

Lab5 7-seg LED

0516039 羅文笙

1. Lab objectives 實驗目的

Understand the principle of using MAX7219.

Design the program of 7-Seg LED.

2. Steps 實驗步驟

- 2.1 在輸出0~9的時候用decode mode,在輸出A~F的時候用 no decode,此外再加一個loop就可以一直顯示了
- 2.2將數字一個個輸出到不同digit上就可以了
- 2.3 這個是要弄費氏數列,這個比較麻煩的點在於要把數字拆開一個一個讓他顯示在上面和長按後將數字重置。對於數字來說我先把數字算出來之後再一個digit求。以123這個數字為例,先將123除以100,得到1,再將1乘以100得到100,用100去減掉123得到23,然後將1顯示在digit2,接著將23除以10得到2,2乘以10得到20,將23-20得到3,最後將2顯示在digit1,而3顯示在digit0。另外,長按重置的部分,我的方式是,先偵測有沒有按下按鈕,接著如果有按的話,就進入判斷是短按或長按,短按就顯示下一個fib number,若是長按一秒以上的話,就歸零並回到main,重新開始。

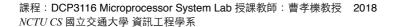


3.Results and analysis 實驗結果與分析

所有的東西都算是簡單,只要將如何把訊息傳進去max7219 裡面的這個部分寫好的話,其實其他的功能都是以前作業 的概念了。

```
300 MAX7219Send:
302//input parameter: r0 is ADDRESS(4-bit,d11~d8) , r1 is DATA(8-bit,d7~d0)
303//TODO: Use this function to send a message to max7219
305
      //make r5 be the serial number which is used to send into max7219
306
      mov r5,r0
307
      lsl r5,#8
308
      add r5,r1
309
310
    //r6 for cnt send_inner_loop,r7 是odr's address, r8 是用來存進odr的值
311
     //odr: odr[5]: clk, odr[4]: cs, odr[3]: din
     movs r6,#0//initialize
312
      //先送4次零來補d15~d12
313
314
      ldr r10,=GPIOB_BSRR
315
      ldr r11,=GPIOB_BRR
316
      ldr r7, =GPIOB_ODR
317
318
      //din=0
319
     mov r8,#(1<<3)
      str r8.[r11]
320
3 328
            mov r8, \#(1 << 5)
  329
            str r8,[r10]
  330
            //clk = 0
  331
            mov r8, \#(1 << 5)
332
            str r8,[r11]
  333
            //clk = 1
  334
            mov r8, \#(1 << 5)
  335
            str r8,[r10]
  336
            //clk = 0
  337
            mov r8, \#(1 << 5)
  338
            str r8,[r11]
  339
            //clk = 1
  340
            mov r8, \#(1 << 5)
  341
            str r8,[r10]
  342
            //clk = 0
  343
            mov r8, \#(1 << 5)
  344
            str r8,[r11]
  345
            //clk = 1
            mov r8, \#(1 << 5)
  346
   347
            str r8,[r10]
  348
            //clk = 0
            mov r8,#(1<<5)
   349
  350
            str r8,[r11]
  351
            //r9 is shift cnt
   352
            mov r9,#11
```

353





```
354 send_inner_loop:
355//r6 for cnt, r8 用來存r5 bitAnd 1 完的結果並放進odr!, r9 is shift cnt
356//r10 = bsrr,r11 = brr(bsrr low 16-bit 如果是1的話飲改1·brr如果是1的話改0)
357
      push {lr}
358
       mov r8, r5
      lsr r8,r9
359
360
       and r8,#1
361
       //if r8==1, 先讓din改變,再讓clk=1,最後變成odr = 10100, else odr = 10000
362
363
       cmp r8,#1
364
       it ea
365
       bleq din_1
366
       cmp r8,#0
367
       it eq
368
       bleg din_0
369
370
       //cnt++
371
      adds r6,#1
       //if cnt < 12, keep doing send_inner_loop, else cs = 1,to letch the d15-d0
372
      cmp r6,#12
373
374
      itt lt
375
       sublt r9,#1
376
       blt send_inner_loop
377
       //set cs = 1,to letch the d15-d0
378
     mov r8,#(1<<4)
379
     str r8,[r10]
      //reset cs = 0
mov r8,#(1<<4)
380
381
```

```
382 str r8,[r11]
383 //
384 pop {pc}
```

4.Conclusions and ideas 心得討論與應用聯想

這次是我們第一次在資工系上接觸到這種顯示器的東西, 我覺得其實還算滿有趣的,至少有東西可以顯示出來,有 種成就感,至少不像之前那種都只是電腦上面的