

Microcontroller for IoT Application

Module 2: C/C++ Basics for Microcontroller II
Minggu ke-2

Nazmi Febrian
Sr. IoT Engineer - Axiata Digital Labs

Komponen yang dibutuhkan

1. ESP32 Board
2. LED dan 220 Ohm Resistor
3. DHT11
4. Potentiometer
5. Kabel Jumper



Outline

1. Fungsi dan Library di Pemrograman C/C++ pada Mikrokontroler
2. *Arrays, Pointers*, dan *String* pada *Arduino Framework*
3. *Typecasting* dan Operasi *Bitwise* pada Pemrograman Mikrokontroler
4. Analog-to-Digital Converter (ADC)



Fungsi dan Library di Pemrograman C/C++ pada Mikrokontroler



Fungsi adalah **sub-program** yang bisa **digunakan kembali** baik di dalam program itu sendiri, maupun di program yang lain.

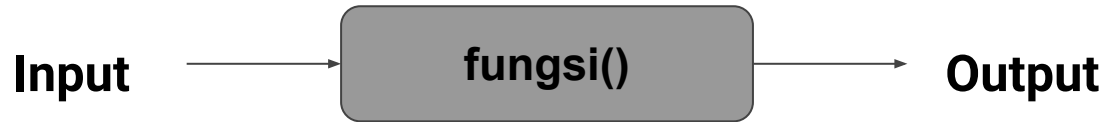
Penggunaan Fungsi

- Mengurangi penumpukan program di main()
- Membungkus program menjadi bagian-bagian kecil
- Memudahkan pembacaan program



Apa itu Fungsi?

Fungsi **adalah** sub-program yang bisa digunakan kembali baik di dalam program itu sendiri, maupun di program yang lain



Struktur Fungsi

tipe data
nilai kembalian

tipe data parameter

```
int nama_fungsi(int parameter) {  
    // tubuh fungsi berisi  
    // kode program dari fungsi  
}
```

petanikode.com

- Ada dua cara: **definisi** dan **deklarasi**
- Gunakan tipe *void* untuk fungsi yang tidak ada nilai balikan (output)
- Variabel lokal dan global?

Let's try a function in Microcontroller!

1. Gunakan rangkaian sesuai dengan tugas terakhir yang diberikan
2. Buat fungsi berikut pada baris terakhir code di main.cpp

```
11 void initLed() {  
12 |   pinMode(BUILTIN_LED, OUTPUT);  
13 | }  
14  
15 void ledOn() {  
16 |   digitalWrite(BUILTIN_LED, HIGH);  
17 | }  
18  
19 void ledOff() {  
20 |   digitalWrite(BUILTIN_LED, LOW);  
21 | }
```


Let's try a function in Microcontroller!

3. Definiskan fungsi yang di buat di bagian atas sebelum fungsi 'void setup()'

```
1  #include <Arduino.h>
2
3  void initLed();
4  void ledOn();
5  void ledOff();
```

4. Panggil fungsi tersebut di dalam 'void setup()' dan 'void loop()'

```
7  void setup() {
8      // put your setup code here, to run once:
9      initLed();
10 }
11
12 void loop() {
13     // put your main code here, to run repeatedly:
14     ledOn();
15     delay(1000);
16     ledOff();
17     delay(1000);
18 }
```

5. Lalu amati Built-in LED

Let's try a function in Microcontroller!

1. Ubah fungsi sebelumnya menjadi seperti berikut. Deklarasikan fungsi yang dibuat di bagian atas sebelum fungsi 'void setup()'

```
1  #include <Arduino.h>
2
3  #define LED1 BUILTIN_LED
4  #define LED2 4
5
6  void initLed(int ledNumber) {
7      pinMode(ledNumber, OUTPUT);
8  }
9
10 void ledOn(int ledNumber) {
11     digitalWrite(ledNumber, HIGH);
12 }
13
14 void ledOff(int ledNumber) {
15     digitalWrite(ledNumber, LOW);
16 }
17
```

Let's try a function in Microcontroller!

2. Panggil fungsi tersebut di dalam 'void loop()' dan 'void setup()' dengan inputan bilangan integer

```
18 void setup() {  
19     // put your setup code here, to run once:  
20     initLed(LED1);  
21     initLed(LED2);  
22 }  
23  
24 void loop() {  
25     // put your main code here, to run repeatedly:  
26     ledOn(LED1);  
27     ledOff(LED2);  
28     delay(1000);  
29     ledOff(LED1);  
30     ledOn(LED2);  
31     delay(1000);  
32 }
```

3. Amati LED-LED yang menyala



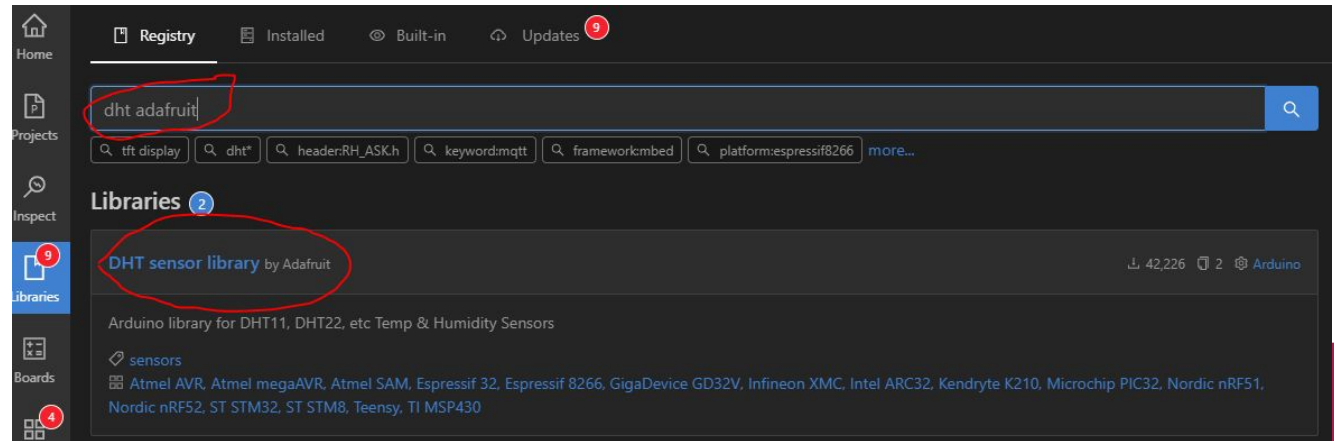
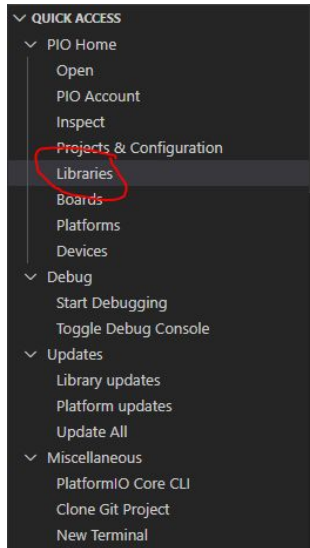
Library pada Framework Arduino

- Library adalah sekumpulan fungsi yang mempermudah pengguna membuat program tertentu.
- Library pada Arduino mempersingkat waktu dalam membuat program untuk *hardware* tertentu



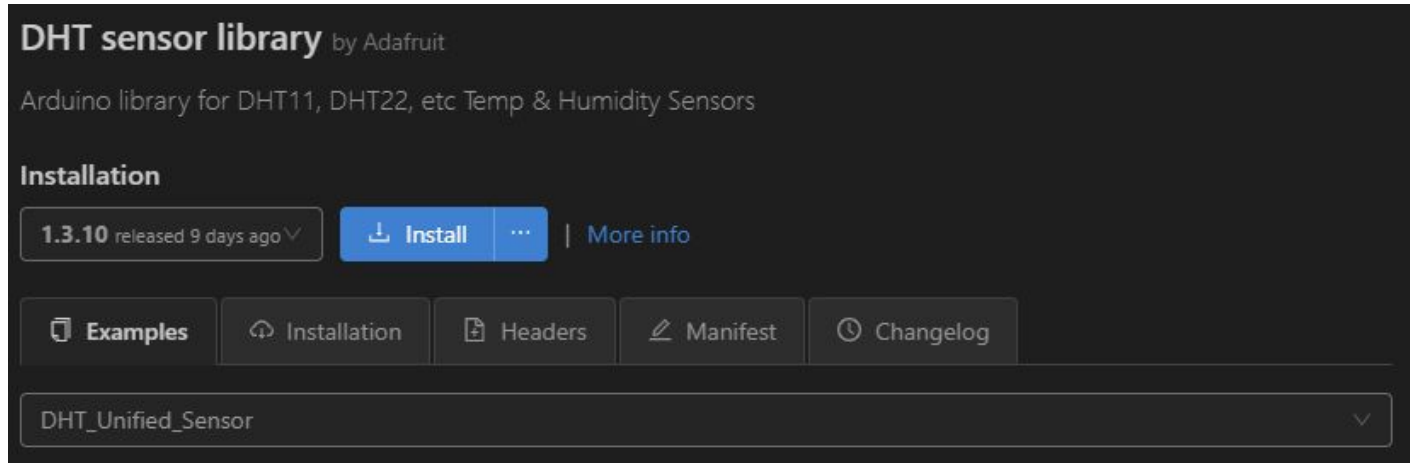
Download Library pada PlatformIO

1. Masuk ke menu 'Libraries' di PlatformIO dan ketik library yang ingin dicari. Dalam hal ini library sensor suhu 'dht adafruit'.



Download Library pada PlatformIO

2. Buka dan klik 'install'



Sensor Suhu DHT11

Rentang pengukuran suhu



Rentang pengukuran kelembaban

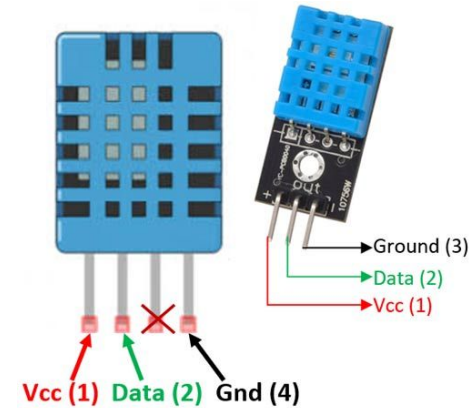
Resolusi

Tegangan input

Suplai arus

Periode sampling

| DHT11 | DHT22 |
|---|---|
|  |  |
| 0 to 50 °C ± 2 °C | -40 to 80 °C ± 0.5 °C |
| 20 to 90% ± 5 % | 0 to 100% ± 2 % |
| Humidity: 1% Temperature: 1°C | Humidity: 0.1% Temperature: 0.1°C |
| 3 – 5.5 V DC | 3 – 6 V DC |
| 0.5 – 2.5 mA | 1 – 1.5 mA |
| 1 second | 2 seconds |

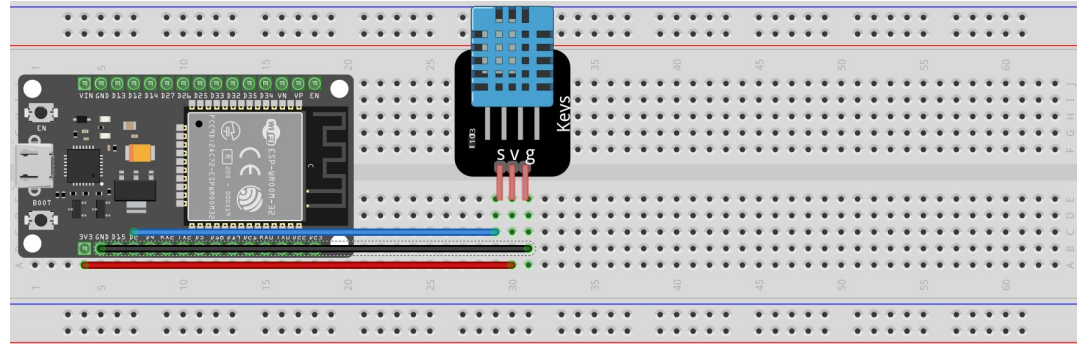


Source: component101

Library Hands-on

1. Buat rangkaian sesuai gambar.
2. Include library pada bagian awal file dan definisikan pin serta tipe DHT yang akan digunakan

```
1  #include <Arduino.h>
2  #include <Wire.h>
3  #include <SPI.h>
4  #include <DHT.h>
5
6  #define DHT_PIN 5
7  #define DHT_TYPE DHT11
```



Library Hands-on

3. Buat fungsi untuk pembacaan nilai suhu DHT11 seperti berikut.

```
27 float readTemp() {  
28     return dht.readTemperature();  
29 }
```

4. Deklarasi variabel yang dibutuhkan pada program serta fungsi yang sudah dibuat.

```
9 DHT dht(DHT_PIN, DHT_TYPE);  
10  
11 float readTemp();
```

Library Hands-on

5. Panggil fungsi tersebut pada 'void loop()'

```
19 void loop() {  
20     // put your main code here, to run repeatedly:  
21     float temperature = readTemp();  
22     Serial.println(temperature);  
23     delay(5000);  
24 }
```

6. Amati keluaran sensor di terminal.

Library Hands-on

7. Ubah posisi variabel local suhu ke variabel global dan ubah fungsi tanpa pengembalian.

8. Amati keluaran sensor di terminal.

```
9   DHT dht(DHT_PIN, DHT_TYPE);
10  float temperature = 0;
11      void readTemp()
12  void readTemp();
13
14  void setup() {
15      // put your setup code here, to run once:
16      Serial.begin(9600);
17      dht.begin();
18  }
19
20  void loop() {
21      // put your main code here, to run repeatedly:
22      readTemp();
23      Serial.println(temperature);
24      delay(5000);
25  }
26
27  void readTemp() {
28      temperature = dht.readTemperature();
29  }
```

Arrays, Pointers, dan String pada Arduino Framework

You'll learn about the relationship between arrays and pointers in C/C++ programming, the beauty of 'String' in Arduino, and variable Casting process in C/C++.

```
void loop() {  
  if (WiFi.status() != WL_CONNECTED) {  
    setupWifi();  
  }  
  
  if (WiFi.status() == WL_CONNECTED && !ESPmqt.connected()) {  
    setupMQTT();  
  }  
}
```

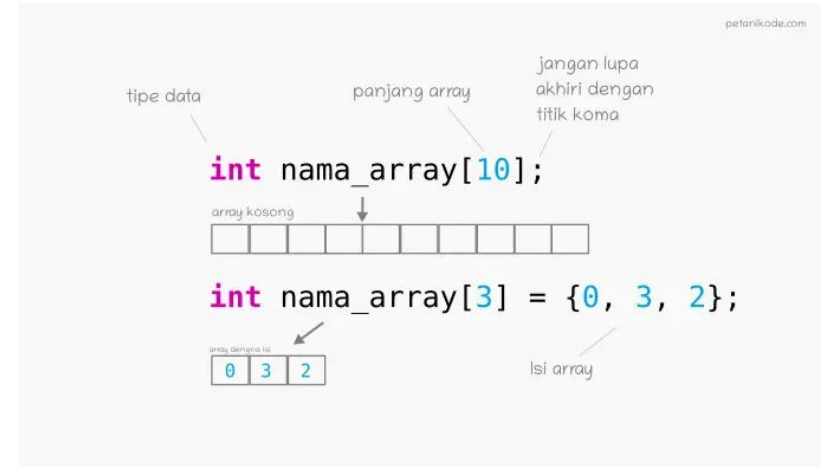
Array di C/C++

```
int suhu_1 = 0;  
int suhu_2 = 0;  
int suhu_3 = 0;  
int suhu_4 = 0;  
int suhu_5 = 0;
```

→

```
int suhu[5] = {0};
```

- digunakan untuk menyimpan **sekumpulan** data dalam **satu tempat**.
- setiap data dalam Array memiliki indeks yang dimulai dari angka 0
- Deklarasi array harus disertai panjang array



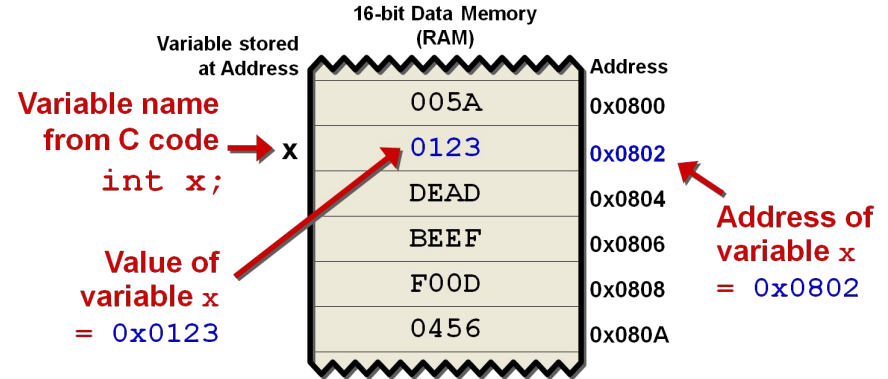
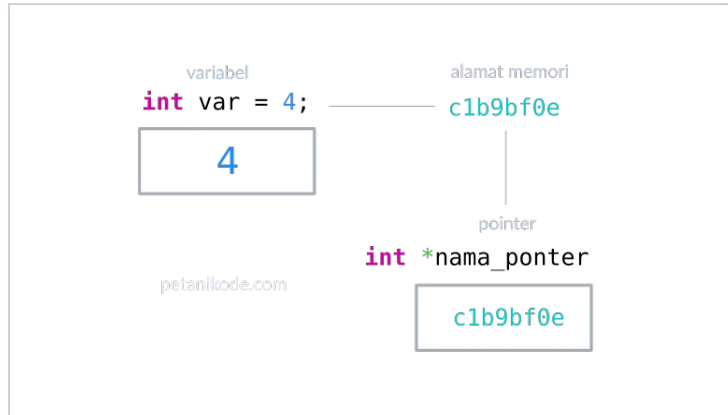
Let's try an Array implementation in uC

- Jalankan program berikut pada ESP32 board
- Amati keluaran UART

```
1  #include <Arduino.h>
2
3  int temp[3];
4  int tempLen;
5  int tempAverage;
6
7  void setup() {
8      // put your setup code here, to run once:
9      Serial.begin(9600);
10     tempLen = sizeof(temp) / sizeof(temp[0]);
11 }
12
13 void loop() {
14     // put your main code here, to run repeatedly:
15     for (int x = 0; x < tempLen; x++) {
16         temp[x] = random(20,30);
17     }
18
19     tempAverage = (temp[0] + temp[1] + temp[2]) / tempLen;
20     Serial.println(tempAverage);
21     delay(5000);
22 }
```

Pointer di C/C++

- Setiap variabel pada program memiliki alamat memori.
- Alamat memori berfungsi untuk menentukan lokasi penyimpanan data pada memori (RAM).
- Alamat memori direpresentasikan dalam bilangan heksadesimal.



Source: [Microchip Developer](#)

Pointer Hands-On

- Jalankan program di samping
- Perhatikan nilai pointer yang ditampilkan
- Perhatikan cara pemanggilan isi pointer

```
1  #include <Arduino.h>
2
3  int temp[3];
4  int tempLen, tempAverage;
5  int *pointerTemp1, *pointerTemp2, *pointerTemp3;
6
7  void setup() {
8      // put your setup code here, to run once:
9      Serial.begin(9600);
10     delay(1000);
11     tempLen = sizeof(temp) / sizeof(temp[0]);
12     pointerTemp1 = &temp[0];
13     pointerTemp2 = &temp[1];
14     pointerTemp3 = &temp[2];
15
16     Serial.print("Alamat data pertama: ");
17     Serial.println((unsigned int)pointerTemp1);
18     Serial.print("Alamat data kedua: ");
19     Serial.println((unsigned int)pointerTemp2);
20     Serial.print("Alamat data ketiga: ");
21     Serial.println((unsigned int)pointerTemp3);
22 }
23
24 void loop() {
25     // put your main code here, to run repeatedly:
26     for (int x = 0; x < tempLen; x++) {
27         temp[x] = random(20,30);
28     }
29
30     tempAverage = (*pointerTemp1 + *pointerTemp2 + *pointerTemp3) / tempLen;
31     Serial.println(tempAverage);
32     delay(5000);
33 }
```


Tipe String pada *Arduino Framework*

- Declaration
 - `String(val)`
 - `String(val, base)`
 - `String(val, decimalPlaces)`
 - Variabel dengan tipe integer dan long integer
 - an integer or long integer variable, using a specified base
 - a float or double, using a specified decimal places
- Beberapa contoh deklarasi String dari tipe data lain:
 - Array of char dengan *double quote*
 - Constant char dengan *single quote*
 - Variabel String lainnya
 - Constant integer atau long integer
 - Constant integer, dengan basis integer

```
String stringOne = "Hello String";
String stringOne = String('a');
String stringTwo = String("This is a string");
String stringOne = String(stringTwo + " with more");
String stringOne = String(13);
String stringOne = String(analogRead(0), DEC);
String stringOne = String(45, HEX);
String stringOne = String(255, BIN);
String stringOne = String(millis(), DEC);
String stringOne = String(5.698, 3);
```

String Hands-on

1. Berdasarkan code dari DHT11, buat fungsi berikut untuk baca suhu dan kelembaban

```
33 void updateData() {  
34     temp = dht.readTemperature();  
35     humid = dht.readHumidity();  
36 }
```

2. Definisikan variable String di awal program

```
11 String printData;  
12 String tempSentence = "Suhu saat ini: ";  
13 String humidSentence = "Kelembaban saat ini: ";
```

String Hands-on

3. Buat code berikut pada 'void loop()'

```
23 void loop() {  
24     // put your main code here, to run repeatedly:  
25     updateData();  
26     printData = tempSentence + String(temp) + String(" C");  
27     Serial.println(printData);  
28     printData = humidSentence + String(humid) + String(" %");  
29     Serial.println(printData);  
30     delay(5000);  
31 }
```

4. Perhatikan serial monitor hasil keluaran ESP32



Typecasting dan Operasi *bitwise* pada Pemrograman Mikrokontroler



The beauty of a high-level language like C is that **you don't need to understand** the details of the microcontroller architecture.

Type Casting pada *Arduino Framework*

- Biasa disebut dengan *type conversion* atau pengubahan tipe satu ke tipe lainnya
- Format yang biasa digunakan:

```
(type_name) expression
```



Operator di C/C++

Operators in C

Unary operator



Binary operator



Ternary operator



| Operator | Type |
|-----------------------|---------------------------------|
| +, -, ++, -- | Unary operator |
| +, -, *, /, % | Arithmetic operator |
| <, <=, >, >=, ==, != | Relational operator |
| &&, , ! | Logical operator |
| &, , <<, >>, ~, ^ | Bitwise operator |
| =, +=, -=, *=, /=, %= | Assignment operator |
| ?: | Ternary or conditional operator |

Bit, Byte, Heksadesimal dan Desimal

- 1 byte = 8 bit = 2 karakter heksadesimal
- 1 byte range:
 - 0 - 255 desimal
 - 00000000 - 11111111 binary
 - 00 - FF heksadesimal
- Desimal - basis 10
- Heksadesimal - basis 16
- Biner - basis 2

| DECIMAL | HEX | BINARY |
|---------|-----|--------|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| 10 | A | 1010 |
| 11 | B | 1011 |
| 12 | C | 1100 |
| 13 | D | 1101 |
| 14 | E | 1110 |
| 15 | F | 1111 |

Contoh Operasi Biner

| | | |
|-----|-----------------|------|
| 177 | 1 0 1 1 0 0 0 1 | |
| 195 | 1 1 0 0 0 0 1 1 | OR / |
| 243 | 1 1 1 1 0 0 1 1 | |

| | | |
|-----|-----------------|---------|
| 177 | 1 0 1 1 0 0 0 1 | |
| 195 | 1 1 0 0 0 0 1 1 | AND / & |
| 129 | 1 0 0 0 0 0 0 1 | |

| | | |
|-----|-----------------|------|
| 179 | 1 0 1 1 0 0 1 1 | >> 4 |
| 11 | 0 0 0 0 1 0 1 1 | |

| | | |
|-----|-----------------|------|
| 179 | 1 0 1 1 0 0 1 1 | << 6 |
| 192 | 1 1 0 0 0 0 0 0 | |

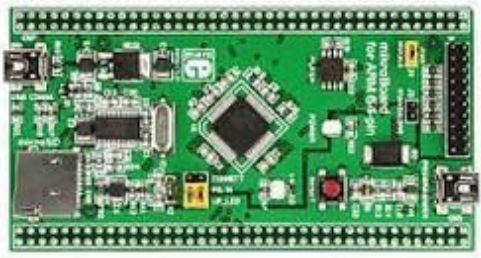
Hands-on Bitwise operator

1. Misal sensor mengirim data suhu via uart sepanjang 16-bit.
2. 16-bit data tersebut harus dibagi 100 agar menjadi data suhu.
3. Kita terjemahkan data tersebut untuk dikirim hasilnya via uart.
4. Jalankan program di sampling
5. Coba kirim dua karakter dengan HTerm
6. Bandingkan nilai karakter yang dikirim dengan balikan mikrokontroler

```
1  #include <Arduino.h>
2
3  byte sensorData[2] = "";
4  byte counter = 0;
5  unsigned short tempShort;
6  float tempFloat;
7
8  void setup() {
9      // put your setup code here, to run once:
10     Serial.begin(9600);
11 }
12
13 void loop() {
14     // put your main code here, to run repeatedly:
15     if (Serial.available()) {
16         sensorData[counter] = Serial.read();
17         counter++;
18         if (counter > 1) {
19             counter = 0;
20             tempShort = (unsigned short)sensorData[0] & 0xFF;
21             tempShort = tempShort | ((unsigned short)(sensorData[1] << 8) & 0xFF);
22             tempFloat = (float)tempShort / 10;
23
24             String data = "Nilai suhu sensor: " + String(tempFloat);
25             Serial.println(data);
26         }
27     }
28 }
```

Microcontroller Peripherals

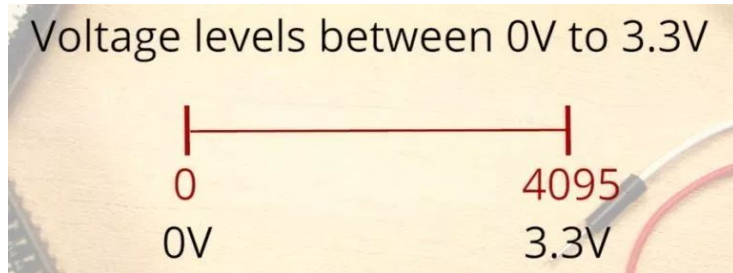
Analog to Digital Converter
(ADC)



ADC berfungsi mengubah nilai analog tegangan menjadi bilangan digital pada rentang tertentu

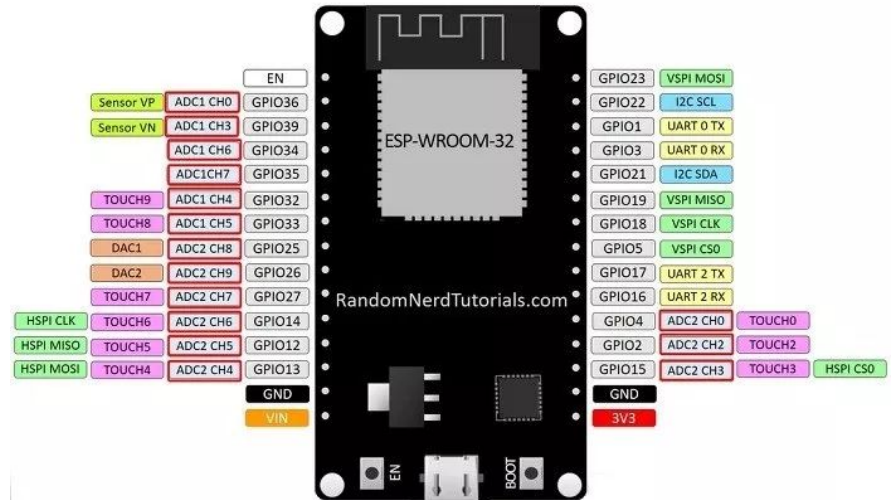
Analog-to-Digital Converter (ADC)

- Melakukan pengukuran tegangan 0 - 3.3V ke dalam bilangan integer
- Resolusi: 12 bit -> 0 - 4095



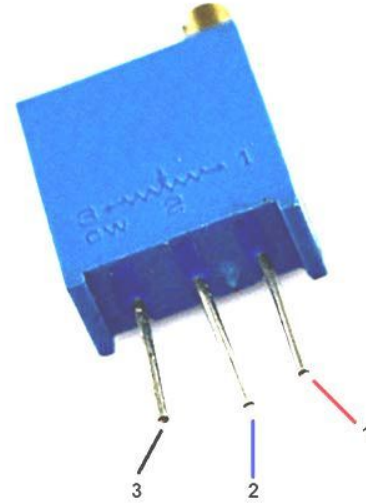
Source: randomnerdstutorial

ESP32 DEVKIT V1 - DOIT



Potentiometer

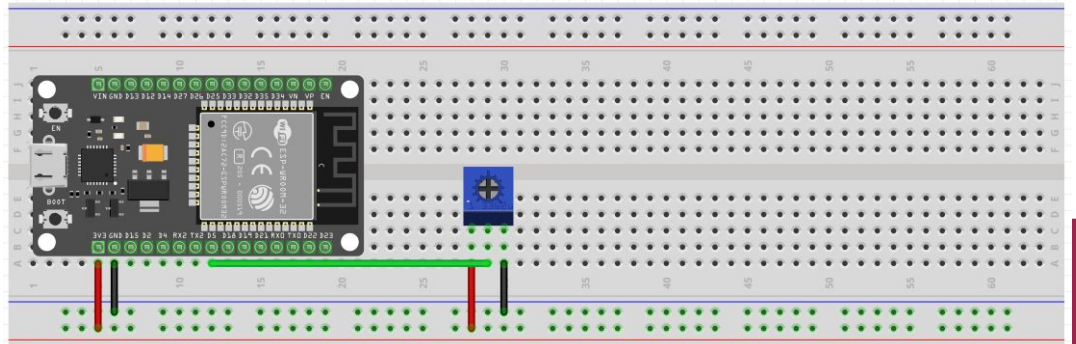
1. PIN1 Potentiometer ke 3.3V ESP32
2. PIN2 Potentiometer ke PIN D5 ESP32
3. PIN3 Potentiometer ke GND ESP32



ADC Hands-On

1. Jalankan program di bawah ini di ESP32.
2. Ubah-ubah potentiometer menggunakan obeng atau apapun
3. Perhatikan hasil bacaan ADC di serial monitor

```
1 #include <Arduino.h>
2
3 #define POT_PIN 4
4
5 int adcValue = 0;
6
7 void setup() {
8   // put your setup code here, to run once:
9   Serial.begin(9600);
10 }
11
12 void loop() {
13   // put your main code here, to run repeatedly:
14   adcValue = analogRead(POT_PIN);
15   Serial.println(adcValue);
16   delay(5000);
17 }
```



Ada Pertanyaan?

