

微處理機系統實習 Lab7

班級：資訊三甲 學號：D1123911 姓名：陳政宏

一、【實驗目的】：

What was your design? What were the concepts you have used for your design?

本次利用 GPIO 的 KEYPAD 和 Buzzer 蜂鳴器和 LCD 和 draw2d，來完成兩項實驗。

實驗一：一開始在 LCD 中間處 $x=64, y=32$ ，畫一顆半徑=3 的圓

按左右鍵(4 跟 6 按)，圓會水平往左/右方向移動，按 3 鍵，圓以 45 度往右上移動，按 9 鍵，圓以 45 度往右下移動，按 1 鍵，圓以 45 度往左上移動，按 7 鍵，圓以 45 度往左下移動，當圓觸到邊界時會反射角=入射角的方式反彈，並且蜂鳴器響一聲。按 S (Stop) 鍵，畫面停止移動如要變換方向，需先按 S 鍵後，再按新的方向鍵(和 lab6-1 判斷方式相同)。

實驗二：在 LCD 中間底部 $x=64, y=60$ 處，畫一顆半徑=3 的圓，並且在其上半部隨機產生一個 5x5 的黑色方塊圓的移動及按鍵使用與實驗一相同，當圓碰觸到 5x5 黑色方塊時 (overlap 重疊)，黑色方塊消失並且圓回到起始的底部位置按 R (Random) 鍵，重新在 LCD 畫面上半部隨機產生一個新的 5x5 黑色方塊

本次實驗主要應用前面實驗的技巧產生亂數、FLAG 的判斷、和 draw2d 的應用。

二、【遭遇的問題】：

實驗一：若在邊界考慮時沒有多考慮一格，邊界會把圓吃掉一個所以在判斷是否超出邊界時，要多+1。

實驗二：要考慮隨機產生一個 5*5 黑色方塊，代表在 8 鬆開時才能產生黑色方塊，若沒有正確判斷會讓畫面上充滿黑色方塊，並且不會消除上一個 draw 出來的黑色方塊。

三、【解決方法】：

How did you solve the problems?

概述：首先也一樣利用 check 來判斷使用者是否有正確按下 s 按鍵才繼續執行以下動作，按下除了 5 以外紀錄 last_press 並將值給到 z 再由 z 判斷現在該往

哪個方向走，當碰到邊界時，改變方向(x|y)，同時把 z 歸 0，讓其一直執行反方向，直到下次按下 5，否則他會只執行一次反方向又往 z 一開始判斷的方向移動，會讓球在邊界卡住。

實驗一程式碼:

```
1 //
2 // LCD_Graph_Pingpong: draw a circle bouncing around between two bars on LCD
3 //
4 // EVB : Nu-LB-NUC140
5 // MCU : NUC140VE3CN
6 //
7 #include <stdio.h>
8 #include <math.h>
9 #include "NUC100Series.h"
10 #include "MCU_init.h"
11 #include "SYS_init.h"
12 #include "LCD.h"
13 #include "Draw2D.h"
14 #include "Scankey.h"
15
16 #define PIXEL_ON 1
17 #define PIXEL_OFF 0
18 #define X0 64 // Circle initial X
19 #define Y0 32 // Circle initial Y
20 #define R_BAR_X LCD_Xmax-1 // Right-Bar X-position
21 #define L_BAR_X 0 // Left-Bar X-position
22
23 int32_t main (void)
24 {
25     int i,last_press=0,check=1,z=0;
26     int dirX, dirY;
27     int movX, movY;
28     uint16_t r;
29     uint16_t x, y;
30     uint16_t fgColor, bgColor;
31     SYS_Init();
32     init_LCD();
33     clear_LCD();
34
35     x = X0; // circle center x
36     y = Y0; // circle center y
37     r = 3; // circle radius
38     movX = 1; // x movement
39     movY = 1; // y movement
40     dirX = 0; // x direction
41     dirY = 0; // y direction
```

```

42
43     bgColor = BG_COLOR;
44     while(1) {
45         fgColor = FG_COLOR;
46         draw_Circle(x, y, r, fgColor, bgColor); // draw a circle
47         CLK_SysTickDelay(50000); // adjustable delay for vision
48         fgColor = BG_COLOR;
49         draw_Circle(x, y, r, fgColor, bgColor); // erase a circle
50         i=ScanKey();
51         if(i==1||i==3||i==4||i==5||i==6||i==7||i==9)
52         {
53             last_press=i;
54             if(i==5)check=1;
55         }
56         if(check==1)
57         {
58             if(last_press==1){z=1;check=0;}
59             if(last_press==3){z=3;check=0;}
60             if(last_press==4){z=4;check=0;}
61             if(last_press==5){z=5;}
62             if(last_press==6){z=6;check=0;}
63             if(last_press==7){z=7;check=0;}
64             if(last_press==9){z=9;check=0;}
65         }
66         if(z==1){dirX = -1;dirY = -1;}
67         else if(z==3){dirX = 1;dirY = -1;}
68         else if(z==4){dirX = -1;dirY = 0;}
69         else if(z==5){dirX = 0;dirY = 0;}
70         else if(z==6){dirX = 1;dirY = 0;}
71         else if(z==7){dirX = -1;dirY = 1;}
72         else if(z==9){dirX = 1;dirY = 1;}
73         if(x + 3 + 1> R_BAR_X || x - 3 - 1< L_BAR_X)
74         {
75             dirX = -dirX;
76             x += dirX * movX;
77             y += dirY * movY;
78             PB11 = 0;
79             CLK_SysTickDelay(5000);
80             PB11 = 1;
81             z=0;
82         }

```

```

82     }
83     if(y + 3 + 1> LCD_Ymax || y - 3 - 1< 0)
84     {
85         dirY = -dirY;
86         x += dirX * movX;
87         y += dirY * movY;
88         PB11 = 0;
89         CLK_SysTickDelay(5000);
90         PB11 = 1;
91         z=0;
92     }
93     x = x + dirX * movX;
94     y = y + dirY * movY;
95
96
97 }
98
99 }

```

概述(實驗二): 延伸實驗一的程式，加入隨機產生 5*5 黑色方塊，當球的 x|y 碰到，黑色方塊的 xx|yy 就重製畫面，球回到最下方，黑色方塊消失。所以只要增加判斷黑色方塊的產生以及是否與小球產生重疊，以及確保鬆開時才能顯示隨機產生黑色方塊的樣子的狀態，就能夠完成該實驗。

實驗二程式碼:

```
1 //
2 // LCD_Graph_Pingpong: draw a circle bouncing around between two bars on LCD
3 //
4 // EVB : Nu-LB-NUC140
5 // MCU : NUC140VE3CN
6 //
7 #include <stdio.h>
8 #include <math.h>
9 #include "NUC100Series.h"
10 #include "MCU_init.h"
11 #include "SYS_init.h"
12 #include "LCD.h"
13 #include "Draw2D.h"
14 #include "Scankey.h"
15
16 #define PIXEL_ON 1
17 #define PIXEL_OFF 0
18 #define X0 64 // Circle initial X
19 #define Y0 32 // Circle initial Y
20 #define R_BAR_X LCD_Xmax-1 // Right-Bar X-position
21 #define L_BAR_X 0 // Left-Bar X-position
22
23 int32_t main (void)
24 {
25     int i, last_press=0, check=1, z=0, count=0, rstart=1;
26     int dirX, dirY;
27     int movX, movY;
28     uint16_t r;
29     uint16_t x, y;
30     uint16_t fgColor, bgColor;
31     SYS_Init();
32     init_LCD();
33     clear_LCD();
34     int xx = 100, yy = 20, state=0;
35     x = X0; // circle center x
36     y = Y0; // circle center y
37     r = 3; // circle radius
38     movX = 1; // x movement
39     movY = 1; // y movement
40     dirX = 0; // x direction
41     dirY = 0; // y direction
```

```

42     int flag=0;
43
44     bgColor = BG_COLOR;
45     while(1) {
46         count++;
47         srand(count);
48         fgColor = FG_COLOR;
49         if(rstart){draw_Rectangle(xx, yy, xx+5, yy+5, fgColor, fgColor);}
50         draw_Circle(x, y, r, fgColor, bgColor); // draw a circle
51         CLK_SysTickDelay(50000); // adjustable delay for vision
52         fgColor = BG_COLOR;
53         draw_Circle(x, y, r, fgColor, bgColor); // erase a circle
54         i=ScanKey();
55         if(i!=2&&state==1&&i!=0)
56         {
57             last_press=i;
58             if(last_press==5)check=1;
59             if(last_press==8)flag=1;
60         }
61         if(i == 0)
62         {
63             state = 1;
64         }
65
66         if(last_press == 8 && state == 1 && flag == 1)
67         {
68             draw_Rectangle(xx, yy, xx+5, yy+5, fgColor, bgColor);
69             xx = rand() % 123;
70             yy = rand() % ((LCD_Ymax/2)+(2*r));
71             flag = 0;
72             state = 0;
73             rstart=1;
74         }
75         if(check==1)
76         {
77             if(last_press==1){z=1;check=0;}
78             if(last_press==3){z=3;check=0;}
79             if(last_press==4){z=4;check=0;}
80             if(last_press==5){z=5;}
81             if(last_press==6){z=6;check=0;}
82             if(last_press==7){z=7;check=0;}

```

```

83         if(last_press==9){z=9;check=0;}
84     }
85     if(z==1){dirX = -1;dirY = -1;}
86     else if(z==3){dirX = 1;dirY = -1;}
87     else if(z==4){dirX = -1;dirY = 0;}
88     else if(z==5){dirX = 0;dirY = 0;}
89     else if(z==6){dirX = 1;dirY = 0;}
90     else if(z==7){dirX = -1;dirY = 1;}
91     else if(z==9){dirX = 1;dirY = 1;}
92     x = x + dirX * movX;
93     y = y + dirY * movY;
94     if(x + 3 + 1> R_BAR_X || x - 3 - 1< L_BAR_X)
95     {
96         dirX = -dirX;
97         x += dirX * movX;
98         y += dirY * movY;
99         PB11 = 0;
100        CLK_SysTickDelay(5000);
101        PB11 = 1;
102        z=0;
103    }
104    if(y + 3 + 1> LCD_Ymax || y - 3 - 1< 0)
105    {
106        dirY = -dirY;
107        x += dirX * movX;
108        y += dirY * movY;
109        PB11 = 0;
110        CLK_SysTickDelay(5000);
111        PB11 = 1;
112        z=0;
113    }
114    if(x+3>=xx&&x-3<=xx+5&&y+3>=yy&&y-3<=yy+5)
115    {
116        x = X0;
117        y = LCD_Ymax-r-1;
118        dirX = 0;
119        dirY = 0;
120        draw_Rectangle(xx, yy, xx+5, yy+5, fgColor, bgColor);
121        z=0;
122        rstart=0;
123        state=0;

```

```

122        rstart=0;
123        state=0;
124    }
125
126
127
128
129
130 }

```

四、【未能解決的問題】：

Was there any problem that you were unable to solve? Why was it unsolvable?

本次實驗雖沒有未解決之問題，實驗一的邊界+1 的判斷以及實驗二要鬆開時才能顯示隨機產生的 5*5 黑色方塊這兩個問題，需要經過一段時間的 debug 才能完成實驗。