EIE3810 Microprocessor System Design Laboratory

Lab 6. Mini Project

School of Science and Engineering

The Chinese University of Hong Kong, Shenzhen

Spring term, 2017-2018

1. Objectives

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

2. Basics

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

- Key
- Buzzer
- USART
- LCD
- External interrupt
- Timer
- Joypad

We have provided _SampleLab6.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 the .hex into the project board. This is a bouncing ball game for two players.

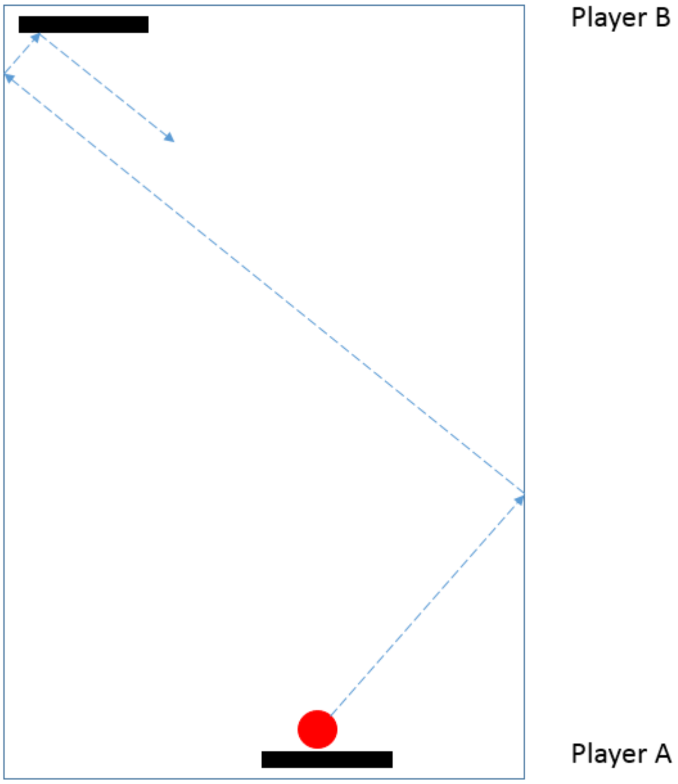
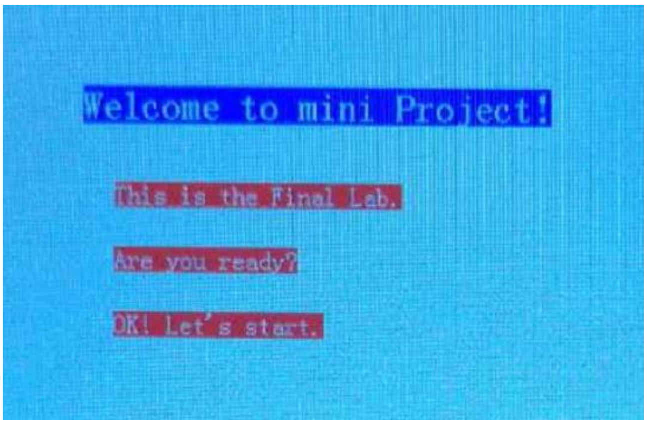


Fig. 1 General layout of the bouncing ball game

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 to power the board by external power supply. Connect joypad to the board through COM3 of the development board. Thereafter, power on the project board. Download the "_SampleLab6.hex" provided in Moodle 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 more stable startup of the system, compared with pressing the red RESET button.

2.2 There will be brief introduction text shown onto the LCD (Fig. 2 and then changed to Fig. 3). Press Key_up and Key1 select the levels of difficulty (i.e. easy or hard), and then press Key0 to choose. In the LCD, there will be one line of text, indicating to receive a random direction from USART (Fig. 4).



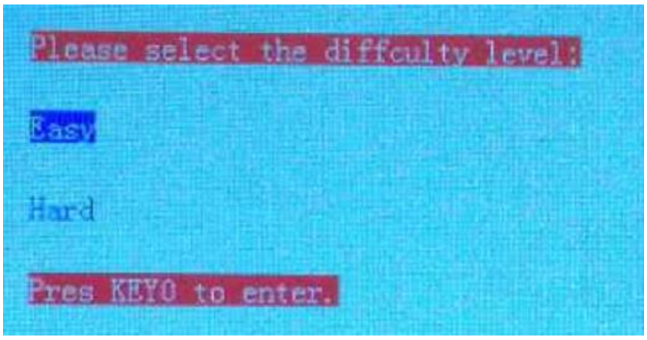
Welcome to mini Project!

This is the Final Lab.

Are you ready?

OK! Let's start.

Fig. 2 First page



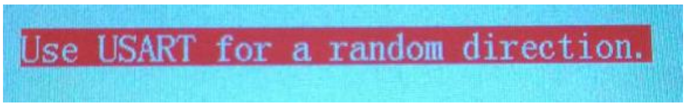
Please select the diffculty level:

Easy

Hard

Pres KEY0 to enter.

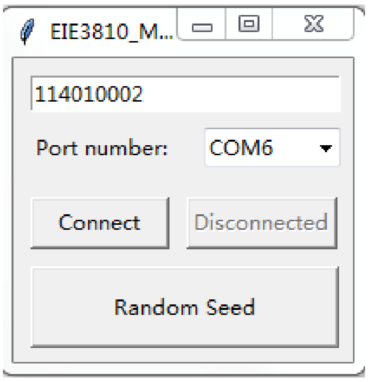
Fig. 3 Second page



Use USART for a random direction.

Fig. 4   Third page

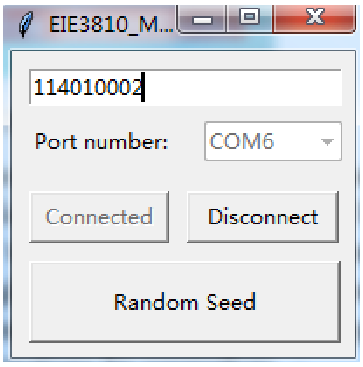
2.3 Run Python code EIE3810_Lab6_PC.exe. In the GUI (Fig. 5), input a sample student ID “114010002”, choose the correct COM port based on your computer setting, and click “Connect”. Then the COM port will be connected to the board (Fig. 6). Press “Random Seed” to generate a random number from the set {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 provides some reference. You can always specify the random directions by yourselves.



114010002

Port number: COM6

Fig. 5 Input student ID and press “Connect”



114010002

Port number: COM6

Fig. 6 The GUI connected to the board

For each frame of USART communication, there are 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 later, when you write your own code. We do not give you the mapping of 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.

2.4 When the random number is received, it will be shown on LCD. Then a 3 second count down will 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. 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 reflection like light.

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

2.7 If either Play A or B fails to bounce back the ball, he/she loses. Some text will be shown on LCD.

2.8 If Player A would like to pause the game, he/she can press Key1. A second press will continue the game. For Play B to pause/continue the game, press Start.

3. Experiment

3.1 Experiment 1: Realize bouncing ball game

Write your own code to realize the game in 2.1-2.8.

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

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

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

3.2 Experiment 2: Improve the bouncing ball game

There is no special requirement in this step, so just use your imagination to propose and realize a slightly upgraded version.

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

Try your best to complete Experiment 3.2 with a short video for demonstration, early in the 3rd lab session. The video should be copied to the teacher's computer in the lab **before the 16:00, May 10**. Then, for those who have already completed this part, present shortly (within 1 minute) to the classmates during the last half hour. Each student audience have maximally 10 votes. We will provide a form, which you will fill and vote for the 10 most excellent versions. We will rank based on the vote numbers, and then linearly map the vote numbers to the score. E.g. maximal vote number => 100% x 20 = 20 points, minimal vote number => 60% x 20 = 12 points. Then the points are linearly mapped based on the vote numbers, between the minimal and maximal vote numbers. Those who failed to complete the video before the aforementioned deadline, he/she will receive no points in this part.

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

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

4. Lab Report and Source Code

Submit the report hardcopy to the lab report mailbox in 6F, Chengdao Building.

Create an empty folder named by your student ID. Copy **ALL** the project files of Experiment 1-2 into this folder, and then compress folder into .zip or .rar. Upload the zipped file to Moodle.

The deadline is:

- 13:00, Thursday, May 17, 2018

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