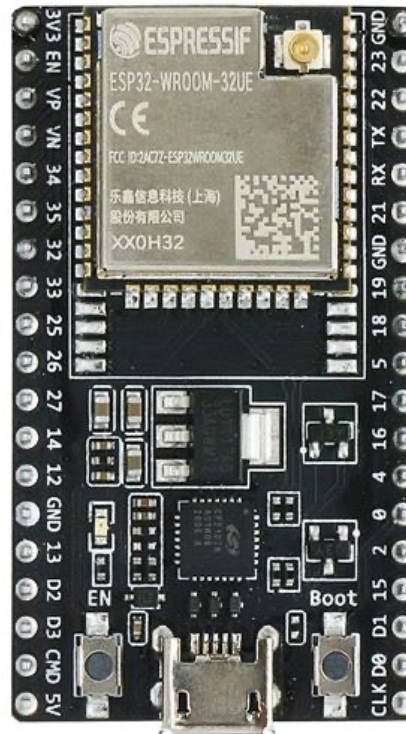


Embedded Systems

Master 's Degree in Informatics Engineering



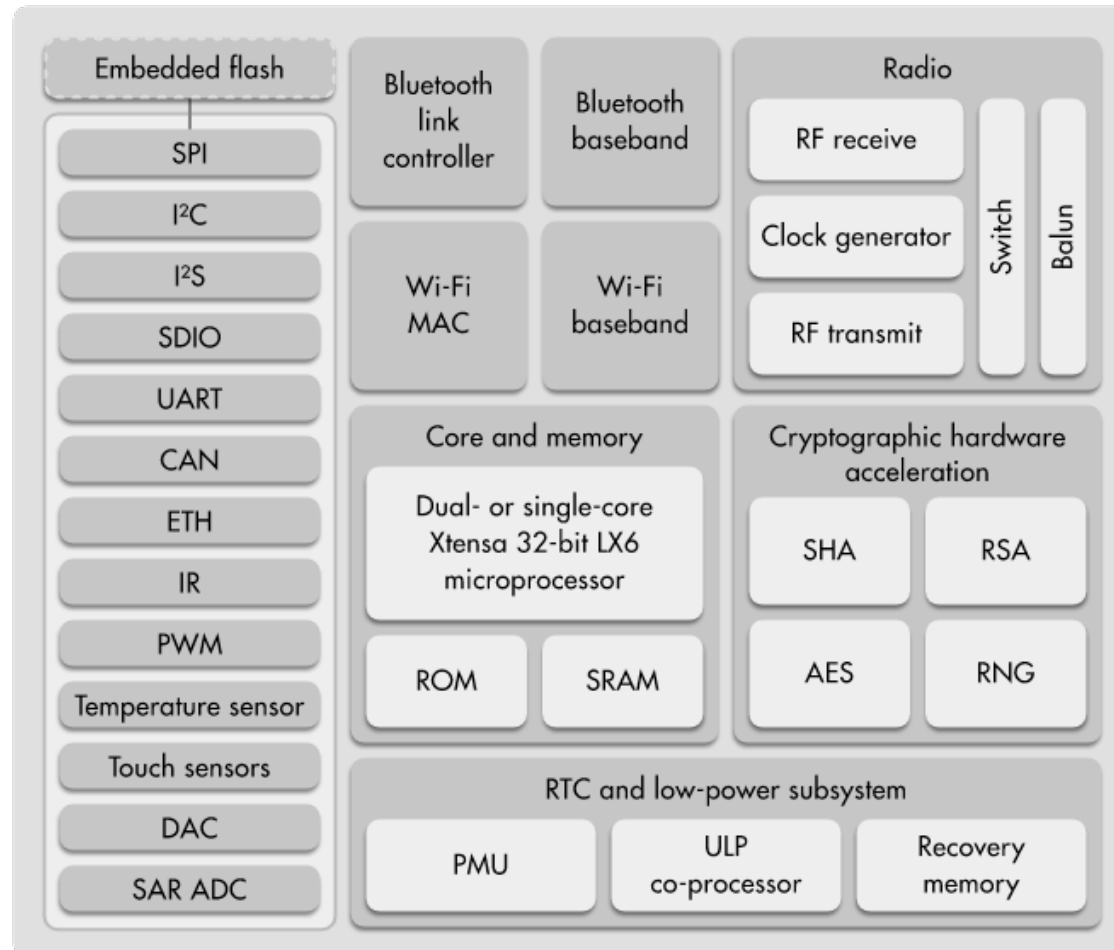
ESP-32 Platform





ESP32 Block Diagram

ESP32 FUNCTION BLOCK DIAGRAM





ESP-01 vs NodeMCU?

	NodeMCU v1.0 (ESP8266)	ESP-32
CPU	Tenselica Xtensa LX106	Tenselica Xtensa LX6
CPU Clock	80MHz/160MHz	up to 240MHz
Instruction SRAM	64KBytes (<36KB Stat Mode)	
Data SRAM	96KBytes	520KBytes
Flash Memory	4MBytes	
GPIO Pins	11	39
ADC	1	18
USB-to-Serial	CH340G	CP210x
UART/SPI/I2C/I2S	1/1/1/0	3/4/2/2
WiFi Built-In	802.11 b/g/n	
Bluetooth	-	V4.2 BLE



ESP-32 WROOM-32D

- Low cost, compact that includes Wi-Fi and BT-BLE Modules
- Power Supply: +3.3V only
- Built-in low power dual core 32-bit MCU from 80MHz to 240MHz
- Extremely low power co-processor to monitor peripherals
- Supports Deep sleep (<5uA)
- Similar WiFi features than the ESP8266 family, and encrypted acceleration.
- Integrates capacitive touch sensors, HALL sensor, SD card interface
- Support freeRTOS

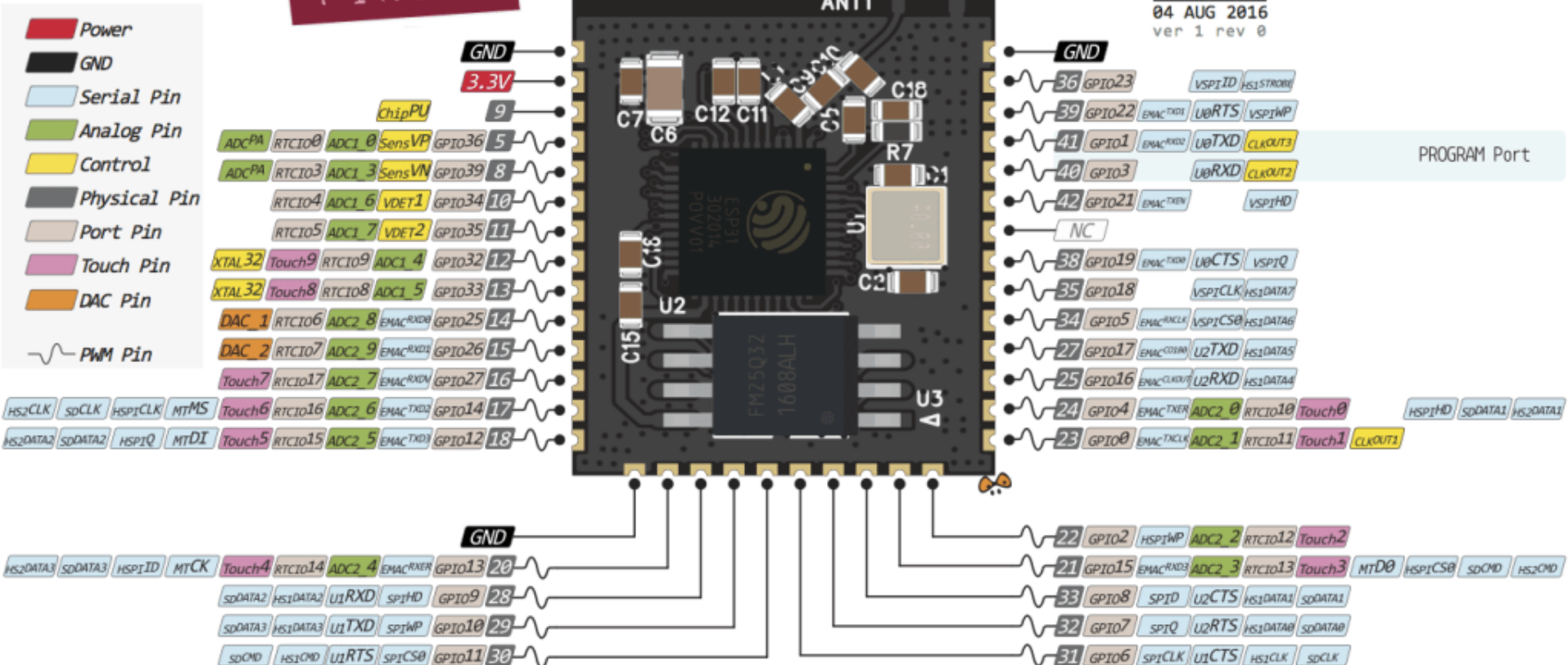




ESP32 – WROOM-32D

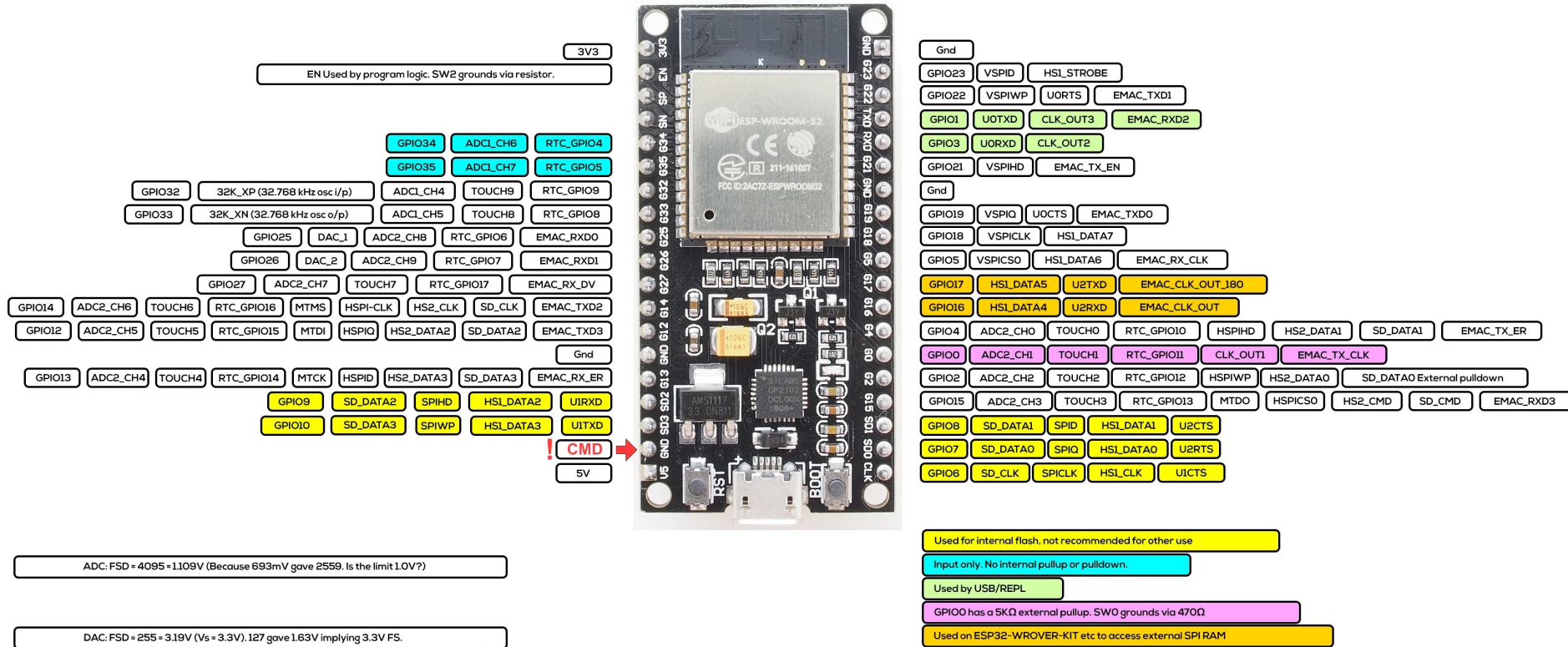
WROOM32

PINOUT





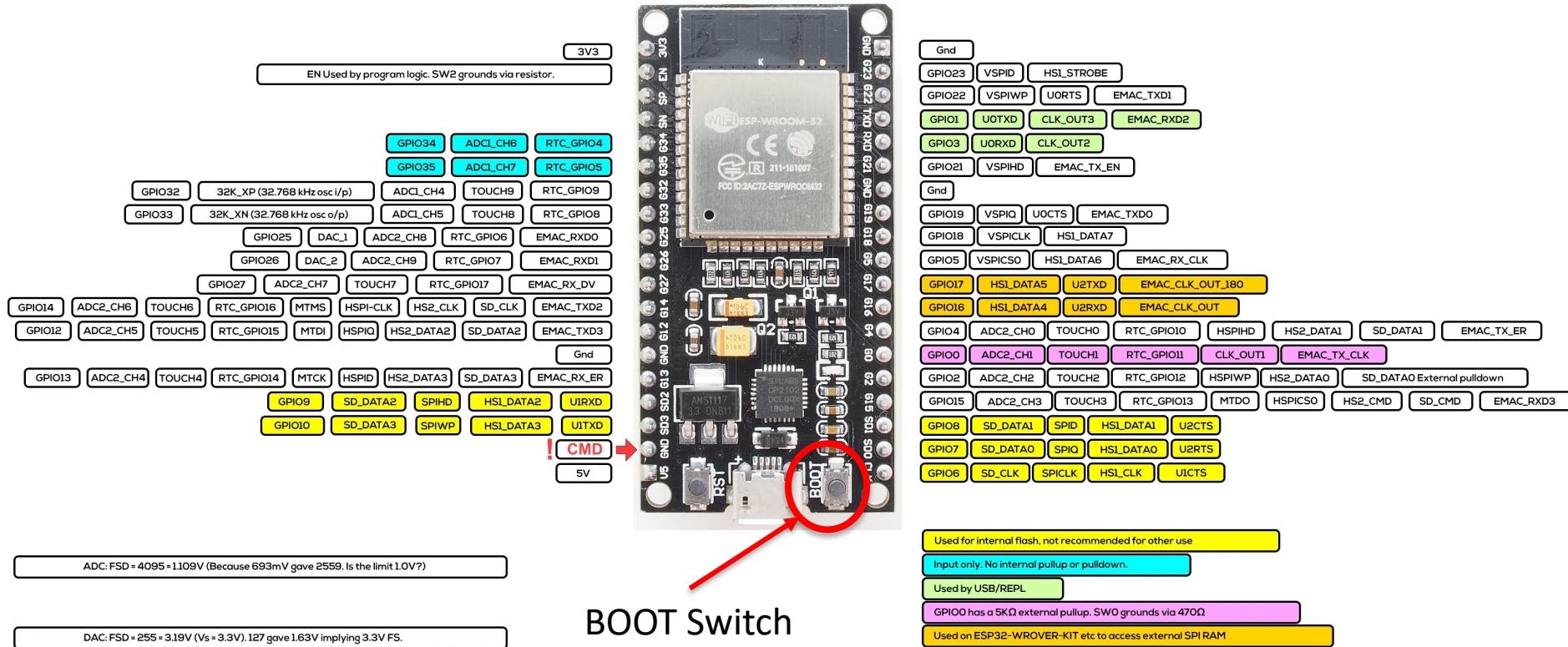
ESP32 – WROOM-32D



ESP32-D2WD is the chip with embedded 2MB flash and the internal flash is connected to different pins (GPIO16, GPIO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1)



ESP32 – WROOM-32D



ESP32-D2WD is the chip with embedded 2MB flash and the internal flash is connected to different pins (GPIO16, GPIO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1)



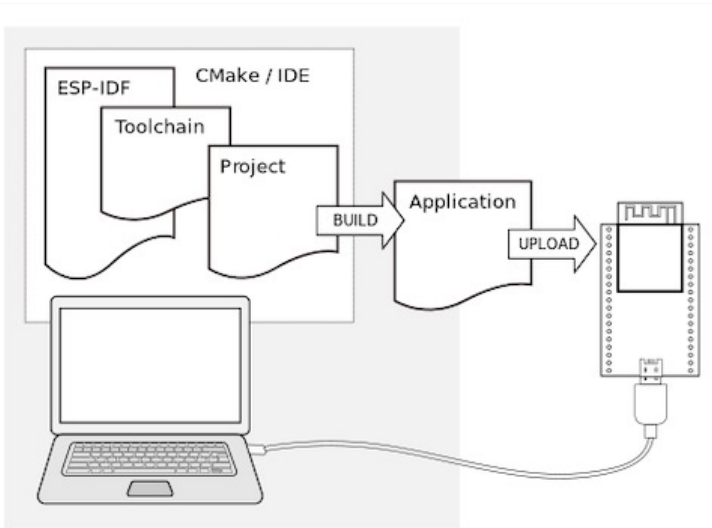
Programming ESP32

ESP – IDF (IoT Development Framework)

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html>

Native Expressig SDK

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/>



Eclipse Plugin

> <https://github.com/espressif/idf-eclipse-plugin>

VS Code Extension

> <https://github.com/espressif/vscode-esp-idf-extension>



Arduino IDE – ESP32 – WROOM32-D

Arduino - Preferences

Preferences

Settings Network

Sketchbook location:
/Users/fernandoguirado/Documents/Arduino **BROWSE**

☐ Show files inside Sketches

Editor font size: 12

Interface scale: ☒ Automatic 100 %

Theme: Light (Arduino) ▾

Language: English ▾ (Reload required)

Show verbose output during ☐ compile ☐ upload

Compiler warnings: None ▾

☐ Verify code after upload

☒ Auto save

☐ Editor Quick Suggestions

Additional boards manager URLs: https://dl.espressif.com/dl/package_esp32_index.json **+**

CANCEL **OK**

https://dl.espressif.com/dl/package_esp32_index.json



Arduino IDE - NodeMCU

Arduino – Tools > Boards > Boards Manager

sketch_oct13a | Arduino IDE 2.0.0

Select Board

BOARDS MANAGER

ESP32

Type: All

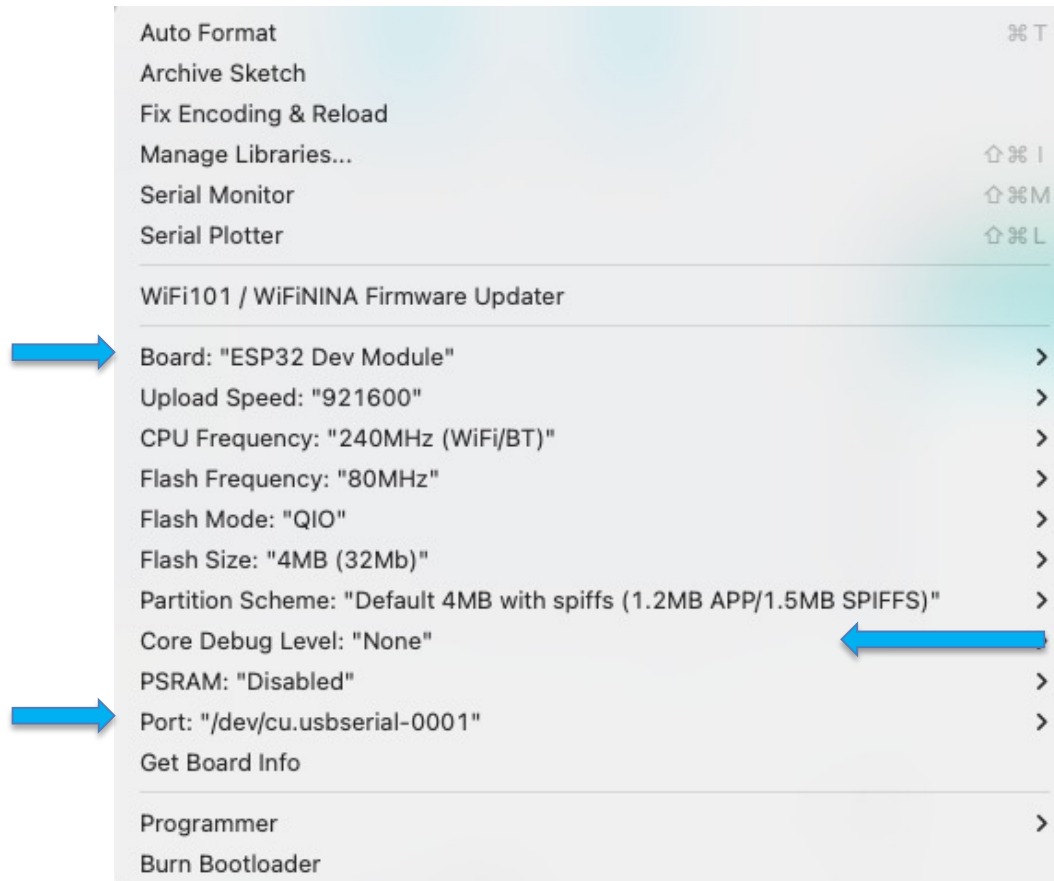
ESP32 Arduino
Version 1.0.6 INSTALLED

Boards included in this package:
M5Stack-ATOM, D-duino-32, TTGO T1, OLIMEX ESP32-GATEWAY, Metro ESP-32, Heltec Wireless Stick Lite, OLIMEX ESP32-DevKit-LiPo, MagicBit, WeMos WiFi&Bluetooth Battery, u-blox NINA-W10 series (ESP32), MGBOT IOTIK 32A, Hornbill ESP32 Minima, ESP32 FM DevKit, KITS ESP32 EDU, WiPy 3.0, ESPea32, DOIT ESPduino32, DOIT ESP32 DEVKIT V1, TTGO T7 V1.3 Mini32, T-Beam, ALKS ESP32, WEMOS D1 MINI ESP32, Adafruit ESP32 Feather, Heltec WiFi Kit 32, Onehorse ESP32 Dev Module, TTGO T7 V1.4 Mini32, M5Stick-C, M5Stack-CoreInk, Heltec WiFi LoRa 32(V2), Widora AIR, OLIMEX ESP32-EVB, M5Stack-Core-ESP32, INEX OpenKB, OLIMEX ESP32-PoE, OLIMEX ESP32-PoE-ISO, Silicognition wESP32, MH ET LIVE ESP32MiniKit, Turta IoT Node, Electronic SweetPeas - ESP320, XinaBox CW02, TTGO LoRa32-OLED v2.1.6, ProtoCentral HealthyPi 4, TTGO LoRa32-OLED V1, Hornbill ESP32 Dev, WEMOS LOLIN32 Lite, M5Stack-Timer-CAM, Senses's WEIZEN, Frog Board ESP32, LOLIN D32 PRO, FireBeetle-ESP32, Dongsen Tech Pocket 32, M5Stack-Core2, BPI-BIT, WEMOS LOLIN32, M5Stack-FIRE, AI Thinker ESP32-CAM, TinyPICO, VintLabs ESP32 Devkit, MGBOT IOTIK 32B, ESP32 Pico Kit, ESP32 Wrover Module, SparkFun LoRa Gateway 1-Channel, S.ODI Ultra v1, MH ET LIVE ESP32DevKIT, Heltec Wireless Stick, Heltec WiFi LoRa 32, LoPy4, SparkFun ESP32 Thing, Microduino-CoreESP32, SparkFun ESP32 Thing Plus, ThaiEasyElec's ESPino32, Piranha ESP-32, Noduino Quantum, ESP32vn IoT Uno, OROCA EduBot, ESP32 Dev Module, TTGO T-Watch, HONEYLemon, ET-Board, Labplus mPython, Nano32, LoPy, LOLIN D32, Pycom GPy, NodeMCU-32S, IntoRobot Fig, IMBRIOS LOGSENS_V1P1, ODROID ESP32, ESPectro32, WiFiduino32, Node32s



Arduino IDE – ESP32

Arduino – Select Board and Serial Port



Debugging feature is controllable over the IDE menu. The new menu points manage the real-time Debug messages.

The Serial port must be initialized by the user at the highest baudrate > **Serial.begin(115200)** and selected in the Debug Port menu

Debugging level is user selected

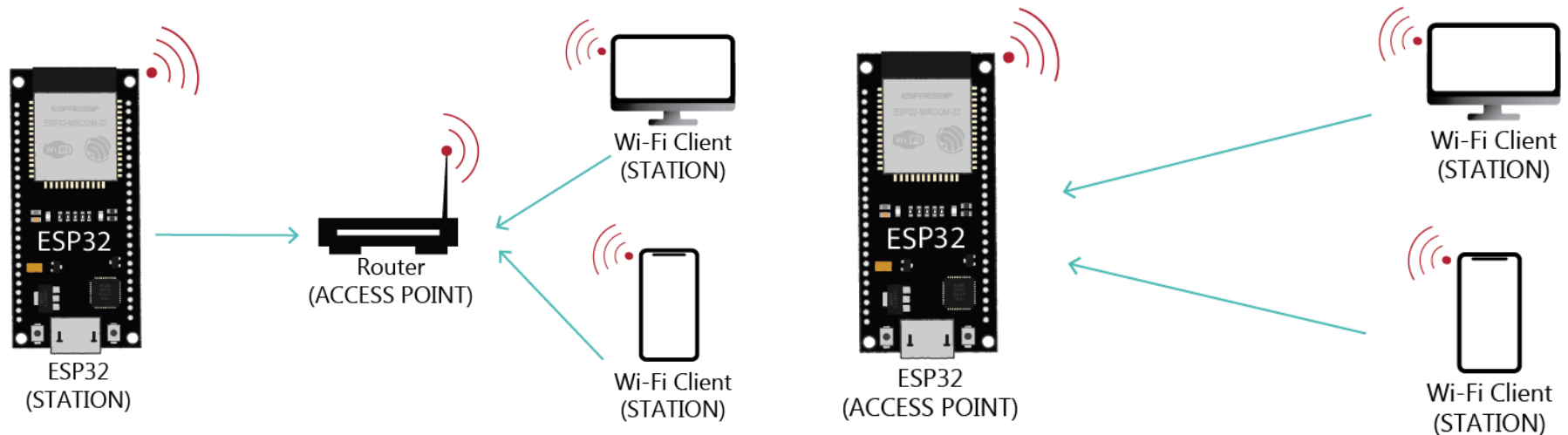


ESP32 WROOM-32D - WiFi

Station Mode (WIFI_STA) → It can connect to the Wi-Fi network as any other device

Access Point (WIFI_AP) → It establishes its own Wi-Fi network, then any other device can connect to the ESP32

Both Station and Access Point (WIFI_AP_STA) → It operates as both a station and a soft access point mode. This provides the possibility of building e.g. mesh networks.



```
#include "WiFi.h"
```



WiFi Station

WiFi.begin(ssid, password, channel, bssid, connect) / WiFi.begin(ssid, password) / WiFi.begin()

- **ssid** - a character string containing the SSID of Access Point we would like to connect to, may have up to 32 characters
- **password** to the access point, a character string that should be minimum 8 characters long and not longer than 64 characters
- **channel** of AP, if we like to operate using specific channel, otherwise this parameter may be omitted
- **bssid** - mac address of AP, this parameter is also optional
- **connect** - a boolean parameter that if set to **false**, will instruct module just to save the other parameters without actually establishing connection to the access point

WiFi.config(local_ip, gateway, subnet, [dns1], [dns2])

> Disable DHCP client and set the IP configuration of station interface to user defined arbitrary values.

- **local_ip** - enter here IP address you would like to assign the ESP station's interface
- **gateway** - should contain IP address of gateway (a router) to access external networks
- **subnet** - this is a mask that defines the range of IP addresses of the local network
- **dns1, dns2** - optional parameters that define IP addresses of Domain Name Servers (DNS) that maintain a directory of domain names (like e.g. www.google.co.uk) and translate them for us to IP addresses



WiFi Access Point

WiFi.softAP(ssid) / WiFi.softAP(ssid, psk, channel, hidden, [max_connection])

- **ssid** - character string containing network SSID (max. 32 characters)
- **psk** - optional character string with a pre-shared key. For WPA2-PSK network it should be minimum 8 characters long and not longer than 64 characters. If not specified, the access point will be open for anybody to connect.
- **channel** - optional parameter to set Wi-Fi channel, from 1 to 13. Default channel = 1.
- **hidden** - optional parameter, if set to true will hide SSID.
- **max_connection** - optional parameter to set max simultaneous connected stations, from 0 to 8. Defaults to 4. Once the max number has been reached, any other station that wants to connect will be forced to wait until an already connected station disconnects.

*There are other overloaded methods

WiFi.softAPConfig(local_ip, gateway, subnet)

- **local_ip** - enter here IP address you would like to assign the ESP station's interface
- **gateway** - should contain IP address of gateway (a router) to access external networks
- **subnet** - this is a mask that defines the range of IP addresses of the local network

>Not using this method the network established by softAP will have default IP address of 192.168.4.1



WiFi Server & Client

It provides functionality to other programs or devices, called clients

Methods come directly from the Arduino library

WiFiServer server(port) → Creates a new server **port**

- **port** - the port to listen on (int)

WiFiClient client → Creates a client able to be connected to any server