

MODUL 2

PRAKTIKUM MENGGUNAKAN ESP32 BOARD

2.1 Praktik-1: Mengakses GPIO pada ESP32

A. Tujuan Percobaan

Dapat mengaktifkan LED *built-in* ESP32

B. Alat dan Bahan

1. ESP32
2. Laptop/komputer
3. Kabel USB *micro* B
4. Arduino IDE

C. Langkah Percobaan

1. Pemrograman ESP32 untuk mengaktifkan LED *built-in*

- a. Jalankan Arduino IDE pada laptop/komputer.
- b. Pada ESP32, LED *built-in* terdapat pada GPIO 2. Untuk mengaksesnya perlu dilakukan inisialisasi pin yang akan digunakan dengan cara menambahkan kode di bawah ini.

```
int LED_BUILTIN = 2;
```

- c. Dalam mengkonfigurasi GPIO tertentu agar dijadikan sebagai *input* atau *output*, digunakan fungsi *pinMode()*. Untuk mengaktifkan LED *built-in*, perlu dilakukan inisialisasi LED_BUILTIN (GPIO 2) sebagai *output* dengan cara menambahkan kode di bawah ini di dalam *void setup()*.

```
pinMode(LED_BUILTIN, OUTPUT);
```

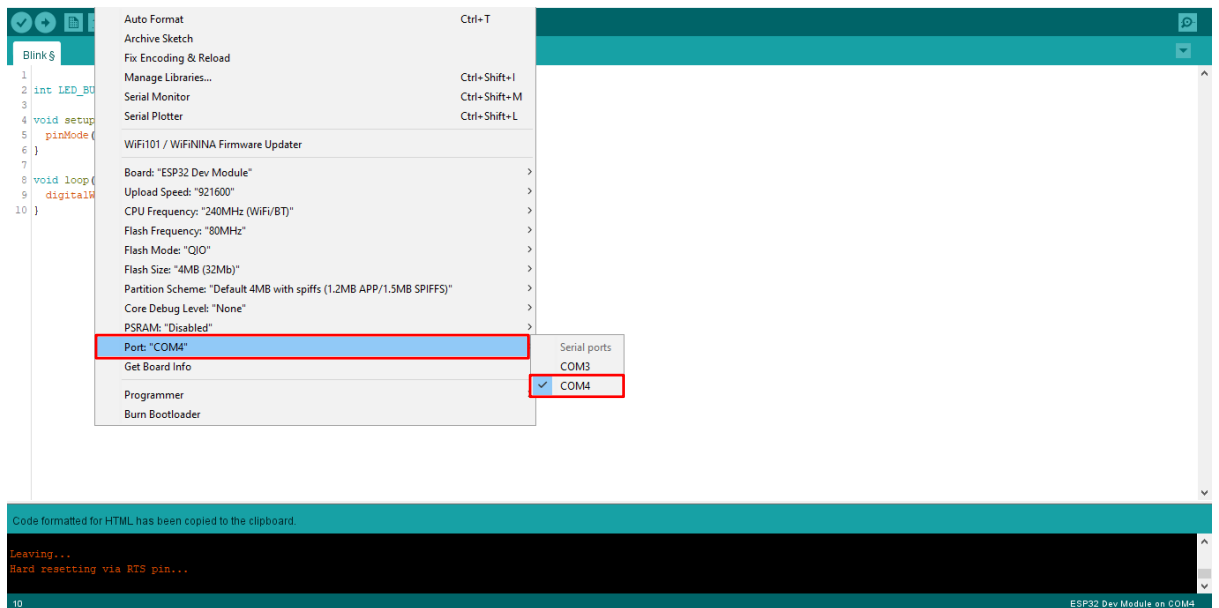
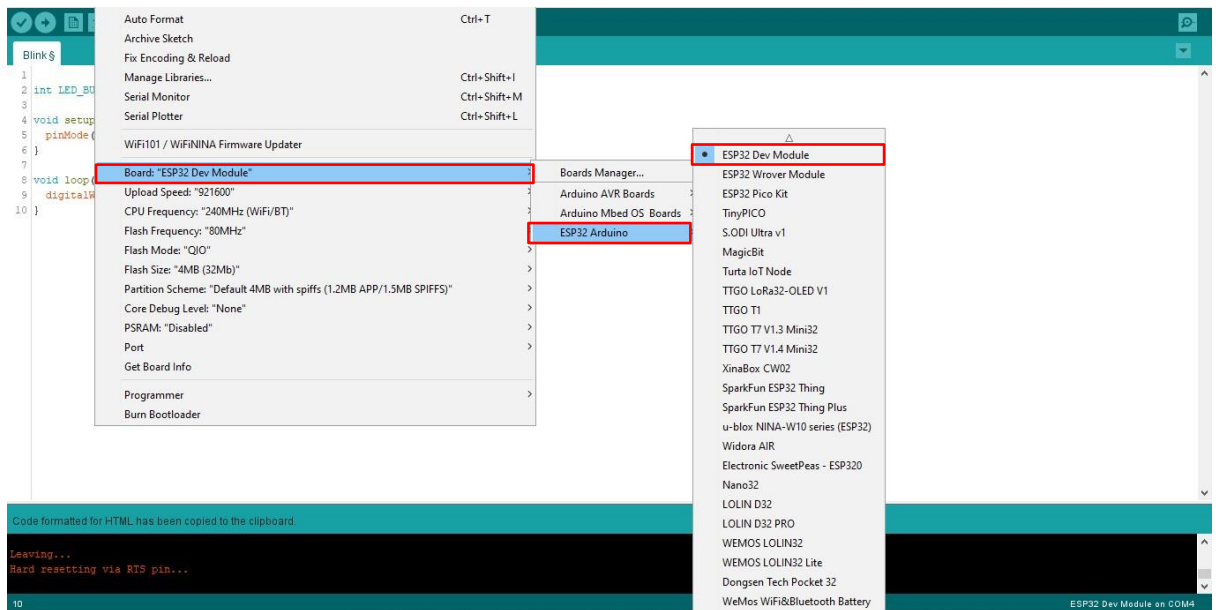
- d. Untuk memberi kondisi HIGH atau LOW pada GPIO, digunakan fungsi *digitalWrite()*. Dalam hal ini, LED_BUILTIN (GPIO 2) diaktifkan dengan cara menambahkan kode di bawah ini ke dalam *void loop()*.

```
digitalWrite(LED_BUILTIN, HIGH);
```

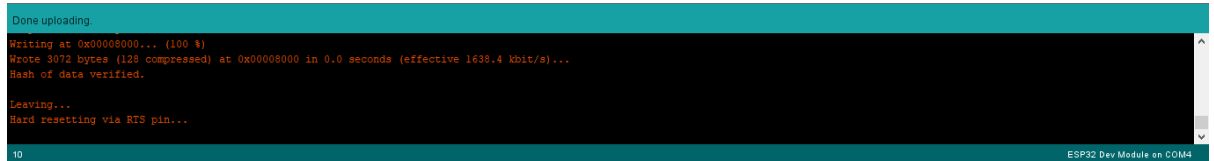
Sehingga kode lengkapnya adalah sebagai berikut.

```
int LED_BUILTIN = 2;
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
}
```

- e. Koneksikan ESP32 dengan laptop/komputer menggunakan kabel USB.
- f. Pastikan *Board* dan *Port* yang dipilih pada bagian Tools Arduino IDE sesuai.



- g. Klik ikon *Verify* (Ctrl + R) pada Arduino IDE untuk mengecek adanya error dan meng-*compile* kode.
- h. Jika tidak terdeteksi *error*, klik ikon *Upload* (Ctrl + U) untuk mengunggah kode ke dalam board ESP32. Setelah proses pengunggahan selesai, konsol Arduino IDE akan menampilkan pesan seperti gambar di bawah ini dan LED *built-in* ESP32 akan aktif (menyala).



```
Done uploading
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1638.4 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
10 ESP32 Dev Module on COM4
```

2.2 Praktik-2: *Blinking* LED *built-in* pada ESP32

A. Tujuan Percobaan

Blinking (mengedipkan) LED *built-in* ESP32.

B. Alat dan bahan

1. ESP32.
2. Laptop/komputer.
3. Kabel USB *micro* B.
4. Arduino IDE.

C. Langkah Percobaan

1. Pemrograman ESP32 untuk membuat LED *built-in* berkedip

- a. Jalankan Arduino IDE pada laptop/komputer.
- b. Pada ESP32, LED *built-in* terdapat pada GPIO 2. Untuk mengaksesnya perlu dilakukan inisialisasi pin yang akan digunakan dengan cara menambahkan kode di bawah ini.

```
int LED_BUILTIN = 2;
```

- c. Dalam mengkonfigurasi GPIO tertentu agar dijadikan sebagai *input* atau *output*, digunakan fungsi *pinMode()*. Untuk mengaktifkan LED *built-in*, perlu dilakukan inisialisasi LED_BUILTIN (GPIO 2) sebagai *output* dengan cara menambahkan kode di bawah ini di dalam *void setup()*.

```
pinMode(LED_BUILTIN, OUTPUT);
```

- d. Untuk mengatur kondisi HIGH atau LOW pada GPIO, digunakan fungsi *digitalWrite()*. Untuk membuat LED_BUILTIN (GPIO 2) berkedip, LED_BUILTIN (GPIO 2) diatur pada kondisi HIGH selama 1 detik, kemudian diatur pada kondisi LOW selama 1 detik, dan seperti itu seterusnya. Untuk memberikan waktu tunda sebelum mengeksekusi kode program selanjutnya, digunakan fungsi *delay()*. Fungsi *delay()* pada Arduino IDE memiliki satuan *millisecond* (ms), sehingga untuk memberikan jeda 1 detik dilakukan dengan cara menambahkan kode *delay(1000);*.

Selanjutnya, tambahkan kode di bawah ini ke dalam *void loop()* untuk membuat kondisi LED *built-in* berkedip tiap 1 detik.

```
digitalWrite(LED_BUILTIN, HIGH);  
delay(1000);  
digitalWrite(LED_BUILTIN, LOW);  
delay(1000);
```

Sehingga kode lengkapnya adalah sebagai berikut.

```
int LED_BUILTIN = 2;  
  
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

- e. Koneksikan ESP32 dengan laptop/komputer menggunakan kabel USB.
- f. Pastikan *Board* dan *Port* yang dipilih pada bagian Tools Arduino IDE sesuai.
- g. Klik ikon *Verify* (Ctrl + R) pada Arduino IDE untuk mengecek adanya error dan meng-*compile* kode.
- h. Jika tidak terdeteksi *error*, klik ikon *Upload* (Ctrl + U) untuk mengunggah kode ke dalam board ESP32. Setelah proses pengunggahan selesai, LED *built-in* ESP32 akan berkedip tiap 1 detik sekali.

2.3 Praktik-3: Menggunakan *Serial Monitor* pada Arduino IDE

A. Tujuan Percobaan

Dapat menampilkan informasi pada *serial monitor* Arduino IDE

B. Alat dan Bahan

1. ESP32.
2. Laptop/Komputer.
3. Kabel USB *micro B*.
4. Arduino IDE.

C. Langkah Percobaan

Pemrograman ESP32 untuk menampilkan informasi dari komunikasi serial

- a. Jalankan Arduino IDE pada Laptop/Komputer.
- b. Buatlah variabel berisi data yang akan ditampilkan pada *serial monitor* dengan menambahkan kode dibawah ini. Variabel dapat dibuat dengan berbagai tipe data, dalam hal ini tipe data yang digunakan adalah tipe data *String* dengan nama variabel *Himbauan*.

```
String Himbauan = "Selalu Patuhi Protokol Kesehatan";
```

- c. Dalam *void setup()* tambahkan kode di bawah ini untuk melakukan inisialisasi *baudrate*.

```
Serial.begin(9600);
```

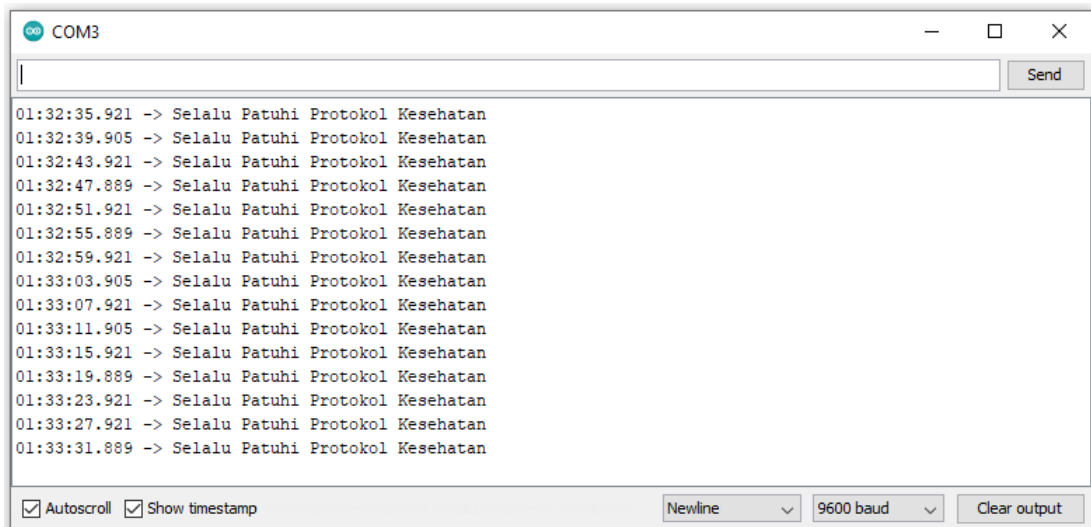
- d. Tambahkan kode dibawah ini dalam *void loop()* untuk mencetak variabel yang telah dibuat sebelumnya pada *serial monitor*. Variabel akan ditercetak setiap 4 detik.

```
Serial.println(Himbauan);  
delay(4000);
```

Sehingga kode lengkapnya adalah sebagai berikut.

```
String teks = "Belajar Internet of Things dengan ESP32";  
  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() { Serial.println(teks);  
    delay(4000);  
}
```

- e. Koneksikan ESP32 dengan Laptop/Komputer menggunakan kabel USB.
- f. Pastikan Board dan Port yang dipilih pada bagian Tools Arduino IDE sesuai.
- g. Klik ikon Verify (Ctrl+R) pada Arduino IDE untuk mengecek adanya error dan meng-*compile* kode.
- h. Jika tidak terdeteksi error, klik ikon Upload (Ctrl + U) untuk mengunggah kode ke dalam board ESP32. Setelah proses pengunggahan selesai, *serial monitor* akan menampilkan pesan seperti gambar di bawah ini.



2.4 Praktik-4: Menghubungkan ESP32 ke Jaringan Internet

A. Tujuan Percobaan

Dapat menghubungkan ESP32 ke jaringan *internet*.

B. Alat dan Bahan

1. ESP32.
2. Jaringan *Internet*.
3. Laptop/Komputer.
4. Kabel USB *micro B*.
5. Arduino IDE.

C. Langkah Percobaan

Pemrograman ESP32 untuk menghubungkan ke jaringan *internet*

- a. Jalankan Arduino IDE pada laptop/komputer.
- b. Untuk memasukan *library* yang akan digunakan tambahkan kode dibawah ini pada awal program. *Library* Wi-Fi digunakan untuk menghubungkan ESP32 dengan jaringan *internet*.

```
#include <WiFi.h>
```

- c. Inisialisasikan *network credentials* dengan menambahkan kode di bawah ini.

```
const char* ssid = "YourNetworkName";  
const char* password = "YourNetworkpassword";
```

Ubah *ssid* dan *password* sesuai dengan jaringan *internet* yang digunakan.

- d. Dalam *void setup()* tambahkan kode di bawah ini karena perlu dilakukan inisialisasi *baudrate*, dan menghubungkan ESP32 dengan Wi-Fi menggunakan `WiFi.begin(ssid, password);`. Fungsi ini digunakan untuk memulai koneksi dengan Wi-Fi. Saat proses menghubungkan ESP32 dengan jaringan *internet* berlangsung serial monitor akan mencetak "." setiap 500 mS (*millisecond*), dan akan mencetak "WiFi connected." dan "IP address" ketika ESP32 berhasil terhubung dengan jaringan *internet*.

```
Serial.begin(115200);  
Serial.print("Connecting to ");  
Serial.println(ssid);  
WiFi.begin(ssid, password);  
  
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}
```



```

// Print local IP address and start web server
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

```

Sehingga kode lengkapnya adalah sebagai berikut.

```

#include <WiFi.h>

const char* ssid = "Redmi Note 8";
const char* password = "12345678";

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

  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
}

```

- e. Koneksikan ESP32 dengan laptop/komputer menggunakan kabel USB.
- f. Pastikan Board dan Port yang dipilih pada bagian Tools Arduino IDE sesuai.
- g. Klik ikon Verify (Ctrl + R) pada Arduino IDE untuk mengecek adanya error dan meng-compile kode.
- h. Jika tidak terdeteksi error, klik ikon Upload (Ctrl + U) untuk mengunggah kode ke dalam board ESP32.
- i. Buka *serial monitor* Arduino IDE kemudian tekan EN pada ESP32 untuk memastikan bahwa ESP32 telah terkoneksi dengan network. Jika berhasil terkoneksi, *serial monitor* akan menampilkan pesan seperti gambar di bawah ini.

```
COM3
09:34:58.560 -> rst:0x10 (RTCWDI_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
09:34:58.560 -> configsip: 0, SPIWP:0xee
09:34:58.560 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
09:34:58.560 -> mode:DIO, clock div:1
09:34:58.560 -> load:0x3fff0018,len:4
09:34:58.560 -> load:0x3fff001c,len:1216
09:34:58.560 -> ho 0 tail 12 room 4
09:34:58.560 -> load:0x40078000,len:10944
09:34:58.560 -> load:0x40080400,len:6388
09:34:58.560 -> entry 0x400806b4
09:34:58.794 -> Connecting to Redmi Note 8
09:34:59.388 -> .....
09:35:09.404 -> WiFi connected.
09:35:09.404 -> IP address:
09:35:09.404 -> 192.168.43.19
```

Autoscroll Show timestamp Newline 115200 baud Clear output