

LED operation with smartphone

- **Implementing Wi-Fi and Web server on ESP32**
- **File operations with SPIFFS**

Table of Contents 《LED operation with smartphone》

1. Overview

2. Wi-Fi connection

3. Web server function

4. LED operation with smartphone

5. SPIFFS function

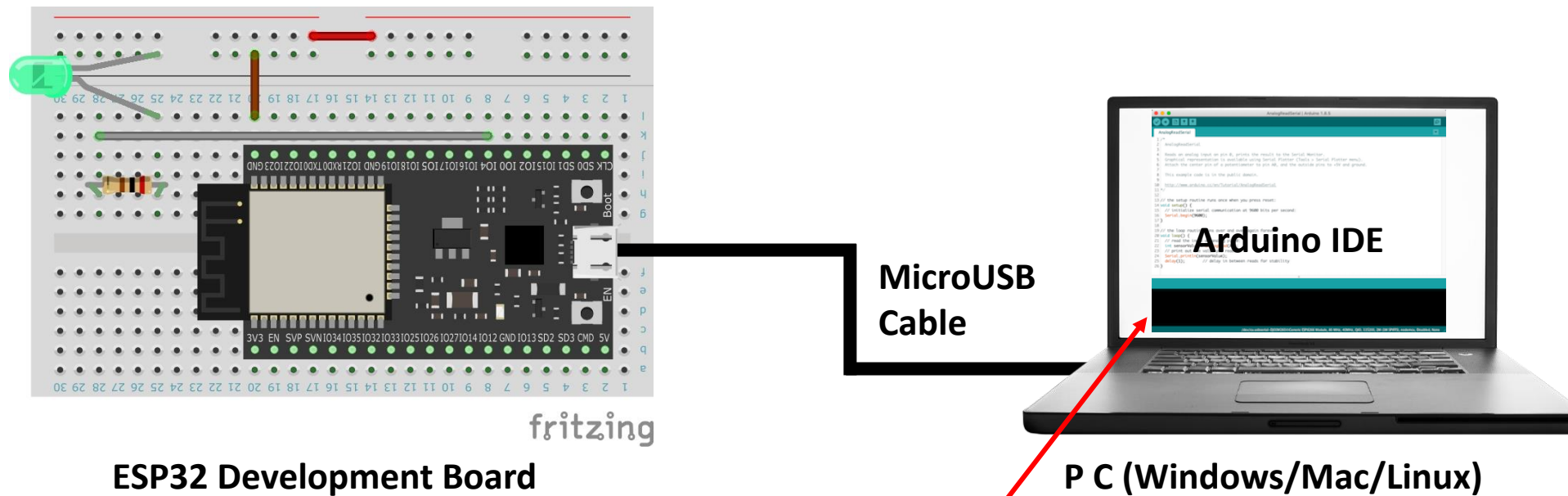
6. LED operation with smartphone (HTML file version)

1-1. Overall flow of Smart Remote Controller production

| No | Item | Content | Hard | Soft | Note |
|----|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|------|------|----------------------------|
| 1 | Overview | Overall flow, system configuration, items used, reasons for selection, development environment, etc. | - | - | Delivered in another video |
| 2 | LED | Learn the basics for beginners. We will make "L blinking" that lights up and blinks the LED. | ○ | ○ | |
| 3 | Infrared receiving sensor | Description of infrared receiving sensor Schematic to Wiring, Software | ○ | ○ | |
| 4 | Infrared transmission LED | Infrared transmission LED description Schematic to Wiring, Software | ○ | ○ | |
| 5 | LED operation with smartphone(at home) | We will create software to operate the LED with smartphone. (Web server function, SPIFFS operation) | - | ○ | this time this video |
| 6 | Remote control with smartphone(at home) | We will create software that to operate the remote control with smartphone indoors. (Button name, signal save/read) | - | ○ | Delivered in another video |
| 7 | Operate from outside And AI speaker cooperation | We will create software to operate the remote control with smartphone from the outdoors, and AI speaker cooperation. | - | ○ | |

1-2. the development environment “Arduino”

We will use Arduino as the development environment.



【Arduino Official site】

<https://www.arduino.cc/>

Downloadable

2. Wi-Fi connection

In ESP32, Wi-Fi connection can be programmed like bellow.

```
const char *ssid      = "##### SSID #####";  
const char *password = "### PASSWORD ###";  
IPAddress ip(192, 168, 1, 123); // IP address (IP used by this machine)  
IPAddress gateway(192, 168, 1, 1); // default gateway  
IPAddress subnet(255, 255, 255, 0); // sub-net mask
```

First define the settings that match your environment.

In the Setup function

```
// Wireless Wi-Fi connection  
WiFi.config( ip, gateway, subnet );  
WiFi.begin ( ssid, password );  
  
// Wi-Fi connection processing (infinite loop until connected)  
while ( WiFi.status() != WL_CONNECTED ) {  
    // Wait for 1 second  
    delay ( 1000 );  
    Serial.print ( "." );  
}  
Serial.print ( "Wi-Fi Connected! IP address: " );  
Serial.println ( WiFi.localIP() );
```

Sets the defined setting value.
Start a connection with the defined SSID, password.

Check the status every second until Wi-Fi is connected.
Repeat until connected.

When connected, the IP address will be displayed on the serial monitor.

3. Web server function

Web server functions are implemented using libraries.

- A library is a "component of a specific function".

Library to use

(There is a link in the summary column)

①ESPAsyncWebServer

<https://github.com/me-no-dev/ESPAsyncWebServer>

②AsyncTCP

<https://github.com/me-no-dev/AsyncTCP>

3. Web server function

After adding the library, program the Web server.

```
#include <ESPAsyncWebServer.h>
```

Import the library so that it can be used.

```
AsyncWebServer webServer ( 80 );
```

Define web server usage and port number
(Using the prescribed HTTP port 80)

Setup function

```
// Set WebServer reception process "/"  
webServer.on("/", HTTP_GET, [](AsyncWebServerRequest *request){  
  sendHtml(request);    // Send web page content  
  Serial.println ( "TOP page" );  
});
```

Program the contents to be processed when “/”,
that is, the TOP of the site is accessed.

```
// Set WebServer reception processing "/on"  
webServer.on("/on", HTTP_GET, [](AsyncWebServerRequest *request){  
  digitalWrite(LED_PIN, HIGH); // LED is lit by setting the LED pin to HIGH  
  Serial.println ( "LED ON" );  
  sendHtml(request);    // Send web page content  
});
```

Program what to do when the “/on” of the site
URL is accessed.
If there is access, we will add the contents to be
processed in the same way.

~ 省略 ~

```
// WebServer startup processing  
webServer.begin();
```

Start the web server with the programmed contents.

5. SPIFFS

Use the SPIFFS function and use Flash memory as a file system.

What is SPIFFS

SPIFFS (SPI Flash File System) is a method of using the connected flash memory as a file system. SPI (Serial Peripheral Interface) is an interface used between a microcontroller and its peripheral ICs, and is used to connect Flash memory in ESP32.

● From the ESP32 data sheet (official)

2 Block Diagram

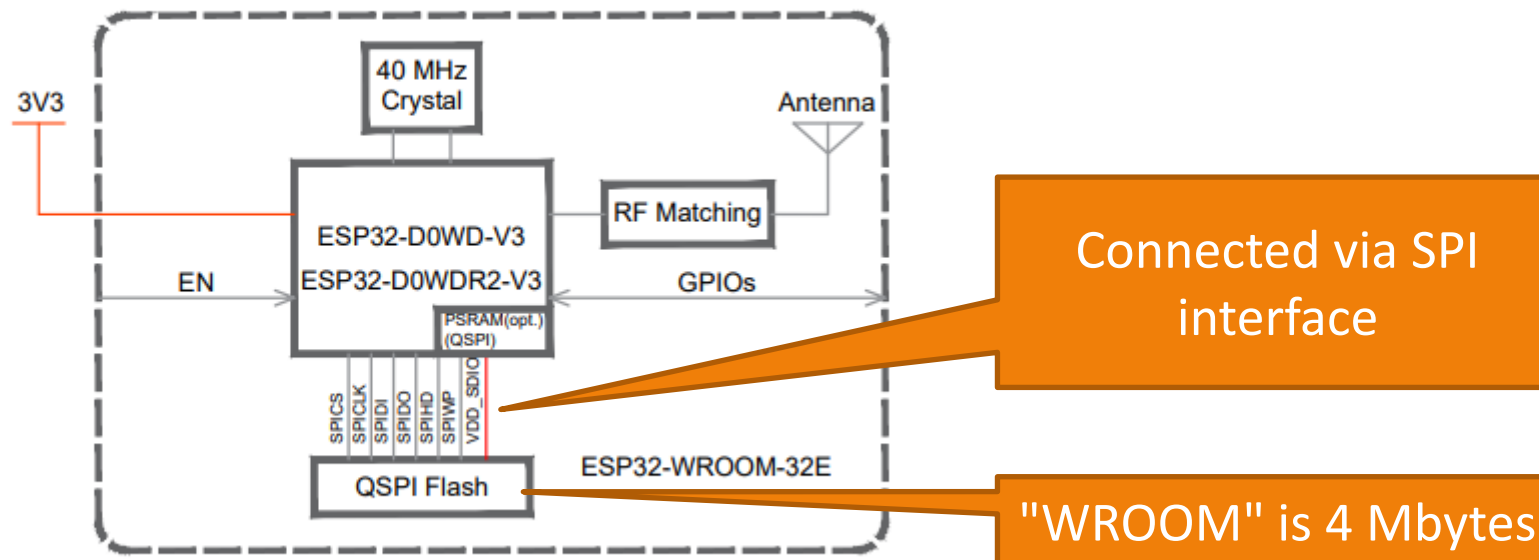


Figure 1: ESP32-WROOM-32E Block Diagram

[SPIFFS]
Use part of this connected Flash memory as a file system