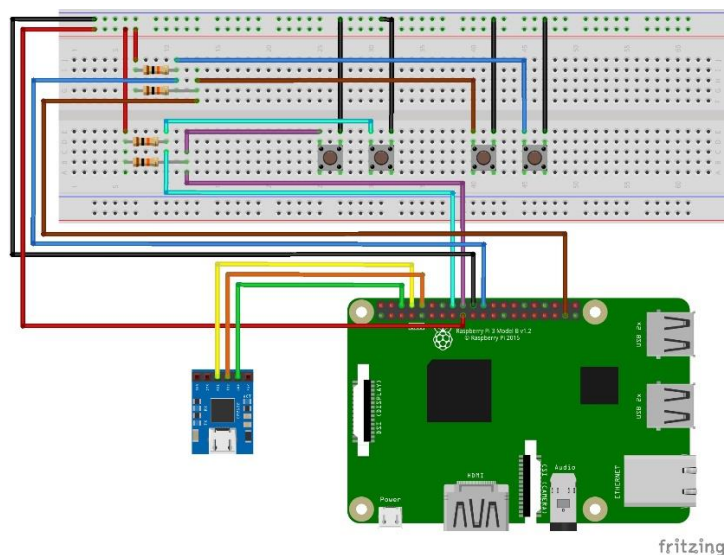


Introduzione

In questo progetto viene utilizzato un Raspberry Pi 3B per realizzare il famoso gioco del Pong. L'interazione con il gioco è stata gestita attraverso l'interfaccia GPIO, mentre la visualizzazione è stata fornita attraverso l'uscita HDMI. Il progetto dimostra come il Raspberry Pi possa essere utilizzato come piattaforma per la realizzazione di giochi, combinando la flessibilità e la potenza del linguaggio di programmazione Forth con l'interazione con il mondo reale reso possibile dalle interfacce GPIO. L'implementazione del gioco del Pong su questo dispositivo dimostra come sia possibile creare esperienze di gioco coinvolgenti e interattive utilizzando il Raspberry Pi come base, e come le sue funzionalità hardware possano essere utilizzate in modo creativo e innovativo. In particolare, l'utilizzo di Forth come linguaggio di programmazione per questo progetto mette in luce le potenzialità del Raspberry Pi come piattaforma di sviluppo versatile e adatta a molteplici scopi.

Il gioco termina quando uno dei due giocatori arriva al punteggio pari a 9.

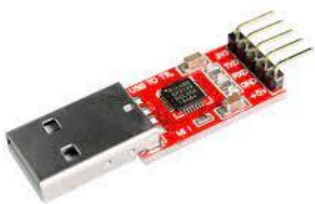
Hardware



[Figura 1]

Il sistema hardware utilizzato per questo progetto è costituito principalmente da un Raspberry Pi 3B, interfaccia UART USB per l'interfacciamento con il target e dei pulsanti di input. In figura 1 è mostrata un'immagine dei componenti hardware utilizzati.

Modulo cp2102 da USB a porta seriale TTL

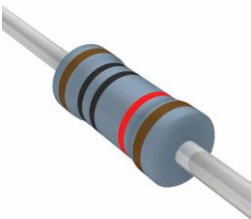


Raspberry Pi modello 3B



Il Raspberry Pi 3B è una scheda computer monocircuito dotata di diverse interfacce, tra cui Ethernet, Wi-Fi, USB e HDMI, che lo rendono ideale per una vasta gamma di applicazioni. Nel nostro caso, è stato utilizzato per eseguire il codice di controllo del gioco del Pong e per comunicare con il target attraverso l'interfaccia UART USB.

Resistori



Resistenza da 10Kohm

I/O choices



I pulsanti di input sono stati collegati al Raspberry Pi 3B attraverso delle resistenze di pull-up. Questo significa che quando il pulsante non viene premuto, la GPIO a cui è collegato il pulsante leggerà il valore di tensione Vcc, ovvero una logica 1. Quando il pulsante viene premuto, la GPIO leggerà il valore di tensione Gnd, ovvero una logica 0. In questo modo, il Raspberry Pi può leggere lo stato del pulsante e agire di conseguenza.

L'utilizzo di resistenze di pull-up per interfacciare i pulsanti con il Raspberry Pi è una scelta comune in molte applicazioni, poiché offre un modo semplice ed efficiente per leggere lo stato degli input. Le resistenze di pull-up sono di solito di valore elevato, in modo da garantire che la tensione in ingresso sia sempre stabile e

non dipenda dalla resistenza del circuito esterno. In questo modo, si garantisce una lettura precisa dello stato del pulsante.

In sintesi, il sistema hardware utilizzato in questo progetto è stato progettato per consentire una facile interfacciamento con il Raspberry Pi e garantire una lettura precisa degli input da parte del sistema. Il collegamento dei pulsanti con resistenze di pull-up ha permesso una lettura affidabile dello stato degli input e ha semplificato la progettazione del circuito

Assegnamenti GPIO

GPIO#	FUNCTION	USAGE
14	TX	UART transmitter
15	RX	UART receiver
23	INPUT	Button_up Player_1
24	INPUT	Button_down Player_1
25	INPUT	Button_up Player_2
26	INPUT	Button_down Player_2

Ambiente

L'Ambiente utilizzato in questo progetto è PijForthOs. PijForthOS è un ambiente di sviluppo Forth progettato appositamente per il Raspberry Pi. Forth è un linguaggio di programmazione compatto e veloce che si presta bene alla programmazione embedded, ovvero alla programmazione di sistemi a basso livello, come i microcontrollori e i dispositivi embedded. Utilizzando un ambiente di sviluppo come PijForthOS, si possono sfruttare appieno le potenzialità del Raspberry Pi, programmando il sistema a basso livello e interagendo direttamente con l'hardware.

ZOC8

Il software ZOC8 è un emulatore di terminale seriale che permette di comunicare con l'interfaccia UART del Raspberry Pi. ZOC8 consente di stabilire una connessione seriale tra il computer e il Raspberry Pi, utilizzando un cavo USB-to-serial o un adattatore seriale. In questo modo, è possibile interagire con il Raspberry Pi come se si stesse utilizzando un terminale seriale, inviando comandi e ricevendo risposte dal sistema. L'utilizzo di ZOC8 semplifica la programmazione del Raspberry Pi, in quanto consente di interagire con il sistema in modo diretto e immediato. Ad esempio, è possibile accedere alla console del Raspberry Pi e interagire con il sistema operativo, eseguire comandi, visualizzare output e risolvere eventuali problemi di configurazione.

CODICE

Il codice che implementa il gioco del Pong su Raspberry Pi usando il linguaggio Forth è suddiviso in diverse sezioni, ognuna delle quali si occupa di una specifica funzionalità del gioco.

La prima sezione si occupa della realizzazione della parte grafica di un rettangolo. Questa primitiva servirà per rappresentare il campo da gioco, la palla e le barre. In questa sezione vengono definiti i parametri del rettangolo, e le istruzioni che serviranno per definire le dimensioni e i colori, e successivamente mediante

un'ulteriore istruzione (`DRAW_RECT`) viene disegnato il rettangolo stesso sulla schermata. Il rettangolo viene disegnato utilizzando una subroutine scritta in Assembly per Arm32. È stata presa questa scelta in modo da migliorare il rendering dei vari componenti grafici.

La seconda sezione si occupa della realizzazione del campo da gioco tramite l'istruzione `DRAW_RECT` precedentemente citata. In questa sezione viene definita la posizione iniziale del campo.

La terza sezione si occupa delle definizioni per il funzionamento con l'interfaccia GPIO del Raspberry Pi, ovvero delle porte di input/output utilizzate per la gestione del gioco. In questa sezione vengono definite le porte GPIO utilizzate per i pulsanti di controllo dei giocatori, e viene implementata la logica di lettura delle porte per la gestione del movimento delle barre.

La quarta sezione si occupa della realizzazione della barra, ovvero del disegno delle due barre di gioco che rappresentano i giocatori. In questa sezione vengono definiti i parametri delle barre, come le dimensioni e i colori, viene disegnata la barra stessa sulla schermata e viene gestito il movimento della barra.

La quinta sezione si occupa della realizzazione del punteggio, ovvero della gestione del conteggio dei punti dei due giocatori. In questa sezione viene implementata la logica di conteggio dei punti e di visualizzazione del punteggio sulla schermata. Anche in questo caso è stata realizzata una Subroutine in Assembly in modo da migliorare il rendering del punteggio. Tramite il codice scritto in questa sezione è anche possibile caricare le bitmap per dei caratteri aggiuntivi in modo da generarli e stamparli a schermo.

La sesta sezione si occupa della realizzazione della palla e delle istruzioni per il suo movimento, ovvero della logica che fa muovere la palla tra i giocatori. In questa sezione vengono definite le regole di movimento della palla, come la velocità e l'angolo di lancio, viene implementata la logica di rimbalzo della palla sulle barre e sui bordi del rettangolo e viene segnato il goal nel caso in cui la palla impatta su una delle due pareti alle spalle delle barre.

Infine, la settima sezione si occupa della definizione del timer del gioco, quindi vengono gestiti i tempi per implementare la velocità delle barre e della palla.

OTTIMIZZAZIONE

Durante lo sviluppo del gioco, si era rilevato che il rendering delle immagini su schermo risultava piuttosto lento. È stato quindi deciso di cercare una soluzione per ottimizzare quella parte del codice, al fine di renderlo più efficiente e veloce.

La soluzione adottata è stata quella di realizzare delle subroutine in assembly per arm32, il set di istruzioni utilizzato dal Raspberry Pi 3B. Le subroutine sono state scritte in assembly, utilizzando l'istruzione di store multiplo (STM) per salvare più registri in una sola operazione, ottenendo delle subroutine molto efficienti dal punto di vista delle prestazioni. In particolare vengono salvati 8 registri con una sola operazione.

Successivamente, gli opcode delle subroutine sono stati memorizzati in un array tramite il linguaggio Forth. Ciò ha permesso di richiamare le subroutine in modo più efficiente, senza dover ogni volta effettuare il caricamento degli opcode dalla memoria.

Per fare ciò, si era utilizzato il comando `"DOES> JSR ;"`, che ha permesso di cambiare l'esecuzione dell'array, in modo che quando questo viene chiamato, ciò che fa è far eseguire direttamente la subroutine al processore.

Questa ottimizzazione ha avuto un impatto notevole sulle prestazioni del gioco, permettendo di ottenere un rendering delle immagini su schermo molto più veloce e fluido.

In generale, l'utilizzo di subroutine in assembly può rappresentare una soluzione efficace per ottimizzare le prestazioni del proprio codice su piattaforme embedded come il Raspberry Pi 3B. Tuttavia, è importante prestare attenzione alla corretta implementazione delle subroutine e alla loro compatibilità con il set di istruzioni della piattaforma in uso.

Setup del gioco

In primo luogo, è necessario effettuare gli appositi collegamenti dell'hardware, in questo caso il Raspberry Pi 3B, l'interfaccia UART USB e i pulsanti di input.

Prima di procedere con la configurazione del software, è necessario installare l'interprete PijForthOS all'interno della microSD del Raspberry Pi.

In seguito, è necessario settare il corretto indirizzo di base dell'interfaccia GPIO, il quale dipende dal modello del Raspberry Pi utilizzato. Nel nostro caso, abbiamo utilizzato il Raspberry Pi 3B, il cui indirizzo di base dell'interfaccia è 0x3F000000.

Una volta impostato l'indirizzo di base dell'interfaccia, abbiamo utilizzato il software ZOC8 per comunicare con il dispositivo. ZOC è un emulatore di terminale professionale che consente di accedere al Raspberry Pi tramite l'interfaccia UART USB.

Successivamente, deve essere caricato il codice `ans.f` e `pong.f`. Questi file contengono le funzioni necessarie per il funzionamento del gioco, inclusi i controlli GPIO per i pulsanti di input e le funzioni per il movimento della palla e delle barre.

Infine, dopo aver caricato il codice, è sufficiente digitare l'istruzione "START" nel terminale per iniziare a giocare.

Configurazione Terminale Zoc8

Per configurare la connessione tra il Raspberry Pi 3B e l'interfaccia UART CP2102 utilizzando il software ZOC8, è necessario seguire i seguenti passaggi:

1. Collegare l'interfaccia UART al Raspberry Pi 3B attraverso uno dei suoi connettori USB.
2. Accendere il Raspberry Pi 3B e avviare il software ZOC8 sul computer esterno.
3. Selezionare la porta seriale corretta all'interno delle impostazioni di connessione di ZOC8. Nel nostro caso, dovremo selezionare la porta seriale associata all'interfaccia UART CP2102.
4. Impostare la velocità di trasmissione dei dati sulla stessa velocità utilizzata dal Raspberry Pi 3B. Di solito, la velocità di default è di 115200 bps.
5. Impostare il segnale RTS su "On" e il segnale DTR su "On" nelle impostazioni della porta seriale. Questo è necessario per garantire che il Raspberry Pi 3B riceva i comandi correttamente.
6. Confermare le impostazioni di connessione e premere il pulsante "Connetti" per stabilire la connessione tra il Raspberry Pi 3B e il computer esterno.

Una volta stabilita la connessione, sarà possibile utilizzare il terminale di ZOC8 per interagire con il Raspberry Pi 3B e caricare il codice Forth all'interno dell'interprete PijForthOS.