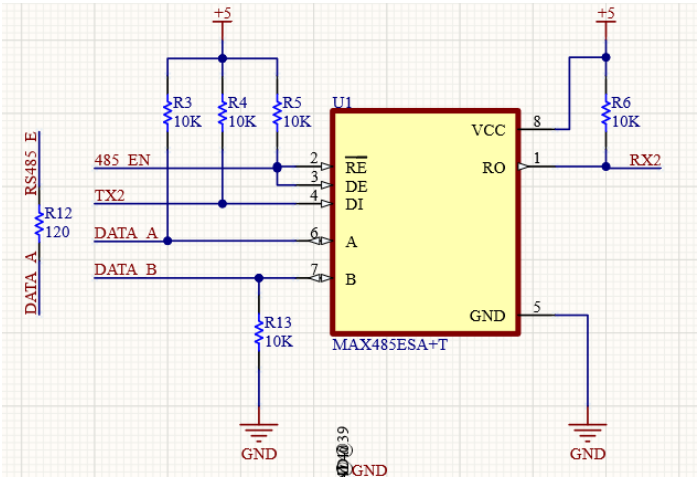
Display direncanakan menggunakan LCD20x4 dengan I2c. sehingga pada PCB hanya diperlukan pin header.



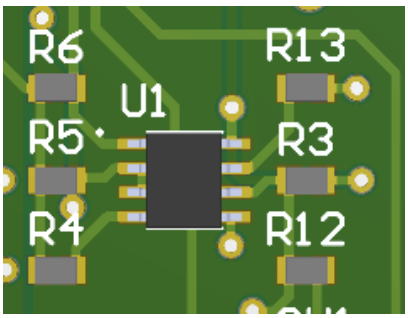
Skema untuk connector I2C

Komunikasi

Untuk komunikasi terdapat RS485. Sehingga data selain dapat dilihat pada display LCD dan IoT. Juga dapat diakses dengan device lain dengan protocol modbus RTU. Digunakan IC Max485.



Rangkaian Max485

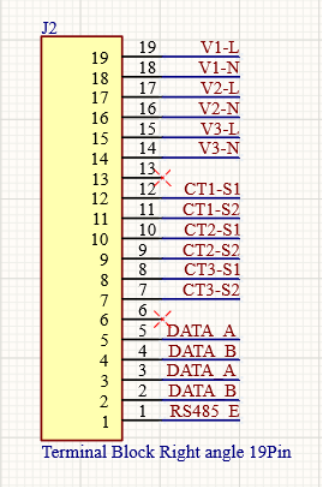


Layout pcb max485

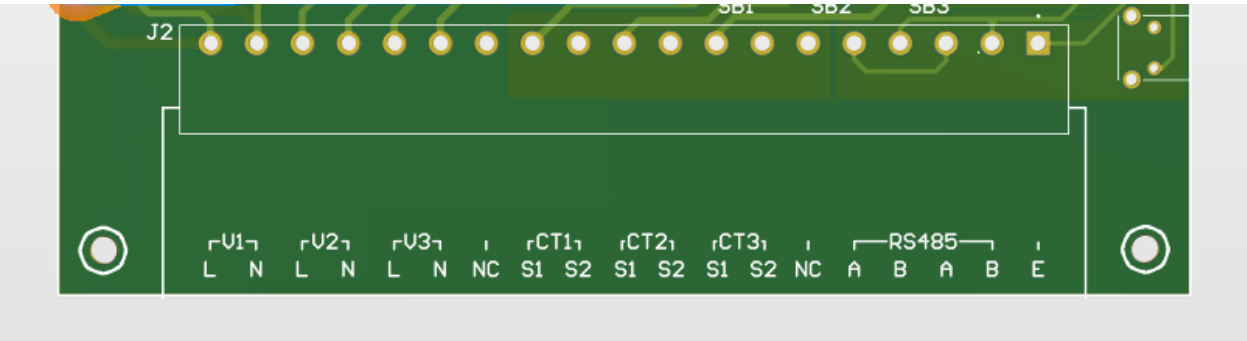
Terminal

Untuk terminal yang digunakan adalah terminal blok hijau siku yang sudah sering digunakan. Terdapat 19 pin (selain kebutuhan, disesuaikan juga dengan box). Dapat mengguanakan 5 buah terminal blok 3pin dan 1 buah terminal blok 4pin.

V1-L dan V1-N untuk line dan netral tegangan ac yang akan diukur. CT-S1 dan CT-S2 untuk input dari CT clamp. Data A dan data B untuk RS485. Dan RS485E untuk resistor terminator RS485.



Pin untuk terminal blok.



Pada overlay pcb juga diberi keterangan kegunaan masing masing pin.

Komunikasi

Komunikasi menggunakan protocol modbus sebagai berikut. Setiap data yang disimpan pada register dikalikan dengan 100.

Function code	Address	Parameter	unit	type
04 (read input register	0000	Voltage 1	V	R
	0002	Voltage 2	V	R
	0004	Voltage 3	V	R
	0006	Current 1	I	R
	0008	Current 2	I	R
	000A	Current 3	I	R
	000C	Real Power 1	W	R
	000E	Real Power 2	W	R
	0010	Real Power 3	W	R
	0012	Apparent Power 1	VA	R
	0014	Apparent Power 2	VA	R
	0016	Apparent Power 3	VA	R
	0018	Energy 1	kWh	R
	001A	Energy 2	kWh	R
	001C	Energy 3	kWh	R
	001E	Energy Total	kWh	R
	0020	Power Factor 1		R
	0022	Power Factor 2		R
	0024	Power Factor 3		

Function Code	Address	Parameter	value	type
06 (Write Single Register)	2000	Communication speed	0: 1200 1: 2400 2: 4800 3: 9600 4: 19200 5: 38400 6: 57600 7: 115200	R/W
	2001	Data length	0: 7 bit 1: 8 bit	R/W
	2002	Parity	0: None 1: Odd 2: Even	R/W
	2003	Stop bit	0: 1bit 1: 2bit	R/W

Firmware

Untuk library yang digunakan adalah sebagai berikut:

```
#include "EmonLib.h"  //https://github.com/Savjee/EmonLib-esp32
#include <Adafruit_SSD1306.h>
#include <esp_modbus_slave.h>
#include <ModbusRTU.h>
#include <ArduinoJson.h>
```

Task yang dibuat.

```
// xTaskCreate(task_readVI1, "readVI line 1", 1024 * 8, NULL, 5, NULL);
xTaskCreate(task_readVI2, "readVI line 2", 1024 * 8, NULL, 5, NULL);
// xTaskCreate(task_readVI3, "readVI line 3", 1024 * 8, NULL, 5, NULL);
xTaskCreate(task_calcKwh, "calculate kwh", 1024 * 4, NULL, 1, NULL);
xTaskCreate(task_display, "display oled", 1024 * 2, NULL, 1, NULL);
xTaskCreate(task_modbus, "modbus", 1024 * 4, NULL, 1, NULL);
```

task yang belum dibuat

- Task button
- Task iot
- Task display (menggunakan lcd)

Task readVI

```
void task_readVI2(void *pvParameter) {

  emon2.voltage(35, 402.6, 1.7);  // Voltage: input pin, calibration, phase_shift
  emon2.current(15, 18.6);        // Current: input pin, calibration.

  while (1) {
    emon2.calcVI(20, 2000);  // Calculate all. No.of half wavelengths (crossings), time-out
    dataL2.realPower = emon2.realPower;
    xQueueSend(calcKwh2_queue, &dataL2.realPower, portMAX_DELAY);
    dataL2.apparentPower = emon2.apparentPower;
    dataL2.powerFactor = emon2.powerFactor;
    dataL2.Vrms = emon2.Vrms;
    dataL2.Irms = emon2.Irms;

    Serial.printf("Vrms : %f;   Irms : %f;    real : %f;   apparent : %f;   pf : %f; \r\n",
    vTaskDelay(1);
    yield();
  }
}
```

Task ini berfungsi untuk mengatasi pembacaan voltage, current dan perhitungan power. Menggunakan library emonLib.h

Emon.voltage() merupakan inisiasi dari pin adc serta konstanta calibration. Sama juga dengan emon.current(). Nilai konstan calilbration disesuaikan dengan nilai pembacaan sensor dan nilai pembacaan calibrator.

Emon.calcVI() digunakan untuk mulai pengambilan data dan perhitungan. Jika selesai, data dapat diambil melalui variabel tersebut.

Terdapat queue send. Digunakan untuk menandakan data siap untuk dilakukan perhitungan kwh. Perhitungan kwh terdapat pada task yang berbeda.

Task calcKwh

```
void task_calcKwh(void *pvParameter) {
  float realPower;
  float kwh1, mwh1, kwh2, mwh2, kwh3, mwh3;
  unsigned long int lastmillis;

  while (1) {
    if (xQueueReceive(calcKwh1_queue, &realPower, portMAX_DELAY)) {
      mwh1 = mwh1 + realPower * (millis() - lastmillis) / 3600.0;
      if (mwh1 / 10000.0 >= 1) {
        Serial.println("here");
        kwh1 = kwh1 + (mwh1 / 1000000.0);
        mwh1 = fmod(mwh1, 10000);
      }
      lastmillis = millis();
      dataL1.kwh = mwh1;
    }
```

Task yang berfungsi melakukan perhitungan kwh. Untuk mendapatkan kwh, digunakan nilai real power yang dikalikan waktu (jam).

Data real power didapat setiap 200-300 ms. Sehingga jika dihitung dalam kwh akan sangat kecil. Maka dari itu dihitung dalam mWh(milli watt hour) terlebih dahulu. Tiap 10000 mWh akan dikonversi menjadi 0.01kWh.

Task display

Untuk saat ini task display hanya digunakan untuk menampilkan data (yang penting tampil dan bisa dilihat datanya). Untuk tampilan sebenenarnya masih belum. Menggunakan oled 64x32 pixel.

```ino
void task_display(void *pvParameter) {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();

  while (1) {
    display.clearDisplay();
    display.setTextColor(1);
    display.setCursor(6, 5);
    display.print("V");
    display.setCursor(18, 5);
    display.print(dataL1.Vrms);
    display.setCursor(6, 20);
    display.print("I");
    display.setCursor(18, 20);
    display.print(dataL1.Irms);
    display.setCursor(69, 5);
    display.print("W");
    display.setCursor(85, 5);
    display.print(dataL1.realPower);
    display.setCursor(64, 20);
    display.print("kWh");
    display.setCursor(84, 20);
    display.print(dataL1.kwh);
    display.drawLine(60, 4, 60, 27, 1);
    display.display();
    vTaskDelay(1000);
  }
}
```

Task modbus

Untuk task modbus menggunakan library ModbusRTU.h
https://github.com/emelianov/modbus-esp8266

```
void task_modbus(void *pvParameter) {
  Serial2.begin(baudrate[mbConf.baudrate] , serialConfig(mbConf.dataLength, mbConf.parity, mbConf.stopBit));
  mb.begin(&Serial2, RS485_ENABLE_PIN);

  mb.server(1);

  for (int i = 0; i < 0x2D; i++) {
    mb.addIreg(i);
  }

  for(int i =0; i<8; i++){
    mb.addHreg(0x2000 + i);
  }

  mb.Hreg(0x2000, mbConf.baudrate);
  mb.Hreg(0x2001, mbConf.dataLength);
  mb.Hreg(0x2002, mbConf.parity);
  mb.Hreg(0x2003, mbConf.stopBit);

  mb.onSetHreg(0x2000, cb_baudrate);
  mb.onSetHreg(0x2001, cb_dataLength);
  mb.onSetHreg(0x2002, cb_parity);
  mb.onSetHreg(0x2003, cb_stopbit);
  // mb.onSetHreg(1, cb1);
```

```
while (1) {
  mb.Ireg(0, (int)(dataL1.Vrms * 100) >> 16);
  mb.Ireg(1, dataL1.Vrms * 100);
  mb.Ireg(2, (int)(dataL2.Vrms * 100) >> 16);
  mb.Ireg(3, dataL2.Vrms * 100);
  mb.Ireg(4, (int)(dataL3.Vrms * 100) >> 16);
  mb.Ireg(5, dataL3.Vrms * 100);

  mb.Ireg(6, (int)(dataL1.Irms * 100) >> 16);
  mb.Ireg(7, dataL1.Irms * 100);
  mb.Ireg(8, (int)(dataL2.Irms * 100) >> 16);
  mb.Ireg(9, dataL2.Irms * 100);
  mb.Ireg(0xA, (int)(dataL3.Irms * 100) >> 16);
  mb.Ireg(0xB, dataL3.Irms * 100);

  mb.Ireg(0xC, (int)(dataL1.realPower * 100) >> 16);
  mb.Ireg(0xD, dataL1.realPower * 100);
  mb.Ireg(0xE, (int)(dataL2.realPower * 100) >> 16);
  mb.Ireg(0xF, dataL2.realPower * 100);
  mb.Ireg(0x10, (int)(dataL3.realPower * 100) >> 16);
  mb.Ireg(0x11, dataL3.realPower * 100);
```

Untuk modbus, harus inisiasi serial terlebih dahulu, disini menggunakan serial2. Kemudian baru dapat digunakan mb.begin();

mb.server(1) digunakan untuk inisiasi bahwa modbus digunakan sebagai slave dengan address 1.

mb.addIreg() digunakan untuk membuat register input baru. Sedangkan untuk mb.Ireg() untuk mengisi data pada register tersebut.