Membaca sensor eksternal (BMP 280 & BMP 180)

Section 1

Selamat berjumpa kembali teman-teman, kali ini kita akan mencoba membaca sensor eksternal dengan menggunakan mikrokontroller ESP-32. Sensor yang akan kita baca adalah sensor BMP 280 dan 180, kedua sensor tersebut sama-sama mengukur tekanan tekanan dan suhu udara, dipasaran bisa kita dapat membeli dengan harga delapan ribu sampai dua puluh ribu.

Beberapakali pengalaman saya saat membaca sensor dengan mikrokontroler tidak selalu menyenangkan, bahkan lebih sering kita harus belajar dan belajar lagi. Namun tidak mengapa teman-teman, yah demi mencapai kuota pengalaman 10.000 jam terbang seperti kata **Pak Sony**. Bahkan sampai saat ini, ketika kita bersama-sama membuat beberapa platform IoT, tetap saja mendebarkan menunggu pembacaan sensor. Karena bagaimanapun *sensing* adalah terkait keterbacaan sensor, ketika sensor sudah terbaca metode yang lain-lainnya dapat kita sesuaikan.

Berkaca dari hal tersebut, saya mencoba berbagi, apa sih yang harusnya pertama kali dilakukan untuk membaca sensor eksternal dengan ESP 32?. Sepertinya pepatah "Tak kenal maka tak sayang" tetap relevan untuk masalah ini, dimana kita harus mengenal sensor yang akan kita gunakan. Setelah kita kenal, dimana *address*-nya, protokol komunikasinya seperti apa, maksimal tegangannya berapa, baru kita mencoba untuk menjalin hubungan. Nah, agar kenal kami sertakan dalam tabel berikut, datasheet untuk mengenal lebih dekat dengan dua sensor yang akan kita gunakan dalam percobaan ini.

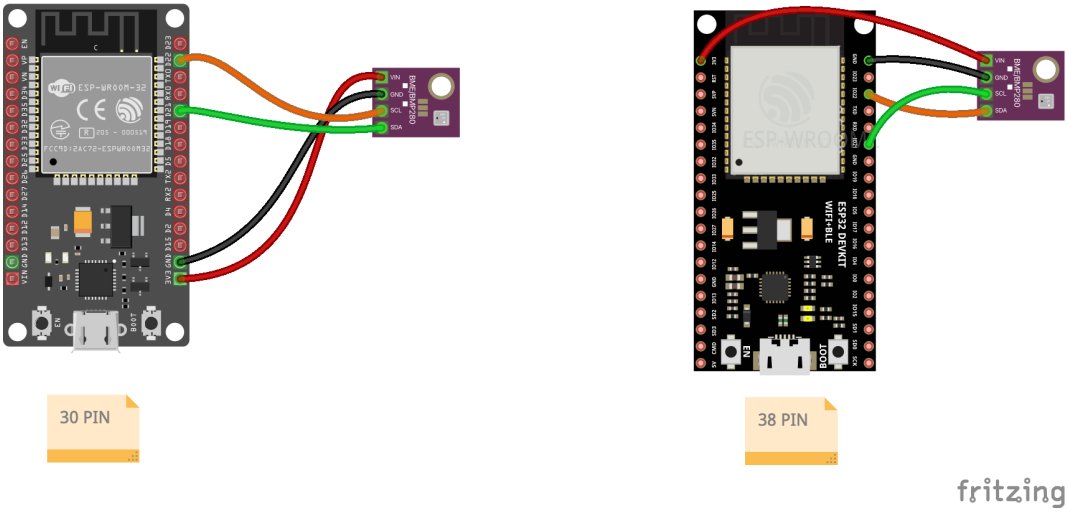
Nama	Datasheet	Nama Arduino IDE Library
BMP - 280	**BST-BMP280-DS001-11.pdf**	Adafruit_BMP280
BMP-180	**BST-BMP180-DS000-09.pdf**	Adafruit_BMP085

Pengalaman saya membaca BMP 280, cukup menguras logika teman-teman, pertama saya sudah tahu alamatnya di 0x76 cuman ketika sudah dipasang, skemanya sudah benar tetap saja di serial monitor disebutkan bahwa sensor saya tidak ditemukan. Sampai beberapa kali mencoba saran dari berbagai forum tetap saja belum bisa terbaca. Setelah sekian lama mencari, ternyata masalahnya adalah library yang saya gunakan adalah tipe BME-280, sementara yang saya miliki adalah BMP-280 dengan 6 pin.

Menurut saya, pertama kali yang harus kita ketahui adalah jenis protokol komunikasi yang akan digunakan, misalnya I2C, SPI atau UART, dari protokol ini kita bisa menerka kira-kira skema rangkaiannya seperti apa. Kemudian layaknya sebuah komunikasi, setiap entitas yang berkomunikasi memiliki sebuah alamat (*address*). Setelah kita ketahui *address*-nya, kemudian kita memilih tools yang tepat untuk menerjemahkan komunikasi kita yaitu sebuah *driver*, nah kumpulan *driver* dan tools yang akan kita gunakan dalam berkomunikasi di Arduino IDE disimpan dalam *Library*.

I2C Scanner

Seringkali sensor yang kita beli BMP-280 dan 180 tidak disertai dengan *datasheet* yang menunjukan di alamat berapa kita dapat mengakses informasi hasil pengukuran. Sehingga ada baiknya kita mengecek alamat sensor kita dengan kode I2C Scanner, yaitu sebuah kode program untuk mencari alamat peripheral yang terhubung dengan I2C ESP32.



ESP32	Senso BMP	Keterangan
GPIO 22	SCL	Serial Clock
GPIO 21	SDA	Serial Data
3V3	VIN	Voltage In
GND	GND	Ground

Rangkai komponen elektronika seperti gambar di atas, jika kita ingin mengecek address sensor BMP-280. Namun jika kita menginginkan untuk mengecek *peripherals* lain, kita dapat mengkoneksikannya seperti diatas dengan catatan GPIO 22 dihubungkan dengan SCL, dan GPIO 21 dihubungkan dengan SDA alat. Langkah selanjutnya *copy-paste* kode berikut dalam *sketch* baru di Arduino IDE, kemudian *compile* dan *upload*.

```
#include <Wire.h>

void setup()
{
  Wire.begin();
  // Wire.begin(0, 2); // ESP8266
  Serial.begin(115200);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}


void loop()
{
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // Scanning alamat dari 1 sampai 126
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
```

```
      Serial.print("0");
     Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(2000);
}
```

Secara rinci penggunaan kode program di atas dapat dilihat di video berikut.

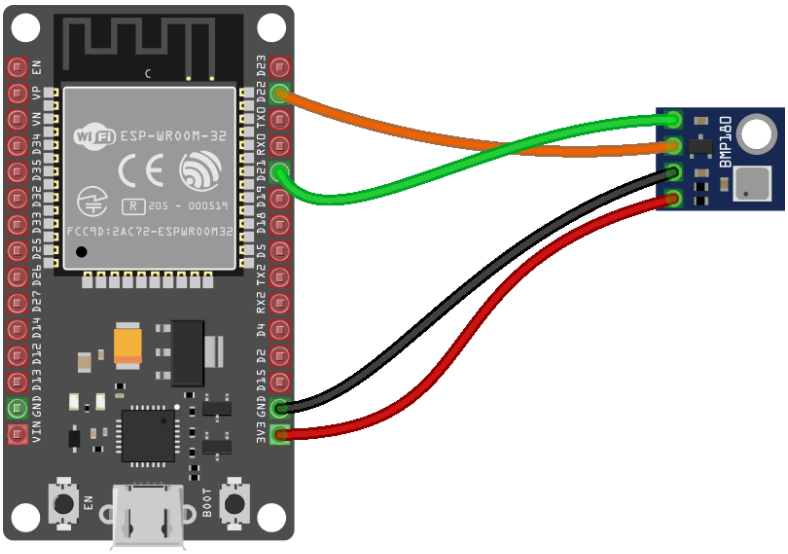
https://www.youtube.com/embed/VEGjV5vhIsk

Membaca Sensor BMP 180

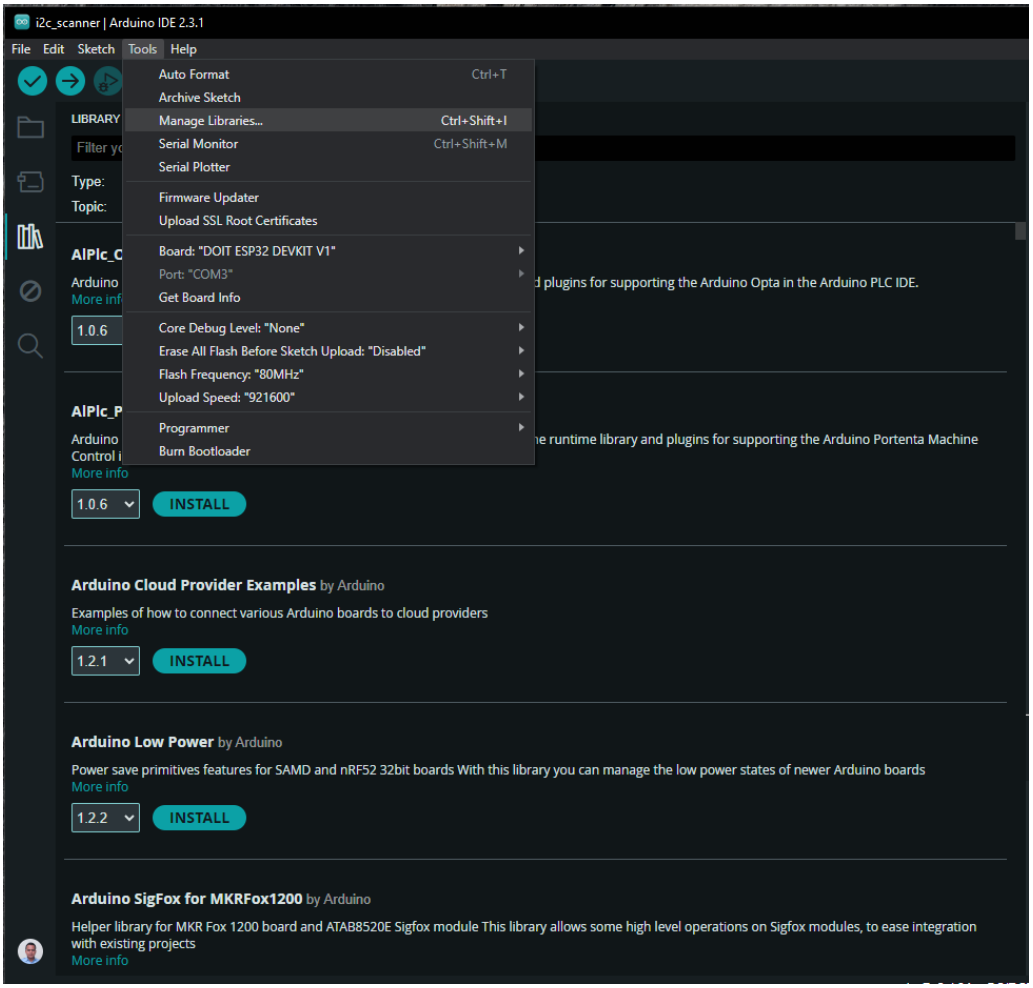
Pada percobaan kali ini kita akan coba membaca sensor BMP 180 dan menampilkan hasil pengukurannya melalui serial monitor. kebutuhan komponen yang akan kita gunakan seperti pada tabel berikut.

No.	Komponen
1	Devboard ESP 32 Devkit V1 30 Pin
2	Sensor BMP 180
3	Micro USB
4	Jumper wires
5	Breadboard

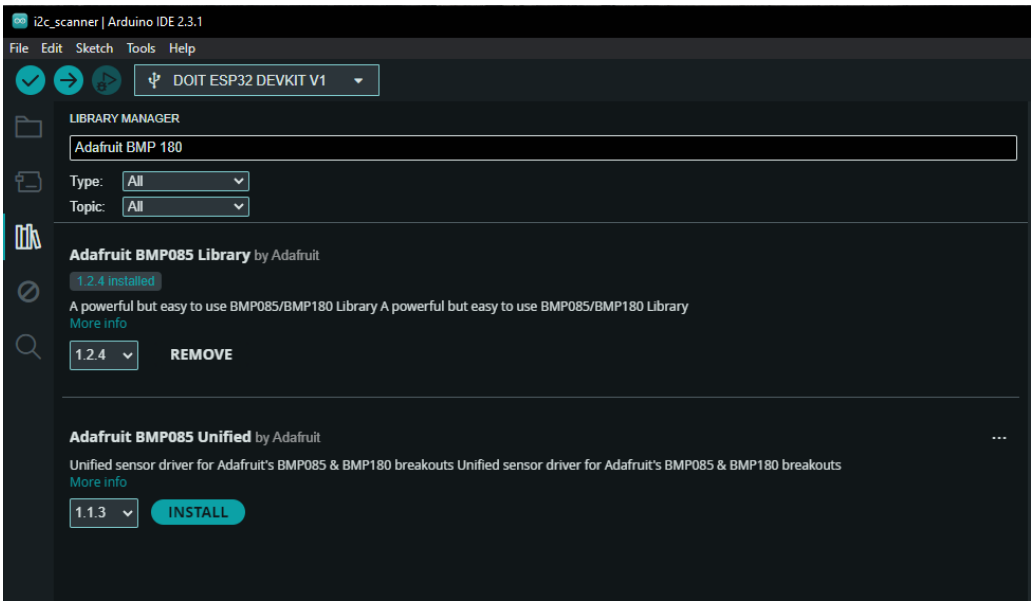
Skema rangkaian sebagai berikut.



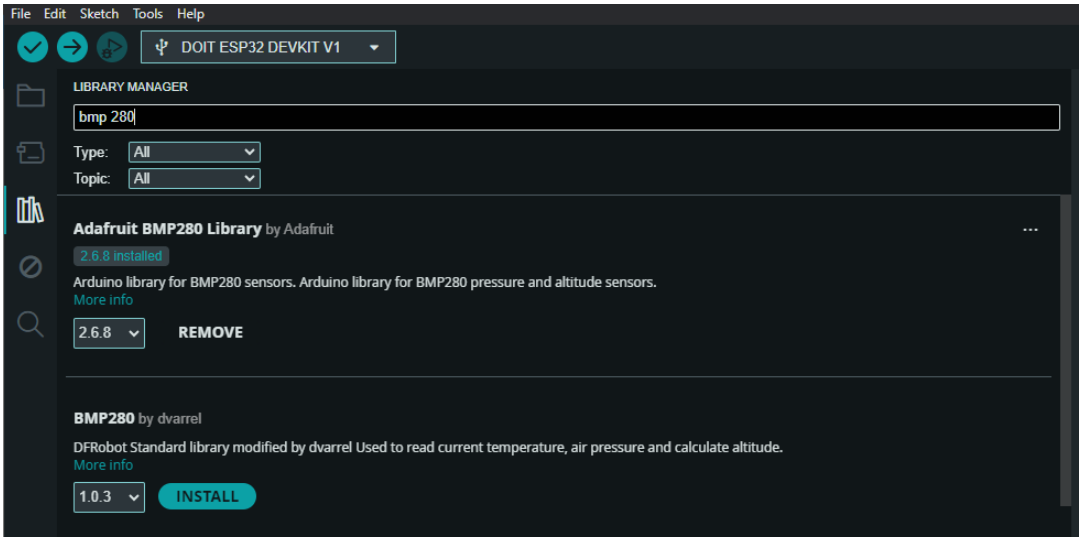
Setelah rangkaian terpasang, kita perlu install library sensor BMP 180 dan BMP 280. Kita perlu menginstall dua duanya karena kita akan ber-eksperimen dengan kedua sensor tersebut di section 2. Langkah-langkah instalasinya adalah sebagai berikut. Pilih **Tools > Manage libraries.**



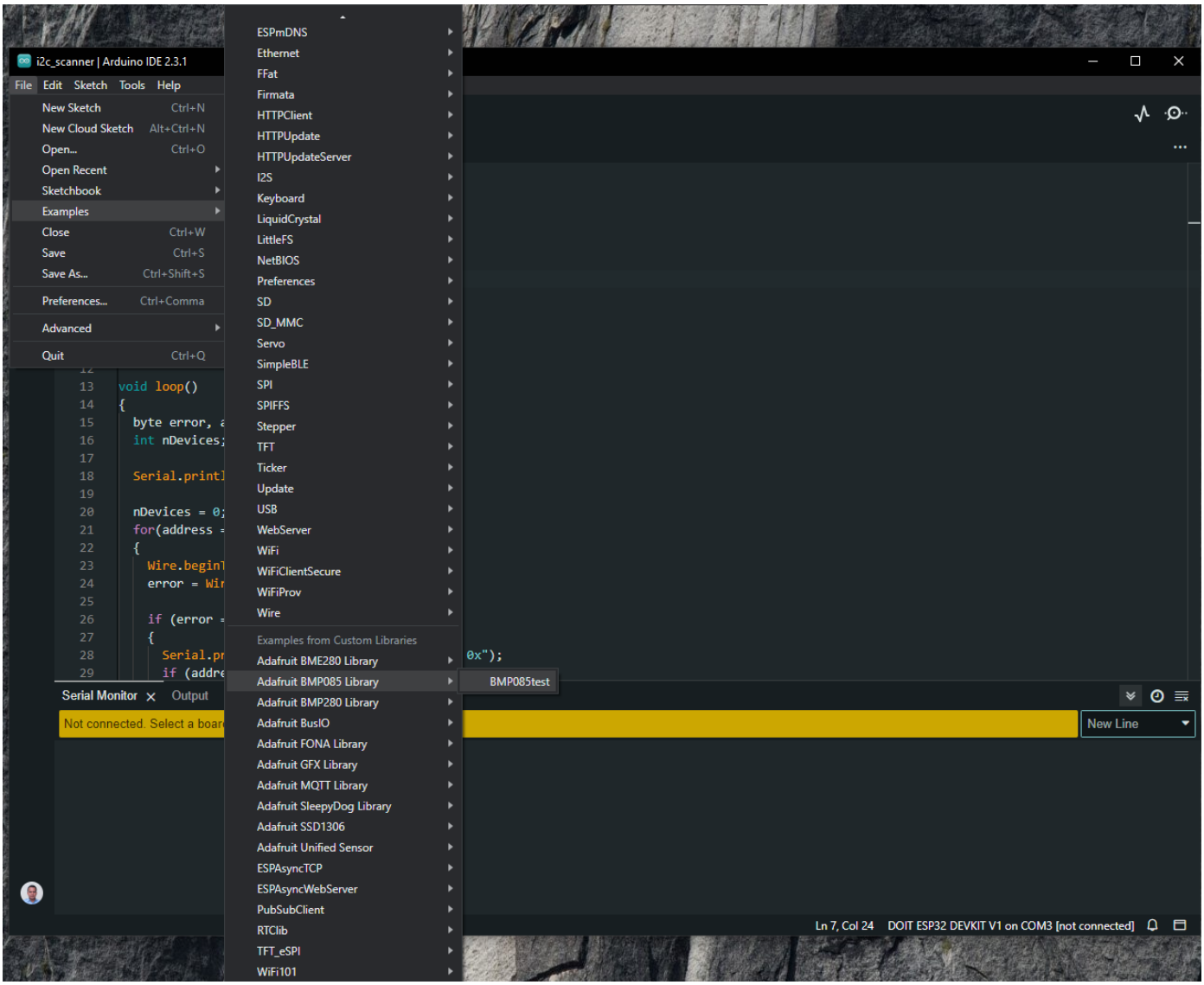
Pada kolom pencarian, ketikan Adafruit BMP 180, kemudian pilih library **Adafruit BMP 085** dari Adafruit dengan menekan tombol install.



Sedangkan untuk sensor BMP 280, pada kolom pencarian ketik Adafruit BMP 280 kemudian pilih dan install pada hasil pencarian **Adafruit BMP 280** by Adafruit.



Setelah menginstall dua library diatas, selanjutnya kita membuka file examples dari library Adafruit sensor yang telah kitan install. Caranya adalah pilih menu **File > Examples > Adafruit BMP085 Library > BMP085Test.**



Atau copy-paste kode berikut, yang diambil dari examples diatas.

```
#include <Adafruit_BMP085.h>

/***************************************************
  This is an example for the BMP085 Barometric Pressure & Temp Sensor

  Designed specifically to work with the Adafruit BMP085 Breakout
  ----> https://www.adafruit.com/products/391

  These pressure and temperature sensors use I2C to communicate, 2 pins
  are required to interface
  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!
```

```
  Written by Limor Fried/Ladyada for Adafruit Industries.
  BSD license, all text above must be included in any redistribution
 ***************************************************/

// Connect VCC of the BMP085 sensor to 3.3V (NOT 5.0V!)
// Connect GND to Ground
// Connect SCL to i2c clock - on '168/'328 Arduino Uno/Duemilanove/etc thats Analog 5
// Connect SDA to i2c data - on '168/'328 Arduino Uno/Duemilanove/etc thats Analog 4
// EOC is not used, it signifies an end of conversion
// XCLR is a reset pin, also not used here

Adafruit_BMP085 bmp;

void setup() {
  Serial.begin(115200);
  if (!bmp.begin()) {
	Serial.println("Could not find a valid BMP085 sensor, check wiring!");
	while (1) {}
  }
}

void loop() {
    Serial.print("Temperature = ");
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");

    Serial.print("Pressure = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");

    // Calculate altitude assuming 'standard' barometric
    // pressure of 1013.25 millibar = 101325 Pascal
    Serial.print("Altitude = ");
    Serial.print(bmp.readAltitude());
    Serial.println(" meters");

    Serial.print("Pressure at sealevel (calculated) = ");
    Serial.print(bmp.readSealevelPressure());
    Serial.println(" Pa");
```

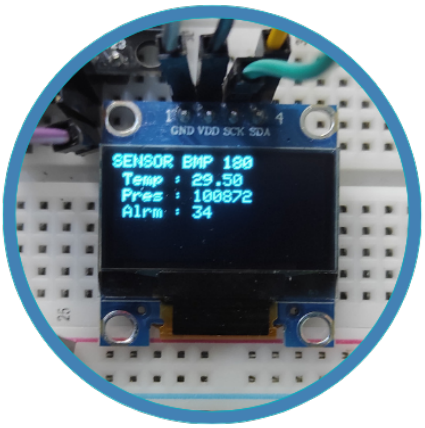
```
  // you can get a more precise measurement of altitude
  // if you know the current sea level pressure which will
  // vary with weather and such. If it is 1015 millibars
  // that is equal to 101500 Pascals.
    Serial.print("Real altitude = ");
    Serial.print(bmp.readAltitude(101500));
    Serial.println(" meters");

    Serial.println();
    delay(500);
}
```

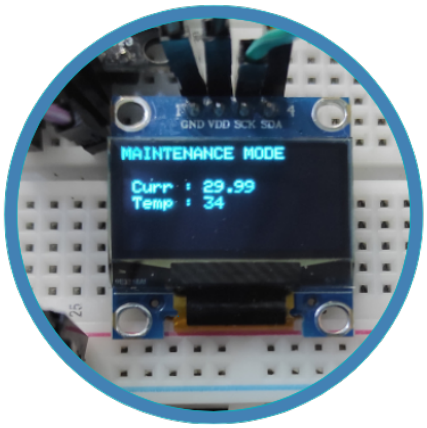
Let's compile !!, lakukan proses uploading dengan memilih icon upload (anak panah ke kanan) dan tunggu hingga proses upload selesai yang ditandai dengan notifikasi **Done uploading.** Hasil dari percobaan ini adalah seperti yang ditampilkan dalam video berikut.

https://www.youtube.com/embed/T3LSGPs8Vlk

Section 2

Pada modifikasi percobaan kali ini, kita akan membuat program kecil berupa alarm suhu. Program ini akan menyalakan buzzer ketika temperatur melebihi batas. *Peripheral* sensor yang akan kita gunakan adalah sensor BMP 280 dan BMP 180. Kedua sensor tersebut akan mengukur suhu secara bersama-sama (*redundant*), program akan mengecek apakah sensor satu dan sensor dua yang lebih dahulu mencapai batas dan jika logika tersebut terpenuhi, program akan menyalakan buzzer. Pengguna dapat melihat hasil pengukuran suhu dan tekanan udara dari layar OLED 128 x 64, pengguna dapat memilih sensor mana yang akan dilihat (BMP 280 atau BMP 180) dengan menekan tombol selektor.

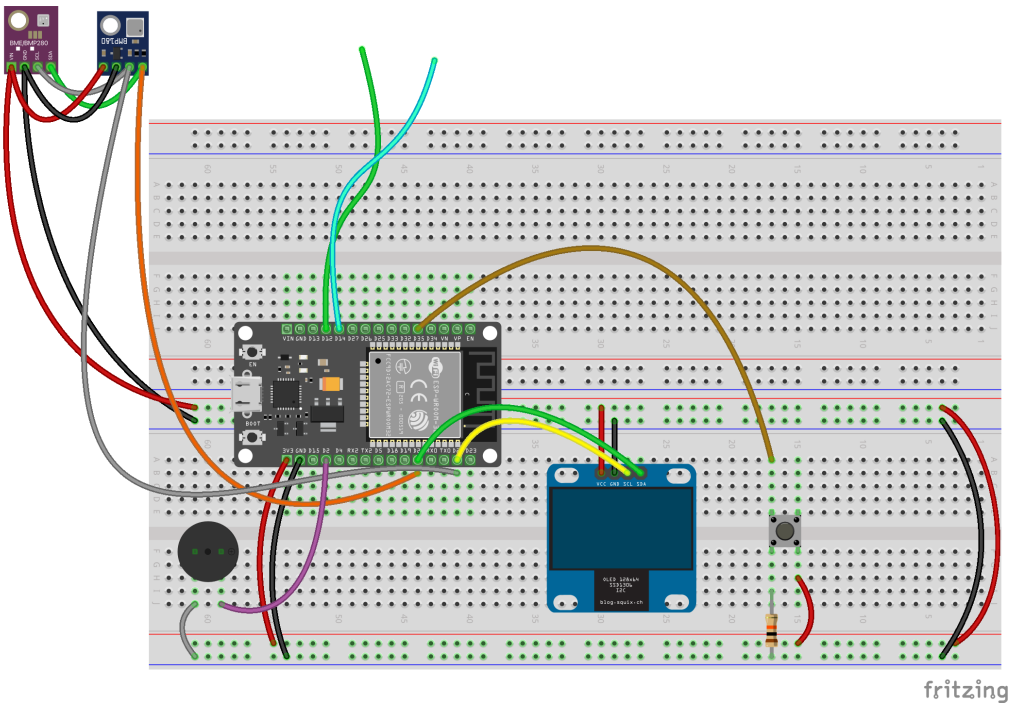




Proses penyetingan batas suhu dilakukan secara langsung dengan menggunakan tombol selektor. Sedangkan untuk menaikan atau menurunkan variabel batas dilakukan dengan menggunakan touch sensor di GPIO 12 dan 14.

No.	Komponen
1	ESP32
2	Jumper wires
3	Breadboard
4	Sensor BMP 280
5	Sensor BMP 180
6	OLED 128 x 64 I2C
7	Buzzer
8	Pushbutton
10	Resistor 10 K

Skema rangkaian seperti di bawah ini, seluruh peripheral sensor dan OLED menggunakan protokol komunikasi I2C. Karena sensor dan OLED memiliki address yang berbeda, maka dapat dirangkai secara paralel sebagai berikut.



Kode program yang akan kita compile adalah seperti di bawah ini. Kita menggunakan empat library yaitu **Adafruit_SSD1306**, **Adafruit_GFX**, **Adafruit_BMP280** dan **Adafruit_BMP085**.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_BMP085.h>
#include <Adafruit_BMP280.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_BMP280 bmp1;
Adafruit_BMP085 bmp2;
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

const int btnPin1 = 35;
const int buzPin = 2;

int btn1State = 0;
int sensor = 1;
int batas = 34;
int batas_set = 34;

void setup() {
  Serial.begin(115200);
  delay(500);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)){
    Serial.println("OLED tidak terkoneksi");
    for(;;);
  }

  if (! bmp1.begin(0x76)) {
    Serial.println("BMP 280 tidak ditemukan");
    while (1) delay(10);
  }

  if (!bmp2.begin(0x77)) {
    Serial.println("BMP 180 tidak ditemukan");
```

```
    while (1) {}
  }

  pinMode(btnPin1, INPUT);
  pinMode(buzPin, OUTPUT);

  // welcome message Smart-x di OLED
  delay(2000);
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(25, 31);
  display.print("SMART-X ITB");
  display.display();
  delay(5000);
  display.clearDisplay();
  delay(500);

}

/* Declare functions */
// Selektor tampilan OLED - 1. Untuk sensor BMP 280, 2. BMP 180 dan 3. Maintennance
void pengukuran(int sensor) {
  if (sensor == 1) {
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print("SENSOR ");
    display.print("BMP 280");
    display.setCursor(0,11);
    display.print(" Temp : ");
    display.print(bmp1.readTemperature());
    display.setCursor(0,21);
    display.print(" Pres : ");
    display.print(bmp1.readPressure());
    display.setCursor(0,31);
    display.print(" Alrm : ");
    display.print(batas);
    display.display();
```

```
  } else if (sensor == 2) {
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print("SENSOR ");
    display.print("BMP 180");
    display.setCursor(0,11);
    display.print(" Temp : ");
    display.print(bmp2.readTemperature());
    display.setCursor(0,21);
    display.print(" Pres : ");
    display.print(bmp2.readPressure());
    display.setCursor(0,31);
    display.print(" Alrm : ");
    display.print(batas);
    display.display();
  } else if (sensor == 3) {
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print("MAINTENANCE MODE ");
    display.setCursor(0,21);
    display.print(" Curr : ");
    display.print(bmp1.readTemperature());
    display.setCursor(0,31);
    display.print(" Temp : ");
    display.print(batas_set);
    display.display();
  }
}

// Fungsi untuk mengeset nilai batas, dentan touch pada GPIO 12 dan 14
void maintenance() {
  if (sensor == 3 && touchRead(12) < 40) {
    -- batas_set;
    beep(1,100);
  } else if (sensor == 3 && touchRead(14) < 40) {
    ++ batas_set;
```

```
    beep(1,100);
  }
}

// Fungsi selektor tombol
void selector() {
  if (btn1State == HIGH) {
    ++ sensor;
    beep(1,100);
    if (sensor > 3) {
      sensor = 1;
      beep(1,1000);
      batas = batas_set;
    }
  }
}

// Fungsi mengecek apakah sensor melebihi batas
void alarm() {
  if (bmp1.readTemperature() >= batas || bmp2.readTemperature() >= batas) {
    beep(5,500);
  }
}

// Fungsi beep generator
void beep (int t, int dl) {
  for (int i = 1; i <= t; ++i) {
    digitalWrite(buzPin, HIGH);
    delay(dl);
    digitalWrite(buzPin, LOW);
    delay(dl);
  }
}

/* Program Utama */
void loop() {
  btn1State = digitalRead(btnPin1);
  delay(100);
  selector();
  pengukuran(sensor);
```

```
  maintenance();
  alarm();
  delay(500);
}
```

Setelah seluruh rangkaian terpasang dan kode program telah diupload dengan Arduino IDE, hasilnya demonya seperti video berikut.

https://www.youtube.com/embed/SAFUQiB8eiY

Revision #44
Created 17 March 2024 02:27:50 by Sandi Wibowo
Updated 19 March 2024 09:32:56 by Sandi Wibowo