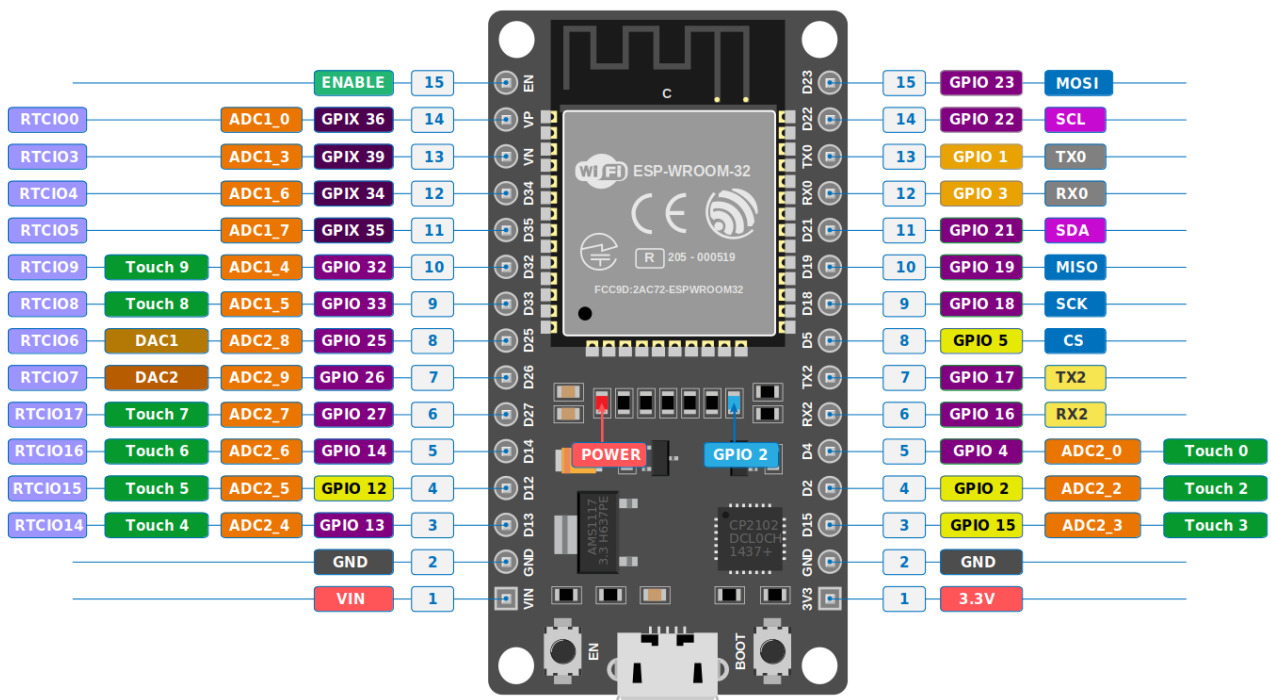


Corso “ESP32 – da Zero a Robot”

Modulo 1 – Dispensa alla Lezione 1

Pinout ESP32

Conoscere il pinout, cioè la funzione di ciascun piedino della scheda ESP32, è fondamentale per poterlo utilizzare e per non danneggiarlo facendo esperimenti.



ATTENZIONE: il pinout può differire da quello mostrato, a seconda del costruttore e/o della versione della scheda. Verificare sempre che i pin corrispondano grazie alle scritte riportate sulla scheda stessa. Per fare un esempio, quella riportata e quelle che utilizzo io hanno i pin VIN e 3,3V dal lato della porta USB, mentre ne ho viste con VIN e 3,3V dal lato opposto.

Nome	GPIO	Si può usare con sicurezza?	Motivo
D0	0	Con attenzione	Deve essere HIGH per il boot e LOW per la programmazione
TX0	1	NO	Tx pin, usato per flashing e debugging
D2	2	Con attenzione	Deve essere LOW e connesso al LED di bordo
RX0	3	NO	Rx pin, usato per flashing e debugging
D4	4	SI	
D5	5	Con attenzione	Deve essere HIGH per il boot
D6	6	NO	Connesso alla memoria Flash
D7	7	NO	Connesso alla memoria Flash
D8	8	NO	Connesso alla memoria Flash
D9	9	NO	Connesso alla memoria Flash
D10	10	NO	Connesso alla memoria Flash
D11	11	NO	Connesso alla memoria Flash
D12	12	Con attenzione	Deve essere LOW per il boot
D13	13	SI	
D14	14	SI	
D15	15	Con attenzione	Deve essere HIGH per il boot, fallisce se LOW
RX2	16	SI	
TX2	17	SI	
D18	18	SI	
D19	19	SI	
D21	21	SI	
D22	22	SI	
D23	23	SI	
D25	25	SI	
D26	26	SI	
D27	27	SI	
D32	32	SI	
D33	33	SI	
D34	34	Con attenzione	GPIO solo input, non configurare come output
D35	35	Con attenzione	GPIO solo input, non configurare come output
VP	36	Con attenzione	GPIO solo input, non configurare come output
VN	39	Con attenzione	GPIO solo input, non configurare come output

MicroPython su ESP32

In tutte le lezioni del Modulo 1 del corso si utilizza MicroPython. Per poter utilizzare questo linguaggio sull'ESP32 è necessario modificare il firmware del dispositivo con un tool come *esptool.py*.

1. Installazione di Python e Strumenti di Flashing

Per caricare il firmware MicroPython sull'ESP32, avrai bisogno di **Python** sul tuo PC (se non è già installato) e dello strumento *esptool.py*.

- **Installa Python:** Scarica e installa l'ultima versione di Python 3 (assicurati di spuntare la casella "**Add Python to PATH**" durante l'installazione su Windows).
- **Installa esptool.py:** Apri il terminale (Prompt dei comandi su Windows, Terminale su macOS/Linux) e digita:

```
pip install esptool
```

2. Scarica il Firmware MicroPython

- Vai al sito ufficiale di MicroPython per l'ESP32:
https://micropython.org/download/ESP32_GENERIC/
- Scarica l'ultima versione del file **firmware binario** (solitamente un file con estensione .bin e nel nome la dicitura generic)

3. Installa l'IDE Thonny

- Scarica e installa **Thonny** dal sito ufficiale (<https://thonny.org/>). Questo IDE è il più semplice da usare per MicroPython perché include già i driver seriali necessari e ha la shell REPL integrata.

4. Connetti l'ESP32

- Collega il tuo ESP32 al PC tramite il cavo **USB-C**.
- **Identifica la Porta Seriale:**
 - **Windows:** Controlla in Gestione Dispositivi (sotto Porte COM e LPT) per trovare la porta COM (es. COM4). Il tuo chip seriale è il **CP2102**.
 - **macOS/Linux:** La porta seriale sarà qualcosa del tipo /dev/tty.usbserial-XXXX o /dev/ttyUSB0.

5. Esegui il Flashing

Apri il terminale del tuo PC e usa esptool.py per caricare il firmware.

- **Cancella la Memoria (Opzionale ma Consigliato)** Per assicurare una memoria pulita, esegui:

```
esptool.py --port [PORTA_COM] erase_flash
```

se Python non è nella path di sistema ("**Add Python to PATH**" del paragrafo 1):

```
python -m esptool --port [PORTA_COM] erase_flash
```

(Sostituisci [PORTA_COM] con la porta che hai identificato, come ad esempio COM4 oppure /dev/ttyUSB0).

- **Carica il Firmware** Esegui il comando per scrivere il file .bin sulla scheda, partendo dall'indirizzo di memoria 0x1000:

```
esptool.py --chip esp32 --port [PORTA_COM]  
write_flash -z 0x1000 [PERCORSO_AL_FILE_BIN]
```

oppure:

```
python -m esptool --port [PORTA] --baud 115200  
write_flash 0 [PERCORSO_AL_FILE_BIN]
```

(Sostituisci [PERCORSO_AL_FILE_BIN] con il percorso completo del file .bin che hai scaricato).

Nota: Durante questa operazione, potresti dover tenere premuto il pulsante **BOOT** (o **FLASH**) sulla scheda ESP32 mentre il programma cerca la connessione, e rilasciarlo una volta che il flashing inizia. Nelle schede più recenti, come quella che ho usato nei video, non è necessario tenere premuto il pulsante.

Configurazione e Test su Thonny

Installato Python, Thonny ed esptool, dopo aver modificato il firmware e riavviato l'ESP32, si può configurare Thonny e procedere col primo test:

1. Configura Thonny

- Apri **Thonny**
- Vai su **Run (Esegui)** → **Select interpreter... (Seleziona interprete...)**
- Nella finestra che si apre:
 - Scegli "**MicroPython (ESP32)**" come interprete.
 - Sotto **Port**, scegli la **Porta Seriale** identificata in precedenza (es. COM4).
- Clicca su "**OK**"

2. Test Interattivo (REPL)

Se il flashing è andato a buon fine, dovresti vedere la **shell REPL** nella parte inferiore di Thonny con il prompt `>>>`.

- Prova a digitare un comando semplice per testare la comunicazione:

```
print('Ciao ESP32!')
```

L'ESP32 dovrebbe rispondere immediatamente con la stringa di testo.

- **Primo Test LED:** Puoi anche provare un comando base per accendere il LED integrato (il pin può variare, ma sui moduli WROOM è spesso il GPIO 2):

```
import machine
led = machine.Pin(2, machine.Pin.OUT)
led.value(1) # Accende il LED
led.value(0) # Spegne il LED
```

Materiali di riferimento e dotazione hardware

La pagina di riferimento a questo corso si trova a questo indirizzo:

<https://www.capirciqualecosa.it/esp32-da-zero-a-robot-presentazione-del-corso>

La pagina di riferimento a questo modulo si trova a questo indirizzo:

<https://www.capirciqualecosa.it/esp32-da-zero-a-robot-modulo-1>

Le lezioni video si trovano anche a questo indirizzo:

<https://bit.ly/ESP32-DaZeroARobot>

Tutto il materiale hardware usato nel corso si può trovare a questo indirizzo:

<https://amzn.to/4ozvITu>



Questo documento è distribuito con licenza [CC-BY](https://creativecommons.org/licenses/by/4.0/)

Utilizzare il link: <https://www.capirciqualecosa.it>

per l'attribuzione di paternità