Embedded System and IoT - a.y. 2023-2024

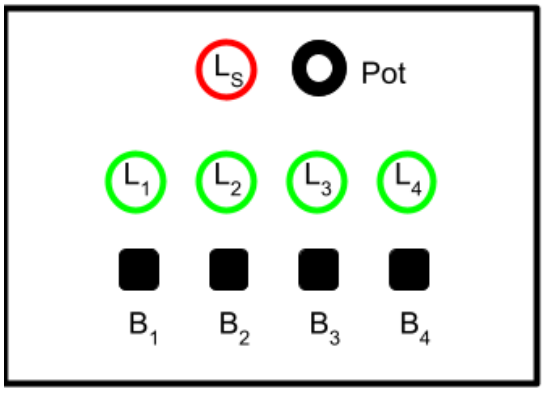
v1.0 - 20231004

Assignment #1 - *Restore the Lights!*

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 B_1, B_2, B_3, B_4 and a potentiometer Pot, displaced in the following layout:



During a game, the leds L_1…L_4 are initially all on. Then, after some random time T_1, the leds start turning off, one by one in sequence, in some random order, taking some T_2 time. As soon as the last led is turned off, the player must turn on the leds *in the reverse order* by pressing the corresponding buttons B_1…B_4, within some time T_3. Each button B_i turns on the corresponding led L_i. So if the order in which the leds were turned off was for instance 3,1,2,4 (that is: the first led to be turned off was L_3, then L_1, etc), then the leds must be turned on again in order 4,2,1,3 (that is: first L_4, then L_2, etc). If the player restores the leds on time, the *score* - starting from zero - is increased and the game goes on, repeating the sequence, but reducing the times T_2 and T_3 of some factor F. If the player does not restore the leds on time in the correct order, the red led L_s is turned on for 1 second and the game ends, displaying the score 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 Restore the Light Game. Press Key B1 to Start”.

If/when the button B_1 is pressed the game starts. If the B_1 button is not pressed within 10 seconds, the system must go into 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…L_4 are turned on some random time T_1
- then, the leds are then turned off one by one in some random order, taking T_2 time
- the player has max T_3 time for restoring the leds in the inverse order, by pressing the buttons B_1…B_4 (each button B_i turns on the corresponding led L_i)
- If the player restores the lights on time and in the correct (inverse) order, 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 restore the lights on time in the correct order, then the red led L_s turned on for 1 second and 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