



實驗六 STM32 Keypad Scanning

0516322朱蝶、0516059劉嘉豪

1. Lab objectives 實驗目的

- Understand the principle of STM32
- Use C code to control STM32
- design program for 7-seg LED and keypad
- 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

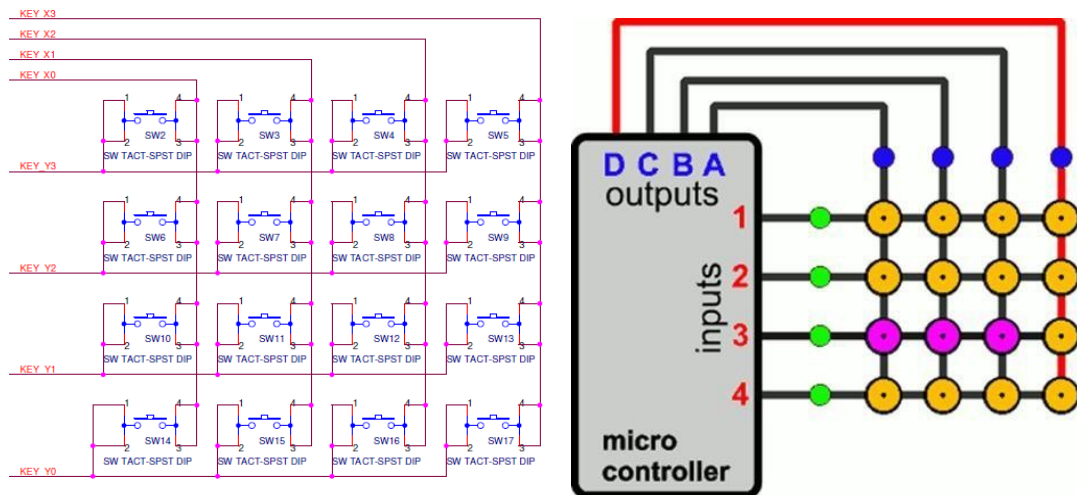
2 Lab principle 實驗原理

The circuit diagram of keypad is given below. You're supposed to use 4 input pins and 4 output pins. Use output pins to determine which row you're scanning. For example, when output value of KEY X0~3 is 1000 and input value of KEY Y0~3 is 1000, then we can say that SW14 is pressed.

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 1000 的話則代表 SW14 按鈕被按下。





KEYPAD PINOUT:

PIN 1: COL 4
 PIN 2: COL 3
 PIN 3: COL 2
 PIN 4: COL 1
 PIN 5: ROW 4
 PIN 6: ROW 3
 PIN 7: ROW 2
 PIN 8: ROW 1

3. Steps 實驗步驟 1

3.1. Lab 6.1: Max7219 displayer (30%)

Modify your code in lab5.2 to make it callable by C. Add a C file to complete the code given below, display your student ID on 7-Seg LED.

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。

1.Problem

首先建立一個project，照作業需求所說將.h檔案放在inc然後環境要設定好（路徑設好），再把lab5.2的檔案分成max7219_init.s, max7219_send.s, GPIO_init.s三個檔案，每個都要先宣告global function，這樣才可以被.c呼叫（附上例圖）

```

.syntax unified
.cpu cortex-m4
.thumb
.data
arr: .byte 0x0, 0x5, 0x1, 0x6, 0x0, 0x5, 0x9

.text
.global GPIO_init
.global max7219_send
.global max7219_init
//GPIO
.equ RCC_AHB2ENR, 0x4002104C
.equ GPIOA_MODER, 0x48000000
.equ GPIOA_OTYPER, 0x48000004
.equ GPIOA_OSPEEDER, 0x48000008
.equ GPIOA_PUPDR, 0x4800000C
.equ GPIOA_IDR, 0x48000010
.equ GPIOA_ODR, 0x48000014 //PA5 6 7 output mode
.equ GPIOA_BSRR, 0x48000018 //set bit -> 1
.equ GPIOA_BRR, 0x48000028 //clear bit -> 0

//Din, CS, CLK offset
.equ DIN, 0b100000 //PA5
.equ CS, 0b1000000 //PA6
.equ CLK, 0b10000000 //PA7

//max7219
.equ DECODE, 0x19 //decode control
.equ INTENSITY, 0x1A //brightness
.equ SCAN_LIMIT, 0x1B //how many digits to display
.equ SHUT_DOWN, 0x1C //shut down -- we didn't use this
.equ DISPLAY_TEST, 0x1F //display test -- we didn't use this

//timer
.equ one_sec, 1000000 //try and error
  
```

下面再打上function。

main.c 檔案要加上

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

好，再來就很簡單了，main function一開始先做gpio, max7210的init，最後再去display

```
int main()
{
    int student_id = 516059;
    GPIO_init();
    max7219_init();
    //display_clear(8);
    display(student_id, 7);
    return 0;
}
```

display將要顯示的數字data、要顯示幾位數num_digs傳進來

設一個for迴圈，跑num_digs次，每次把data的最後一位印出來(max7219_send(i, data%10)，前面是顯示在第幾個led後面是數值，最後data/=10一直印直到最後一次，把0印出來(max7219_send(i, 0)。如果要顯示的數字大於7seg-led可以顯示的八位 (>99999999) 或是小於 (-9999999) 則return -1 不然就return 0

2.遇到的問題

第一次學到如何用.c檔案call function，其實這題蠻簡單的，算是熱身題xdddd

3.2. Lab6.2: KeypadScanning (30%)



Use 4 input GPIO pins and 4 output GPIO pins to connect with keypad. Show the corresponding number of pressed button on 7-SegLED.

Note: Use C to init GPIO used by keypad. Please refer to stm32l476xx.h for GPIO register address and structure define.

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。

Note: keypad 所使用到的 GPIO 請利用 C 語言的方式初始化，各 GPIO register

address 與 structuredefine 請參考 stm32l476xx.h

```
#include "stm32l476xx.h"
//TODO: define your gpio pin
#define X0
#define X1
#define X2
#define X3
#define Y0
#define Y1
#define Y2
#define Y3

unsigned int x_pin[4] = {X0, X1, X2, X3}; unsigned int
y_pin[4] = {Y0, Y1, Y2, Y3};

/* TODO: initial keypad gpio pin, X as output and Y as input
*/
void keypad_init()
{
}

/* TODO: scan keypad value return:
    >=0: key pressedvalue
    -1: no keypress
*/
char keypad_scan()
```

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	15	0	14	13

1.Problem

這題的main跟上一題差不多，只是要先把數字清掉，不然就會顯示剛剛的學號
display function傳入要清掉幾個位置，寫一個for迴圈共跑那個數值次，每次都
max7219_send(i, 0xF)，1111是blank，所以是改成不顯示任何數字。

再來init keypad我們的output接portC 0-3, input 接portB 0-3，這個init就跟以前一樣，
RCC->AHB2ENR調成enable port a, b, c，portC的moder改成output mode, pull-up,
mid-speed, odr等等，portB則是input mode, pull-down, idr(如下圖)。

```

GPIO_init(); //have initialized in arm
RCC->AHB2ENR |= 0b00000000000000000000000000000111;
GPIOC->MODER &= 0b11111111111111111111111110000000;
GPIOC->MODER |= 0b00000000000000000000000000001010101;
GPIOC->PUPDR &= 0b11111111111111111111111110000000;
GPIOC->PUPDR |= 0b00000000000000000000000000001010101;
GPIOC->OSPEEDR &= 0b11111111111111111111111110000000;
GPIOC->OSPEEDR |= 0b00000000000000000000000000001010101;
GPIOC->ODR |= 0b000000000000000000000000000011110000;

GPIOB->MODER &= 0b11111111111111111111111110000000;
GPIOB->PUPDR &= 0b11111111111111111111111110000000;
GPIOB->PUPDR |= 0b00000000000000000000000000001010101;

```

再來就是keypad scan了，這個函數裡面有一個while跑無限次，每次掃column跟row，用兩個迴圈。迴圈裡面清空portC的odr然後用left shift將那個row的data送出去。至於portB的input right shift row次再and 1可以知道有沒有被按到，比如輸入值100則right shift 2 次使值變成1就可以知道那個row被按到了(如圖)

```

int keypad_row=0,keypad_col=0;
char key_val=-1;
while(1)
{
    for(keypad_col=0;keypad_col<4;keypad_col++)
    {
        for(keypad_row=0;keypad_row<4;keypad_row++)
        {
            GPIOC->ODR&=0; //clear output
            GPIOC->ODR|=(1<<keypad_col); //shift and send data to that row
            int masked_value=GPIOB->IDR&0xf, is_pressed=(masked_value>>keypad_row)&1;

```

如果被按到的話，就把值用display顯示出來，除了值大於9以外要顯示兩個，其他都只要顯示一個led就好，如果沒有被按就清掉數值。

2. Questions

一開始明明第一個環境就可以跑，main.c寫完卻一直跑出錯，然後找了半天以為我是哪裡觀念有問題這題要重新用不同環境，結果是我把作業講義上的sample code複製貼上的時候，雙引號自己跑掉了！！！！超級生氣的浪費我好多時間，有夠智障，好拉是我智障。

然後這題比較難的是接線，那8個pin 怎麼接一開始頭腦有夠混亂的，所以出來的數字都很奇怪哈哈，而且上課講義跟作業的y3-y0好像是相反的，頭很痛，反正最後還是搞懂了，我們8-1分別是接pb0123pc0123。



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

Show pressed button of keypad on 7-Seg LED. Each value of corresponding button is given below.

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

When multiple buttons are pressed, show the sum of values that buttons pressed representing. If shown value is greater than 99999999, don't modify the number showing on 7-Seg LeD until button C is pressed.

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



1. Problem

第三題的main都差不多，一開始也是要init gpio, keypad, max7219然後把八個 led 位置的顯示都清掉。keypad scan這個函數也是一個while迴圈，裡面設定兩個state，如果是state 0代表沒有被按過，會先跟上題一樣做兩個for迴圈，大部分都一樣，就不再多說，比較特別的是：

如果被按過了(is_pressed==1)而且<14的話，就把那一格的map設1代表按過，不然的話就是按到歸零(一開始設歸零按鈕的值是15)，則輸出的值設為0 (outsum)，那再用一個for迴圈，剛剛map第幾個設為1，sum的值就是多少。

再根據outsum顯示對應的數值，比如說0-9就顯示一個display(outsum, 1)，10000000-99999999就顯示八個數字這樣

如果沒有東西被按，就把map歸零，然後把state設為1跑去下一個case，這個case在沒人按的時候會讓led一直顯示那個數值，

第二個case把state=0使下次回到上一個state看有沒有被按，然後把按過的數字加起來，最後一樣根據 outsum 顯示對應的數值，比如說0-9就顯示一個 display(outsum, 1)，10000000-99999999就顯示八個數字這樣（如下圖），結束就回到while迴圈繼續跑。

```
switch(out_sum){
case 0 ... 9:
    if(out_sum!=0 || clear!=
        display(out_sum,1);
    else
        display_clr(8);
    break;
case 10 ... 99:
    display(out_sum,2);
    break;
case 100 ... 999:
    display(out_sum,3);
    break;
case 1000 ... 9999:
    display(out_sum,4);
    break;
case 10000 ... 99999:
    display(out_sum,5);
    break;
case 100000 ... 999999:
    display(out_sum,6);
    break;
case 1000000 ... 9999999:
    display(out_sum,7);
    break;
case 10000000 ... 99999999:
    display(out_sum,8);
    break;
}
```

2.Question

這題遇到的問題就是複雜好多喔！！要想很久，不過雖然寫得很慢好像也沒有什麼大問題，問題比較大的應該是明天的補考qq

心得：這次lab用的這個keypad比較不會有之前dip壞掉的情況，接線部分沒什麼問題，但是明天還有補考qq覺得很恐怖qqqqqq然後身體又很不舒服好想吐嗚嗚嗚不知道怎麼辦，總之明天加油！！！！