微處理機實習

Lab8

班級：資訊三甲

學號：D1109023

姓名：楊孟憲

一、 實驗目的

此次實驗利用程式邏輯的編寫操作 GPIO 上的 LCD 畫面捲動和 Keyboard + GPIO 中斷操作。

二、 遭遇的問題

沒有未能解決的問題。

三、 解決方法

- 實驗一 畫面捲動

```
#include <stdio.h>
#include "NUC100Series.h"
#include "MCU_init.h"
#include "SYS_init.h"
#include "LCD.h"
#include "Scankey.h"


int flag = 2;


void GPAB_IRQHandler(void)
{
    if (PA->ISRC & BIT0 && flag == 2) {         //
        check if PB12 interrupt occurred
```

```
15          PA->ISRC |= BIT0;          // clear PB12
                interrupt status
16              flag = 3;              // set a
                    flag for PB12(KEY1)
17      } else if (PA->ISRC & BIT1) { // check if Pb13
            interrupt occurred
18          PA->ISRC |= BIT1;          // clear PB13
                interrupt status
19          flag = 2;                  // set a flag
                for PB13(KEY2)
20      } else if (PA->ISRC & BIT2 && flag == 2) { //
            check if PB14 interrupt occurred
21          PA->ISRC |= BIT2;          // clear PB14
                interrupt status
22          flag = 1;                  // set a flag
                for PB14(KEY3)
23      } else {                       // else it is
            unexpected interrupts
24          PA->ISRC = PA->ISRC;           // clear all
                GPB pins
25                  PA0 = PA1 = PA2 = 1;
26          }
27  }
28
29  void Init_KEY(void)
30  {
31      GPIO_SetMode(PA, (BIT3 | BIT4 | BIT5),
            GPIO_MODE_OUTPUT);
32          GPIO_SetMode(PA, (BIT0 | BIT1 | BIT2),
```

```
            GPIO_MODE_QUASI);
    GPIO_EnableInt(PA, 0, GPIO_INT_FALLING);
    GPIO_EnableInt(PA, 1, GPIO_INT_FALLING);
    GPIO_EnableInt(PA, 2, GPIO_INT_FALLING);

    NVIC_EnableIRQ(GPAB_IRQn);
    GPIO_SET_DEBOUNCE_TIME(GPIO_DBCLKSRC_LIRC,
       GPIO_DBCLKSEL_64);
    GPIO_ENABLE_DEBOUNCE(PA, (BIT0 | BIT1 | BIT2))
       ;
      PA3 = PA4 = PA5 = 0;
}


unsigned char map[128*8] = { // Nuvoton Logo
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
       x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
       ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
       x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
       ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
       x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
       ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
       x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
       ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
       x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
       ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
       x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
       ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
       x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
```

```
    ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
45  0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xC0,0xE0,0
    xE0,0xE0,0x60,0x70,0x70,0x60,0x60,0xE0,0xE0
    ,0xE0,0xC0,0x80,0x00,0x00,0xE0,0xE0,0xE0,0
    xE0,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0xE0
    ,0xE0,0xE0,0xE0,0xE0,0x00,0x00,0x60,0xE0,0
    xE0,0xE0,0x80,0x00,0x00,0x00,0x00,0x80,0xE0
    ,0xE0,0xE0,0x60,0x00,0x00,0x80,0xC0,0xE0,0
    xE0,0xE0,0x60,0x60,0x70,0x60,0xE0,0xE0,0xE0
    ,0xC0,0xC0,0x80,0x10,0x18,0x18,0x18,0x18,0
    x18,0xF8,0xF8,0xF8,0xF8,0x18,0x18,0x18,0x18
    ,0x18,0x18,0x00,0x80,0xC0,0xE0,0xE0,0xE0,0
    x60,0x60,0x70,0x60,0xE0,0xE0,0xE0,0xC0,0xC0
    ,0x80,0x00,0x00,0x00,0x80,0xC0,0xE0,0xE0,0
    xE0,0x60,0x70,0x70,0x70,0x60,0xE0,0xE0,0xE0
    ,0xC0,0x80,0x00,0x00,0x00,0x00,0x00,0x00,
46  0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0x7F,0x7F,0
    x7F,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x7F
    ,0x7F,0x7F,0x7F,0x00,0x00,0x0F,0x3F,0x3F,0
    x7F,0x7F,0x70,0xE0,0xE0,0xE0,0xE0,0x70,0x7F
    ,0x7F,0x3F,0x3F,0x0F,0x00,0x00,0x00,0x01,0
    x03,0x0F,0x1F,0x7E,0x78,0x78,0x7E,0x1F,0x0F
    ,0x03,0x00,0x00,0x00,0x0F,0x1F,0x3F,0x7F,0
    x79,0x70,0xE0,0xE0,0xE0,0xE0,0x60,0x70,0x7F
    ,0x7F,0x3F,0x1F,0x00,0x00,0x00,0x00,0x00,0
    x00,0x7F,0x7F,0x7F,0x7F,0x00,0x00,0x00,0x00
    ,0x00,0x00,0x0F,0x1F,0x3F,0x7F,0x79,0x70,0
    xE0,0xE0,0xE0,0xE0,0x60,0x70,0x7F,0x7F,0x3F
    ,0x1F,0x00,0x00,0x00,0x7F,0x7F,0x7F,0x7F,0
```

```
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0x7F
,0x7F,0x7F,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
xC0,0xE0,0x20,0x20,0x20,0x60,0xE0,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x20,0xE0,0xE0,0x00,0x00,0xE0,0
xE0,0xE0,0x00,0x00,0x80,0xE0,0x60,0x20,0x60
,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x20,0xE0,0xE0,0x00,0x00
```

```
    ,0xE0,0xE0,0xE0,0x00,0x00,0xC0,0xE0,0x20,0
    x20,0x20,0x60,0xE0,0x00,0x00,0x20,0xE0,0xE0
    ,0x00,0x00,0xE0,0xE0,0x00,0x00,0x00,0x00,
49  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0C,0
    x7F,0xFF,0x80,0x00,0x00,0x80,0xE0,0x20,0x10
    ,0xFC,0xCE,0x02,0x02,0x02,0xC6,0xFC,0x10,0
    x00,0x02,0xFE,0xFE,0x06,0x02,0x02,0x02,0x00
    ,0x00,0x02,0x02,0xFF,0xFF,0x02,0x82,0x82,0
    x00,0x00,0x7C,0xFE,0x92,0x12,0x12,0x9E,0xCC
    ,0x00,0x00,0x02,0xC6,0x6E,0x38,0xFE,0xC6,0
    x02,0x00,0x00,0x08,0x08,0x08,0x08,0x08,0x08
    ,0x08,0x00,0x00,0xFF,0x0F,0xFE,0x3C,0x0F,0
    xFF,0xFF,0x00,0x00,0x3F,0xFF,0x80,0x00,0xC0
    ,0xFF,0x3F,0x00,0x00,0x00,0x00,0x00,0x00,0
    x00,0x00,0x00,0x00,0x00,0xFF,0x0F,0xFE,0x3C
    ,0x0F,0xFF,0xFF,0x00,0x0C,0x7F,0xFF,0x80,0
    x00,0x00,0x80,0xE0,0x20,0x00,0x00,0xFF,0xFF
    ,0x00,0x00,0xFF,0x7F,0x00,0x00,0x00,0x00,
50  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
    x00,0x00,0x01,0x01,0x01,0x01,0x00,0x00,0x00
    ,0x00,0x01,0x01,0x01,0x01,0x01,0x00,0x00,0
    x00,0x01,0x01,0x01,0x01,0x00,0x00,0x00,0x00
    ,0x00,0x00,0x00,0x00,0x01,0x01,0x01,0x00,0
    x00,0x00,0x00,0x00,0x01,0x01,0x01,0x01,0x00
    ,0x00,0x01,0x01,0x01,0x01,0x00,0x01,0x01,0
    x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00
    ,0x00,0x00,0x01,0x01,0x01,0x00,0x00,0x01,0
    x01,0x01,0x01,0x00,0x00,0x00,0x01,0x01,0x01
    ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
```

```
        x00,0x00,0x00,0x00,0x01,0x01,0x01,0x00,0x00
        ,0x01,0x01,0x01,0x01,0x00,0x00,0x00,0x01,0
        x01,0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x01
        ,0x01,0x01,0x01,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
        x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
        ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
        x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
        ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
        x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
        ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
        x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
        ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
        x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
        ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
        x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
        ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
        x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
        ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};

void scrollRight() {
    unsigned char tmp[10];
    int i = 0, j = 0;
    for(i = 0, j = 127; i < 8; i++) {
        tmp[i] = map[j];
        j += 128;
    }

```

```
62      for(i = 127; i >= 1; i--) {
63          for(j = 0; j < 8; j++) {
64              map[128 * j + i] = map[128 * j + i -
                    1];
65          }
66      }
67
68      for(i = 0; i < 8; i++) {
69          map[128 * i] = tmp[i];
70      }
71
72      return;
73  }
74
75  void scrollLeft() {
76
77      unsigned char tmp[10];
78      int i = 0, j = 0;
79      for(i = 0; i < 8; i++) {
80          tmp[i] = map[128 * i];
81      }
82
83      for(i = 0; i < 128 - 1; i++) {
84          for(j = 0; j < 8; j++) {
85              map[128 * j + i] = map[128 * j + i +
                    1];
86          }
87      }
88
```

```
89      for(i = 0; i < 8; i++) {
90          map[128 * i + 127] = tmp[i];
91      }
92
93      return;
94  }
95
96
97
98  int main(void)
99  {
100     int i = 5, keyPressed = 0;
101     int currentStatus = 0;
102     SYS_Init();
103     Init_KEY();
104     OpenKeyPad();
105     init_LCD();
106     clear_LCD();
107
108
109     while(1) {
110         draw_LCD(map);
111
112         if ( flag == 1 ) {
113             scrollLeft();
114         } else if ( flag == 3 ) {
115             scrollRight();
116         }
117     }
```

```
118
119 }
```

- 實驗二 5pt 數字會跑步

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "NUC100Series.h"
4  #include "MCU_init.h"
5  #include "SYS_init.h"
6  #include "LCD.h"
7  #include "Scankey.h"
8  #include "Seven_Segment.h"
9
10 #define INF (int)(1e9)
11 #define MAXN (int)(1e5 + 10)
12
13 int onRestart = 0;
14 int onStart   = 0;
15
16
17 int players[4] = {0};
18 int speedOfPlayers[4] = {0};
19 int posOfPlayers[4][2] = {
20     {0, 0},
21     {0, 16},
22     {0, 32},
23     {0, 48}
```

```
};

int onEnd[4] = {0, 0, 0, 0};

const int staticSpeed[4] = {2, 4, 6, 8};

int seed = 0;

void show_LCD() {
    int i = 0;
    clear_LCD();
    for(i = 0; i < 4; i++) {
        char player = (char)(players[i] + '0');
        printC(posOfPlayers[i][0], posOfPlayers[i
           ][1], player);
    }
    return;
}

int cmp(const void* a, const void* b) {
    return (*(int*)a - *(int*)b);
}


void restart() {

    int i = 0;
    int temp[4];
    int map[20] = {0};
```

```
52      int visited[20] = {0};
53
54
55      onRestart = 0;
56      PC12 = PC13 = PC14 = PC15 = 1;
57      for(i = 0; i < 4; i++) {
58          posOfPlayers[i][0] = 0;
59          onEnd[i] = 0;
60      }
61
62      for(i = 0; i < 4; i++) {
63
64          while( 1 ) {
65              int newPlayer = rand() % 9 + 1;
66              if ( !visited[newPlayer] ) {
67                  visited[newPlayer] = 1;
68                  players[i] = newPlayer;
69                  break;
70              }
71              seed = (seed + 1) % INF;
72          }
73
74      }
75
76      for(i = 0; i < 4; i++) {
77          temp[i] = players[i];
78      }
79
80      qsort(temp, 4, sizeof(int), cmp);
```

```
81
82      for(i = 0; i < 4; i++) {
83          map[temp[i]] = staticSpeed[i];
84      }
85
86      for(i = 0; i < 4; i++) {
87          speedOfPlayers[i] = map[players[i]];
88      }
89
90      show_LCD();
91
92      return;
93  }
94
95  void GPAB_IRQHandler(void)
96  {
97      int flag, i;
98      if (PA->ISRC & BIT0) {        // check if PB12
            interrupt occurred
99          PA->ISRC |= BIT0;         // clear PB12
               interrupt status
100                                 // set a flag for PB12(
                                       KEY1)
101     } else if (PA->ISRC & BIT1) { // check if Pb13
            interrupt occurred
102         PA->ISRC |= BIT1;         // clear PB13
               interrupt status
103                             // set a flag for PB13(KEY2)
104     } else if (PA->ISRC & BIT2) { // check if PB14
```

```
        interrupt occurred
105         PA->ISRC |= BIT2;          // clear PB14
            interrupt status
106                             // set a flag for PB14(
                                KEY3)
107     } else {                        // else it is
        unexpected interrupts
108         PA->ISRC = PA->ISRC;            // clear all
             GPB pins
109                 PA0 = PA1 = PA2 = 1;
110
111         }
112
113         if ( onRestart ) restart();
114 }
115
116 void Init_KEY(void)
117 {
118     GPIO_SetMode(PA, (BIT3 | BIT4 | BIT5),
        GPIO_MODE_OUTPUT);
119         GPIO_SetMode(PA, (BIT0 | BIT1 | BIT2),
            GPIO_MODE_QUASI);
120     GPIO_EnableInt(PA, 0, GPIO_INT_FALLING);
121     GPIO_EnableInt(PA, 1, GPIO_INT_FALLING);
122     GPIO_EnableInt(PA, 2, GPIO_INT_FALLING);

123     NVIC_EnableIRQ(GPAB_IRQn);
124     GPIO_SET_DEBOUNCE_TIME(GPIO_DBCLKSRC_LIRC,
        GPIO_DBCLKSEL_64);
```

```
125     GPIO_ENABLE_DEBOUNCE(PA, (BIT0 | BIT1 | BIT2))
            ;
126       PA3 = PA4 = PA5 = 0;
127 }
128
129
130
131 void EINT1_IRQHandler(void)
132 {
133     GPIO_CLR_INT_FLAG(PB, BIT15);   // Clear GPIO
            interrupt flag
134     if ( onRestart ) {
135         restart();
136         return;
137     }
138
139     onStart = 1;
140 }
141
142 void Init_EXTINT(void)
143 {
144     // Configure EINT1 pin and enable interrupt by
            rising and falling edge trigger
145     GPIO_SetMode(PB, BIT15, GPIO_MODE_INPUT);
146     GPIO_EnableEINT1(PB, 15, GPIO_INT_RISING); //
            RISING, FALLING, BOTH_EDGE, HIGH, LOW
147     NVIC_EnableIRQ(EINT1_IRQn);
148
149     // Enable interrupt de-bounce function and
```

```
        select de-bounce sampling cycle time
150     GPIO_SET_DEBOUNCE_TIME(GPIO_DBCLKSRC_LIRC,
        GPIO_DBCLKSEL_64);
151     GPIO_ENABLE_DEBOUNCE(PB, BIT15);
152 }
153
154
155
156
157 void showEnd(int winner) {
158     winner += 1;
159     if ( winner == 1 ) PC12 = 0;
160     else if ( winner == 2 ) PC13 = 0;
161     else if ( winner == 3 ) PC14 = 0;
162     else if ( winner == 4 ) PC15 = 0;
163     return;
164 }
165
166
167
168 void checkEndGame() {
169     int amountOfEnd = 0, i = 0;
170     for(i = 0; i < 4; i++) {
171         if ( posOfPlayers[i][0] >= 120 ) {
172             onEnd[i] = 1;
173             showEnd(i);
174         }
175     }
176
```

```
177     for(i = 0; i < 4; i++) {
178         amountOfEnd += onEnd[i] != 0 ? 1 : 0;
179     }
180 
181     if ( amountOfEnd >= 4 ){
182         onRestart = 1;
183         onStart   = 0;
184     }
185     return;
186 }
187 
188 
189 int main(void)
190 {
191     int i;
192     SYS_Init();
193     Init_KEY();
194     Init_EXTINT();
195     OpenKeyPad();
196     init_LCD();
197     clear_LCD();
198 
199     restart();
200 
201     while(1) {
202         show_LCD();
203         CLK_SysTickDelay(2000000);
204 
205         checkEndGame();
```

```
206             seed = (seed + 1) % INF;
207             srand(seed);
208
209             for(i = 0; i < 4; i++) {
210                 if ( !onEnd[i] && onStart ) {
211                     posOfPlayers[i][0] +=
                            speedOfPlayers[i];
212                 }
213             }
214         }
215
216         return 0;
217     }
```

四、 未能解決的問題

沒有未能解決的問題。