Lab5 STM32 Keypad Scanning
實驗五 STM32 Keypad Scanning

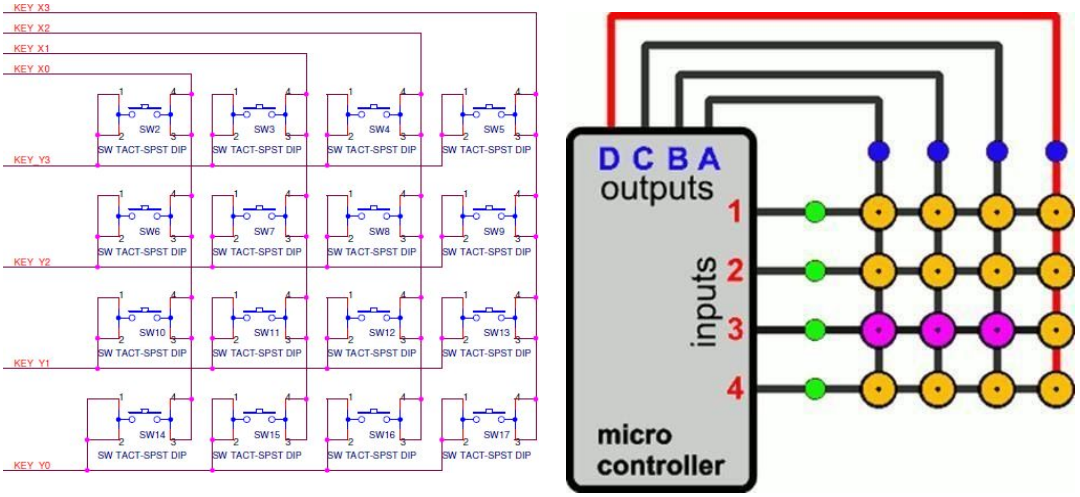
1. Lab objectives 實驗目的

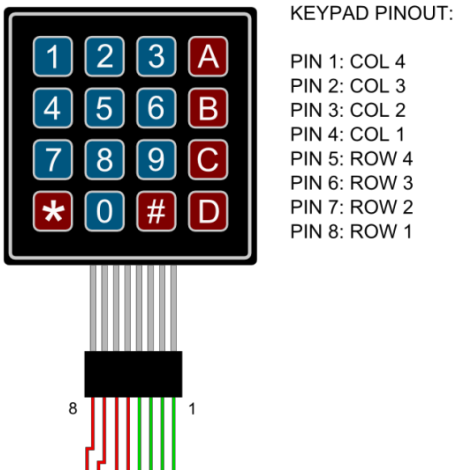
- Understand the principle of STM32
- Use C code to control STM32
- design program for 7-seg LED and keypad
- 了解 STM32 使用原理
- 了解如何使用 C code 控制 STM32
- 設計 7-Seg LED 和 keypad 程式

2. Lab principle 實驗原理

The circuit diagram of the 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 the 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。





Please check the course material of lab5_note

請參考 lab5_note 課程講義。

3. Steps 實驗步驟

3.1. Max7219 displayer

Requirement: Modify GPIO_init(), max7219_init() and max7219_send() finished in Lab4 to make it callable by C. Add a C file to complete code below. Finally, display your student ID on 7-Seg LED.

將 Lab4 所完成的 GPIO_init(), max7219_init() 與 max7219_send() 改成可以**被 C 所呼叫**的版本，並新增一個 C file 完成以下程式碼。最終將學號顯示於 7 段顯示器上。

```
//These functions inside the asm file
extern void GPIO_init();
extern void max7219_init();
extern void max7219_send(unsigned char address, unsigned char
data);
/**
* TODO: Show data on 7-seg via max7219_send
*   Input:
*     data: decimal value
*     num_digs: number of digits will show on 7-seg
*   Return:
*     0: success
*     -1: illegal data range(out of 8 digits range)
*/
int display(int data, int num_digs){

}
void main(){
   int student_id = 1234567;
   GPIO_init();
   max7219_init();
   display(student_id, 8);
}
```

3.2. KeypadScanning

Requirement: Use 4 input GPIO pins and 4 output GPIO pins to connect with keypad. Show the corresponding number of the pressed button on 7-Seg LED. Don't show any numbers when the button is released.

利用 4 個 input GPIO pins 與 4 個 output GPIO pins 來連接 keypad。當按住 keypad 時利用七段顯示器顯示所對應的數字。放開 keypad 時則不顯示數字。

Note: Please refer to stm32l476xx.h. To initialize the GPIO configuration using C, you may need the structure defined in the head file to access the corresponding GPIO registers.

Note: 請參考stm32l476xx.h。使用C語言初始化 GPIO 配置時，您可能需要標頭檔中定義的 struct 結構來訪問對應的 GPIO 暫存器。

```
#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-value pressed, -1: keypad is free
*/
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

3.3. multi buttons 處理多按鍵

Requirement: Corresponding to the previous question, please modify your implement of "`keypad_scan`". Then, it can still work when we press two keys or less at the same time, and shows the sum of the key value(s) on the 7-segment display.

乘上題，請修改你的 "`keypad_scan`" 實作。讓它可以在我們同時按住一個鍵或兩個鍵時正常運作，並且將鍵值相加的結果顯示於七段顯示器上。

e.g.

- In the beginning, don't display any numbers on 7-Seg
- Press 1, display 1
- Press 3 without releasing 1, display 4
- Release 1 with 3 pressed, display 3
- Press 9 without releasing 3, display 12
- Release all, don't display number
- Press 1, 5 at same time, display 6

範例

- 初始，沒有顯示任何數字在七段顯示器
- 按住 1，顯示 1
- 按住 1 之下按住 3，顯示 4
- 按住 3 之下放開 1，顯示 3
- 按住 3 之下按住 9，顯示 12
- 放開全部，沒有顯示任何數字
- 按住 1, 5，顯示 6

3.4. Question 實驗課問題

Question 1: In Requirement 3-1, we used the function, "`max7219_send`" which is implemented in arm asm. How did we pass the arguments? That is, Where were the arguments "address" and "data" stored?

在 Requirement 3-1 中我們使用了透過 arm 組合語言實做的 "`max7219_send`"。請問這個函式的引數是如何被傳遞的？即 address 和 data 會被存在哪裡？

Question 2: In stm32l476xx.h, variables are defined with the keyword "**volatile**" (__IO). Please describe its function? What problems can be avoided?

在 stm32l476xx.h 中，變量被使用關鍵字 "**volatile**" (__IO) 來定義。請說明它的功能是什麼？可以避免甚麼問題？

3.5. Reference & Hint 參考資料與提示

Hint 1: When we press key 1, key 2, key 5 at the same time, we find that key 4 is also detected. This phenomenon is called "Phantom Key" or "Ghost Key". It's because of the defects in hardware design. Can you figure that out?

當我們同時按下鍵1，鍵2，鍵5時，我們發現同時會檢測到鍵4。這種現象被稱為 “幽靈鍵” 或 “鬼鍵”。 這是由於硬件設計中的缺陷。 你能指出它（的原因）嗎？

3.5. Reference & Hint 參考資料與提示

Hint 1: When we press key 1, key 2, key 5 at the same time, we find that key 4 is also detected. This phenomenon is called "Phantom Key" or "Ghost Key". It's because of the defects in hardware design. Can you figure that out?

當我們同時按下鍵1，鍵2，鍵5時，我們發現同時會檢測到鍵4。這種現象被稱為 “幽靈鍵” 或 “鬼鍵”。 這是由於硬件設計中的缺陷。 你能指出它（的原因）嗎？