

Expected delivery of extrapoints1.zip must include:

- zipped project folder;
- 4 minutes video that describe your project.

Using the LandTiger Emulator available on Keil uVision, you are requested to implement the following single-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.

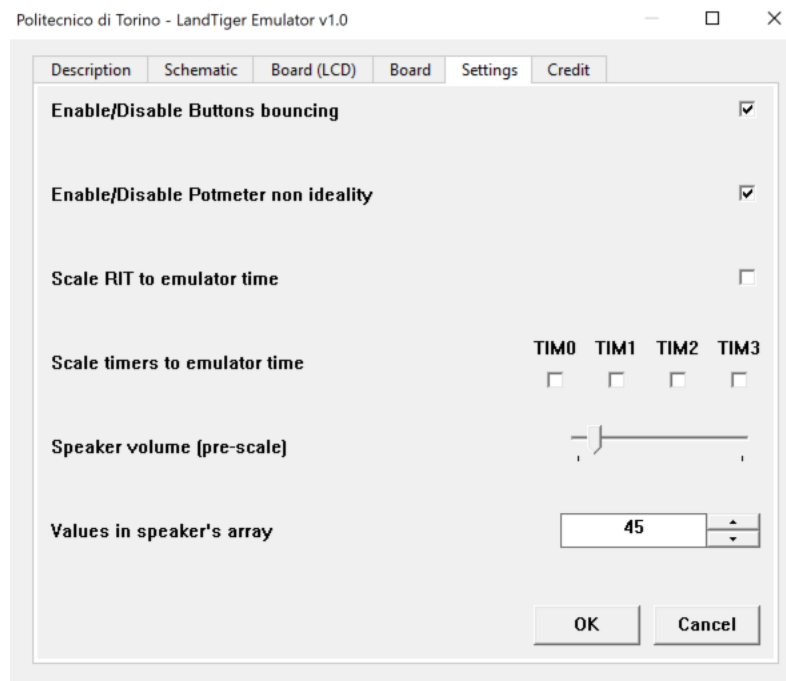


Figure 1 - Emulator Configuration

The game called “Blind Labyrinth” implemented by your software must reproduce the behaviour of a two-wheel robot equipped with a proximity sensor (which signals the distance from obstacles) and controlled remotely by a controller:

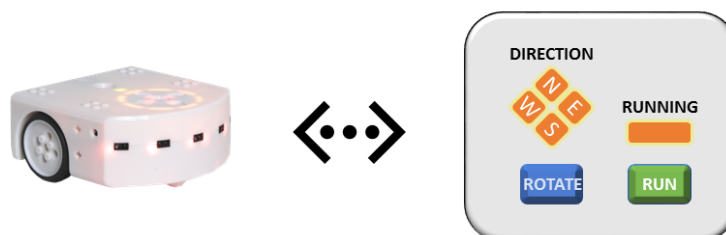


Figure 2 - Robot and controller

The player drives the robot through the controller in the labyrinth, trying to reach one of four possible exits. The player does not know the labyrinth structure, nor the obstacle it contains. Since the game is called “Blind Labyrinth”, the player has not any visual control over the robot. He decides the robot movements depending on the information returned by the proximity sensor.

The labyrinth (with the obstacles) in which the robot moves is the following:

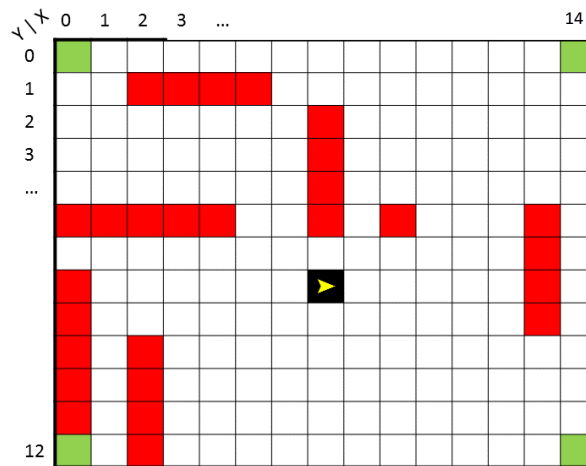


Figure 3 - The labyrinth

The labyrinth can be encoded as a 13x15 matrix and the robot can proceed at the speed of **1 slot per second**. Each slot is univocally identified by its coordinates Y,X and can be either an exit (for example slot 0,0 in green), an obstacle (slots 1,2 and 1,3 in red) or free ground (white slots). The robot cannot move over red slots, but on the white ones only. When it reaches one of the green slots, the game terminates.

The robot starting position is 7,7 with direction EAST.

HINT: The labyrinth can be encoded as a constant 13x15 matrix in the program, being each slot one byte wide. The free ground can be encoded with a 0, the obstacles with a 1 and the exits with a 2.

The controller, which sends commands to the robot and receives information about the proximity sensor, is made of five LEDs and two buttons.

The LEDs are:

- 1 RUNNING LED;
- 4 DIRECTION LEDs: North-East-West-South.

The Running LED blinks at 1Hz **when the robot moves** (when not moving is OFF). It blinks at 5Hz when the robot is blocked due to:

- Labyrinth perimeter;
- Obstacle.

The DIRECTION LEDs **are active one at a time depending on the robot current direction**. Whenever (**i.e., at any time instant**) the proximity sensor detects an obstacle **in front of the robot**, the DIRECTION LED currently active blinks with the following specifications according to the distance from the obstacle (Figure 4):

- Greater than 5 slots: No blink;
- Between 5 (included) and 3 (included) slots: 2Hz blinking (Figure 4.a);
- 2 slots: 4Hz blinking (Figure 4.b);
- 0 slots (robot right in front of the obstacle): 8Hz (Figure 4.c). **NOTE: Also the RUNNING LED must blink at 5Hz.**

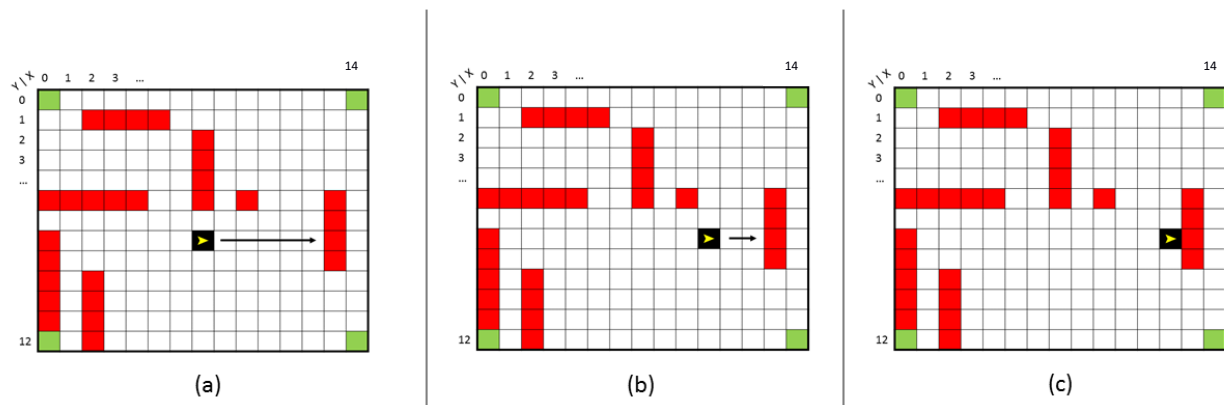


Figure 4 - Distance from obstacles and led blinking

Please note that the perimeter walls must be considered as obstacles. Therefore, the same specifications above are equally valid. Thus, considering Figure 5.b, the led must blink to signal the wall in front of the robot. **However, if the robot is pointing towards one of the exits (Figure 5.a) then the led must not blink.**

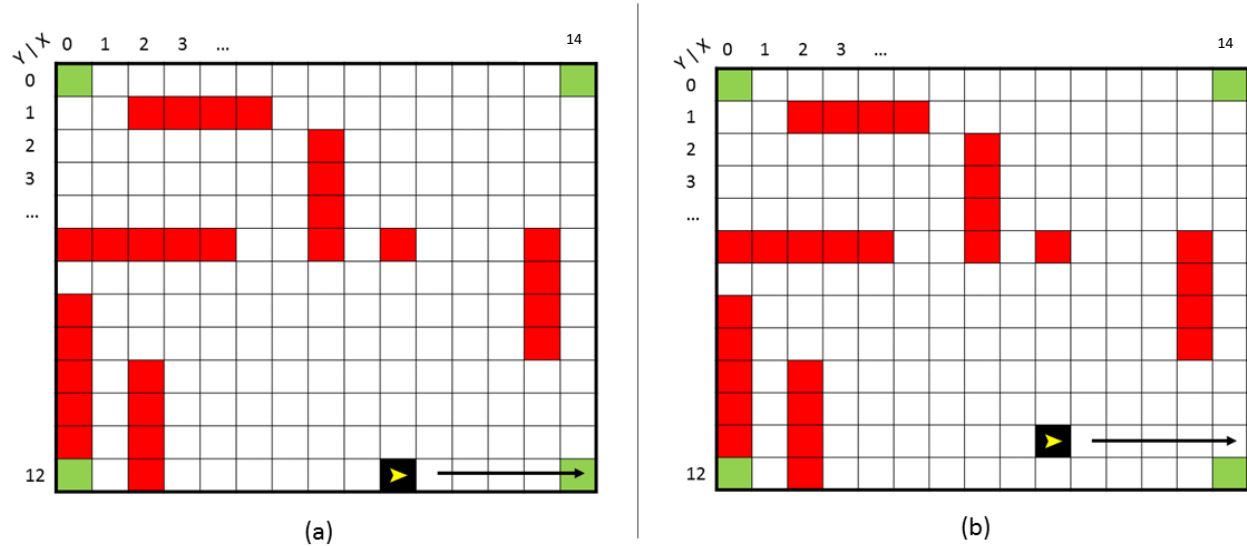


Figure 5 - Perimeter behavior

To move the robot, two buttons are available on the controller:

- ROTATE button: Press the button to change robot direction (clockwise rotation);
- RUN button: **Press and Hold** the button to move the robot in the current direction. When the button is released, the robot stops its movement. When pressed again, the robot continues to move. The RUNNING LED must behave accordingly.

When the robot is in the position of Figure 4.c, the player must not be allowed to proceed in that direction. That is, any pressure of the RUN button does not change the robot position. The player must first release the RUN button. Then, press the ROTATE button (the rotation must be applied when the button is actually released). Once released, press again RUN button to continue the movement.

Considering the LandTiger board, the five LEDs and the buttons are mapped as in Figure 6.

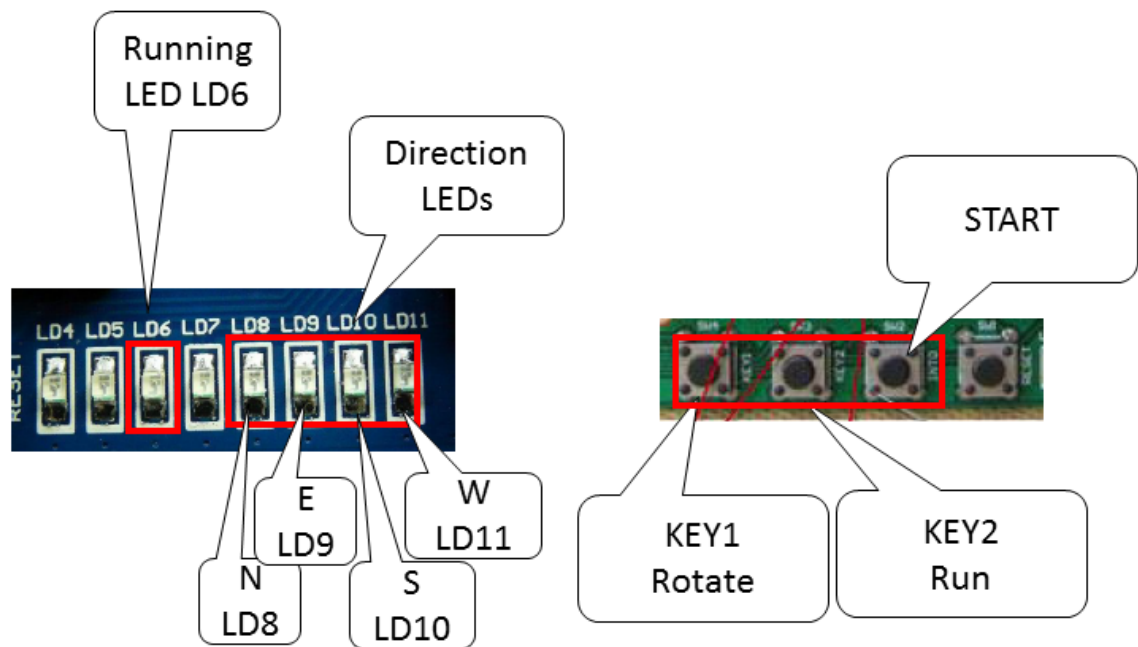


Figure 6 - LandTiger HW mapping

The START button on INT0 serves the purpose of starting the game. When the player reaches one of the exits, all the LEDs endlessly blink at 1Hz. To start a new game, the player must press the RESET button.

Appendix

The blink frequency must be intended:

- Duty cycle of 50%
- Half period low, half period high

For example:

- 1Hz: 0.5s off, 0.5s on – Period 1s
- 2Hz: 0.25s off, 0.25s on – Period 0.5s
- 0.5Hz: 1s off, 1s on – Period 2s