實驗四 STM32 GPIO System

0411306 范鑫

1. Lab objectives 實驗目的

了解STM32基本輸出入I/O port使用原理

設計簡易LED跑馬燈程式

了解按鈕與指撥開關使用原理

*2.*實驗原理

第一個實驗主要是瞭解GPIO的基本設定

利用

.equ RCC_AHB2ENR, 0x4002104C

```
.equ GPIOB_MODER, 0x48000400
.equ GPIOB_OTYPER, 0x48000404
.equ GPIOB_OSPEEDR, 0x48000408
.equ GPIOB_PUPDR, 0x4800040C
.equ GPIOB_ODR, 0x48000414

.equ GPIOC_MODER, 0x48000800
.equ GPIOC_IDR,  0x48000810
```

來決定GPIOx的位址設定 然後主要是進行ＧＰＩＯ的初始：

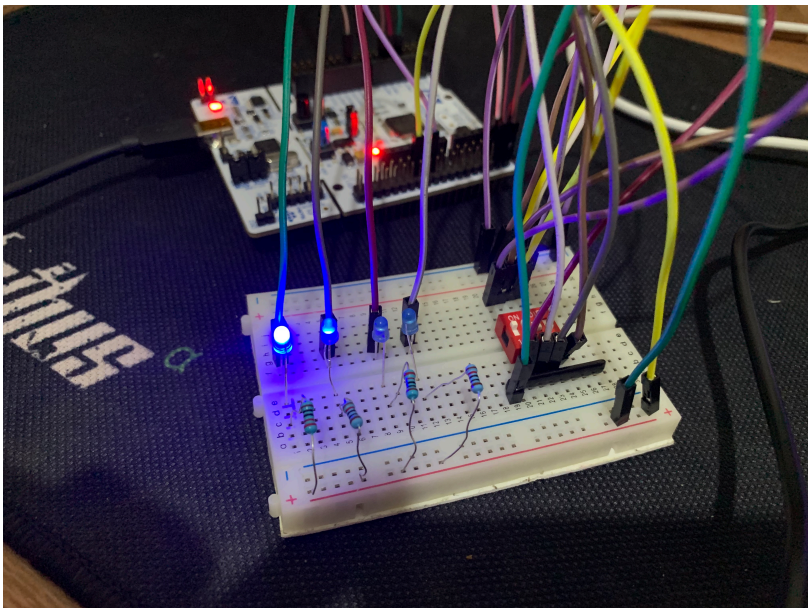
將他們的moder設定為我們所需要的input或output

```
GPIO_init:
//TODO: Initial LED GPIO pins as output
    movs r0, #0x6
    ldr r1, =RCC_AHB2ENR
    str r0, [r1]

    movs r0, #0x1540
    ldr  r1, =GPIOB_MODER
```

```
    ldr  r2, [r1]
    and  r2, #0xFFFFC03F//這邊使用pin 3,4,5,6
    orrs r2, r2, r0
    str  r2, [r1]
    //-----------
    ldr r1, =GPIOC_MODER
    ldr r0,[r1]
    ldr r2, =#0xF3FFFFFF
    and r0,r2
    str r0,[r1]
//---------------
    movs r0, #0x2A80
    ldr  r1, =GPIOB_OSPEEDR
    str  r0, [r1]

    ldr  r8, =GPIOB_ODR
    movs r1 , #(1<<3)
    str  r1,[r8]
    BX LR
```



如何解決按鍵彈跳：如果讀到使用者按鈕 則進入一段delay不進行接收按鍵訊號,則可以避免按鍵彈跳.

3. *Results and analysis* 實驗結果與分析

可以執行跑馬燈和密碼鎖,並且解決了按鍵彈跳

4.Conclusions and ideas 心得討論與應用聯想

這次實驗讓我們第一次的使用到了ＬＥＤ之類的硬體,讓我們更瞭解ＧＰＩＯ的輸出和輸入原理,是一次很棒的實驗,一開始著手很困難 ,但瞭解原理後就上手多了.

5. Code

1.

```
.syntax unified
.cpu cortex-m4
.thumb
.data
    leds: .long 0
    RL:.byte 1
    X: .long 1000
    Y: .long 500
    button:.byte 0



.text
    .global main
    .equ RCC_AHB2ENR, 0x4002104C
    .equ GPIOB_MODER, 0x48000400
    .equ GPIOB_OTYPER, 0x48000404
    .equ GPIOB_OSPEEDR, 0x48000408
    .equ GPIOB_PUPDR, 0x4800040C
    .equ GPIOB_ODR, 0x48000414
```

```
    .equ GPIOC_MODER, 0x48000800
    .equ GPIOC_IDR,  0x48000810

main:

    BL GPIO_init

    MOVS R1, #1
    LDR R0,=leds


    LSL r1,#3
    add r1,#4
    str r1,[r0]



Loop:
//TODO: Write the display pattern into leds
variable
    BL DisplayLED

    ldr r0,=leds
    ldr r1,[r0]

    ldr r0,=RL
    ldrb r5,[r0]

    cmp r1 ,#12
    beq turn_left
    cmp r1 ,#192
    beq turn_right

    B Loop2
turn_right:
    movs r5,#0//right
```

```
    B Loop2

turn_left:
    movs r5,#1//left

Loop2:
    cmp r5,#1
    beq left

right:
    LSR r1,#1
    b Loop3
//--------------------
left:
    LSL r1,#1

//--------------------

Loop3:


    ldr r0,=RL
    strb r5,[r0]

    ldr r0,=leds
    str r1,[r0]
    BL Delay
```

```
    B Loop


GPIO_init:
//TODO: Initial LED GPIO pins as output
    movs r0, #0x6
    ldr r1, =RCC_AHB2ENR
    str r0, [r1]

    movs r0, #0x1540
    ldr  r1, =GPIOB_MODER
    ldr  r2, [r1]
    and  r2, #0xFFFFC03F//3,4,5,6
    orrs r2, r2, r0
    str  r2, [r1]
    //-----------
    ldr r1, =GPIOC_MODER
    ldr r0,[r1]
    ldr r2, =#0xF3FFFFFF
    and r0,r2
    str r0,[r1]
//---------------
    movs r0, #0x2A80
    ldr  r1, =GPIOB_OSPEEDR
    str  r0, [r1]

    ldr  r8, =GPIOB_ODR
    movs r1 , #(1<<3)
    str  r1,[r8]
    BX LR


DisplayLED:
//TODO: Display LED by leds
    //LDR r0,=leds
    ldr r0,=leds
```

```
    ldr r3,[r0]
    eor r3 , -1

    ldr r0,=GPIOB_ODR

    //str r3,[r8]
    str r3, [r0]
    BX LR


Delay:
//TODO: Write a delay 1 sec function
    ldr  r3, =X
    ldr  r4, =Y
    movs r5,#1
    lsl r5,#13
    ldr  r0, =GPIOC_IDR
    ldr r6, [r3]
    L1:
        ldr r7, [r4]

    L2:
        ldr r1, [r0]
        ands r1,r5
        beq Stop
        subs r7, #1
        bne  L2
        subs r6, #1
        bne  L1

    BX LR


Stop:
        movs r3,#500
    L1111:
```

```
        ldr r7,[r4]
    L2222:
        subs r7, #1
        bne  L2222
        subs r3, #1
        bne  L1111
wait:
        ldr r1, [r0]
        ands r1,r5
        beq Stop2
        b wait

Stop2:
        movs r3,#500
    L11111:
        ldr r7,[r4]
    L22222:
        subs r7, #1
        bne  L22222
        subs r3, #1
        bne  L11111
    ldr r7,[r4]
    b L2
```

2.

```
.syntax unified
.cpu cortex-m4
.thumb
.data
    password: .byte 0xC
    X: .long 250
    Y: .long 1000
    temp: .long 0xF
    led: .long 0x78
```

```
.text
      .global main
      .equ RCC_AHB2ENR, 0x4002104C
      .equ GPIOB_MODER, 0x48000400
      .equ GPIOB_OTYPER, 0x48000404
      .equ GPIOB_OSPEEDR, 0x48000408
      .equ GPIOB_PUPDR, 0x4800040C
      .equ GPIOB_ODR, 0x48000414

      .equ GPIOC_MODER, 0x48000800
      .equ GPIOC_IDR,  0x48000810

main:
      BL GPIO_init

wait:
      ldr  r3, =X
      ldr  r4, =Y
      movs r5,#0
      //lsl r5,#13
      ldr  r0, =GPIOC_IDR

      ldr r1, [r0]
      lsr r1, #13
      cmp r1,r5
      beq Stop2
      b wait

Stop2:
            ldr r6,[r3]
      L11111:
            ldr r7,[r4]
      L22222:
            subs r7, #1
            bne  L22222
            subs r6, #1
            bne  L11111
pass:
      ldr r0,=temp
      ldr r1,[r0]

      ldr r0,=password
      ldr r2,[r0]
```

```
      ldr r0,=GPIOC_IDR
      ldr r3,[r0]

      ands r2,r1
      ands r3,r1

      cmp r2,r3
      beq Yes

No:
      bl Display1
      bl Delay
      bl Display2

      b wait

Yes:
      bl Display1
      bl Delay
      bl Display2
      bl Delay
      bl Display1
      bl Delay
      bl Display2
      bl Delay
      bl Display1
      bl Delay
      bl Display2

      b wait




Display1:
      ldr r1,=GPIOB_ODR

      movs r0,#0
      str r0,[r1]
      bx lr

Display2:
      ldr r1,=GPIOB_ODR
      ldr r0,=led
      ldr r8,[r0]
```

```
      //movs r0,#(1<<5)
      str r8,[r1]
      bx lr



GPIO_init:
//TODO: Initial LED GPIO pins as output
      movs r0, #0x6
      ldr r1, =RCC_AHB2ENR
      str r0, [r1]

      movs r0, #0x1540
      ldr  r1, =GPIOB_MODER
      ldr  r2, [r1]
      and  r2, #0xFFFFC03F//3,4,5,6
      orrs r2, r2, r0
      str  r2, [r1]
      //-----------
      ldr r1, =GPIOC_MODER
      ldr r0,[r1]
      ldr r2, =#0xF3FFFF00
      and r0,r2
      str r0,[r1]
//---------------
      movs r0, #0x2A80
      ldr  r1, =GPIOB_OSPEEDR
      str  r0, [r1]

      ldr r1,=GPIOB_ODR
      ldr r0,=led

      BX LR

Delay:
      ldr  r3, =X
      ldr  r4, =Y
      ldr r6, [r3]
      L1:
            ldr r7, [r4]

      L2:
            subs r7, #1
            bne  L2
            subs r6, #1
            bne  L1
```

```
BX LR
```