**Embedded System and IoT - a.y. 2024-2025**

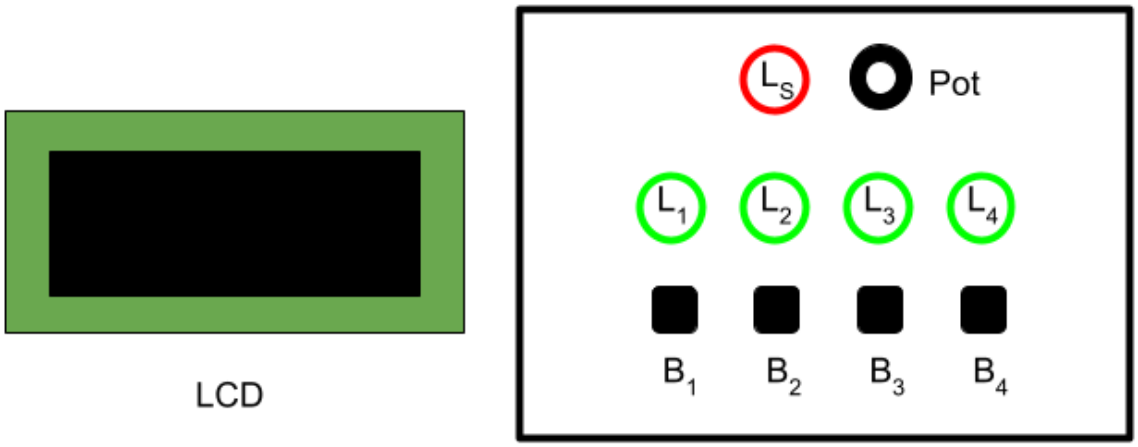
**v0.9 - 20241003**

# Assignment #1 - *Give Me the Binary! (GMB)*

We want to realise an embedded system implementing a game called *Turn on the binary code.*

## Description

The game board includes 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, and an LCD. This is the suggested layout:




In the game, leds represent the binary digits of a number (0..15). During the game, the system repeatedly displays a number in decimal notation on the LCD and the player must turn on the proper leds defining the same number in binary notation (L1 is the most significant bit, L4 the less significant bit). So for instance: if the number 13 is displayed on the LCD, the player must turn on the leds L1, L2 and L4. To turn on the leds, the player can use the tactile buttons. Each button $B_i$ turns on the corresponding led $L_i$. Each game involves multiple rounds. At each round, a number (at random) is displayed and the player must compose the binary version, within some maximum time $T_1$. If the player does it right, a *score* - starting from zero - is increased and the game goes on, with another round, but reducing the times $T_1$ of some factor F. If the player does not compose the correct number on time, the red led $L_s$ is turned on for 1 second and the game ends, displaying the score on the LCD.

### *Game behaviour in detail*

In the initial state, all green leds should be off but led $L_S$ that pulses (fading in and out), waiting for some player to start the game. The LCD should display the message “Welcome to GMB! Press B1 to Start” (on multiple lines).

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 turned off and a “Go!” message is displayed on the LCD. The score is set to zero.

During the game, at each round:

- The leds $L_1$…$L_4$ are turned off and a random number between 0 and 15 is displayed on the LCD
- Then, the player has max $T_1$ time for composing the binary version, by pressing the buttons $B_1$…$B_4$ (each button $B_i$ turns on the corresponding led $L_i$)
- If the player composes the correct number on time, then:
    - the score is increased and a message "GOOD! Score: XXX" (where XXX is the current score) is displayed on the LCD
    - the game goes on with another round, by reducing the time $T_1$ of some factor F.
- If the player does not compose the correct number on time, then the red led $L_s$ is turned on for 1 second and the game ends: a message "Game Over - Final Score XXX" (where XXX is the final score) is displayed on the LCD (on multiple lines) 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