

Integrazione di sistemi embedded

Laboratorio 7: Misuratore di riflessi

Trufini Andrea, Ribaldone Elia, Carta Federica, Marchesin Andrea

Anno accademico 2018-2019



Contents

1	Introduzione	2
2	Funzionalità e modalità d'uso	2
2.1	Sviluppo dell'interfaccia utente	2
2.1.1	GUI	3
2.1.2	Terminal	6
3	Struttura del codice	6
3.1	Software - PC	6
3.1.1	Software - GUI	6
3.1.2	Software - Terminal	7
3.2	Firmware - Nucleo board	8

1 Introduzione

Il progetto, al quale è stato assegnato il nome *The reaction game*, si basa sulla realizzazione di un misuratore di riflessi attraverso una comunicazione mediante interfaccia seriale tra la scheda *Nucleo F401RE* e il *PC*. In particolare il gioco si basa sulla misura del tempo che il giocatore impiega a pigiare il pulsante sulla board, dopo il verificarsi dell'accensione/spegnimento di un led sulla board stessa con un tempo casuale compreso tra i 10 ed i 20 secondi.

2 Funzionalità e modalità d'uso

Il misuratore di riflessi è stato sviluppato in due parti principali:

1. sviluppo dell'interfaccia utente;
2. programmazione del microcontrollore.

2.1 Sviluppo dell'interfaccia utente

L'interfaccia utente, ovvero il software utilizzabile sul PC, è stato sviluppato interamente in **Python**, fatta eccezione solo per lo script **Bash** che viene eseguito inizialmente e permette di scegliere quale tra le due possibili modalità di funzionamento utilizzare, come mostrato in Figura 1.

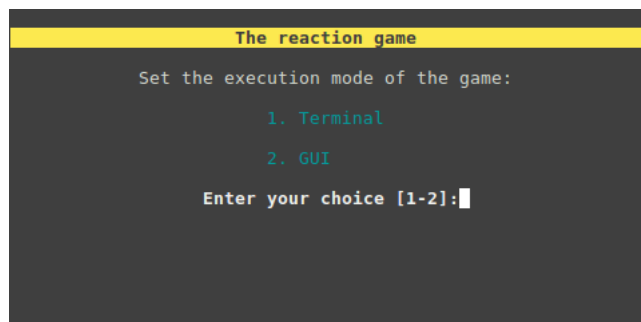


Figure 1

In particolare, come mostrato in figura, si vede che le due possibili modalità sono:

- GUI;
- Terminal.

2.1.1 GUI

All'avvia dell'interfaccia grafica, la schermata iniziale si presenta come quella di Figura 2 in cui è possibile inserire il proprio *username* e fare click sul pulsante *Begin* oppure semplicemente premere *Invio*. All'avvio dell'interfaccia grafica parte anche un sottofondo musicale con una modulazione crescente del volume, gestita nel codice.

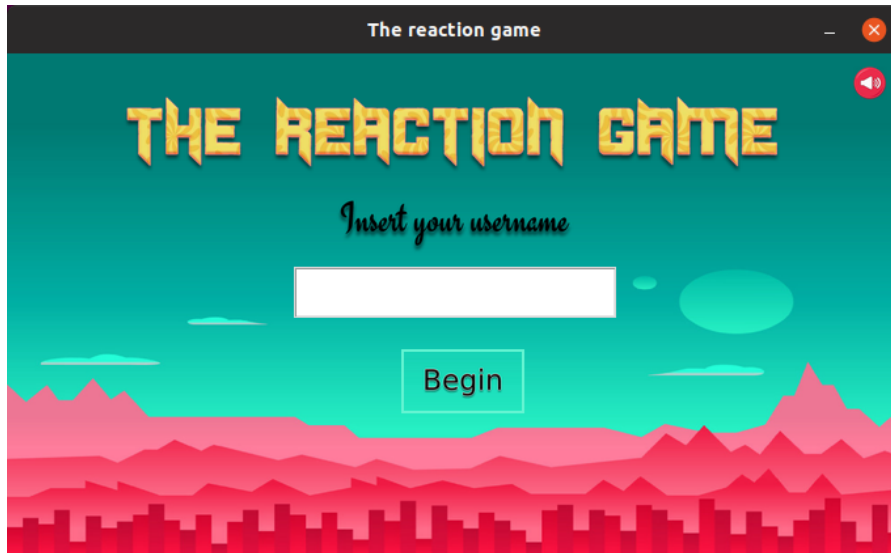


Figure 2

Dopo aver scelto ed inserito il proprio *username* si viene direzionati su una nuova schermata, che si presenta come quella di Figura 3, in cui è possibile procedere con la misura del tempo di reazione. Innanzitutto, è necessario selezionare la porta sulla quale è collegata la board attraverso l'apposito menù a tendina. Una volta selezionata la porta, è possibile procedere cliccando sul pulsante rotondo presente al centro della schermata, il quale emula gli eventi di accensione e spegnimento del LED presente sulla board Nucleo.

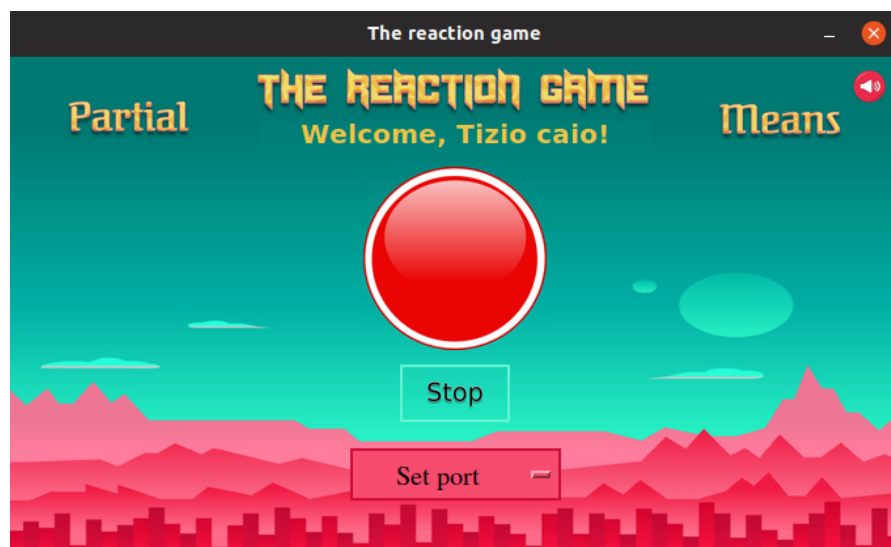


Figure 3

Una volta pigiato il pulsante sullo schermo, questo diventerà ombreggiato e, dopo un ritardo casuale compreso tra 10 e 20 secondi, cambierà colore in modo sincronizzato con il LED. Dal momento

in cui avviene il toggle del LED, il microcontrollore inizia il conteggio del tempo che viene interrotto nel momento in cui l'utente pigia il pulsante presente sulla board e, in questo modo, il tempo viene visualizzato nella colonna *Partial*. Effettuando delle misure ripetute, arrivati ad un numero di 7 tempi, viene calcolata una media che viene riportata nell'apposita colonna *Means*. Il gioco continua fino a quando si raggiunge un numero di 7 medie, oppure si può decidere di interrompere prima la partita pigiando sul pulsante *Stop*. Un esempio di schermata con diversi tempi misurati è riportato in Figura 4.

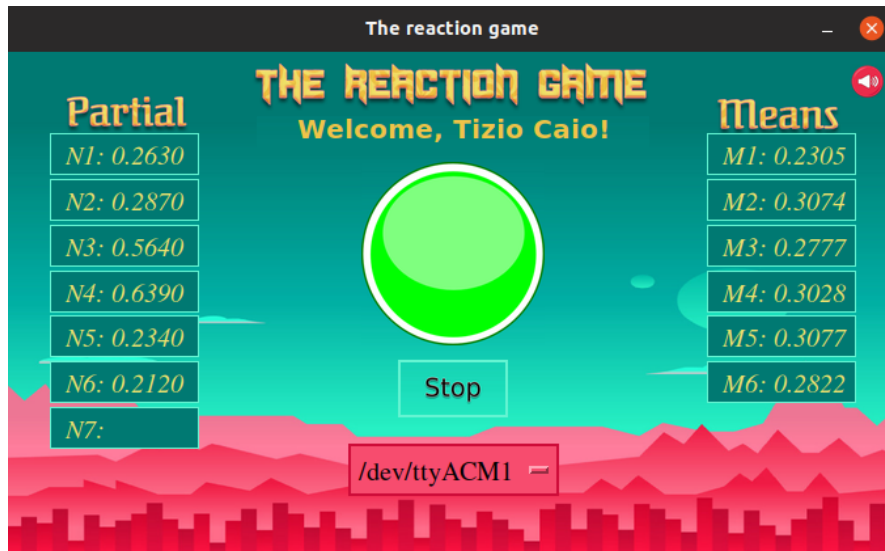


Figure 4

Può però accadere che quando l'interfaccia viene avviata la board non è collegata e, in questo caso non si viene ridirezionati sulla schermata di Figura 3, bensì su una apposita schermata, come riportato in Figura 5. Il software effettua un continuo controllo della presenza di un dispositivo e, nel momento del collegamento della board, tale schermata si chiude automaticamente e nello stesso momento compare una barra di caricamento che attende che la board stessa sia pronta all'uso prima di mostrarla come una porta disponibile nell'apposito menù a tendina.

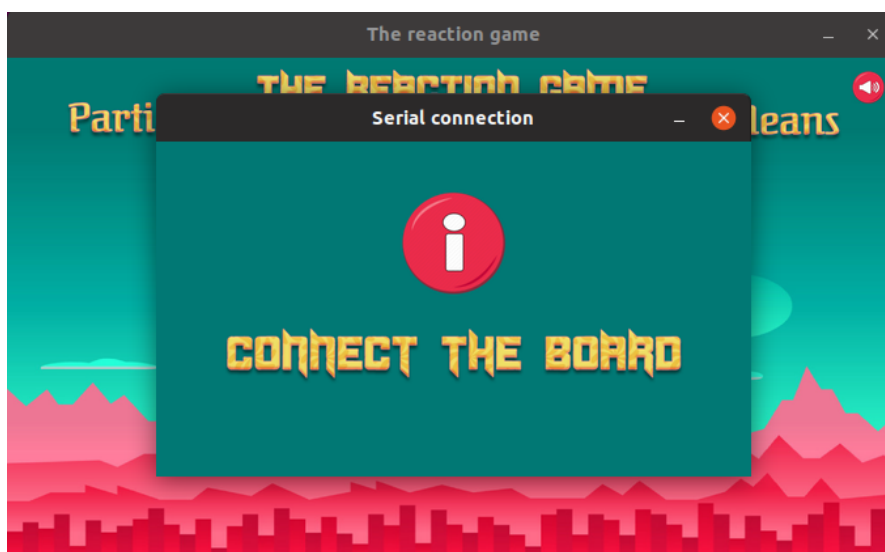


Figure 5

Una volta terminata la partita, si viene ridirezionati su un'ultima schermata, come quella di Figura 6, in cui vengono mostrati il miglior tempo di reazione e la miglior media della partita. Da questa schermata è possibile terminare l'esecuzione pigiando sul pulsante *Exit*, oppure si può ripetere la partita facendo click sull'apposito pulsante. Si può infine notare che in tutte le schermate, in alto a destra, è presente un pulsante raffigurante un altoparlante con il quale è possibile mettere in pausa l'esecuzione del sottofondo musicale.

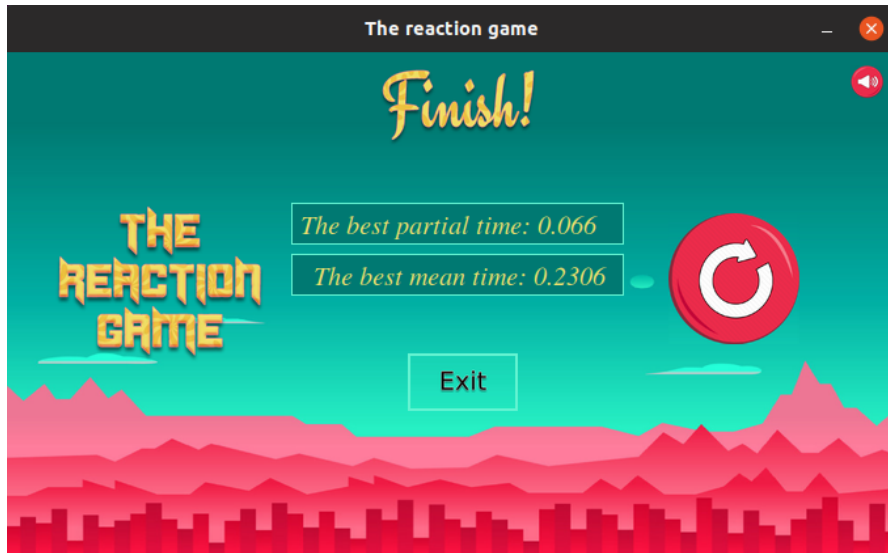


Figure 6

2.1.2 Terminal

Come previsto dalle specifiche di progetto è stata sviluppata anche un'interfaccia da terminale. Il software è stato redatto in linguaggio *Python* e la sua esecuzione ripercorre tutti gli step previsti dalla versione provvista di interfaccia grafica. In termini di esecuzione all'avvio del software vengono mostrate le regole del gioco e successivamente tutte le varie porte seriali connesse al calcolatore.

Come già evidenziato inizialmente, prima dell'esecuzione del codice è fondamentale che l'utente appartenga al gruppo *dialout* in modo tale da avere i permessi per la gestione della porta seriale a cui è stata connessa la scheda Nucleo.

Dopo aver selezionato la porta corretta, è possibile iniziare l'esecuzione del gioco "vero e proprio". Viene richiesto quindi all'utente di fornire il proprio nome, del quale si terrà traccia per tutta la durata della manche. Lo stesso nome verrà poi memorizzato nel caso in cui l'utente riesca ad ottenere il miglior tempo di reazione, a cui verrà associato anche quest'ultimo dato per identificare la statistica del "campione" del gioco. Si ricorda come il dato relativo al miglior punteggio ottenuto venga inizializzato ad ogni avvio del programma, in quanto non ne viene tenuta memoria nelle sue varie esecuzioni.

A questo punto il software inizializza la schermata di gioco e inizia a richiedere in sequenza i vari comandi da fornire al microcontrollore tramite interfaccia seriale. L'algoritmo è stato sviluppato in modo da identificare e scartare tutti i comandi errati forniti dall'utente e presentare di volta in volta indicazioni su come correggere il comando inserito. Nel caso invece si forniscano al software più comandi in successione, l'algoritmo eseguirà solo il primo della lista, ignorando di conseguenza tutti gli altri. Per portare poi il microcontrollore in uno stato noto alla macchina, viene mandato di default il comando di reset, per spegnere dunque il LED.

Dopo l'inserimento di un comando corretto il programma invocherà una funzione utile alla generazione di un ritardo casuale compreso tra 10s e 20s, con la risoluzione di 1ms. Questo tempo verrà poi sfruttato per mettere in delay il software prima di inoltrare il comando sulla linea seriale. Successivamente, il programma rimane in attesa del tempo di reazione restituito dal microcontrollore, del quale ne effettuerà la conversione per renderlo disponibile direttamente all'utente con unità di misura espressa in secondi.

Il gioco procede richiedendo all'utente di accendere o spegnere il led per sette volte in totale, registrando ad ogni turno il suo tempo di reazione. Al termine di tutti i tentativi delle manche del gioco, il programma restituisce all'utente il suo miglior punteggio e il valore del suo tempo di risposta medio. In aggiunta, come già anticipato, il programma determinerà se quanto totalizzato dall'utente risulti essere il miglior punteggio assoluto e, nel caso, aggiornerà i dati relativi al "campione" visualizzandone il messaggio informativo. In conclusione, il programma proporrà all'utente di procedere con una nuova partita richiedendo il nominativo di un nuovo utente oppure di uscire.

È importante sottolineare come per tutta la durata del gioco l'utente possa procedere all'interruzione della partita. Il comando in questione da fornire è "Q", al quale il programma risponderà fornendo sul display il dato relativo al miglior punteggio parziale ottenuto dal giocatore per poi richiedere se l'utente abbia intenzione di ricominciare una nuova partita.

3 Struttura del codice

Nelle seguenti sottosezioni si riporta una piccola analisi del codice scritto per la gestione delle diverse parti del progetto.

3.1 Software - PC

3.1.1 Software - GUI

L'interfaccia grafica è stata realizzata con l'utilizzo del package *Tkinter*. In generale, gli sfondi con le diverse scritte statiche, come anche la grafica dei vari pulsanti, sono stati realizzati mediante l'utilizzo di un apposito software per l'elaborazione grafica. Tutte le coordinate e dimensioni utilizzate sono relative e permettono quindi la corretta esecuzione su schermi con diversa dimensione e/o diversa risoluzione. Inoltre, per la gestione contemporanea di più processi è stato effettuato in più momenti

un *multithreading*, mediante l'utilizzo delle funzionalità del package *threading*, come per esempio nella gestione della modulazione del volume della musica e per la barra di caricamento.

Il codice è stato strutturato in classi:

- Una prima classe per la musica che gestisce il pulsante on/off e il crescendo iniziale del volume;
- Una seconda classe che gestisce l'interfaccia col microcontrollore;
- Una terza classe che gestisce tutta la restante parte di interfaccia; la classe è stata creata in questo caso per semplificare la gestione delle variabili comuni e rendere di fatto il programma un modulo.

3.1.2 Software - Terminal

Il software Python per la gestione dell'interfaccia da terminale si compone di due script separati, in virtù del principio di modularità. Vi è infatti il codice relativo al programma in sé e alla sua corretta esecuzione denominato "*Terminal.py*" e quello contenente le funzioni utili all'algoritmo, denominato "*TerminalSupport.py*". Il contenuto dei due script viene sintetizzato nelle due sezioni che seguono.

Contenuti dello script "*Terminal.py*"

Come accennato, questo file contiene l'algoritmo alla base del gioco. Quando esso viene chiamato dall'utente tramite lo script Bash di avvio, il codice definisce e inizializza le variabili più importanti che risulteranno utili durante la sua esecuzione, tra cui *ledstatus* utile a tenere traccia dello stato del led e *GameSet*, il quale definisce il numero di prove a cui l'utente dovrà partecipare. Assieme a queste si inizializza anche un piccolo dizionario, che servirà a contenere il nome e il relativo punteggio del futuro campione del gioco.

La parte di codice che segue, si occupa di indicare all'utente l'elenco delle porte seriali disponibili sul calcolatore in cui si sta eseguendo il gioco. A selezione compiuta, il codice procede nell'opportuna configurazione della porta con i seguenti parametri di comunicazione:

1. Baudrate = 112500 Baud/s;
2. Bitsize = 8 bit;
3. Numero di stop bit = 1;
4. Bit di parità disabilitato;
5. Protocolli di *handshake* disabilitati.

Se non venisse trovata alcuna porta seriale, il codice restituirebbe un messaggio di errore per poi interrompere l'esecuzione.

Viene dunque mandato un segnale di reset al microcontrollore predisposto a supporto del gioco e si procede entrando in un loop infinito che contiene il codice del gioco vero e proprio. La prima operazione svolta è costruire una semplice interfaccia utente, di tipo testuale, in cui vengono spiegati tutti i meccanismi di gioco. Più internamente è contenuto un ulteriore loop, questa volta ripetuto per un numero di cicli pari al numero di prove che l'utente dovrà affrontare. Questo contiene, oltre ai semplici passaggi di acquisizione e presentazione dei dati da e verso l'utente, anche la chiamata alle funzioni presenti nel file "*TerminalSupport.py*" che verificano la correttezza dei comandi inseriti e di generare i ritardi casuali necessari al funzionamento del gioco. Altre funzioni importanti sono quelle relative alle trasmissioni con il microcontrollore, quella che pulisce lo stream in ingresso ogni volta che si deve procedere con l'inserimento di un nuovo comando e infine quella che si occupa di identificare la richiesta dell'utente di uscire dal gioco prima dei tempi previsti dall'algoritmo. In questo contesto si tiene traccia dei vari risultati ottenuti dal giocatore così come descritto nella sezione di apertura. Al termine dell'esecuzione di questo loop interno, l'algoritmo valuta il tempo di risposta medio dell'utente e verifica se il suo migliore risultato sia a sua volta migliore di quanto raggiunto dal

campione del gioco, eventualmente aggiornandone i dati. Solo a questo punto il programma presenta all'utente i vari risultati.

È importante da sottolineare come i risultati relativi al miglior tempo di reazione vengano presentati anche se si esce prematuramente dal gioco. Tuttavia non verranno fornite, né la media né le informazioni relative alla vittoria o meno del gioco da parte dell'utente.

Al termine, il codice richiede se procedere con una nuova partita facendo quindi ripartire il loop di gioco, oppure se uscire interrompendolo.

Contenuti dello script *"TerminalSupport.py"*

Lo specifico script include al proprio interno le dichiarazioni di cinque specifiche funzioni utili all'algoritmo di gioco. In particolare, vi sono:

1. `availableSerialPort()` : Funzione utile all'identificazione delle varie porte seriali disponibili sul calcolatore di riferimento; ritorna una lista contenente i nomi degli elementi ricavati;
2. `Matchcommand(str,int)` : Funzione utile alla verifica della correttezza del comando fornito dall'utente in rapporto allo stato del led; ritorna un flag necessario ad identificare la condizione di corretta impostazione del comando e il flag aggiornato che codifica lo stato del led sul microcontrollore;
3. `DelayGenerator()` : Funzione utile alla generazione del ritardo casuale compreso tra 10s e 20s con risoluzione di 1ms;
4. `PrintMenu(list)` : Funzione utile a stampare a video l'elenco delle porte seriali disponibili sul calcolatore di riferimento e ad acquisirne la scelta dell'utente;
5. `PrintTitle()` : Funzione utile a pulire la schermata del terminale e a stampare il titolo del gioco.

Per un maggiore approfondimento di quanto trattato in questa sezione, si invita il lettore a considerare direttamente lo script fornito. Quest'ultimo è stato opportunamente commentato per esplicitarne le funzionalità principali.

3.2 Firmware - Nucleo board

Il codice scritto per il microcontrollore resta in attesa di un comando dalla seriale *USART1* (piedini PA10(rx) e PA9(tx)), i comandi inviati dal computer possono essere:

- **L1**: il led si accende e si aspetta che l'utente prema il pulsante;
- **L0**: il led si spegne e si aspetta che l'utente prema il pulsante;
- **R**: nel microcontrollore viene resettato lo stato del led al valore di default (LED spento); questo è un comando non presente nelle specifiche ma permette di ricominciare il gioco senza resettare manualmente la board e, inoltre, non pregiudica l'utilizzo della board stessa con altri programmi (per il PC) che non utilizzino R per eseguire il reset.

Oltre a queste funzionalità il programma rileva se l'utente sta barando, controllando lo stato del pulsante mentre si accende o si spegne il led; in particolare, se il pulsante è premuto il programma aspetta finché il pulsante non viene rilasciato e premuto nuovamente, in questo modo non è possibile riscontrare un ritardo nullo. Viene in pratica effettuato il rilevamento del fronte relativo alla pressione del tasto e non del livello logico.