

## MODUL 2

### PENGENALAN ANTARMUKA MIKROKONTROLER IO, ADC, PWM ARDUINO UNO

#### 1. TUJUAN

- 1.1 Praktikan dapat memahami tentang definisi IO, ADC, dan PWM
- 1.2 Praktikan dapat mengetahui fungsi analog read dan analog write.
- 1.3 Praktikan dapat mempraktikkan Digital Read and Digital Write di Arduino IDE
- 1.4 Praktikan dapat mempraktikkan pengontrolan kecerahan LED dengan PWM di Arduino IDE

#### 2. ALAT DAN BAHAN PRAKTIKUM

- Laptop
- ESP32
- Arduino UNO
- Software Arduino IDE dan Thonny
- LED
- Kabel Jumper
- Resistor
- Project Board
- Potensiometer

#### 3. DASAR TEORI

##### 3.1 I/O (Input Output)

###### 1. DigitalRead()

DigitalRead() pada Arduino atau ESP32 digunakan untuk membaca status logika dari pin digital tertentu. Fungsi ini memeriksa apakah pada pin tersebut terdapat tegangan tinggi (HIGH), yang biasanya setara dengan 3.3V atau 5V, atau tegangan rendah (LOW), yang setara dengan 0V. Hasil pembacaan ini akan mengembalikan nilai HIGH (1) jika pin berada dalam kondisi tegangan tinggi, atau LOW (0) jika pin berada dalam kondisi tegangan rendah. digitalRead (pin) merupakan bentuk umum dari digitalRead().

###### 2. DigitalWrite()

DigitalWrite() berfungsi untuk memberi nilai High (1) atau Low (0) ke pin digital. Jika pin telah di konfigurasi sebagai Output dengan pinMode(), tegangan akan diatur ke nilai yang sesuai 5V atau 3.3V untuk High , 0V untuk Low atau ground, digitalWrite (pin, value) merupakan bentuk umum dari digitalWrite().

###### 3. AnalogWrite()

AnalogWrite() berfungsi untuk menuliskan suatu nilai yang berupa angka pada sebuah komponen, misalnya LED , kita bisa mengatur seberapa ternag LED tersebut

menyala, bergantung terhadap nilai yang kita tuliskan, analogWrite (pin, value) merupakan bentuk umum dari analogWrite()

4. analogRead()

AnalogRead() berfungsi untuk membaca sinyal pada suatu pin, di analogRead akan menghasilkan nilai dari 0 hingga 1023, yang mempresentasikan voltase 0V hingga 5V, analogRead (pin) merupakan bentuk umum dari analogRead().

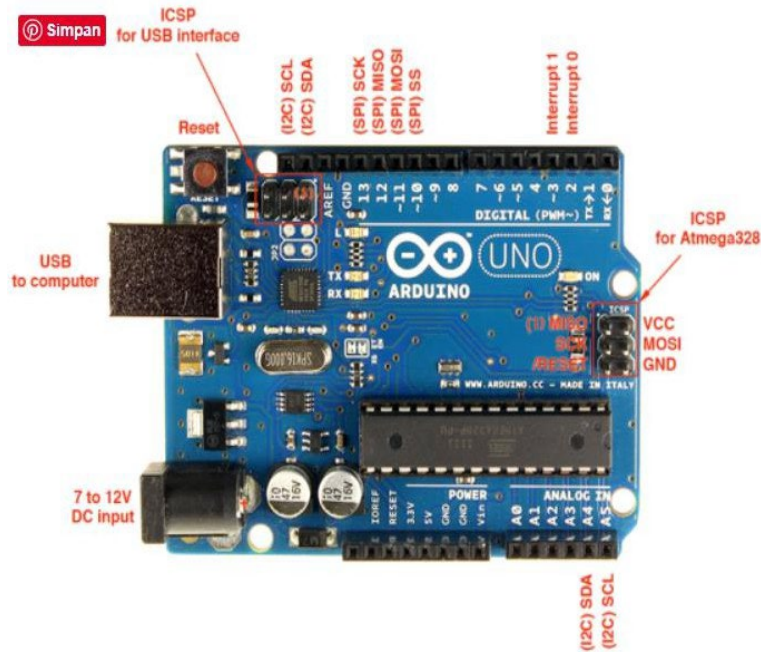
3.2 ADC ( Analog to Digital Converter)

ADC ( Analog to Digital Converter ) berfungsi untuk mengubah sinyal masukan analog menjadi sinyal masukan digital, ADC dapat digunakan untuk berkomunikasi antara mikrokontroller dengan perangkat eksternal (sensor) yang memiliki gelombang sinyal analog (sinus).

3.3 PWM ( Pulse Width Modulation)

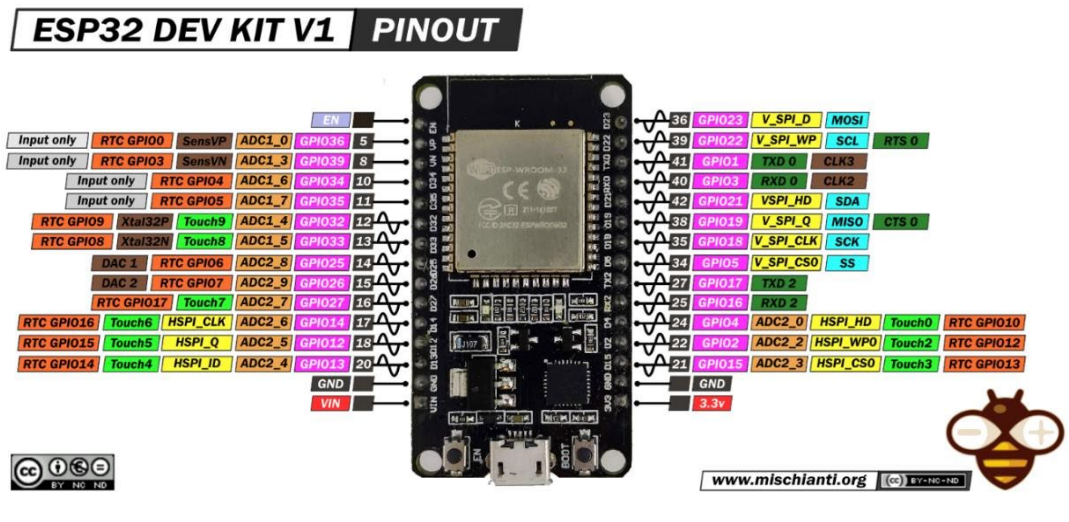
PWM (Pulse Width Modulation) berfungsi untuk mendapatkan bentuk sinyal analog dari sinyal digital sehingga dapat membuat transisi antar state hidup dan mati menjadi lebih halus. PWM banyak digunakan pada berbagai perangkat, seperti motor DC untuk mengatur kecepatan, lampu LED untuk mengontrol tingkat kecerahan, dan kipas komputer untuk menyesuaikan kecepatan putaran sesuai kebutuhan pendinginan. Selain itu, teknologi PWM juga digunakan dalam perangkat audio untuk menghasilkan suara yang lebih halus melalui speaker.

3.4 PIN IO, ADC, dan PWM pada Arduino UNO



Kategori	Jumlah Pin	Pin yang Digunakan
Digital I/O	14 pin (D0 - D13)	D0 - D13
PWM (Pulse Width Modulation)	6 pin	D3, D5, D6, D9, D10, D11
Analog Input (ADC)	6 pin (A0 - A5)	A0 - A5
Pin I <sup>2</sup> C	2 pin	A4 (SDA), A5 (SCL)
Pin SPI	4 pin	D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK)
Power Pin	6 pin	3.3V, 5V, GND (2), VIN, AREF
Interrupt Pin	2 pin	D2 (Interrupt 0), D3 (Interrupt 1)

### 3.5 PIN IO, ADC, dan PWM pada ESP32

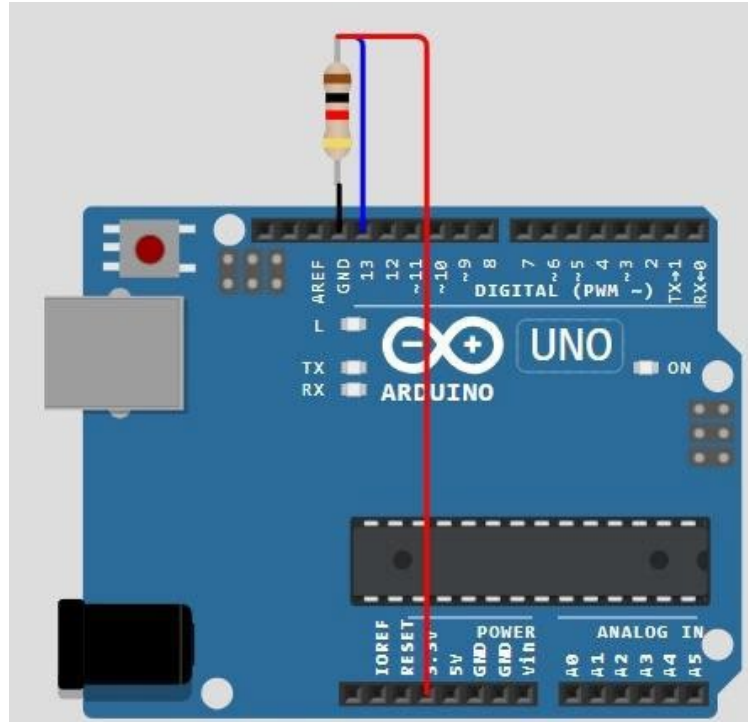


Kategori	Jumlah Pin	Pin yang Digunakan
Digital I/O	34 pin GPIO	GPIO0 hingga GPIO39 (kecuali GPIO20 dan GPIO24)
Pin ADC (Analog to Digital Converter)	18 pin	GPIO0, GPIO2, GPIO4, GPIO12 hingga GPIO15, GPIO32 hingga GPIO39
Pin DAC (Digital to Analog Converter)	2 pin	GPIO25 (DAC1), GPIO26 (DAC2)
Pin PWM (Pulse Width Modulation)	Hampir semua GPIO pin	GPIO2 hingga GPIO39 mendukung PWM
Pin Touch	10 pin	GPIO4, GPIO0, GPIO2, GPIO15, GPIO13, GPIO12, GPIO14, GPIO27, GPIO33, GPIO32
Pin SPI (Serial Peripheral Interface)	6 pin	GPIO18 (SCK), GPIO19 (MISO), GPIO23 (MOSI), GPIO5 (SS)
Pin I <sup>2</sup> C	2 pin	GPIO21 (SDA), GPIO22 (SCL)
Pin UART (Serial Communication)	3 pin	GPIO1 (TX0), GPIO3 (RX0), GPIO16 (TX2), GPIO17 (RX2)
Pin RTC (Real Time Clock)	16 pin	GPIO0 hingga GPIO17 (berbeda fungsi tergantung konfigurasi RTC)
Power Pin	3 pin	VIN, 3.3V, GND

## 4. PERCOBAAN

### 4.1 Arduino IDE Digital Read and Digital Write pada Arduino UNO

1. Siapkan alat dan bahan
2. Buka Arduino IDE
3. Rangkailah ESP32 dan komponennya menjadi seperti berikut:



5. Ketikkan Code berikut lalu upload

```
const int pin13 = 13;
const int pin2 = 2;

void setup() {
  Serial.begin(115200);
  pinMode(pin2, OUTPUT);
  pinMode(pin13, INPUT);
}

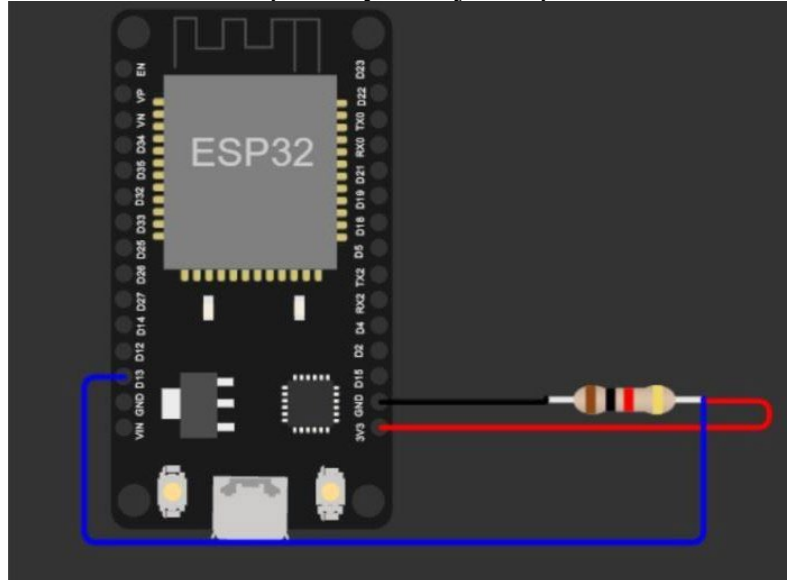
void loop() {
  delay(1000);

  if (digitalRead(pin13) == HIGH) {
    digitalWrite(pin2, HIGH);
    Serial.println(digitalRead(pin2));
  }
  else {
    digitalWrite(pin2, LOW);
    Serial.println(digitalRead(pin2));
  }
}
```

5. Lihatlah pada LED IO 2 di Arduino UNO. Jika menyala cobalah untuk melepas kabel merah VCC agar mati.
6. Lihat hasil pada Shell / Serial Monitor.

## 4.2 Implementasi Digital Input dan Output pada ESP32 Menggunakan MicroPython

1. Siapkan alat dan bahan
2. Buka Thonny IDE
3. Pastikan sudah Configure Interpreter
4. Rangkailah ESP32 dan komponennya menjadi seperti berikut:



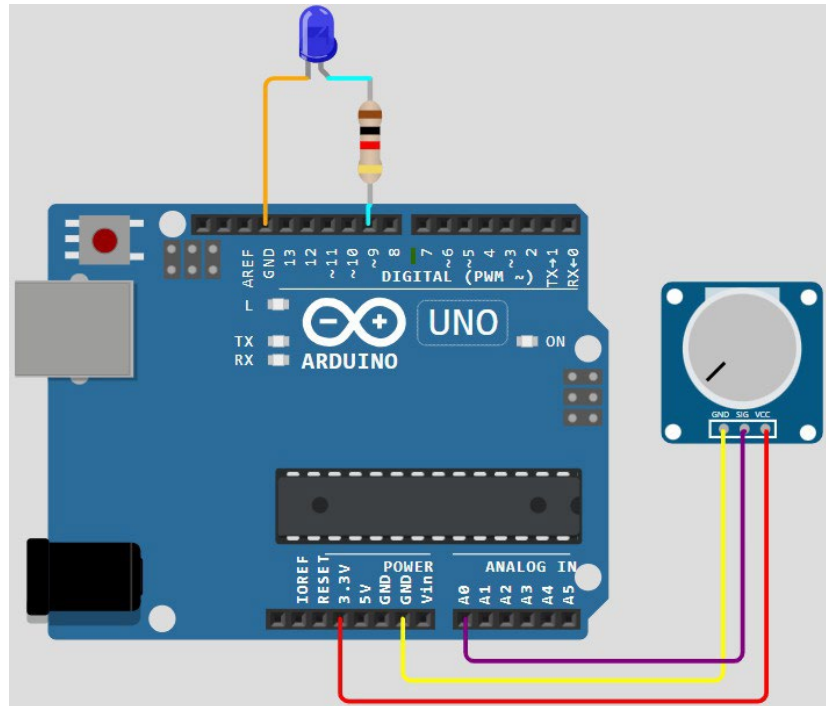
6. Ketikkan Kode berikut dan simpan di main.py

```
1 from machine import Pin
2 import utime
3
4
5 led = Pin(2, Pin.OUT)
6 button = Pin(13, Pin.IN)
7
8 while True:
9     utime.sleep(1)
10
11     if button.value() == 1:
12         led.value(1)
13         print(led.value())
14     else:
15         led.value(0)
16         print(led.value())
```

6. Lihatlah pada LED IO 2 di ESP32. Jika menyala cobalah untuk melepas kabel merah VCC agar mati.
7. Lihat hasil pada Shell / Serial Monitor

## 4.3 Kontrol Kecerahan LED dengan PWM di Arduino IDE

1. Buka Arduino IDE
2. Rangkailah Arduino UNO dan komponennya menjadi seperti berikut:



3. Buat pemrograman seperti di bawah ini:

```
const int analogInPin = A0;
const int analogOutPin = 9;

int sensorValue = 0;
int outputValue = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    sensorValue = analogRead(analogInPin);

    outputValue = map(sensorValue, 0, 1023, 0, 255);

    analogWrite(analogOutPin, outputValue);

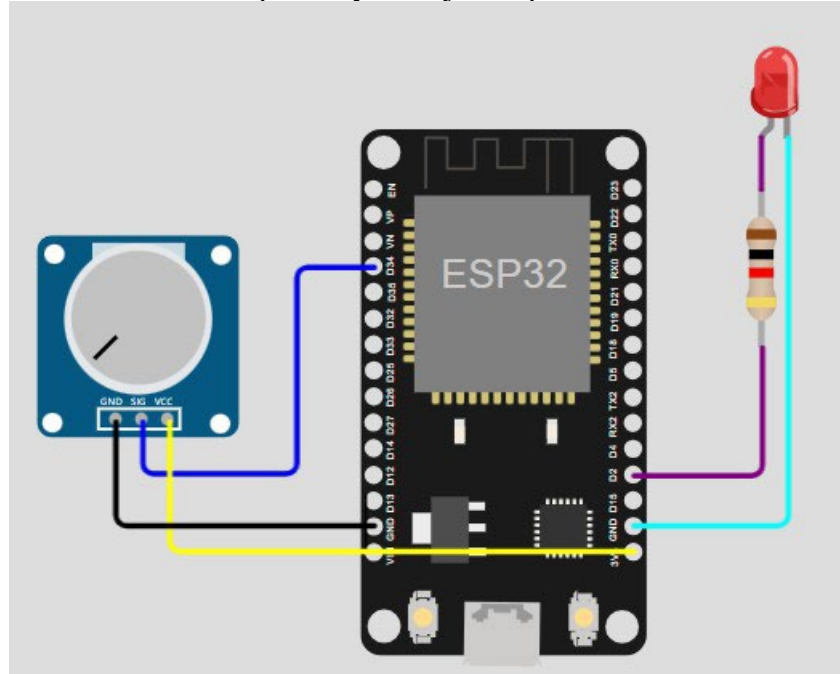
    Serial.print("Sensor (ADC) = ");
    Serial.print(sensorValue);
    Serial.print("\t Output (PWM) = ");
    Serial.println(outputValue);

    delay(2);
}
```

1. Lalu, compile dan upload program tersebut. Tunggu hingga proses upload program selesai hingga bertuliskan “Done Uploading”.

### 1.4 Kontrol Kecerahan LED dengan PWM di ESP32

1. Siapkan alat dan bahan
2. Buka Thonny IDE
3. Pastikan sudah Configure Interpreter
4. Rangkailah ESP32 dan komponennya menjadi seperti berikut:



2. Ketikkan Kode berikut

```

1 from machine import Pin, ADC, PWM
2 import time
3
4 analog_in_pin = ADC(Pin(34))
5 analog_in_pin.atten(ADC.ATTN_11DB)
6
7 analog_out_pin = PWM(Pin(2))
8 analog_out_pin.freq(500)
9
10
11 while True:
12     sensor_value = analog_in_pin.read()
13
14     output_value = int(sensor_value / 4095 * 255)
15
16     analog_out_pin.duty(output_value)
17
18     print("Sensor (ADC) =", sensor_value, "\tOutput (PWM) =", output_value)
19
20
21     time.sleep(0.1)

```

### 5. KESIMPULAN

Mikrokontroler berfungsi sebagai pengendali utama yang dapat membaca input, memproses data, dan menghasilkan output sesuai program. Arduino IDE digunakan untuk memprogram Arduino Uno dengan bahasa C/C++ dan Thonny digunakan untuk memprogram ESP32 dengan MicroPython. Kedua mikrokontroler dapat mengolah input, mengendalikan output digital,

serta menggunakan PWM untuk mengatur kecerahan LED, sehingga mendukung pemahaman dasar dalam pengembangan sistem berbasis mikrokontroler dan IoT.

## **6. LATIHAN**

1. Jelaskan fungsi utama dari perintah `digitalRead()` dan `digitalWrite()` pada mikrokontroler!
2. Apa perbedaan utama antara `analogRead()` dan `analogWrite()`?
3. Apa fungsi dari ADC (Analog to Digital Converter) pada mikrokontroler?
4. Sebutkan salah satu contoh penerapan PWM pada Arduino Uno atau ESP32 dan jelaskan fungsinya!