



UNIVERSITÀ
DEGLI STUDI
DI PALERMO

GIOCO DI RIFLESSI

Studente

Antonio Pio Sciacchitano
0759278

Professore:

Daniele Peri

SOMMARIO

1. DESCRIZIONE DEL PROGETTO	2
2. COMPONENTI DEL PROGETTO	2
2.1 Hardware.....	2
2.2 Software.....	2
3. DETTAGLIO DEI COMPONENTI	2
3.1 Raspberry Pi 3 Model A+	2
3.2 GPIO e registri.....	3
3.4 Interfaccia seriale UART FT232RL	4
3.3 Buzzer, LED e bottoni	4
4. SCHEMA DEL PROGETTO	5
5. PROCEDURA DI INSTALLAZIONE	7
5.1 Preparazione del Raspberry Pi.....	7
5.2 Configurazione del Dispositivo Target.....	7
6. FUNZIONAMENTO DEL SISTEMA	7
6.1 Inizializzazione GPIO	7
6.2 Gestione LED e buzzer	8
6.3 Sistema di debouncing.....	8
6.4 Sequenza di gioco	8
7. LOGICA DI CONTROLLO	9
8. POSSIBILI SVILUPPI FUTURI	9
9. FOTO	10

1. DESCRIZIONE DEL PROGETTO

Il progetto rappresenta un gioco interattivo sviluppato su una piattaforma **Raspberry Pi 3A+**, programmata in **Forth**. Il sistema utilizza i GPIO per controllare una serie di LED, pulsanti e un buzzer, offrendo un'esperienza ludica che combina segnali visivi e acustici. Gli obiettivi principali includono:

- Attivazione casuale dei LED.
- Risposta dell'utente tramite pulsanti.
- Segnali acustici e visivi per indicare stati del gioco.

L'obiettivo didattico è mostrare l'uso avanzato dei GPIO in un contesto pratico e coinvolgente.

2. COMPONENTI DEL PROGETTO

2.1 Hardware

- **Raspberry Pi 3 Model A+** (512 MB di RAM)
- **LED**
- **LED RGB** (rosso, verde, blu)
- **BUZZER**
- **PULSANTI DI INPUT** (rosso, verde, blu)
- Cavi jumper per i collegamenti
- resistenze da 220 ohm

2.2 Software

- Sistema operativo **pijFORTHos**
- Software di terminale:

ZOC (Windows)

Per il target è stato utilizzato il sistema operativo **pijFORTHos**, che consente una gestione diretta dell'hardware tramite il linguaggio **Forth**, ottimizzato per l'interazione a basso livello. La comunicazione seriale tra il dispositivo e il PC è stata realizzata utilizzando **ZOC8 Terminal** su Windows o software equivalenti su altre piattaforme per il trasferimento e l'esecuzione del codice.

3. DETTAGLIO DEI COMPONENTI

3.1 Raspberry Pi 3 Model A+

Il **Raspberry Pi 3A+** è un single board computer progettato per essere accessibile, versatile e ideale per una vasta gamma di progetti di elettronica e programmazione.

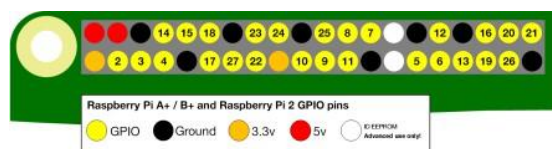
Con un processore quad-core ARM Cortex-A53 da 1.4 GHz e 512 MB di RAM, il Raspberry Pi 3A+ è perfetto per applicazioni leggere come l'automazione domestica, piccoli server o dispositivi IoT.

La presenza di un'interfaccia GPIO con 40 pin consente il collegamento di sensori, attuatori e altri componenti esterni, rendendolo uno strumento potente per sviluppare progetti innovativi in modo semplice ed efficace.



3.2 GPIO e registri

I GPIO del **Raspberry Pi 3A+** offrono una grande versatilità e sono configurati per soddisfare diverse esigenze. Possiamo utilizzarli sia come **OUTPUT**, ad esempio per controllare **LED o buzzer**, sia come **INPUT**, collegandoli a pulsanti per ricevere segnali esterni.



Per gestire i **GPIO**, utilizziamo registri specifici:

- **GPSET0** e **GPCLR0** ci permettono di attivare o disattivare i pin configurati come output.
- **GPLEV0**, invece, è dedicato alla lettura dello stato dei pin configurati come input.

La scheda dispone anche di vari tipi di pin, fondamentali per i nostri progetti:

I pin di alimentazione, come quelli a 3.3V, 5V e GND, sono indispensabili per fornire energia ai componenti.

I pin GPIO generici possono essere configurati sia come input che come output, a seconda delle necessità.

Infine, i pin GPIO specializzati ci offrono funzionalità avanzate come I2C, SPI e UART, rendendo il Raspberry Pi 3A+ ideale per la comunicazione con dispositivi esterni e la gestione di progetti complessi.

In questo modo, possiamo sfruttare al massimo le potenzialità della scheda, adattandola a una vasta gamma di applicazioni

3.4 Interfaccia seriale UART FT232RL

L'adattatore **USB to Serial con FT232RL** consente di collegare un PC a qualsiasi sistema a microcontrollore tramite la porta USB. Oltre ai segnali **TX** e **RX**, include anche le linee di handshaking come **CTS**, **RTS** e altre, garantendo un supporto completo per la comunicazione seriale.

Una volta collegata alla porta USB del PC, la scheda viene riconosciuta automaticamente come una porta COM, permettendo di stabilire la connessione con il dispositivo target senza necessità di modifiche. Per il collegamento, è sufficiente connettere il **GND** della scheda con il GND del target e incrociare le linee di segnale, collegando il **TX** della scheda con l'**RX** del target e viceversa.

Questo adattatore semplifica la comunicazione tra PC e microcontrollori, rendendolo uno strumento indispensabile per lo sviluppo e il debug di progetti embedded.



3.3 Buzzer, LED e bottoni

- **LED**

Sono presenti tre led gialli ai quali sono collegati tre resistenze da 220 ohm



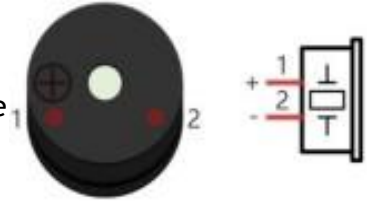
- **LED RGB:**

è un dispositivo che emette luce attraverso tre diodi a colori separati: rosso, verde e blu. Combinando diverse intensità di questi tre colori, è possibile ottenere una vasta gamma di colori. Ogni colore del LED è controllato separatamente, permettendo di ottenere qualsiasi combinazione. Come il buzzer, il LED RGB funziona semplicemente alimentandolo con una tensione continua (DC) e può essere facilmente controllato tramite un pin digitale o un circuito di commutazione. Funziona con tensioni che generalmente variano da 3V a 5V



BUZZER:

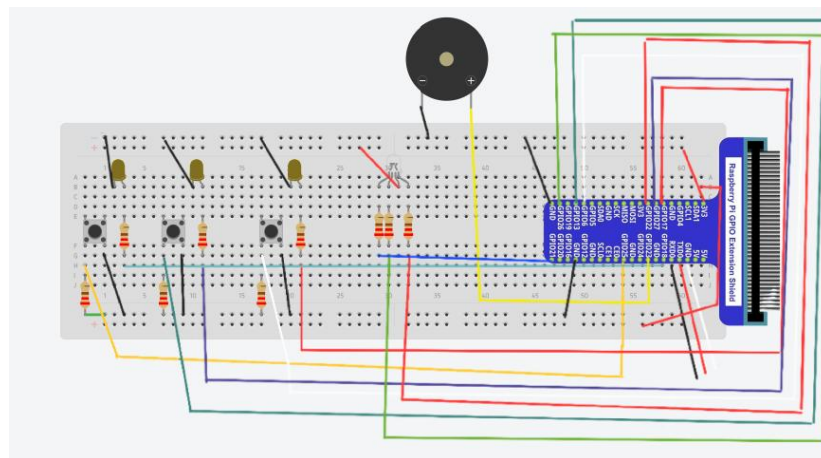
è un dispositivo elettromeccanico che emette un suono quando viene alimentato. Al suo interno è presente un circuito oscillatore che genera la frequenza del suono. Per far suonare un buzzer attivo, basta fornire una tensione continua (DC) al dispositivo. Non è necessario inviare segnali di controllo esterni o impulsi specifici, quindi può essere facilmente collegato direttamente a una fonte di alimentazione o controllato tramite un interruttore o un pin digitale.



- **PULSANTI:** L'utente deve premere il bottone corrispondente al colore del LED acceso. Quando il bottone giusto viene premuto, il LED si spegne e il **LED RGB** si accende con il colore del bottone premuto (Rosso, Verde o Blu) per indicare il vincitore.

4. SCHEMA DEL PROGETTO

Lo schema circuitale risulta come segue:



Funzione	GPIO	Pin	Pin	GPIO	Funzione
-	3V3 Power	1	2	5V Power	-
-	GPIO 2 (SDA)	3	4	5V Power	-
-	GPIO 3 (SCL)	5	6	Ground	-
-	GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)	-
-	Ground	9	10	GPIO 15 (RXD)	-
LED1	GPIO 17	11	12	GPIO 18 (PCM_CLK)	-
LED3	GPIO 27	13	14	Ground	-
-	GPIO 22	15	16	GPIO 23	BUZZER
-	3V3 Power	17	18	GPIO 18	LED2
-	GPIO10(MOSI)	19	20	Ground	-
RGB Blu	GPIO9(MISO)	21	22	GPIO 25	RGB Rosso
-	GPIO11(SCLK)	23	24	GPIO8(CE0)	-
-	Ground	25	26	GPIO7(CE1)	BOTTONE BLU
-	GPIO0(ID_SD)	27	28	GPIO1(ID_SC)	-
-	GPIO5	29	30	Ground	-
BOTTONE ROSSO	GPIO6	31	32	GPIO12(PWM0)	-
BOTTONE VERDE	GPIO13(PWM1)	33	34	Ground	-
-	GPIO19(PCM_FS)	35	36	GPIO16	-
-	GPIO 26	37	38	GPIO 20 (PCM_DIN)	RGB Verde
-	Ground	39	40	GPIO 21 (PCM_DOUT)	-

5. PROCEDURA DI INSTALLAZIONE

5.1 Preparazione del Raspberry Pi

Prima di utilizzare il sistema è necessario preparare l'ambiente di sviluppo sia per il dispositivo Target sia per il PC utilizzato per la programmazione interattiva tramite il protocollo di comunicazione UART.

5.2 Configurazione del Dispositivo Target

Per questo progetto è stato utilizzato un Raspberry Pi 3 A+. La configurazione richiede di formattare la microSD che conterrà il sistema operativo pijFORTH.

Formattazione della microSD

- Utilizzare il software **Raspberry Pi Imager** e installare **Raspberry Pi OS Lite**.

Preparazione del sistema operativo

- Al termine della formattazione, accedere alla directory della microSD e completare i seguenti passaggi:
- Eliminare tutti i file denominati "kernelX.img".
- Copiare il file kernel7.img (il file del sistema operativo pijFORTH opportunamente preparato) nella microSD.
- Rinominare il file **kernel7.img** in **kernel.img**.

Abilitazione della comunicazione UART

- Modificare il file config.txt presente nella directory principale della microSD aggiungendo la seguente riga alla fine del file:
enable_uart=1
- Salvare le modifiche.

A questo punto, il dispositivo Target è pronto per comunicare tramite il protocollo UAR

6. FUNZIONAMENTO DEL SISTEMA

6.1 Inizializzazione GPIO

Il codice Forth configura i GPIO per le varie periferiche:

- **GPIO17_OUT**: LED1 (Pin 17)
- **GPIO18_OUT**: LED2 (Pin 18)
- **GPIO27_OUT**: LED3 (Pin 27)
- **GPIO23_OUT**: Buzzer (Pin 23)
- **GPIO22_OUT**: RGB Rosso (Pin 22)
- **GPIO26_OUT**: RGB Verde (Pin 26)
- **GPIO21_OUT**: RGB Blu (Pin 21)
- **GPIO6_IN**: Bottone Rosso (Pin 6)
- **GPIO13_IN**: Bottone Verde (Pin 13)
- **GPIO25_IN**: Bottone Blu (Pin 25)

6.2 Gestione LED e buzzer

I LED vengono controllati attraverso registri GPSET0 e GPCLR0 usando maschere predefinite:

- LED1_MASK (GPIO 17)
- LED2_MASK (GPIO 18)
- LED3_MASK (GPIO 27)

Per il LED RGB

- SEG_R_MASK
- SEG_G_MASK
- SEG_B_MASK

Il buzzer utilizza BUZZER_MASK (GPIO 23) per i segnali sonori.

6.3 Sistema di debouncing

Il sistema implementa un debouncing hardware-software con:

- **DEBOUNCE-DELAY** costante impostata a 50000
- Variabili **ULTIMO-ROSSO**, **ULTIMO-BLU**, **ULTIMO-VERDE** per tracciare l'ultima attivazione valida
- **MILLIS-COUNTER** per la temporizzazione ogni pressione del bottone, viene validata solo se è trascorso il tempo di debouncing dall'ultima pressione valida."

6.4 Sequenza di gioco

- **1. Inizializzazione del sistema:**
 - Esecuzione di INIT-STATE che spegne tutti i LED
 - Inizializzazione delle variabili di stato e del flag PARTITA-ATTIVA
- **Sequenza di avvio BLINK:**
 - LED1 si accende con feedback sonoro
 - LED3 si accende con feedback sonoro
 - LED2 si accende con feedback sonoroLa sequenza si ripete due volte
- **Fase di gioco con tre stati:**
 - Stato 0: Accensione casuale di uno dei tre LED principali
 - Stato 1: Attesa dell'input utente
 - Stato 2: Gestione della risposta e feedback

- **Gestione della risposta:**
 - Controllo del bottone premuto
 - Accensione del segmento RGB corrispondente
 - Esecuzione del suono di fine partita
 - Preparazione per una nuova partita"

7. LOGICA DI CONTROLLO

Il sistema utilizza tre variabili principali di controllo:

STATO-GIOCO: Controlla il flusso del gioco (0-2)

LED-ATTIVO: Tiene traccia del LED attualmente acceso (-1 per nessuno, 0-2 per i LED)

RGB-COLORE: Memorizza il colore RGB selezionato dall'utente

PARTITA-ATTIVA: Flag booleano per il controllo del ciclo di gioco

8. POSSIBILI SVILUPPI FUTURI

- Espansione del sistema con più LED o colori.
- Implementazione di un display per mostrare informazioni di gioco come punteggi.
- Integrazione con sensori aggiuntivi per rendere l'interazione più ricca

9. FOTO

