

PassChain

Gruppo 13 - G. Spina, A. Montefusco, O. Szuba

Internet Of Things - 20/05/2022

Password Managers

Cosa sono?

Servizi online che permettono agli utenti di
memorizzare le password
e sincronizzarle su tutti i dispositivi personali

Password Managers

LCD DISPLAY

FileToolsSyncExtensionsHelp

Search...

PASSWORD MANAGER

PasswordsSecurity DashboardSecure Notes

WALLET

Personal InfoPaymentsIDsReceipts

CONTACTS

Sharing CenterEmergency

Sync: On

You're Premium

Add newPassword ChangerShare

Most used

airbnbairbnb.comstew.rob007@gmail.c...

facebookfacebook.comstew.rob007

amazonamazon.comROBSt007

Adobeadobe.comstew.rob007@gmail.c...

Etsyetsy.comrob@stewartworks.com

boxbox.comstew.rob007@gmail.c...

GROUPONGroupon.comstew.rob007@gmail.c...

PayPalpaypal.comstew.rob007@gmail.c...

RunKeeperrunkeeper.comstew.rob007@gmail.c...

TwitterRobstewartD

LinkedInrob@stewartworks.com

linkedin.comstew.rob007@gmail.c...

Doodledoodle.comstew.rob007@gmail.c...

hotmail.comstew.rob007

Nortonstew.rob007@g...

uber.com

ft.com

netflix.com

kayak.co.uk

Youtube.com

Favorites

Air Canadawendy.appleseed@gmail.com

Applewendy.appleseed@icloud.com

Ebaywendyappleseed

Gmailwendy.appleseed@gmail.com

Instagramwendy.appleseed

TD - Checking Accountwendyappleseed

Twitterwendy_appleseed

Air Canada

Personal

Go

username

wendy.appleseed@gmail.com

password

.....

website

https://aircanada.com

Last modified: Jul 17, 2019, 10:02 AM

Created: May 13, 2018, 9:27 AM

Edit

1Password keeps you safe online

Dashlane e 1Password sono i password manager più popolari

Password Managers

Pro

- App disponibili per tutte le piattaforme
- Le estensioni browser permettono di inserire automaticamente le password
- Sincronizzati su tutti i dispositivi

Contro

- Servizi online e cloud quindi violabili anche da remoto
- Necessità di installare un'applicazione o estensione su ognuno dei propri dispositivi personali
- Poco pratico in caso di utilizzo di dispositivi non personali

Il disastro del 2016

LCD DISPLAY

2016

- **MyPasswords, InformatiCore, LastPass, Keeper, F-Secure Key, Dashlane, KeepSafe, Avast Passwords, and 1Password:** This was a busy year in terms of password management vulnerabilities. **TeamSIK (Security Is Key)**, a group of people interested in IT security from the Fraunhofer Institute for Secure Information Technology, discovered serious security **flaws in the most popular password management apps** developed for the Android platform.
- **LastPass:** Google Project Zero Hacker Tavis Ormandy discovered a **critical zero-day flaw** that allowed any remote attacker to compromise accounts completely.

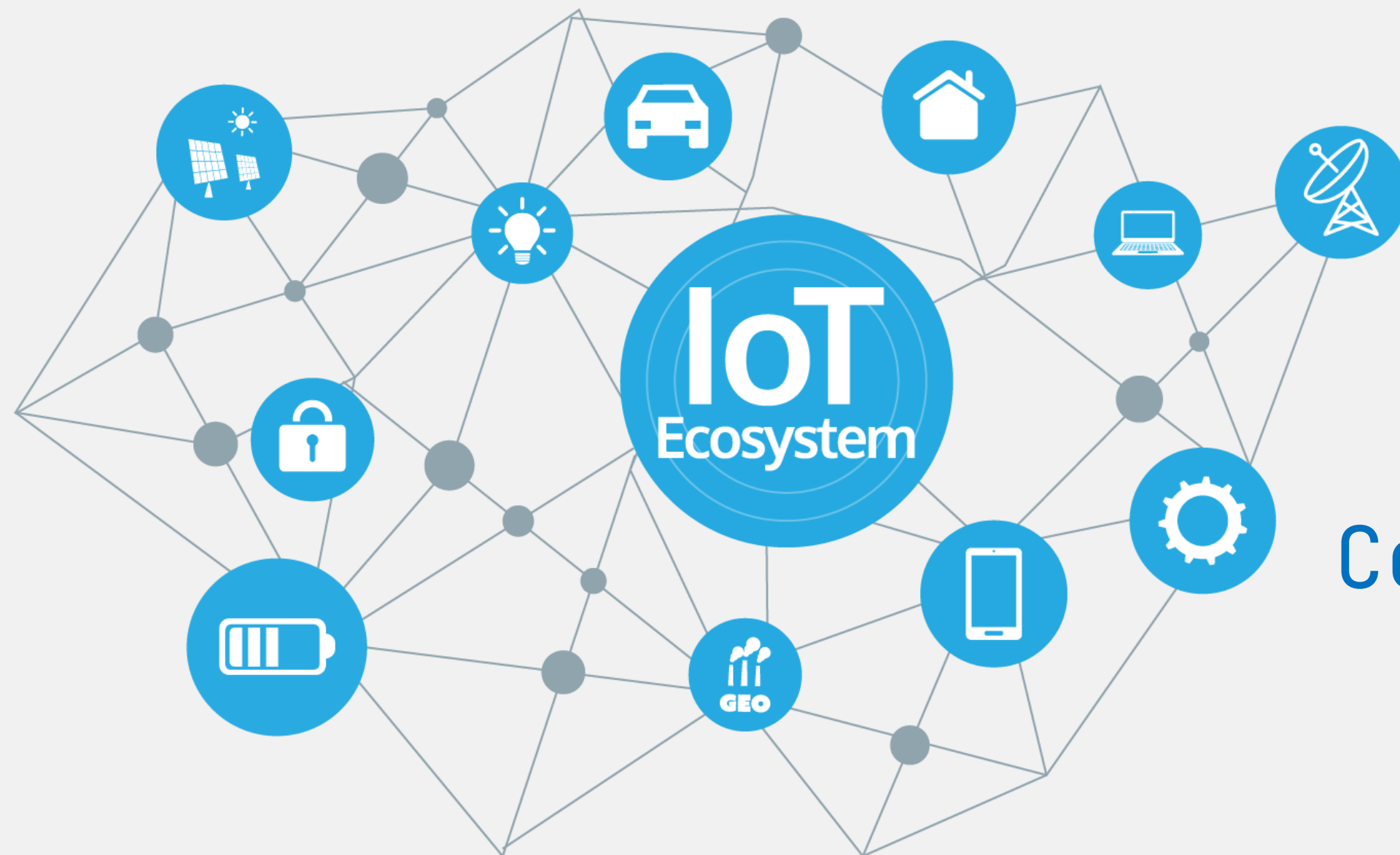
Nel 2016 sono state scoperte vulnerabilità in quasi tutti i password manager più popolari
Fonte: <https://password-managers.bestreviews.net/faq/which-password-managers-have-been-hacked/>



Cos'è PassChain

PassChain: IoT

L'Internet of Things
è il processo di connessione a
Internet di oggetti fisici di
utilizzo quotidiano.

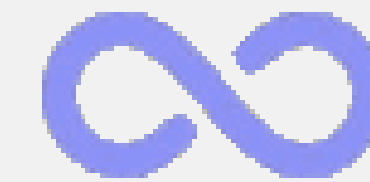


Con l'acronimo IoT si indica qualsiasi
sistema di dispositivi fisici che
ricevono e trasferiscono i dati su
reti wireless.

PassChain: IoT



Un tipico **sistema IoT** funziona grazie all'invio, alla ricezione e all'analisi dei **dati** in un ciclo continuo di feedback.



A seconda del tipo di sistema IoT, l'analisi può essere eseguita tramite intervento manuale o da tecnologie di:

→ **intelligenza artificiale;**

→ **machine learning**

PassChain: cosa fa?

PassChain ha come obiettivo quello di facilitare l'utente nell'autenticazione digitale ma anche di assicurare la sicurezza attraverso la sua funzione di password manager.



Tramite PassChain, l'utente può:

→ connettersi ad altri dispositivi tramite Bluetooth;



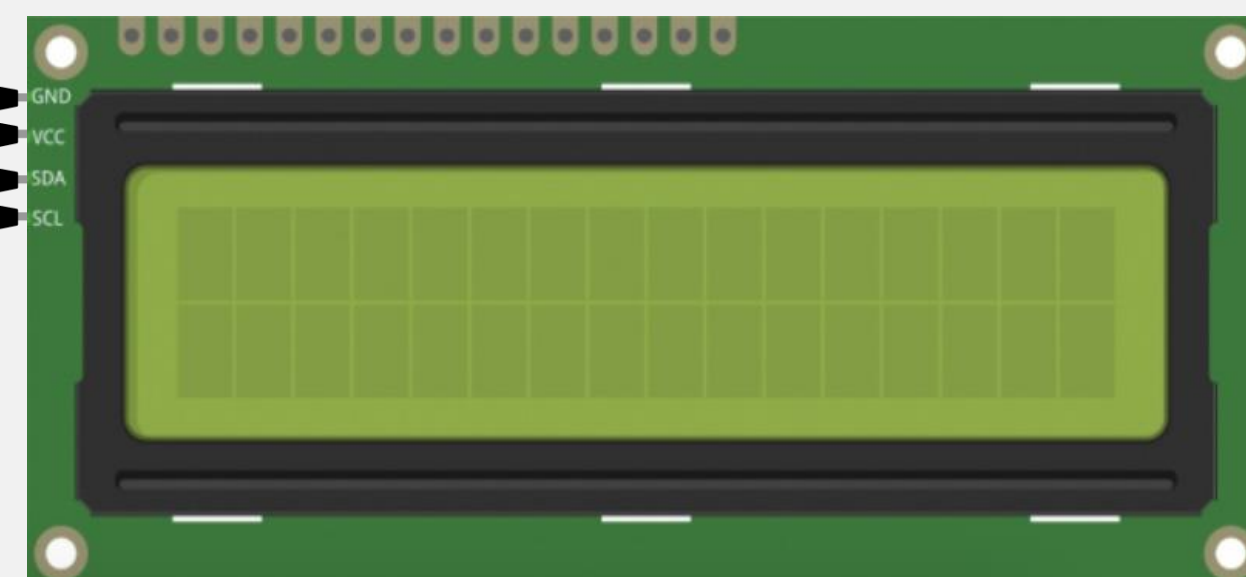
→ autenticarsi tramite un tastierino numerico;

→ collegarsi all'App desktop «PassChain» tramite protocollo MQTT.

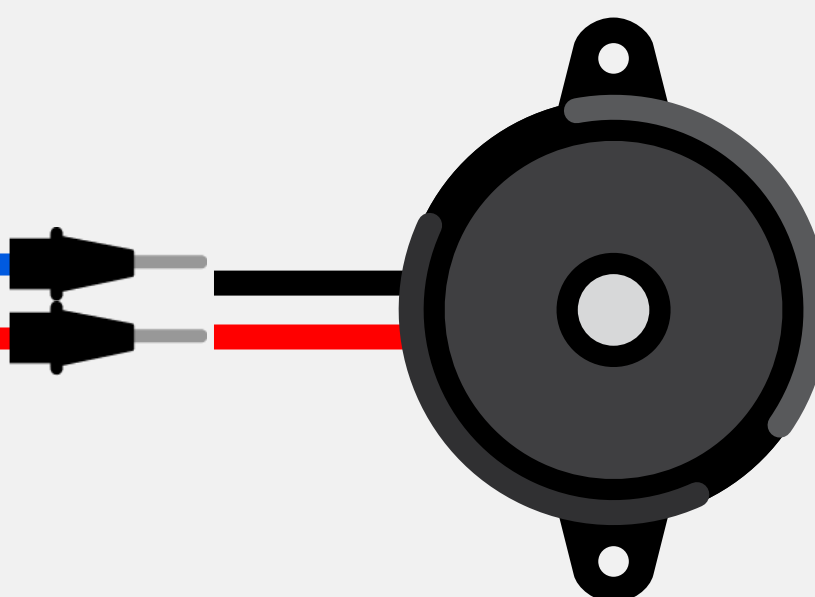


Il nostro Hardware

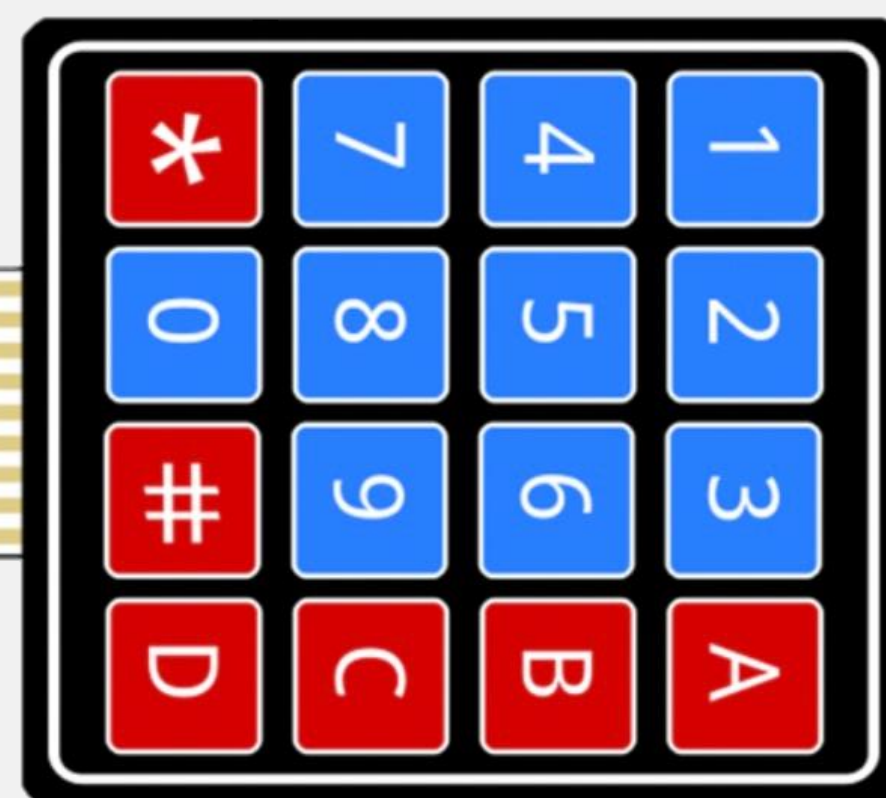
Hardware utilizzato



Display LCD con modulo
di comunicazione I2C



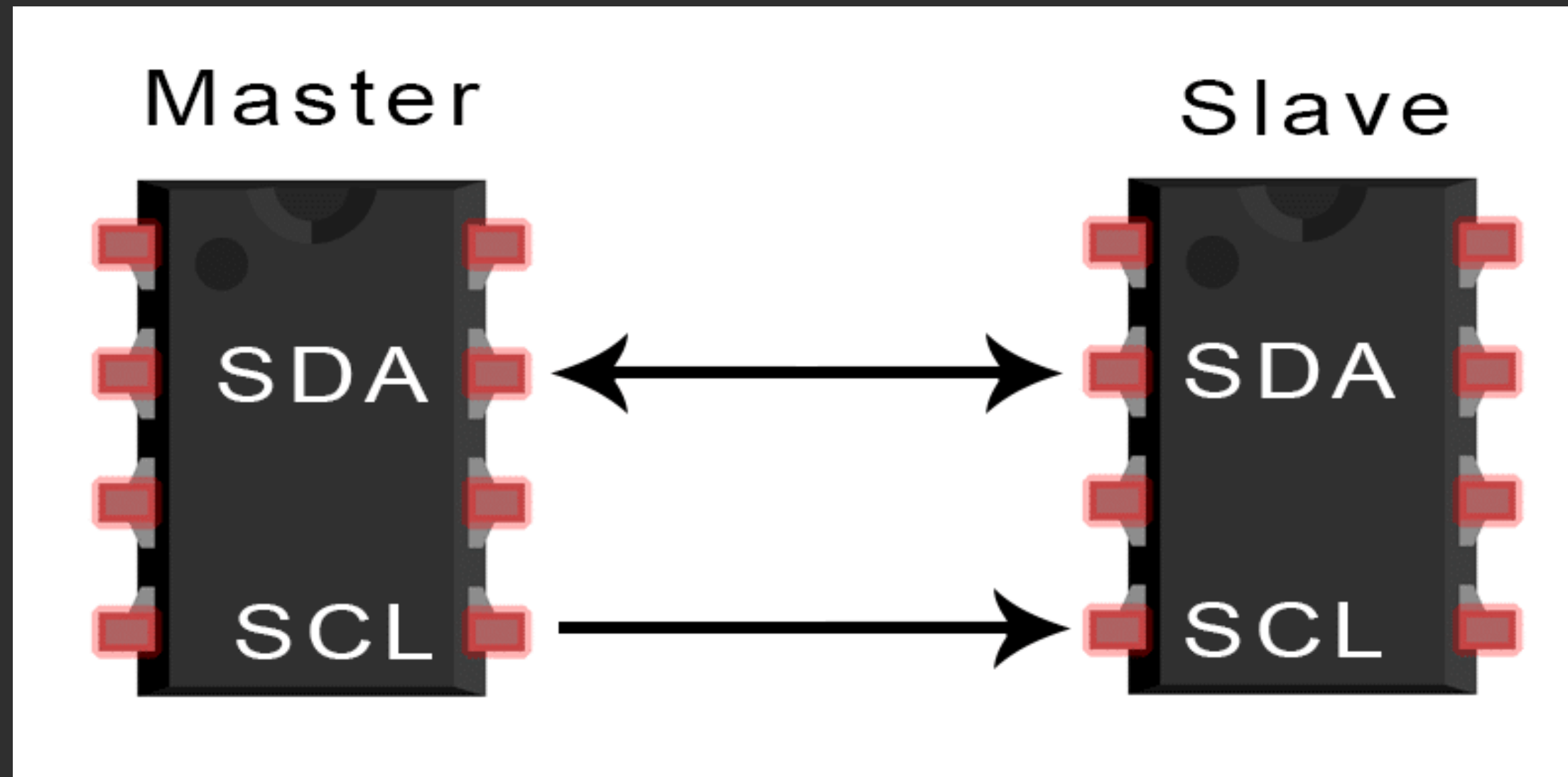
Buzzer passivo
con controllo PWM



Tastierino
alfanumerico

Protocollo I2C

LCD DISPLAY



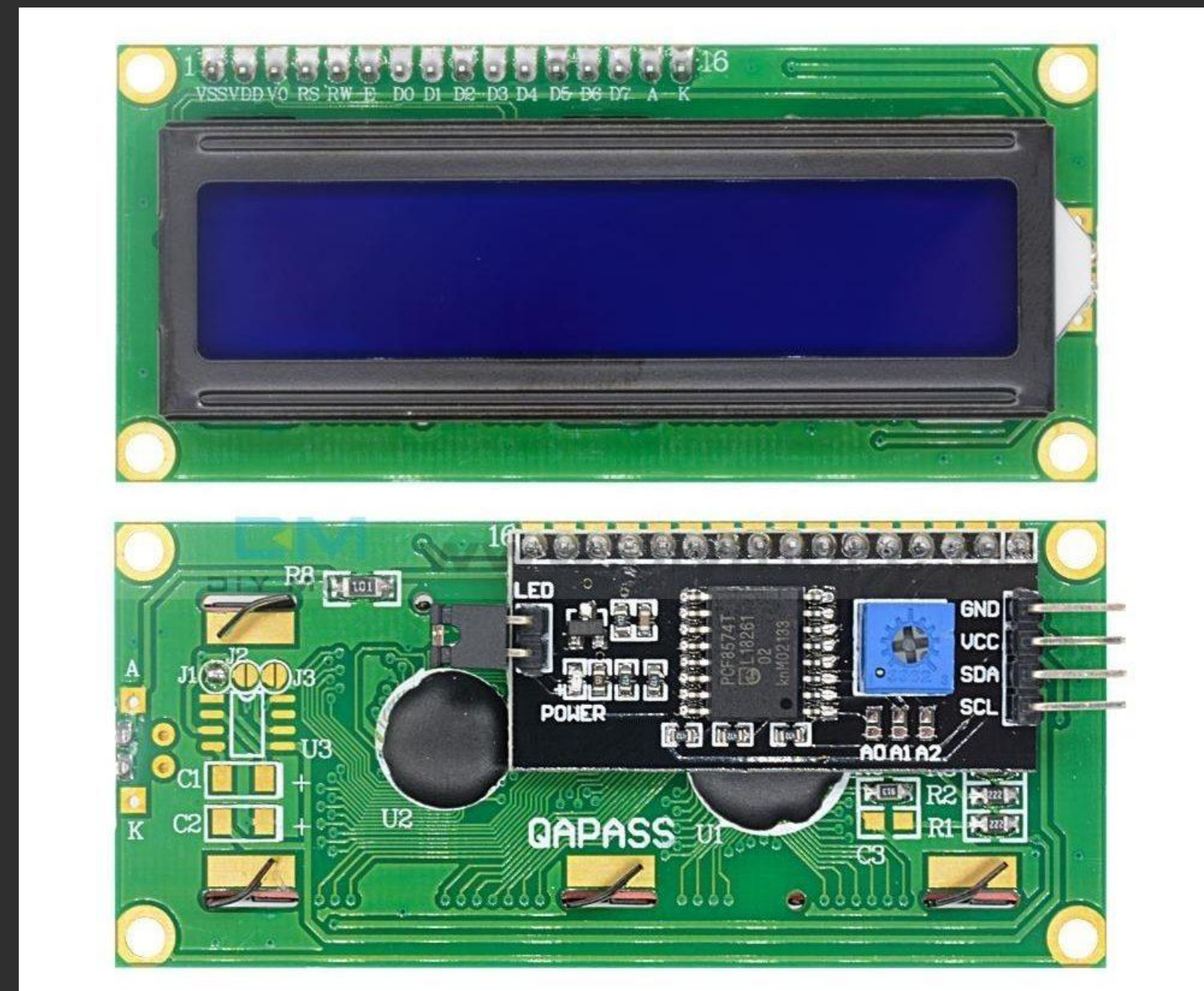
Inter-Integrated Circuit è un protocollo sincrono, half-duplex, master-slave e con linee condivise.

Ogni slave necessita di un "indirizzo" separato che viene trasmesso sulla linea dal master per selezionare lo slave. Richiede meno linee di SPI, ma è più lento anche quando si utilizza la stessa frequenza di clock.

Protocollo I2C

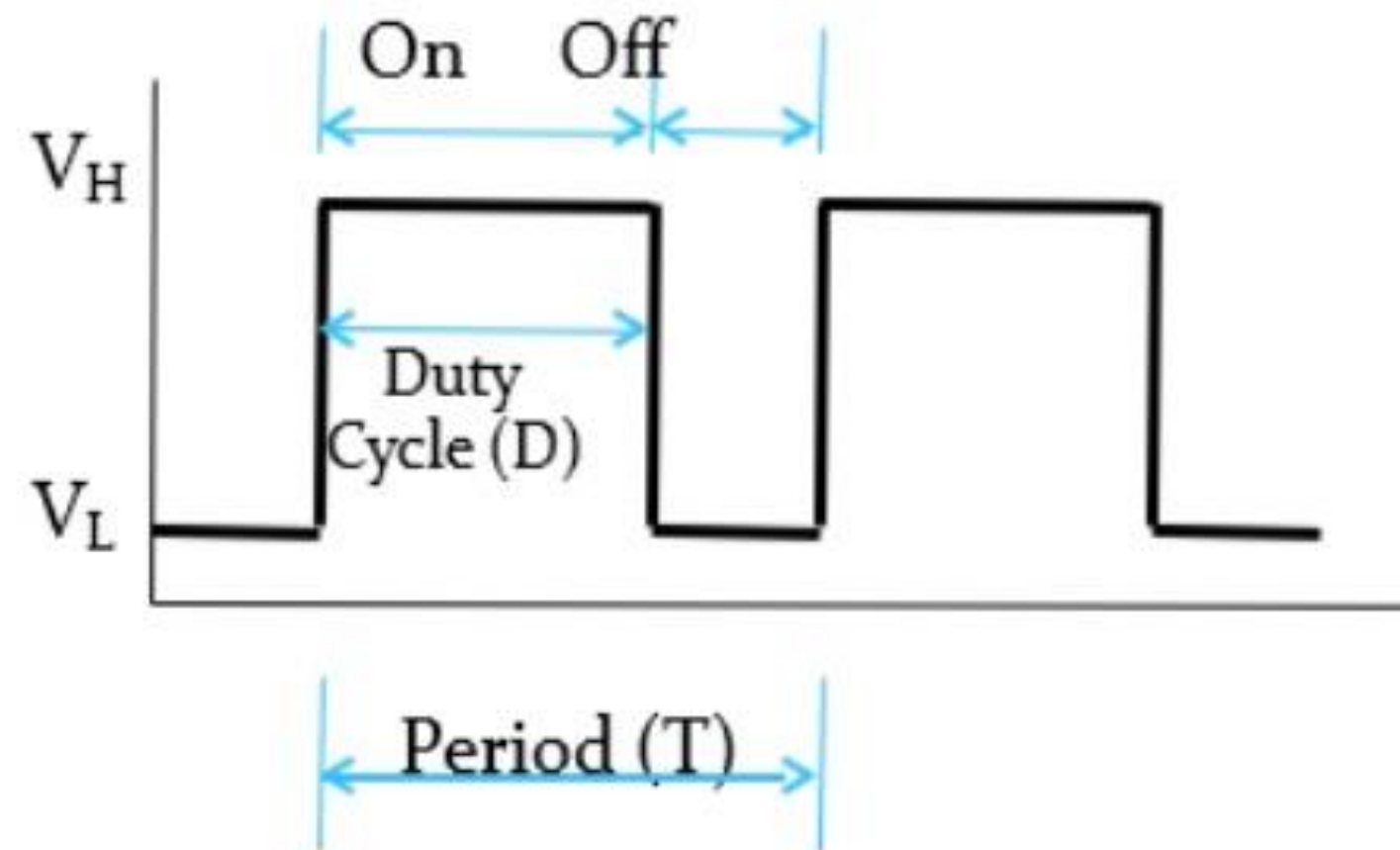
LCD DISPLAY

Nel nostro caso l'abbiamo utilizzato per poter comunicare con il display LCD, così da poterlo utilizzare mediante l'uso di soli 4 pin, a differenza dei 16 di norma richiesti



I display LCD possono avere il chip per usufruire del protocollo I2C integrato nella PCB, o come in foto le due schede sono successivamente assemblate.

Pulse Width Modulation (PWM) è un tipo di modulazione digitale che permette di ottenere una tensione media variabile che dipende dal rapporto tra la durata dell'impulso positivo e dell'intero periodo (duty cycle): il duty cycle non è altro che l'ampiezza del segnale e indica quanto tempo il segnale rimane su on rispetto al periodo.



$$\text{Duty Cycle (D)} = \frac{\text{On Time}}{\text{Period}} \times 100\%$$

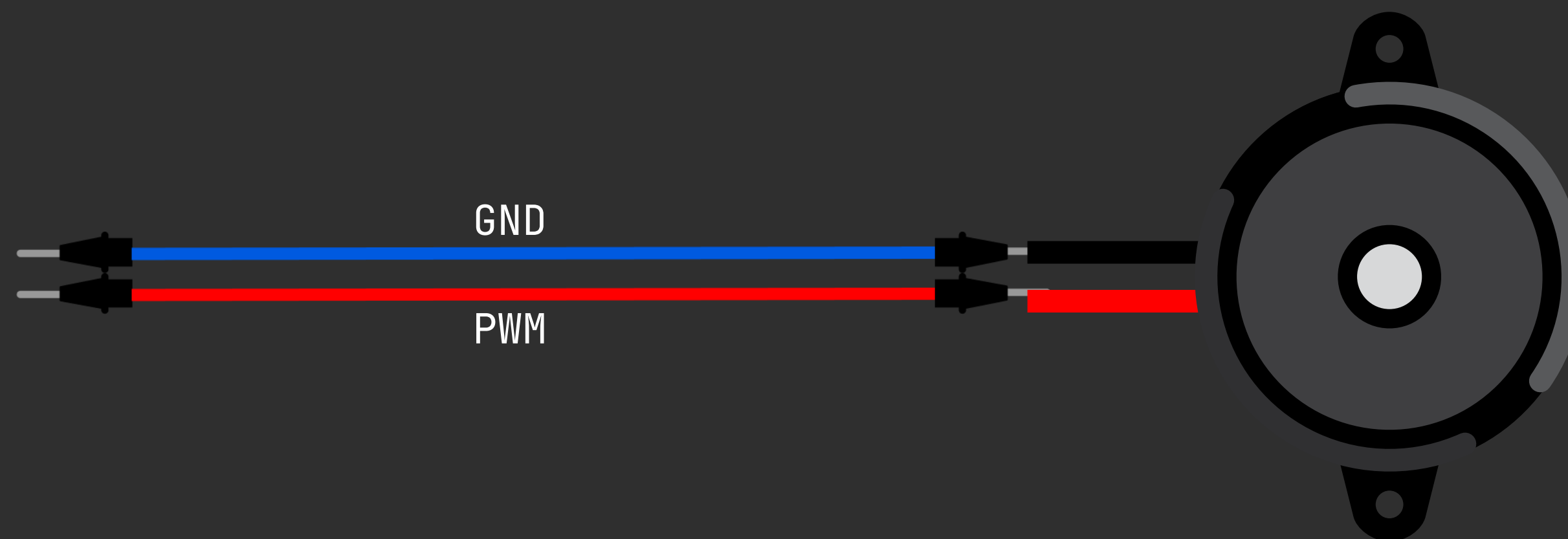
Media del segnale:

$$V_{avg} = D \cdot V_H + (1 - D) \cdot V_L$$

di solito $V_L = 0V$ per semplicità

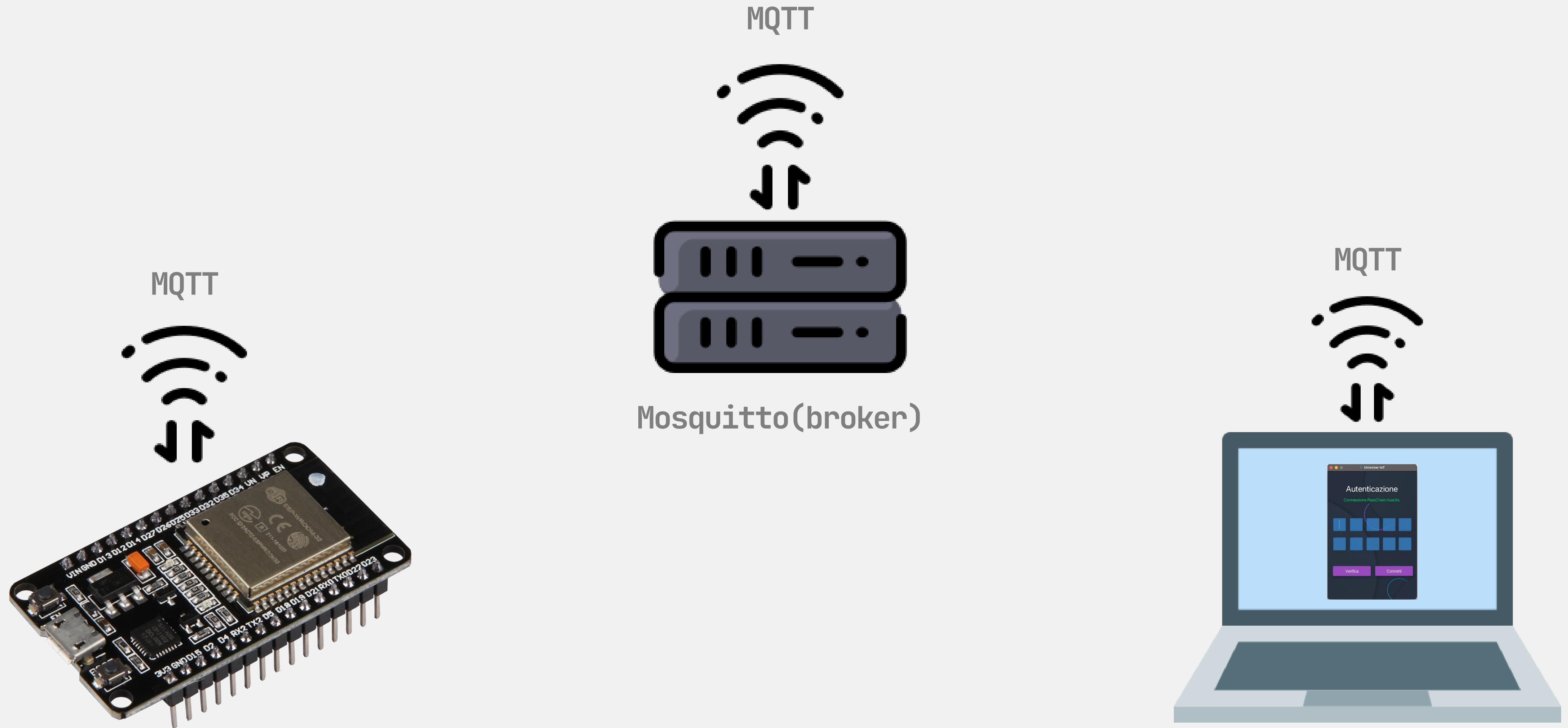
LCD DISPLAY

Noi abbiamo usufruito del PWM per controllare il nostro attuatore: un buzzer di tipo passivo, che mediante il PWM permette di controllare il dispositivo variandone il tono emesso.



Il nostro Software

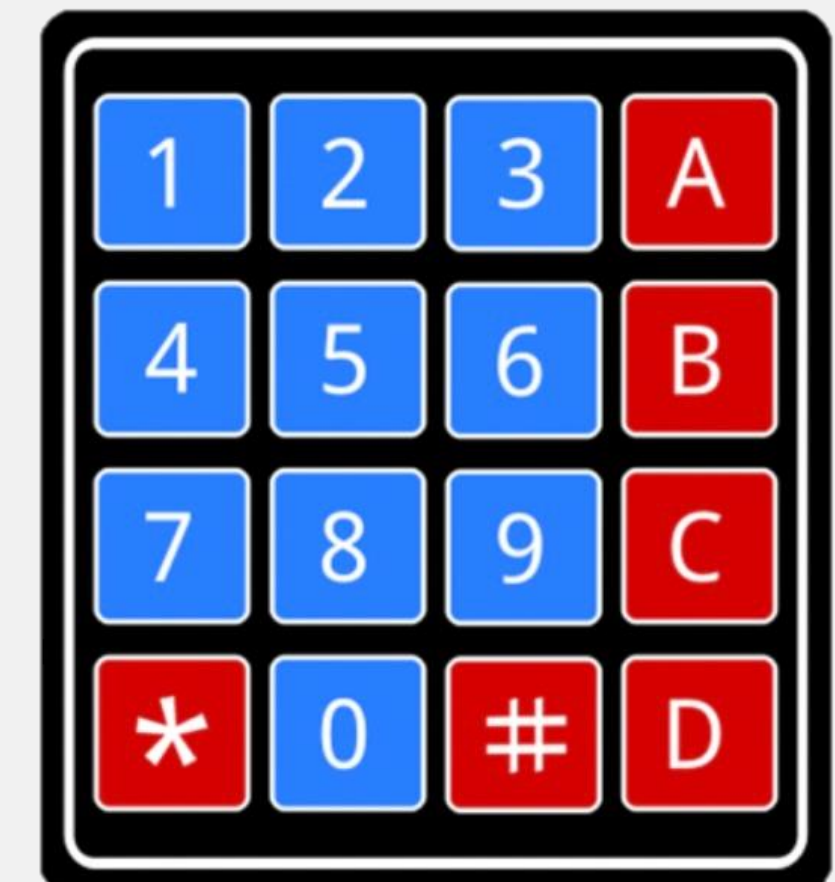
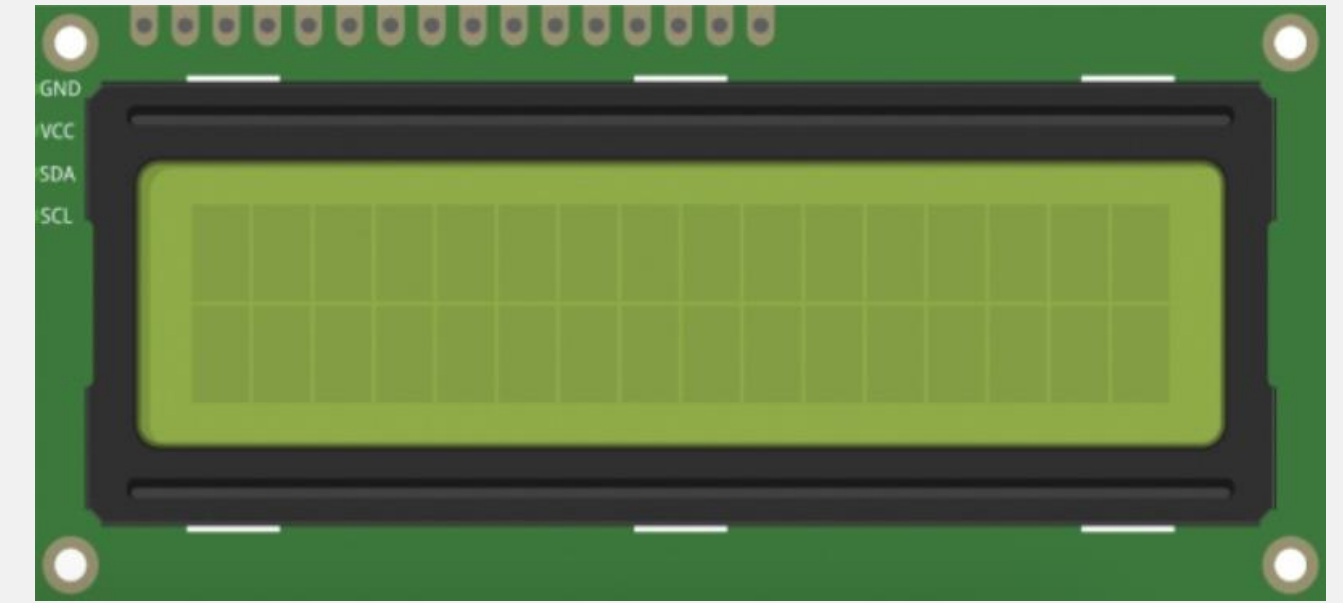
Software overview e funzionalità



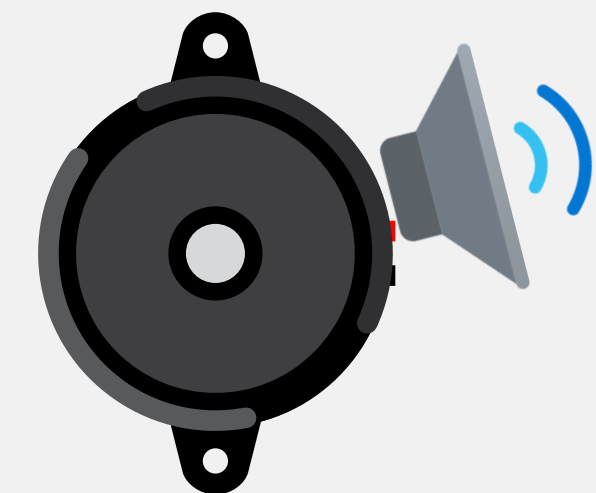
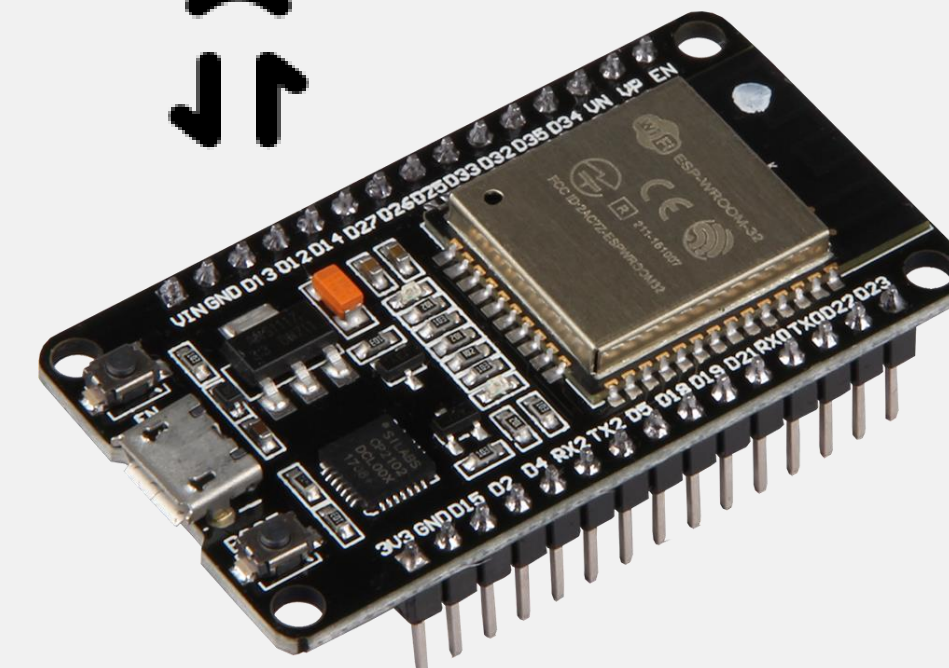
Software overview e funzionalità

L'utente mediante il display LCD e un tastierino alfanumerico può interagire con il sistema navigando nel menù del dispositivo e scegliere se:

inviare credenziali mediante bluetooth;

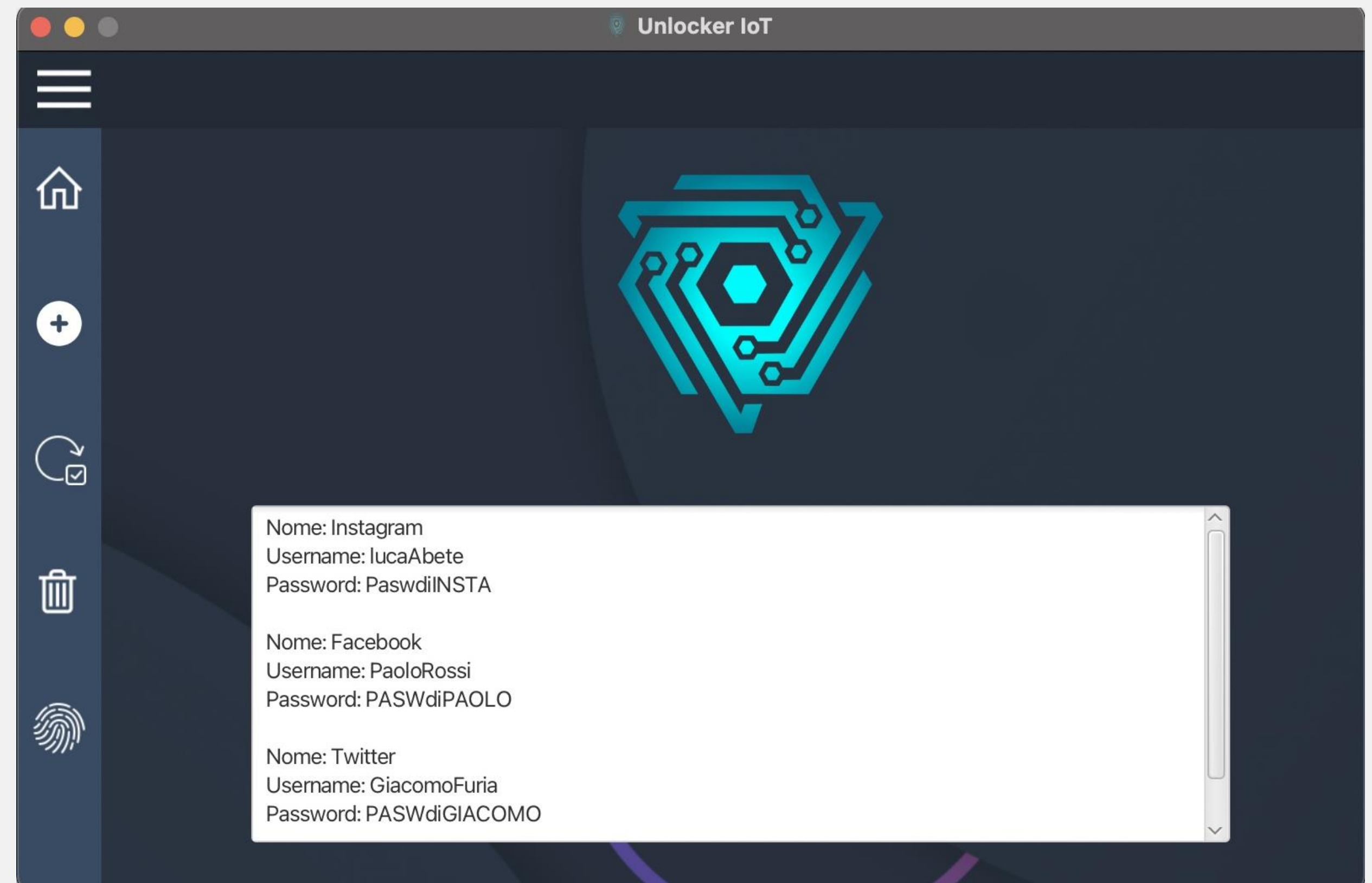
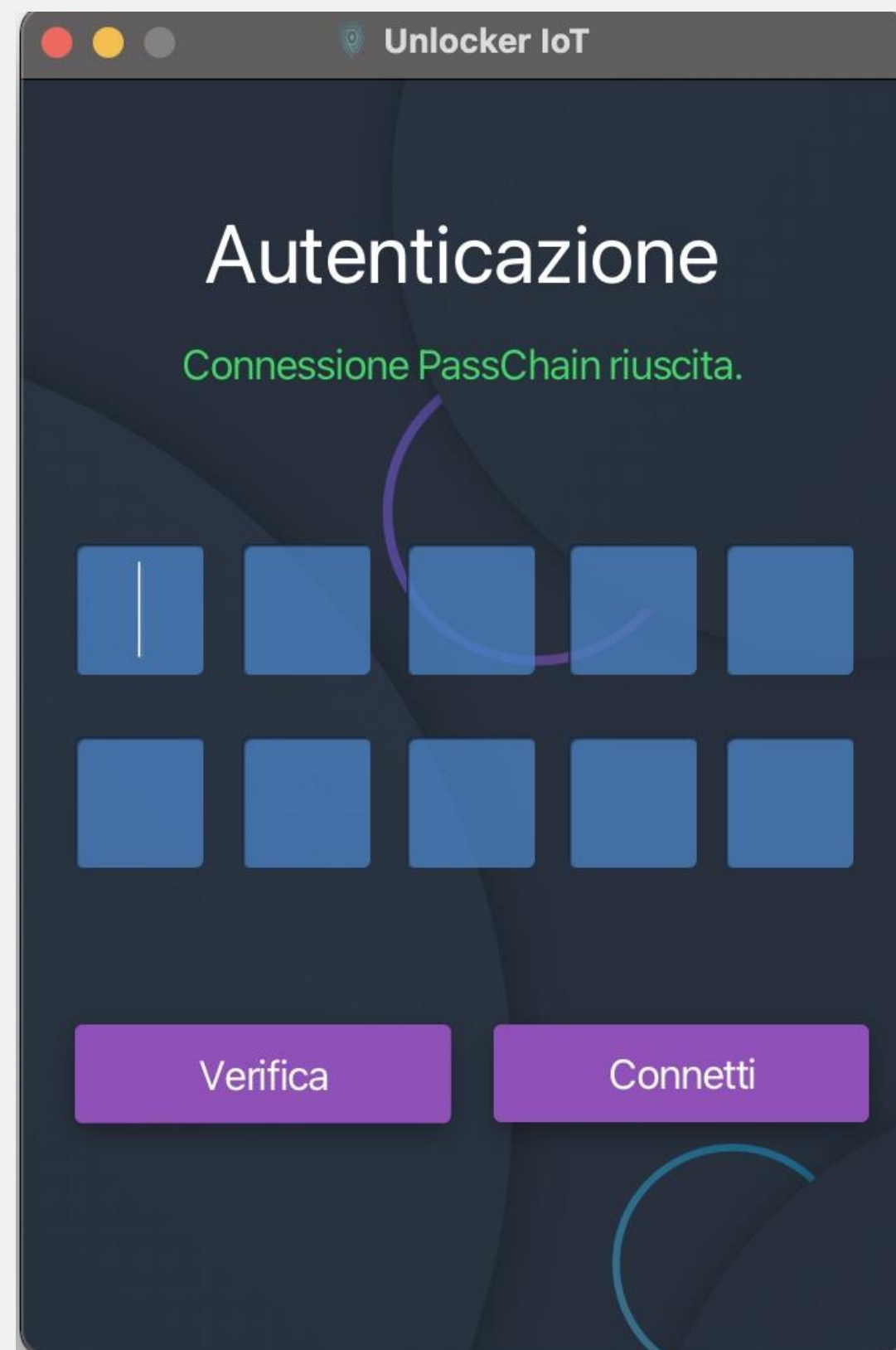


MQTT
Bluetooth



Software overview e funzionalità

Connettere il dispositivo alla GUI per poter gestire le password (aggiunta, rimozione e aggiornamento) e per poter modificare il codice pin per l'autenticazione.

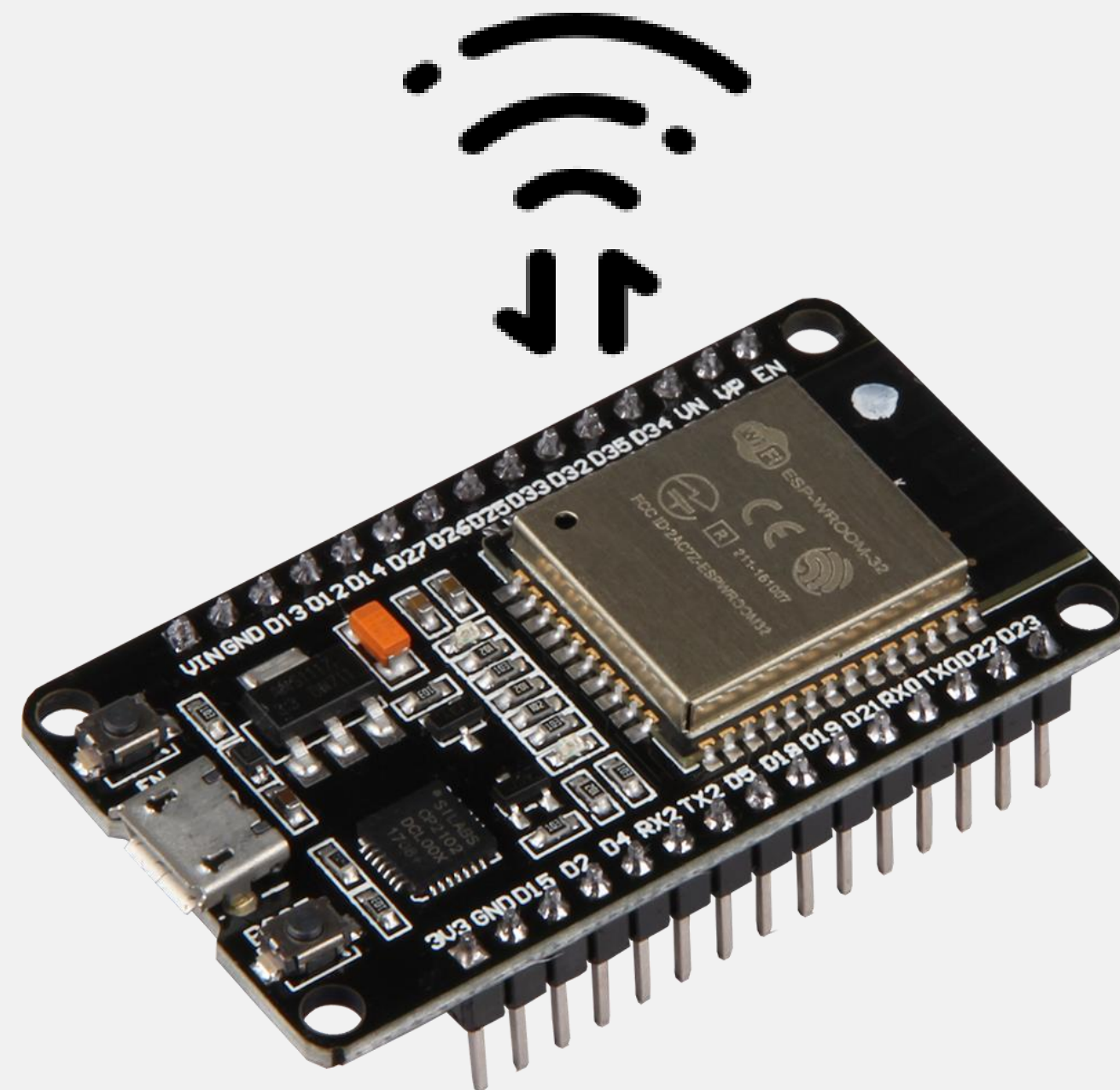


Bluetooth

Software overview e funzionalità

La board mediante l'uso della libreria « MicroPythonBLEHID » può essere utilizzata per simulare il comportamento di una tastiera bluetooth, quindi, permette l'inserimento di testo in qualsiasi campo di testo, su qualsiasi dispositivo che supporti tastiere bluetooth.

Bluetooth



Bluetooth

A screenshot of a MicroPython LCD display with a green border and four orange corner markers. The display shows a Python function named `send_char` that takes `self` and `char` as arguments. The function uses a series of `if` and `elif` statements to map specific characters to a numeric code. For lowercase letters 'a' through 'z', it uses a modulus of 0 and a base code of 0x2C. For uppercase letters 'A' through 'Z', it uses a modulus of 1 and a base code of 0x04. For digits '0' through '9', it uses a modulus of 2 and a base code of 0x04. For punctuation characters '!' through '/', it prints a message and uses a modulus of 0. Any other character triggers an `assert 0` statement.

```
def send_char(self, char):  
    if char == " ":  
        mod = 0  
        code = 0x2C  
    elif ord("a") <= ord(char) <= ord("z"):  
        mod = 0  
        code = 0x04 + ord(char) - ord("a")  
    elif ord("A") <= ord(char) <= ord("Z"):  
        mod = 1  
        code = 0x04 + ord(char) - ord("A")  
    elif ord("0") <= ord(char) <= ord("9"):  
        mod = 2  
        code = 0x04 + ord(char) - ord("0")  
    elif ord("!") <= ord(char) <= ord("/"):   
        print("Ecco il char: "+ str(ord("/")))  
        mod = 0  
        code = (ord(char) - ord("!"))  
    else:  
        assert 0
```

« MicroPythonBLEHID »
è una libreria basata
sull'uso di questa
funzione, la quale
traduce i caratteri
che le vengono
passati in codice
ASCII.

Successivamente verrà utilizzata per la simulazione
della tastiera bluetooth.

Protocollo MQTT

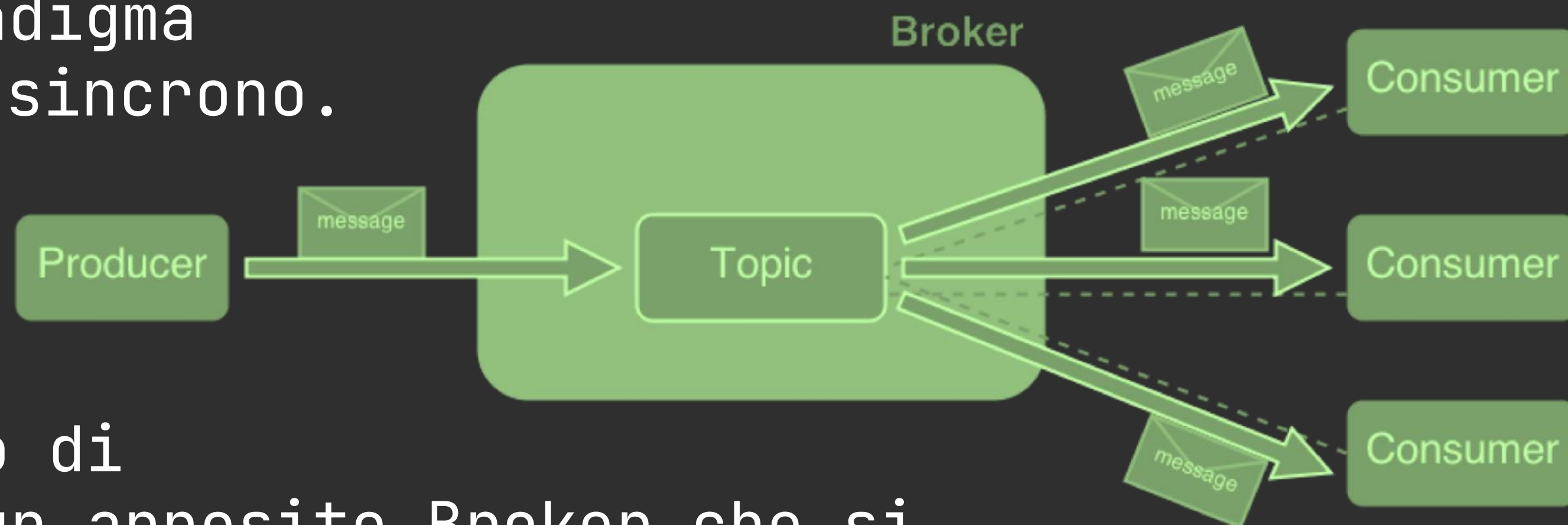
MQTT

LCD DISPLAY



Protocollo usato per lo scambio di messaggi con l'obiettivo di minimizzare il traffico sulle reti e richiedere poche risorse ai dispositivi per la sua gestione.

L'**MQTT** usa il paradigma di *pub/sub* che è asincrono.



Prevede lo scambio di messaggi tramite un apposito Broker che si occupa di consegnare il messaggio soltanto per i topics sottoscritti dal subscriber.

MQTT: mosquitto

Mosquitto è un **broker** di messaggi open source leggero ed è adatto per l'uso su tutti i dispositivi, dai computer a scheda singola a bassa potenza ai server completi.

Il protocollo MQTT fornisce un metodo leggero per eseguire la messaggistica utilizzando il modello **publish/subscribe**:

adatto per la messaggistica Internet of Things



Topic utilizzati:

topic_pub : «ESPcredentials»

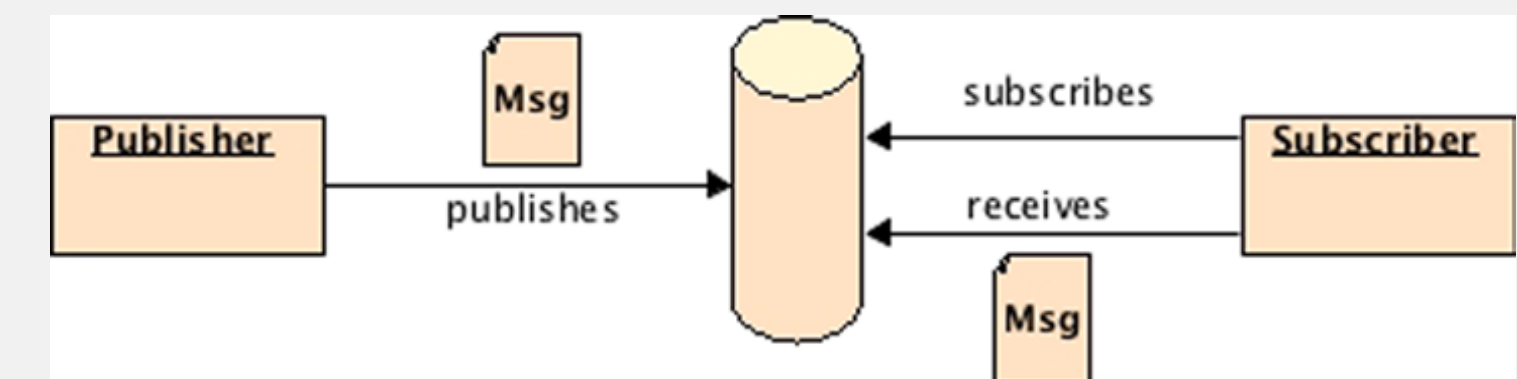
topic_sub : «APPcredentials»

Board ESP32

topic_sub : «ESPcredentials»

topic_pub : «APPcredentials»

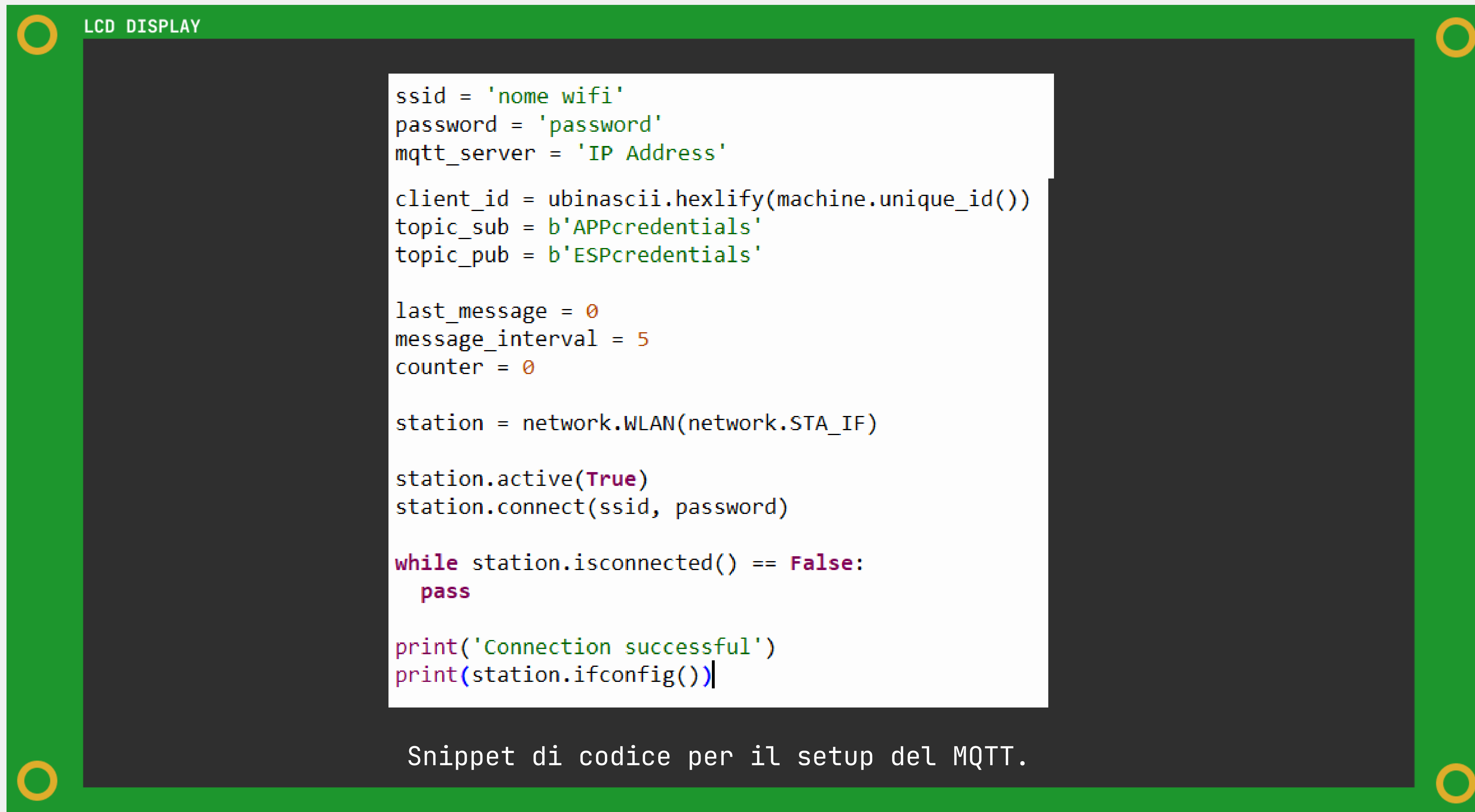
App Desktop



MQTT: codice ESP32

Setup MQTT in ESP32:

boot.py



```
ssid = 'nome wifi'
password = 'password'
mqtt_server = 'IP Address'

client_id = ubinascii.hexlify(machine.unique_id())
topic_sub = b'APPcredentials'
topic_pub = b'ESPcredentials'

last_message = 0
message_interval = 5
counter = 0

station = network.WLAN(network.STA_IF)

station.active(True)
station.connect(ssid, password)

while station.isconnected() == False:
    pass

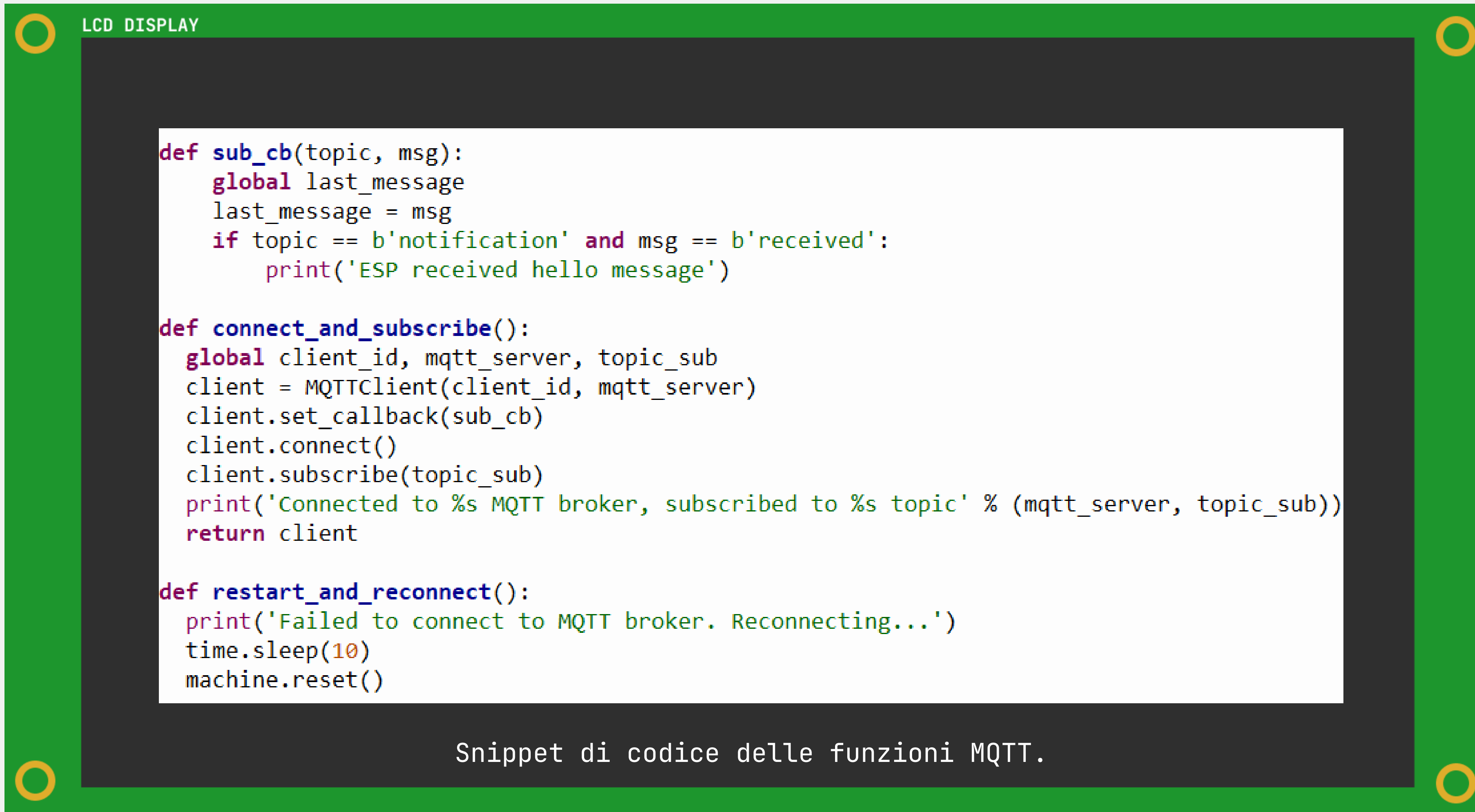
print('Connection successful')
print(station.ifconfig())
```

Snippet di codice per il setup del MQTT.

MQTT: codice ESP32

Setup MQTT in ESP32:

main.py

A green rectangular frame representing an LCD display. Inside the frame, on a dark grey background, is a white rectangular area containing Python code. The code defines three functions: sub_cb, connect_and_subscribe, and restart_and_reconnect. The sub_cb function checks for a specific topic and message. The connect_and_subscribe function sets up an MQTT client and subscribes to a topic. The restart_and_reconnect function handles connection failures by sleeping and resetting the machine. The text 'LCD DISPLAY' is in the top left corner of the green frame. The text 'Snippet di codice delle funzioni MQTT.' is at the bottom of the white area.

```
LCD DISPLAY

def sub_cb(topic, msg):
    global last_message
    last_message = msg
    if topic == b'notification' and msg == b'received':
        print('ESP received hello message')

def connect_and_subscribe():
    global client_id, mqtt_server, topic_sub
    client = MQTTClient(client_id, mqtt_server)
    client.set_callback(sub_cb)
    client.connect()
    client.subscribe(topic_sub)
    print('Connected to %s MQTT broker, subscribed to %s topic' % (mqtt_server, topic_sub))
    return client

def restart_and_reconnect():
    print('Failed to connect to MQTT broker. Reconnecting...')
    time.sleep(10)
    machine.reset()

Snippet di codice delle funzioni MQTT.
```

MQTT: codice Java

Setup MQTT in GUI Java:

MQTT_connection.java

LCD DISPLAY

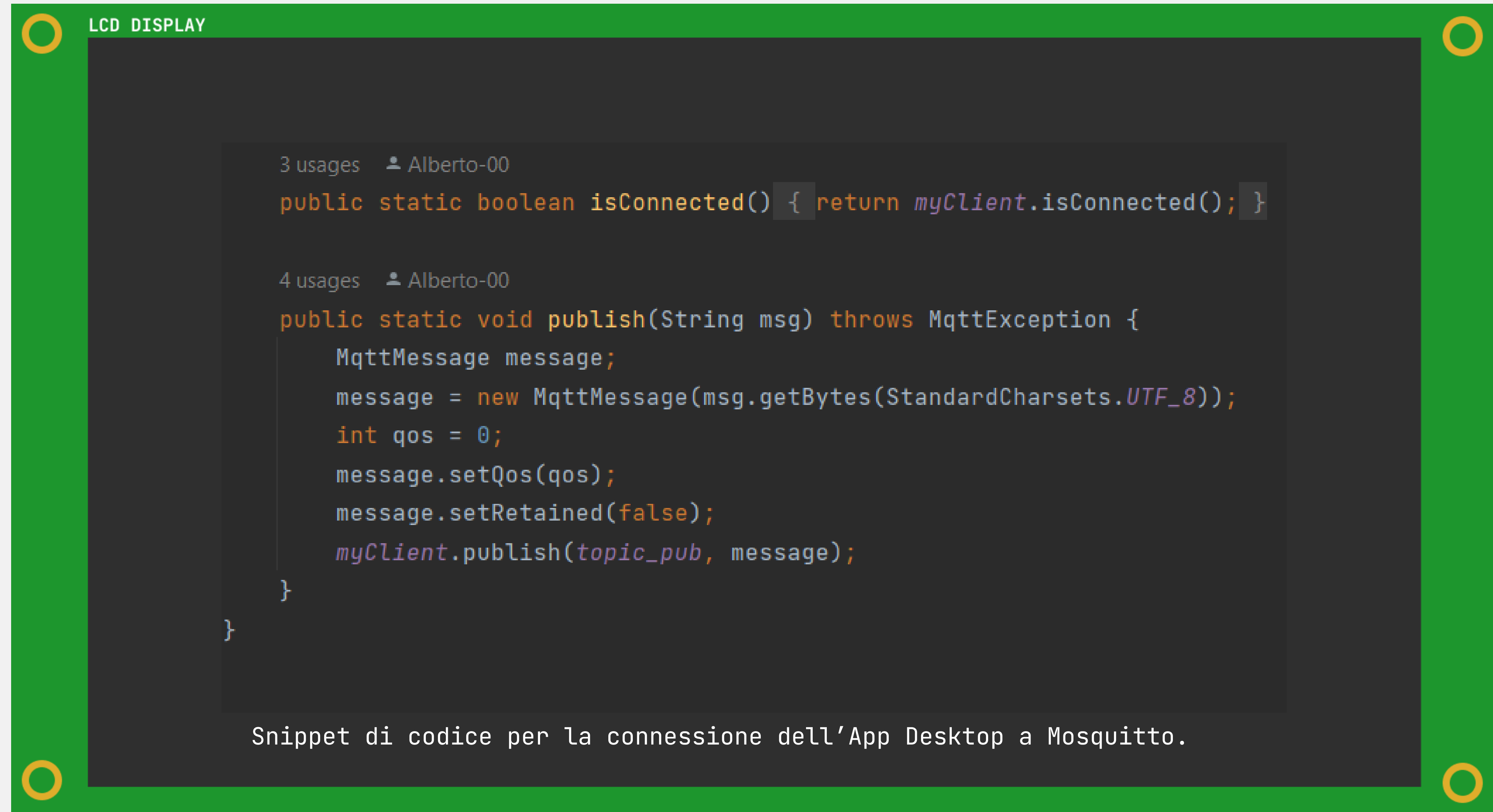
```
public class MQTT_comunication {  
    8 usages  
    private static MqttAsyncClient myClient;  
    1 usage  
    private static final String topic_sub = "ESPcredentials";  
    2 usages  
    private static final String topic_pub = "APPCredentials";  
  
    2 usages  Alberto-00 *  
    public static void connect() throws MqttException, UnknownHostException {  
        System.out.println(InetAddress.getLocalHost().getHostAddress());  
        myClient = new MqttAsyncClient( serverURI: "tcp://" + "172.20.10.7" + ":1883",  
            UUID.randomUUID().toString());  
  
        MyCallback myCallback = new MyCallback();  
        myClient.setCallback(myCallback);  
  
        IMqttToken token = myClient.connect();  
        try {  
            TimeUnit.SECONDS.sleep( timeout: 2);  
        } catch (InterruptedException e) {  
            throw new RuntimeException(e);  
        }  
        if (myClient.isConnected()){  
            token.waitForCompletion();  
            System.out.println("Connected to the broker !\n");  
  
            MqttMessage message = new MqttMessage("reconnect".getBytes());  
            int qos = 0;  
            message.setQos(qos);  
            message.setRetained(false);  
            myClient.publish(topic_pub, message);  
            myClient.subscribe(topic_sub, qos: 0);  
        }  
    }  
}
```

Snippet di codice per la connessione dell'App Desktop a Mosquitto.

MQTT: codice Java

Setup MQTT in GUI Java:

MQTT_connection.java



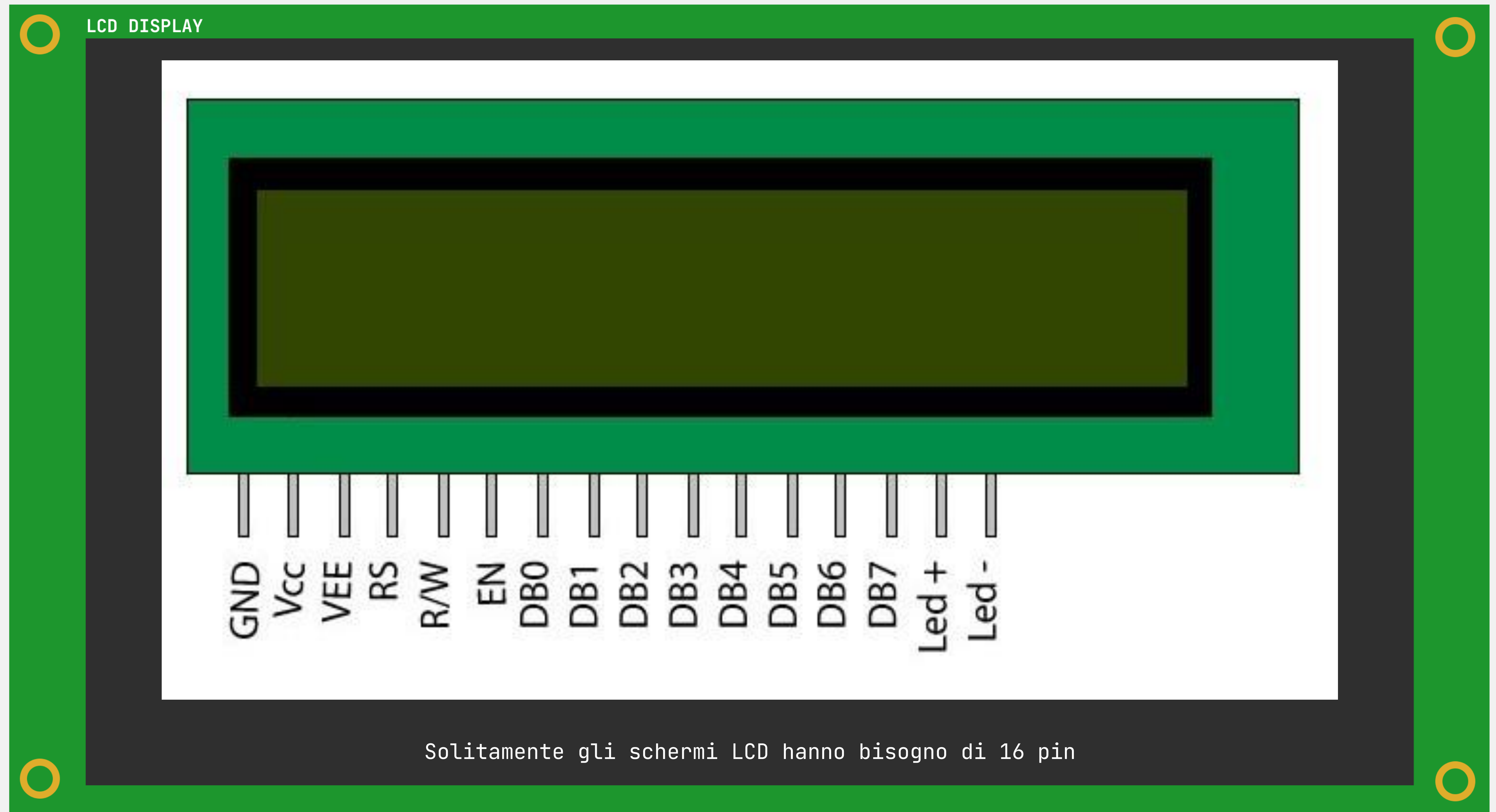
```
3 usages  👤 Alberto-00
public static boolean isConnected() { return myClient.isConnected(); }

4 usages  👤 Alberto-00
public static void publish(String msg) throws MqttException {
    MqttMessage message;
    message = new MqttMessage(msg.getBytes(StandardCharsets.UTF_8));
    int qos = 0;
    message.setQos(qos);
    message.setRetained(false);
    myClient.publish(topic_pub, message);
}
}
```

Snippet di codice per la connessione dell'App Desktop a Mosquitto.

IL nostro driver LCD

Schermo LCD



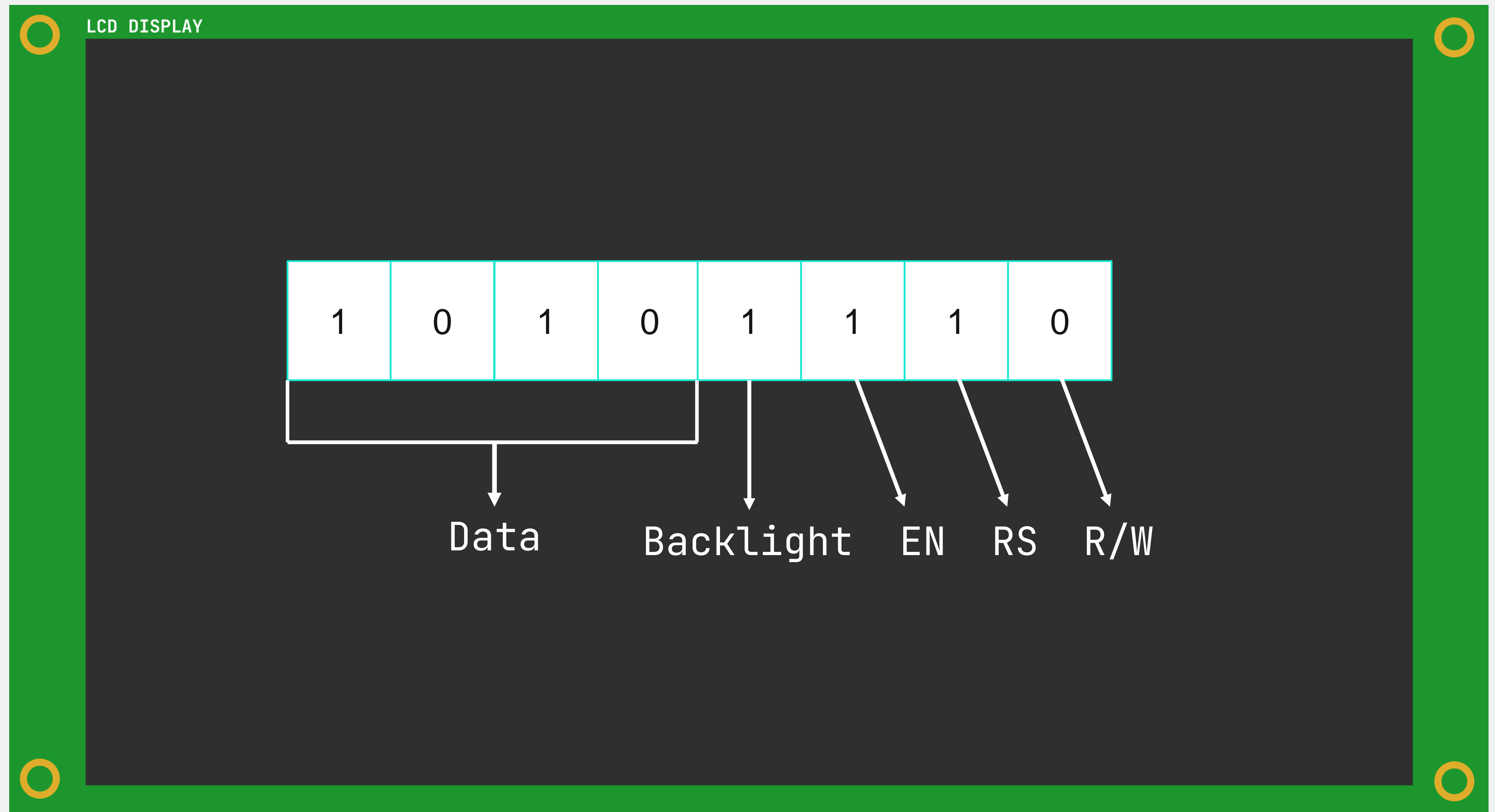
Schermo LCD

LCD DISPLAY



Utilizzando I2C possiamo ridurli a quattro!

Interpretazione dei bit



Schermo LCD

LCD DISPLAY

```
def write_data(self, data):  
    """Scrivo dati sull'LCD."""  
    byte = (MASK_RS |  
            (self.backlight << SHIFT_BACKLIGHT) |  
            (((data >> 4) & 0x0f) << SHIFT_DATA))  
    self.i2c.writeto(self.i2c_addr, bytes([byte | MASK_E]))  
    self.i2c.writeto(self.i2c_addr, bytes([byte]))  
    byte = (MASK_RS |  
            (self.backlight << SHIFT_BACKLIGHT) |  
            ((data & 0x0f) << SHIFT_DATA))  
    self.i2c.writeto(self.i2c_addr, bytes([byte | MASK_E]))  
    self.i2c.writeto(self.i2c_addr, bytes([byte]))  
    gc.collect()
```

Snippet di codice per scrivere dati sull'LCD



Perché scegliere PassChain?

PassChain vs Password Managers

PassChain

Password Managers

Dispositivo fisico poco vulnerabile e con credenziali criptate

Servizi cloud soggetti a continui tentativi di hacking

La connessione bluetooth lo rende compatibile con tutti i dispositivi moderni

La necessità di installare un programma apposito rende difficile utilizzarlo su dispositivi non personali

Password Manager fisici

LCD DISPLAY



Non hanno nessuna funzionalità smart, solo un pin di protezione



Grazie per l'attenzione!

Gruppo 13 - G. Spina, A. Montefusco, O. Szuba

Internet Of Things - 20/05/2022

