La scheda **AZ-delivery ESP32 Dev Kit C V2** basata sul modulo ESP32-WROOM

Riferimenti:

- ESP-32 Dev Kit C V2_EN eBook.pdf
- Esp32-wroom-32_datasheet_en.pdf
- esp32_datasheet_en.pdf

Preparare Arduino IDE per ESP32 Dev Kit C V2

Preparare Arduino IDE 1/6

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

Downloads

1. Scaricare ed Installare Arduino versione 1.8

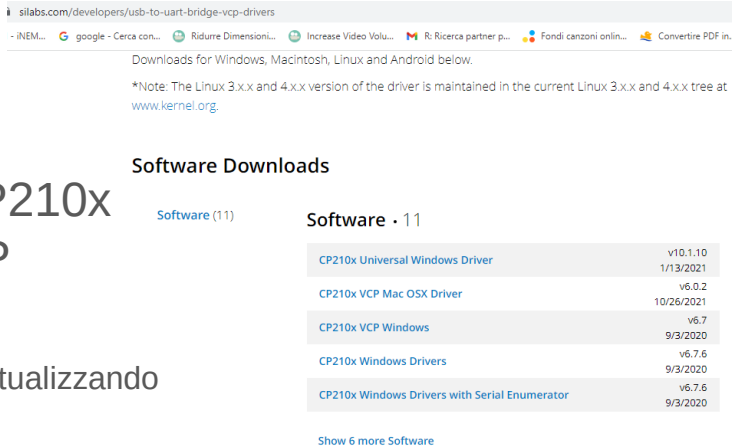
- WINDOWS WIN7 and newer
- Mac OS X 10.10 or newer

NOTA: di seguito si fa riferimento a arduino-1.8.16-windows.exe

https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers

2. Scaricare il driver CP210x Universal Windows Driver o CP210x VCP MAC OSX Driver e UNZIP (cartella)

- Indispensabile per comunicare via USB virtualizzando una COM (porta seriale di tipo UART)



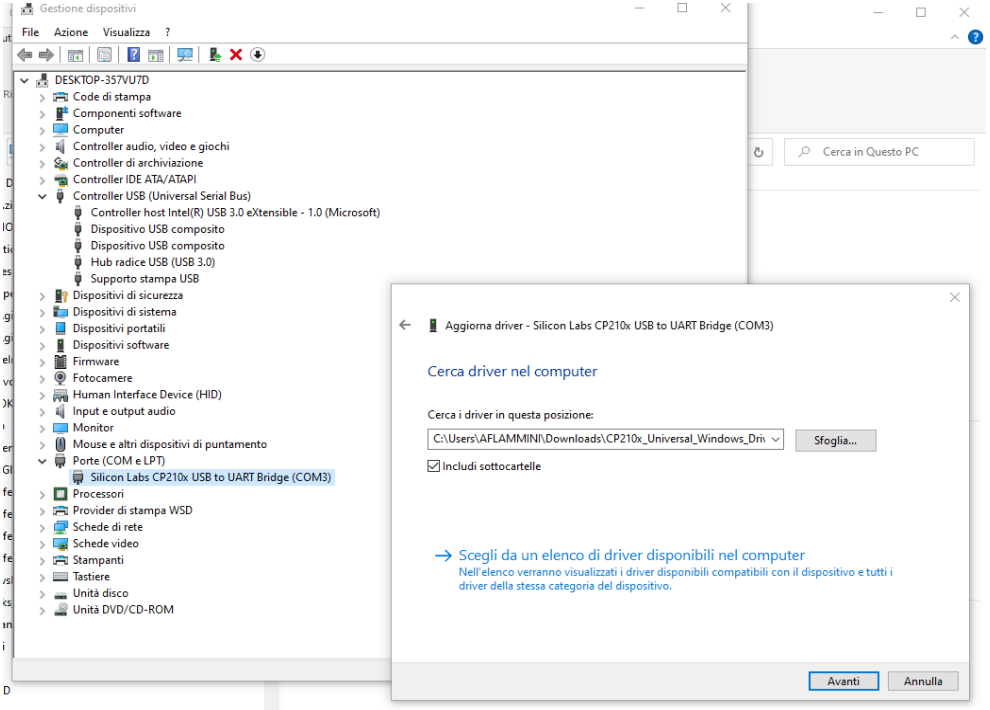
Preparare Arduino IDE 2/6

3. Collegare la scheda ad una porta USB

4. Andare su Gestione dispositivi e verificare su Porte (COM e LPT) che vi sia un driver, anche vecchio, di Silicon Labs CP210x USB to UART Bridge e segnarsi il numero della COM

5. Cliccare col destro e aggiornare il driver cercando nel PC la cartella dezippata

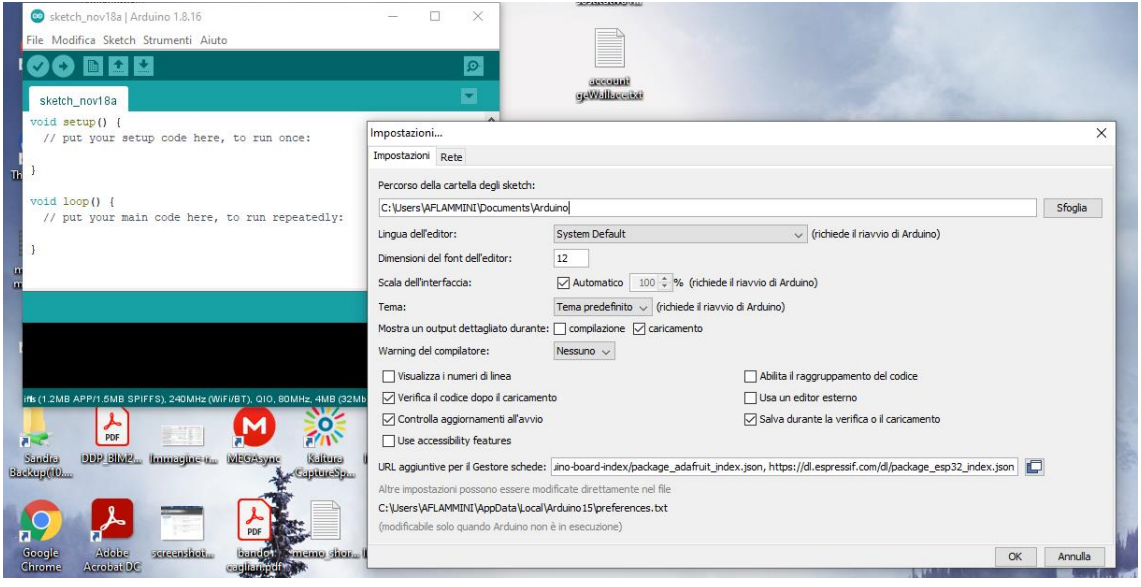
6. Riavviare il PC



Preparare Arduino IDE 3/6

7. Avviare Arduino

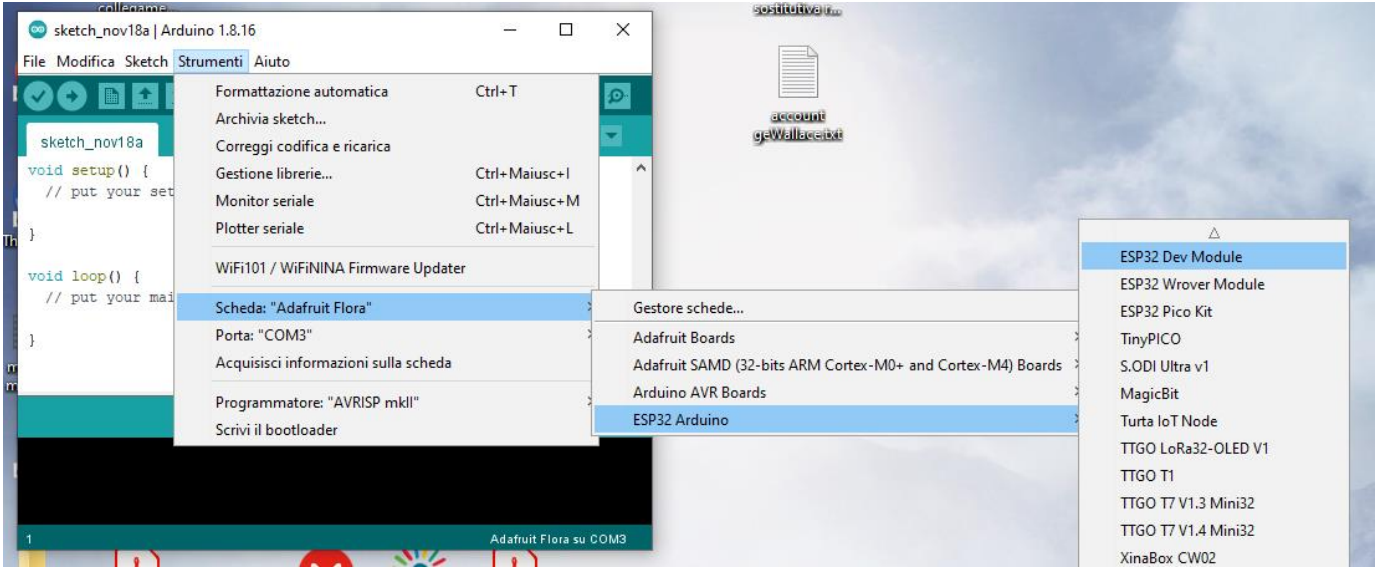
8. Andare su File -> Impostazioni (preferences) e aggiungere a URL aggiuntive per il Gestore schede https://dl.espressif.com/dl/package_esp32_index.json quindi chiudere e riavviare Arduino



Preparare Arduino IDE 4/6

9. Andare su Strumenti (Tools) -> Scheda (Board) -> Gestore schede e individuare e installare esp32

10. Andare su Strumenti (Tools) -> Scheda (Board) -> e selezionare ESP32 Dev Module e verificare che "porta" sia la stessa individuata nel punto 4.



Gli sketch Arduino

I programmi in Arduino si chiamano "sketch" e vengono memorizzati normalmente in Documenti/Arduino all'interno di cartelle che hanno lo stesso nome del file.ino (il programma)

Gli sketch sono composti da due parti:

- setup() eseguita una sola volta al reset
- loop() eseguita in modo ciclico asincrono

Nota: prima del setup è possibile includere librerie (#include <mylib.h>)definire costanti (#define <label> <value>), variabili (es. int Var1, Var2;), dichiarare sottoprogrammi (void myprogr() {)

Regole generali di sintassi:

- Ogni istruzione deve essere seguita da ;
- I commenti sono identificati da //
- Le funzioni (vedi https://www.arduino.cc/reference/en/), se scritte correttamente, sono in rosso

Arduino e lo strumento "Monitor seriale"

Quando realizzo un programma e lo carico sulla scheda, posso verificare il funzionamento a livello di segnali, ma non sempre è sufficiente (es. Come faccio a sapere quanto vale una variabile?)

Arduino ha previsto lo strumento "Monitor seriale" che permette di interagire con il programma mentre è in esecuzione.

In particolare, attivando lo strumento Monitor seriale, si apre una finestra mediante la quale è possibile inviare e ricevere dati.

https://www.arduino.cc/reference/en/language/functions/communication/serial/

Naturalmente il processore impiega tempo ad inviare i dati su seriale e quindi nei programmi definitivi si tende ad eliminare questa funzione diagnostica.

La seriale deve avere lo stesso baud rate (es. Serial.begin(9600)

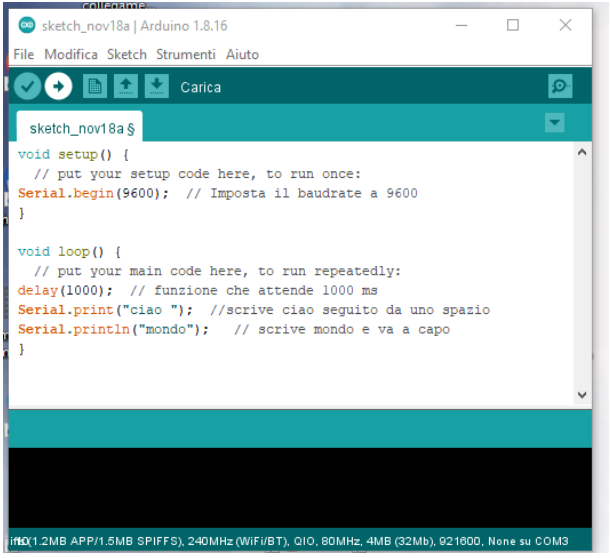
Per scrivere i dati sulla finestra si usa Serial.print() o Serial.println

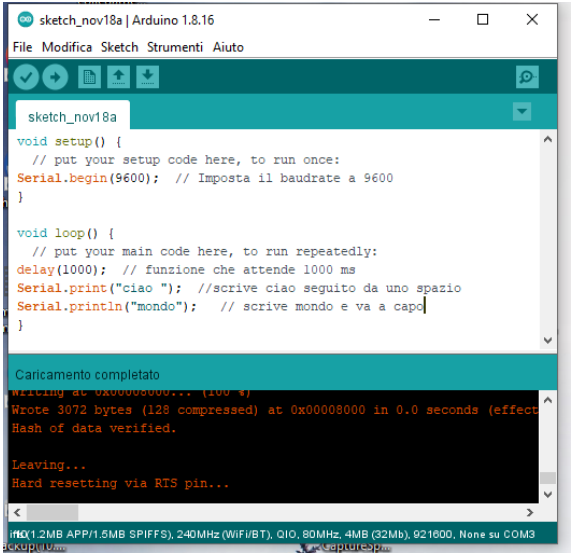
Preparare Arduino IDE 5/6

11. Realizzare un programma che scriva "ciao mondo" sul monitor seriale ad ogni secondo e caricarlo su scheda

12. Andare su Strumenti -> Monitor seriale e si apre la finestra; verificare quale baud rate usa la virtual COM (es. 9600)

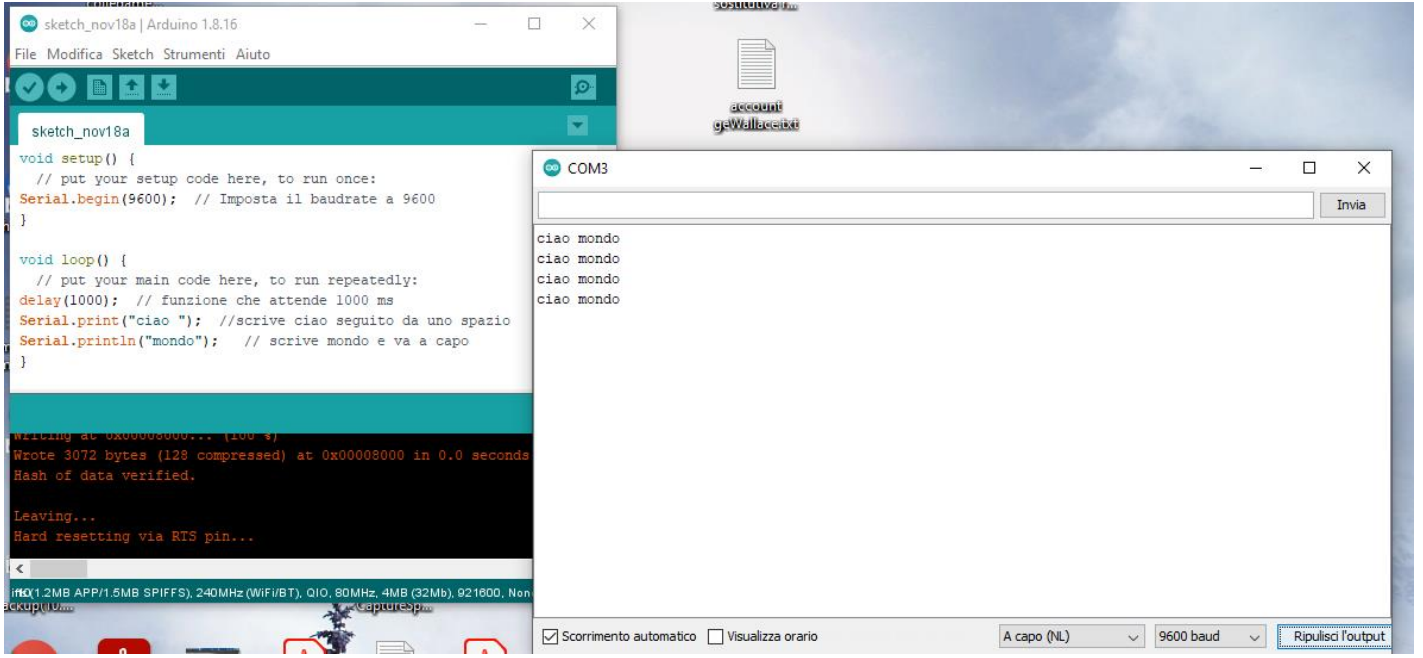
13. Scrivere il programma e caricarlo su scheda





Preparare Arduino IDE 6/6

14. Premere il pulsantino di reset (RST) su scheda a sinistra dell'USB. Se tutto è andato bene questo è il risultato



Riassumendo

1. Installare Arduino dal sito ufficiale e collegare la scheda al PC
2. Installare i driver del dispositivo CP210x (SerialCOM) e riavviare il PC
3. Avviare Arduino e impostare il nuovo set di schede, quindi usare lo strumento gestore schede per installarle e selezionare la scheda ESP32 Dev Module
4. Controllare la porta COM ed adeguarla a quanto visto su gestione dispositivi del PC
5. Scrivere un file di esempio basato sul monitor, caricarlo e verificarne il funzionamento

Possibili errori e relative soluzioni

- **Nel caricare lo sketch si incanta su COM e fallisce**
  - ✓ Premere 2-3 volte il pulsantino di reset (RST) su scheda
  - ✓ Premere il pulsante di Boot durante il caricamento
  - ✓ Verificare in "Strumenti" che ci sia la COM corretta
  - ✓ Staccare l'USB e riattaccare eventualmente cambiando porta USB
  - ✓ Andare da Windows in "Gestione dispositivi" -> tasto destro su Dispositivo Silicon Labs -> Disinstalla dispositivo. Riavviare PC e verificare porta COM su "Gestione dispositivi" e, se COM cambiata, aggiornare su Arduino IDE

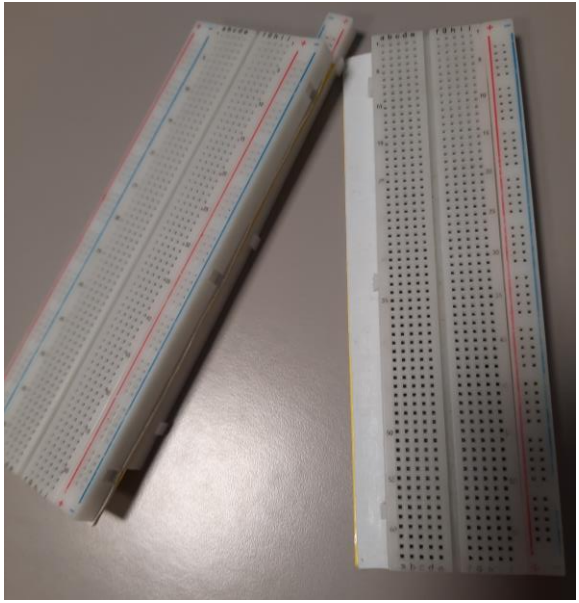
- **Windows da dispositivo USB non riconosciuto**
  - ✓ Verificare che la scheda non abbia connessioni esterne "critiche" (Ad esempio gnd, 3V, USB, Reset)
  - ✓ Se in Strumenti non appare come scheda "ESP32 Dev Module", allora attivare gestore schede, disinstallare la scheda esp32, riavviare Arduino e reinstallarle come da punti 7-8-9-10
  - ✓ Reinstallare i drivers per windows come da punti 2-3-4-5-6

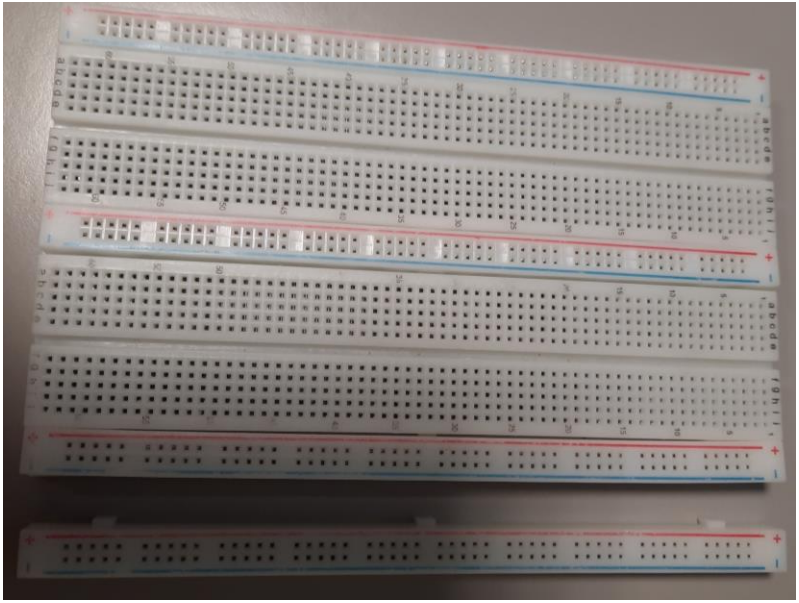
Preparare la basetta per ESP32 Dev Kit C V2

Preparare la basetta

La scheda ESP32 Dev Kit C V2 è troppo larga per la nostra basetta quindi dobbiamo realizzare una composizione di due basette

1. Si liberi la basetta da tutti i componenti, incluso l'alimentatore e l'oscillatore
2. Si scolleghi delicatamente la barra verticale sinistra dalla basetta e, scollando leggermente il fondo, si unisca alla seconda basetta ricevuta





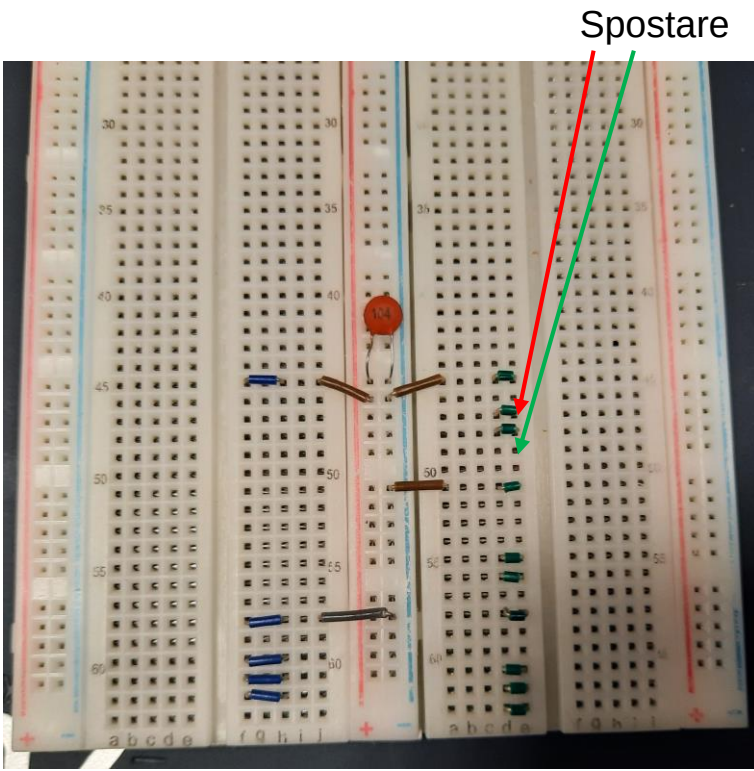
Preparare la basetta

La scheda ESP32 Dev Kit C V2 alimenta la basetta con una tensione a 3,3V; inoltre alcuni pin sono già utilizzati da altre funzioni e non è il caso che siano utilizzati in esperienze didattiche. Facciamo alcuni collegamenti:

La scheda ESP32 Dev Kit C V2 alimenta la basetta con una tensione a 3,3V; inoltre alcuni pin sono già utilizzati da altre funzioni e non sono utilizzati in esperienze didattiche. Fare alcuni collegamenti:

Sinistra: 45f-45h, 45l-46barra+, 58f-58h, 58l-58barra-, 60f-60h, 61f-61h, 62f-62h

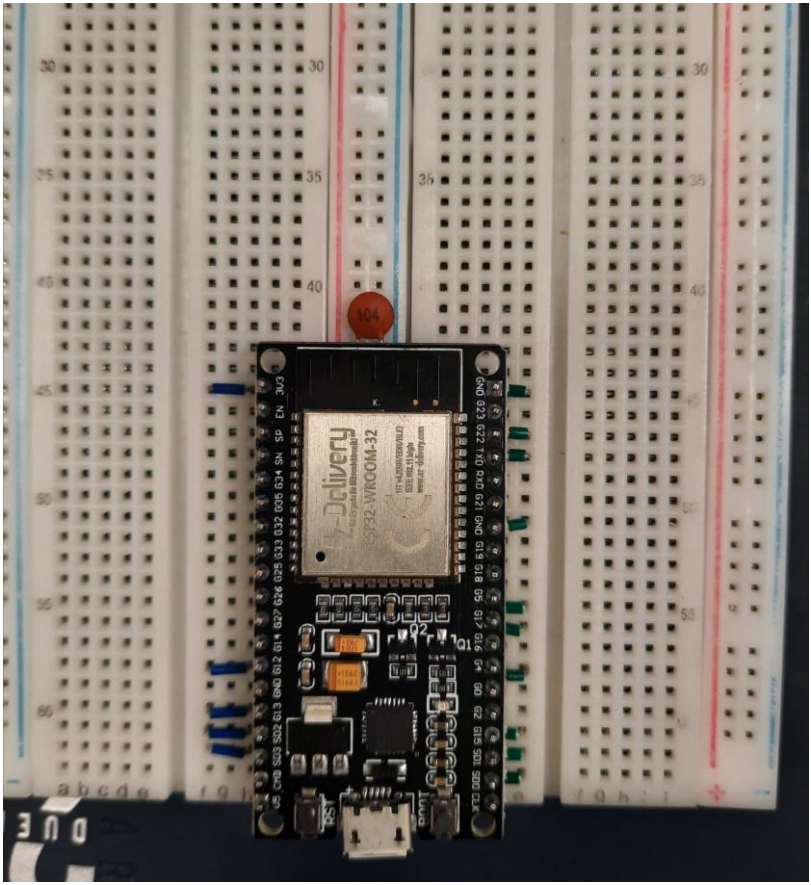
Destra: 45d-45e, 45a-46barra-, 47d-47e (rimuovere, GPIO22 è usabile), 48d-48e, 49d-49e (GPIO3 non è usabile), 51d-51e, 51a-51barra-, 55d-55e, 56d-56e, 58d-58e, 61d-61e, 62d-62e, 63d-63e, Aggiungere un condensatore da 100nF tra 45barra+ e 45barra-



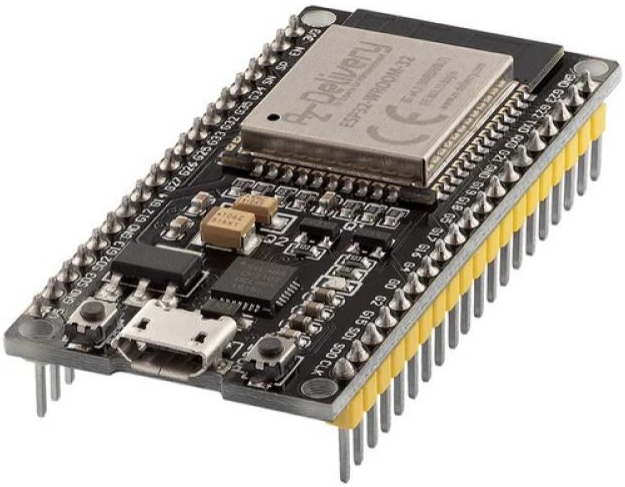
Preparare la basetta

Inserire la scheda ESP32 Dev Kit C V2 a sinistra sulla fila i da 45 a 63 e a destra sulla fila c da 45 a 63.

Inserire il cavo USB e verificare che il programma "ciao mondo" di cui al punto 11 funzioni correttamente.

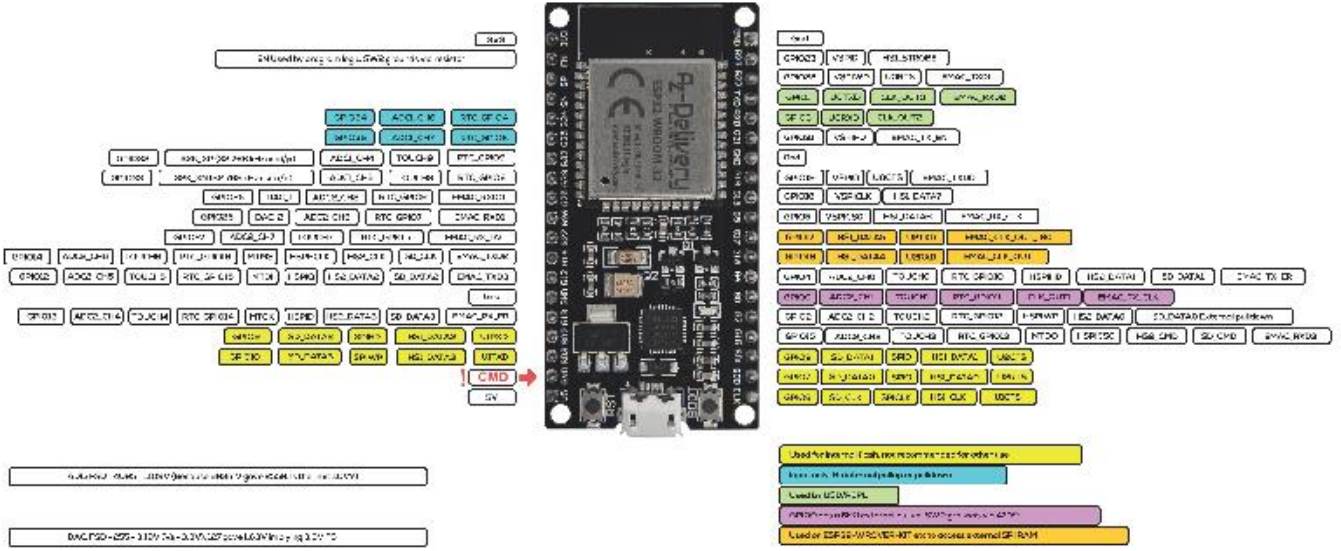


Caratteristiche tecniche



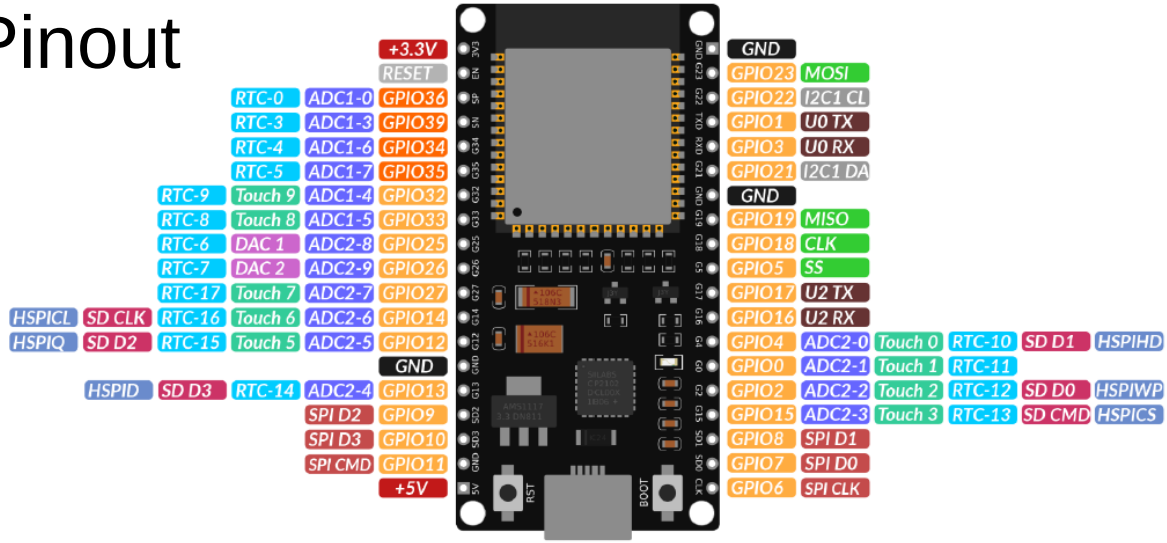
- Measures 2.2" x 1.1" x 0.5" (56mm x 28mm x 13mm)
- SoC ESP32-WROOM-32/QFN-38 based on CPU ESP32-D0WDQ6 (ATTENZIONE! Altre schede sono basate su questo SoC ma hanno pinout diverso)
- 2 uP (dual core) Xtensa 32-bit LX6 @ 80-240MHz with 3.3V logic I/O
- 3.3V regulator from 5V (USB) with 500mA peak current output
- 512kB SRAM, 4MB SPI Flash
- 19+19 pins, 17 I/O available
- 12-bit ADC, 8-bit DAC
- Hardware Serial: SPI, I2C, I2S, CAN, UART
- RTC GPIO che possono essere usati nei modi a basso consumo
- GPIO, a parte GPIO34-39
- Integrated 802.11b WiFi, Bluetooth and BLE (Class 1-2-3) communication
- Reset button, Boot button

ESP-32 NodeMCU Developmentboard
Pinout Diagram



Value	Expected	Actual	Error %
10	0.13	0.21	2.4
20	0.26	0.33	2.1
127	1.64	1.63	-0.3
200	2.58	2.53	-1.5
240	3.11	3.01	-3
255	3.3	3.19	-3.3

Pinout



- *Digital In/Out ports (all support PWM)*
- *Digital Input ports*
- *Analog Input 12 bits, 0 to 3.3V*
- *Analog Output 8 bits, 0 - 3.3V*
- *Capacitive Touch Sensor ports*
- *I/O -pins from RTC ultra low power processor, usable in deep sleep mode*
- *SD card interface*
- *SPI bus for Flash-memory, do not use*

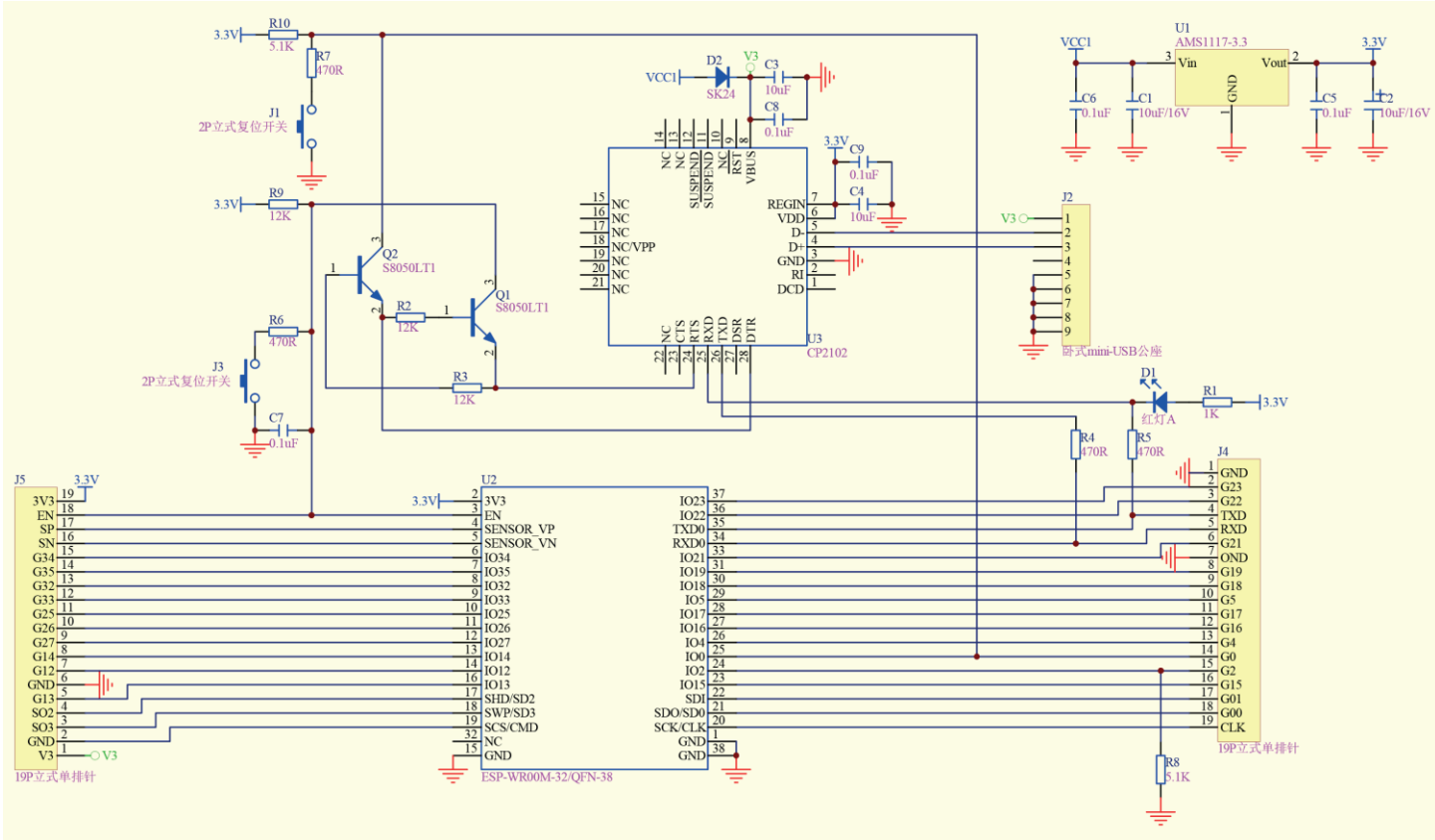
The following pins show the default assignment. All these signals can be changed to any In/Out port. This applies also to UART0 and UART1, which cannot be accessed in the default assignment.

- *I2C bus (Wire)*
- *VSPI bus*
- *Serial interfaces*
- *HSPI bus*

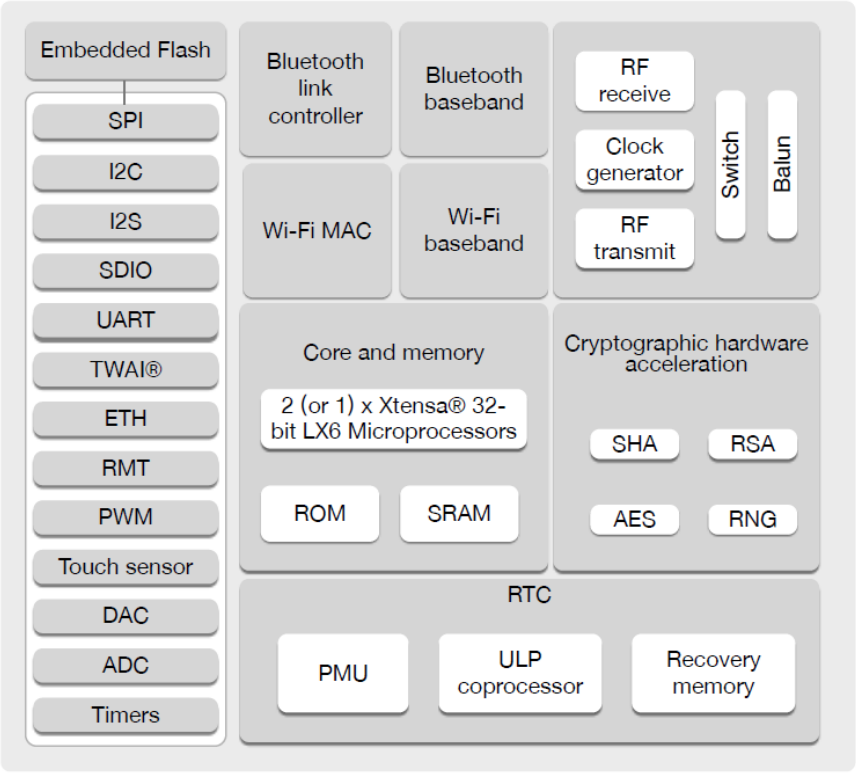
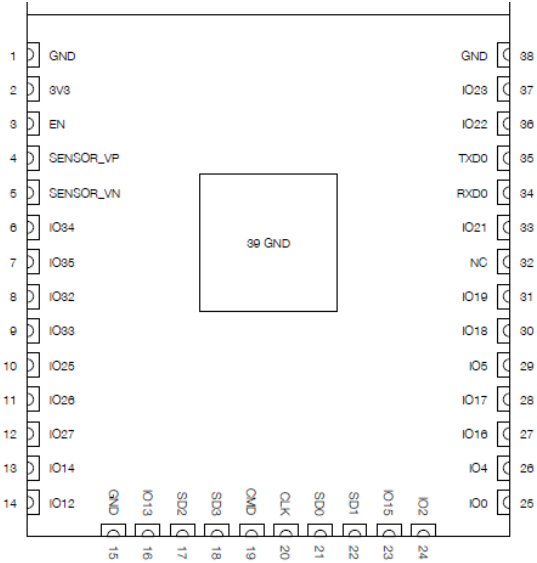
Schemi funzionali scheda

La scheda ESP32 Dev Kit C V2 si basa sul SoC ESP-WROOM-32/QFN-38

Attenzione! G2 può inibire il caricamento di programmi, usare solo per OUTled



Pinout ESP-WROOM-32/QFN-38

La scheda ESP32 Dev Kit C V2 si basa sul SoC ESP-WROOM-32/QFN-38

Vedi esp32_datasheet_en.pdf per la descrizione delle risorse