Relazione Led Pong

Introduzione

Lo scopo del progetto era quello di realizzare, tramite la piattaforma Arduino, un simulatore del classico gioco "Pong". Tale gioco prende il nome di "LedPong", in quanto implementato attraverso dei led.

Architettura e Progettazione Iniziale

Siamo partiti ragionando che avremmo avuto due routine principali:

- Pulsazione led flash: Questo piccolo loop avrebbe dovuto gestire la pulsazione del led e attendere la pressione del pulsante per far avviare la seconda routine. Al pulsante non abbiamo deciso di assegnarli un interrupt in quanto essendo un operazione non fondamentale al funzionamento responsivo del gioco, abbiamo inserito un semplice controllo dentro al ciclo, sapendo che a volte si rischia di non rilevare l'evento.
- Movimento della pallina: Per la sequenza di rimbalzo della pallina abbiamo deciso che sarebbe stato il loop centrale di gioco ad eseguirlo. Per rispettare tutte le specifiche richieste abbiamo impostato che all'arrivo a qualsiasi delle estremità dei led, si assume che il gioco sia perso, in modo che se il bottone non viene premuto il gioco termini. I pulsanti verranno gestiti attraverso gli interrupt, ogni bottone avrà il suo. In questo modo al loro click verrà avviata una funzione di check, che imposterà le variabili di gioco in modo tale da interrompere il gioco se non fosse il proprio turno o farlo procedere in caso contrario. Questo controllo della funzione check, avviene attraverso una variabile di stato booleana che indica se sia o meno il proprio turno per premere il pulsante e che verrà impostata all'interno del loop centrale.

Logica Implementativa

Per implementare il gioco abbiamo strutturato varie funzioni :

- **waitStart()**: In modo ciclico esegue la pulsazione del led flash, eseguendo una sequenza di istruzioni tratte dallo sketch di fading all'interno di un ciclo while. Tale ciclo termina solamente quando viene premuto il pulsante associato al led, per avviare il loop di gioco.
- **randomDirection()** : Viene avviata appena termina il waitStart() e ha lo scopo di calcolare, attraverso una funzione randomica, la direzione di partenza della pallina. Dopo essere partita dal centro, decide se spostarsi a destra o a sinistra.
- **loop()**: All'interno del loop di funzionamento di Arduino, abbiamo inserito una serie di controlli quali il controllo sulla direzione e sullo stato di gioco. L'esito di tali controlli servono per scegliere quale routine di gioco avviare.
  Le routine sono:
  - runPlayerOne(): effettua lo scorrimento verso sinistra.

  - runPlayerTwo(): effettua lo scorrimento verso destra.

  All'interno di queste due routine sono impostate le seguenti variabili di stato:

  - pressedButtonOne
  - pressedButtonTwo

  che consentono l'interazione con gli interrupt associati alla pressione dei pulsanti T1 e T2.

- **decrementRT()**: Effettua il decremento del react time all'avvenire di ogni scambio. Dovendo effettuare una riduzione di 1/8, quindi effettuo una moltiplicazione per 7/8.
- **readPotent()**: Effettua la lettura del valore del potenziometro, e in base al range in cui si trova il valore letto, va a settare la variabile "Speed" con il valore deciso.
- **blinky()**: Al termine della partita, questa funzione fa lampeggiare il led del giocatore perdente per 2 secondi. Al termine risetta le variabili "game" e "loser" a 0.

Gestione degli interrupt

Gli interrupt all'interno del nostro programma sono associati ai bottoni di gioco T1 e T2. Tale pulsanti sono collegati ai pin digitali 2 e 3 di Arduino con lo scopo di sfruttare le funzionalità di interrupt integrate nel sistema.
I due interrupt quando eseguiti, avviano un funzione di check specifica per il bottone da cui avviato.
Tale funzione controlla lo stato delle variabili prima illustrate pressedButtonOne e pressedButtonTwo, in modo tale da capire se il tasto è stato premuto nel momento giusto e quindi il gioco continua o se il pulsante è stato premuto nel momento sbagliato e quindi il gioco deve terminare.

Enrico Gnagnarella, Anis Lico, Tommaso Ghini