

ESP32

Що е то и как се пуска?



Георги Маринов



<https://github.com/georgi-marinov>



ESPRESSIF



ESPRESSIF

Espressif Systems
Founded

2008

2013
Espressif Releases Its First Product -
ESP8089
(Wi-Fi Direct (P2P), Miracast, SoftAP)

2016
Espressif Releases its
Flagship Chip ESP32

2017
100-Million Target for IoT
Chip Shipments

2020
ESP32-C3 Launched

Announcing ESP32-H4:
Low-power SoC with
802.15.4 + Bluetooth 5.4
(LE) Connectivity

2024

32 ESP32



S - series

S2 ESP32-S2



S3 ESP32-S3



C - series

C2 ESP32-C2



C3 ESP32-C3



C5 ESP32-C5



C61 ESP32-C61



C6 ESP32-C6



H - series

H2 ESP32-H2



P - series

P4 ESP32-P4





ESP32

Xtensa® single-/dual-core 32-bit LX6 microprocessor(s) (up to 240Mhz)
Wi-Fi, Bluetooth 4.2



ESP32-C3

32-bit RISC-V single-core processor, (up to 160 MHz)
Wi-Fi, Bluetooth 5



ESP32-C6

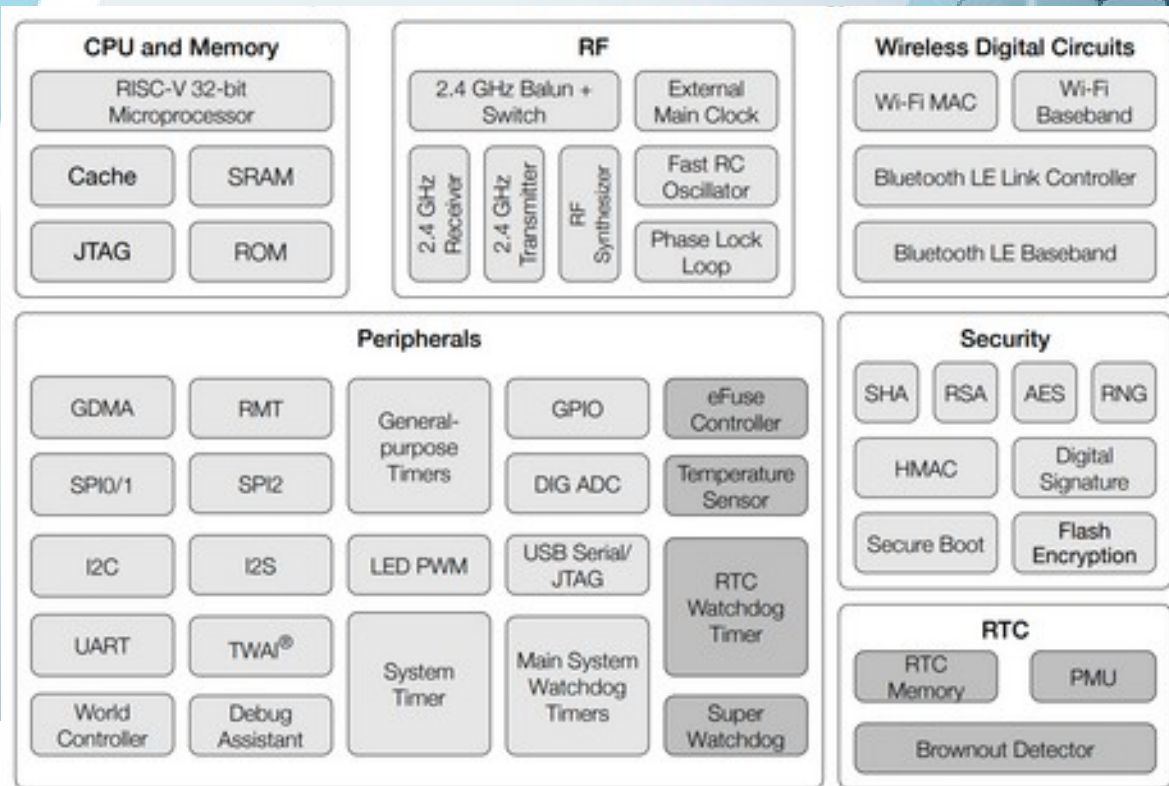
32-bit RISC-V single-core processor, (up to 160 MHz) + LP 32-bit RISC-V (up to 20 MHz)
Wi-Fi, Bluetooth 5.3, IEEE 802.15.4



ESP32-S3

Xtensa® dual-core 32-bit LX7 microprocessor (up to 240Mhz)
Wi-Fi, Bluetooth 5

ESP32-C3



32-bit RISC-V single-core

(up to 160 MHz)

400 KB SRAM

Internal / External FLASH

(SPI, Dual SPI, Quad SPI, and QPI)

22 or 16 GPIOs

Cryptographic

AES 128/256, SHA, RSA, RNG, HMAC

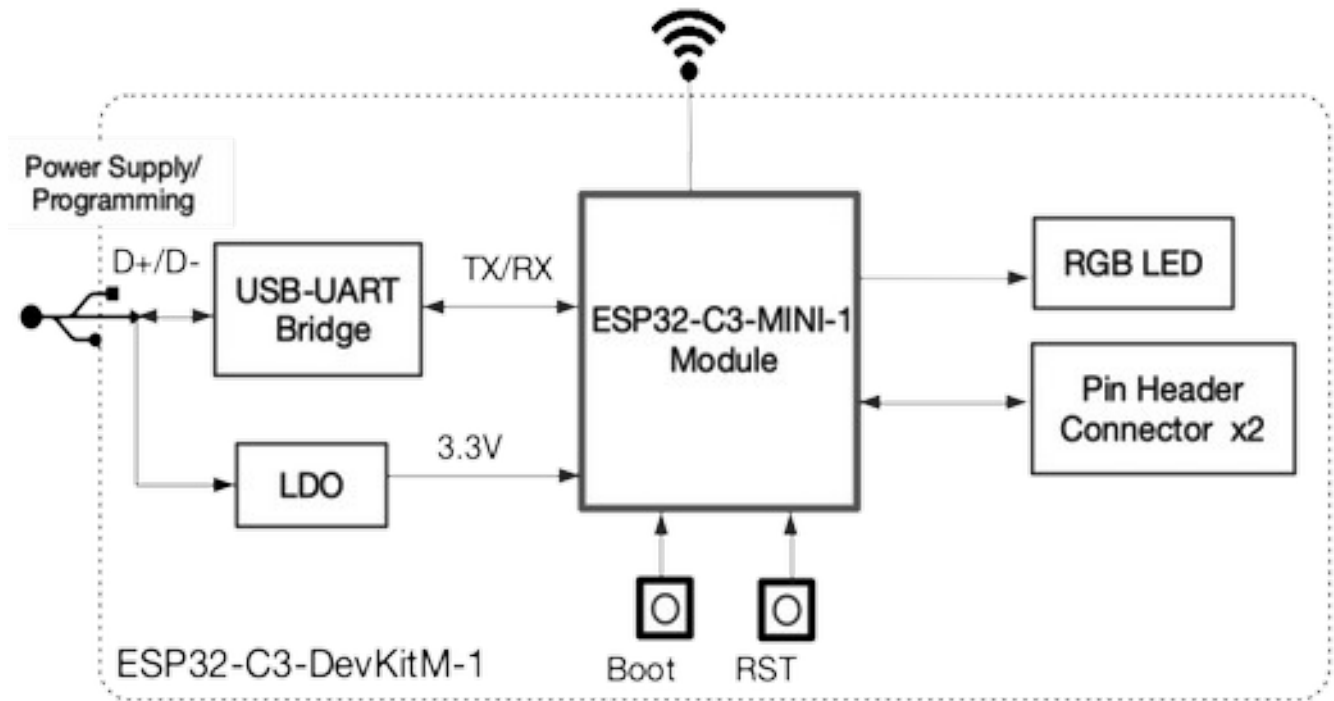
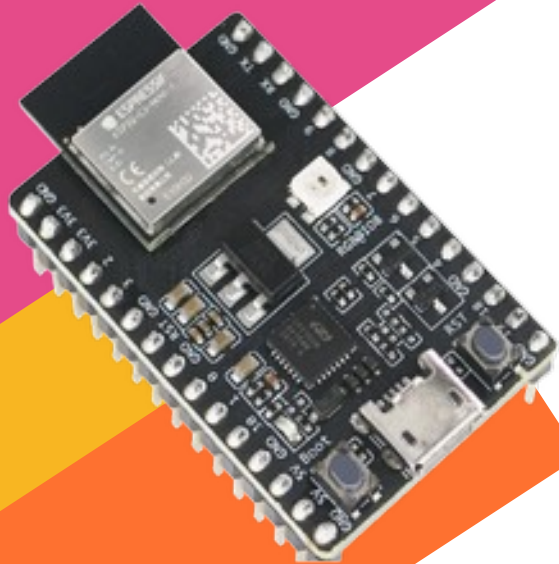
ESP32-C3

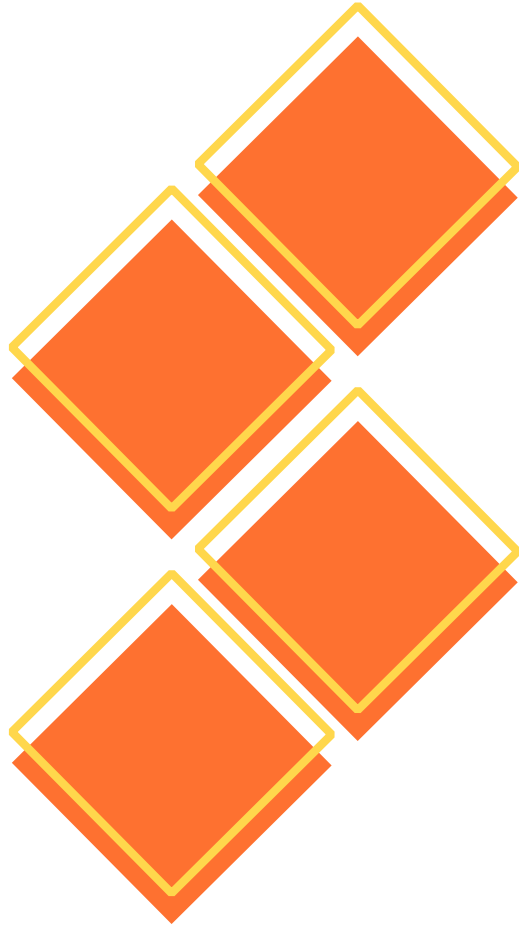
- UART (x2)
- SPI (x3)
- I2C
- I2S
- RMT (4 channels)
- LED PWM (16 channels)
- USB Serial/JTAG Controller
- DMA (6 channels)
- TWAI (CAN 2.0)
- ADC 12bit (x2)
- Temp sensor
- Timers (x 7)



ESP32-C3-DevKitM-1

<https://docs.espressif.com/projects/esp-dev-kits/>





Development tools



PlatformIO



Arduino



ESP-IDF

Arduino setup

<https://docs.espressif.com/projects/arduino-esp32/>

1

Download Arduino IDE

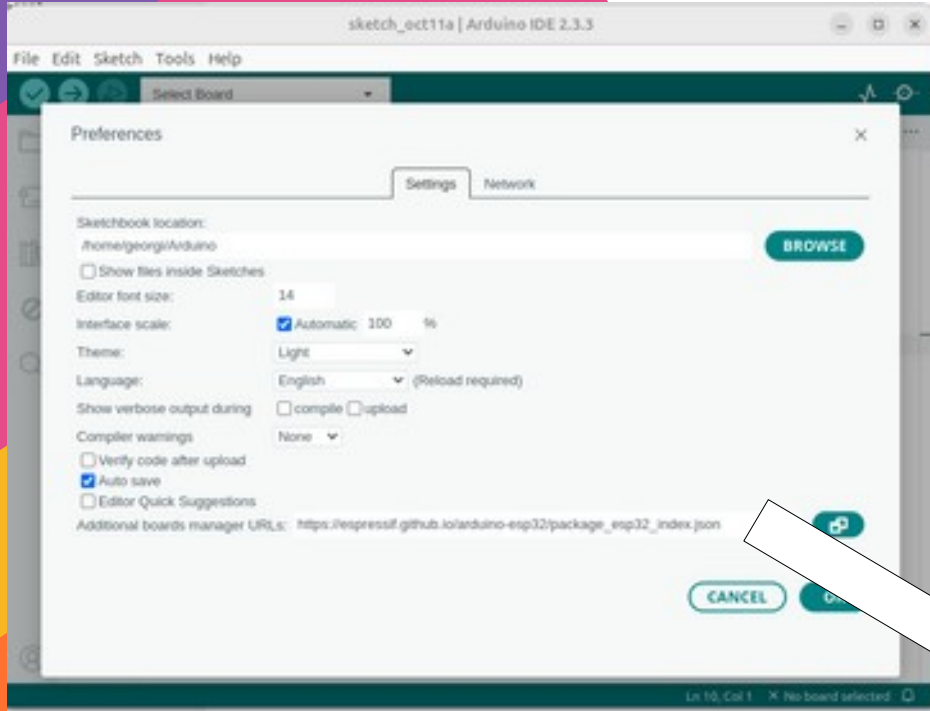


<https://www.arduino.cc/en/software>

Arduino setup

<https://docs.espressif.com/projects/arduino-esp32/>

2 Add ESP32 package json



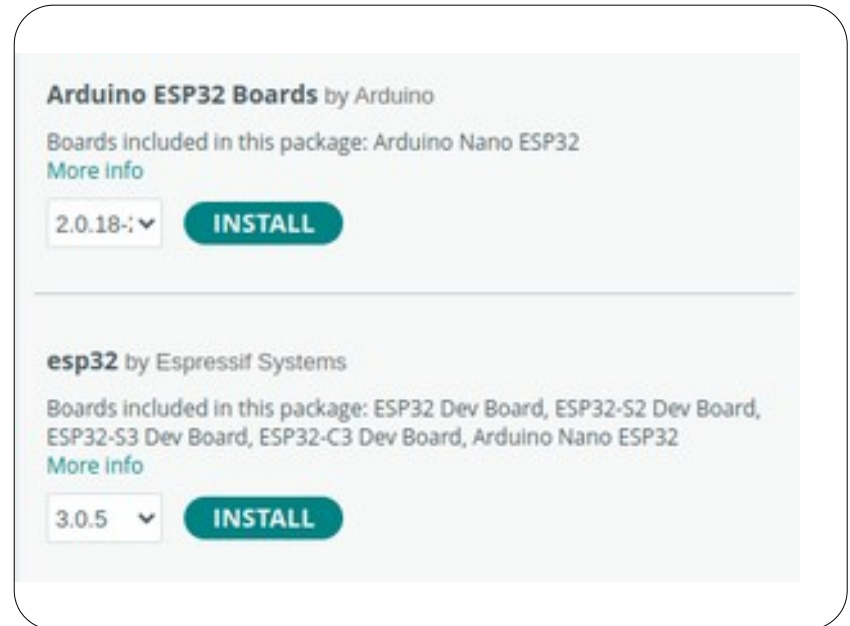
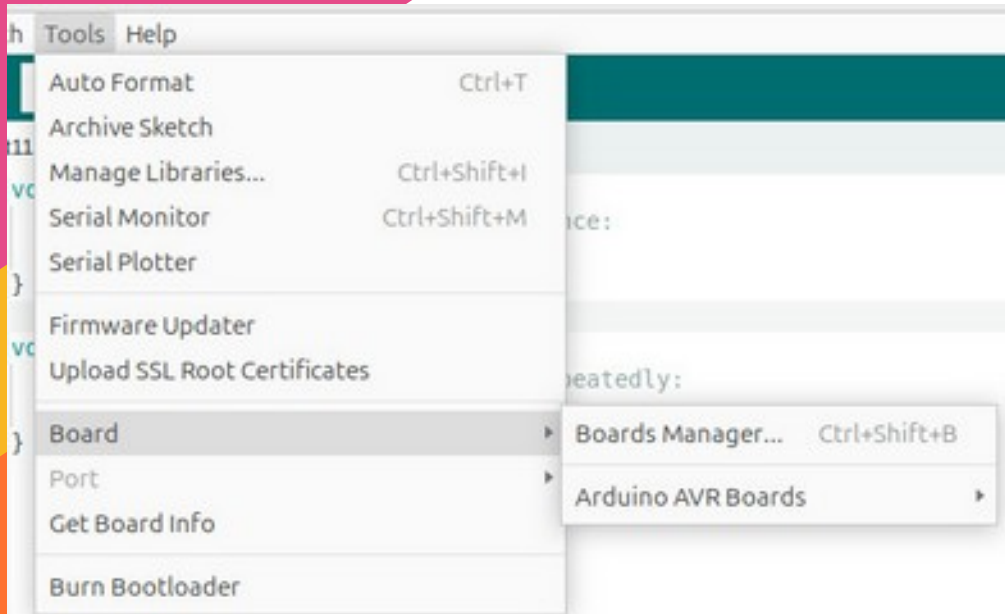
Additional board manager URLs:

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

Arduino setup

<https://docs.espressif.com/projects/arduino-esp32/>

3 Install ESP32 boards



Arduino setup

<https://docs.espressif.com/projects/arduino-esp32/>

4

Configure board

Board: "ESP32C3 Dev Module"

Port

Get Board Info

USB CDC On Boot: "Disabled"

CPU Frequency: "160MHz (WiFi)"

Core Debug Level: "None"

Erase All Flash Before Sketch Upload: "Disabled"

Flash Frequency: "80MHz"

Flash Mode: "QIO"

Flash Size: "4MB (32Mb)"

JTAG Adapter: "Disabled"

Partition Scheme: "Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS)"

Upload Speed: "921600"

Zigbee Mode: "Disabled"

Programmer

Burn Bootloader

Configuring platform.
Platform esp32:esp32@3.0.5

Windows

- ✓ Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS)
- Default 4MB with ffat (1.2MB APP/1.5MB FATFS)
- 8M with spiiffs (3MB APP/1.5MB SPIFFS)
- Minimal (1.3MB APP/700KB SPIFFS)
- No FS 4MB (2MB APP x2)
- No OTA (2MB APP/2MB SPIFFS)
- No OTA (1MB APP/3MB SPIFFS)
- No OTA (2MB APP/2MB FATFS)
- No OTA (1MB APP/3MB FATFS)
- Huge APP (3MB No OTA/1MB SPIFFS)
- Minimal SPIFFS (1.9MB APP with OTA/190KB SPIFFS)
- 16M Flash (2MB APP/12.5MB FATFS)
- 16M Flash (3MB APP/9.9MB FATFS)
- RainMaker 4MB
- RainMaker 4MB No OTA
- RainMaker 8MB
- Zigbee ZCZR 4MB with spiiffs

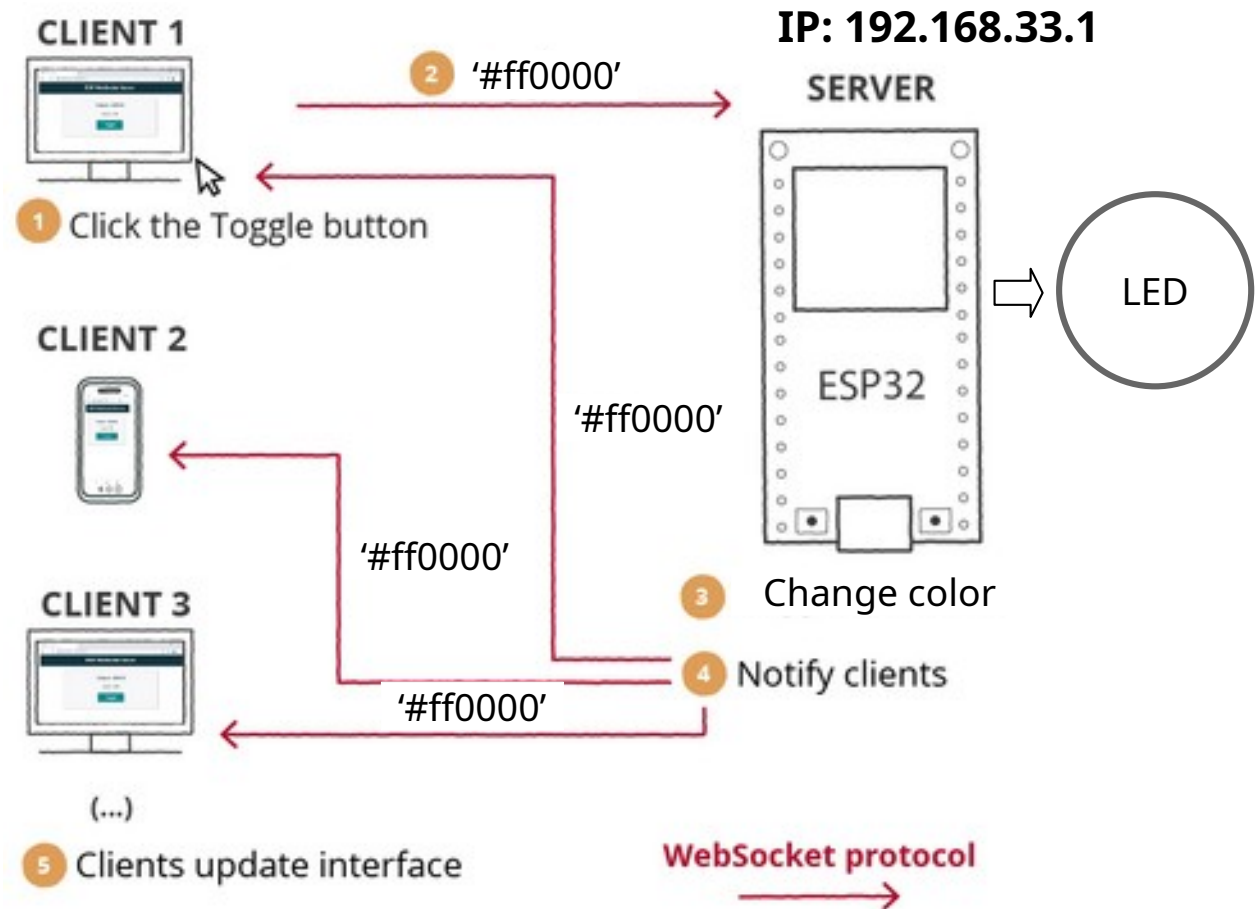
ESPAsyncTCP

AsyncTCP

ESPAsyncWebServer

Freenove WS2812 Lib

Application



Application

```
IPAddress AP_LOCAL_IP(192, 168, 33, 1);  
IPAddress AP_GATEWAY_IP(192, 168, 33, 254);  
IPAddress AP_NETWORK_MASK(255, 255, 255, 0);
```

```
WiFi.softAPConfig(AP_LOCAL_IP, AP_GATEWAY_IP, AP_NETWORK_MASK);  
WiFi.softAPsetHostname("espdemo");
```

```
// To initiate the Soft AP, pause the program if the initializat  
if (!WiFi.softAP(ssid, password))  
{  
    Serial.println("Soft AP creation failed.");  
    while (1);  
}
```

Application

```
struct LedState {  
    uint8_t red = 0;  
    uint8_t green = 0;  
    uint8_t blue = 0;  
} led_state;
```

```
// Led  
Freenove_ESP32_WS2812 strip =  
    Freenove_ESP32_WS2812(LED_COUNT, LED_PIN, CHANNEL);
```

```
// LED strip start  
strip.begin();  
strip.setLedColor(0, led_state.red, led_state.green, led_state.blue);
```

Application

```
// Web server
AsyncWebServer server(80);
AsyncWebSocket ws("/ws");
```

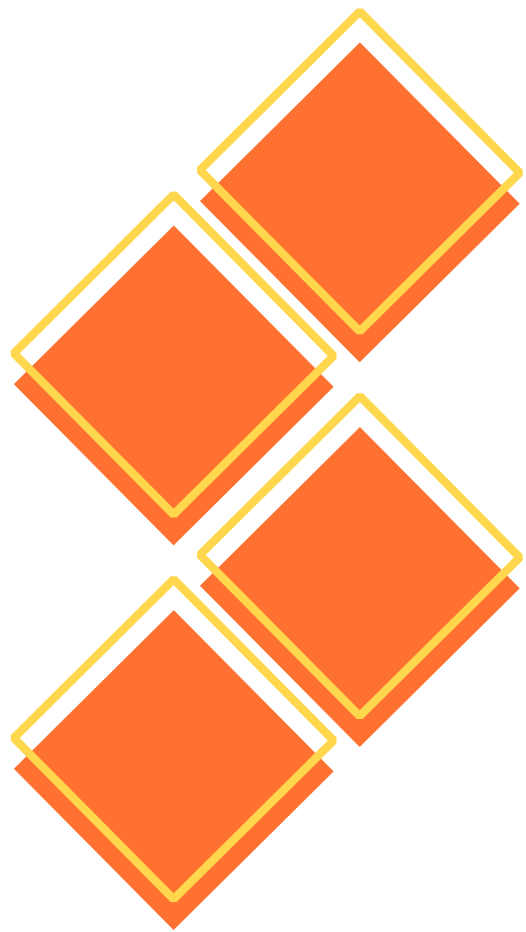
```
void initWebSocket() {
    ws.onEvent(onEvent);
    server.addHandler(&ws);
}
```

```
initWebSocket();
```

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(200, "text/html", index_html);
});
```

```
server.on("/led", HTTP_GET, LedHandler);
server.onNotFound(notFound);
```

```
server.begin();
```

Благодаря за вниманието



Георги Маринов



<https://github.com/georgi-marinov>

