

ESP32 아두이노 기초 웨비나

1. 아두이노 ESP32 보드 패키지 설치

관련 설명 블로그 링크

<https://blog.naver.com/chandong83/223981020228>

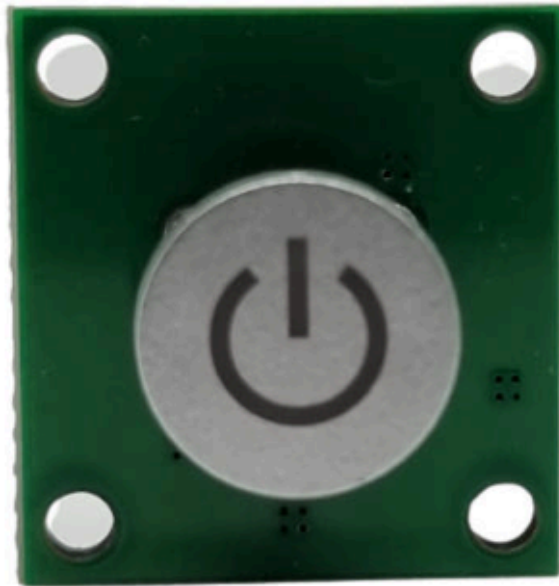
<https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>

안정화 버전 패키지

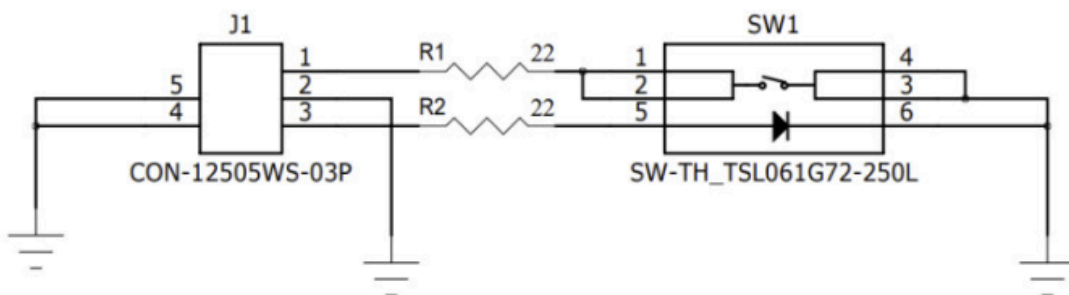
https://espressif.github.io/arduino-esp32/package_esp32_index.json

2. GPIO In/Out

LED 스위치 모듈



모듈 회로도



2.1. ESP32 GPIO OUTPUT

```
void setup() {
  pinMode(12, OUTPUT);
}

void loop() {
  digitalWrite(12, HIGH);
  delay(500);
  digitalWrite(12, LOW);
  delay(500);
}
```

2.2. ESP32 GPIO INPUT

```
void setup() {
  pinMode(12, OUTPUT);
  pinMode(14, INPUT_PULLUP);
}

void loop() {
  uint32_t delayTime;
  if(digitalRead(14) == LOW)
  {
    delayTime = 500;
  }
  else
  {
    delayTime = 1000;
  }
  digitalWrite(12, HIGH);
  delay(delayTime);
  digitalWrite(12, LOW);
  delay(delayTime);
}
```

3. 시리얼 디버그

```
void setup() {  
  Serial.begin(115200);  
  Serial.print("Hello");  
  Serial.println(" ESP32");  
}
```

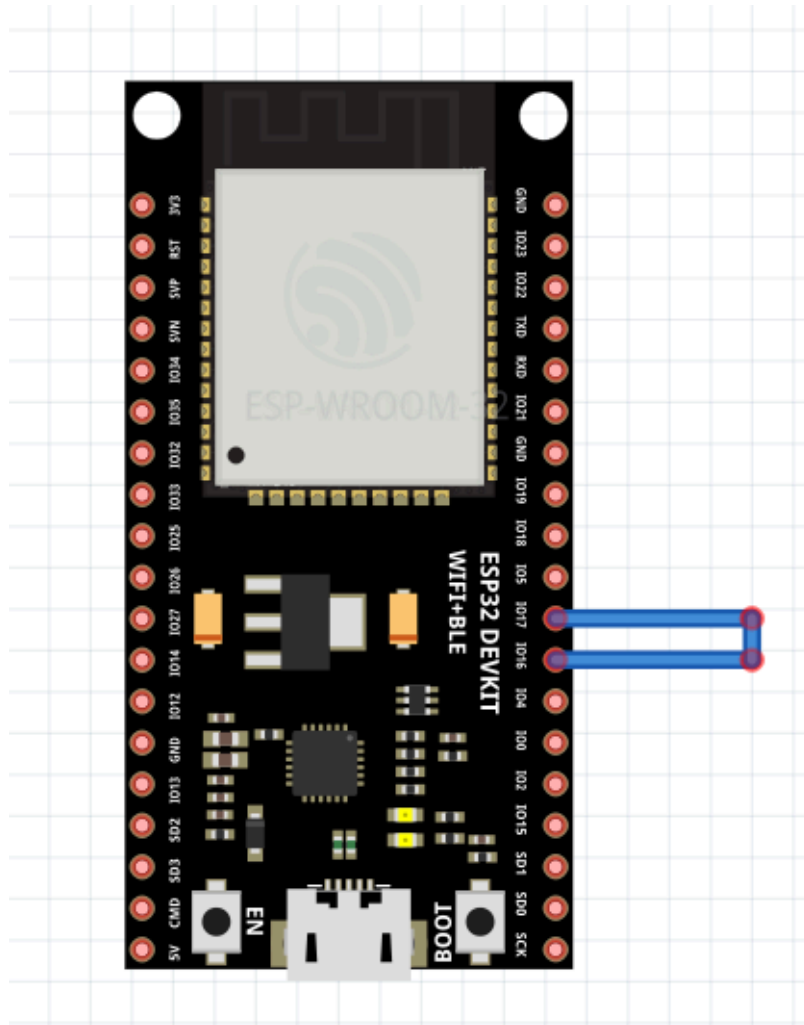
```
}

void loop() {
  static uint32_t tm = 0;
  static uint8_t count = 0;
  if((millis()-tm) > 1000)
  {
    tm = millis();
    count++;
    Serial.printf("Count: %d\n", count);
  }
  if(Serial.available()) {
    Serial.write(Serial.read());
  }
}
```

4. ESP32 UART 통신 테스트

관련 설명 블로그 링크

<https://blog.naver.com/chandong83/222630154615>



```

void setup() {
  // USB에 연결도니 UART RX: 3, TX: 1
  Serial.begin(115200);
  // 기본 설정된 RX: 16, TX: 17 핀 적용
  //Serial2.begin(115200);

  // RX 16핀, TX 17핀, 8비트, 페리티 비트 없음, Stop 비트 1
  Serial2.begin(115200, SERIAL_8N1, 16, 17);
}

void loop() {
  /*
  * 에코 기능 구현
  */
}

```

```
*/  
//모니터 프로그램으로 입력이 들어오면  
if(Serial.available()){  
  //들어온 데이터를 Serial2로 그대로 전송  
  Serial2.write(Serial.read());  
}  
//시리얼 2번에 데이터가 들어오면  
if(Serial2.available()){  
  //들어온 데이터를 Serial로 그대로 전송  
  Serial.write(Serial2.read());  
}  
}
```

5. 블루투스 SPP



블루투스 SPP 모듈(HC-05 / HC-06)

관련 설명 블로그 링크

<https://blog.naver.com/chandong83/222609516442>

BluetoothSerial > SerialToSerialBT 예제

```
// 블루투스 시리얼(SPP) 사용 헤더
#include "BluetoothSerial.h"

// BT와 BLUEROID 가 설정에서 활성화 되어있는지 확인하는 코드 일단 무시해도 좋음
// (보드 설정만 잘하면 문제 없기 때문)
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif
```

```

// SPP 클래스 선언
BluetoothSerial SerialBT;

void setup() {
  // 모니터 프로그램 시리얼 시작
  Serial.begin(115200);
  // SPP 시작 인자로 입력하는 것은 상대방 기기에 스캔시 표시되는 이름이다.
  SerialBT.begin("ESP32test");
  // 모니터 프로그램에 준비 완료 알려줌
  Serial.println("The device started, now you can pair it with bluetooth!");
}

void loop() {
  // 모니터 프로그램으로 받은 데이터가 있다면
  if (Serial.available()) {
    // SPP로 데이터 전송
    SerialBT.write(Serial.read());
  }
  // SPP로 받은 데이터가 있다면
  if (SerialBT.available()) {
    // 그대로 모니터 프로그램쪽으로 전송
    Serial.write(SerialBT.read());
  }
  // 20ms 대기 (해당부분은 코드 구성상 없는 것이 좋을 것 같다.)
  delay(20);
}

```

6. WebServer

WiFi > SimpleWiFiServer

```
#include <WiFi.h>

const char *ssid = "e4ds_HiGenis_2.4G";
const char *password = "e4ds20251125";

NetworkServer server(80);

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

```

pinMode(12, OUTPUT); // set the LED pin mode

delay(10);

// We start by connecting to a WiFi network

Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

server.begin();
}

void loop() {
  NetworkClient client = server.accept(); // listen for incoming clients

  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the
client
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then

```

```

Serial.write(c);      // print it out the serial monitor
if (c == '\n') {     // if the byte is a newline character

    // if the current line is blank, you got two newline characters in a row.
    // that's the end of the client HTTP request, so send a response:
    if (currentLine.length() == 0) {
        // HTTP headers always start with a response code (e.g. HTTP/1.1 200
OK)
        // and a content-type so the client knows what's coming, then a blank l
ine:
        client.println("HTTP/1.1 200 OK");
        client.println("Content-type:text/html");
        client.println();

        // the content of the HTTP response follows the header:
        client.print("Click <a href=\"/H\">here</a> to turn the LED on pin 12 o
n.<br>");
        client.print("Click <a href=\"/L\">here</a> to turn the LED on pin 12 off.
<br>");

        // The HTTP response ends with another blank line:
        client.println();
        // break out of the while loop:
        break;
    } else { // if you got a newline, then clear currentLine:
        currentLine = "";
    }
} else if (c != '\r') { // if you got anything else but a carriage return chara
cter,
    currentLine += c;   // add it to the end of the currentLine
}

// Check to see if the client request was "GET /H" or "GET /L":
if (currentLine.endsWith("GET /H")) {
    digitalWrite(12, HIGH); // GET /H turns the LED on
}

```

```
    if (currentLine.endsWith("GET /L")) {  
        digitalWrite(12, LOW); // GET /L turns the LED off  
    }  
}  
}  
// close the connection:  
client.stop();  
Serial.println("Client Disconnected.");  
}  
}
```