

APPLICATION NOTE: RIPRODUZIONE MUSICA E GESTIONE VOLUME

EXTRA POINT 2 A.A. 2022/23 – TAMAGOTCHI

DE ROSSI DANIELE – S314796

OVERVIEW:

Tra le varie funzionalità implementate in questo progetto, è sicuramente degna di nota l'implementazione dei suoni attraverso il loudspeaker integrato, nonché la gestione del volume, attraverso l'utilizzo dei convertitori ADC e DAC presenti nella scheda **LandTiger LPC1768**.

Per ogni azione compiuta dal personaggio (in questo caso, **Super Mario**) viene riprodotta una sequenza diversa di **note**. Ad esempio, all'avvio del gioco viene riprodotta una sequenza di poche note estrapolate dal tema originale di Super Mario. In linea generale, viene riprodotta una sequenza di note più possibilmente coerente con l'azione da eseguire (inclusa la pressione del tasto SELECT).

Attraverso il **potenziometro**, invece, viene regolato il **volume** di tali suoni. In base alla **posizione** del potenziometro, una diversa icona del volume comparirà sul lato sinistro dello schermo, a partire dall'assenza di suono, per arrivare a valori intermedi fino al valore massimo possibile.

IMPLEMENTAZIONE:

Per implementare tali funzionalità si è fatto uso della libreria **music.h** contenuta all'interno del progetto **11_sample_MIDI_MUSIC**, e delle funzioni incluse in libreria **adc.h**, adattandole al contesto del progetto già precedentemente sviluppato.



Le quattro icone mostrate a schermo in base al valore del volume desiderato.



Il buzzer della scheda LandTiger attraverso il quale vengono riprodotti i suoni.

FUNZIONAMENTO:

La libreria **music.h** contiene al suo interno le definizioni delle note (basandosi sulle diverse frequenze di queste ultime) e dei tempi (minima, semiminima, croma, semicroma, ecc...). Essa inoltre definisce la funzione vera e propria di riproduzione della nota **playNote(NOTE note)**. Questa funzione riproduce la nota andando a impostare di conseguenza i tempi del **TIMER2** e del **TIMER3** rispettivamente in base a frequenza e durata desiderate. Questi timer sono utilizzati solamente per questo scopo.

Ad ogni raggiungimento del tempo sul **TIMER2**, il suo **handler** metterà in uscita sul **DAC** (e dunque un suono sullo speaker) il valore di una sinusoide definita con valori da 0 a 1024 (questi valori vengono scalati in base al valore corrente del volume). Il **TIMER3** viene inizializzato alla durata desiderata della nota, di fatto il suo **handler** disabiliterà il **TIMER2** andando dunque a fermare la riproduzione del suono.

Per poter riprodurre suoni, sono state definite una serie di note sui file **IRQ_Timer.c** e **IRQ_RIT.c**, in quanto le funzioni che usano la libreria **music.h** e riproducono suoni sono definite in questi file [**TIMER1_IRQHandler(void)** e **RIT_IRQHandler(void)**].

Il funzionamento del volume, invece, è basato sul valore in ingresso letto dall'**ADC** [**ADC_IRQHandler(void)**], ricavato dalla posizione del potenziometro. Una funzione all'interno del file di handling (**IRQ_adc.c**) definirà qual è l'icona del volume da disegnare tra le 4 disponibili (si trasforma il valore letto dall'ADC su **4 livelli diversi**, dove il minimo indica l'assenza di suono e il massimo i valori di volume più alti) [**check_volume(void)**].

Un'ulteriore funzione serve al **TIMER2** per tarare il valore della sinusoide in uscita sul **DAC** [**getVolume(void)**]. Più alto è il valore in ingresso sull'**ADC**, più alto sarà il volume del suono sul buzzer. Il valore in ingresso sull'ADC viene letto ogni 50ms dal **RIT**, altrimenti, nel caso in cui quest'ultimo stesse gestendo un comando, dal **TIMER1**, tramite la funzione **ADC_start_conversion(void)**.