

# Giuseppe Ciacco - S295982

## Architetture dei Sistemi di Elaborazione

Report Special Project  
A.A. 2021/2022

### 1 FASE 1 - CAN bus

#### 1.1 Comunicazione tra schede singole

L'interfaccia è realizzata a partire da quella fornita da KEIL ( \KEIL\ARM\Boards\Keil\MCB1700\CAN ) per LPC17xx. Sono state modificate le funzioni dell'interfaccia per adattarle alla scheda Landtiger Sono state aggiunte funzioni per trasmissione/ricezione di dati di dimensione arbitraria (>8 byte).

Commit di riferimento: **bb015b93** (vedere README.md)

/gitlab/nxp-landtiger/specialprojectphase1\_2022/tree/bb015b93c9c52adc6d9112c4d830440dc53fe4cc

#### 1.2 Comunicazione tra più schede (propagazione)

Per ogni scheda si usa una porta in input e una porta in output. Ogni messaggio ricevuto sulla porta in input viene inviato sulla porta in output, con struttura ad anello aperto. Le schede propagano il segnale senza conoscerne l'origine e senza avere un identificativo univoco proprio.

Commit di riferimento: **bb015b93** (vedere README.md)

/gitlab/nxp-landtiger/specialprojectphase1\_2022/tree/bb015b93c9c52adc6d9112c4d830440dc53fe4cc

##### 1.2.1 DEMO

Verifica della trasmissione di dati di dimensione arbitraria (16 blocchi da 8 byte), verifica della corretta propagazione dei messaggi

**Configurazione schede (vd. video)** Il file sample.c deve essere modificato in CAN\_receiveBuffers(...) e CAN\_transmitBuffers(...) nel ciclo while per far corrispondere i collegamenti.

**Disposizione:**

D    A  
C    B

**Collegamenti:**

CAN\_1(A) <-> CAN\_1(B)  
CAN\_2(A) <-> CAN\_2(D)  
CAN\_2(B) <-> CAN\_1(C)  
CAN\_2(C) <-> CAN\_1(D)

(C impostato per non ricevere il messaggio, disabilitando CAN\_receiveBuffers)

Premendo il pulsante INT-0 su C, il messaggio (124 byte) è passato da C a D direttamente, propagato da D ad A e da A a B

C -> D -> A -> B

**Video DEMO:** <https://www.youtube.com/watch?v=o5lLoL9SIAE>

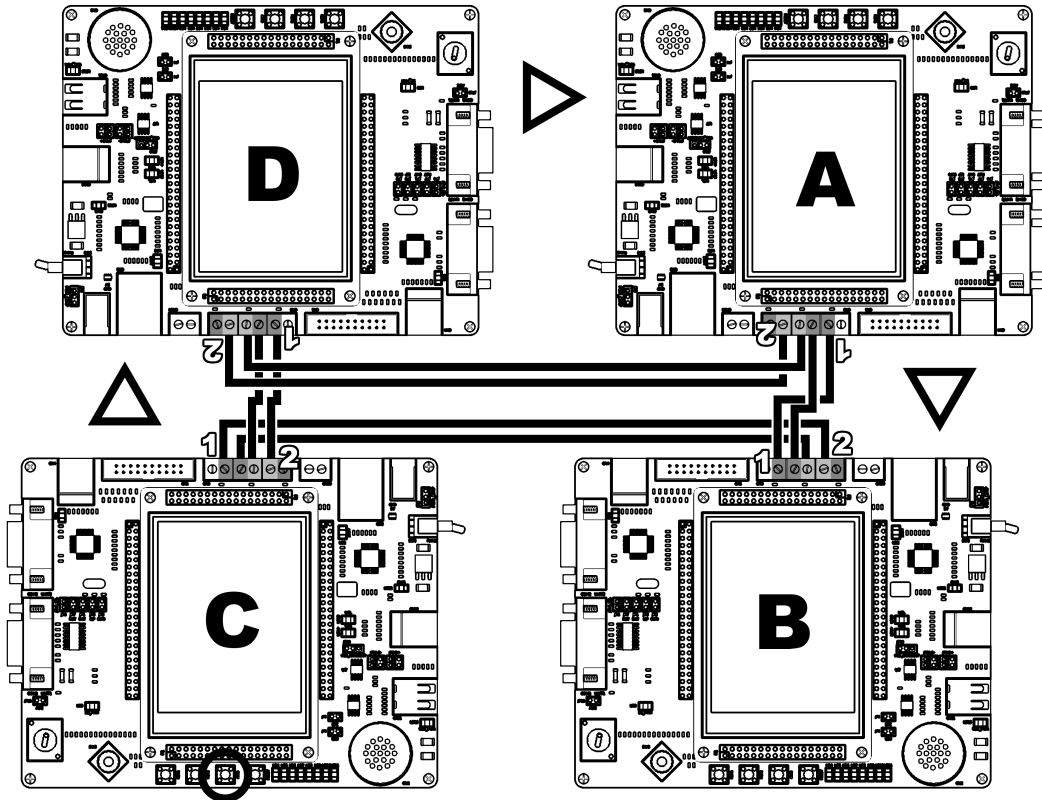


Fig. 1: Configurazione schede DEMO 1.2.1

### 1.3 Comunicazione tra più schede (anello e identificatore univoco)

Per ogni scheda si usa una porta in input e una porta in output. I messaggi vengono propagati alla scheda successiva fino a destinazione. In caso la destinazione non sia presente, il messaggio torna al mittente e non viene propagato ulteriormente.

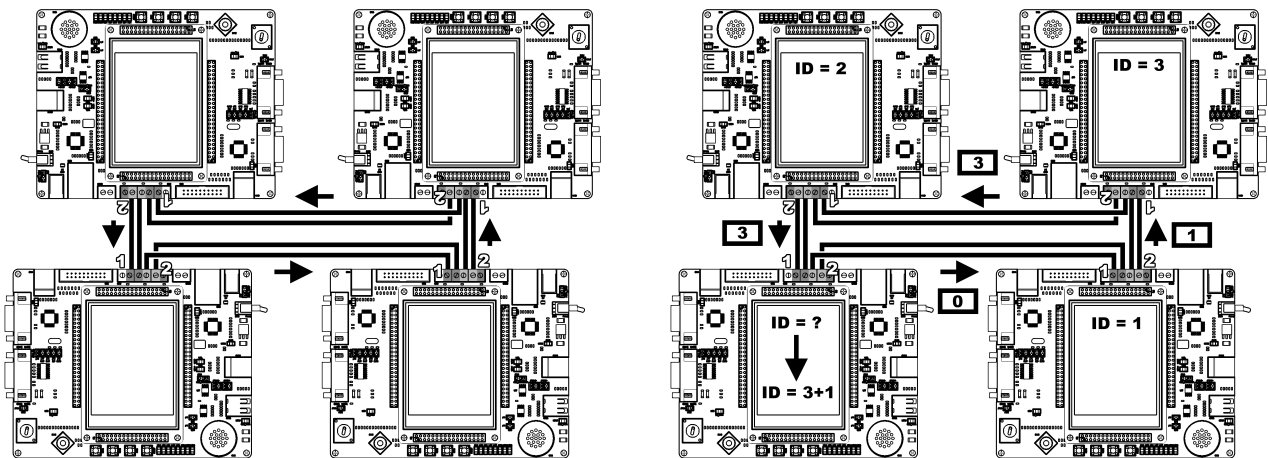
Alle schede viene assegnato un identificativo univoco, attraverso propagazione di un messaggio con cui si ottiene l'identificativo su cui calcolare il successivo.

Commit di riferimento: **6a1f1c4e** (vedere README.md)

/gitlab/nxp-landtiger/specialprojectphase1\_2022/tree/6a1f1c4e2025801bcf018a7860f4c96ce48bf241

#### 1.3.1 DEMO

Configurazione:



**Pulsante INT0:** Connessione alla rete ad anello (assegnazione ID)

**Joystick:** Selezione dell'ID su cui inviare (joystick su/giù) e invio messaggio (joystick select)

Gestione messaggio verso ID non sulla rete: percorre interamente l'anello e viene scartato

**Video DEMO 1:** <https://www.youtube.com/watch?v=X3Ao-fzE2yc>

**Video DEMO 2:** <https://www.youtube.com/watch?v=fmzs-KFtZm0>



### 2.2.1 Schermate

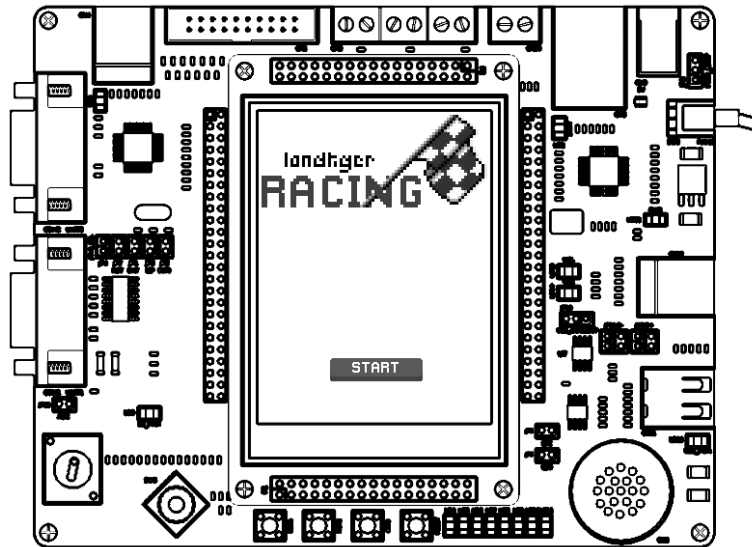


Fig. 5: Schermata iniziale  
Comandi: click joystick per avviare il gioco

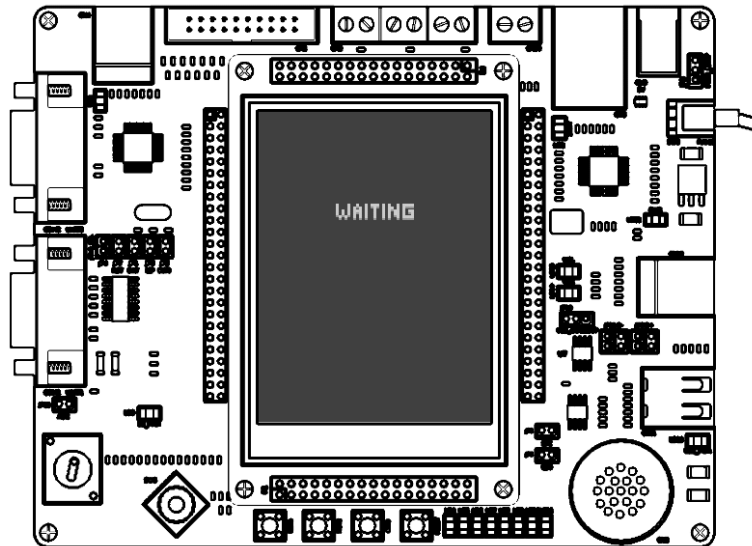


Fig. 6: Schermata attesa  
Attesa di collegamento del secondo giocatore

#### Punteggi:

- Verde: +1
- Rosso: -1
- Giallo  $\pm 2$

#### Elementi a schermo:

- Auto dei due giocatori
- Punteggi (in alto a sx)
- Distanza percorsa (in alto a dx, fino a 100)
- Indicatore della distanza percorsa (barra a dx)
- Collezionabili verdi/rossi/gialli

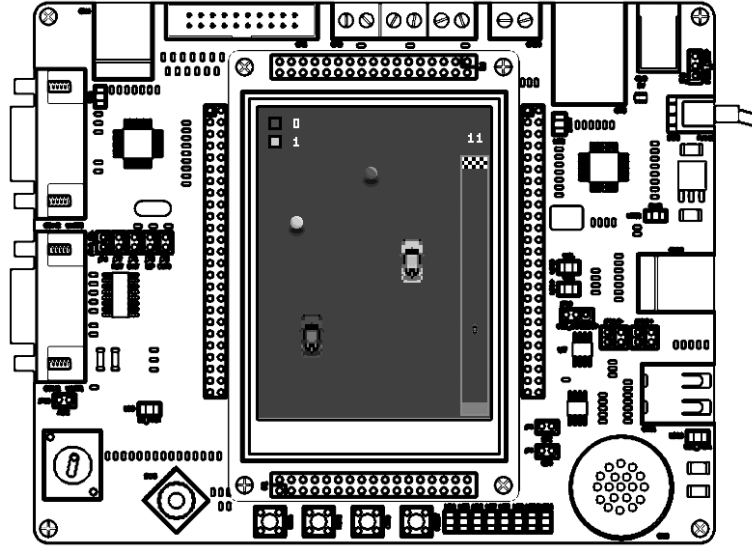


Fig. 7: Schermata di gioco  
Comandi movimento joystick per spostamento auto del giocatore

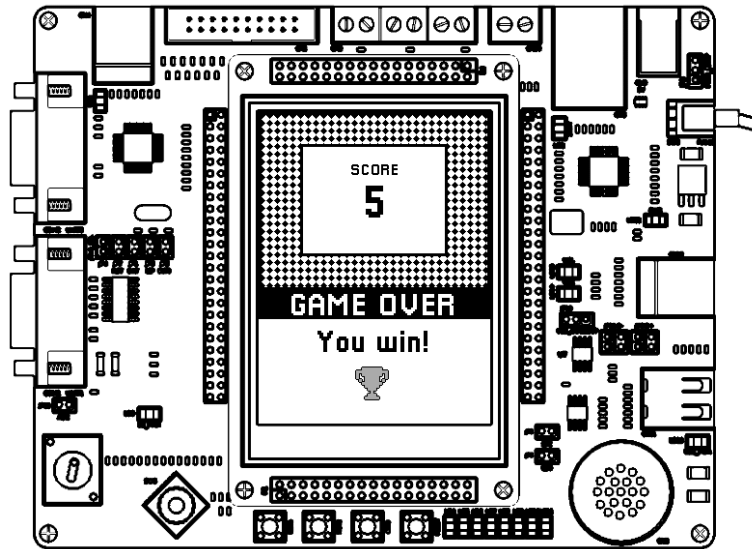


Fig. 8: Schermata finale  
Comandi click del joystick per tornare al menu principale

## 2.3 Sincronizzazione tramite CAN

### 2.3.1 Sincronizzazione a inizio partita

A inizio partita una delle due schede rimane in stato di WAIT fino a quando una seconda scheda non si collega al CAN. Per fare questo la prima scheda a connettersi invia un messaggio sul canale e rimane in attesa di ricevere una risposta da un'altra scheda. La seconda scheda riceve il messaggio dalla prima, senza bloccarsi in attesa e la sblocca inviando un messaggio sul canale.

### 2.3.2 Assegnazione del giocatore

A una assegnazione tramite comando del giocatore (tramite numero player selezionato dal giocatore prima dell'esecuzione del gioco o con uno schema simile a quello della DEMO 1.3.1, premendo un pulsante per ricevere un ID univoco) si è preferita un'assegnazione dinamica, basata sui tempi di attesa della sincronizzazione iniziale. Dal momento che entrambe le schede inviano il messaggio sul canale prima di entrare in attesa, solo una di esse entrerà in stato di wait, pertanto ogni scheda può identificare sè stessa senza dover comunicare con le altre. Questo schema **non** funziona in una configurazione a 3+ schede, per cui è necessario un protocollo di assegnazione degli ID con comunicazione tra le schede

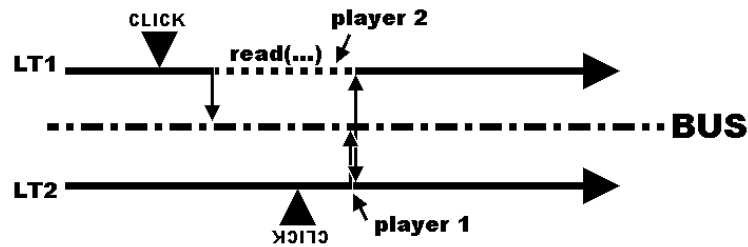


Fig. 9: Sincronizzazione iniziale e assegnazione del giocatore

### 2.3.3 Sincronizzazione durante l'esecuzione

Durante l'esecuzione del gioco deve essere effettuata la sincronizzazione dei seguenti valori

- **Posizione del giocatore:** una scheda invia all'altra la posizione del proprio giocatore e viceversa.  
data[0] e data[1]
- **Distanza percorsa:** le due schede inviano la loro distanza, in questo modo si risolvono disallineamenti dovuti a durate differenti dei frames  
data[2] e data[3]
- **Punteggio:** una scheda invia all'altra il proprio punteggio e viceversa  
data[4] e data[5]
- **Collezionabili:** i collezionabili sono generati casualmente (massimo uno per frame) da **una sola** scheda (quella del giocatore 1) e inviati all'altra scheda  
data[6] e data[7]

## 2.4 Estensione a 3/4 giocatori - progetto

Per estendere il gioco a 3/4 giocatori è possibile usare lo stesso codice del gioco con 2 giocatori, al quale dovrebbero essere apportate le seguenti modifiche

- **Grafica:** Punteggio dei 4 giocatori a schermo e riduzione delle dimensioni delle auto per permettere mobilità maggiore sullo schermo
- **Sincronizzazione a inizio partita:** in questo caso è sufficiente estendere lo schema su più schede con una configurazione ad anello
- **Assegnazione del giocatore:** non si può adottare lo schema usato nel codice per due giocatori, ma si può includere lo schema usato nella DEMO 1.3.1 in cui ogni scheda effettua richiesta per un id univoco e, in base all'ID assegnato si seleziona un giocatore
- **Sincronizzazione durante l'esecuzione:** anche in questo caso è sufficiente estendere a uno schema ad anello in cui però è necessario trasmettere più messaggi per trasferire tutti i dati necessari

**Video DEMO:** <https://www.youtube.com/watch?v=SB6NhghxTP4>