



Lab4 MAX7219 7-Seg LED

實驗四 MAX7219 7-Seg LED

1. Lab objectives 實驗目的

- Understand the principle of using MAX7219.
- Design the program of 7-Seg LED.
- 了解 MAX7219 使用原理。
- 設計 7-Seg LED 程式。

2. Lab principle 實驗原理

Please check the course material of lab4_note

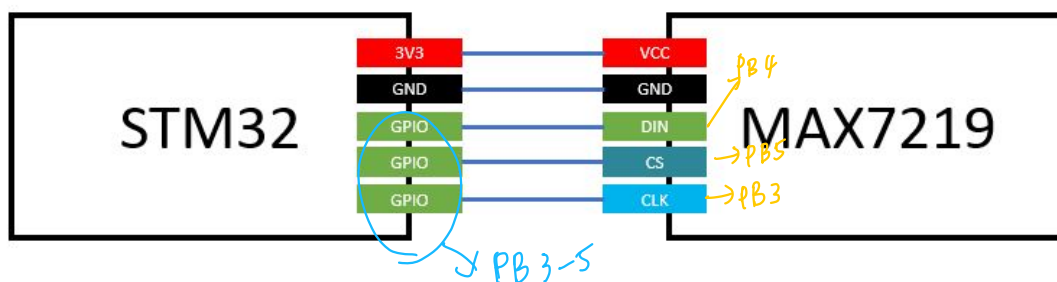
請參考 lab4_note 課程講義。

3. Steps 實驗步驟

3.1. Practice of Max7219 and 7-Seg LED with no-decode mode

Requirement: Please connect 3.3V, GND pins on STM32 to VCC, GND port on MAX7219, and pick up the GPIO pins on STM32 for DIN, CS and CLK on MAX7219.

將 stm32 的 3.3V 接到 7-Seg LED 板的 VCC，GND 接到 GND，並選擇三個 GPIO 接腳分別接到 DIN、CS 和 CLK。



Then, complete the code provided below and display 0, 1, 2, 3..., 9, A, b, C, d, E, F at the first digit of 7-Seg LED at 1 second interval. Example video link is given above.

接著，完成以下程式碼，並利用 GPIO 控制 Max7219 並在 7-Seg LED 上的第一位依序顯示 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, b, C, d, E, F (時間間隔1秒)，範例影片如下。

https://drive.google.com/file/d/1l3jPmqgVzfIKZtJ_ou2HL3Ek7x_hf522/view?usp=sharing



```
.syntax unified
.cpu cortex-m4
.thumb
.data
//TODO: put 0 to F 7-Seg LED pattern here
arr: .byte 0x0, 0x0, ...

.text
.global main
main:
    BL    GPIO_init
    BL    max7219_init
loop:
    BL    DisplayDigit
    B loop

GPIO_init:
//TODO: Initialize GPIO pins for max7219 DIN, CS and CLK
BX LR

DisplayDigit:
//TODO: Display 0 to F at first digit on 7-SEG LED.
BX LR

MAX7219Send:
//input parameter: r0 is ADDRESS , r1 is DATA
//TODO: Use this function to send a message to max7219
BX LR

max7219_init:
//TODO: Initialize max7219 registers
BX LR

Delay:
//TODO: Write a delay 1sec function
BX LR
```

3.2. Practice of Max7219 and 7-Seg LED with code B decode mode

Requirement: In this lab, you are asked to set Max7219 to code B decode mode. Then, please refer to the code provided below, please put your student ID in array student_id and show these numbers on 7-Seg LED. For example, the 7-Seg LED shows the pattern like the picture below in case that your student ID is 1234567. Note that the unused digits are set to blank.

在本實驗中，您被要求將 Max7219 設置為 **code B decode mode**。然後，請參考下面提供的代碼請參閱下面提供的代碼，請將您的學生 ID 放置在陣列 student_id 中，並將這些數字顯示在 7 段 LED 上。例如，如果您的學號是 1234567，則 7 段 LED 會顯示如下圖所示的圖案。沒使用到的 digits 被設成了空白。



```
.syntax unified
.cpu cortex-m4
.thumb

.data
//TODO: put your student id here
student_id: .byte 1, 2, 3, 4, 5, 6, 7

.text
.global main
main:
    BL    GPIO_init
    BL    max7219_init
    //TODO: display your student id on 7-Seg LED
Program_end:
    B Program_end

GPIO_init:
    //TODO: Initialize three GPIO pins as output for max7219 DIN,
    CS and CLK
    BX LR

MAX7219Send:
    //input parameter: r0 is ADDRESS , r1 is DATA
    //TODO: Use this function to send a message to max7219
    BX LR

max7219_init:
    //TODO: Initial max7219 registers.
    BX LR
```



3.3. Show the Fibonacci number

Requirement: Design a program to detect input signals of the user button on STM32. When the button is clicked N times, display the Nth fibonacci number on 7-Seg LED and set unused digits to blank. When the user button is held down for more than 1 second, reset the displayed number to 0. If the value is out of the range of display, show "9999 9999". Example video link is given above.

(Note: Please remember to deal with the bouncing problem.)

設計一個程式來檢測 STM32 上用戶按鈕的輸入信號。點擊按鈕 N 次後，在 7 段 LED 上顯示第 N 個斐波那契數並將沒使用到的位數設為空白。按住用戶按鈕 1 秒鐘以上時，將顯示的數字重置為 0。如果數值超出顯示範圍，請顯示“9999 9999”，範例影片如下。

(Note: 請記得處理 User button 開關彈跳的問題。)

Link to demo video:

https://drive.google.com/file/d/1AAH4FBO6Xma4eTWrEtsSFi9M8TP_oYAc/view?usp=sharing

udi

Fibonacci number:

The fibonacci number is defined as following,

$$F_0 = 0, F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}, n > 2$$

3.4. Question 實驗課問題

Question 1: What's the functions of DIN, CLK, CS pins on MAX7219 7-Seg LED?

MAX7219 7-Seg LED 上的 DIN, CLK, CS 腳位分別有什麼作用？
data → 當 CS 下降沿時，表示我要告訴 max7219 我的資料準備好了，它會把 internal shift register 中的資料 (最新的 16 bit) 去 decode address 然後送到正確的 register 存起來。
CLK → 每次 CLK 上升沿時，會把 DIN 1 bit 的資料 shift 進 internal shift register

Question 2: Each time we send a command to MAX7219, we need to encode our command into 2-bytes, Address, and Data. What are the functions of Address(D8~D15) and Data(D0~D7)?

每次向 MAX7219 發送命令時，都需要將命令編碼為 2 個字節，地址和數據。地址 (D8~D15) 和數據 (D0~D7) 的功能是什麼？

↓
L1 那個 register 裡面所存資料
告訴 max7219 我現在存的是什麼
data 是要存到哪個 register (decode mode, scan limit...)



3.5. Reference & Hint 參考資料與提示

Hint 1: If we disconnect GND between MAX 7219 7-Seg LED and our development board, what will be the voltage detected on pin VCC of MAX 7219 7-Seg LED.

如果我們斷開七段顯示器與開發版上的 GND 訊號線。我們在七段顯示器上的 VCC 腳位測到的電壓值會是多少？

Hint 2: In addition to GPIOx_ODR, we can also use GPIOx_BSRR_OFFSET and GPIOx_BRR_OFFSET to set the value which is the output value to STM32. Please explain the difference between these registers.

除了 GPIOx_ODR，我們也可以使用 GPIOx_BSRR_OFFSET 和 GPIOx_BRR_OFFSET 設置要輸出到 STM32 的值，請說明兩者之間的差異。

BSRR 只有後 16 位有效，對這 16 位的某位寫「1」，對應的 port 32 bit 高 16 bit 就設「1」，對應的 port 的對應 pin 就設「1」。
和 BRR 的前 16 bit 有相同效果。對這 16 位的某位寫「1」，對應的 port 32 bit 低 16 bit 就設「1」，對應的 port 的對應 pin 就設「1」。

Hint 3: STM32 does not directly communicate with the 7-segment LED, but uses the driving IC - Max7219. All operations on MAX7219 are triggered by 16-bit commands to update the values of its registers. Please explain the result after executing the following command.

STM32 不會直接溝通 7-Seg LED，而是利用驅動 IC - Max7219。MAX7219 的所有操作均由 16-bit 命令觸發，以更新其暫存器的值。請說明執行完以下命令後所產生的結果。

digit 2-5 for code B decode

D15-12	D11-8	D7-0	備註
xxxx	1001 <i>decode mode</i>	0011 1100	
xxxx	1111 <i>display test</i>	xxxx xxx1 <i>全亮</i>	
xxxx	1011 <i>scan limit</i>	xxxx x101 <i>display digit 0-5</i>	
xxxx	0110 <i>digit 5</i>	0000 1110 <i>P</i>	with Code B
xxxx	0110 <i>digits</i>	0000 1110 <i>L</i>	without Code B

(x: don't care)