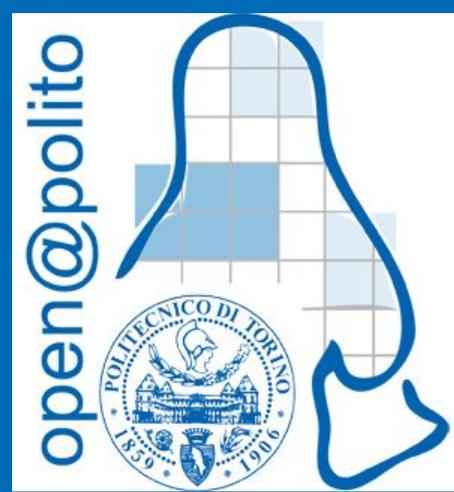


OPEN DOMOTICA

Gestire casa con hardware e
software Open Source



weeeopen

NETSTUDENT

whoami

Francesco Tucci

Lavoro nei datacenter di una **grande** multinazionale.

Uso tutti i sistemi operativi.

Lavoro nell'IT da molto tempo.

(sono sopravvissuto al millennium bug, all'avvento dell'Euro e al grande blackout del 2003)



WARNING!

Questa presentazione potrebbe portare a farvi spendere grandi quantità di **denaro** e molte **ore** al posto di dormire



Ma non potrete dare la colpa a me, ok?

QUIZ INTRODUTTIVO 1

Lo hai mai visto? Lo hai usato?



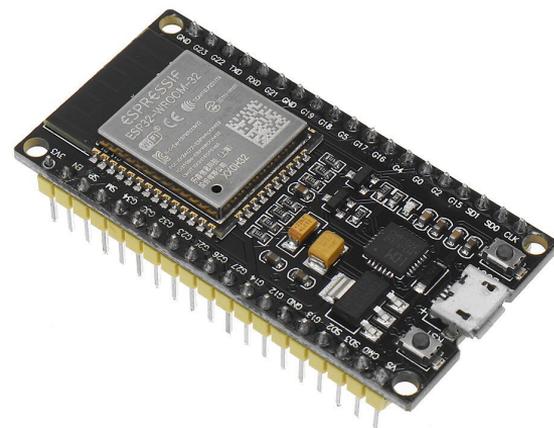
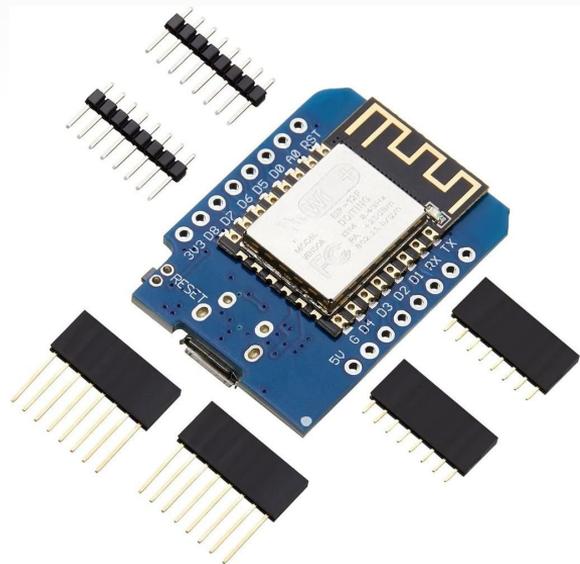
QUIZ INTRODUTTIVO 2

Lo hai mai visto? Lo hai usato?



QUIZ INTRODUTTIVO 3

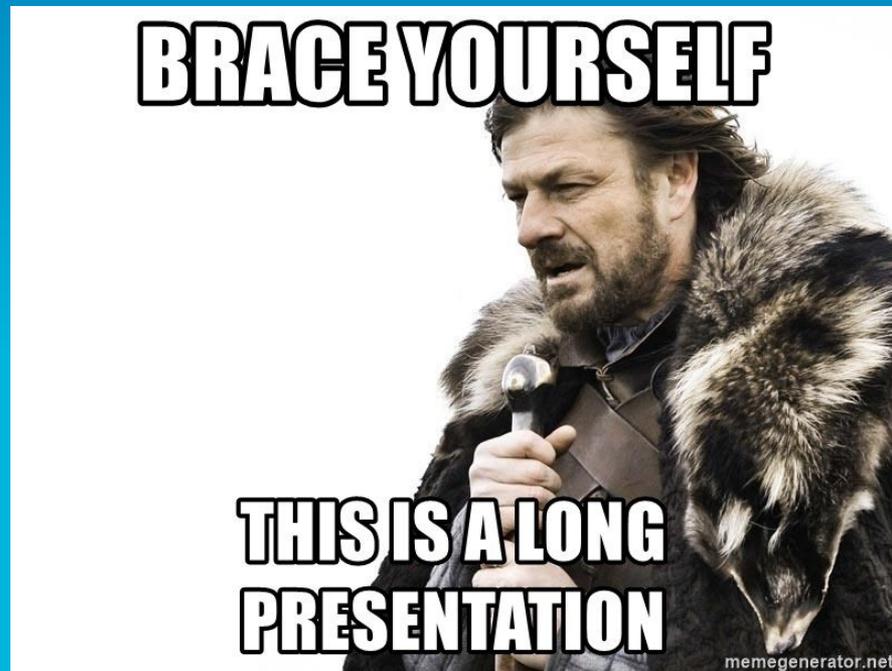
Le hai mai viste? Le hai usate?





Oggi parliamo di:

1. Raspberry Pi
2. Sistemi operativi
3. Microcontrollori
4. Sensori
5. MQTT

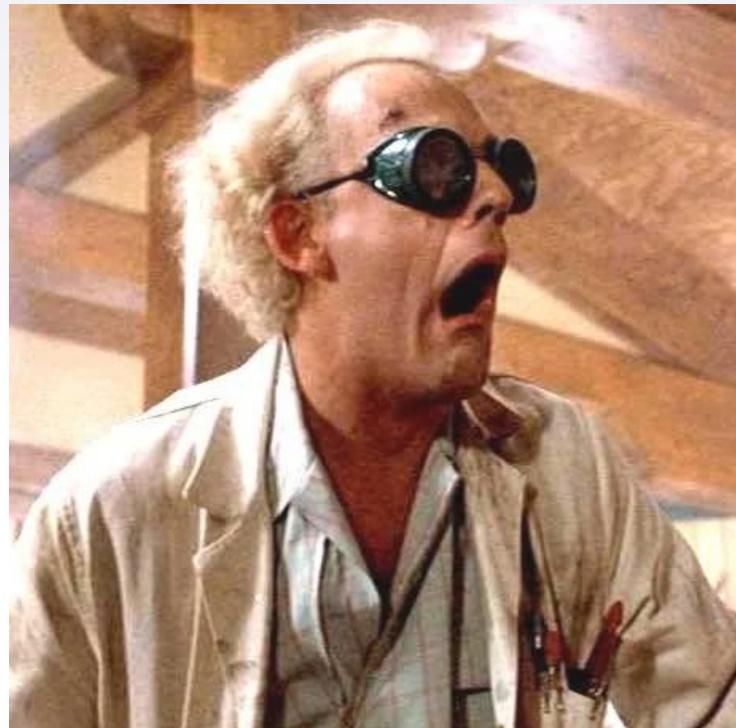


Avete portato il sacco a pelo?

1

Raspberry Pi

La miglior invenzione dal 2012



lo avevo 34 anni, voi?

Perché è interessante?

- ▶ È piccolo
- ▶ È completo (SOC)
- ▶ Fa tutto quello che serve
- ▶ Consuma poco
- ▶ È espandibile
- ▶ Ha un GPIO su cui si installano cose fantastiche
- ▶ È molto ben supportato



► Perché è interessante?

- ▶ **È piccolo**
- ▶ È completo (SOC)
- ▶ Fa tutto quello che serve
- ▶ Consuma poco
- ▶ È espandibile
- ▶ Ha un GPIO
- ▶ È molto ben supportato

È grande come una carta di credito, circa.

Ma ne esiste una versione ancora più piccola, che manca di USB e di Ethernet, ma è comodo per fare cose a bassissima potenza.

Si chiama **Raspberry Pi Zero W** (quello senza la W non ha la WiFi, evitatelo)

► Perché è interessante?

- ▶ È piccolo
- ▶ **È completo (SOC)**
- ▶ Fa tutto quello che serve
- ▶ Consuma poco
- ▶ È espandibile
- ▶ Ha un GPIO
- ▶ È molto ben supportato

System On a Chip, in un unico integrato c'è tutto quello che serve per far funzionare il piccolo computer, quello che di solito è sparso in vari integrati all'interno di una scheda madre. Questo ha aiutato a renderlo piccolo e soprattutto economico (all'inizio).

► Perché è interessante?

- ▶ È piccolo
- ▶ È completo (SOC)
- ▶ **Fa tutto quello che serve**
- ▶ Consuma poco
- ▶ È espandibile
- ▶ Ha un GPIO
- ▶ È molto ben supportato

È di fatto un computer, ne ha tutte le componenti e assolve a tutti i compiti, con la connettività di base e la capacità di calcolo. Dalla prima alla quarta versione le prestazioni sono aumentate, ma l'idea di base non è cambiata.

Supporta i sistemi operativi standard e permette una facile interazione con i GPIO (più facile di Arduino)

► Perché è interessante?

- ▶ È piccolo
- ▶ È completo (SOC)
- ▶ Fa tutto quello che serve
- ▶ **Consuma poco**
- ▶ È espandibile
- ▶ Ha un GPIO
- ▶ È molto ben supportato

Pochi componenti, prestazioni non all'ultimo grido, funziona anche abbandonato in un angolo senza monitor, mouse e tastiera, consuma davvero poco (no, se ci mettete 4 dischi USB consuma molto, *solo la versione 4B*).

Scarico: **2,7W. 23kWh/y**

A pieno carico: **6,4W. 56kWh/y**

Spesa tra **7 e 17€/y** in bolletta

► Perché è interessante?

- ▶ È piccolo
- ▶ È completo (SOC)
- ▶ Fa tutto quello che serve
- ▶ Consuma poco
- ▶ **È espandibile**
- ▶ Ha un GPIO
- ▶ È molto ben supportato

Con 4 porte USB e una ethernet ci potete fare un po' quello che volete. Dalla versione 4 le porte sono USB3 e la eth è davvero Gb, non su USB come le precedenti.

Ha anche un'interfaccia dedicata per un display e una per una camera, le ultime cam sono eccezionali.

Ma il meglio viene dopo

Perché è interessante?

- ▶ È piccolo
- ▶ È completo (SOC)
- ▶ Fa tutto quello che serve
- ▶ Consuma poco
- ▶ È espandibile
- ▶ **Ha un GPIO**
- ▶ È molto ben supportato

General Purpose Input Output.

È una serie di PIN digitali che permettono di interfacciare un milione di milioni di dispositivi via seriale (sì, si usa ancora) o con una specie di contatti puliti. Potete leggere dei valori da sensori o dispositivi (pulsanti) o comandarne altri (apri porta, accendi luce, attiva motore, ...)

GPIO 2	3	GPIO 10	19
GPIO 3	4	GPIO 11	20
GPIO 4	5	GPIO 12	21
GPIO 5	6	GPIO 13	22
GPIO 6	7	GPIO 14	23
GPIO 7	8	GPIO 15	24
GPIO 8	9	GPIO 16	25
GPIO 9	10	GPIO 17	26
GPIO 10	11	GPIO 18	27
GPIO 11	12	GPIO 19	28
GPIO 12	13	GPIO 20	29
GPIO 13	14	GPIO 21	30
GPIO 14	15	GPIO 22	31
GPIO 15	16	GPIO 23	32
GPIO 16	17	GPIO 24	33
GPIO 17	18	GPIO 25	34
GPIO 18	19	GPIO 26	35
GPIO 19	20	GPIO 27	36
GPIO 20	21	GPIO 28	37
GPIO 21	22	GPIO 29	38
GPIO 22	23	GPIO 30	39
GPIO 23	24	GPIO 31	40
GPIO 24	25	GPIO 32	
GPIO 25	26	GPIO 33	
GPIO 26	27	GPIO 34	
GPIO 27	28	GPIO 35	
GPIO 28	29	GPIO 36	
GPIO 29	30	GPIO 37	
GPIO 30	31	GPIO 38	
GPIO 31	32	GPIO 39	
GPIO 32	33	GPIO 40	
GPIO 33	34		
GPIO 34	35		
GPIO 35	36		
GPIO 36	37		
GPIO 37	38		
GPIO 38	39		
GPIO 39	40		
GPIO 40			

► Perché è interessante?

- ▶ È piccolo
- ▶ È completo (SOC)
- ▶ Fa tutto quello che serve
- ▶ Consuma poco
- ▶ È espandibile
- ▶ Ha un GPIO
- ▶ **È molto ben supportato**

La prima scheda uscita nel 2012 la si può usare con l'ultima versione di Raspberry OS. Ho provato altre schede concorrenti che sono state abbandonate con Linux vecchi e mai più aggiornati. Attenti a cosa comprate.

▶ Ma c'è qualcosa di negativo?

- ▶ Il sistema sta su una microSD
- ▶ Al momento c'è poca disponibilità

▶ Ma c'è qualcosa di negativo?

- ▶ **Il sistema sta su una microSD**
- ▶ Al momento c'è poca disponibilità

Le memorie flash si bruciano. Fate sempre un backup e compratele di ottima qualità. O cercate di fare boot da un disco SSD collegato a una USB (dalla versione 4 è più facile)

▶ Ma c'è qualcosa di negativo?

- ▶ Il sistema sta su una microSD
- ▶ **Al momento c'è poca disponibilità**

I prezzi sono alle stelle, non è il momento buono per comprarli. Si dovrà aspettare ancora un po'. Comprare un miniPC costa di meno.

Per sapere dove c'è disponibilità si può usare RPi Locator:

<https://rpilocator.com/?instock>

La lista della spesa

Scheda Raspberry Pi

Il computer, senza di questa non si fa niente. Meglio se versione 4 con almeno 1GB di RAM.

Se la Zero, almeno Zero W

Memoria

MicroSD di buona qualità, con adattatore per poterla mettere nel computer per scriverci il sistema operativo. Almeno 16GB

Alimentatore

Scegliere quello ufficiale o uno con potenza equivalente (5V 3A)

Fino al 3: Micro USB

Dal 4: USB-C

Cavo Ethernet (opzionale)

Per collegarlo alla rete e raggiungerlo via SSH la prima volta.

Vi serve anche una porta libera sullo switch/router e sapere che IP ha preso

Funziona anche via WiFi dal primo avvio

Case (quello originale)

Fa figo, non impegna e prende meno polvere.

Attenzione che a seconda delle versioni le schede cambiano.

Cavo video, tastiera, mouse

Se lo volete configurare come se fosse un PC normale.

Via SSH fa tutto più nerd

2

Installazione

Un computer ha bisogno di un sistema operativo



Casa madre o più leggero?

Raspberry OS

- Fatto apposta per RPi
- Facile per tutti
- Sempre aggiornato
- Pesantuccio

DietPi

- Leggero
- Senza fronzoli
- Installi quel che ti serve
- Backup di serie
- Pacchetti pronti da installare



Raspberry Pi



DietPi

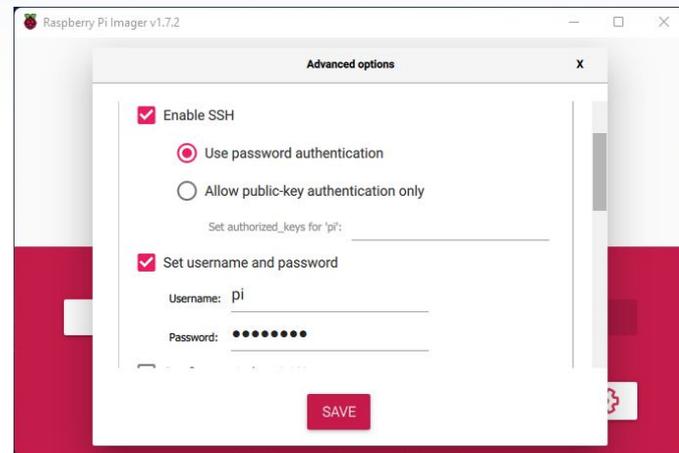
#LightweightJustice for your SBC

Scrivere il sistema sulla microSD

Per **Raspberry OS**

- Scaricate il tool Raspberry Pi Imager (<https://www.raspberrypi.com/software/>)
- Inserite la microSD nel computer
- Avviate il SW
- Nelle opzioni abilitate SSH

Si preoccupa lui di scaricare l'immagine del sistema e scriverla sulla scheda di memoria



► Scrivere il sistema sulla microSD

Per **Diet Pi** *io vi consiglio questo, alla grande!*

- Scaricate Balena Etcher (<https://www.balena.io/etcher>)
- Scaricate la ISO di DietPi (<https://dietpi.com/#download>)
- Inserite la microSD nel computer
- Scrivete la ISO nella memoria con Etcher



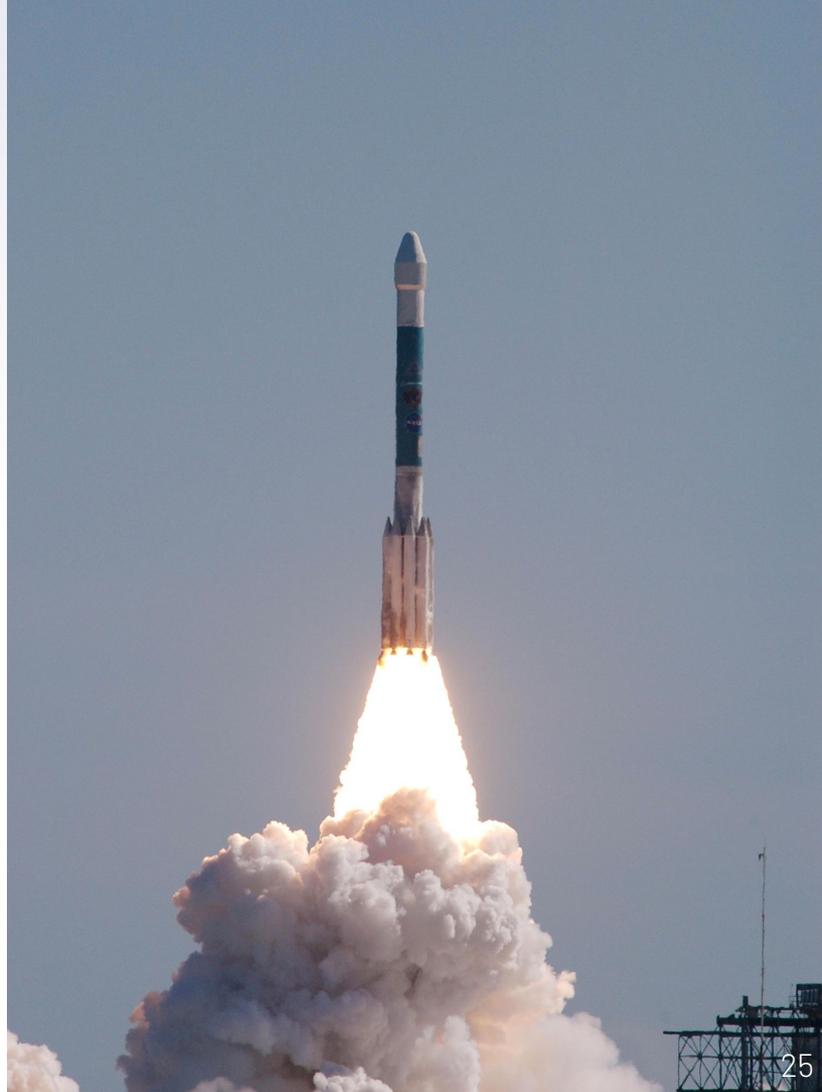
▶ Boot!

- Si mette la microSD nello slot
- Si collega il cavo di rete (meglio eth per iniziare)
- Si dà alimentazione (non c'è interruttore)
- Si guarda sul router che IP ha preso

```
$ ssh pi@192.168.88.25
```



```
$ ssh root@192.168.88.25
```



E cosa ci faccio?

DVR

Server VPN
(PI-VPN)

Desktop
economico

Filtro pubblicità
(PI-Hole)

BOT Telegram

Domotica
(Home Assistant)

Retrogaming

Docker server
(con 4-8GB di RAM)

Sensoristica

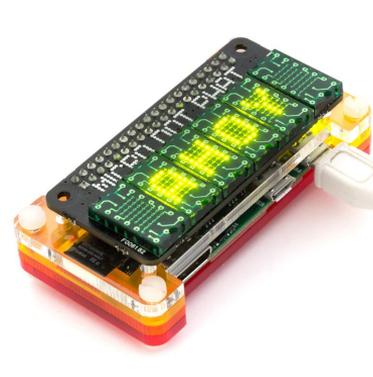
Display

Diffusione audio
(Volumio)

E ancora molto
altro...



Ci sono espansioni?



3

Microcontrollori

Arduino e schede ESP8266 e ESP32



Qui ci si sporca le mani

► Le differenze

Raspberry

Ha un sistema operativo

Lo raggiungo via SSH al volo
mentre sta lavorando

Consuma abbastanza

Il GPIO è solo digitale

Microcontrollore

NON ha un sistema operativo

Lo programmo e fa quello che deve fino
a che non lo riprogrammo

Può arrivare a consumare μA

I PIN sono anche analogici

► Come funziona

Mi serve un IDE, solitamente Arduino IDE

Devo saper programmare in un certo linguaggio, tipo il C (o Micro Python)

Mi servono le librerie della scheda che ho comprato (l'IDE deve compilare)

Devo sapere cosa può fare la scheda che ho in mano (datasheet)

Devo conoscere i sensori che uso e le loro librerie

Faccio il circuito, scrivo il codice, lo carico e spero che vada

L'IDE

È un semplice editor di linguaggi e riesce a scrivere, via USB/seriale sulle schede che colleghiamo al PC, c'è anche una versione Cloud che permette di lavorare ovunque siamo.

Ogni scheda ha bisogno di una sua libreria per insegnare all'IDE come deve compilare il codice da caricare nella (poca) memoria del dispositivo

Ogni pezzo di codice si chiama "sketch" e ha estensione ".ino"

<https://www.arduino.cc/en/software> (si usa anche per le ESP)

Il linguaggio

Senza display non si può scrivere Hello world, ma si può far lampeggiare un LED

```
// the setup function runs once when you press reset or power the board

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Le librerie

La libreria ha funzioni già scritte che permettono di fare chiamate hardware su componenti con comandi facili, senza dover gestire la comunicazione bit per bit

Ogni sensore, scheda di espansione, componente, ha le sue librerie.

Questo ci semplifica il lavoro.

Per leggere i dati da un sensore noi gli diciamo *“usa la sua libreria e vai a leggere dall’indirizzo seriale 0x45 temperatura e umidità, mettili nelle variabili TEMP e HUM”* tutto con un solo comando.

Quando comprate un componente, **PRIMA**, guardate se ha le librerie

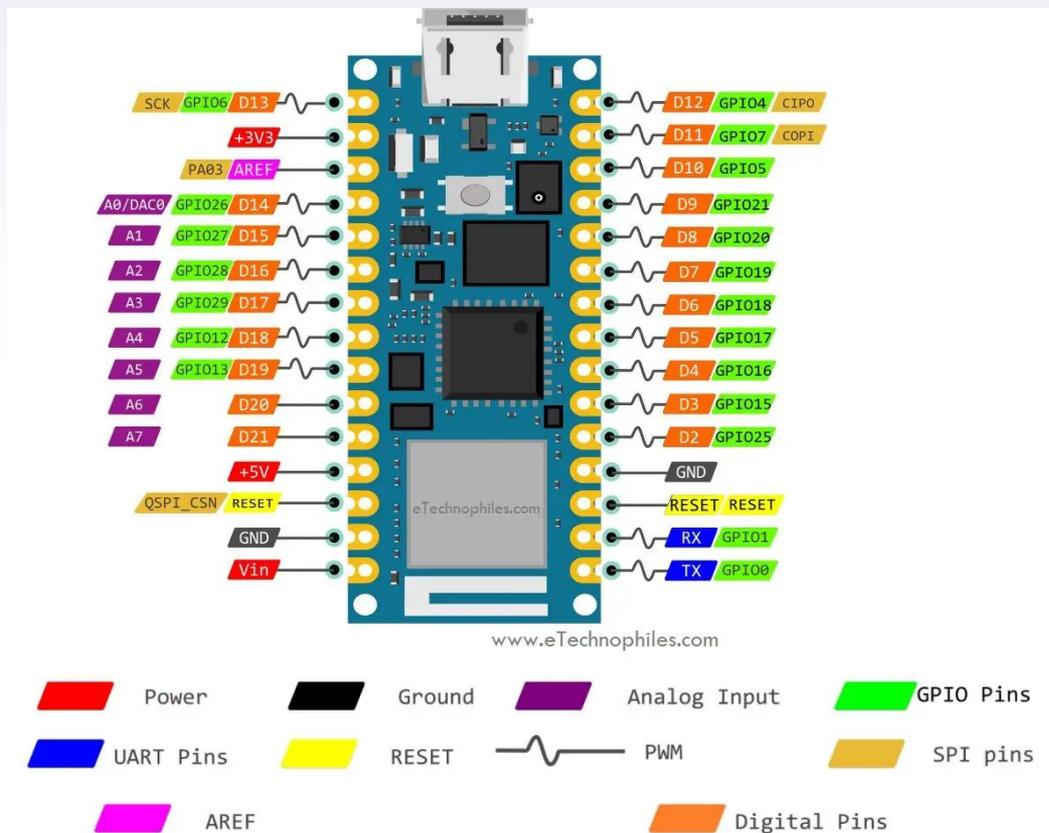
La scheda

Il microcontrollore DEVE comunicare con l'esterno (sensori, bottoni, lettori RFID, motori, ...).

Dobbiamo sapere con quali PIN e come.

Si deve studiare

Arduino Nano RP2040 Connect



3

Sensori

Interagire con il mondo reale

80 Degrees

Fahrenheit



Celsius



Kelvin



► Sensori e altro

Ogni sensore va connesso con fili e dispositivi a due o più PIN al microcontrollore

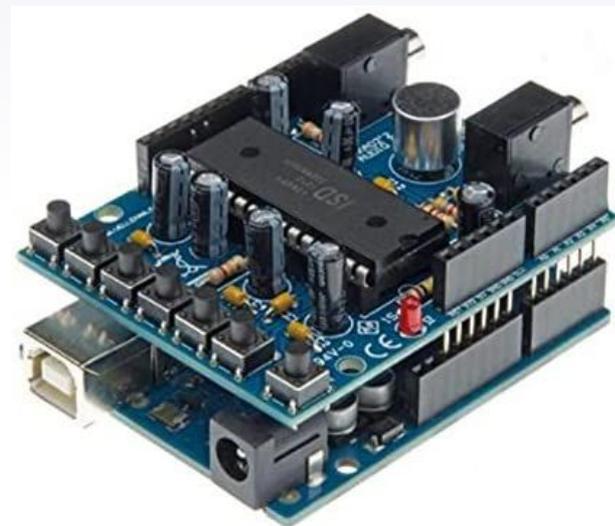
Servono datasheet e librerie compatibili col microcontrollore (da verificare PRIMA di comprarlo)

Lavorare con i microcontrollori, sensori e attuatori è maledettamente divertente, spesso potrebbe portare a frustrazione, voi non arrendetevi.

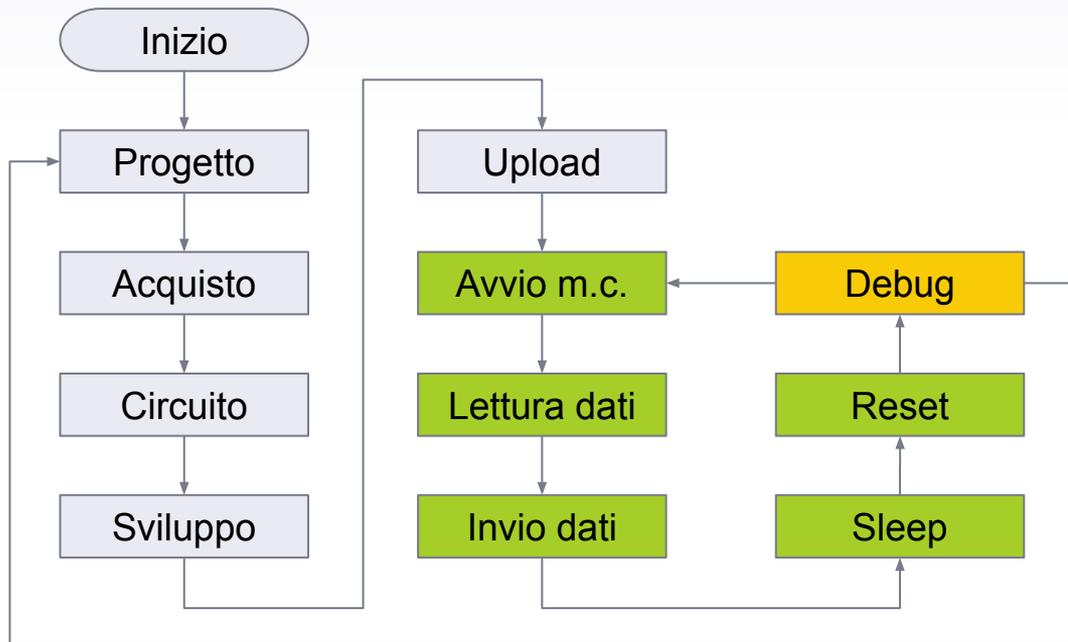
Shields

Molti produttori, a partire da Arduino, hanno i connettori disposti in modo standard, così che i produttori di componenti possano fare schede che si incastrano sui microcontrollori, senza cablaggi.

Shield = scudo = ci sta sopra



Ciclo di vita del sistema



4

Come li facciamo comunicare?

I bit viaggiano, fortunelli



un po' di basi di reti, spiace...

► Importante

Mezzo di trasmissione: supporto sul quale viaggiano i dati.

Protocollo: regole condivise con le quali ci si scambiano i dati.

Vari protocolli possono viaggiare su più mezzi di trasmissione (e possono cambiare durante il percorso)

Ethernet può essere sia su fibra che su rame

TCP/IP può essere su rame come su etere (Wi-Fi o altro standard)

Sullo stesso tipo di fibra può passare Ethernet o Fibre Channel

► Memorizzare i dati di molti sensori

Pacchetti di pochi byte

Diradati nel tempo

Provenienti da molte fonti

Sensori tipicamente a batteria dove ogni byte consuma e pesa

Serve un protocollo facile, leggero e con poco overhead (dati di contorno che non sono l'informazione)

MQTT

Message

Queue

Telemetry

Transport

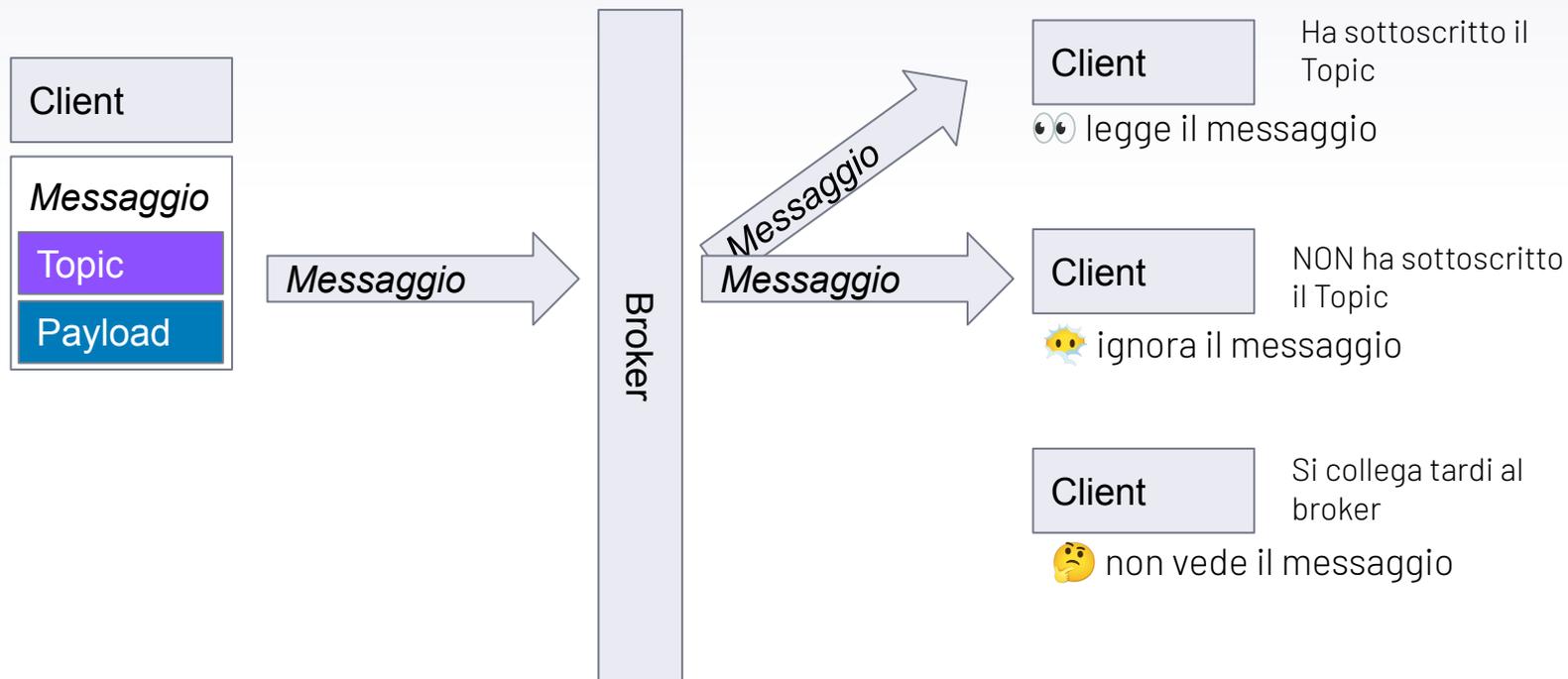


È facile e bellissimo

- Client:** chi legge e scrive i dati
- Broker:** chi riceve e dispaccia i messaggi
- Topic:** argomento del messaggio
- Payload:** dato da trasmettere
- Subscriber:** sottoscrittore ad un certo topic

Viaggia su ogni rete: cablata, Wi-Fi, LoRa, GPRS, 3/4/5G...

Il giro dei dati



► Creare il broker

```
$ sudo apt install mosquitto mosquitto-clients  
$ sudo systemctl is-enabled mosquitto  
$ sudo systemctl status mosquitto
```

Installato e verificato che è attivo.

Adesso potete usarlo come broker nella vostra rete.

Potete anche usare un broker su qualche servizio cloud, se avete sensori in giro per Internet

► Publicare una temperatura

Sono pezzetti di codice, si trova documentazione ovunque

```
#include <PubSubClient.h> //Libreria MQTT
WiFiClient espClient; //se uso la WiFi, potrei usare altro
PubSubClient clientMQTT(espClient); //oggetto MQTT sulla WiFi
const char* mqtt_server = "192.168.88.11"; //IP del broker
String clientId = "balcone"; //nome del sensore
clientMQTT.setServer(mqtt_server, 1883); //a che broker:porta mi connetto
clientMQTT.connect(clientId.c_str()); //attiva la connessione
client.publish("temperature", "18"); //mando il messaggio
```

► Pubblicare una temperatura

A questo punto il Broker riceve da “balcone”, con topic “temperature” il valore “18” (<– attenzione! Si trasmettono solo stringhe) e lo dice a tutti quelli connessi allo stesso broker.

Chi ha sottoscritto il topic “temperature” leggerà il messaggio inviato dal sensore del balcone e rimandato dal broker.

Una volta inviato, il messaggio è andato e non è più recuperabile.

Memorizzare la temperatura

(esempio in Python)

```
import paho.mqtt.client as mqtt
import sqlite3

client = mqtt.Client() # crea il client MQTT
client.connect("ip_server", 1883) # si collega al broker
conn = sqlite3.connect("balcone.db") # connessione al DB SQLite dove c'è già la tabella
cur = conn.cursor() # crea il cursore per lavorare nel DB
client.subscribe("balcone") # sottoscrizione al topic 'balcone'
while True: # aspetta l'arrivo dei messaggi, per sempre (fino a Ctrl+C)
    message = client.receive() # memorizza il messaggio
    # Salva il messaggio nella tabella
    cur.execute('''
INSERT INTO messages (topic, payload) VALUES (?, ?)
''', (message.topic, message.payload))
    conn.commit()

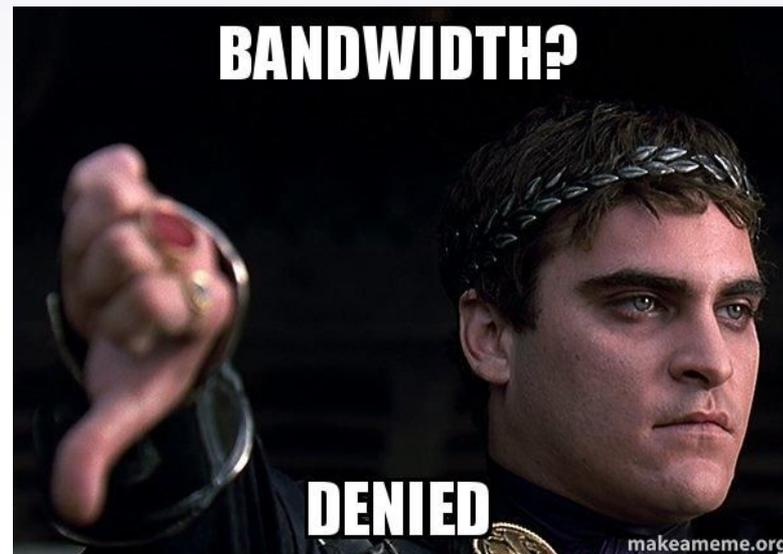
client.disconnect() #chiude la connessione (quando qualcuno lo fa uscire dal ciclo
```

► Quanto si risparmia?

	HTTP	MQTT	%
Handshake	2.261	5.572	+146%
Un messaggio	3.285	388	-88%

Dati in Bytes

Fonte: <https://www.hivemq.com/blog/mqtt-vs-http-protocols-in-iiot/>



► Quanto si risparmia?

Meno banda usata = meno batteria

Codice più semplice = meno batteria

Trasmissione più veloce = meno batteria

Meno batteria = il sensore dura di più prima che io debba cambiarla.

► Chi usa MQTT?

Facebook messenger

NodeRed

Home Assistant

Tutti i dispositivi Shelly, in modo nativo

BMW (quelli delle automobili)

I luoghi del male

Pimoroni: <https://shop.pimoroni.com/>

Tantissimi accessori per Raspberry Pi (da UK senza dogana)

The Pi Hut: <https://thepihut.com/>

Anche qui tanta roba interessante per Raspberry Pi

Wemos: <https://www.wemos.cc/en/latest/>

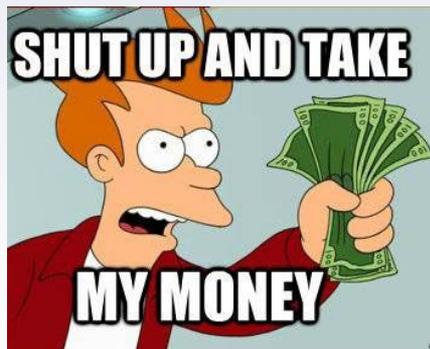
ESP8266/32 con espansioni ben supportate (da Aliexpress)

Arduino: <https://store.arduino.cc/>

Store ufficiale di Arduino (prezzi IVA esclusa, occhio!)

Shelly: <https://shelly.cloud>

Dispositivi domotici aperti a ogni protocollo



► Un OT importante

Non fate questo errore, quando inizierete la tesi, fate backup e versioning:

- ▶ Comprate un disco esterno (non flash)
- ▶ Fate backup sia sul disco che nel cloud
- ▶ Salvate versioni settimanali della tesi e mantenete lo storico
- ▶ Testate il recupero dei documenti salvati.

**le hanno rubato la borsa:
“Tenetevi tutto, ma ridatemi
la tesi altrimenti non mi
posso laureare”**

L'appello lanciato sui social per recuperare il lavoro di mesi

GRAZIE!

Domande?

Potete trovarmi...

- ▶  ft@tucci.boo
- ▶ www.pilloledib.it (il podcast)
- ▶ www.iltucci.com (il blog)
- ▶ [@cesco_78@mastodon.social](https://mastodon.social/@cesco_78)
- ▶ [@cesco_78](https://twitter.com/cesco_78) (Twitter)



Credits

Special thanks to all the people who made and released these awesome resources for free:

- ▶ Presentation template by [SlidesCarnival](#)
- ▶ Illustrations by [Sergei Tikhonov](#)
- ▶ Photographs by [Unsplash](#)