

微算機原理及應用

單元五：8051的輸入與輸出埠控制

授課老師：林淵翔 老師

大綱

- 8051輸入輸出埠的特性和功能介紹(8051 I/O port)
- 8051輸入輸出埠的控制方法(8051 I/O port control)
- 8051的C語言程式設計方法(8051 programming in C)

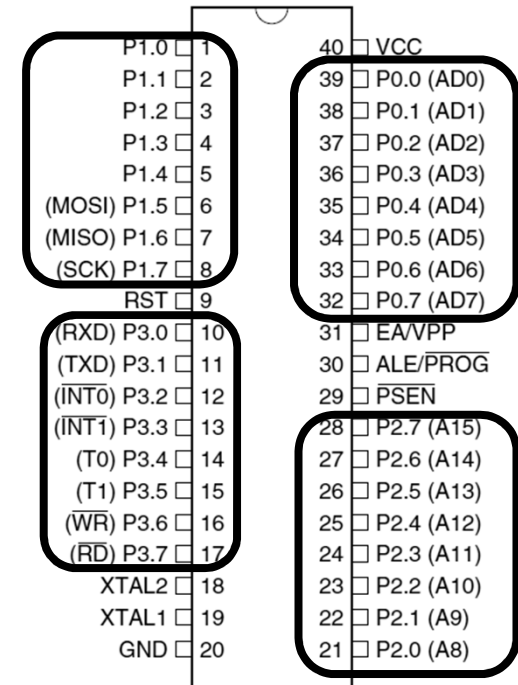
大綱

- 範例一: 跑馬燈實驗
- 範例二: 七段顯示器控制實驗
- 範例三: 開關輸入實驗
- 範例四: 按鈕控制實驗
- 參考文獻

單元五
8051的輸入與輸出埠控制
PART A

5.1 8051 I/O port

- 總共有四個埠(port)可以做輸入輸出埠控制(I/O control)
 - P0, P1, P2, and P3
- 每個埠有8隻腳，共有32隻腳(32 pins)



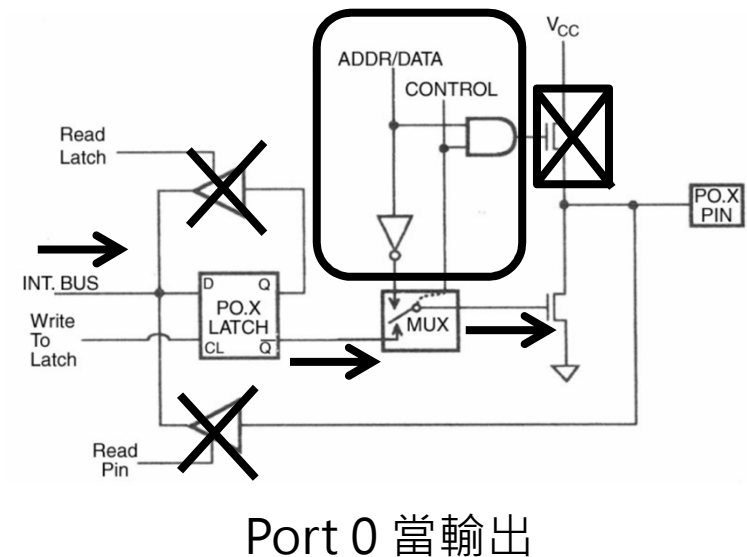
5.1 8051 I/O port (continued)

- 在 **RESET** 之後，所有埠都被設為 **inputs**，可以用來做輸入埠使用。
- 輸出之後，要重新設定**I/O port**為輸入埠，必須將 **1** 送到該埠，把輸出端的**output driver FET** 關掉。

5.1 8051 I/O port (continued)

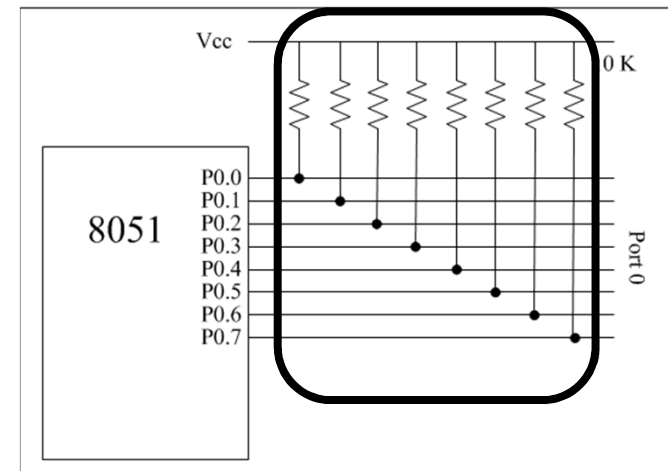
- **Port 0**

- Port 0 共有8隻腳(pins 32~39)。
- 可以用來作為input/output 或 address 和 data。
- 要使用這個埠必須外接提升電阻 (pull-up resistor)。因為 P0 是 open drain。



5.1 8051 I/O port (continued)

- Port 0
 - Port 0 共有8隻腳(pins 32~39)。
 - 可以用來作為input/output 或 address 和 data。
 - 要使用這個埠必須外接提升電阻 (pull-up resistor)。因為 P0 是 open drain。



5.1 8051 I/O port (continued)

- Port 0 當輸出

;Toggle all bits of P0

BACK: MOV A,#0AAH

MOV P0,A

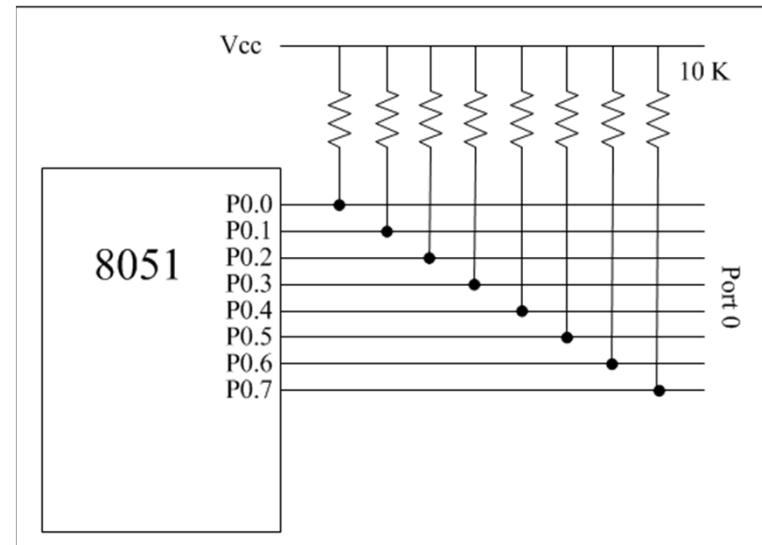
ACALL DELAY

MOV A,#055H

MOV P0,A

ACALL DELAY

SJMP BACK



5.1 8051 I/O port (continued)

• Port 0 當輸入

- 整個埠要當成input，則整個port要寫1 (writing 1)到每一個位元。

;Get a byte from P0 and send it to P1

```
MOV A,#0FFH ;A = FF hex
```

```
MOV P0,A ;make P0 an input port
```

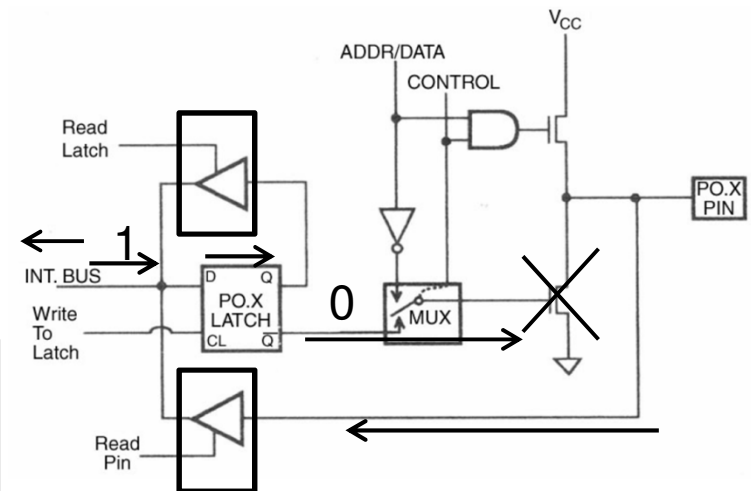
;by writing all 1s to it

BACK:

```
MOV A,P0 ;get data from P0
```

```
MOV P1,A ;send it to port 1
```

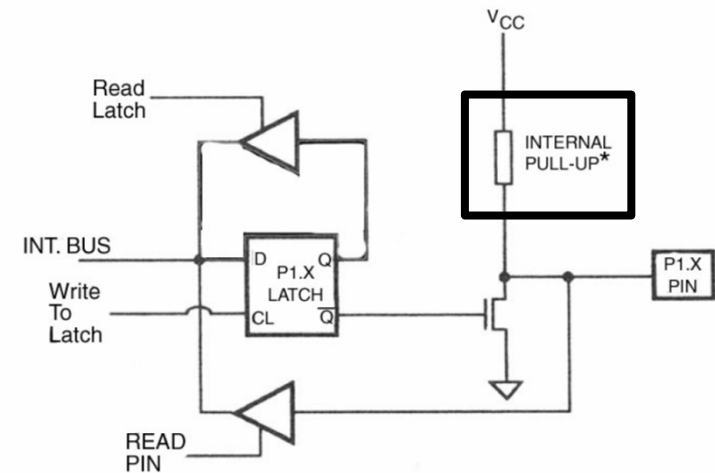
```
SJMP BACK ;keep doing it
```



5.1 8051 I/O port (continued)

- **Port 1**

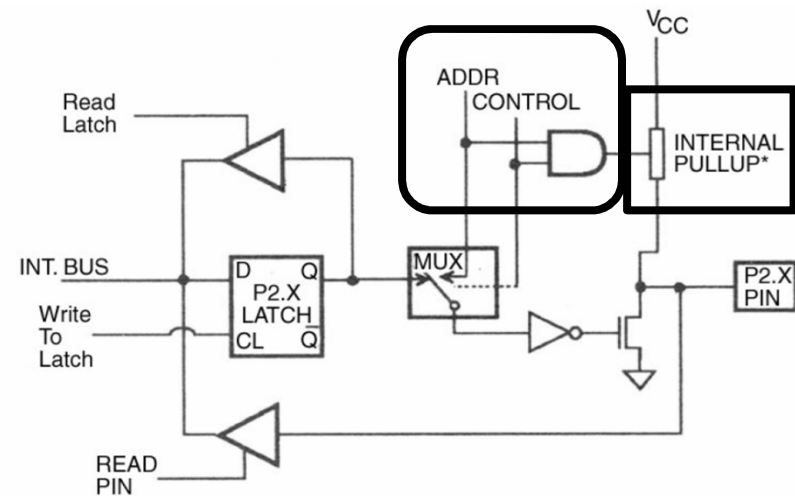
- Port 1 共有8隻腳(pins 1~8)。
- 只用來作為input/output。
- 相對於P0，P1 不需要外接提升電阻(pull-up resistors)。因為 P1 是內建提升電阻(pull-up resistors internally)。



5.1 8051 I/O port (continued)

• Port 2

- Port 2共有8隻腳(pins 21~28)
- 可以用來作為input/output 或 address。
- 就像P1，P2 不需要外接提升電阻 (pull-up resistors)。因為 P2 是內建提升電阻 (pull-up resistors internally)。



5.1 8051 I/O port (continued)

- **Port 3**

- **Port 3**共有8隻腳(pins 10~17)。
- 可以用來作為input/output 或 **alternate output function**。
- 就像P1, P2, P3 不需要外接提升電阻(pull-up resistors)。因為 **P3 是內建提升電阻(pull-up resistors internally)**

P3 Bit	Function	Pin
P3.0	RxD	10
P3.1	TxD	11
P3.2		12
P3.3		13
P3.4	T0	14
P3.5	T1	15
P3.6		16
P3.7		17

單元五
8051的輸入與輸出埠控制
PART B

5.2 I/O port 控制

- **Output**

```
MOV A,#0AAH
```

```
MOV P0,A
```

- **Input**

```
MOV A,#0FFH ;A = FF hex
```

```
MOV P0,A      ;make P0 an input port  
              ;by writing all 1s to P0
```

```
MOV A,P0      ;get data from P0
```

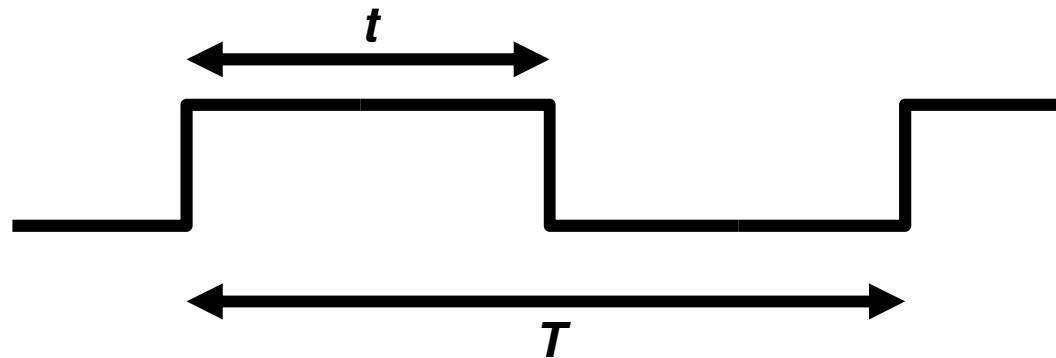
5.2 I/O port 控制 (continued)

- I/O control
- Bit-addressability
 - SETB P1.0
 - CLR P1.1

Mnemonic	Example
ANL Px	ANL P1, A
ORL Px	ORL P2, A
XRL Px	XRL P0, A
JBC PX.Y, TARGET	JBC P1.1, TARGET
CPL PX.Y	CPL P1.2
INC Px	INC P1
DEC Px	DEC P2
DJNZ PX.Y, TARGET	DJNZ P1, TARGET
MOV PX.Y, C	MOV P1.2, C
CLR PX.Y	CLR P2.3
SETB PX.Y	SETB P2.3
<i>Note: x is 0, 1, 2, or 3 for P0 – P3.</i>	

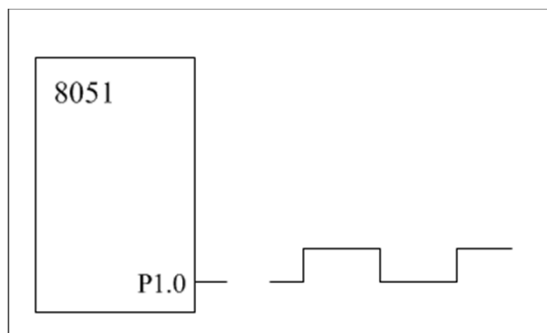
5.2 I/O port 控制 (continued)

- 工作週期(duty cycle)



5.2 I/O port 控制 (continued)

- 在Port 1的bit 0(P1.0)建立一個工作週期(duty cycle)為50%的方波

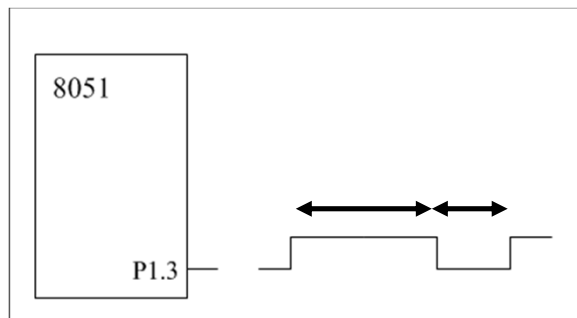


```
HERE:  SETB P1.0  ← ← ;set to high bit 0 of port 1
        LCALL DELAY ← ← ;call the delay subroutine
        CLR P1.0  ←   ;P1.0 = 0
        LCALL DELAY ←
        SJMP HERE ←   ;keep doing it
```

```
HERE:  CPL P1.0  ← ← ← ;complement bit 0 of port 1
        LCALL DELAY ← ← ← ;call the delay subroutine
        SJMP HERE ← ← ;keep doing it
```

5.2 I/O port 控制 (continued)

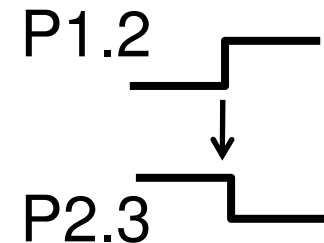
- 在Port 1的bit 3(P1.3)建立一個工作週期(duty cycle)為66.6%的方波



```
BACK:  SETB P1.3      ;set port 1 bit 3 high
        LCALL DELAY   ;call the delay subroutine
        LCALL DELAY   ;call the delay subroutine again
        CLR P1.3      ;clear bit 2 of port 1(P1.3=low)
        LCALL DELAY   ;call the delay subroutine
        SJMP BACK     ;keep doing it
```

5.2 I/O port 控制 (continued)

- 持續監控 P1.2 直到他變成 high
- 當 P1.2變成 high ，送 45H 到 port 0 並送一個 high-to-low (H-to-L) pulse 到 P2.3



	SETB P1.2	←	;set P1.2 an input
	MOV A,#45H	←	;A = 45H
HERE:	JNB P1.2,HERE	←	;get out when P1.2 = 1
	MOV P0,A	←	;issue A to P0
	SETB P2.3	←	;make P2.3 high
	CLR P2.3	←	;make P2.3 low for H-to-L

單元五
8051的輸入與輸出埠控制
PART C

5.3 8051 C語言程式設計

- **Unsigned char**

- 是一個 8-bit 無號數的資料型態(data type)，數值表示範圍為 0 - 255 (00 - FFH)。

- 寫一個 8051 C 語言程式把 00 - FF 送到 P1。

```
#include <REGX51.H>  ←  
void main(void)  ←  
{  ←  
    unsigned char z;  ←  
    for(z=0;z<=255;z++)  ←  
        P1 = z;  ←  
}  ←
```

5.3 8051 C語言程式設計 (continued)

- 8051 C語言程式常用的資料型態(data type)

	Data Type	Size in Bits	Data Range/Usage
→	unsigned char	8-bit	0 to 255
→	(signed) char	8-bit	-128 to +127
→	unsigned int	16-bit	0 to 65535
→	(signed) int	16-bit	-32,768 to +32,767
→	sbit	1-bit	SFR bit-addressable only
→	bit	1-bit	RAM bit-addressable only
→	sfr	8-bit	RAM addresses 80 – FFH only

REGX51.H

```
/*-----  
AT89X51.H
```

```
Header file for the low voltage Flash Atmel AT89C51 and AT89LV51.  
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.  
All rights reserved.
```

```
-----*/
```

```
#ifndef __AT89X51_H__  
#define __AT89X51_H__
```

```
/*-----  
Byte Registers
```

```
-----*/
```

```
sfr P0    = 0x80;  
sfr SP    = 0x81;  
sfr DPL   = 0x82;  
sfr DPH   = 0x83;  
sfr PCON  = 0x87;  
sfr TCON  = 0x88;  
sfr TMOD  = 0x89;  
sfr TL0   = 0x8A;  
sfr TL1   = 0x8B;
```

```
sfr TH0    = 0x8C;  
sfr TH1    = 0x8D;  
sfr P1     = 0x90;  
sfr SCON   = 0x98;  
sfr SBUF   = 0x99;  
sfr P2     = 0xA0;  
sfr IE     = 0xA8;  
sfr P3     = 0xB0;  
sfr IP     = 0xB8;  
sfr PSW    = 0xD0;  
sfr ACC    = 0xE0;  
sfr B      = 0xF0;
```

```
/*-----  
P0 Bit Registers
```

```
-----*/
```

```
sbit P0_0 = 0x80;  
sbit P0_1 = 0x81;  
sbit P0_2 = 0x82;  
sbit P0_3 = 0x83;  
sbit P0_4 = 0x84;  
sbit P0_5 = 0x85;  
sbit P0_6 = 0x86;  
sbit P0_7 = 0x87;
```


5.3 8051 C語言程式設計 (continued)

- Time delay
 - 使用“for” 迴圈
 - 使用8051 timers

寫一個 8051 C 語言程式把P1的每個位元做 toggle ，並加上 time delay 。

```
#include <REGX51.H>
void main(void) ←
{ ←
    unsigned int x; ←
    for(;;) ← //report forever
    {
        P1 = 0x55; ←
        for(x=0;x<4000;x++); ← //delay size unknown
        P1 = 0xAA; ←
        for(x=0;x<4000;x++); ←
    }
} ←
```

5.3 8051 C語言程式設計 (continued)

- 可位元定址(Bit-addressable)的 I/O 控制

寫一個 8051 C 語言程式去監控 bit P1.5，如果是 high，送55H到P0；否則送 AAH 到 P2。

```
#include <REGX51.H>
sbit mybit = P1^5;  ← //notice the way single bit is declared
void main(void) ←
{ ←
    mybit = 1; ← // configure P1.5 as input
    while (1) ← {
        if (mybit == 1) ←
            P0 = 0x55; ←
        else ←
            P2 = 0xAA; ←
    }
} ←
```

5.3 8051 C語言程式設計 (continued)

- **Checksum**

- 1. 將所有位元組加起來，並捨去進位部分
- 2. 把加起來的和(sum)取 2's 補數(complement)
- 3. 這就是 **checksum byte**，是這一串數列的最後一個 byte

5.3 8051 C語言程式設計 (continued)

寫一個 8051 C 語言程式去計算 checksum byte 。

```
#include <REGX51.H>
void main(void) ←
{ ←
    unsigned char mydata[] = {0x25,0x62,0x3F,0x52}; ←
    unsigned char sum = 0; ←
    unsigned char x; ←
    unsigned char checksumbyte; ←
    for(x=0;x<4;x++) ←
    {
        P2 = mydata[x]; ←           //issue each byte to P2
        sum = sum + mydata[x]; ←    //add them together
        P1 = sum; ←                 //issue the sum to P1
    }
    checksumbyte = ~sum+1; ←        //make 2's complement
    P1 = checksumbyte; ←           //show the checksum byte
} ←
```

5.3 8051 C語言程式設計 (continued)

- 二進制【Binary (hex)】
對 十進制【decimal】
和 ASCII 的轉換

Key	ASCII (hex)	Binary	BCD (unpacked)
0	30	0011 0000	0000 0000
1	31	0011 0001	0000 0001
2	32	0011 0010	0000 0010
3	33	0011 0011	0000 0011
4	34	0011 0100	0000 0100
5	35	0011 0101	0000 0101
6	36	0011 0110	0000 0110
7	37	0011 0111	0000 0111
8	38	0011 1000	0000 1000
9	39	0011 1001	0000 1001

5.3 8051 C語言程式設計 (continued)

寫一個 8051 C 語言程式去轉換 11111110(FE hex) 到十進制(decimal) 並顯示數字(digits)在 P0, P1, and P2. (FEH→254)

```
#include <REGX51.H>
```

```
void main(void)
```

```
{
```

```
    unsigned char x, binbyte, d1, d2, d3;
```

```
    binbyte = 0xFE; ←
```

```
    x = binbyte / 10; ←
```

```
    d1 = binbyte % 10; ←
```

```
    d2 = x % 10; ←
```

```
    d3 = x / 10; ←
```

```
    P0 = d1; ←
```

```
    P1 = d2; ←
```

```
    P2 = d3; ←
```

```
}
```

```
//binary(hex) byte
```

```
//divide by 10, x = 25
```

```
//find remainder (LSD), d1 = 4
```

```
//middle digit, d2 = 5
```

```
//most significant digit (MSD), d3 = 2
```

單元五
8051的輸入與輸出埠控制
PART D

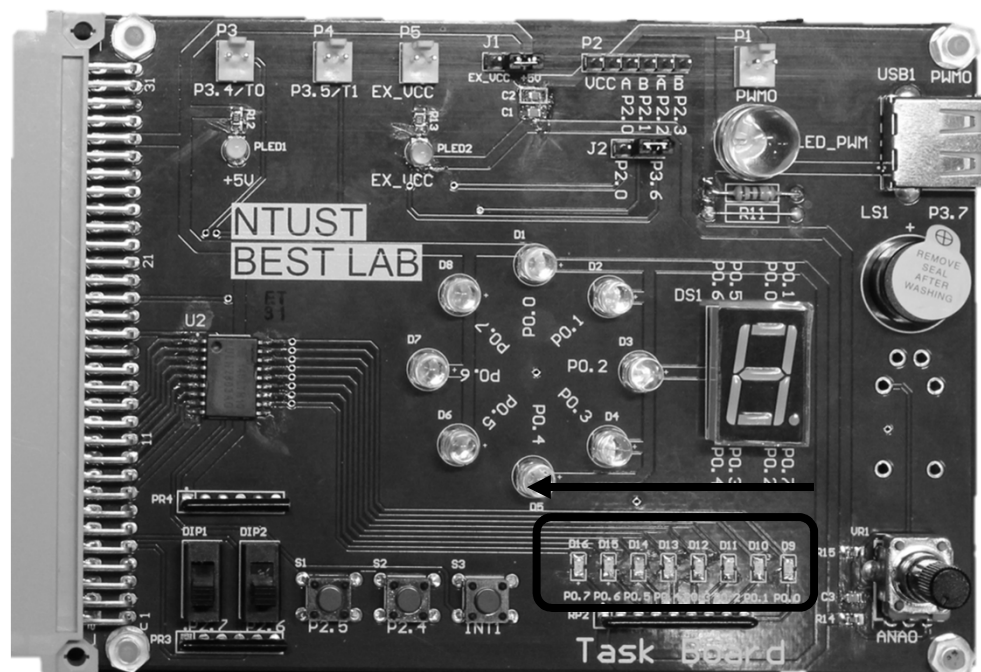
微算機原理及應用實習

範例一：跑馬燈實驗

範例說明

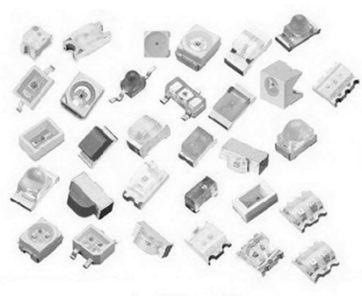
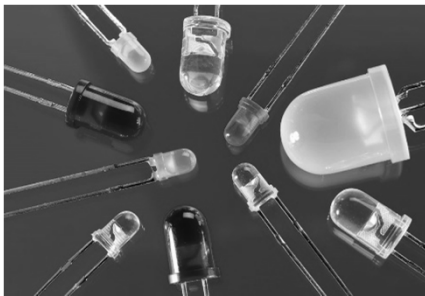
- 實驗目的：
瞭解**8051**之輸出控制方法，並練習搬移與移位旋轉指令。
- 功能說明：
由**AT89S51**的**Port0**輸出控制**8**個**LED**，從**D9(LSB)**開始依序往左邊點亮，每次亮一顆，即向左旋轉。

IO應用電路板(Task board 1)



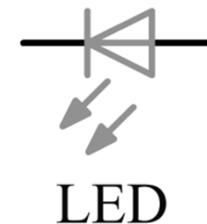
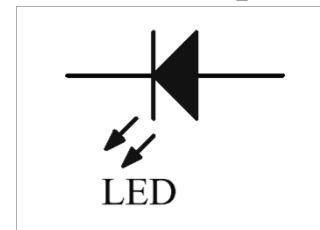
元件介紹

- 各種LED外觀



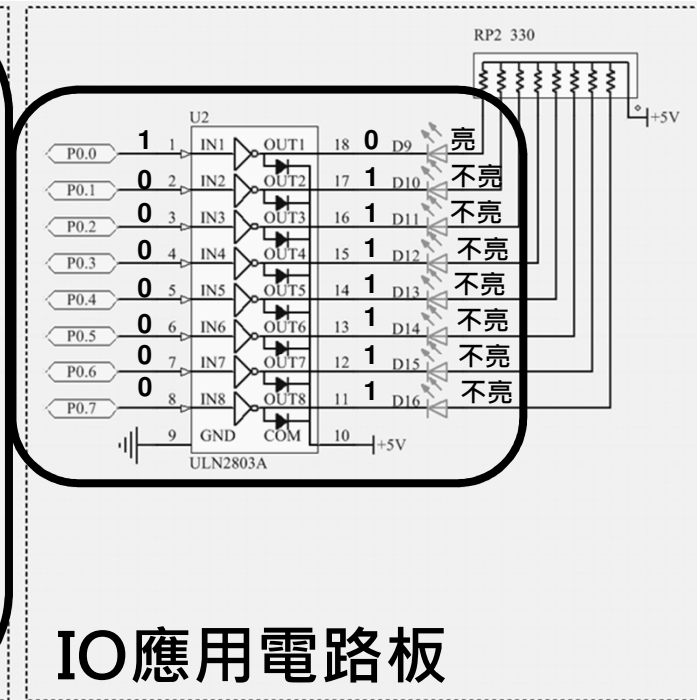
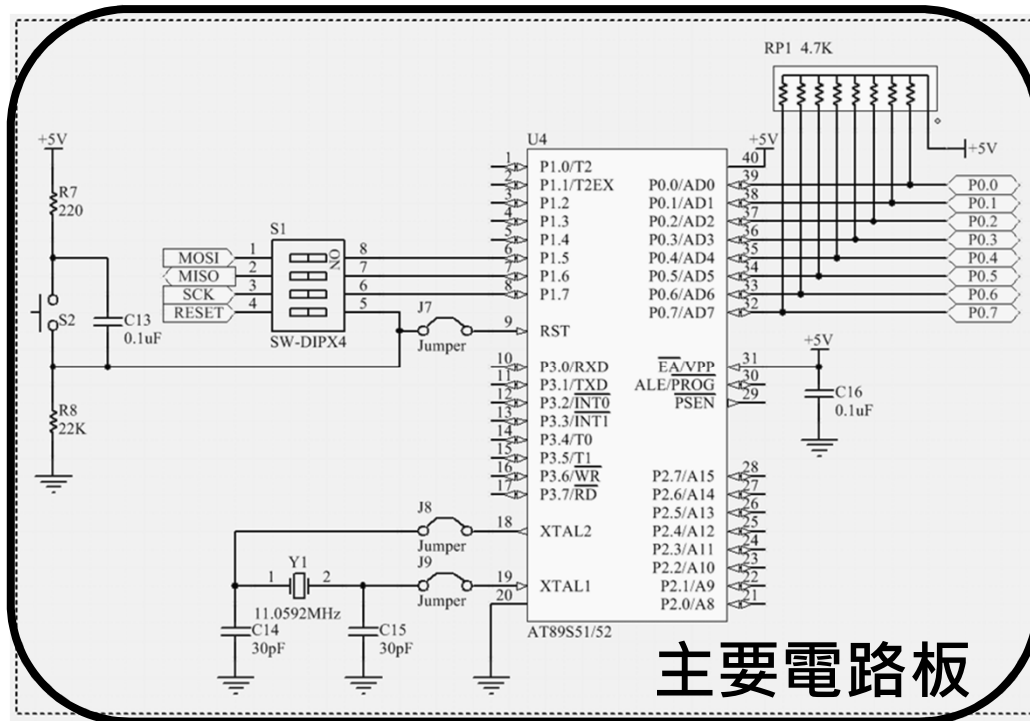
- LED符號

N P

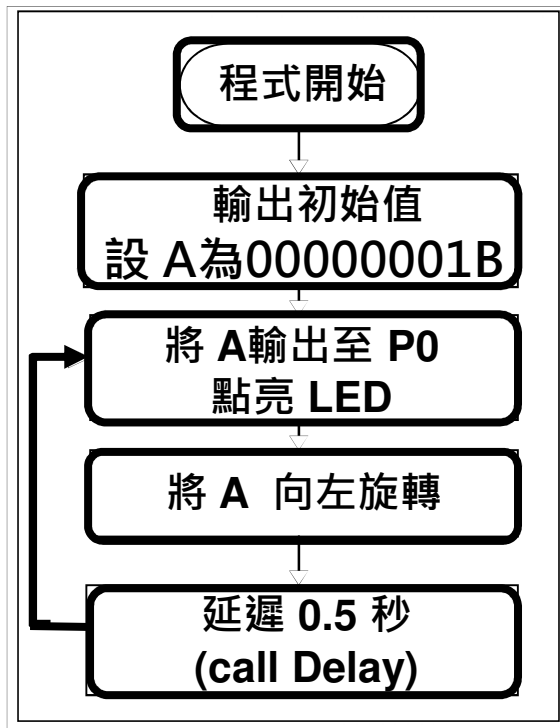


資料來源: <http://intl-lighttech.com/> and LEDKE.COM

電路圖



程式碼與流程圖(組合語言)



	ORG	0	←	
START:	MOV	A,#00000001B	←	;輸出初始值
LOOP:	MOV	P0,A	←	;將A輸出至P0
	RL	A	←	;將資料向左轉
	CALL	DELAY	←	;延遲0.5秒
	JMP	LOOP	←	
DELAY:	MOV	R5,#5		;延遲副程式
DL1:	MOV	R6,#200		
DL2:	MOV	R7,#230		
	DJNZ	R7,\$		
	DJNZ	R6,DL2		
	DJNZ	R5,DL1		
	RET			
	END			

程式碼與流程圖(C語言)

```
→ #include <REGX51.H>
void Delay_ms(int tx)          //延遲副程式
{
    char ti;
    while(tx--)    for(ti=0;ti<101;ti++);
}
→ main(void)
{
→   P0 = 0x01;                //輸出初始值
→   while(1)
→   {
→       char P0_MSB = P0>>7;    //存入MSB
→       Delay_ms(500);          //延遲 0.5秒
→       P0 <<= 1;                //將資料左移
→       P0 |= P0_MSB;            //放入原始MSB到LSB
→   }
}
```

練習題

- **功能說明：**

由AT89S51的Port0輸出控制8個LED，從D9(LSB)開始依序往右邊點亮，每次亮一顆，即向右旋轉。

單元五
8051的輸入與輸出埠控制
PART E

微算機原理及應用實習

範例二：七段顯示器控制實驗

範例說明

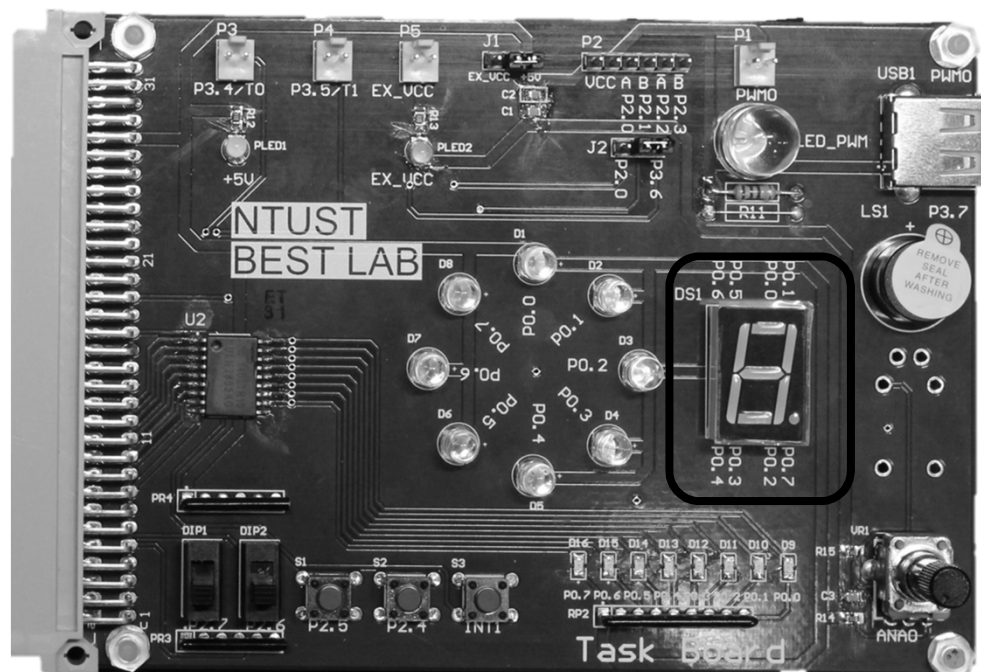
- **實驗目的：**

認識七段顯示器元件及其顯示方法。

- **功能說明：**

由AT89S51的Port0控制七段顯示器輸出，使七段顯示器由0計數到9並循環。

IO應用電路板(Task board 1)

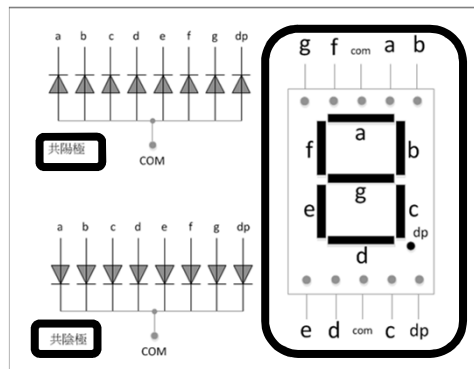


元件介紹

- 各種七段顯示器外觀



- 七段顯示器接腳



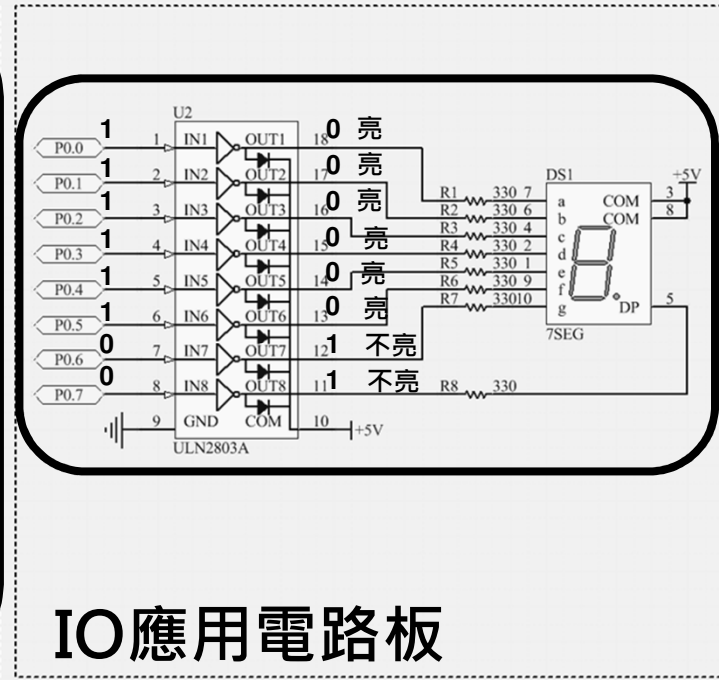
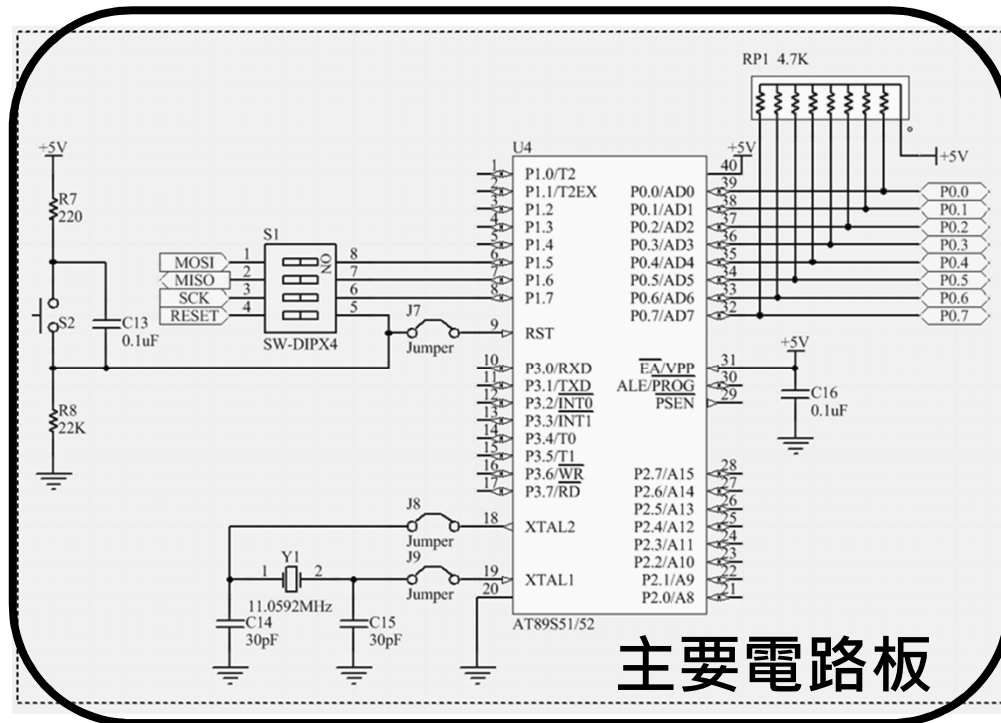
- 七段顯示器顯示數值

輸入				輸出							顯示
D	C	B	A	a	b	c	d	e	f	g	數字
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	1	1	0	0	9

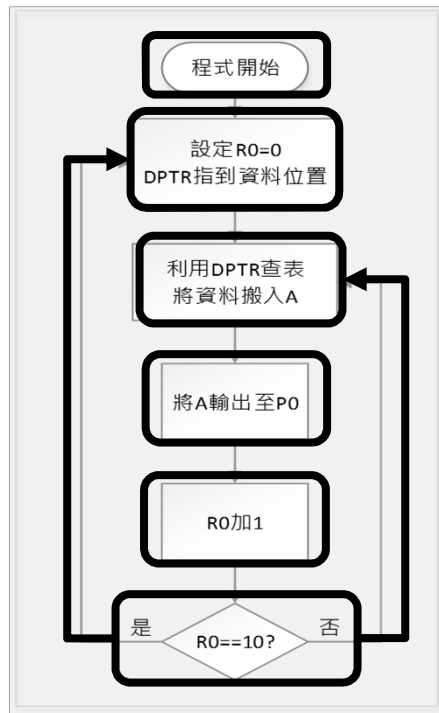


資料來源:
www.cpu.com.tw

電路圖



程式碼與流程圖(組合語言)

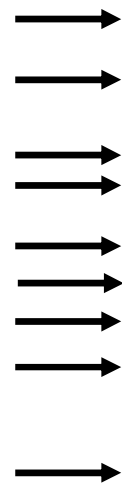


```
ORG 0
START: MOV R0,#0           ;設定輸出初始值
      MOV DPTR,#TABLE1     ;DPTR指到資料位置
LOOP:  MOV A,R0
      MOVC A,@A+DPTR        ;利用DPTR查表
      MOV P0,A              ;將資料搬入A
      INC R0                ;將A輸出至P0
      CALL DELAY            ;上數(加1)
      CJNE R0,#10,LOOP      ;呼叫延遲副程式
      JMP START             ;判斷R0是否超過10

DELAY: MOV R5,#5
DLOOP: MOV R6,#200
DLOOP2: MOV R7,#230
      DJNZ R7,$
      DJNZ R6,DLOOP2
      DJNZ R5,DLOOP
      RET
```

```
TABLE1: → DB 00111111B;
          DB 00000110B;
          DB 01011011B;
          DB 01001111B;
          DB 01100110B;
          DB 01101101B;
          DB 01111100B;
          DB 00000111B;
          DB 01111111B;
          DB 01100111B;
          END
```

程式碼與流程圖(C語言)



```
#include <REGX51.H>
void Delay_ms(int);
char SEG_table[]={0x3f,0x06,0x5b,0x4f,0x66,    //0~4
                  0x6d,0x7c,0x07,0x7f,0x67};    //5~9 .gfedcba
main(void)
{
    char i=0;
    while(1)
    {
        if(i>=10) i=0;
        P0=SEG_table[i];
        Delay_ms(500);
        i=i+1;
    }
}
void Delay_ms(int tx)
{
    char ti;
    while(tx--) for(ti=0;ti<101;ti++);
}
```

練習題

- **功能說明：**

由**AT89S51**的**Port0**控制七段顯示器輸出，使七段顯示器由**16**進制的“**F**”往下計數到“**0**”並循環。

單元五
8051的輸入與輸出埠控制
PART F

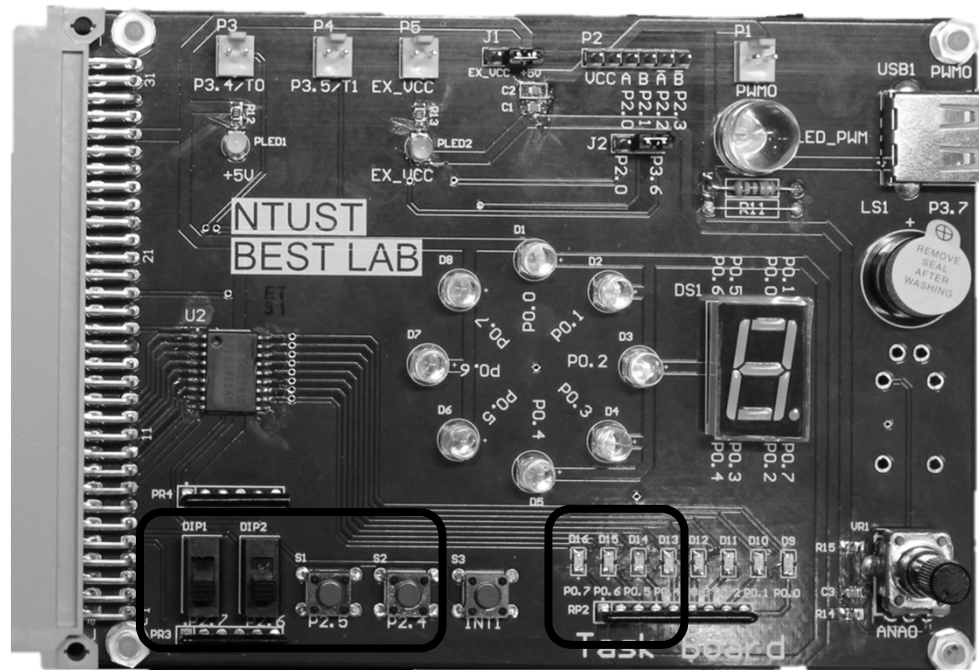
微算機原理及應用實習

範例三：開關輸入實驗

範例說明

- **實驗目的：**
瞭解8051讀取開關輸入之方法。
- **功能說明：**
AT89S51的Port2為開關輸入，將開關狀態讀入並透過Port0輸出至對應的LED。即開關導通時LED須點亮，開關不通時則LED不亮。

IO應用電路板(Task board 1)

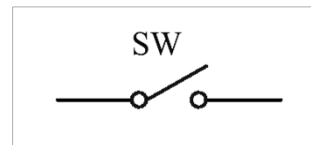
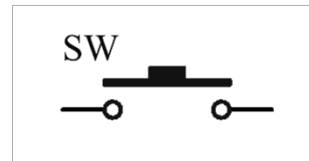


元件介紹

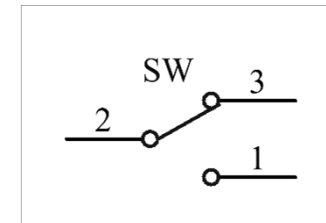
- 各種按鈕與開關外觀



- 按鈕開關符號

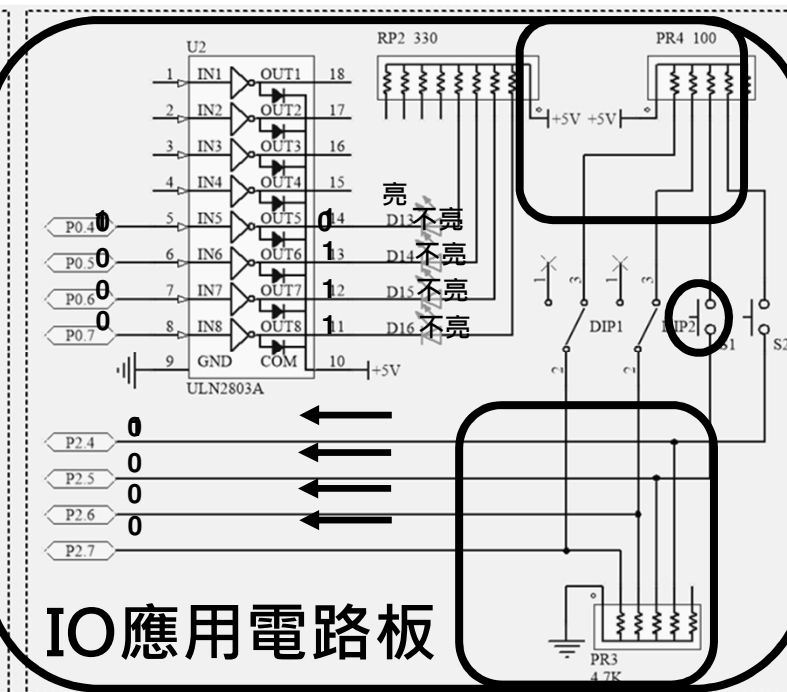
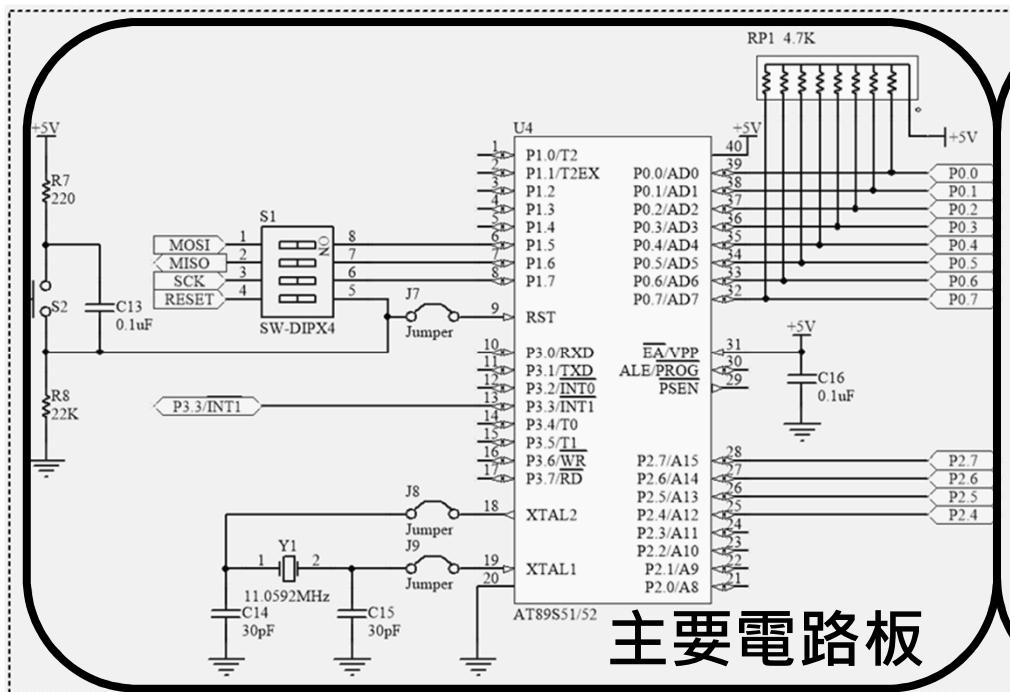


- 滑動開關符號

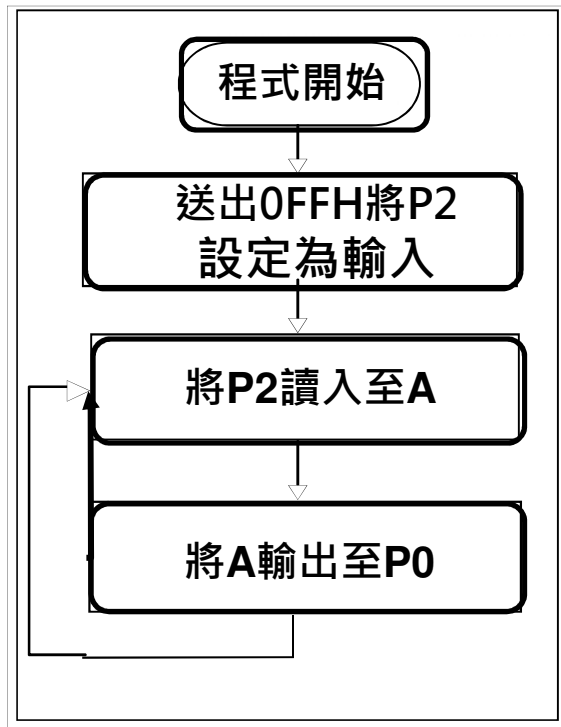


資料來源: ygic.com, icshop.com.tw

電路圖

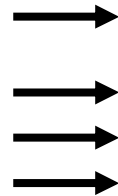


程式碼與流程圖(組合語言)



	ORG	0	←	
START:	MOV	P2,#0FFH	←	;將P2控制為輸入
SCAN:	MOV	A,P2	←	;將P2讀入
	ANL	A,#0F0H	←	
	MOV	P0,A	←	;將讀入值輸出至P0
	JMP	SCAN	←	
	END		←	

程式碼與流程圖(C語言)



```
#include <REGX51.H>
main(void)
{
    P0=0x00;          //關掉LED
    P2=0xFF;          //將P2控制為輸入
    while(1)
    {
        P0_4=P2_4;    // ;將P2讀入值輸出至P0
        P0_5=P2_5;
        P0_6=P2_6;
        P0_7=P2_7;
    }
}
```


練習題

- 功能說明：

AT89S51的Port2為按鈕輸入，當S1按下時，使Port0輸出的LED開始向左旋轉(可參考範例一)，放開S1按鈕時則停止。

單元五
8051的輸入與輸出埠控制
PART G

微算機原理及應用實習

範例四：按鈕控制實驗

範例說明

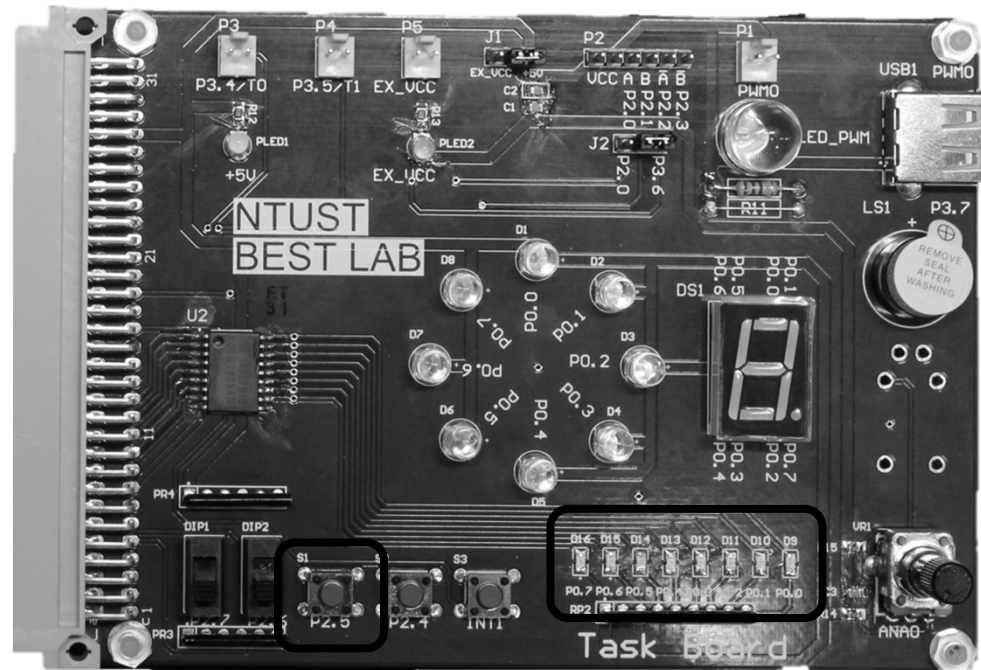
- **實驗目的：**

瞭解8051讀取按鈕輸入之方法與除彈跳原理。

- **功能說明：**

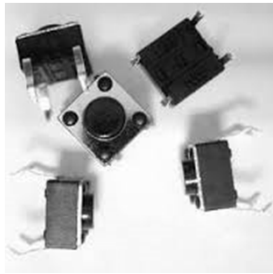
AT89S51的Port2為按鈕輸入，判斷按鈕之狀態並透過Port0輸出至對應LED。每按下一次，輸出的值加1。

IO應用電路板(Task board 1)

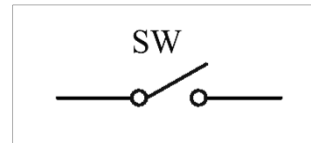
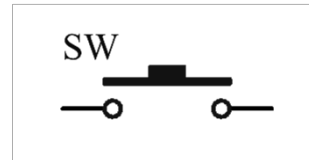


元件介紹

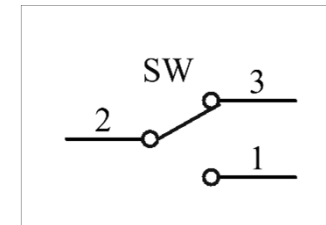
- 各種按鈕與開關外觀



- 按鈕開關符號

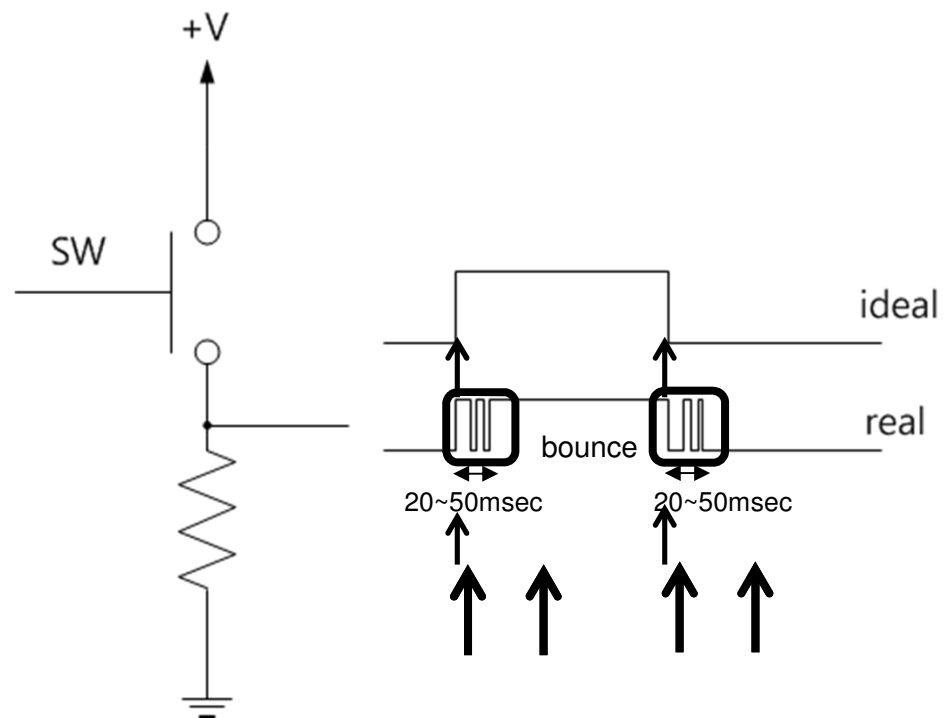


- 滑動開關符號

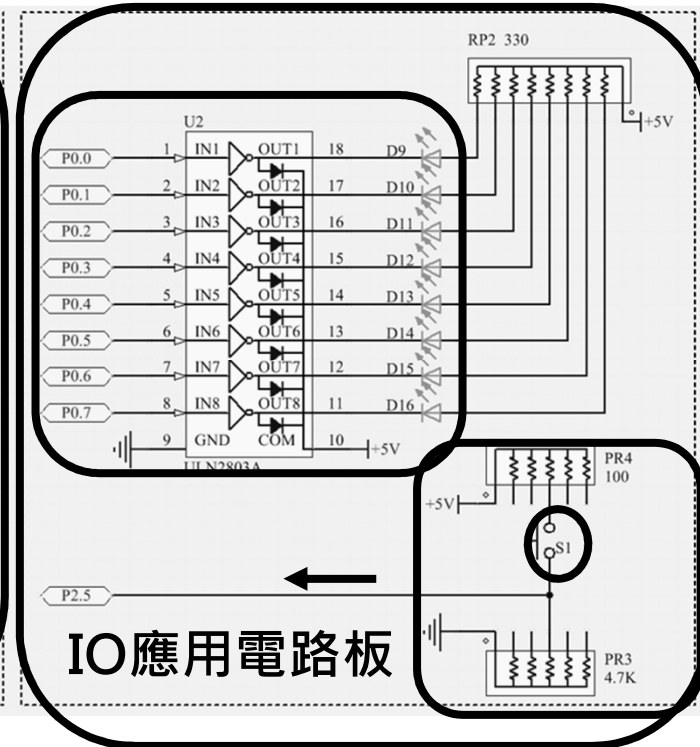
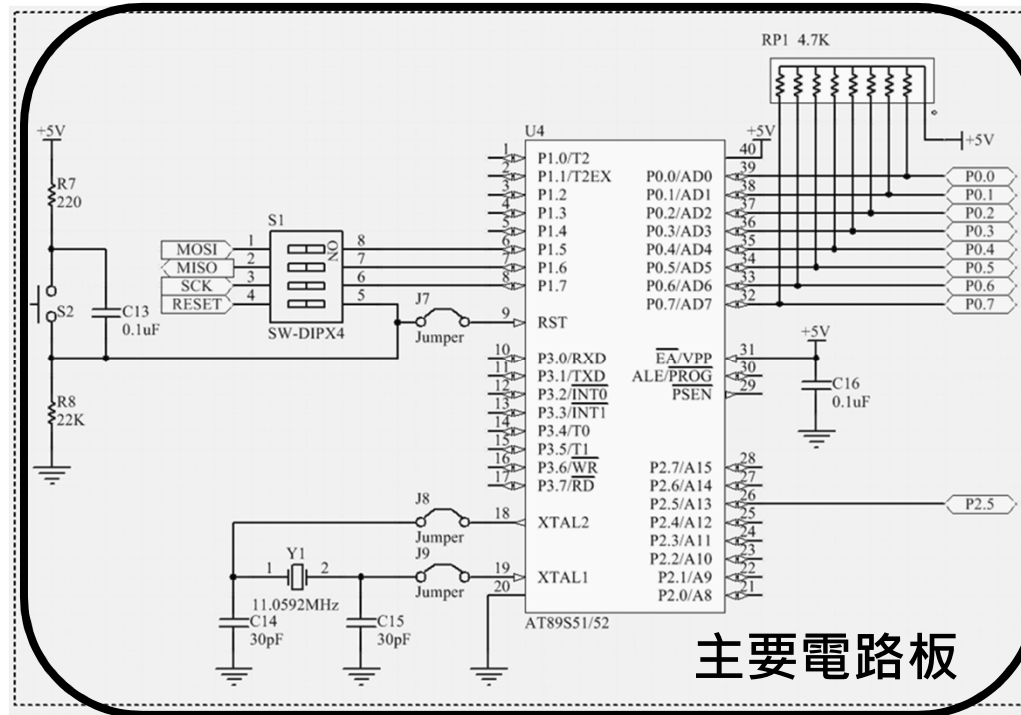


資料來源: ygic.com, icshop.com.tw

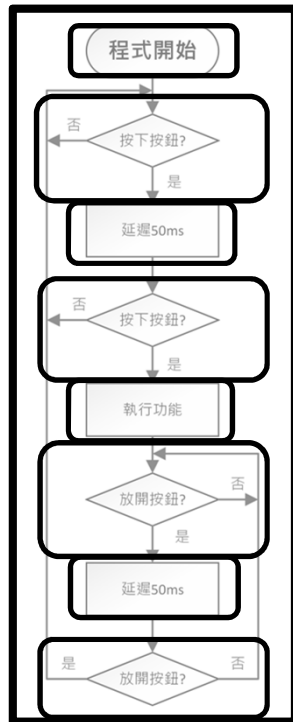
防彈跳原理



電路圖



程式碼與流程圖(組合語言)



	ORG	0	←←←	;程式開始
	MOV	R0,#0	←←←	;設定計數初值
	MOV	P0,#0	←←←	;輸出清為0
	SETB	P2.5	←←←	;設定P2.5為讀入
START:	JNB	P2.5,\$	←←←	;判斷P2.5有無按下
	CALL	DELAY	←←←	;延遲20ms
	JNB	P2.5, START	←←←	;判斷有無彈跳
DOUP:	INC	R0	←←←	;計數值加1
	MOV	P0,R0	←←←	;將計數值輸出
DEBOUN2:	JB	P2.5,\$	←←←	;判斷P2.5有無放開
	CALL	DELAY	←←←	;延遲20ms
	JB	P2.5,DEBOUN2	←←←	;判斷有無彈跳
	JMP	START	←←←	;延遲副程式
DELAY:	MOV	R6,#100		
DLOOP:	MOV	R7,#230		
	DJNZ	R7,\$		
	DJNZ	R6,DLOOP		
	RET			
	END			

程式碼與流程圖(C語言)

```
→ #include <REGX51.H>
void Delay_ms(int tx)
{
    char ti;
    while(tx--) for(ti=0;ti<101;ti++);
}
→ main(void)
{
    → char Counter=0;
    → P0=Counter; //P0顯示初始值
    → while(1)
    {
        → while(!P2_5);
        → Delay_ms(50);
        → if(P2_5)
        {
            → Counter++;
            → P0=Counter;
            → while(P2_5);
        }
    }
}
```

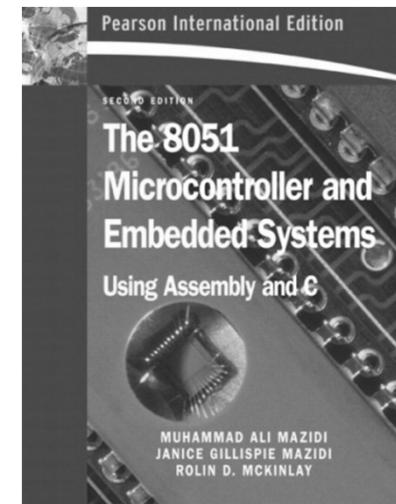
練習題

- **功能說明：**

AT89S51的Port2為按鈕輸入，每當S1按放一次，使Port0輸出的LED向左旋轉一位元(如範例一)。

5.4 參考文獻

- **ATMEL AT89S51 datasheet (doc2487.pdf)**
- **ATMEL 8051 Microcontrollers Hardware Manual (doc4316.pdf)**
- **ATMEL 8051 Microcontroller Instruction Set (doc0509.pdf)**
- **The 8051 Microcontroller and Embedded Systems Using Assembly and C, Second Edition, by Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay.**



5.5 複習題

- **8051輸入輸出埠的特性和功能？**
- **8051輸入輸出埠的控制方法？**
- **8051 C語言程式設計？**
- **如何控制LED？**
- **如何判斷按鈕輸入？**
- **如何控制七段顯示器？**