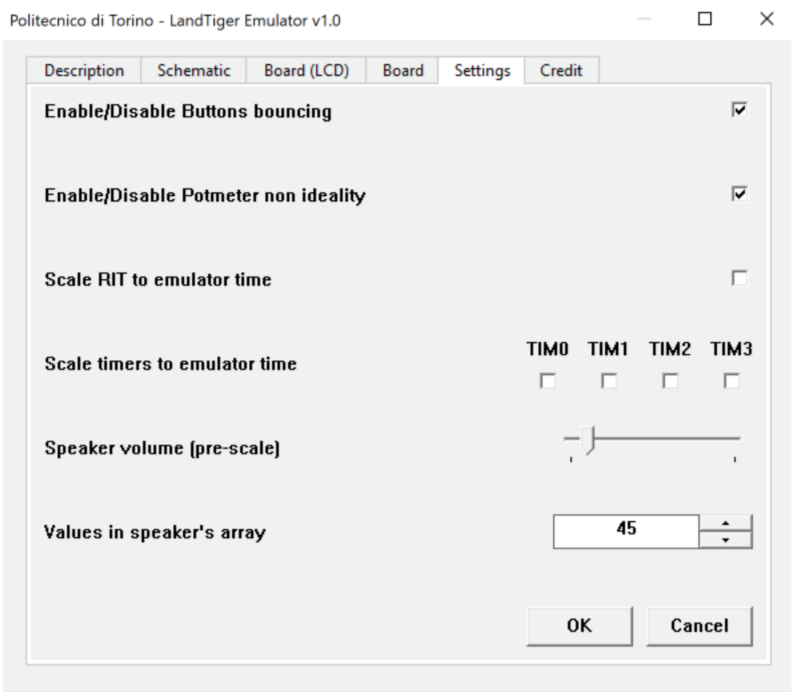
Architetture dei Sistemi di Elaborazione	Delivery date: 23/01/2022
Extra Points Part 2	Solving this track gets you a *max of 2 additional points*. Expected delivery of extrapoints1.zip must include: - Zipped project folder

Using the LandTiger Emulator available on Keil uVision, you are requested to implement the following **multi-player** game. Please note that all the non-ideality behaviors (see Figure 1 below) must be enabled and considered by your software. The RIT and Timers scaling switches can be enabled for debugging faster the project.


Politecnico di Torino - LandTiger Emulator v1.0

Description | Schematic | Board (LCD) | Board | Settings | Credit

Enable/Disable Buttons bouncing	✓
Enable/Disable Potmeter non ideality	✓
Scale RIT to emulator time	☐
Scale timers to emulator time	TIM0 ☐ TIM1 ☐ TIM2 ☐ TIM3 ☐
Speaker volume (pre-scale)	
Values in speaker's array	45

Figure 1 - Emulator Configuration

The game called "*Pong*" implemented by your software must reproduce the behaviour of the classic table tennis-themed arcade game, originally released in 1972 by Atari.

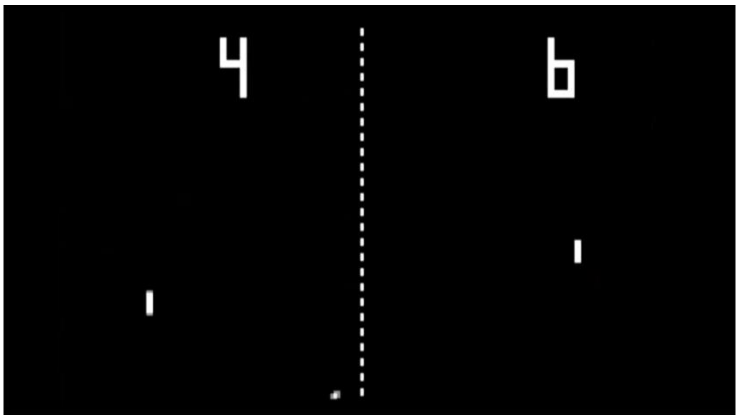


Figure 2 – Screenshot of a Pong match

For this project, start by implementing a **multi-player** version of the game where the player must not let the ball fall below the paddle. The bottom paddle is driven by the **first player**, and it can only move horizontally. The first player operates it through the potentiometer available in the LandTiger board. On the contrary, the top paddle is driven by the **second player**. It is **fully automatic** and, when the play starts, it only moves horizontally **at a constant speed.** The choice on **how fast** the top paddle moves is left unconstrained.

The game field should be implemented vertically, i.e., the bottom paddle will be in the bottom portion of the LCD display with red walls (5px thick) on the left and right portion of the display for the ball to bounce on, as represented in the Figure below. The top paddle will be in the top portion of the LCD display and has the same size of the bottom paddle.

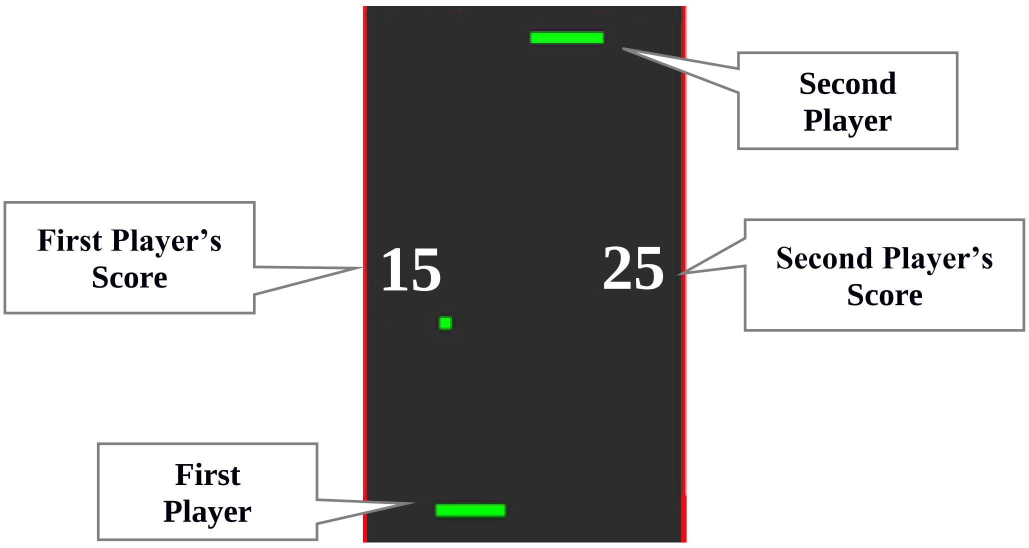


*Figure 3 – Game field for a **multi-player** match*

As for the previous lab track, when the ball hits a wall, it should bounce with a reflection angle equal to the incident one. For instance, if a ball going downwards hits the left **wall** with an incident angle of 30°, then it will bounce downwards with the same 30° angle, as shown below.

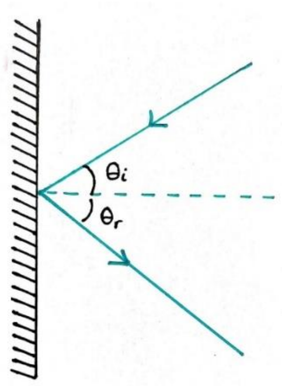


Figure 4 – Ball bouncing on a wall

When the ball hits the paddles, the reflection angle **must not always be equal** to the incident one, for example, depending on what part of the paddle is hit by the ball. It is up to you to pick and implement a strategy. The choice on **how fast** the ball moves is left unconstrained. Complex solutions, e.g., spinning the ball by hitting it with the paddle at a certain angle, will be considered when evaluating the project.

The bottom paddle should be **32px high from the bottom part** of the screen and **10px thick**, and the **ball should be a 5x5px square**. The top paddle should be **32px lower from the top part** of the screen and **10px thick**.

Every time the ball falls below one's paddle, the other player's score is incremented by 1. The game ends as soon as one player scores 5 points, with the message "You win" or "You lose" displayed on the screen for both players. The score must be printed on the left side of the screen for the first player and on the right side for the second player, at mid high (160px from both the bottom and top part of the LCD).

The user should **start the game** by pressing the button KEY1. When the game starts, the ball should be touching the right wall at mid high (160px from both the bottom and top part of the screen) and go downwards with an angle of 45°. To **pause the game and later resume it**, the user should press the button KEY2. In case of **game over**, to **prepare a new game** the player should press the INT0 button, followed by KEY1 to start it.

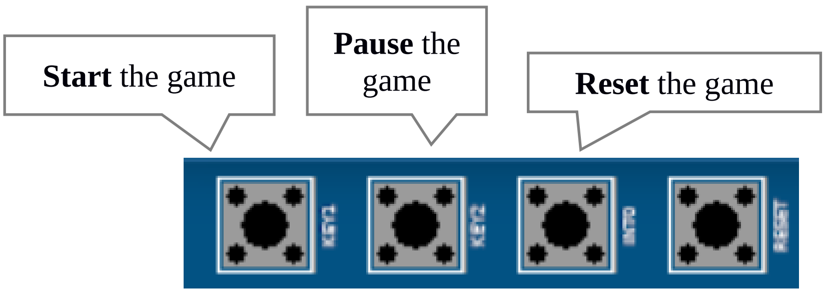


Figure 6 – Buttons configuration