Cazzola Michele s323270

**APPLICATION NOTE**

Il progetto extrapoint 2 è stato testato utilizzando lo stesso codice su entrambe le schede.

**Comunicazione**

La comunicazione avviene sul canale CAN1, configurato per ricevere solo messaggi con id pari a 1; il canale CAN2 non è attivato. Non si considerano i casi di invio simultaneo di messaggi: di conseguenza, si ipotizza che nella fase iniziale i due giocatori premano INT0 in momenti diversi, senza provocare collisioni sul canale.

## **Handshake**

Avviene sul CAN1, con messaggi aventi *id=1* e prefisso *data[0] = 0xFF*; si effettuano le seguenti ipotesi iniziali

* Se un giocatore entra nel menù, è sempre intenzionato a giocare e nella stessa modalità dell’avversario (single/double board)
* Se i giocatori giocano in single-board, pur essendo collegati, svolgono le partite in modo indipendente, ma al termine della partita si allineano, ovvero ognuno attende che l’avversario abbia concluso la propria

Ogni board ha due variabili di riferimento:

* Modalità di gioco: se WAITING, il giocatore è nella fase iniziale di reset, se CHOOSING, il giocatore deve ancora effettuare le scelte, se READY il giocatore ha effettuato tutte le scelte ed è pronto per giocare, se PLAYING il giocatore sta giocando
* Stato dell’handshake: se HANDSHAKE\_OFF, non è in corso, se HANDSHAKE\_ON, è in corso la fase iniziale per scoprire se è presente una seconda board, se HANDSHAKE\_DONE, le board si sono scoperte a vicenda, se HANDSHAKE\_READY, l’avversario è pronto per giocare (e inizia il turno)

Il protocollo segue i seguenti passi:

* Inizialmente, le board sono in WAITING e HANDSHAKE\_OFF
* Quando la prima preme INT0, passa in CHOOSING ed HANDSHAKE\_ON e lo invia all’altra, inserendo l’informazione del suo stato in data[1]; avvia un timer di 5 secondi (timer 1), entro cui deve ricevere la risposta dalla seconda board: se non arriva, si considera solo la modalità di gioco single-board; inoltre, memorizza di essere il primo giocatore, in caso di double-board
* Quando l’altra board riceve il messaggio, entra in HANDSHAKE\_ON; dopo aver premuto anch’essa INT0, entra in CHOOSING e HANDSHAKE\_DONE, inviando tale stato alla prima board; inoltre, memorizza di essere il secondo giocatore, in caso di double-board
* Quando la prima board riceve il messaggio, entra in HANDSHAKE\_DONE ed ottiene la possibilità di muoversi tra le due opzioni
* Se entrambe scelgono double-board, la prima che conferma la scelta del tipo del proprio giocatore (human/NPC), entra in READY ed invia un HANDSHAKE\_READY all’altra, senza modificare il proprio stato di handshake
* La seconda entra in HANDSHAKE\_READY: dopo aver ricevuto il messaggio e confermato la scelta, entra in READY ed invia un HANDSHAKE\_READY alla prima
* La prima riceve il HANDSHAKE\_READY e il protocollo termina

## **Cambio turno**

In modalità single-board, il cambio di turno avviene come nell’extrapoint 1, ovvero invocando la funzione *setPlayer*; non si invia nessuna informazione sul bus CAN.

In modalità double-board, il cambio di turno avviene come segue:

* Reset informazioni del giocatore della board corrente
* Impostazione evidenziazione del giocatore avversario (si colora la sua cella in giallo)
* Invio della mossa all’avversario

**NPC**

Il giocatore automatico (NPC) è realizzato mediante una funzione dedicata (*NPC\_playTurn*), che effettua la risoluzione di un

problema di ottimizzazione locale; esso, ad ogni turno:

* Esplora tutte le mosse possibili
* Considera solo quelle valide (ovvero che rispettano le regole del gioco)
* Calcola, per ogni mossa, il numero di mosse necessarie a sé stesso e all’avversario per arrivare a destinazione, in seguito all’ipotetica esecuzione di quella considerata all’iterazione corrente: in particolare, esplora la matrice di gioco mediante una visita in ampiezza, inserendo le adiacenze valide in una coda FIFO, realizzata con buffer circolare e definita nella cartella di progetto *queue*
* Effettua la mossa che massimizza la metrica seguente:

dove *posNPC* è la posizione del NPC in seguito alla mossa effettuata

* Caso particolare: si utilizza un’euristica, per cui se il NPC può vincere in una mossa, muove la propria pedina, anche se inserire un muro risultasse vantaggioso, in riferimento alla metrica utilizzata

Non si impostano timer, in quanto l’esecuzione del turno è nell’ordine dei decimi di secondo, anche nel caso peggiore in cui tutte le mosse risultano essere valide.