

EE212 Spring 2024–2025 Lab Assignment 5: Ping-Pong Game

Furkan Genç

Deadline: 14.04.2025 Monday 13:30

Please read the notes and the assignment requirements carefully since they are essential for evaluation.

Assignment Requirements

- You need FRDM-KL25Z board and component kits for this lab assignment. You will show your demo on FRDM-KL25Z board.
- The deadline is strict. Submit your code before the deadline. **You could not change your uploaded codes during lab. You will show your demos based on your uploaded codes.** You will not change even one line of the code during the demo.
- You need to get a check from one of the lab assistants. The check consists of small demonstrations and questions evaluating your knowledge of the lab. Indeed, note that getting a check from all parts does not mean you get the full grade. Your grade will also be based on your answers to questions and code efficiency.
- Please upload your codes files with '.txt'. TAs may run your codes after lab hours, so please upload the correct files. Failing to send the correct file results in a deduction of your grades.
- This is an individual assignment. You can cooperate but must submit your **OWN** code. Any plagiarism will not be tolerated. After the lab, the codes will be compared manually by assistants and by Turnitin software.

Introduction

In this lab, you will implement a “ping-pong” style LED game on the FRDM-KL25Z board using five LEDs placed horizontally and three buttons for control.

The default behavior is a single “ball” of light bouncing from the first LED to the fifth LED and back again. Concretely:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow \dots$$

Here, “1” means only the first LED is lit (others off), “2” means only the second LED is lit, and so on. After reaching the fifth LED, it bounces back in the opposite direction.

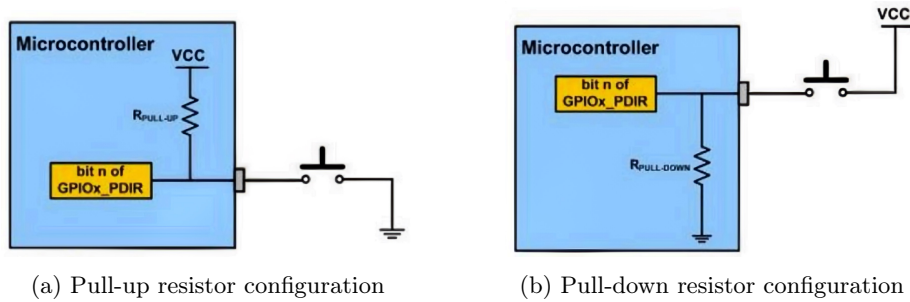


Figure 1: Possible button configurations for the FRDM board.

Equipments and Connections

- **5 LEDs**, connected to 5 distinct GPIO pins on the FRDM board (configured as outputs).
- **3 Buttons**, connected to 3 distinct GPIO pins on the FRDM board (configured as inputs).
- You may use either internal pull-up or internal pull-down resistors for your buttons. Figure 1 shows examples of each wiring configuration.

1 Single-Ball Ping-Pong (35 pts)

Implement the baseline LED ping-pong effect with a single LED (“ball”) traversing from LED 1 to LED 5 and back to LED 1 repeatedly, with a **1000 ms delay** between each move. Specifically:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow \dots$$

Notes:

- Use a delay function (no timers) for the 1000 ms pause.
- No button input is required for this base mode; just demonstrate the ping-pong pattern.

2 Speed Control (15 pts)

Use the first button to toggle the LED movement speed between **1000 ms** (normal) and **500 ms** (fast):

- Pressing this button once changes the delay to 500 ms.
- Pressing it again returns the delay to 1000 ms.
- *Important:* Do not reset the LED index or pattern when switching speeds.

3 Direction Control (15 pts)

Use the second button to reverse the current direction of LED movement:

- For example, if the sequence is going forward as $\dots 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$, and the button is pressed while LED 4 is lit, the next LED should be LED 3 (inverting the direction).
- A more detailed illustration:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \xrightarrow{\text{press}} 3 \rightarrow 2 \xrightarrow{\text{press}} 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3$$

- You must continue from the currently lit LED at the moment of pressing; do not reset to LED 1 or LED 5.

4 Two-Ball Mode (35 pts)

Use the third button to toggle a two-ball mode, so that two adjacent LEDs are lit at once. For instance:

$$(1, 2) \rightarrow (2, 3) \rightarrow (3, 4) \rightarrow (4, 5) \rightarrow (5, 4) \rightarrow (4, 3) \rightarrow (3, 2) \rightarrow (2, 1) \rightarrow \dots$$

In the state $(1, 2)$, LED 1 and LED 2 are lit while the remaining LEDs remain off. Similarly, in state $(2, 3)$, LED 2 and LED 3 are lit, and so forth. This mode creates the effect of two ping-pong balls moving together.

Notes:

- You need to implement speed control (10 pts), direction control (10 pts) as in the case of single ball ping-pong. If you only implement two ball functionality without any speed control or direction control you get 15 pts.
- You need to reset the position to $(1, 2)$ whenever two-ball mode is activated.
- Pressing the button again returns to single-ball mode (see Section 1), restarting the sequence from LED 1.

5 General Requirements

1. You must be able to change the speed and direction at run-time without restarting the program; when either is changed, continue from the currently lit LED(s) rather than resetting to the first LED.
2. You are not allowed to use timer functionality; you need to write your own delay function. You can find a similar delay function in the tutorial code.
3. Precise timing is not required, but the difference between 1000 ms and 500 ms must be clearly observed.