

ECE3080 Introduction to Embedded Systems

Project

School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen

2025-2026 Term 1

1. Objectives

In this project, we will spend the 3-week session on the following:

- To design and build up a 2-player bouncing ball game based on the knowledge learned through ECE3810 Lab 1-5.
- To propose and improve the game based on your design.

2. Basics

In this lab, you are required to include all you have learned through Lab 1-5, including

- GPIO
- USART
- LCD
- External interrupt
- Timer

We have provided _Sample.hex, Font.H (including 12×6 , 16×8 and 24×12 sized fonts), and part of EIE3810_TFTLCD.c (including functions to draw a circle, and to display 24×12 sized fonts). You can download _Sample.hex into the project board. This is a bouncing ball game for two players.

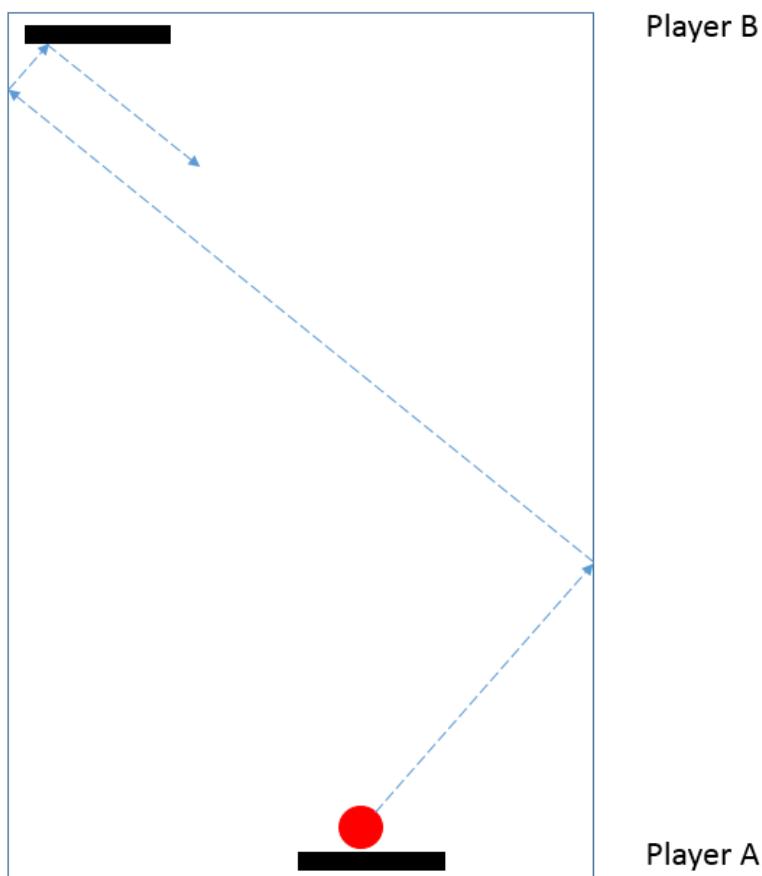


Fig. 1 General layout of the bouncing ball game

Here below is the simple manual to play it.

2.1 Connect the project board (USART1) to the computer through a USB cable. We also recommend powering the board with an external power supply. Connect JOYPAD to the board through COM3 of the project board. After that, power on the project board. Download the “_Sample.hex” provided in Blackboard into the board. Then power off the board, and power it on again by pressing the blue button at the right upper corner of the board. This will allow for a more stable system startup, compared with pressing the red RESET button.



Fig. 2 Hardware settings

2.2 There will be a brief introduction text shown on the LCD (Fig. 3 and then changed to Fig. 4). Player A press Key_Up and Key1 to select the difficulty levels (i.e., easy or hard), then press Key0 to confirm. Player B can also press UP and DOWN button in the JOYPAD to select the difficulty levels, then press SELECT to confirm. In the LCD, there will be one line of text, indicating to receive a random direction from USART (Fig. 4).

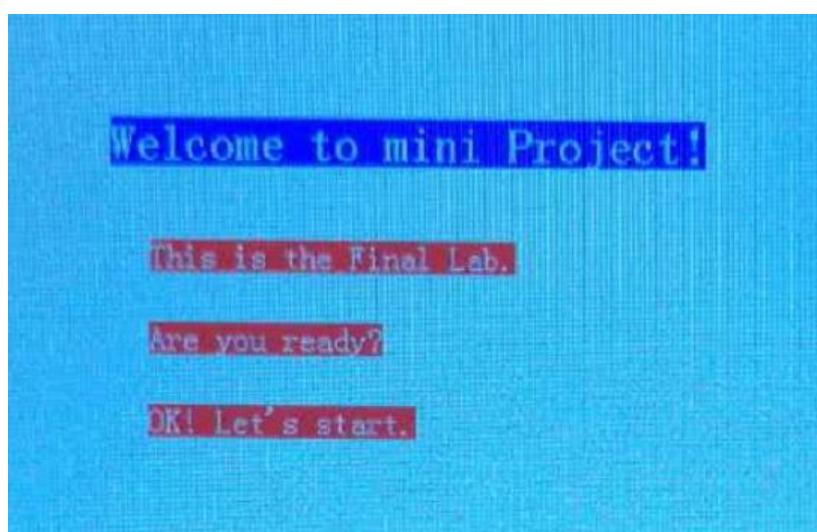


Fig. 3 First page

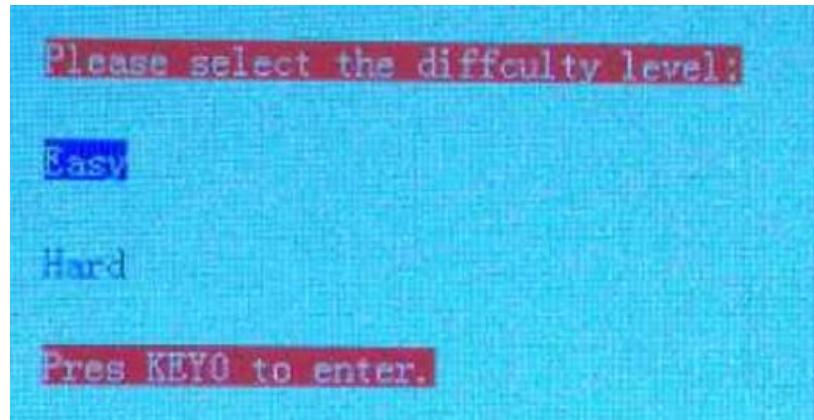


Fig. 4 Second page

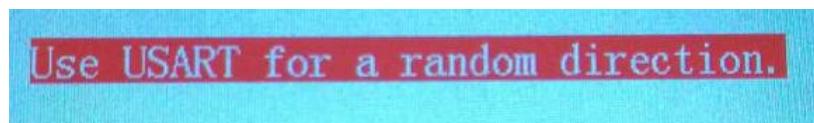


Fig. 5 Third page

2.3 Run Python code ECE3080 _PC.exe (provided in Blackboard). In the GUI (Fig. 6), input a sample student ID “114010002” (for testing _Sample.hex only), choose the correct COM port based on your computer setting, and click “Connect”.

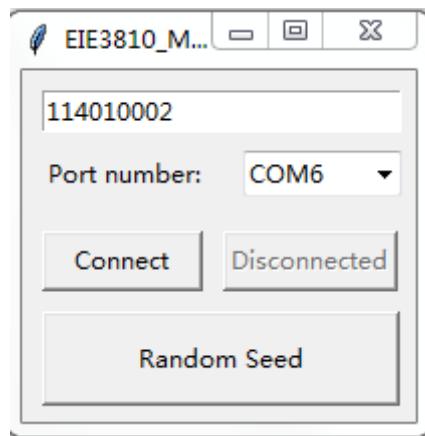


Fig. 6 Input student ID and press “Connect”

Then, the COM port will be connected to the board (Fig. 7). Press “Random Seed” to generate a random number from the set {0, 1, 2, 3, 4, 5, 6, 7} by Python, and then transmit it through USART to the microprocessor. You can use this random number to specify 8 random directions when starting the ball. The sample code “_Sample.hex” provides some references. You can always specify the random directions by yourself.

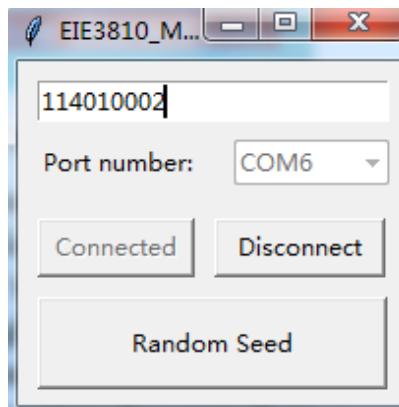


Fig. 7 The GUI connected to the board

Each frame of USART communication has 8 data bits (for the digits from 0 to 7) and 1 stop bit, with no parity bit. However, the baud rate is chosen according to the student ID input to the GUI based on some computation. You are required to use **your own student ID for your own code**. We do not give you the mapping of the baud rate to your student ID directly and would like you to get the baud rate by yourself. (Hint: use the oscilloscope to observe, and then set that to the baud rate in your coding.) When the random number is received, it will be shown on LCD.

2.4 A 3-second countdown will then appear on the LCD, and Player A will automatically kick off the ball in the specified random direction.

2.5 Player A will use Key2 and Key0 to change the position of the pad to bounce back the ball, while Player B will use LEFT and RIGHT in the JOYPAD. When there is a bounce, i.e., the ball collides onto the two vertical boundaries or one of the two player's pads, it will change directions based on some rules, and the buzzer will make a sound. A simple rule is a reflection like a light.

2.6 The elapsed time and the number of bounces will be shown on the LCD.

2.7 If either Player A or B fails to bounce back the ball, he/she loses. Some text will be shown on LCD. After several seconds, it will go back to Step 2.2.

2.8 Player A can press Key1 to pause the game, a second press will continue the game. For Player B to pause/continue the game, press START.

3. Experiment

3.1 Experiment 1: Realize the bouncing ball game

Write **your own code** to realize the game in 2.2-2.8. (Remember to use your own student ID at EIE3080_ PC.exe)

[Demonstration] When you have completed each step (2.2, 2.4, 2.5, 2.6, 2.7, and 2.8), demonstrate to the instructor or TA that your program works.

[In Report] Use an evidential way to show in your report that you have completed this experiment.

[Source code] Provide the source code with adequate comments.

3.2 Experiment 2: Improve the bouncing ball game

Use your creativity to make the game **more challenging**. Try your best to complete Experiment 2 with a 1-minute video explaining (verbally) and demonstrating your improvement. The video should be uploaded to Blackboard before the end of the last lab session.

[Presentation] We will not have questions in this lab report. The 20 points in question (of the report) will be substituted by Experiment 2 based on the criteria below.

- 5 points will be obtained for those who have improved the game and uploaded the video before the end of the lab session.
- An extra 5-15 points are given to the selected students whose improvement and video presentation are outstanding.

[In Report] Elaborate your proposed improvement, how you realize it, and then present the result that you have achieved.

[Source code] Provide the source code with adequate comments.

4. Lab Report and Source Code

Submit the report softcopy and your code (complete project folder of each experiment) in zip format to Blackboard by the deadline below:

- L01: 18:00, Monday, December 15, 2025
- L02: 18:00, Wednesday, December 17, 2025

Each day of late submission will result in 10% deduction in the report and source code raw marks.