

UNIVERSITÀ DEGLI STUDI DI SALERNO

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Documentazione Progetto IoT

Annamaria Scermino, Anuar Zourhi, Gerardo Selce

TheBox

31 maggio 2025

Indice

1	Introduzione	2
2	Architettura del Sistema	2
3	Descrizione del Codice	2
3.1	File: main.ino	3

1 Introduzione

Il progetto sviluppato ha come obiettivo la realizzazione di uno smart caveaux. Il caveaux è dotato di un meccanismo di apertura che può essere controllato sia tramite un tastierino fisico posto all'esterno, sia da remoto attraverso una piattaforma di controllo. Il sistema integra inoltre sensori per la rilevazione di temperatura e umidità, con la capacità di attivare allarmi in caso di valori fuori soglia o tentativi di intrusione. Gli allarmi possono essere gestiti localmente o a distanza. Un semaforo posto accanto alla porta mostra lo stato del caveaux.

2 Architettura del Sistema

I componenti hardware utilizzati sono i seguenti:

- Un servomotore;
- Un tastierino numerico;
- Uno schermo OLED;
- Due breadboard;
- Due push-button;
- Un semaforo;
- Un sensore ad ultrasuoni;
- Un sensore di umidità e temperatura;
- Un buzzer passivo;
- Una ESP32
- Un relay

Dal punto di vista software, il sistema è stato sviluppato interamente in micropython. Per il controllo remoto e il monitoraggio in tempo reale è stata realizzata una dashboard interattiva con Node-RED, che consente di visualizzare lo stato del sistema, ricevere notifiche e inviare comandi. Inoltre alcune funzionalità, come l'apertura del caveaux, sono accessibili anche da un'app mobile. I protocolli utilizzati sono: MQTT ed I2C.

3 Descrizione del Codice

Spiegazione dettagliata del codice, divisa per moduli o file.

3.1 File: main.ino

```
1      #include <WiFi.h>
2      #include <PubSubClient.h>
3
4      // Connessione WiFi
5      const char* ssid = "NomeRete";
6      const char* password = "password";
7
8      // Setup
9      void setup() {
10         Serial.begin(115200);
11         WiFi.begin(ssid, password);
12         while (WiFi.status() != WL_CONNECTED) {
13             delay(500);
14             Serial.print(".");
15         }
16         Serial.println("Connesso!");
17     }
```

Listing 1: File principale per ESP32