

Microcontroller Boards

Useful information.

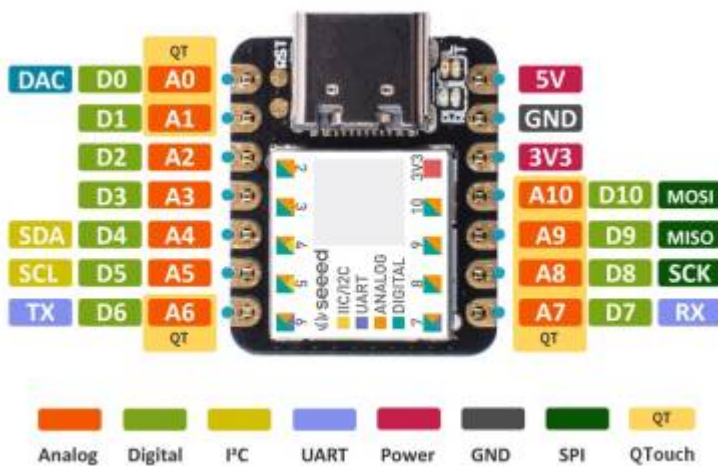
Seeed Studio's XIAO

Based on SAMD21 (Cortex-M0)

Note: *Arduino IDE Board Setup*

(https://files.seeedstudio.com/arduino/package_seeeduino_boards_index.json)

Pinouts shown below (*Disclaimer: Image copied from some online page.*).



Seeed Studio's ESP32-C3

Using ESP32 core (?duh!)

Note: *Arduino IDE Board Setup (Espressif's)*

Pinouts shown below (*Disclaimer: Image copied from some online page.*).



ESP32 DEVKIT

Espressif's ESP32 - got this board to play around with...

[esp32_devkit1.txt](#)

```
-----  
ESP32-DEVKITV1  
-----
```

- clone of DOIT's ESP32 DEVKIT V1?
- ESP-WROOM-32 (ESP32-D0WDQ6)
 - = ESP32-D0WD-based modules with integrated flash
- DECISION20200822:
 - = toolchain and build environment is simply not worth it!
 - = WILL USE micropython instead!

```
-----  
micropython  
-----
```

- get some tools
 - > pip install esptool rshell
- download binary from <https://micropython.org/download/esp32/>
 - = selected esp32-idf3-20191220-v1.12.bin
 - = idf4 seems to be missing LAN? stick with idf3 for now...
- erase flash
 - > esptool.py --chip esp32 --port /dev/ttyUSB0 erase_flash
- flash micropython
 - > esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 460800 write_flash -z 0x1000 <bin>
- use rshell to manage code (vfs)
 - > rshell --port /dev/ttyUSB0 --buffer-size=32
- download file in rshell
 - > cp <file> /pyboard[/<more_path>]

```
-----  
board info  
-----
```

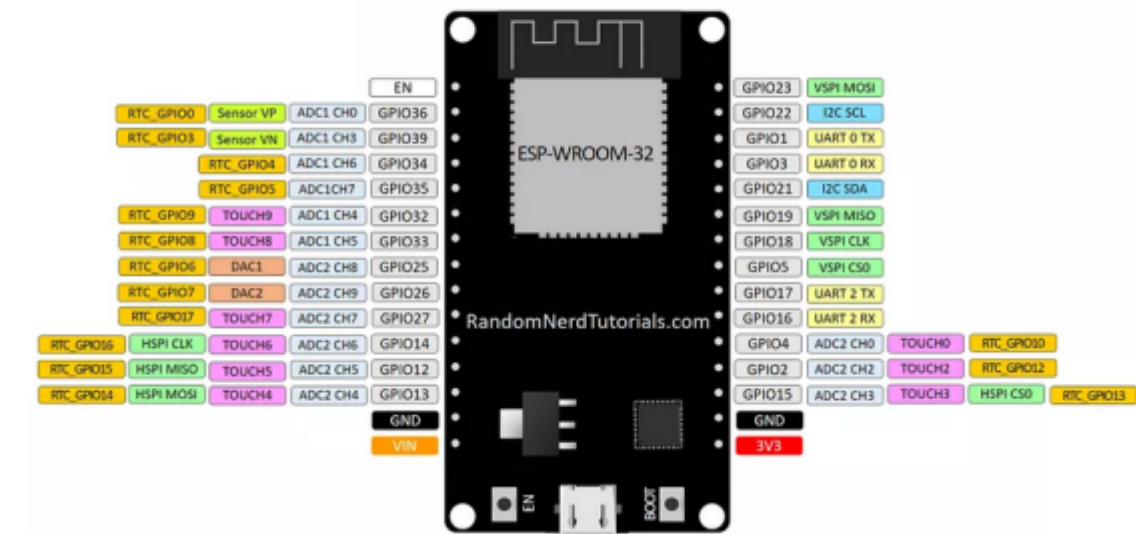
```
esp32 doit devkit v1  
vid = 0x10c4  
pid = 0xea60  
sn = 0001
```

```
esp32 doit devkit v1 (aliexpress - usb type c)  
vid = 0x1a86  
pid = 0x7523  
sn = (null)
```

Decided to use [micropython](#). Pinouts shown below (*Disclaimer: Image copied from [RandomNerdTutorials](#)*).

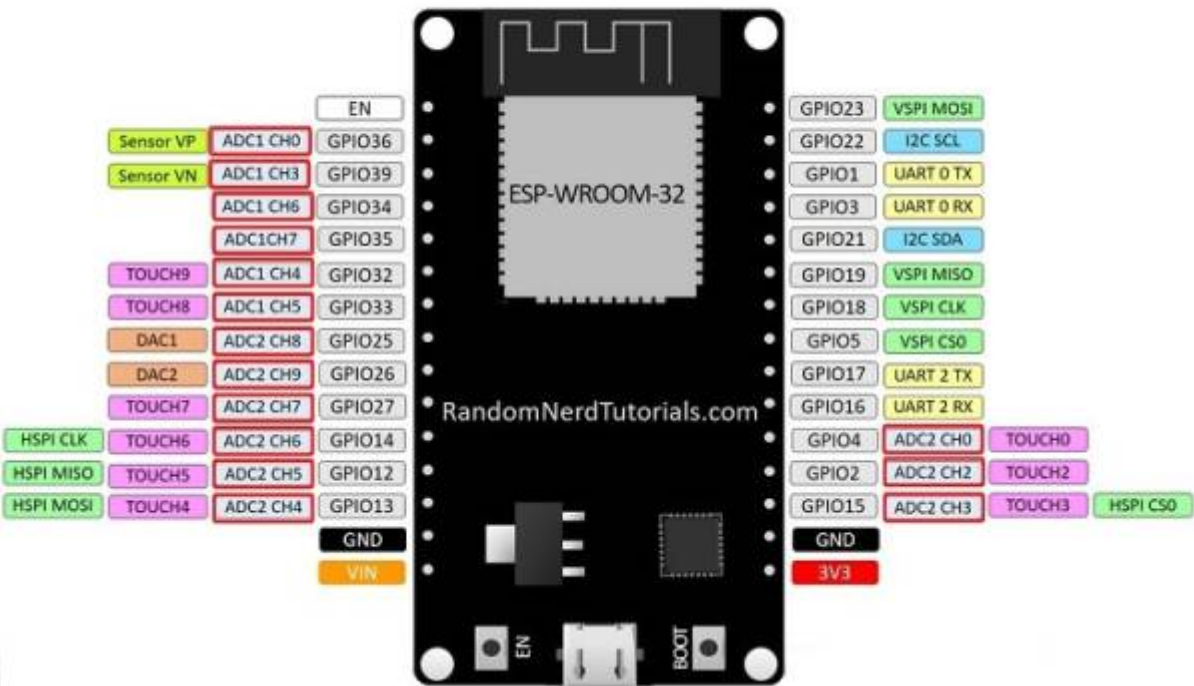
ESP32 DEVKIT V1 - DOIT

version with 30 GPIOs



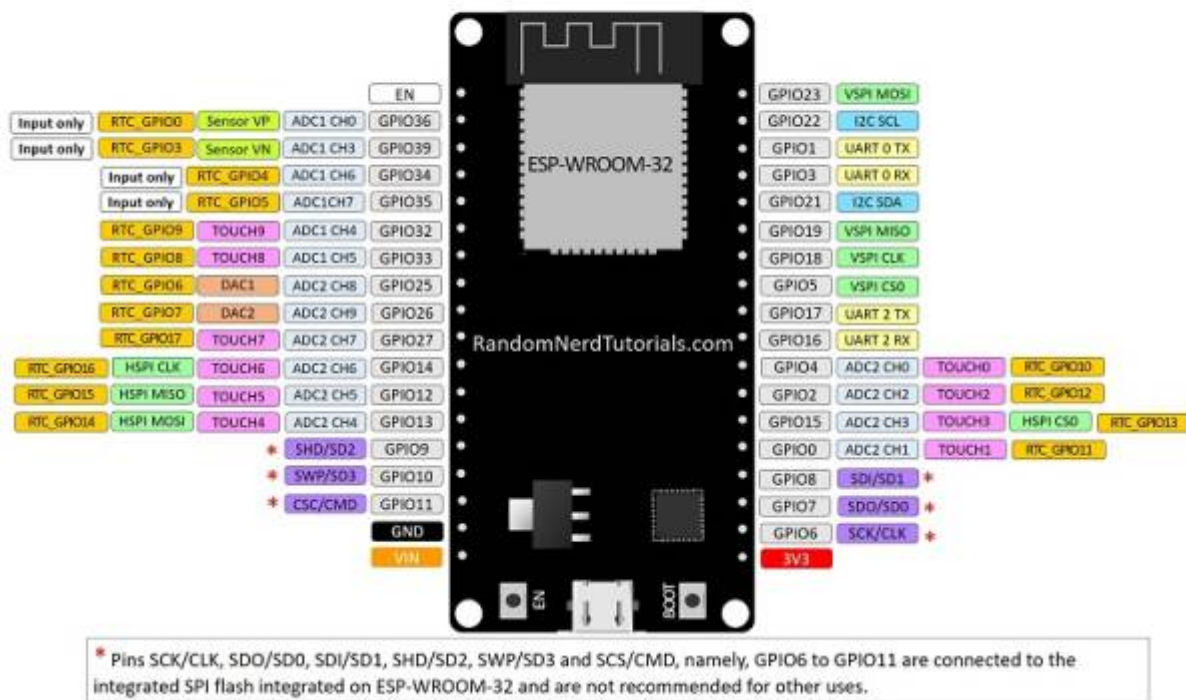
(Disclaimer: Sorry, I can't remember where I got the following two from...).

ESP32 DEVKIT V1 - DOIT



ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



Below is a simple python script to test SSD1306 OLED display which makes use of [this](#) micropython library code.

test.py

```
from machine import Pin, I2C
from ssd1306 import SSD1306_I2C
from time import sleep

def run_this():
    i2c = I2C(-1, scl=Pin(22), sda=Pin(21))
    oled_width = 128
    oled_height = 64
    oled = SSD1306_I2C(oled_width, oled_height, i2c)

    oled.text('Hello, World 1!', 0, 0)
    oled.text('Hello, World 2!', 0, 10)
    oled.text('Hello, World 3!', 0, 20)
    oled.show()
```

Using Arduino IDE:

- select Preferences>Additional Board Manager URLs
- enter https://espressif.github.io/arduino-esp32/package_esp32_index.json
- open Boards Manager and install esp32 (search for it)
- when selecting board for upload, select DOIT ESP32 DEVKIT V1

Note: esp32 flash tool requires python serial module (e.g. `pip install pyserial`)

ESP01S WiFi Module

Actually has an esp8266 controller (can be a standalone... e.g. NodeMCU). I primarily use this as WiFi module for 8051 systems.

[esp01s_info.txt](#)

Upgraded ESP-01S ESP8266 Serial WIFI Wireless Module Wireless Transceiver

- ESP01S ESP8266-01S (stronger antenna signal compared to esp01)
- 1MB flash (from 512M)
- Integrated WEP, TKIP, AES, and WAPI engines. 802.11 b/g/n
- Wake up and transmit packets in 2ms
- standby power consumption of 1.0mW (DTIM3)

Feature:

- Power down leakage current <10uA
- Integrated low power 32-bit CPU could be used as application processor

```
-----  
|ANT.|  
|    |  
|    |  
|8765|  
|1234|  
-----
```

```
1|tx|  
2|ch_pd|1-enable  
3|rst|0-reset  
4|vcc|3.3v  
5|rx|  
6|gpio0|0-flash mode (def), 1-uart mode (prog)  
7|gpio2|gpio  
8|gnd|
```

From:

<https://www.azman.my1matrix.cc/> - Azman @MY1

Permanent link:

<https://www.azman.my1matrix.cc/doku.php?id=notes:devs:board&rev=1777079188>

Last update: **2026/04/25 01:06**

