

**Progetto Sistemi Embedded**

**Automatic Firedoor Alarm**

**A.F.A.**

***Prof. Daniele Peri***

***Corso di Laurea Magistrale Ingegneria Informatica (LM-32)***

***A cura di***

***Luca La Barbera***

***Salvatore Drago***

Sommario

[Descrizione progetto 2](#_Toc120633454)

[Introduzione 2](#_Toc120633455)

[**Descrizione componenti** 2](#_Toc120633456)

[**Configurazione** 2](#_Toc120633457)

[**Schema del sistema** 3](#_Toc120633458)

## **Descrizione progetto**

## **Introduzione**

L’obiettivo finale del nostro progetto è quello di realizzare un sistema di monitoraggio di temperatura umidità, rilevazioni di anomalie e incendi che permettono la chiusura immediata delle porte tagliafuoco di una sala server, consentendo l’isolamento delle aree durante gli incendi e preservando le altre sale macchine adiacenti dall’avanzare dell’incendio.

### **Descrizione componenti**

Questo è l’elenco dei componenti elettronici utilizzati per la realizzazione del progetto:

* x 1 Raspberry Pi 3B+
* x 1 Servo motore sg90
* x 1 Sensore di fiamma A-Z Delivery KY-026
* x 1 BreadBoard
* x 1 FT232RL USB Interfaccia seriale UART
* x 1 Mini USB Cable
* x 1 Sensore di umidità e temperatura DHT-11
* x 1 Active Buzzer
* x 1 RGB LED
* x1 LCD Display 16x02
* Resistenze 10 kΩ 1kΩ 220kΩ

### **Configurazione**

Si procede con la descrizione della configurazione iniziale e la descrizione del corretto avvio del dispositivo AFA.

il progetto è realizzato in FORTH su ambiente PijForthOS.  
PijForthOS è un interprete FORTH bare-metal per Raspberry Pi (modello B) che utilizza la console seriale RPi. Installando pijForthOS su una scheda SD nell'RPi, è possibile collegare la Raspberry a un'altra macchina utilizzando un cavo USB-seriale.

Quando l'RPi è acceso, un programma terminale sulla macchina host consente l'accesso alla console FORTH. In questo caso si utilizza il software ZOC8 Terminal.  
Tramite terminale si procede ad importare le librerie a disposizione, in seguito descritte.

#### Import Librerie

Descrizione delle librerie importate per l’attivazione e l’utilizzo del dispositivo AFA.

|  |  |  |
| --- | --- | --- |
| Ordine esecuzione | File.forth | Descrizione breve |
| 1° | 0-SE-ANS | File di configurazione iniziale del sistema |
| 2° | 1\_AUTOCONFIG | File di configurazione che inizializza i registri riconoscendo il tipo di dispositivo Raspberry Pi tra 3b+ e 4 |
| 3° | 2\_UTILS | File di configurazione che istanzia alcune funzioni di utilità, che serviranno dopo per l’implementazione delle funzionalità del dispositivo AFA |
| 4° | 3\_CONFSENS | File di configurazione dei sensori, abilitazione dei Pin GPIO e creazione delle variabili utili e degli array per la gestione dei sensori. |
| 5° | 4\_RGB\_LED | File che si occupa dell’implementazione dei comandi per RGB LED |
| 6° | 5\_THERMO\_SENSOR | File che si occupa dell’implementazione dei comandi per sensore di temperatura e umidità |
| 7° | 6\_BUZZER | File che si occupa dell’implementazione dei comandi per Active Buzzer |
| 8° | 7\_SERVO | File che si occupa dell’implementazione dei comandi per il Servo motore |
| 9° | 8\_LCD | File che si occupa dell’implementazione dei comandi per accendere e pilotare LCD |
| 10° | 9\_BUTTONS | File che si occupa dell’implementazione dei comandi per utilizzare i bottoni |
| 11° | 10\_FLAMESENS | File che gestisce il sensore di fiamma |
| 12° | 11\_MAIN | File principale per il funzionamento del sistema AFA. |

#### Schema di collegamento

### **Schema del sistema**

Si procede alla stesura dei dettagli del sistema dal punto di vista progettuale

\+++++++Schema collegamenti sistema +++++++++\

#### Descrizione

\+++++++Schema collegamenti +++++++++\

Il collegamento delle componenti è stato effettuato seguendo lo schema precedentemente descritto.

In particolare, le componenti interagiscono fra di loro nel seguente modo:

* Il Raspberry Pi coordina l’intero sistema e contiene la logica del software
* La Breadboard costituisce il ponte di collegamento per i servomotori ed il sensore IR
* L’alimentazione dell’intero sistema è fornita dal Raspberry Pi, che a sua volta è alimentato tramite il proprio alimentatore da una presa di corrente

#### Componenti

In questa sezione descriviamo le componenti utilizzate, il loro scopo all’interno del nostro progetto e il software da noi scritto per il loro funzionamento.

##### **Raspberry Pi 3b+**



Il Raspberry Pi è un computer a scheda singola (single-board computer) progettato dalla Raspberry Pi Foundation. Si basa sul system-on-chip (SoC) BCM2837B0, che include un processore ARMv8 quad-core a 64 bit da 1,4 GHz, RAM di circa 1 GB e una potente GPU VideoCore IV. Raspberry Pi con i suoi 40 pin GPIO funziona anche come controller programmabile in un'ampia varietà di applicazioni di robotica ed elettronica.

##### **Breadboard**



La Breadboard rappresenta un mezzo per realizzare montaggi di circuiti elettronici senza saldature. È costituita da una basetta provvista di una serie di fori disposti secondo righe e colonne e distanziati del passo standard di 2,54 mm, tipico dei pin dei circuiti integrati.

I fori di una colonna, generalmente 5, sono internamente collegati fra loro mediante una barretta metallica a molla, ma non con i fori delle colonne adiacenti.

Lungo i due lati maggiori della basetta sono disposte due file di fori (dette binari di alimentazione) per i collegamenti di alimentazione e di terra.

##### **FT232RL USB Interfaccia Seriale UART**



È utilizzato per collegare la Raspberry con interfaccia seriale (UART) a un'interfaccia USB del proprio PC, garantendo robustezza in trasmissioni ad alta velocità. È dotata di 6 pin:

* DTR: Data Terminal Ready, un'uscita utilizzata per il controllo di flusso
* RX: Serial Data Receive Pin
* TX: Serial data Transmit Pin
* VCC: Uscita in tensione positiva
* CTS: Clear To Send, un ingresso utilizzato per il controllo di flusso
* GND: Messa a terra o 0V

Per la maggior parte dei casi, basta connettere i pin RX, TX, GND ai rispettivi pin della macchina target.

##### **SG90**



SG90 è un servomotore piccolo e leggero con un'elevata potenza di uscita. Il servo può ruotare di circa 180 gradi (90 in ciascuna direzione).

Il codice che gestisce questo componente è contenuto nel file 7\_SERVO.forth, qui viene implementato un metodo FIREDOOR che prende in input:

1. Il numero della Stanza
2. una costante ( APRI/CHIUDI), dichiarate all’inizo del file

Questo metodo quando invocato consente di attivare il servomotore e ruota di 90° la piccola asta montatagli sopra. Questo simula l’attivazione di attuatori che in caso di emergenza servono a chiudere le porte tagliafuoco.