EIE3810 Microprocessor System Design Laboratory

Lab Report 6
Mini Project

Name: Ding Xiao

Student ID: 116010034

Date: 2018/5/13

The Chinese University of Hong Kong, Shenzhen

Content

Experiment 1: Realize the bouncing ball game

1.1 Design

In this experiment, a 2-player bouncing ball game based on the knowledge learned through Lab 1-5 was designed and built up. To complete the task, following steps have been taken:

1) (Written in main function in main.c) Besides “stm32f10x.h”, the previous files “EIE3810_TFTLCD.h”, “EIE3810_key.h”, “EIE3810_Buzzer.h”, “EIE3810_LED.h”, “EIE3810_USART.h”, “EIE3810_JOYPAD.h”, and “EIE3810_TFTLCD.h” were included in the main.c file, and "Font.H" which including 12*6, 16*8 and 24*12 sized fonts was included in the EIE3810_TFTLCD.c file.

2) (Written in main function in main.c) The clock tree, TFTLCD, Key LED, Buzzer, USART1 were initialized. And priority group was set to 5. Then the TFTLCD was cleared to white.

3) (Subroutine “Step1” in main.c) In the welcome interface, a brief introduction text was shown onto the LCD. “Welcome to mini project!” was written in 24*12 sized fonts, and “This id Final Lab”, “Are you ready?”, and “OK! Let’s start” were written in 16*8 sized fonts. The time interval between every two sentence was set around 1 second.

4) **(Subroutine “Step2” in main.c)** Then in the level selecting interface, there were two modes (easy and hard) could be selected. Keys Key1 and Key_Up were used to change the selected mode, and Key0 was used to confirm the mode and go to next interface. In this step, a while (1) loop was used. Inside the loop two triggers of Key1 and Key_Up were used to change the level value between 0 and 1. And Key0 was used to break the loop and to continue other subroutines.

5) **(Subroutine “Step3” in main.c)** Then after the level was chosen, it would go to the USART interface, and the sentence “use USART for a random direction” was shown on the TFTLCD. In this step, USART1 is used for communication. After my student ID “116010034” was sent by program EIE3810_Lab6_PC.exe, the Tx pin was be measured by oscilloscope, and the time for transmitting one bit was measured 25.6μs. So the bit rate of USART1 was set to 1/[25.6*10^ (-6)], which is 39062 Hz. Then bit 5 of USART1_SR was set and a random number from 0 to 7 would be transmitted and stored in USART_DR register. The random number was read for deciding the beginning direction of the ball. After the number was sent, the word

“SUCCESS” would be shown on the TFTLCD, and the program continue to execute next subroutines.

6) **(Subroutine “Step4” in main.c)** Then was the count down interface. The time count down from 3 to 0 in 7-segment form on TFTLCD. And then the game will start. The JOYPAD, TIM2, TIM4, and TIM5 would be initialized for supporting following game subroutines..

7) **(Subroutine “Game2”, “GameBall” in main.c)**

In the game, “**Game2**” subroutine was used to control the two pads. The subroutine was controlled by TIM4. First, FillRectangle function was used to draw the two rectangular pads on the TFTLCD. Then at every count of the timer, both of the pads can be moved horizontally by two triggers, and the previous ball was covered by white. The upper pad (for player B) can be moved by triggers of LEFT and RIGHT keys on the JOYPAD, and the lower pad (for player A) can be moved by triggers of Key2 and Key0.

“**GameBall**” subroutine was used to control the trajectory of the ball and judge whether the game is over and which player won. The subroutine was controlled by TIM5. First, DrawCircle function was used to draw the solid ball on the TFTLCD, and the ball starts with the initial direction decided by the random number in previous step. Then at every count of the timer, the ball move once for several pixels horizontally and vertically, respectively, and the previous ball was covered by white. As long as the ball hit the boundary, it would change the direction as the light reflection. When the ball arrived the up edge or down edge, the program would check the ball’s and pads’ position. If the ball was in the range of one of the pad, the ball would reflect as light reflection. If not, the game would be over.

9) **(Subroutine “GameOver” in main.c)** If the ball was not in the pad’s range when arriving up or down edge, the game would be over. The game would be paused and then all of the TIMs would be disabled. Then the TFTLCD screen would be cover by black and red sentence “GAME OVER” would be shown on the screen. And the loser of the game would be shown on the screen too, which was judged by which edge the ball arrived.

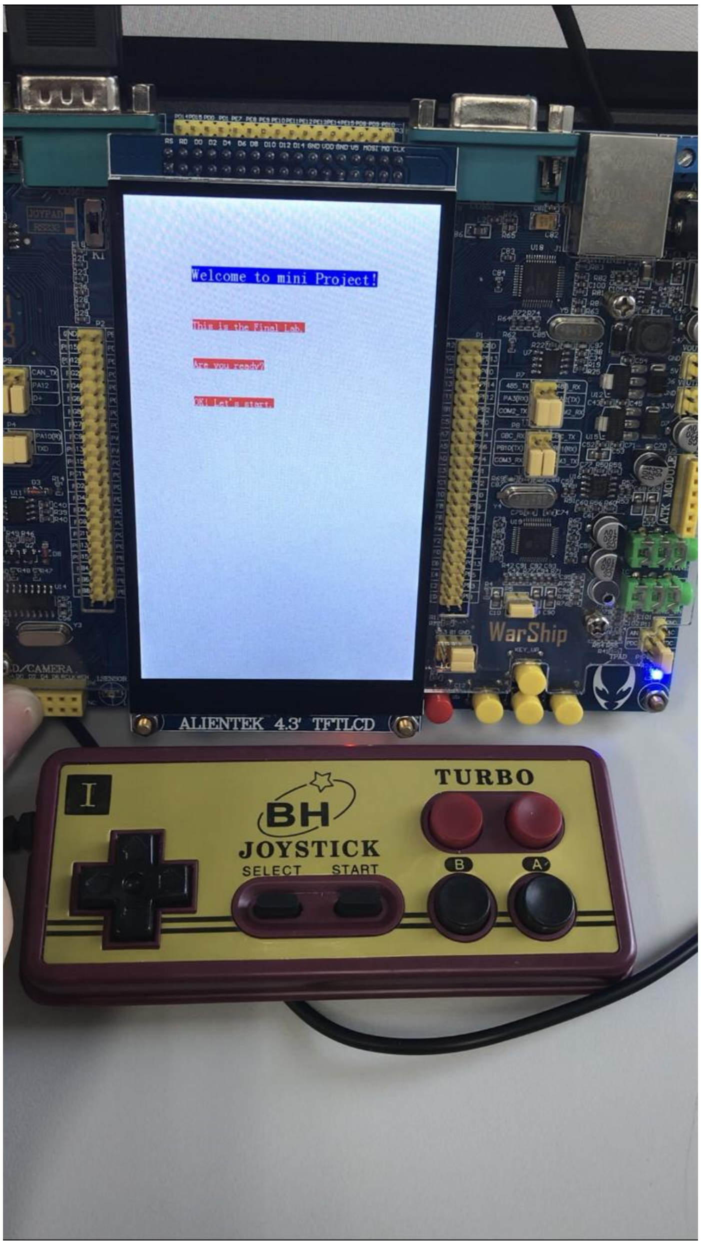
10) (**Subroutine “show_time_bounces” in main.c**) When the game began, a timer would start to count the time spent in game in second. The timer was controlled by TIM5, and the PSR and ARR of it were set to 9999 and 7199 to make that the timer count every second. When the ball hits the down panel, the number of “bounces” would plus one.

11) (**Subroutine “pauses” in main.c**) When the game was continuing, it can be paused by pressing START on JOYPAD or the Key1. The two keys implement this

function by toggling the value of “pause”. If pause = 0, the game would continue, and if pause = 1, the game and the timer would pause.

1.2 Result

When the program was executing, the TFTLCD, keys and JOYPADs are like following.



Welcome to mini Project!

This is the Final Lab.

Are you ready?

OK! Let's start.

Figure1 subroutine **Step1**

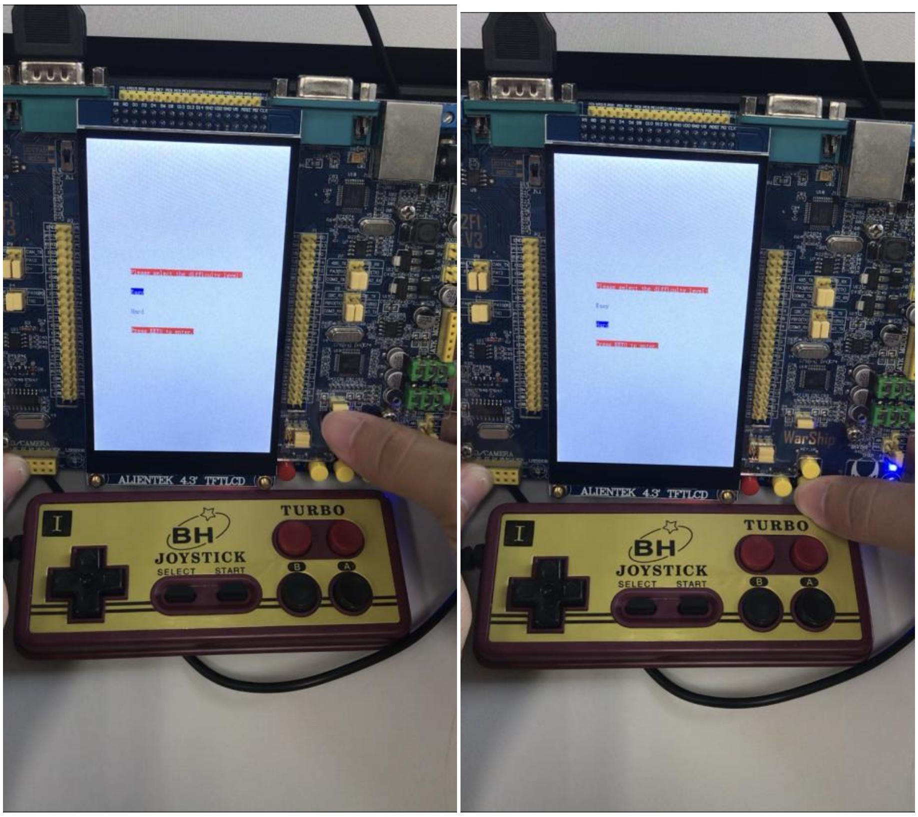


Figure2-3 subroutine **Step2**

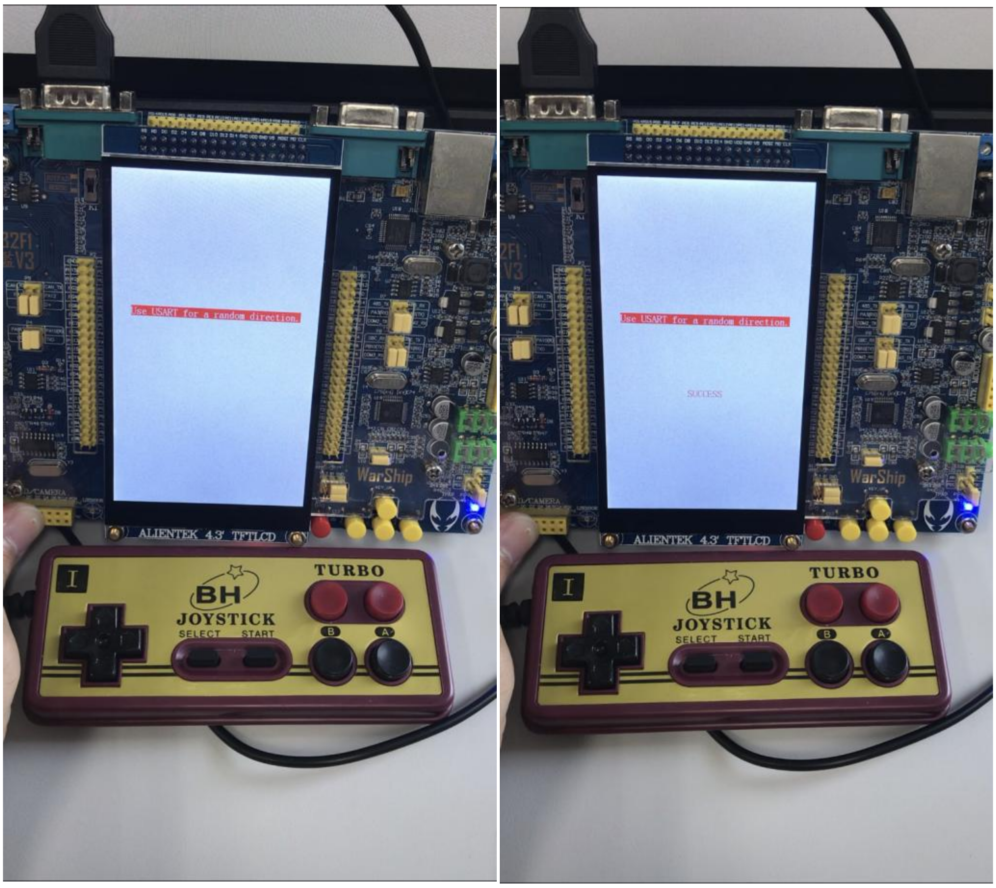


Figure4-5 subroutine **Step3**

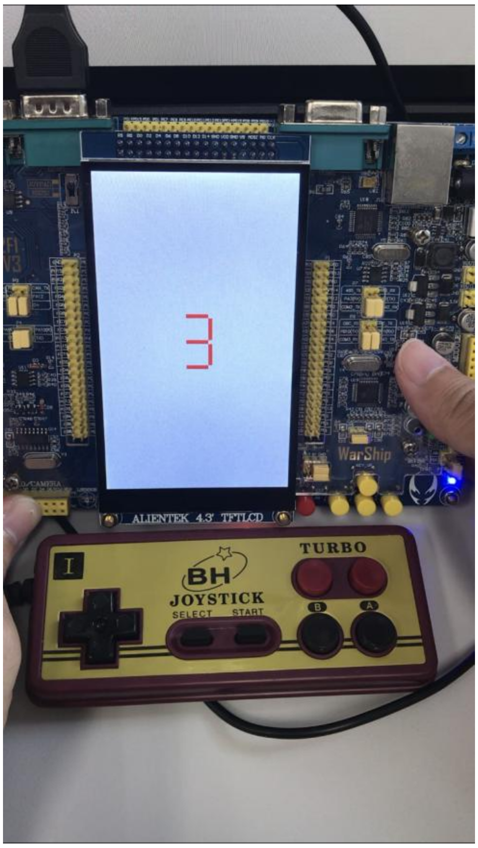


Figure6 subroutine **Step4**

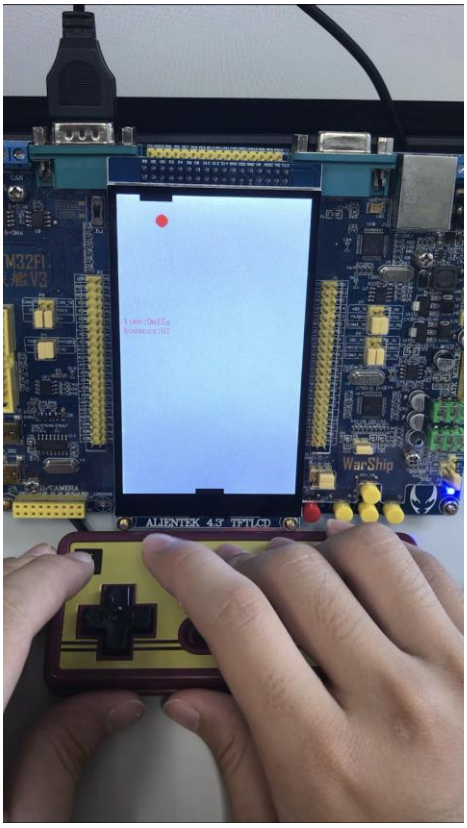
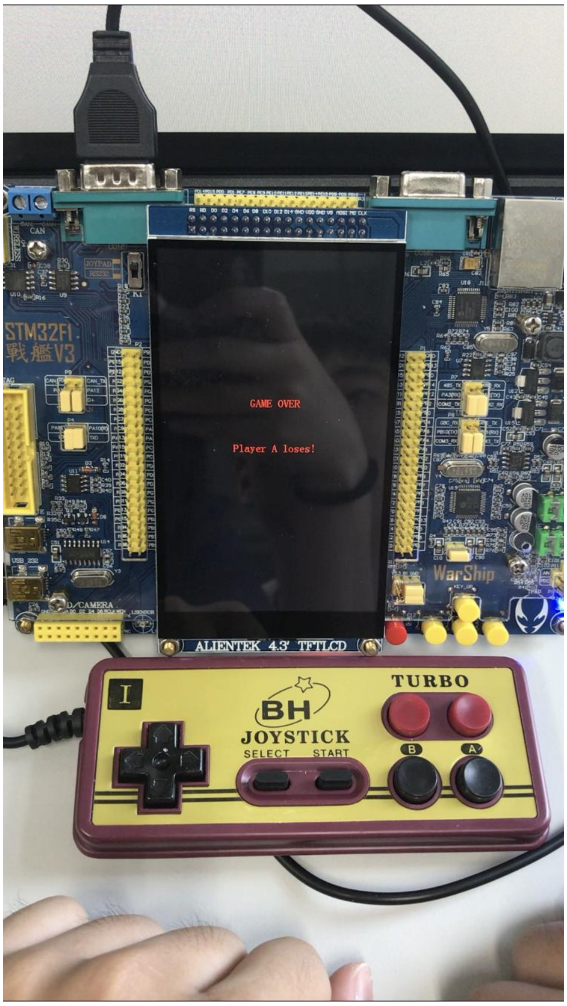


Figure7 subroutine **Game2** and **GameBall**



GAME OVER

Player A loses!

Figure7 subroutine **GameOver**

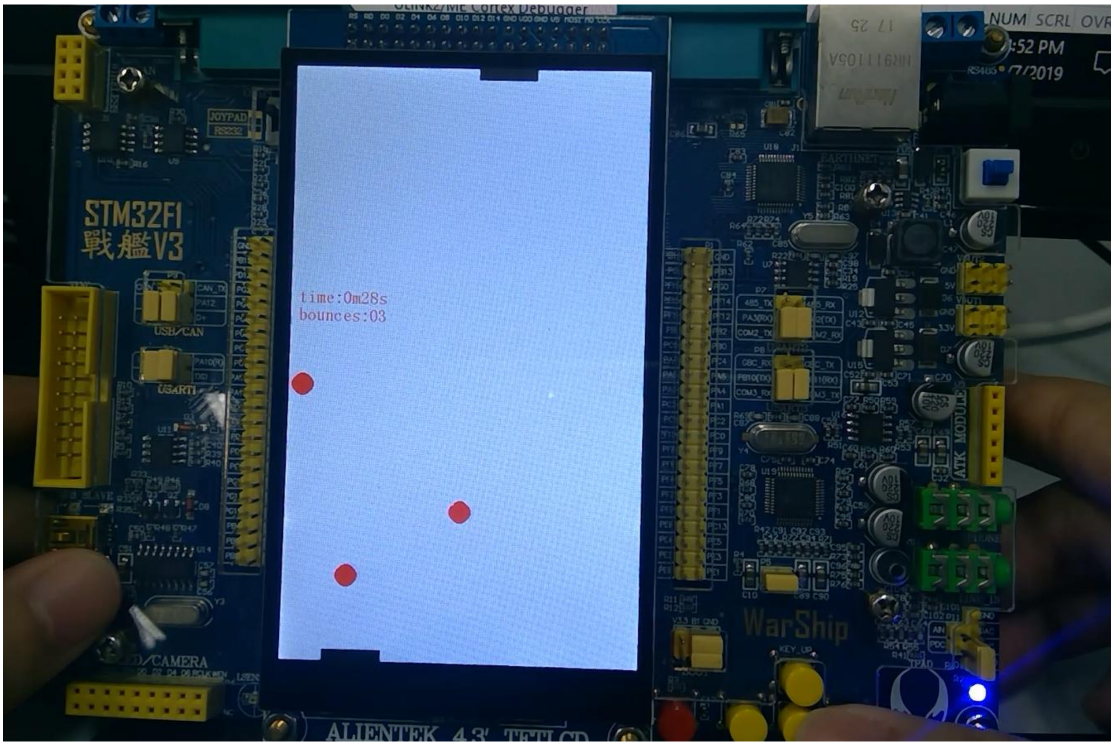
Experiment 2: Improve the bouncing ball game

2.1 Design

My improvement is to add two balls by adding two subroutines “**GameBall2”** and **“GameBall3”** into **main.c** which are similar with “**GameBall”**. The directions of 3 balls have different corresponding relations with the random number gotten in subroutine **“Step3”**, in order to make the 3 balls always move in different directions. All of **“GameBall, GameBall2, GameBall3”** are controlled by TIM2 IRQHandler. When one of the ball arriving the edge and doesn’t meet the pad, the game will be over..

2.2 Result

The game with 3 balls is following.



```
time:0m28s
bounces:03
```

Figure8 improved ball game with 3 balls

Conclusion

In Lab 6, a small project about the 2-players' ball game was finished by applying all we have learned before. During this experiment, I have learned:

(1) How to design the structure of a program by myself.

(2) There are actually concerns in developing a mature program that we should consider about. We should plan for the program at the beginning.