

1 Display

1.1 Inizio

Nel file *sample.c* ho inizializzato tutti i componenti necessari e chiamato la funzione *start()*, definita in *IRQ_RIT.c*, che setta i valori iniziali e disegna la schermata iniziale. Al suo interno è inoltre attivato il *timer0*, il quale gestisce il tocco dell'intera schermata per far avviare la partita.

1.2 Restart

La restart è gestita dal *timer1*. Una volta cliccata l'area del pulsante, viene semplicemente richiamata la funzione *start()*.

1.3 Clear

Anche la clear è gestita dal *timer1*, ma in questo caso viene controllato anche il flag di vittoria *win*. Una volta cliccato nell'area del pulsante, viene richiamata una versione modificata della *LCD_Clear()*, la *clearMatrix()* (in *GLCD.c*), che pulisce la parte di schermo dove si trova la matrice di gioco e ridisegna matrice e robot. Ho agito in tale modo per non occupare memoria salvando la posizione degli ostacoli scoperti e per non usare la *LCD_DrawLine()*, che risulta più lenta.

2 Joystick

Le varie funzionalità del joystick sono gestite dal *RIT* in *IRQ_RIT.c* e attivabili solo se il flag *win* è a zero.

2.1 Select

La select modifica la modalità di gioco e ridisegna il triangolo nella stessa posizione, ma di colore differente (giallo per la explore, verde per la move) con la funzione *triangle()*, alla quale passo la posizione del pixel dell'angolo superiore sinistro della casella dove va disegnato e il colore, mentre l'orientazione è definita dalla variabile globale *dir*.

2.2 Explore

La explore è gestita nel primo if di ogni pressione del joystick. Sfruttando la funzione *triangle()*, cancella e ridisegna il robot secondo la nuova direzione. Viene poi richiamata la funzione *obst()* che, sfruttando la distanza precedentemente calcolata con *dist()* e il flag *wall* che indica la presenza di un muro/ostacolo (per differenziare dal caso in cui sia terminata la matrice), riempie la cella contenente l'ostacolo con la *LCD_DrawLine()*.

2.3 Move

La move è gestita per ogni pressione del joystick come alternativa alla explore. Se ci si vuole spostare in una direzione diversa da quella attuale semplicemente ridisegna il robot con la *triangle()*, mentre se la direzione è la stessa controlla che la distanza sia maggiore di zero, quindi che non ci siano ostacoli frontali, e solo in quel caso avanza ridisegnando il robot. Alla fine, se sono nel caso in cui il robot si è spostato, controllo se ho raggiunto l'uscita con la funzione *check()*, che disegnerà una schermata di vittoria e setterà a 1 il flag di vittoria *win*.