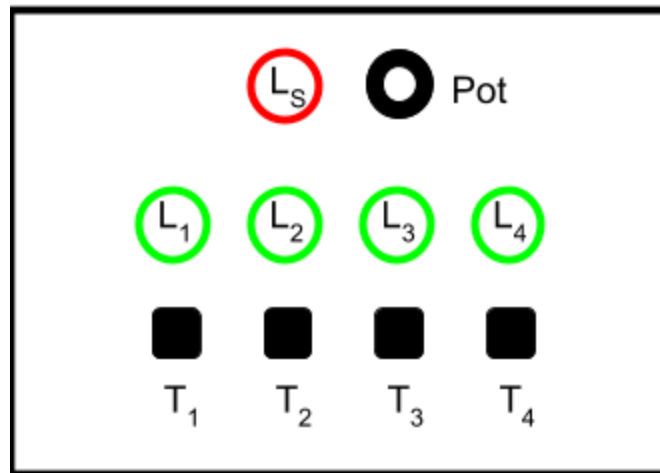


## Assignment #1 - *Catch the Led Pattern!*

We want to realise an embedded system implementing a game called *Catch the Led Pattern!*.

### Description

The game board is based on 4 green leds  $L_1$ ,  $L_2$ ,  $L_3$ ,  $L_4$  and red led  $L_S$ , four tactile buttons  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  and a potentiometer Pot, displaced in the following layout:



During the game, repeatedly the leds  $L_1 \dots L_4$  are first turned off for some random time  $T_1$ , and then turned on according to some random configuration or pattern (called here pattern). An example of pattern is:  $L_1$  and  $L_3$  on,  $L_2$  and  $L_4$  off. After some time  $T_2$  the leds are turned off and the player must recreate the pattern by pressing the corresponding buttons  $T_1 \dots T_4$  within some time  $T_3$ . Each button  $T_i$  turns on the corresponding led  $L_i$ . If the player recreates the correct pattern on time, the score is increased and the game goes on, by reducing the times  $T_2$  and  $T_3$  of some factor  $F$ . If the player does not recreate the correct pattern on time, she/he gets a penalty. The penalty is signalled by the red led  $L_S$  is turned on for 1 second. After three penalties the game ends and the score is displayed on the serial line.

### Game detailed behaviour

In the initial state, all green leds are off but led  $L_S$  that pulses (fading in and out), waiting for some player to start the game. On the serial line, it must be sent the message "Welcome to the Catch the Led Pattern Game. Press Key T1 to Start".

If/when the button  $T_1$  is pressed the game starts. If the  $T_1$  button is not pressed within 10 seconds, the system must go in deep sleeping. The system can be awoken back by pressing any button. Once awoken, the system goes in the initial state and the led  $L_s$  starts pulsing again.

When the game starts, all leds are switched off and a "Go!" message is sent on the serial line. An initial score is set to zero.

During the game:

- the leds  $L_1 \dots L_4$  are turned off for some random time  $T_1$
- the leds are then turned on according to some random pattern, for some time  $T_2$  and then turned off again
- the player has max  $T_3$  time for recreating the pattern by pressing the buttons  $T_1 \dots T_4$  (each button  $T_i$  turns on the corresponding led  $L_i$ )
- the player can act only after the leds are turned off
  - if the player press buttons before, then she/he gets a penalty (see later)
- If the player recreates the correct pattern on time, then:
  - the score is increased and a message "New point! Score: XXX" (where XXX is the current score) is sent on the serial line
  - the game goes on, by reducing the times  $T_2$  and  $T_3$  of some factor  $F$ .
- If the player does not recreate the correct pattern on time, she/he gets a penalty: the red led  $L_s$  turned on for 1 second and a message "Penalty!" is sent on the serial line.
- After three penalties the game ends: a message "Game Over. Final Score: XXX" (where XXX is the final score) is sent on the serial line for 10 seconds, then the game restarts from the initial state.

Before starting the game, the potentiometer Pot device can be used to set the difficulty  $L$  level which could be a value in the range 1..4 (1 easiest, 4 most difficult). The level must affect the value of the factor  $F$  (so that the more difficult the game is, the greater the factor  $F$  must be).

---

### The assignment:

- Develop the game on the Arduino platform, implementing the embedded software in C using the Wiring framework. The game must be based on a superloop control architecture.
  - Choose concrete values for parameters in order to have the best game play.
  - For any other aspect not specified, make the choice that you consider most appropriate.
- The deliverable must a zipped folder **assignment-01.zip** including two subfolders:
  - src
    - including the Arduino project source code

- doc, including
  - a representation of the schema/breadboard using tools such as TinkerCad or Fritzing or Eagle
  - a short video (or the link to a video on the cloud) demonstrating the system