APPLICATION NOTE

La specifica che ho deciso di descrivere è la specifica del display touch (Specifica 1). Innanzitutto, per verificare se lo schermo è stato toccato va verificato tramite un timer con elevata frequenza o tramite il RIT, in quanto il tocco non genera un interrupt, ciò significa che il programma stesso deve accedere ad un'opportuna variabile e verificare se il valore contenuto in essa ricade nell'area di interesse. Nel mio programma la verifica del display avviene tramite il RIT in quanto l'interruzione con il timer generava dei conflitti con lo speaker non facendogli riprodurre le note correttamente. Ogni volta che il RIT HANDLER è invocato, vi è una funzione chiamata touchFunction(), la quale verifica se lo schermo non sia stato toccato recentemente, perché bisogna controllare se non ci sia già una coccola in corso, e in tal caso recuperare le coordinate di dove è stato premuto. Una volta recuperate, testa se il punto ricade nell'area di interesse ovvero in prossimità del disegno del personaggio. Nel caso in cui questa verifica sia soddisfatta viene invocata la funzione printCuddles(0xF800), che permette di stampare sullo schermo due immagini di colore rosso, il colore è dato dal parametro della funzione, e aggiorna una variabile cuddlesPrinted=1 per indicare che vi è una coccola in corso. Questa variabile è dichiarata nel modulo del RIT come extern int in quanto presente anche nel IRQ_timer.c come volatile int dato che il suo valore è aggiornato in due punti distinti del programma. Lo scopo della variabile cuddlesPrinted è quello di far durare l'immagine 2 secondi nel caso in cui è stato toccato lo schermo. La verifica della durata viene gestita dal timer0 il quale genera un interrupt ogni secondo. Di conseguenza se la variabile cuddlesPrinted assume valore uno consente di eseguire un blocco di istruzioni interne al timer0. Supponendo che abbia valore 1 e l'Handler del timer0 è invocato si ha l'aggiornamento della variabile cuddlesTime, la quale viene semplicemente incrementata di uno. Nel caso in cui assume valore 1 l'immagine viene cancellata e ridisegnata spostata. Questa operazione avviene tramite una chiamata alla funzione moveCuddles(int xMove, int yMove) la quale cancella la figura, o meglio la ridisegna di colore bianco, e la disegna nuovamente di colore rosso ma spostata di un offset x, y passati come parametri alla moveCuddles. La sua implementazione è posta nel modulo character.c, la quale effettua una chiamata a deleteCuddles(), aggiornare due variabili necessarie per cambiare le coordinate di dove disegnare e richiama la printCuddles() con il relativo colore. Questo è ciò che accade quando il timer0 è eseguito per la prima volta dopo aver toccato lo schermo, ciò significa che è trascorso 1 secondo. Quando l'handler del timer0 viene eseguito nuovamente significa che sono trascorsi 2 secondi, allora le coccole devono terminare quindi cuddlesTime assume valore 2, allora il timer0 cancella la figura, ripristina le variabili cuddlesPrinted=0 così nessuna coccola è disegnata e cuddlesTime=0. Una volta che le coccole terminano bisogna incrementare di un livello l'happiness, di conseguenza si verifica se tale batteria sia piena o meno. Nel caso in cui è piena non bisogna aggiornare nulla, in caso contrario bisogna disegnare una nuova barra nella batteria dell'happiness tramite la funzione increaseHappinessLevel() la quale disegna il nuovo livello della batteria e aggiornare la variabile happinessLevel che tiene traccia delle barre presenti nella relativa batteria.