



: GAS DET ;

Forth Embedded Systems Project



WALTER MADONIA

Anno accademico 2022/23

Matr. 0757990

walter.madonia@community.unipa.it

SOMMARIO

Descrizione del Progetto	2
Ambiti applicativi e Sviluppi futuri	2
Componenti del progetto	3
Componenti Hardware	3
Componenti Software	3
Descrizione Componenti	4
Target: Raspberry Pi 3A+	4
Modulo con sensore per Gas MQ-2	5
Analog To Digital Converter (ADC) ADS1115 16bit	6
FT232RL USB Interfaccia Seriale UART	6
Attuatori: Buzzer acustico e LEDs.....	7
Schema del Progetto	8
Collegamenti componenti	8
Schema Header GPIO del Raspberry Pi 3A+	9
Altri collegamenti	9
Funzionamento del Sistema	10
GPIO del Raspberry Pi 3A+	10
Sensore MQ-2 e ADC ADS1115	10
Target e ADS1115 attraverso il bus I2C.....	11
Configurazione ADS1115	11
Rilevazione dei Gas.....	13
Calibrazione e Limite di Allerta.....	14
Flusso degli Eventi	15
Eventi degli Attuatori.....	15
Procedura di utilizzo del Sistema	16
Preparazione dell'ambiente di sviluppo	16
Preparazione del Target	16
Preparazione PC	17
Procedura con LINUX e picocom-3.1.....	17
Foto Testing	19

Descrizione del progetto

Il progetto consiste nella realizzazione di un sistema di rilevazione di Gas altamente infiammabili in ambienti chiusi con appositi componenti per la segnalazione acustica e visiva alla presenza dei Gas rilevati. Il sistema inoltre è dotato di funzionalità di auto-calibrazione in base alle condizioni ambientali circostanti e permette la rilevazione di diversi Gas infiammabili più comuni in commercio sia ad impiego casalingo che industriale come ad esempio il Butano, Metano e Gas Propano Liquido (GPL) ed inoltre permette la rilevazione di fumo causato dalle combustioni di qualsiasi genere e/o tipo.

Dopo aver appositamente preconfigurato il sistema basterà semplicemente immettere il comando apposito per l'avvio digitato da Tastiera per inizializzare il sistema che eseguirà:

- **Prima fase:** configurazione dei componenti collegati al sistema.
- **Seconda fase:** auto-calibrazione ambientale circostante prelevando dei dati di test in completa assenza di Gas con la sola presenza di "Aria Pulita".
- **Terza fase:** modalità di scansione segnalando opportunamente attraverso segnali visivi e acustici l'eventuale presenza o assenza del Gas infiammabile.

A causa delle limitazioni dello sviluppo e realizzazione prettamente accademico il seguente sistema è stato realizzato in piccola scala utilizzando come piattaforma principale il dispositivo Target **Raspberry Pi 3A+** che attraverso i componenti circuitali, computazionali e periferiche di cui è dotato permette la realizzazione di sistemi informatici completamente programmabili ed automatizzati.

Ambiti applicativi e Sviluppi futuri

Come precedentemente detto il seguente progetto è realizzato su piccola scala con impiego in ambito domestico ma ideato e progettato per essere implementato in ambiti applicativi di scala superiore come ad esempio nel settore aziendale e industriale per la sicurezza degli ambienti di lavoro, settore dei trasporti pubblici e privati oppure nel settore automobilistico per vetture alimentate a Gas infiammabili. Inoltre è possibile implementare il seguente sistema all'interno di un altro Macro sistema più complesso a supporto delle operazioni di quest'ultimo.

Componenti del progetto

Componenti Hardware

I componenti elettronici e computazionali utilizzati per la realizzazione del progetto sono i seguenti:

- Raspberry Pi 3A+ 512MB RAM
- Modulo con sensore per Gas MQ-2
- Analog To Digital Converter (ADC) ADS1115 16bit Interfaccia I2C
- FT232RL USB Interfaccia Seriale UART
- Buzzer segnalatore acustico da 3V fino a 24V
- BreadBoard e Cavi connessione M/F e M/M
- 3 LED vari colori
- 3 Resistori ceramidi da 200 ohm
- Scheda MicroSD 16GB
- Cavo MicroUSB
- Cavo MiniUSB
- Personal Computer (PC)

Componenti Software

Per la programmazione e lo sviluppo software del progetto sono stati adottati due componenti software contemporaneamente per permettere la programmazione interattiva attraverso un dispositivo Host (PC) direttamente sul Target.

Installato nel Target:

- **pijFORTHos**: sistema operativo con interprete FORTH basato su linguaggio con grammatica Post Fissa che permette un'alta interazione con l'Hardware sottostante per soluzioni Bare Metal e programmazione di basso livello.

A seconda del sistema operativo installato sul Host è necessario utilizzare uno dei due componenti software per l'interazione col dispositivo Target attraverso l'interfaccia seriale UART.

Windows e MacOS:

- **ZOC8 Terminal**

Linux:

- **picocom-3.1**

In alternativa è possibile utilizzare un qualsiasi software open-source che permette la comunicazione seriale tramite interfaccia UART tuttavia in questa documentazione non verranno illustrate altre procedure se non per i software sopra citati.

Descrizione Componenti

Target: Raspberry Pi 3A+

Il **Raspberry Pi 3A+** è un Single Board Computer di dimensioni ridotte, progettato per essere economico, versatile e altamente programmabile. (Fig. 1)

La scheda utilizza un System on a Chip (SoC) **Broadcom BCM2837B0**, che integra un processore quad-core **ARM Cortex-A53 a 1,4 GHz** e una GPU **VideoCore IV a 400 MHz**. Questo SoC è stato sviluppato specificamente per il Raspberry Pi, ed è stato progettato per offrire un equilibrio tra prestazioni elevate e consumo energetico ridotto.

Inoltre, la scheda dispone di una porta HDMI per la connessione a un display, una porta USB 2.0 per la connessione di periferiche esterne, una porta camera per la connessione di una fotocamera e una porta display per la connessione di un display a basso costo.

Il Raspberry Pi 3A+ dispone di un **Header General Purpose Input/Output** a 40 pin (Fig. 2) che consente di connettere la scheda a una vasta gamma di dispositivi esterni, come sensori, attuatori e display. I pin GPIO possono essere programmati per funzionare come ingressi o uscite digitali consentendo al Raspberry Pi 3A+ di acquisire dati da sensori come la temperatura, la pressione o la luminosità. L'Header GPIO comprende anche pin per la comunicazione seriale, **I2C** e **SPI**, che possono essere utilizzati per la comunicazione con dispositivi esterni come sensori di temperatura, accelerometri, schermi OLED e molto altro ancora.

Il dispositivo è adatto a una vasta gamma di progetti, dall'automazione domestica alla robotica, dalla prototipazione di prodotti all'istruzione sulla programmazione. Grazie alla sua flessibilità, è possibile utilizzare la scheda per creare un computer completo, un media center, un server di gioco e molto altro ancora. Inoltre, il Raspberry Pi 3A+ è compatibile con una vasta gamma di sistemi operativi, tra cui Raspberry Pi OS, Ubuntu, Fedora ed altri.

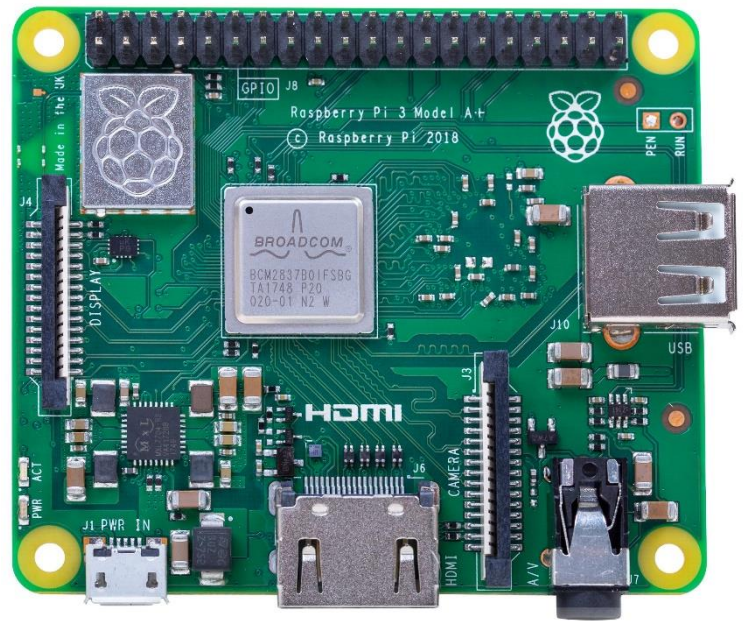


Fig. 1 – Raspberry Pi 3A+ e la sua PCB

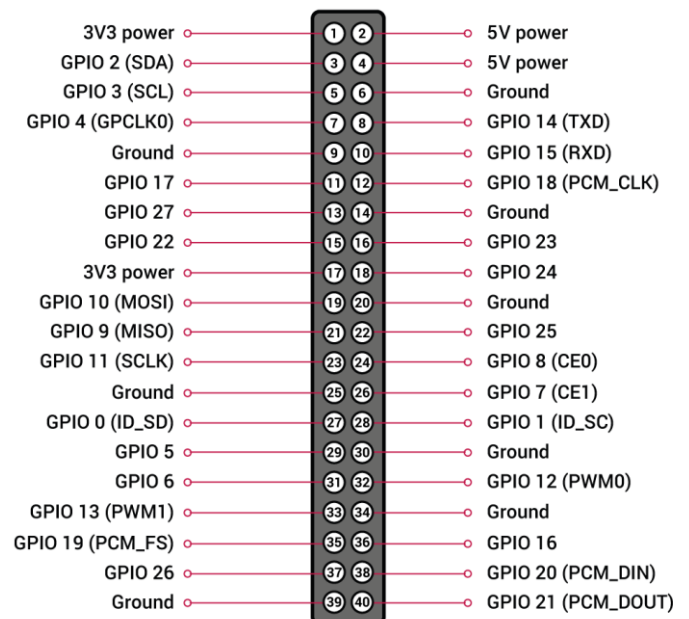
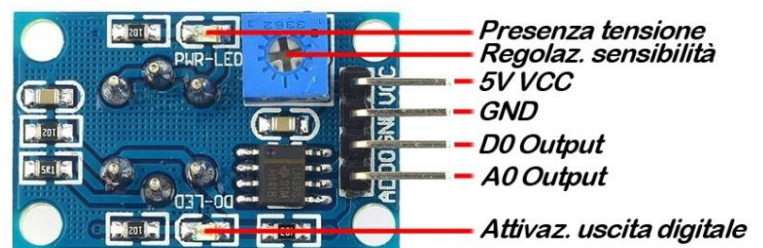
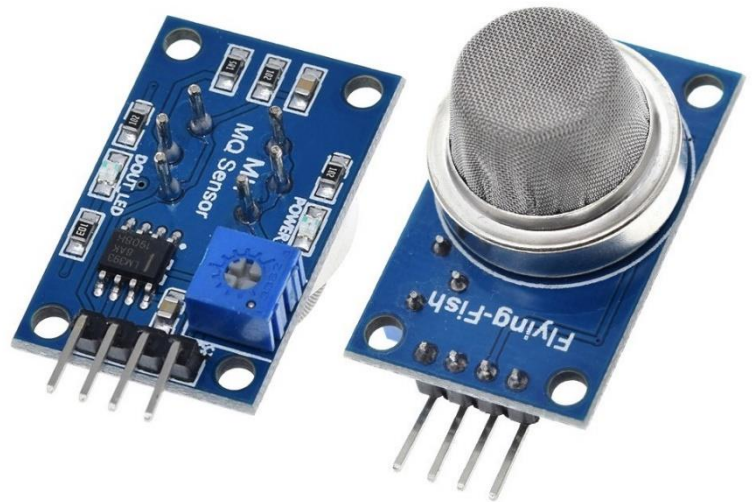


Fig. 2 – Schema Header GPIO con descrizione di ogni singolo pin.

Modulo con sensore per Gas MQ-2

Il sensore utilizzato per la realizzazione del progetto è un Modulo con a bordo un **sensore MQ-2** che permette la rilevazione di Gas infiammabili più comuni in commercio sia per utilizzo domestico che industriale. Oltre al sensore nel Modulo vi è la presenza di altri elementi circuitali che permettono l'uscita in Output di segnali sia Analogici che Digitali per poter permettere la lettura delle informazioni che determinano la presenza o meno del Gas e della sua quantità misurata in **Parti Per Milione (PPM)**. Inoltre vi è nel Modulo la presenza di un potenziometro che permette la regolazione della **Resistenza di Carico RL** e dunque permette di regolare la sensibilità dell'individuazione del Gas attraverso il valore della tensione in uscita dal sensore MQ-2, vi è anche la presenza di 2 LED che segnalano la corretta circolazione di corrente elettrica all'interno del Modulo (**PWR_LED**) sempre acceso e la segnalazione del segnale Digitale (**DO_LED**) che si attiva soltanto quando viene individuata la concentrazione massima di Gas in PPM (per la realizzazione di questo progetto tale segnale viene ignorato in quanto non permette un controllo sensibile all'individuazione di Gas).



All'interno della capsula metallica vi è Il sensore MQ-2 per rilevare la presenza di diversi tipi di gas come:

- **Butano;**
- **Metano;**
- **Propano;**
- **Alcohol allo stato gassoso;**
- **Idrogeno;**
- **GPL;**
- **Fumo prodotto da combustioni.**

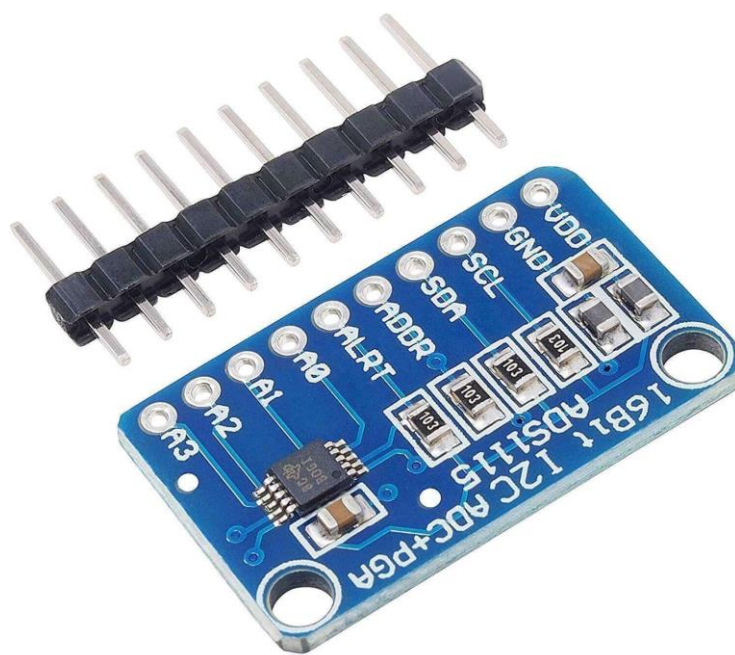
Il sensore MQ-2 è costituito da un elemento sensibile semiconduttore di **Diossido di Stagno** (SnO_2) che viene riscaldato da un circuito integrato a bassa potenza. Quando il gas entra in contatto con l'elemento sensibile esso reagisce con il Diossido di Stagno alterando la resistenza elettrica dell'elemento. Il circuito di controllo del sensore MQ-2 rileva la variazione di resistenza e produce una tensione proporzionale alla concentrazione di gas rilevata e tale informazione viene portata in uscita attraverso un **segnale Analogico**.

Analog To Digital Converter (ADC) ADS1115 16bit

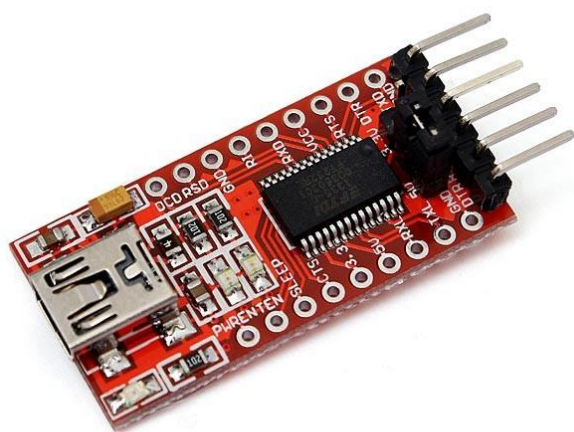
L'**ADS1115** è un convertitore analogico-digitale (ADC) a **16 bit** di precisione che consente di acquisire segnali analogici con una precisione elevata ed erogare in uscita un segnale Digitale di grandezza 16 bit. Esso è in grado di convertire fino a quattro segnali analogici simultaneamente e di trasmetterli in digitale attraverso la comunicazione **I2C**.

L'**ADS1115** è dotato di una tensione di riferimento interna, ma è anche possibile utilizzare una tensione di riferimento esterna per una maggiore precisione. Dispone inoltre di un'ampia gamma di guadagni di amplificazione selezionabili che consentono di adattare la sensibilità dell'ADC al segnale in ingresso.

Il dispositivo è compatibile con molte schede di sviluppo, come il **Raspberry Pi**, ed è spesso utilizzato in applicazioni che richiedono una conversione ADC di alta precisione, come il controllo di sensori di temperatura, la rilevazione di corrente e tensione, e altro ancora. Grazie alla sua elevata precisione, l'**ADS1115** è in grado di fornire letture affidabili e accurate anche per segnali analogici molto deboli o rumorosi, rendendolo un componente molto utile per molte applicazioni in campo industriale, scientifico e hobbistico.



FT232RL USB Interfaccia Seriale UART



La **USB to Serial Adapter with FT232RL** permette di collegare i PC con qualsiasi sistema a microcontrollore attraverso la porta USB. Usa l'integrato **FT232RL** della FTDI, dotato di buffer in ricezione da 128 byte e buffer in trasmissione da 256 byte che garantiscono robustezza in trasmissioni ad alta velocità. Oltre ai segnali **TX** e **RX**, sono presenti anche le linee CTS, RTS e le altre linee di Handshaking. Collegando la scheda alla porta USB, il PC la riconoscerà come una **VirtualCOM Port (VCP)** attraverso la quale possiamo stabilire la connessione con il target emulando la porta seriale RS232, senza bisogno di alcuna modifica. Per il collegamento è sufficiente fornire la linea di alimentazione dalla Breadboard e collegare le linee segnale TX ed RX alle rispettive RX e TX del target.

Attuatori: Buzzer acustico e LEDs

Gli attuatori impiegati per la realizzazione del progetto sono 3 LED colorati per la segnalazione visiva dello stato di rilevazione di Gas da parte del sensore MQ-2 e da un Buzzer acustico.



Un **Buzzer acustico** da 3V a 24V è un dispositivo elettronico che produce un suono acustico quando viene alimentato con una tensione compresa tra 3 e 24 volt. Esso è costituito da un oscillatore elettronico, un'unità di controllo e un elemento di produzione del suono come un **altoparlante**. Il buzzer acustico può essere utilizzato in diversi progetti e applicazioni, come allarmi di sicurezza, indicatori acustici per segnali elettronici, suonerie per telefoni e dispositivi di segnalazione.

La segnalazione luminosa a 3 LED (Fig. 3), di cui ognuno è collegato in serie ad un Resistore da 200 ohm, sono configurati come segue:

- LED **BLU** (Lampeggiante): segnala lo svolgimento della fase di calibrazione del sistema.
- LED **VERDE** (Fisso): segnala l'assenza di Gas nell'aria.
- LED **ROSSO** (Lampeggiante): segnala la presenza di Gas captata del sensore.

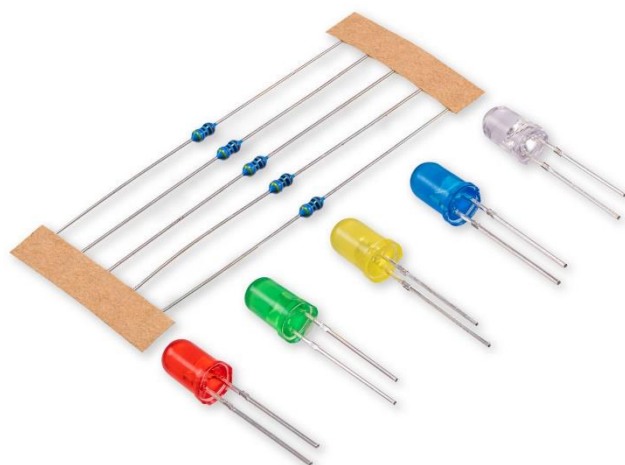
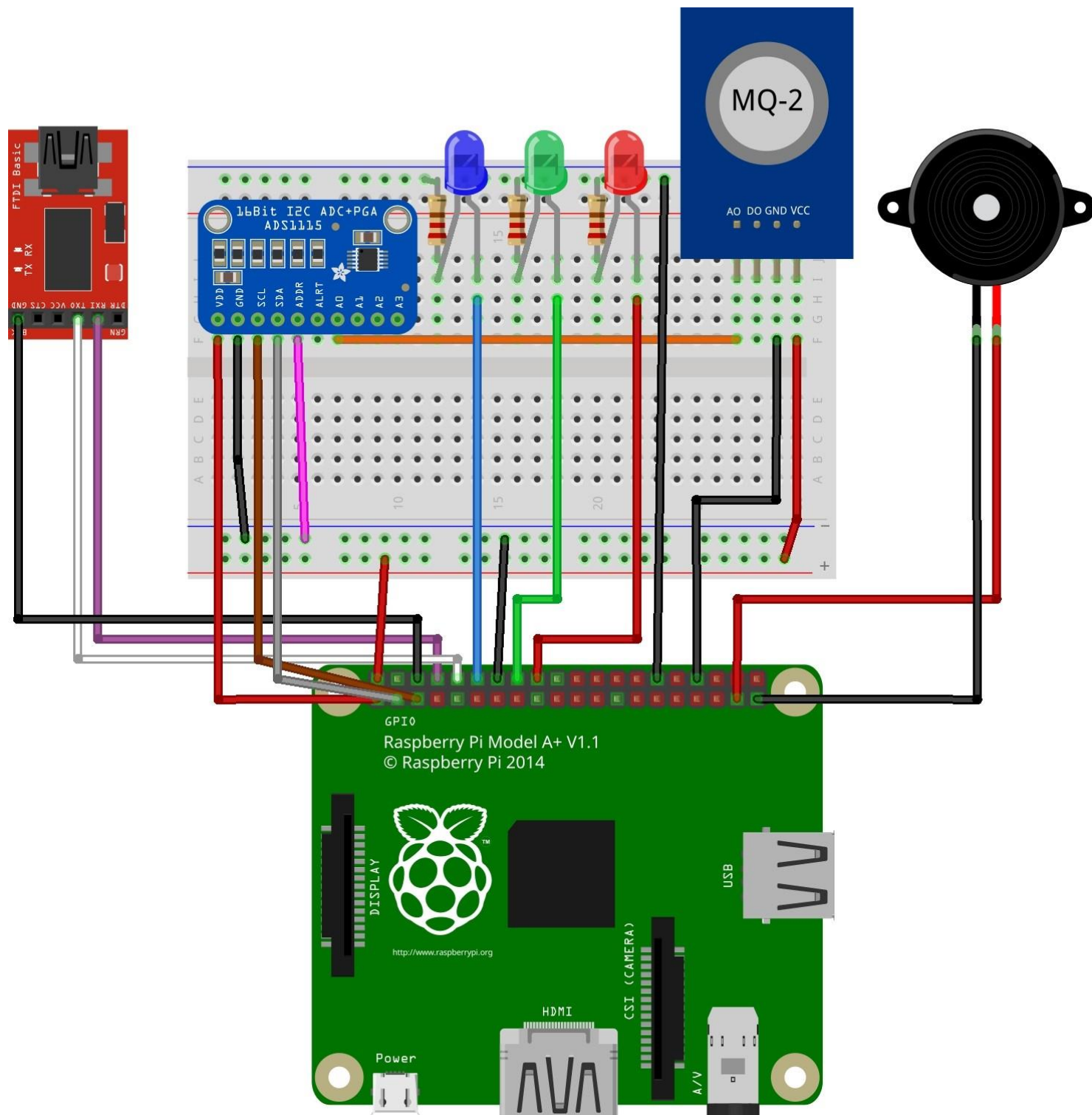


Fig. 3 – Immagine a scopo illustrativo. Per la realizzazione del progetto sono stati impiegati soltanto 3 LED: Blu, Rosso e Verde.

Schema del progetto

Collegamenti componenti



Schema Header GPIO del Raspberry Pi 3A+

CONN.	HEADER	PIN		HEADER	CONN.
ADC - VCC	3V3 Power	1	2	5V Power	MQ2 - VCC
ADC - SDA	GPIO 2 (SDA)	3	4	5V Power	
ADC - SCL	GPIO 3 (SCL)	5	6	Ground	FTDI - GND
	GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)	FTDI - RX
	Ground	9	10	GPIO15 (RXD)	FTDI - TX
	GPIO 17	11	12	GPIO 18 (PCM_CLK)	BLUE LED+
	GPIO 27	13	14	Ground	ADC - GND
	GPIO 22	15	16	GPIO 23	GREEN LED+
	3V3 Power	17	18	GPIO 24	RED LED+
	GPIO 10 (MOSI)	19	20	Ground	
	GPIO 9 (MISO)	21	22	GPIO 25	
	GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)	
	Ground	25	26	GPIO 7 (CE1)	
	GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)	
	GPIO 5	29	30	Ground	LEDs - GND
	GPIO 6	31	32	GPIO 12 (PWM0)	
	GPIO 13 (PWM1)	33	34	Ground	MQ2 - GND
	GPIO19 (PCM_FS)	35	36	GPIO 16	
BUZZER - VCC	GPIO 26	37	38	GPIO 20 (PCM_DIN)	
BUZZER - GND	Ground	39	40	GPIO 21 (PCM_DOUT)	

Altri collegamenti

CONN.	CAVO	CONN.
MQ2 - AO		ADC - A0
ADC - ADDR		ADC - GND

Funzionamento del Sistema

Dopo aver eseguito correttamente la prima fase preliminare connettendo appositamente tutti i componenti seguendo gli schemi visti precedentemente, attraverso la **FT232RL** è possibile instaurare una connessione tra il Target e il PC attraverso la connessione seriale **Universal Asynchronous Receiver-Transmitter (UART)** che permetterà la programmazione interattiva direttamente sul Target utilizzando il Terminale FORTH attraverso il **pijFORTHos** installato nella MicroSD appositamente formattata per la corretta lettura da parte del Target.

GPIO del Raspberry Pi 3A+

Il Target in questione possiede diverse porte **General Purpose Input/Output** di cui la maggior parte sono collegate ai 40 pin presenti sull'Header del PCB. Di default le GPIO sono impostate in modalità **Input** dunque è necessario andare a cambiare la modalità delle GPIO interessate in base alle **Funzioni** per il quale sono destinate.

Oltre alle funzioni **Input** e **Output** le GPIO posseggono delle **Alternative Functions** che variano a seconda della GPIO. Per il seguente sistema le GPIO collegate ai componenti sono state impostate secondo il seguente schema:

GPIO	FUNCTION	CONN.
GPIO 18, 23, 24, 26	OUTPUT	LEDs e Buzzer Acustico
GPIO 2	ALTO (SDA)	ADC - SDA
GPIO 3	ALTO (SCL)	ADC - SCL
GPIO 14	ALTO (TXD)	FTDI - RX
GPIO 15	ALTO (RXD)	FTDI - TX

Si nota che le GPIO 2, 3, 14, 15 sono necessarie per la ricezione/invio di dati sia alla **FT232RL** che al **ADC ADS1115** precisamente:

- **GPIO 2, 3** sono necessarie per utilizzare una connessione attraverso il bus **I2C** tra il Target e il **ADS1115**.
- **GPIO 14, 15** sono necessarie per utilizzare il collegamento **UART** tra il Target e il PC attraverso la **FT232RL**.

Sensore MQ-2 e ADC ADS1115

Come descritto precedentemente il modulo con Sensore **MQ-2** porterà in uscita un segnale Analogico che rappresenta l'informazione relativa al **valore di tensione in uscita** al variare della resistenza dell'elemento sensibile ai Gas infiammabili. Tale segnale in uscita verrà passato in ingresso al **ADC ADS1115** che si occuperà di convertire il segnale analogico in digitale di lunghezza 16 bit.

Target e ADS1115 attraverso il bus I2C

Al fine di poter ricevere ed analizzare il valore ottenuto dalla conversione effettuata dal **ADS1115** è necessario instaurare una connessione tra il Target e il ADC utilizzando una comunicazione attraverso il bus **Inter-Integrated Circuit (I2C)** ovvero un protocollo di comunicazione seriale utilizzato per collegare tra loro dispositivi elettronici su una stessa scheda di sviluppo o su una stessa rete di dispositivi basato su un architettura di tipo **Master-Slave** dove in questo caso il Master sarà il Target e lo Slave l'ADS1115.

Per effettuare un collegamento **I2C** tra due dispositivi, sono necessari almeno due cavi: SDA (**Serial Data**) e SCL (**Serial Clock**). Il cavo SDA viene utilizzato per trasferire i dati in formato seriale, mentre il cavo SCL viene utilizzato dal Master per trasmettere il segnale di clock che sincronizza la trasmissione dei dati. Tuttavia è possibile inviare/ricevere solamente 1 Byte alla volta (8 bit) dove per 1 impulso di clock sul SCL viene trasmesso 1 bit sul SDA, alla trasmissione completa del Byte viene inviato un messaggio da parte dello Slave di **Acknowledgement (ACK)** oppure di **Negative-Acknowledgement (NACK)** in caso di errore durante la trasmissione.

Configurazione ADS1115

Come riportato nella Documentazione del ADS1115 (non presente in questo documento) prima di utilizzare e ricevere i valori digitali generati dalla conversione del segnale Analogico prodotto dal sensore MQ-2, è necessario configurare l'ADC andando ad interagire con i due **Registri da 16 bit** del ADS1115 che sono precisamente:

- **Config Register:** registro dedicato alla configurazione della tipologia di conversione ed altri parametri.
- **Conversion Register:** registro che conterrà all'interno il valore digitale di 16 bit (2 Byte) convertito dal segnale Analogico.

Per poter accedere ad entrambi i registri sarà necessario utilizzare un **Address Pointer** ovvero un valore di 8 bit specifico che identifica in quale registro vogliamo accedere dell'ADS1115. Prima di procedere con la ricezione del valore convertito all'interno del Conversion Register è necessario configurare al primo avvio l'ADC attraverso il Config Register che possederà per ogni bit all'interno del registro tutta una serie di impostazioni dedicate alla tipologia di conversione come:

- **Inizializzare una nuova conversione se è in Power State Mode**
- **Configurazione Input Multiplexer**
- **Configurazione Gain Amplifier**
- **Modalità di conversione**
- **Data rate**
- **Modalità comparatore**
- **Polarità comparatore**
- **Latching comparatore**
- **Coda comparatore e disabilitazione**

In questo sistema l'ADC è stato configurato come riportato dalla tabella:

IMPOSTAZIONE	VALORE
Input Multiplexer	AIN(p) = AIN0 and AIN(n) = GND
Gain Amplifier	+/- 2.048 V
Modalità di conversione	Conversione continua
Data rate	128 SPS (Sample Per Second)
Modalità comparatore	Traditional
Polarità comparatore	Active Low
Latching comparatore	Non-latching
Coda comparatore e disabilitazione	Disabilitata

Dunque attraverso il bus **I2C** sarà possibile mettere in comunicazione il Target e l'ADS1115 che avverrà attraverso 3 trasmissioni differenti: i primi due sono preliminari da eseguire solo al primo avvio del sistema e l'ultimo, che verrà effettuato continuamente, permetterà per ogni istante di tempo la ricezione del valore digitale convertito.

Le trasmissioni da effettuare saranno:

- 1) **Target invia 3 Byte all'ADC** di cui il **Primo Byte** sarà l'Address Pointer per accedere al Config Register e il **Secondo e Terzo Byte** saranno il valore di 16 bit che permetterà di configurare l'ADC come riportato precedentemente.
- 2) **Target invia 1 Byte all'ADC** che conterrà l'Address Pointer per poter richiedere l'accesso al Conversion Register.
- 3) **Target riceve 2 Byte dall'ADC** che insieme rappresentano il valore di 16 bit generato dalla conversione.

Una volta che viene ricevuto il valore convertito per poterlo rappresentare come il **valore di tensione** di uscita dal sensore MQ-2 è necessaria la trasformazione:

$$Conversion\ Voltage = 2,048\ V \times \frac{ConvertValue}{2^{15}}$$

ConvertValue è il valore ricevuto dall'ADS1115 mentre **2,048 V** è il valore del **Gain Amplifier** che è stato impostato nel Config Register (in questo caso di +/-2,048 V)

Rilevazione dei Gas

Una volta che è stata instaurata la connessione tra il Target e il Modulo MQ-2 attraverso l'ADC è necessario utilizzare le informazioni relative al **valore di tensione** in uscita dal sensore MQ-2 al fine di poter stabilire tutti quei parametri necessari per il Rilevamento sensibile del Gas e definire una **soglia di avvertenza** che a sua volta permetterà l'attivazione degli attuatori e la segnalazione della presenza di Gas infiammabile.

Analizzando il circuito elettrico che definisce la struttura del Modulo MQ-2 si evidenzia che esso non è altro che un **Partitore di Tensione** ovvero un tipo di circuito costituito da due o più componenti passivi collegati in serie ai capi dei quali, se viene applicata una tensione, essa si ripartirà sulle stesse componenti in base al loro valore. In questo caso i due elementi in questione saranno:

- L'elemento sensibile di **Diossido di Stagno** che produrrà una Resistenza **RS** che varierà in base alla concentrazione di Gas presente nel sensore.
- Il **Potenziometro** presente nel modulo che invece produrrà una Resistenza di Carico **RL** che rimane fissa.

Dalla documentazione del sensore MQ-2 viene riportato il grafico dove nell'asse dell'ordinate vi è il **Rapporto** tra la Resistenza del Sensore **RS** con la Resistenza **R0**, che sarebbe la resistenza dell'elemento sensibile alla sola presenza di Aria Pulita e dunque con totale assenza di Gas infiammabili, mentre nell'asse delle ascisse vi è la concentrazione di Gas in **PPM** (Parti Per Milione). Come si evince dal grafico (Fig.4) per ogni Gas infiammabile è definita una Curva che definisce l'andamento decrescente del Rapporto **RS/R0** all'aumentare dei PPM del Gas e dunque all'aumentare della concentrazione di Gas rilevata la Resistenza dell'elemento sensibile diminuisce e quindi ne consegue che il **Valore di tensione in uscita sarà più elevato**. Il valore di **R0** viene ricavato dal valore di **RS/R0** in presenza di Aria Pulita che nel grafico è riportato come valore di circa **9,83**.

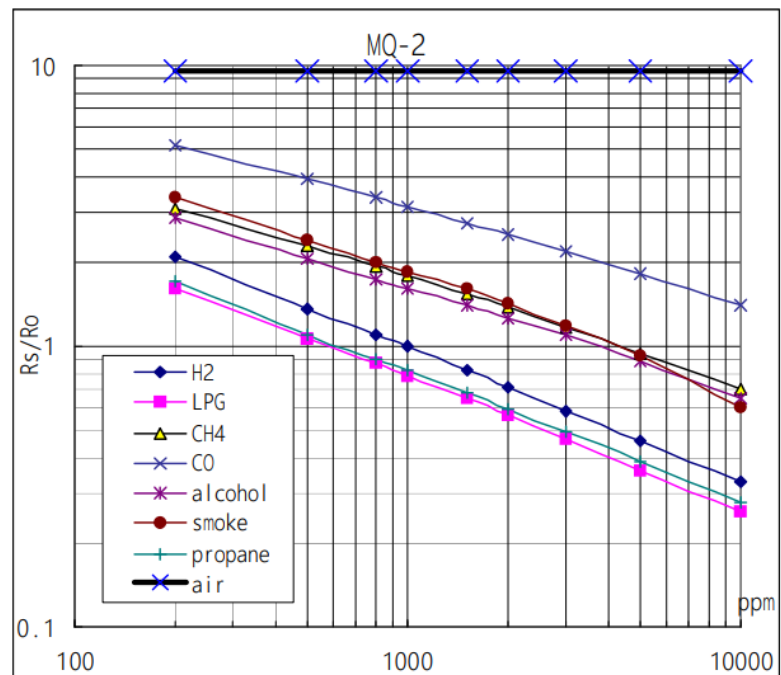


Fig.4 – Grafico delle Curve dei Gas che mette in relazione il Rapporto R_S/R_0 e i PPM dei Gas.

Dunque con la formula inversa **$R_0 = RS/9,83$** è possibile stabilire il valore di riferimento in assenza di Gas. Tuttavia essendo dipendente dal valore di **RS** quest'ultimo viene calcolato attraverso la **formula del Partitore di Tensione** che mette in relazione il Rapporto tra le due Resistenze del Circuito RS/RL con i valori di tensione in Ingresso ed in Uscita.

$$\frac{RS}{RL} = \frac{V_{ingresso} - V_{uscita}}{V_{uscita}}$$

Dove il valore di **RL** è costante in quanto rappresenta la Resistenza di Carico del Potenzimetro mentre **$V_{ingresso}$** è un valore di tensione fisso ovvero di 2.048V che sarebbe il valore del VCC del modulo MQ-2 riscaldato dal Gain Amplifier del ADC.

Dunque **RS** dipenderà soltanto dal **V_{uscita}** che sarà generato dalla conversione dell'ADC.

Calibrazione e Limite di Allerta

Per ragioni fisiche legate all'elemento sensibile del sensore MQ-2 il valore di **RS** in assenza di Gas è soggetto alle condizioni ambientali quali **Temperatura** dell'ambiente e **% di umidità** nell'aria dunque è necessaria una fase di **Calibrazione** in base all'ambiente circostante andando a calcolare come **V_{uscita}** un Valore Medio tra 100 valori di Tensioni calcolati dall'ADC in ambiente privo di Gas e ricavare quindi il Valore di **R0** attraverso le formule precedentemente viste.

Una volta trovato il valore di **R0** legato ad uno specifico ambiente bisogna ora definire dei **Limiti di Allerta** che permettono l'individuazione del Gas e quindi una Soglia minima che determina l'esecuzione degli eventi che attiveranno gli attuatori.

A causa dei limiti progettuali del sensore MQ-2 che non gli permettono una calibrazione simultanea per la rilevazione di più Gas contemporaneamente, ai fini dimostrativi accademici è stato scelto come Gas per la calibrazione il **GPL** (o LPG in inglese) che è un Gas infiammabile ampiamente diffuso in commercio presente all'interno dei comuni **Accendini** liberamente vendibili presso i Tabacchi. Come riportato nel grafico precedente (Fig.4) per la curva del **LPG** il valore minimo di **PPM** riporta in ordinata il valore del rapporto **RS/R0** di circa **1,62**.

Dunque, avendo calcolato precedentemente **R0** attraverso la fase di calibrazione, dalla formula:

$$\frac{RS}{R_0} = 1,62$$

Attraverso il calcolo delle formule inverse si può calcolare il valore di **RS** e di conseguenza dalla formula del Partitore di Tensione si può ricavare il valore della Tensione in Uscita che sarà dunque il nostro **Limite di Avvertenza**.

Flusso degli Eventi

Descritti gli aspetti tecnici del sistema adesso è possibile descrivere il **Flusso degli Eventi** che definiscono tutte le fasi del Sistema:

- 1) **Configurazione del Sistema:** il sistema predispone la configurazione del Target attraverso le GPIO, la configurazione degli attuatori e la comunicazione **I2C** con l'ADS1115.
- 2) **Configurazione dell'ADC:** il sistema abilita il bus I2C e vengono effettuate tutte le trasmissioni necessarie per la configurazione e l'abilitazione dell'ADC e dunque la ricezione continua dei Valori convertiti.
- 3) **Calibrazione del Sensore:** si effettua una media di 100 valori convertiti dall'ADC e si calcola il Valore di R_0 in assenza di Gas.
- 4) **Calcolo Soglia minima di attivazione:** trovato R_0 si calcola il valore di R_S e di conseguenza della Tensione in uscita del sensore al valore minimo di PPM del Gas infiammabile.
- 5) **Scansione di controllo:** il sistema è attivo e calibrato completamente e dunque vengono prelevati continuamente i valori convertiti dall'ADC e confrontati con la Soglia minima di attivazione che se uguali o superiori attiveranno gli attuatori per la segnalazione.

Eventi degli Attuatori

I 4 attuatori presenti nel sistema ovvero i 3 LED e il Buzzer seguiranno delle azioni ben specifiche in base all'evento specifico rilevato dal sistema.

CALIBRAZIONE	
ATTUATORE	STATO
LED BLU	BLINK
LED VERDE	OFF
LED ROSSO	OFF
BUZZER	OFF

ASSENZA GAS ($V_{ADC} < Soglia$)	
ATTUATORE	STATO
LED BLU	OFF
LED VERDE	ON
LED ROSSO	OFF
BUZZER	OFF

PRESENZA GAS ($V_{ADC} \geq Soglia$)	
ATTUATORE	STATO
LED BLU	OFF
LED VERDE	OFF
LED ROSSO	BLINK
BUZZER	BLINK

Procedura di utilizzo del Sistema

Preparazione dell'ambiente di sviluppo

Prima di poter effettivamente utilizzare il Sistema sono necessari una serie di passaggi per la preparazione dell'ambiente di sviluppo sia del Target che del PC che utilizzeremo per effettuare la programmazione Interattiva utilizzando il protocollo di comunicazione **UART** della **FT232RL**.

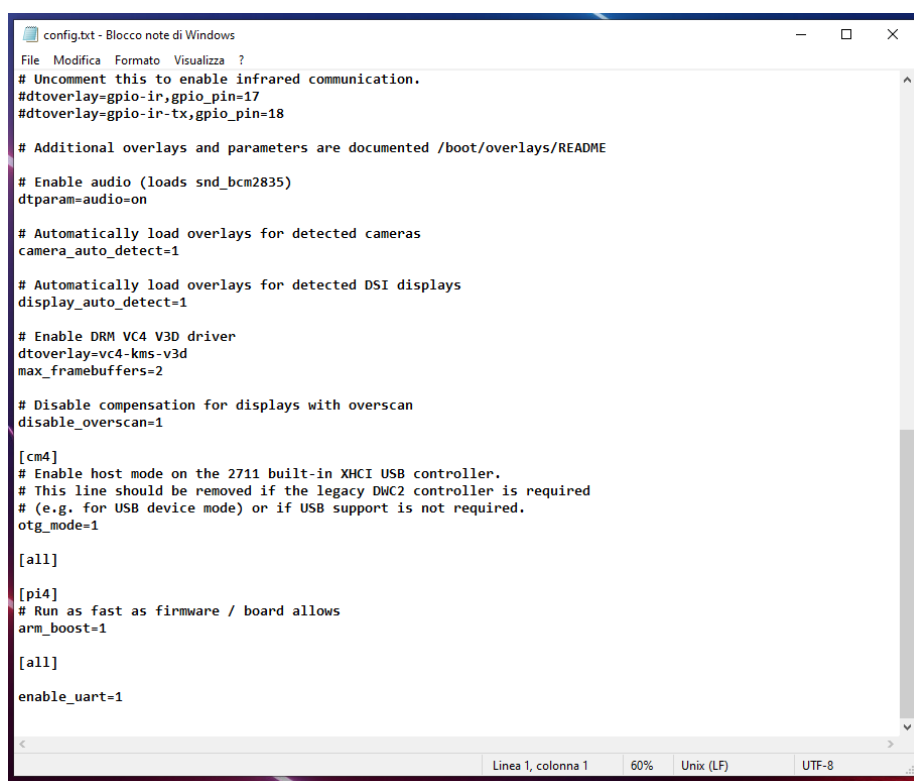
Preparazione del Target

Il **Raspberry Pi 3A+** utilizza come memoria di archiviazione di massa una MicroSD che necessita di essere correttamente formattata.

Per la formattazione della MicroSD si utilizza il software gratuito **Raspberry Pi Imager** (<https://www.raspberrypi.com/software/>) che di Default andrà ad installare il Raspberry Pi OS che è il Sistema Operativo proprietario della piattaforma. Questo ci permetterà al primo avvio, di partizionare la SD in modo corretto e nello stesso tempo verranno caricati tutti i files necessari al **bootstrapping** del nostro target. Scendendo nel dettaglio tra tutti i files presenti, solo alcuni sono utilizzati al nostro scopo mentre altri andranno eliminati o sostituiti.

Una volta terminata la procedura di formattazione accedendo alla directory principale della MicroSD di dovrà procedere come segue:

- 1) **Eliminare tutti i files denominati come "kernelX.img"** in quanto dovranno essere sostituiti col file "kernel7.img" di pijFORTHos che dovrà essere l'unico kernel presente nella MicroSD.
- 2) Successivamente andiamo a **modificare il file "config.txt"** andando ad aggiungere alla fine del file di testo la stringa **enable_uart=1** salvare la modifica e chiudere il file.



```
config.txt - Blocco note di Windows
File Modifica Formato Visualizza ?
# Uncomment this to enable infrared communication.
#dtoverlay=gpio-ir,gpio_pin=17
#dtoverlay=gpio-ir-tx,gpio_pin=18

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# Automatically load overlays for detected cameras
camera_auto_detect=1

# Automatically load overlays for detected DSI displays
display_auto_detect=1

# Enable DRM VC4 V3D driver
dtoverlay=vc4-kms-v3d
max_framebuffers=2

# Disable compensation for displays with overscan
disable_overscan=1

[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[all]

[pi4]
# Run as fast as firmware / board allows
arm_boost=1

[all]

enable_uart=1
```

- 3) Copiare nella directory principale il file **kernel7.img** di pijFORTHos ottenuto su gentile concessione del professore Daniele Peri.

A questo punto il Target è pronto per essere utilizzato attraverso un qualsiasi software che permette la comunicazione tramite protocollo UART.

Preparazione PC

Successivamente bisogna preparare l'ambiente di sviluppo sul PC che utilizzeremo per programmare il Target col terminale FORTH che in base al sistema operativo installato sul PC si dovrà utilizzare differenti software. Come già detto precedentemente in questa documentazione per le Piattaforme Windows e MacOS si potrà utilizzare **ZOC8** mentre per piattaforme Linux si utilizzerà **picocom-3.1**. In questa documentazione verrà illustrata la procedura su piattaforma Linux.

Procedura con LINUX e picocom-3.1

Per installare picocom-3.1 è necessario aprire una nuova finestra del Terminale e digitare i seguenti comandi in ordine:

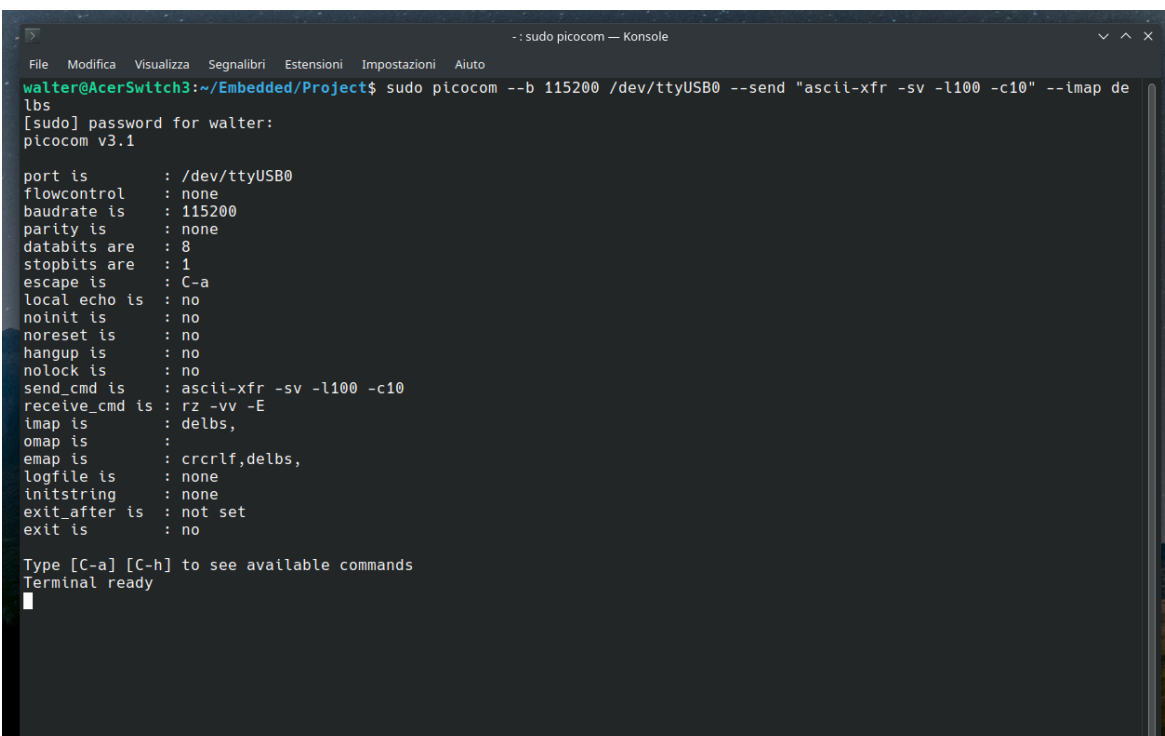
```
sudo apt-get install minicom
```

```
sudo apt-get install picocom
```

L'installazione di **minicom** è necessaria per la corretta esecuzione di picocom in quanto quest'ultimo è basato su minicom. Al fine di stabilire la connessione con il target useremo picocom digitando la stringa che ci permette di avviare la comunicazione, permettere l'invio di file con il supporto ASCII e formattare il terminale in maniera corretta.

Collegare adesso il dispositivo al PC utilizzando la **FT232RL** attraverso un cavo di connessione di tipo MiniUSB. Successivamente digitare il seguente comando nel terminale per avviare picocom:

```
sudo picocom --b 115200 /dev/ttyUSB0 --send "ascii-xfr -sv -l100 -c10" --imap delbs
```



```
--: sudo picocom -- Konsole
File  Modifica  Visualizza  Segnalibri  Estensioni  Impostazioni  Aiuto
walter@AcerSwitch3:~/Embedded/Project$ sudo picocom --b 115200 /dev/ttyUSB0 --send "ascii-xfr -sv -l100 -c10" --imap de
lbs
[sudo] password for walter:
picocom v3.1

port is          : /dev/ttyUSB0
flowcontrol     : none
baudrate is     : 115200
parity is       : none
databits are    : 8
stopbits are    : 1
escape is       : C-a
local echo is   : no
noinit is       : no
noreset is      : no
hangup is       : no
nolock is       : no
send_cmd is     : ascii-xfr -sv -l100 -c10
receive_cmd is  : rz -vv -E
imap is         : delbs,
omap is         :
emap is         : crclrf,delbs,
logfile is      : none
initstring      : none
exit_after is   : not set
exit is         : no

Type [C-a] [C-h] to see available commands
Terminal ready
```


Visualizzata la schermata che descrive tutti i parametri della comunicazione da adesso è possibile utilizzare il terminale FORTH ed interagire col Target.

Adesso è necessario caricare nel Target il codice sorgente del progetto utilizzando una combinazione di tasti appositi di picocom in sequenza:

[CTRL] + A

[CTRL] + S

```
noexec is      : no
nolock is      : no
send_cmd is    : ascii-xfr -sv -l100 -c10
receive_cmd is : rz -vv -E
imap is       : delbs,
omap is       :
emap is       : crcrlf,delbs,
logfile is    : none
initstring    : none
exit_after is : not set
exit is       : no

Type [C-a] [C-h] to see available commands
Terminal ready

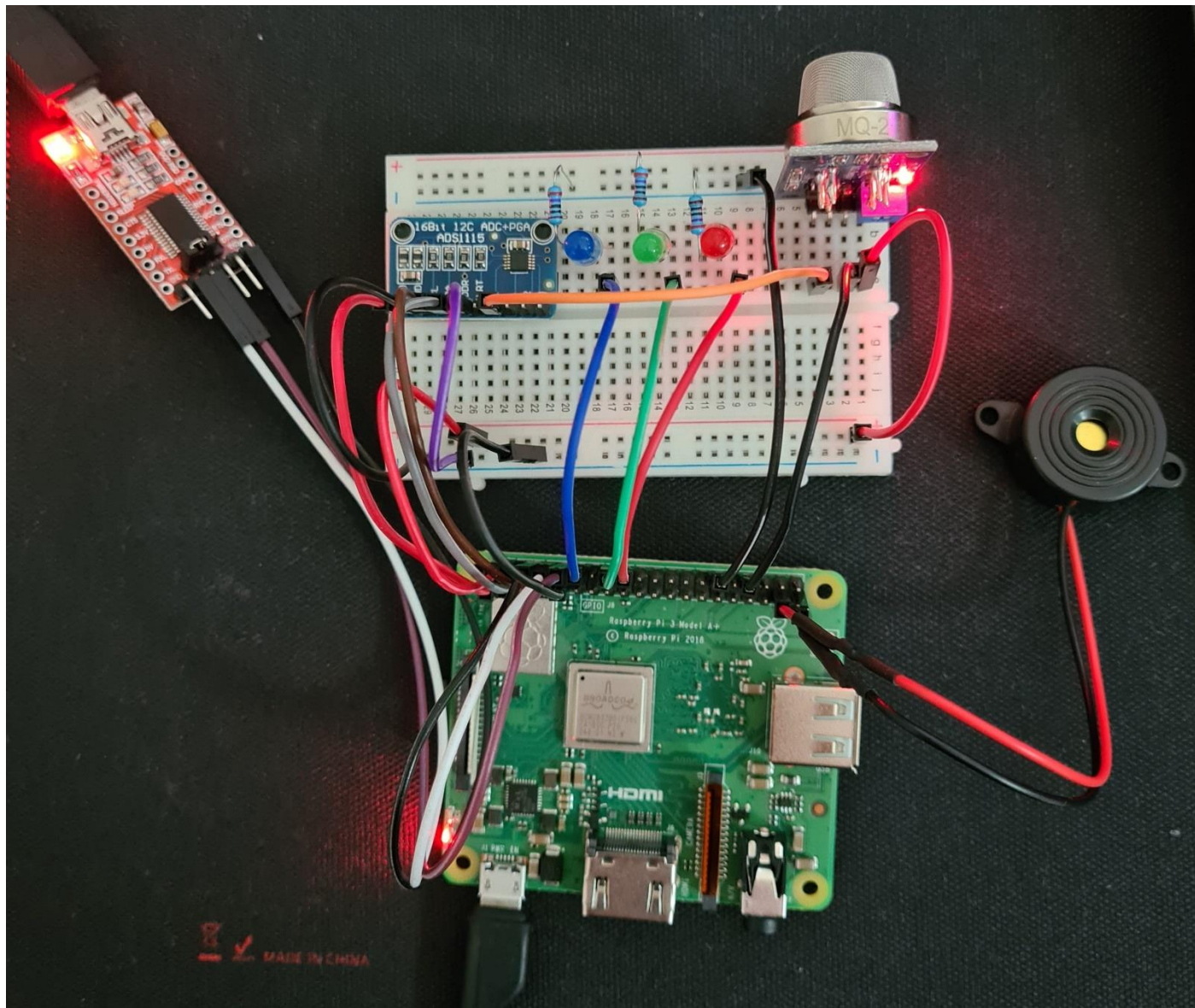
*** file: main.f
```

Con questa combinazione picocom vi chiederà di scrivere il nome del file da inviare al Target che dovrà comprendere anche la sua estensione. Da questo momento è sufficiente attendere fino al completamento dell'operazione.

Una volta che il programma sarà caricato sarà possibile avviarlo eseguendo il comando **START** ovvero la Word definita in FORTH di alto livello che si occuperà di eseguire tutte le operazioni necessarie per avviare il sistema.

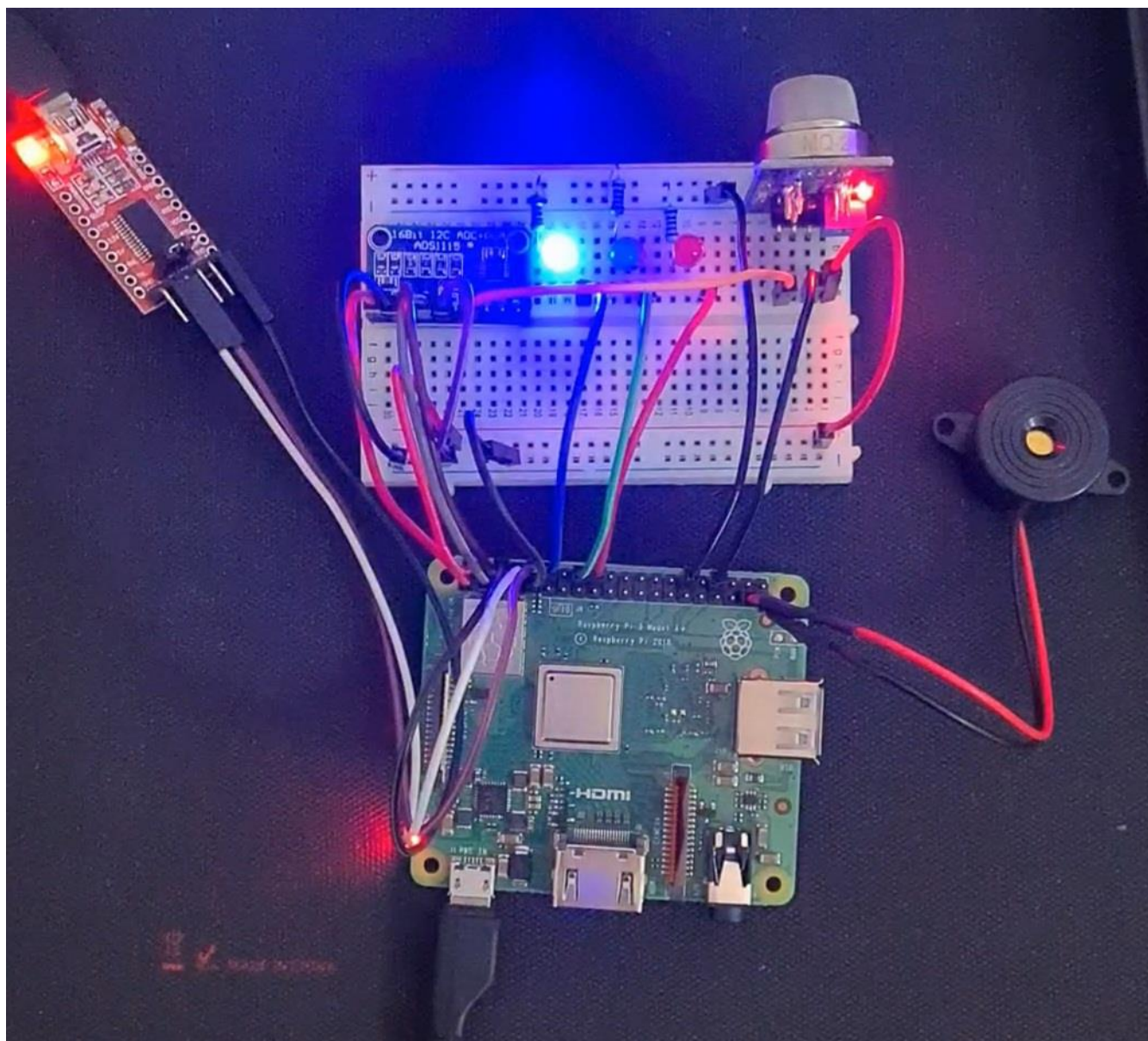
Foto Testing

Panoramica del Sistema



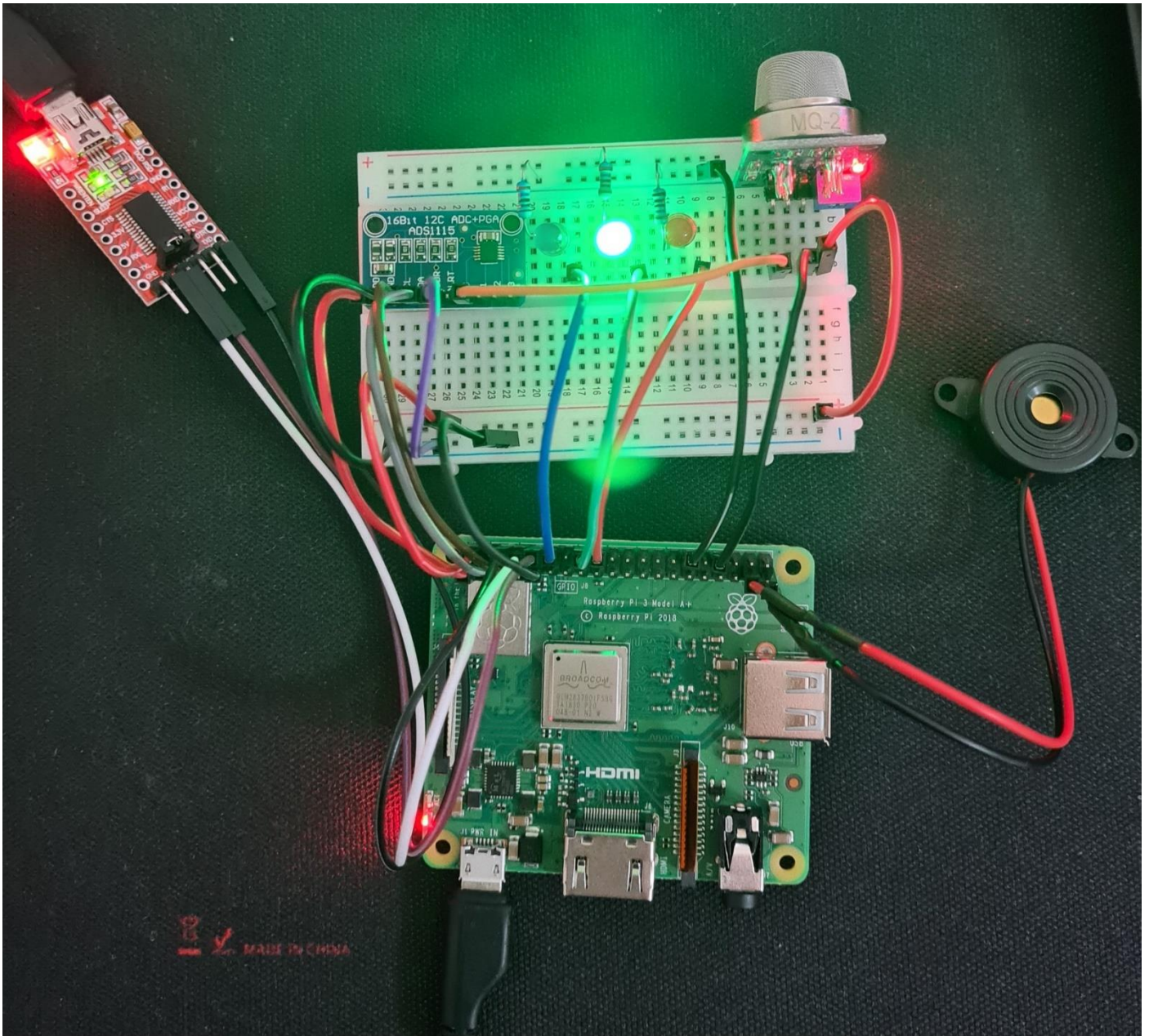
Fase di Calibrazione Ambientale

Durante la fase di Calibrazione il LED Blu lampeggia.



Fase di Scansione: Assenza di Gas

Durante la fase di Scansione in assenza di Gas il LED Verde rimane acceso.



Fase di Scansione: Presenza di Gas

Durante la fase di Scansione in presenza di Gas il LED Rosso e il Buzzer lampeggiano.

