

APPLICATION NOTE – EXTRAPOINT 2

Lorenzo Ceccarelli s269134

1. Struttura del progetto

1.1 Organizzazione del codice

Il codice è organizzato in cartelle e sottocartelle divise per funzionalità. (i.e. La cartella timer contiene i files per la gestione dei timer).

Inoltre il progetto è diviso in due macrosezioni che sono rispettivamente:

-sample.c il quale gestisce il setup al reset della scheda e

-IRQ_RIT.c che gestisce e coordina tutti i moduli che implementano le diverse funzionalità.

1.2 Gestione delle specifiche

- Maintenance Mode: tale modalità è gestita dalle funzioni RIT_IRQHandler() e TIMER3_IRQHandler().
- Emergency Mode: tale modalità è gestita dalle funzioni Entra_in_EmergencyMode() e Esci_da_EmergencyMode() [entrambe richiamate da RIT_IRQHandler].
- Spostamento: tale modalità è gestita all'interno delle funzioni RIT_IRQHandler() e Spostamento(). TIMER0_IRQHandler() emula lo spostamento dell'ascensore quando prenotato.
- Lampeggio Leds: tale modalità è gestita all'interno della funzione TIMER1_IRQHandler();
- Select non premuto dopo prenotazione: tale specifica è gestita con le funzioni GestionePrenotato_NoSelect() e TIMER3_IRQHandler();
- Gestione del loudspeaker: Tale specifica è gestita dalle funzioni AttivaSpeaker() e DisattivaSpeaker() richiamati all'interno delle funzioni che gestiscono l'Emergency Mode.
- Gestione del touch display: Il touch display viene gestito nel TIMER3_IRQHandler() che richiama funzioni di libreria per disegnare e aggiornare la GUI in seguito ai vari eventi.

N.B. Le funzioni che implementano le varie specifiche richiamano a loro volta delle sottofunzioni che gestiscono i casi in modo da rendere il codice modulare e il più facilmente comprensibile.

Inoltre sempre a tale scopo si consiglia di consultare il codice dove sono fornite informazioni più dettagliate sui diversi moduli, funzioni e variabili.

2 Gestione Potenzimetro

2.1 Funzionamento

Il potenziometro viene abilitato durante la Maintenance Mode quando l'utente seleziona il tono da modificare tra i due. La funzione che si occupa di questo è TIMER3_IRQHandler() che ogni 50 ms legge il valore in uscita dal ADC.

Tale valore viene mappato in un numero compreso tra 0 e numero_note-1 attraverso una divisione per DIV ottenuto come $DIV = \frac{4095}{numero_note}$ dove 4095 è il valore massimo restituito dal convertitore e numero_note è pari a 13. Il risultato viene usato come indice nel vettore di struct nota per ottenere la nota selezionata con il potenziometro. Da come si può dedurre ad una singola nota sono associati diversi valori del potenziometro.

2.2 Organizzazione del codice rispetto alla selezione del tono

Per quanto riguarda l'organizzazione del codice, quando l'utente seleziona il tono da modificare, viene richiamata la funzione Aggiorna_GUI_pressioneBottone(int n_bottone), dove n_bottone è la nota che si vuole modificare, che aggiorna l'interfaccia grafica andando ad inserire un bordo bianco nei campi selezionati. Dopo, ADC_start_conversion() e ADC_IRQHandler() leggono il valore in uscita dal potenziometro e dal ADC e lo mappano in un valore tra 0 e 12. Infine se il valore selezionato è diverso dalla nota corrente usata dal loudspeaker si aggiorna l'interfaccia grafica richiamando Aggiorna_GUI_selezioneTono().