實驗六 STM32 Keypad Scanning

1. 實驗目的

- 了解STM32使用原理
- 了解如何使用C code控制STM32
- 設計7-SegLED和keypad程式

2.實驗原理

Keypad電路組成如下，主要是一個4x4的鍵盤按鈕所組成會用到4個Input pin與 4個Output pin，其控制原理是利用Output pin掃描的方式來決定目前所選擇到 的 是哪一行按鍵，例如當KEY X0~3輸出1000而此時若KEY Y0~3所讀到的值是1 000的話則代表SW14按鈕被按下。

3. 實驗步驟

3.1. Lab 6.1: Max7219 displayer (30%)

將Lab5所完成的GPIO_init()與MAX7219_send()改成可以被C所呼叫的版本， 並新增一個Cfile完成displayfunction及利用max7219_send()將學號顯示於7段 顯示器上。

※細節都再註解中

main.s

```
#include "../inc/stm32l476xx.h"

extern void GPIO_init();
extern void max7219_init();
extern void max7219_send(unsigned char address, unsigned char data);

int main(){
	GPIO_init();
	max7219_init();
	int student_id[] = {0,6,1,6,4,2,1};

	for(int i=0; i<7; i++){
			max7219_send(7-i, student_id[i]);
	}
}
```

init.s

```
        .syntax unified
        .cpu cortex-m4
        .thumb
.text
        .global GPIO_init
        .global max7219_init
        .global max7219_send
        .equ RCC_AHB2ENR, 0x4002104C
        .equ GPIOB_MODER, 0x48000400
        .equ GPIOB_OSPEEDR, 0x48000408
        .equ GPIOB_PUPDR,0x4800040C
        .equ GPIOB_BSRR, 0x48000418 // set
        .equ GPIOB_BRR, 0x48000428 // reset
        .equ DATA, 0b1000 // PB3
        .equ LOAD, 0b10000 // PB4
        .equ CLK, 0b100000 // PB5
        .equ DECODE_MODE, 0x09 // 0xX9
        .equ DISPLAY_TEST, 0x0F // 0xXF
        .equ SCAN_LIMIT, 0x0B // 0xXB
        .equ INTENSITY, 0x0A // 0xXA
        .equ SHUTDOWN, 0x0C // 0xXC
        .equ X, 1
        .equ Y, 1000000

// GPIO_init()
GPIO_init:
        /* AHB2 */
        ldr r0, =RCC_AHB2ENR
        movs r1, #2
        str r1, [r0]

        /* GPIOA */
        // MODER output 3 4 5
        ldr r0, =GPIOB_MODER
        ldr r1, [r0]
        and r1, 0xFFFFF03F
        movs r2, 0x540
        orr r1, r2
        str r1, [r0]

        // SPEEDR high speed 10
```

```
        ldr r0, =GPIOB_OSPEEDR
        ldr r1, [r0]
        and r1, 0xFFFFF03F
        movs r2, 0xA80
        orr r1, r2
        str r1, [r0]

        // PUPDR pull up 01
        ldr r0, =GPIOB_PUPDR
        ldr r1, [r0]
        and r1, 0xFFFFF03F
        movs r2, 0x540
        orr r1, r2
        str r1, [r0]
        bx lr
// end of GPIO_init

// max7219_send(adr , data)
max7219_send:    //input parameter: r0 is ADDRESS , r1 is DATA
        push {r0-r9}
        lsl r0, #8
        add r0, r1
        ldr r1, =GPIOB_MODER
        ldr r2, =LOAD
        ldr r3, =DATA
        ldr r4, =CLK
        ldr r5, =GPIOB_BSRR
        ldr r6, =GPIOB_BRR
        mov r7, #16 // r7 = i
max7219_send_loop:
        mov r8, #1
        sub r9, r7, #1
        lsl r8, r8, r9 // r8 = mask
        str r4, [r6] // CLK = 0
        tst r0, r8
        beq bit_not_set
        str r3, [r5] // DATA = 1
        b if_done
bit_not_set:
        str r3, [r6] // DATA = 0
if_done:
        str r4, [r5] // CLK = 1
        subs r7, #1 // i--
        bgt max7219_send_loop // i > 0
```

```
        str r2, [r6] // Load = 0
        str r2, [r5] // Load = 1
        pop {r0-r9}
        bx lr
// end of max7219_send(adr, data)

// max7219_init()
max7219_init:
        push {lr}
        ldr r0, =DECODE_MODE
        mov r1, 0xFF // 0xFF => decode all digits 7-0
        bl max7219_send
        ldr r0, =DISPLAY_TEST
        mov r1, 0x00 // 0 => normal operation
        bl max7219_send
        ldr r0, =SCAN_LIMIT
        mov r1, 0x06 // 0xX6 => display digits 0-6
        bl max7219_send
        ldr r0, =INTENSITY
        mov r1, 0x0F // 0xX0 => 1/32
        bl max7219_send
        ldr r0, =SHUTDOWN
        mov r1, 0x01 // 1 => normal operation
        bl max7219_send
        pop {lr}
        bx lr
```

3.2. Lab6.2: KeypadScanning (30%)

利用4個input GPIO與4個output GPIO pin連接keypad，當按下keypad利用 兩顆七段顯示器顯示所對應的數字。

※細節都再註解中

main.c (init.s同Lab6.1, 但scan_limit由0x6改為0x7)

```
#include "stm32l476xx.h"
#define X0 6
#define X1 7
#define X2 8
#define X3 9
#define Y0 11
```

```
#define Y1 12
#define Y2 13
#define Y3 14
unsigned int x_pin[4] = {X0, X1, X2, X3}; // GPIOA 6 7 8 9
unsigned int y_pin[4] = {Y0, Y1, Y2, Y3}; // GPIOB 11 12 13 14

extern void GPIO_init();
extern void max7219_init();
extern void max7219_send(unsigned char adr, unsigned char data);

int output=0;
int Table[4][4] = {{1,2,3,10},{4,5,6,11},{7,8,9,12},{15,0,14,13}};

/* TODO: initial keypad gpio pin, X as output and Y as input */
void keypad_init() {
	/*============== AHB2=================*/
	RCC->AHB2ENR = RCC->AHB2ENR | 0b11;
	/*============== GPIOA ================*/
	// MODER: output mode (01): 6 7 8 9
	GPIOA->MODER = GPIOA->MODER & 0xFFF00FFF;
	GPIOA->MODER = GPIOA->MODER | 0x00055000;
	// OSPEEDR: medium speed (01)
	GPIOA->OSPEEDR = GPIOA->OSPEEDR & 0xFFF00FFF;
	GPIOA->OSPEEDR = GPIOA->OSPEEDR | 0x00055000;
	// PUPDR: pull up (01)
	GPIOA->PUPDR = GPIOA->PUPDR & 0xFFF00FFF;
	GPIOA->PUPDR = GPIOA->PUPDR | 0xFFF55000;
	/*============== GPIOB ================*/
	// MODER: input mode (00): 11 12 13 14
	GPIOB->MODER = GPIOB->MODER & 0xC03FFFFF;
	// OSPEEDR: medium speed (01)
	GPIOB->OSPEEDR = GPIOB->OSPEEDR & 0xC03FFFFF;
	GPIOB->OSPEEDR = GPIOB->OSPEEDR | 0x15400000;
	// PUPDR: pull down (10)
	GPIOB->PUPDR = GPIOB->PUPDR & 0xC03FFFFF;
	GPIOB->PUPDR = GPIOB->PUPDR | 0x2A800000;
}
/* TODO: scan keypad value return: >=0: key pressed_value -1: no key_press */
int keypad_scan() {
	int k,flag_keypad, flag_debounce, flag_keypad_r;
	// detect next button
	GPIOA->ODR = GPIOA->ODR | (0b1111 << 6);
	while(1){
		flag_keypad=GPIOB->IDR & (0b1111<<11);
```

```
                if(flag_keypad!=0){ // check the button is pressed
                        k=45000;
                        while(k!=0){ // check the debounce
                                flag_debounce = GPIOB->IDR&(0b1111 << 11);
                                k--;
                        }
                        if(flag_debounce!=0){
                                for(int i=0; i<4; i++){ // scan row
                                        GPIOA->ODR =
(GPIOA->ODR&0xFFFFFC3F)|(1<<x_pin[i]);
                                        for(int j=0; j<4; j++){ // scan column
                                                flag_keypad_r = GPIOB->IDR&(1<<y_pin[j]);
                                                if(flag_keypad_r != 0){
                                                        return Table[i][j];
                                                }
                                        }
                                }
                        }
                }
                else {
                        return -1;
                }
        }
}

void display(int data){
        if(data==-1){ // display nothing while no button is pressed
                for(int i=1; i<=8; i++)
                        max7219_send(i,0xF);
        }
        else{
                for(int i=1; i<=8; i++){
                        if(data > 0)
                                max7219_send(i,data%10);
                        else{
                                if(i==1)
                                        max7219_send(1,0);
                                else
                                        max7219_send(i,0xF); // send space
                        }
                        data /= 10;
                }
        }
}
```

```
int main(){
	GPIO_init();
	keypad_init();
	max7219_init();
	display(0);
	while (1) {
		int keypad_value = keypad_scan();

		if (keypad_value != -1)
			display(keypad_value);
		else
			display(-1);
	}
}
```

3.3. Lab6.3 single and multi buttons處理單或多按鍵 (40%)

利用keypad輸入數字並在七段顯示器顯示，各按鍵對應值為：

	X0	X1	X2	X3
Y0	1	2	3	10
Y1	4	5	6	11
Y2	7	8	9	12
Y3	C	0	C	13

當按多按鍵時，會將按鍵值相加並顯示出來(按1、5、9則顯示15)，若準備顯示 的值>99999999，則不更動原本七段顯示器上顯示的數字，直到按下消除鍵 (C)。

※細節都在註解中

main.c (紅色為Lab6.2更改的部分, init.s同Lab6.2)

```
#include "stm32l476xx.h"
#define X0 6 // PA6
```

```
#define X1 7 // PA7
#define X2 8 // pAB
#define X3 9 // PA9
#define Y0 11 // PB11
#define Y1 12 // PB12
#define Y2 13 // PB13
#define Y3 14 // PB14
unsigned int x_pin[4] = {X0, X1, X2, X3}; // GPIOA 6 7 8 9
unsigned int y_pin[4] = {Y0, Y1, Y2, Y3}; // GPIOB 11 12 13 14

extern void GPIO_init();
extern void max7219_init();
extern void max7219_send(unsigned char adr, unsigned char data);

int output=0;
int Table[4][4] = {{1,2,3,10},{4,5,6,11},{7,8,9,12},{15,0,14,13}};

/* TODO: initial keypad gpio pin, X as output and Y as input */
void keypad_init() {
	/*====================== AHB2 ======================*/
	RCC->AHB2ENR = RCC->AHB2ENR | 0b11;
	/*====================== GPIOA ======================*/
	// MODER: output mode (01): 6 7 8 9
	GPIOA->MODER = GPIOA->MODER & 0xFFF00FFF;
	GPIOA->MODER = GPIOA->MODER | 0x00055000;
	// OSPEEDR: medium speed (01)
	GPIOA->OSPEEDR = GPIOA->OSPEEDR & 0xFFF00FFF;
	GPIOA->OSPEEDR = GPIOA->OSPEEDR | 0x00055000;
	// PUPDR: pull up (01)
	GPIOA->PUPDR = GPIOA->PUPDR & 0xFFF00FFF;
	GPIOA->PUPDR = GPIOA->PUPDR | 0xFFF55000;
	/*====================== GPIOB ======================*/
	// MODER: input mode (00): 11 12 13 14
	GPIOB->MODER = GPIOB->MODER & 0xC03FFFFF;
	// OSPEEDR: medium speed (01)
	GPIOB->OSPEEDR = GPIOB->OSPEEDR & 0xC03FFFFF;
	GPIOB->OSPEEDR = GPIOB->OSPEEDR | 0x15400000;
	// PUPDR: pull down (10)
	GPIOB->PUPDR = GPIOB->PUPDR & 0xC03FFFFF;
	GPIOB->PUPDR = GPIOB->PUPDR | 0x2A800000;
}
/* TODO: scan keypad value return: >=0: key pressed_value -1: no key_press */
int keypad_scan() {
	int k, flag_keypad, flag_debounce, flag_keypad_r, sum=0;
```

```
        // detect next button
        GPIOA->ODR = GPIOA->ODR | (0b1111 << 6);
        while(1){
                flag_keypad=GPIOB->IDR & (0b1111<<11);
                if(flag_keypad!=0){ // check the button is pressed
                        k=45000;
                        while(k!=0){ // check the debounce
                                flag_debounce = GPIOB->IDR&(0b1111 << 11);
                                k--;
                        }
                        if(flag_debounce!=0){
                                for(int i=0; i<4; i++){ // row
                                        GPIOA->ODR =
(GPIOA->ODR&0xFFFFFC3F)|(1<<x_pin[i]);
                                        for(int j=0; j<4; j++){ // column
                                                flag_keypad_r = GPIOB->IDR&(1<<y_pin[j]);
                                                if(flag_keypad_r != 0){
                                                        if((i==3&&j==0) || (i==3&&j==2))
                                                                return -1;
                                                        else
                                                                sum+=Table[i][j];
                                                }
                                        }
                                }
                                return sum;
                        }
                }
        }
}

void display(int data){
        for(int i=1; i<=8; i++){
                if(data > 0)
                        max7219_send(i,data%10);
                else{
                        if(i==1)
                                max7219_send(1,0);
                        else
                                max7219_send(i,0xF);
                }
                data /= 10;
        }
}
```

```
int main(){
        GPIO_init();
        keypad_init();
        max7219_init();
        display(0);
        int sum=0;
        while (1) {
                display(sum);
                int keypad_value = keypad_scan();

                if (sum+keypad_value > 99999999)
                        continue;
                if (keypad_value != -1)
                        sum += keypad_value;
                else
                        sum = 0;
        }
}
```