

# APPLICATION NOTE

Il progetto 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.

## Handshake

Avviene mediante messaggi aventi *id* = 1 e prefisso *data[0] = 0xFF*; si effettuano le seguenti ipotesi iniziali:

- sul canale non si verificano mai collisioni: nella fase iniziale, i giocatori premono *INT0* in istanti diversi;
- 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 le schede collegate, svolgono le partite in modo indipendente, ma al termine della partita si allineano, ovvero ognuno attende che l'avversario abbia concluso la propria, prima di rientrare nel menù iniziale premendo *INT0*.

Ogni scheda ha due variabili di riferimento:

- modalità di gioco: *WAITING* (fase iniziale), *CHOOSING* (fase di scelta nei menù), *READY* (scelte effettuate, giocatore in attesa dell'avversario), *PLAYING* (fase di gioco o attesa del turno di gioco avversario);
- stato dell'handshake: *HANDSHAKE\_OFF* (fase iniziale, non in corso), *HANDSHAKE\_ON* (in corso la prima fase, per scoprire se è presente una seconda scheda), *HANDSHAKE\_DONE* (le schede si sono scoperte a vicenda), *HANDSHAKE\_READY* (l'avversario è pronto per giocare e il primo giocatore inizia il turno).

Il protocollo di comunicazione è il seguente:

1. le schede si scambiano un messaggio di handshake iniziale, per scoprirsi a vicenda: tale fase inizia, per ogni scheda, premendo su *INT0* (ed entrando in *CHOOSING*); se la differenza tra gli istanti di pressione è superiore a 3 secondi, si considera la seconda scheda come assente; se tale fase termina correttamente, le schede sono in *HANDSHAKE\_DONE*;
2. in single-board, i giocatori eseguono le operazioni di scelta in modo indipendente; in double-board, essi effettuano le scelte richieste, ponendosi in *READY*, inviando un messaggio di *HANDSHAKE\_READY* all'avversario ed attendendo da esso una risposta con lo stesso dato: al termine di tale fase, il primo giocatore inizia la partita.

## 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. Se l'avversario è il NPC, non si mostrano le mosse possibili né i muri in corso di posizionamento, ma si evidenzia solo in giallo la cella in cui la sua pedina si trova.

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

- reset informazioni del giocatore della scheda corrente;
- impostazione evidenziazione del giocatore avversario: si colora la sua cella in giallo, senza mostrare adiacenze valide né eventuali muri in corso di posizionamento;
- invio della mossa all'avversario.

## NPC

Il giocatore automatico (NPC) è realizzato mediante una funzione dedicata (*NPC\_playTurn*, definita in *NPC.c*, nella cartella di progetto *quoridor*), che effettua la risoluzione di un problema di ottimizzazione locale; esso, ad ogni turno:

- esplora tutte le mosse possibili, filtrando solo quelle valide (ovvero che rispettano le regole del gioco);
- calcola, per ogni mossa futura possibile, il numero di mosse necessarie a se stesso e all'avversario per arrivare a destinazione: 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*; tale esplorazione è effettuata mediante la funzione *measureDistance*, definita in *NPC.c*;
- effettua la mossa che massimizza la metrica seguente:

$$dist(pos_{avversario}, dest_{avversario}) - dist(pos_{NPC}, dest_{NPC})$$

dove  $pos_{NPC}$  è la posizione del NPC in seguito alla mossa considerata;

- caso particolare: si utilizza un'euristica, per cui se il NPC può vincere in una mossa, muove la propria pedina nella direzione opportuna, anche se inserire un muro risultasse vantaggioso, in riferimento alla metrica utilizzata.

Non si impostano timer, in quanto l'esecuzione del turno ha una durata nell'ordine dei decimi di secondo, anche nel caso peggiore dal punto di vista computazionale in cui tutte le mosse risultano essere valide.