

微算機原理及應用

單元八：8051的中斷

授課老師：林淵翔 老師

大綱

- **中斷與輪詢(Interrupt vs. Polling)**
- **中斷服務程式和中斷向量(Interrupt Service Routine and Interrupt Vector)**
- **8051 有哪些中斷源(8051 Interrupts)**
- **8051 中斷之程式設計(Programming for Interrupt)**
- **8051 中斷之C程式設計(Programming Interrupt in C)**

大綱

- 範例一：外部中斷
- 範例二：計時器中斷
- 範例三：串列傳輸(傳送)中斷
- 範例四：串列傳輸(接收)中斷
- 參考文獻

單元八：8051的中斷

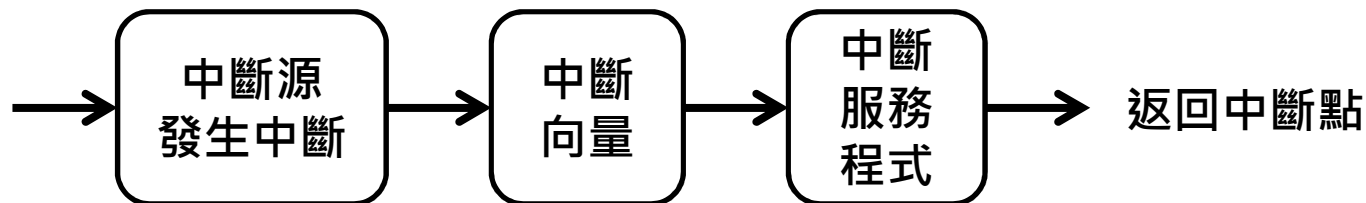
PART A

8.1 中斷與輪詢

- 一個微控制器服務周邊裝置的方法有中斷與輪詢(*Interrupt and Polling*)兩種。
- 輪詢方法：微控制器連續監控裝置的狀態，當狀態條件成立，就去執行這個服務。
- 中斷方法：當周邊裝置需要服務時，會送一個中斷信號通知微控制器。
 - 當微控制器收到中斷信號，無論他正在做甚麼，會馬上去服務這個裝置。
 - 執行和中斷相關的程式就稱為中斷服務程式(*Interrupt Service Routine (ISR)*)。

8.2 中斷服務程式與中斷向量

- 每一個中斷源，必須有一個對應的 *Interrupt Service Routine (ISR)*，or *Interrupt Handler*。
- 每一個中斷源，會有一個記憶體位址(中斷向量)用來指到不同的中斷服務程式位置，這一群記憶體位址稱為中斷向量表 (*Interrupt Vector Table*)。



8.3 8051 有哪些中斷源

- **8051 有 6 種中斷源**
 - **系統重置 (RESET)**
 - **2 個計時器中斷 (TF0 and TF1)**
 - **2 個外部硬體中斷 (INT0 and INT1)**
 - **1 個串列通訊中斷 (TI and RI)**

8.3 8051 有哪些中斷源

- 中斷向量表 (*Interrupt Vector Table*)

Interrupt Vector Table for the 8051			
Interrupt	ROM Location (Hex)	Pin	Flag Clearing
Reset	0000	9	Auto
External hardware interrupt 0 (INT0)	0003	P3.2 (12)	Auto
Timer 0 interrupt (TF0)	000B		Auto
External hardware interrupt 1 (INT1)	0013	P3.3 (13)	Auto
Timer 1 interrupt (TF1)	001B		Auto
Serial COM interrupt (RI and TI)	0023		Programmer clear it.

8.3 8051 有哪些中斷源

- 中斷的致能與除能(**Enabling and Disabling**)
 - 在**RESET**後，所有的中斷是**disabled (masked)**的。
 - 可以使用軟體去致能或除能中斷。
 - **IE (Interrupt Enable)** 暫存器就是用來 **enabling (unmasking) and disabling (masking)** 這些中斷的。

8.3 8051 有哪些中斷源

- IE (Interrupt Enable) 暫存器

D7				D0			
EA	--	ET2	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disable all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.					
--	IE.6	Not implemented, reserved for future use.*					
ET2	IE.5	Enables or disables Timer 2 overflow or capture interrupt (8052 only).					
ES	IE.4	Enables or disables the serial interrupt.					
ET1	IE.3	Enables or disables Timer 1 overflow interrupt.					
EX1	IE.2	Enables or disables external interrupt 1.					
ET0	IE.1	Enables or disables Timer 0 overflow interrupt.					
EX0	IE.0	Enables or disables external interrupt 0.					

單元八：8051的中斷

PART B

8.4 8051 中斷之程式設計

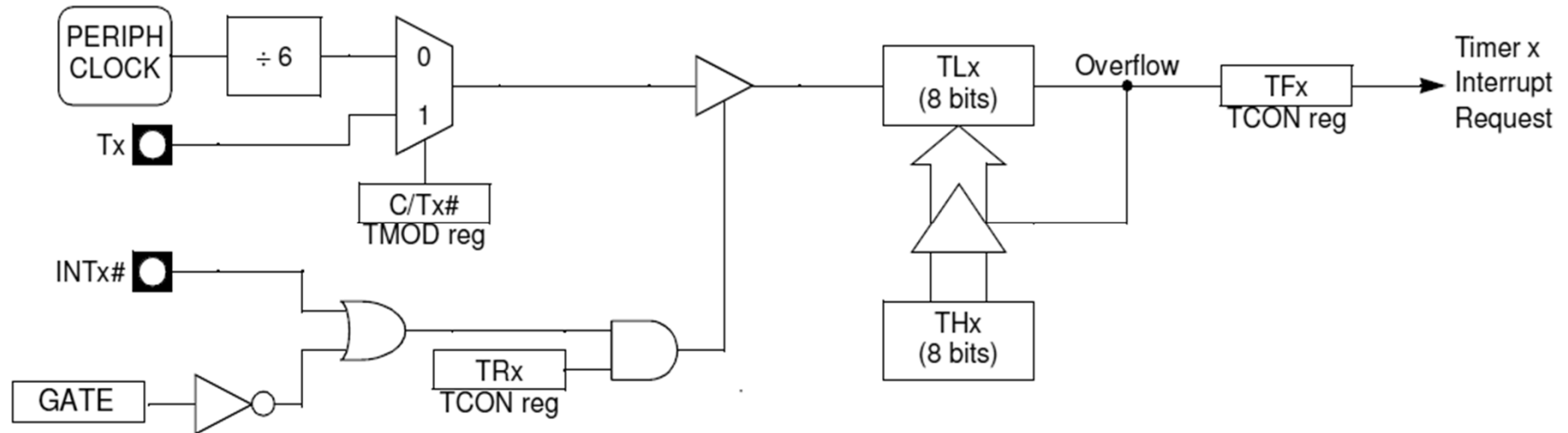
D7				D0			
EA	--	ET2	ES	ET1	EX1	ET0	EX0

Example:

- **MOV IE,#10010110B** ;enable Serial, Timer 0, EX1
- **CLR IE.1** ;mask(disable) Timer 0 interrupt only
- **CLR IE.7** ;disable all interrupts
- **SETB IE.7** ;EA = 1, Global enable
- **SETB IE.4** ;enable Serial interrupt
- **SETB IE.1** ;enable Timer 0 interrupt
- **SETB IE.2** ;enable EX1

8.4 8051 中斷之程式設計

- 8051 有兩個 Timer 中斷: TF0 and TF1



8.4 8051 中斷之程式設計

- 計時中斷範例：

a. 寫一程式將 8 位元的資料連續從 P0 送到 P1。

b. 同時輸出一週期為 200 μ s 的方波到 P2.2。



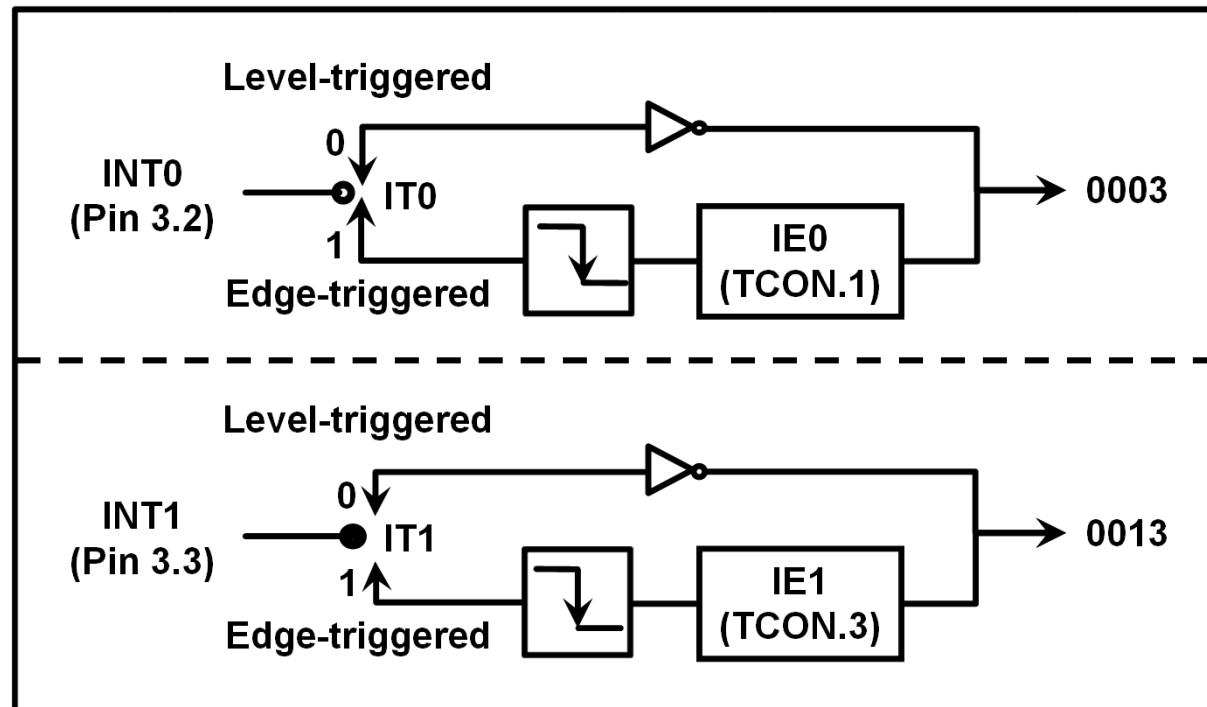
c. 請用 Timer 0 去產生此方波。

d. 假設 XTAL = 11.0592 MHz。

```
ORG 0000H
LJMP MAIN ;bypass interrupt vector table
;--ISR for Timer 0 to generate square wave
ORG 000BH ;Timer 0 interrupt vector table
CPL P2.2 ;toggle P2.2 pin
RETl ;return from ISR
;--The main program
ORG 0030H ;after vector table space
MAIN: MOV TMOD,#02H ;Timer 0, mode 2 (auto-reload)
MOV P0,#0FFH ;make P0 an input port
MOV TH0,#-92 ;TH0=A4H for -92 (100/1.085 = 92)
MOV IE,#82H ;IE=10000010(Bin) enable Timer 0
SETB TR0 ;Start Timer 0
BACK: MOV A,P0 ;get data from P0
MOV P1,A ;issue it to P1
SJMP BACK ;keep doing it
;loop unless interrupted by TF0
END
```

8.4 8051 中斷之程式設計

- 8051 有兩個外部硬體中斷
 - INT0
 - INT1



8.4 8051 中斷之程式設計

- **TCON 暫存器**

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit Number	Bit Mnemonic	Description
7	TF1	Timer 1 Overflow Flag Cleared by hardware when processor vectors to interrupt routine. Set by hardware on timer/counter overflow, when the timer 1 register overflows.
6	TR1	Timer 1 Control Bit Clear to turn off timer/counter 1. Set to turn on timer/counter 1.
5	TF0	Timer 0 Overflow Flag Cleared by hardware when processor vectors to interrupt routine. Set by hardware on timer/counter overflow, when the timer 0 register overflows.
4	TR0	Timer 0 Run Control Bit Clear to turn off timer/counter 0. Set to turn on timer/counter 0.
Reset Value = 0000 0000b		

8.4 8051 中斷之程式設計

• TCON 暫存器

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

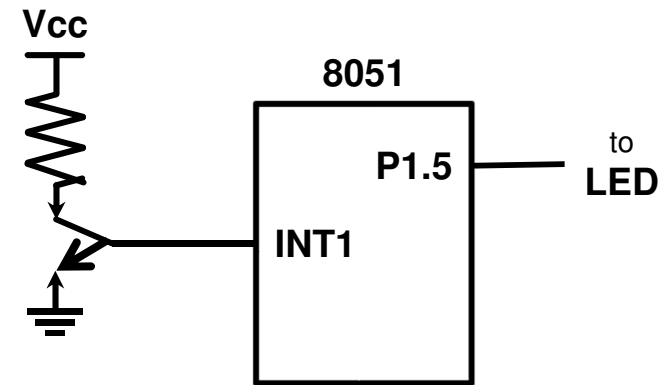
Bit Number	Bit Mnemonic	Description
3	IE1	Interrupt 1 Edge Flag Cleared by hardware when interrupt is processed if edge-triggered (see IT1). Set by hardware when external interrupt is detected on INT1# pin.
2	IT1	Interrupt 1 Type Control Bit Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.
1	IE0	Interrupt 0 Edge Flag Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.
0	IT0	Interrupt 0 Type Control Bit Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.
Reset Value = 0000 0000b		

8.4 8051 中斷之程式設計

- **準位觸發中斷(Level-triggered Interrupt)**
 - 一般狀態的 INT0 和 INT1 pins 是 High，如果有低準位(Low)信號就會觸發產生中斷。
 - 在 INT pin 的低準位信號必須在ISR的最後一個指令 RETI前被移除; 否則會產生另一個中斷。

8.4 8051 中斷之程式設計

- 外部硬體中斷範例：
 - a. 假設 INT1 接到一個開關並且一般狀態為 High。每當 INT1 變 0 時就會打開一 LED。
 - b. LED 接到 P1.5，且一般狀態為 off。當被打開時，會亮一段時間後自動關閉。只要開關一直按著，LED 就會一直亮著。



8.4 8051 中斷之程式設計

```
ORG    0000H
LJMP    MAIN                ;bypass interrupt vector table
;--ISR for hardware interrupt INT1 to turn on the LED
ORG    0013H                ;INT1 ISR
SETB    P1.5                ;turn on LED
MOV     R3,#255              ;load counter
BACK:   DJNZ    R3,BACK      ;keep LED on for a while
CLR     P1.5                ;turn off the LED
RETI                    ;return from ISR
;--The main program
ORG    0030H
MAIN:   MOV     IE,#10000100B ;enable external INT1
HERE:   SJMP    HERE        ;stay here until interrupted
END
```

8.4 8051 中斷之程式設計

- 邊緣觸發中斷(Edge-triggered Interrupt)
 - 在 TCON 暫存器的 IT0 (bit 0) 和 IT1(bit 2) 用來決定是 Level- or Edge-triggered 模式。
 - 例如: “SETB TCON.2” 可以使 設為 *Edge-triggered Interrupt* , 因此當有一個 High-to-Low 信號加入P3.3 時, 微控制器會被中斷並被強制跳到中斷向量表中 0013H 的位置去執行中斷服務程式(ISR)。(假設IE暫存器的EX1有致能)

8.4 8051 中斷之程式設計

- 外部硬體中斷範例：
 - a. 假設 INT1 接到一個脈波產生器，每當脈波的負緣時就會送一個 High 到 P1.5。
 - b. P1.5 接一個 LED 或蜂鳴器。

8.4 8051 中斷之程式設計

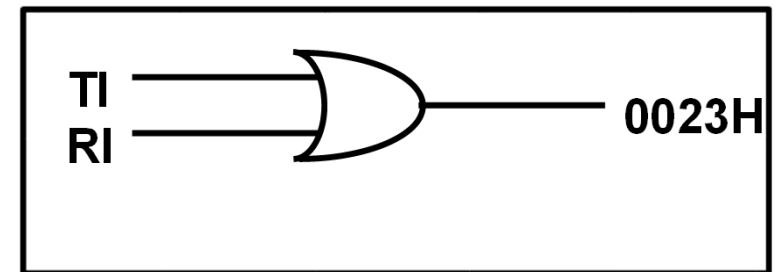
```
ORG    0000H
LJMP    MAIN        ;bypass interrupt vector table
;--ISR for hardware interrupt INT1 to turn on the LED
ORG    0013H        ;INT1 ISR
SETB    P1.5        ;turn on the LED
MOV     R3,#255
BACK:   DJNZ    R3,BACK ;keep the LED on for a while
CLR     P1.5        ;turn off the LED
RETI                    ;return from ISR
;--The main program
ORG    0030H
MAIN:   SETB    TCON.2 ;make INT1 edge-trigger interrupt
MOV     IE,#10000100B ;enable External INT1
HERE:   SJMP    HERE  ;stay here until interrupted
END
```

單元八：8051的中斷

PART C

8.4 8051 中斷之程式設計

- 串列通訊中斷
 - TI (Transfer Interrupt) 變成 1 表示資料傳完，SBUF 暫存器可以再傳另一個 byte 資料。
 - RI (Received Interrupt) 變成 1 表示收到一個 byte 資料。
 - 8051 只有一個中斷給串列通訊。
 - 所以 ISR 必須自己判斷是 TI 或 RI 造成中斷。



Serial interrupt is invoked by TI or RI flags

8.4 8051 中斷之程式設計

- 串列通訊中斷範例：
 - a. 寫一程式將 8 位元的資料連續從 P1 送到 P2。
 - b. 並複製一份資料送到 COM port。
 - c. 假設XTAL = 11.0592 MHz。Baud rate 設為 9600。

8.4 8051 中斷之程式設計

```
ORG      0
LJMP     MAIN
; Serial interrupt vector
ORG      23H
LJMP     SERIAL      ;jump to serial interrupt ISR
; The main program
ORG      30H
MAIN:    MOV     P1, #0FFH      ;make P1 an input port
         MOV     TMOD, #20H     ;timer1, mode 2(auto-reload)
         MOV     TH1, #0FDH    ;9600 baud rate
         MOV     SCON, #50H    ;8-bit, 1 stop, REN enabled
         MOV     IE, #10010000B ;enable serial interrupt
         SETB    TR1           ;start timer 1
```

8.4 8051 中斷之程式設計

```
LOOP:  MOV    A, P1          ;read data from port 1
        MOV    SBUF, A      ;give a copy to SBUF
        MOV    P2, A        ;send it to P2
        SJMP   LOOP         ;stay in loop indefinitely

;--Serial port ISR
        ORG    200H
SERIAL: JB    TI, TRANS      ;jump if TI is high
        MOV    A, SBUF      ;otherwise due to receive
        CLR    RI           ;clear RI since CPU does not
        RETI                ;return from ISR
TRANS:  CLR    TI           ;clear TI since CPU does not
        RETI                ;return from ISR
        END
```

8.4 8051 中斷之程式設計

- 在 **RETI** 指令前清除 **RI** 和 **TI** 旗標
 - 這是需要的，因為 **8051** 只有一個中斷給傳送和接收，所以 **8051** 不知道是誰產生的中斷，因此，需要在 **ISR** 中去清除旗標。

8.4 8051 中斷之程式設計

8051/52 的中斷旗標位元		
Interrupt	Flag	SFR Register Bit
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 0	TF0	TCON.5
Timer 1	TF1	TCON.7
Serial port	TI, RI	SCON.1, SCON.0
Timer 2	TF2	T2CON.7 (AT89C52)
Timer 2	EXF2	T2CON.6 (AT89C52)

8.4 8051 中斷之程式設計

- 串列通訊和計時中斷範例：
 - a. 接收串列資料並送到 P0。
 - b. 讀 P1 資料並傳送到串列埠。
 - c. 使用 Timer 0 產生一 5 kHz 的方波在 P2.1。
 - d. 假設XTAL = 11.0592 MHz。Baud rate 設為 4800。

8.4 8051 中斷之程式設計

```

                                ORG      0
                                LJMP     MAIN
                                ORG      000BH          ;ISR for Timer 0
                                CPL       P2.1          ;toggle P2.1
                                RETI
                                ORG      23H          ;ISR for UART
                                LJMP     SERIAL          ;jump to serial int. ISR

MAIN:  ORG      30H
        MOV     P1,#0FFH          ;make P1 an input port
        MOV     TMOD,#22H          ;timer 0&1, mode 2, auto-reload
        MOV     TH1,#0F6H          ;4800 baud rate
        MOV     SCON,#50H          ;8-bit, 1 stop, REN enabled
        MOV     TH0,#-92           ;for 5 kHz wave
        MOV     IE,#10010010B     ;enable serial, timer 0 int.
        SETB    TR1                ;start timer 1
        SETB    TR0                ;start timer 0
```


8.4 8051 中斷之程式設計

BACK:	MOV	A,P1	;read data from port 1
	MOV	SBUF,A	;give a copy to SBUF
	SJMP	BACK	;stay in loop indefinitely
;-----Serial port ISR			
	ORG	100H	
SERIAL:	JB	TI,TRANS	;jump if TI is high
	MOV	A,SBUF	;otherwise due to receive
	MOV	P0,A	;send serial data to P0
	CLR	RI	;clear RI since CPU doesn't
	RETI		;return from ISR
TRANS:	CLR	TI	;clear TI since CPU doesn't
	RETI		;return from ISR
	END		

單元八：8051的中斷

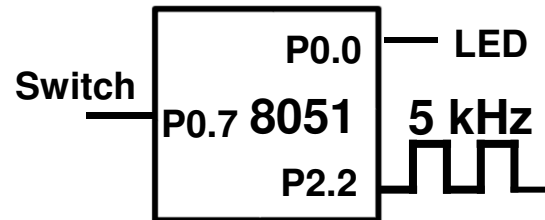
PART D

8.5 8051 中斷之C程式設計

8051/52 Interrupt Number in C		
Interrupt	Name	Numbers used by 8051 C
External Interrupt 0	(INT0)	0
Timer Interrupt 0	(TF0)	1
External Interrupt 1	(INT1)	2
Timer Interrupt 1	(TF1)	3
Serial Communication	(RI + TI)	4
Timer 2 (8052 only)	TF2	5

8.5 8051 中斷之C程式設計

- **C程式中斷範例 1：**
 - a. 寫一程式將 1 位元的資料連續從 P0.7 送到 P0.0。
 - b. 同時輸出一週期為 200 μ s 的方波到 P2.2。
 - c. 請用 Timer 0 去產生此方波。
 - d. 假設 XTAL = 11.0592 MHz。



```
#include <REGX51.H>
sbit SW    = P0^7;
sbit IND    = P0^0;
sbit WAVE = P2^2;
void timer0(void) interrupt 1 {
    WAVE = ~WAVE;    //toggle pin
}
void main(){
    SW = 1;
    TMOD = 0x02;
    TH0 = 0xA4;      //TH0 = -92;
    TR0 = 1;
    IE = 0x82; //enable interrupts for timer 0
    while(1){
        IND = SW; //send switch to LED
    }
}
```

8.5 8051 中斷之C程式設計

- **C程式中斷範例 2：**
 - a. 寫一程式將 1 位元的資料連續從 P0.7 送到 P0.0。
 - b. 同時輸出一週期為 200 μ s 的方波到 P2.2。
 - c. 請用 Timer 0 去產生此方波。
 - d. 送一次字元 'A' 到串列埠。Baud rate 設為 9600。
 - e. 假設 XTAL = 11.0592 MHz。

8.5 8051 中斷之C程式設計

```
#include <REGX51.H>
sbit SW    = P0^7;
sbit IND   = P0^0;
sbit WAVE  = P2^2;
void timer0(void) interrupt 1 {
    WAVE = ~WAVE;    //toggle pin
}
void serial0() interrupt 4 {
    if(TI == 1){
        TI = 0;      //clear interrupt
    }
    else{
        RI = 0;      //clear interrupt
    }
}
```

```
void main(){
    SW = 1;           //make switch input
    TH1 = -3;         //9600 baud
    TMOD = 0x22;      //mode 2 for both timers
    TH0 = 0xA4;        //-92 = A4H for timer 0
    SCON = 0x50;
    TR0 = 1;
    TR1 = 1;          //start timer
    IE = 0x92; //enable interrupt for T0, serial
    SBUF = 'A'; //send A to serial port
    while(1){
        IND = SW; send switch to LED
    }
}
```

8.5 8051 中斷之C程式設計

8051/52 Interrupt Priority Upon Reset (中斷優先權)	
Highest to Lowest Priority	
External Interrupt 0	(INT0)
Timer Interrupt 0	(TF0)
External Interrupt 1	(INT1)
Timer Interrupt 1	(TF1)
Serial Communication	(RI + TI)
Timer 2 (8052 only)	TF2

單元八：8051的中斷

PART E

微算機原理及應用實習

範例一：外部中斷

範例說明

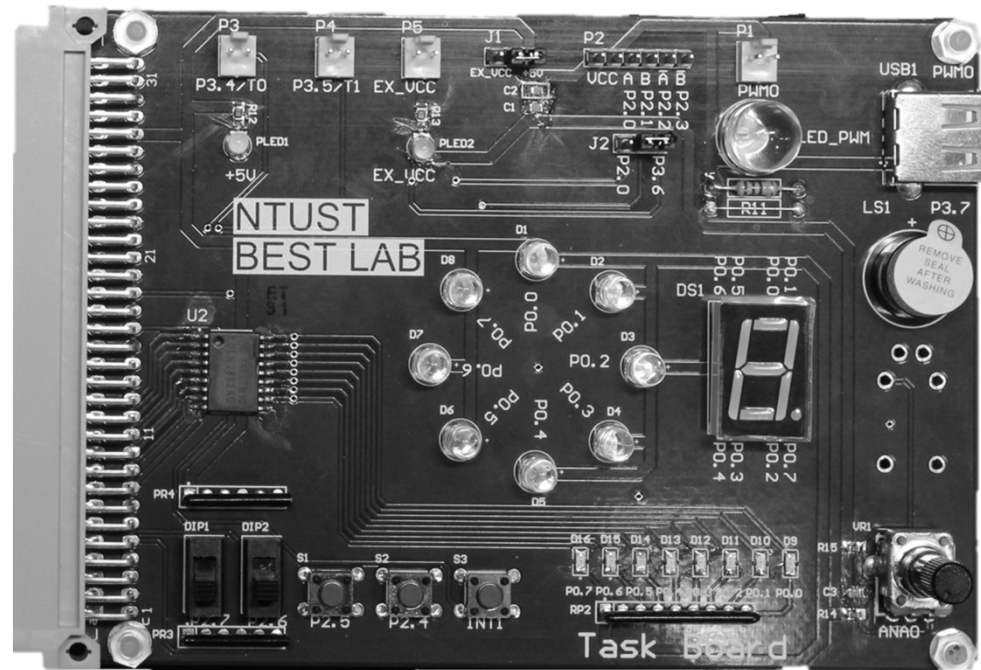
- **實驗目的：**

瞭解 8051 之外部中斷控制方法，並練習使用外部中斷來執行中斷功能。

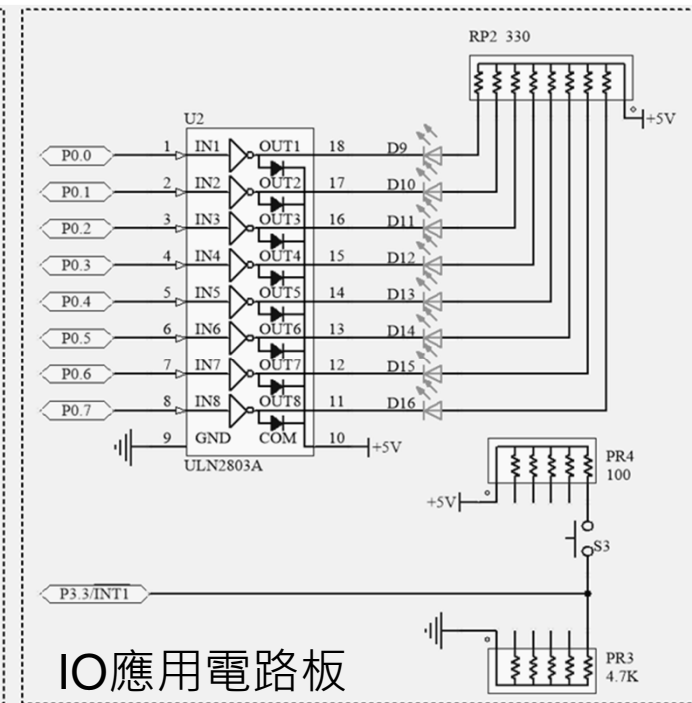
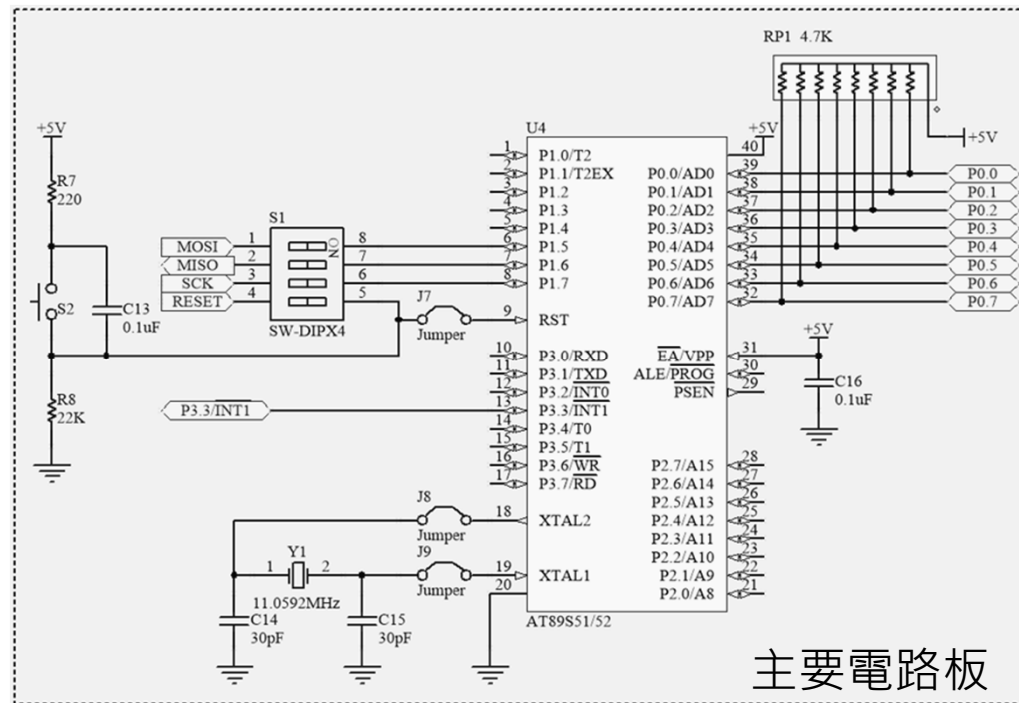
- **功能說明：**

由 AT89S51 的 Port 0 輸出控制 LED 燈，平時 LED 每 1 秒全部亮滅一次，當外部中斷 INT1 發生時，LED 由 D9 開始向左旋轉直到 D16 後回到全部亮滅狀態。

IO應用電路板(Task board 1)



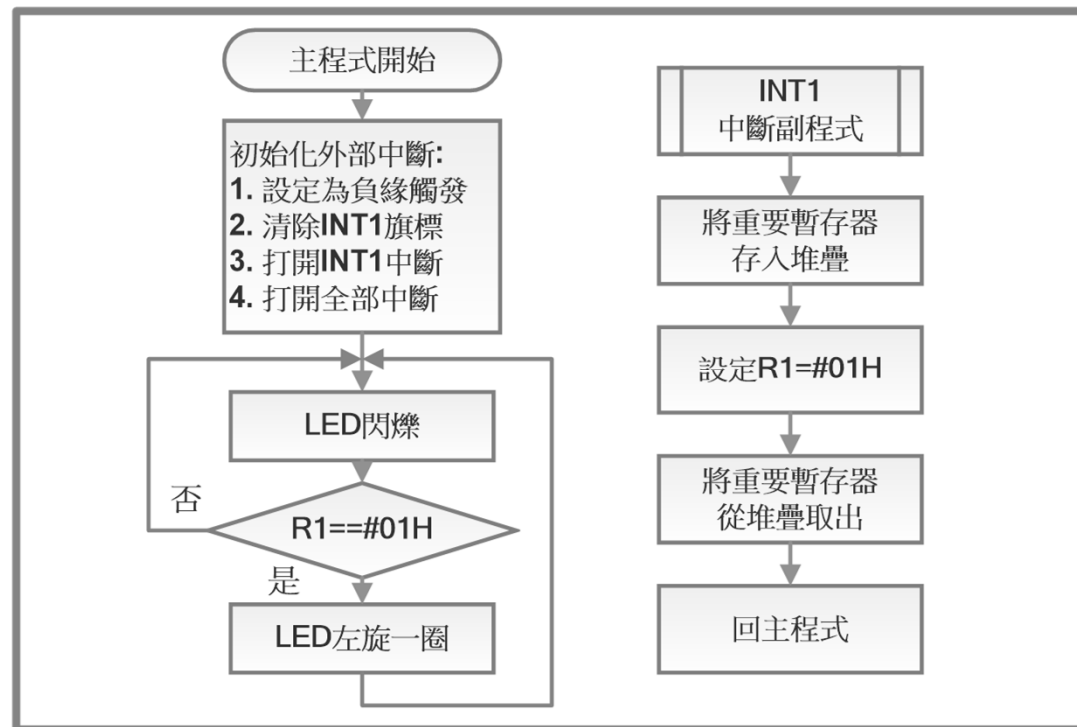
電路圖



外部中斷設定方式

組合語言	C語言
<pre>;初始化要做的事 START: SETB IT1 ;設定負緣觸發 CLR IE1 ;清除INT1旗標 SETB EX1 ;打開INT1中斷 SETB EA ;打開全部中斷 ;中斷服務要做的事 EX0_INT: PUSH PSW ;將PSW送入堆疊 PUSH ACC ;將ACC送入堆疊 ;中斷服務程式 POP ACC ;將ACC取出堆疊 POP PSW ;將PSW取出堆疊 RETI</pre>	<pre>//初始化要做的事 IT1 = 1; //設定負緣觸發 IE1 = 0; //清除INT1旗標 EX1 = 1; //打開INT1中斷 EA = 1; //打開全部中斷 //中斷服務要做的事 void EX1(void) interrupt 2 { //中斷服務程式 }</pre>

程式流程圖



程式碼(組合語言)

```

ORG      0
JMP      START
ORG      13H      ;外部中斷INT1向量
JMP      INT_1    ;跳躍執行中斷副程式
START:   SETB     IT1      ;外部中斷負緣觸發
        CLR      IE1      ;清除INT1旗標
        SETB     EX1      ;開啟外部中斷開關
        SETB     EA      ;開啟中斷總開關
        MOV      R0,#0
        MOV      R1,#0
LOOP:    MOV      A,R0      ;全部LED亮滅
        CPL      A
        MOV      R0,A
        MOV      P0,R0
        MOV      R5,#5
D500:    CALL     DELAY
        DJNZ     R5,D500
        CJNE     R1,#01H,LOOP ;判斷是否要左旋
    
```

```

LOOP2:   MOV      P0,R1      ;LED左旋部分
        CALL     DELAY
        MOV      A,R1
        RL       A
        MOV      R1,A
        CJNE     R1,#01H,LOOP2
        MOV      R1,#0
        JMP      LOOP
INT_1:   PUSH     PSW      ;將PSW送入堆疊
        PUSH     ACC      ;將ACC送入堆疊
        MOV      R1,#01H  ;設定要左旋
        POP      ACC      ;將ACC取出堆疊
        POP      PSW      ;將PSW取出堆疊
        RETI
DELAY:   MOV      R6,#200
DLOOP:   MOV      R7,#230
        DJNZ     R7,$
        DJNZ     R6,DLOOP
        RET
        END
    
```

程式碼(C語言)

```
#include <REGX51.H>
unsigned char P0_rl=0x00; //宣告變數與常數
void Delay_ms(int);       //宣告副程式型態
main(void)                //主程式開始
{
    unsigned char P0_buf=0;
    IT1=1;                 //設定INT1負緣觸發
    IE1=0;                 //清除INT1旗標
    EX1=1;                 //開啟INT1中斷
    EA=1;                  //開啟總中斷
    while(1)
    {
        P0_buf=~P0_buf;    //取反向
        P0=P0_buf;         //P0的LED持續閃爍
        Delay_ms(500);
        while(P0_rl)       //若發生中斷進入此迴圈
        {
            P0=P0_rl;
            Delay_ms(100);
            P0_rl<<=1;
        }
    }
}
```

```
// -----中斷副程式開始-----
void INT1_isr(void) interrupt 2
{
    P0_rl=0x01;    //設定P0_rl初始值
}

//-----副程式開始-----
void Delay_ms(int tx)
{
    char ti;
    while(tx--)
        for(ti=0;ti<101;ti++);
}
```


影片

練習題

- **功能說明：**

由 **AT89S51** 的 **Port 3.7** 輸出控制蜂鳴器，並修改參考程式快樂頌 ("**HappySong.c**")，按下 **S1** 播放後，請使用外部中斷讓音樂播放到一半可以立即按下 **S3** 停止播放。

影片

單元八：8051的中斷

PART F

微算機原理及應用實習

範例二：計時器中斷

範例說明

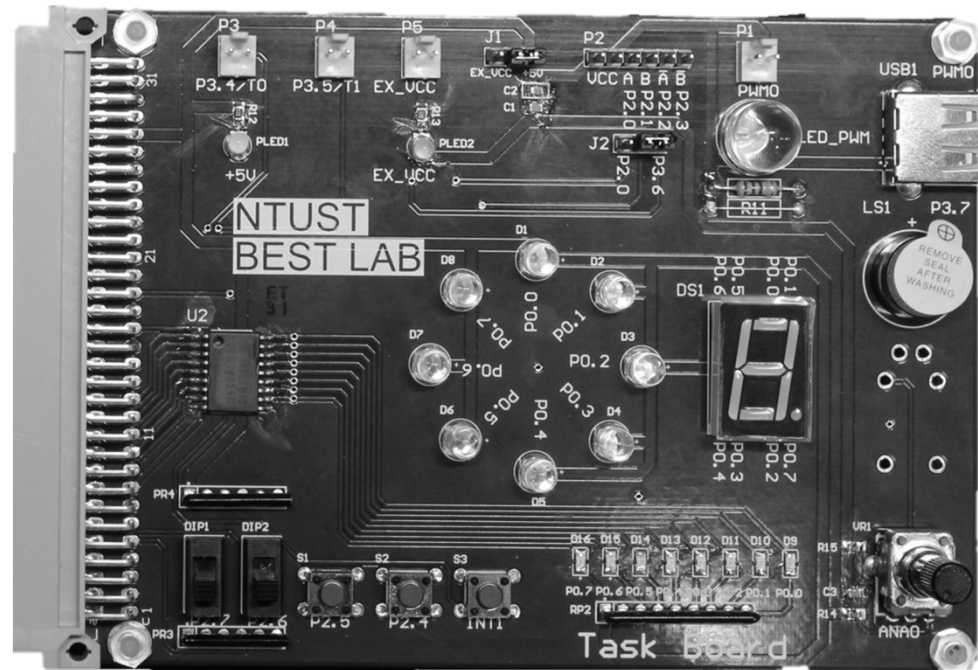
- **實驗目的：**

瞭解 **8051** 之計時器中斷控制方法，並練習使用計時器中斷實作時間延遲副程式。

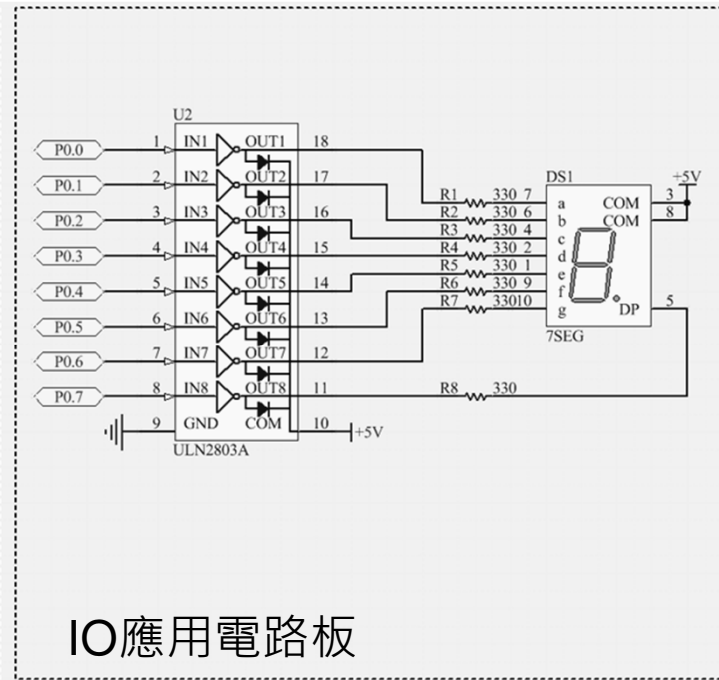
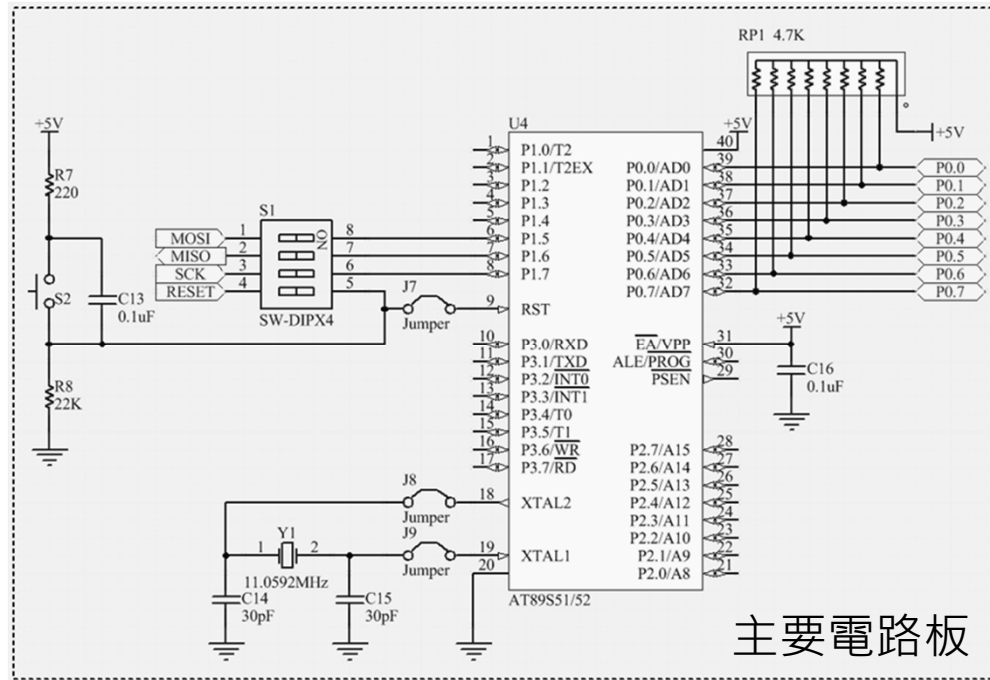
- **功能說明：**

使用計時器中斷實作 **1** 秒的時間延遲副程式，並由 **AT89S51** 的 **Port 0** 輸出控制七段顯示器，使其每一秒依序從 **0~9** 上數並顯示。

IO應用電路板(Task board 1)



電路圖

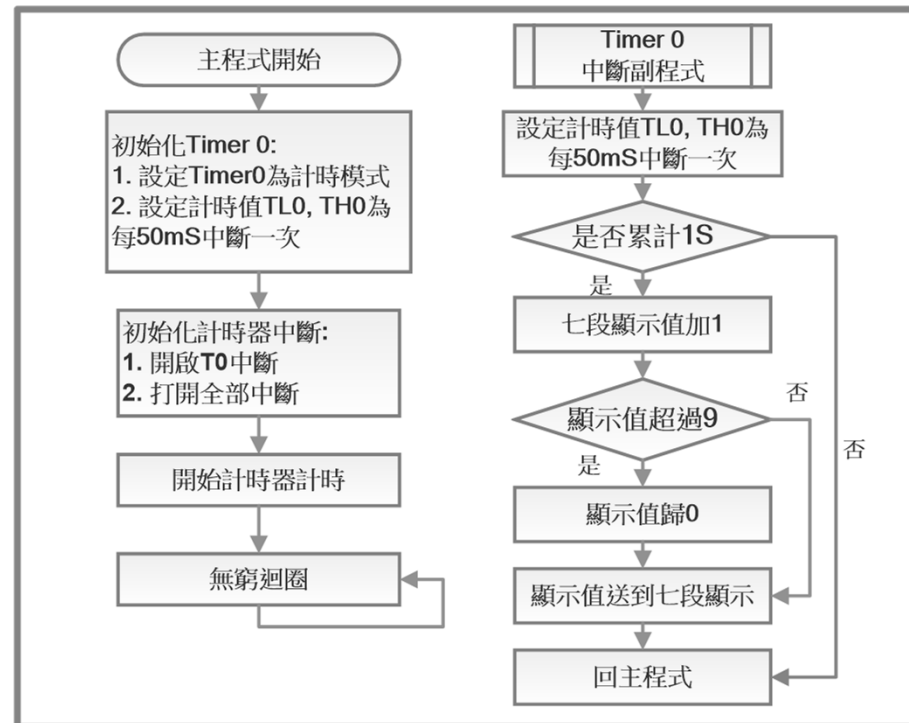


影片

計時器中斷設定方式

組合語言	C語言
<pre>;初始化要做的事 START: MOV TMOD,#01H ;將TIMER0設為模式1 MOV TL0,#LOW(65536-46080) ;設定計時/計數值 MOV TH0,#HIGH(65536-46080) SETB ET0 ;打開TIMER0中斷 SETB EA ;打開全部中斷 SETB TR0 ;設定開始計數 ;中斷服務要做的事 TIMER0_INT: MOV TL0,#LOW(65536-46080) ;設定計時/計數值 MOV TH0,#HIGH(65536-46080) ;中斷服務程式 TIMER0_INT_EXIT: RETI ;離開中斷</pre>	<pre>;初始化要做的事 TMOD = 0x01 ;將TIMER0設為模式1 TL0 = (65536-46080)%256 ;設定計時/計數值 TH0 = (65536-46080)/256 ET0 = 1; ;打開TIMER0中斷 EA = 1; ;打開全部中斷 TR0 = 1; ;設定開始計數 ;中斷服務要做的事 void Timer0(void) interrupt 1 { TL0 = (65536-46080)%256 ;設定計時/計數值 TH0 = (65536-46080)/256 ;中斷服務程式 }</pre>

程式流程圖



程式碼(組合語言)

```
ORG    0
JMP     START
ORG    0BH
JMP     TIMER0
```

START:

```
MOV     DPTR,#TABLE      ;七段顯示器查表
MOV     R0,#0
MOV     R1,#0
MOV     TMOD,#01H        ;設定計時器模式
MOV     TH0,#HIGH(65536-46080) ;設定計時數值
MOV     TL0,#LOW(65536-46080)
MOV     P0,#3FH
SETB    ET0              ;開啟T0中斷
SETB    EA               ;開啟中斷總開關
SETB    TR0              ;開始計時
JMP     $                ;無窮迴圈
```

TIMER0:

```
MOV     TH0,#HIGH(65536-46080) ;設定計時數值
MOV     TL0,#LOW(65536-46080)
INC     R1
CJNE    R1,#20,EXIT        ;20 * 50ms = 1s
MOV     R1,#0
INC     R0                  ;上數 加1
CJNE    R0,#10,DISPLAY
MOV     R0,#0              ;上數超過9 → 歸0
```

DISPLAY:

```
MOV     A,R0               ;查表顯示七段
MOVC    A,@A+DPTR
MOV     P0,A
```

EXIT:

```
RETI
```

TABLE:

```
DB      3FH, 06H, 5BH, 4FH, 66H
DB      6DH, 7CH, 07H, 7FH, 67H
END
```

程式碼(C語言)

```
#include <REGX51.H>
// -----宣告變數與常數-----
code char SEG_table[]={0x3f,0x06,0x5b,0x4f,0x66, //0~4
                        0x6d,0x7c,0x07,0x7f,0x67}; //5~9

char xi=0,div_t0;
//Tsys=12/Fosc=12/11.0592MHz=1.085069444us
//計數Timer值 = 50ms/Tsys = 46080
#define T0Val (65536-46080)
// -----主程式開始-----
void main(void)
{
    P0=SEG_table[xi];
    TMOD=0x01; //設定T0計時器模式
    TH0=T0Val/256; //設定計時數值
    TL0=T0Val%256;
    ET0=1; //開啟T0中斷
    EA=1; //開啟中斷總開關
    TR0=1; //開始計時
    while(1); //無窮迴圈
}
```

```
// -----中斷副程式開始-----
void T0_isr(void) interrupt 1
{
    TH0=T0Val/256; //設定計時值
    TL0=T0Val%256;
    div_t0++;
    if(div_t0>=20) //20 * 50ms = 1s
    {
        div_t0=0;
        xi++; //上數 加1
        if(xi>9)xi=0; //上數超過9->歸0
        P0=SEG_table[xi]; //查表顯示七段
    }
}
```

練習題

- **功能說明：**

倒數計時器，由 **AT89S51** 的 **P2.5** 讀取按鈕狀態，當按下按鈕後，**Port 0** 輸出控制七段顯示器從 **5** 開始每一秒下數一次，到 **0** 即停止。請使用計時器中斷來做時間延遲副程式。

影片

單元八：8051的中斷

PART G

微算機原理及應用實習

範例三：串列傳輸(傳送)中斷

範例說明

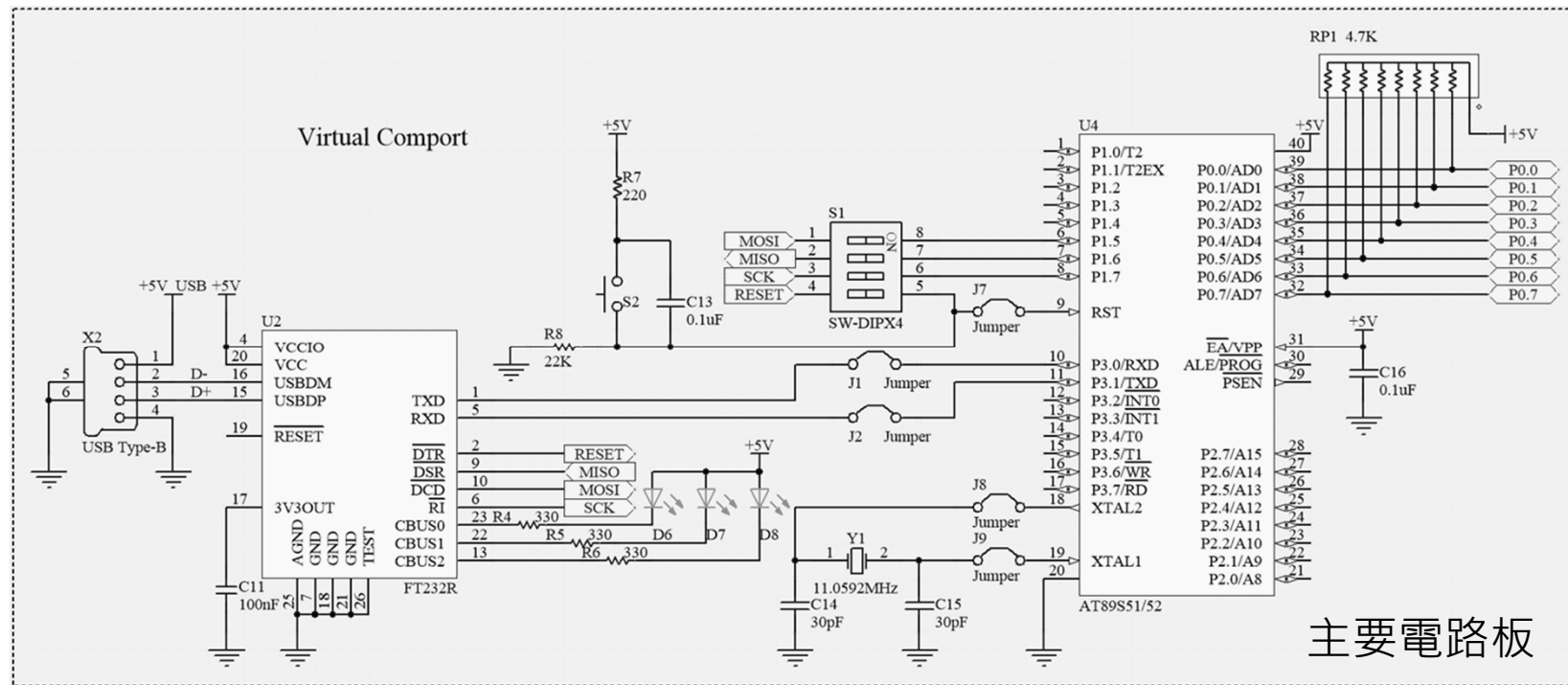
- **實驗目的：**

瞭解 **8051** 之串列傳輸(傳送)中斷控制方法，並練習使用串列傳輸(傳送)中斷來傳送字串功能。

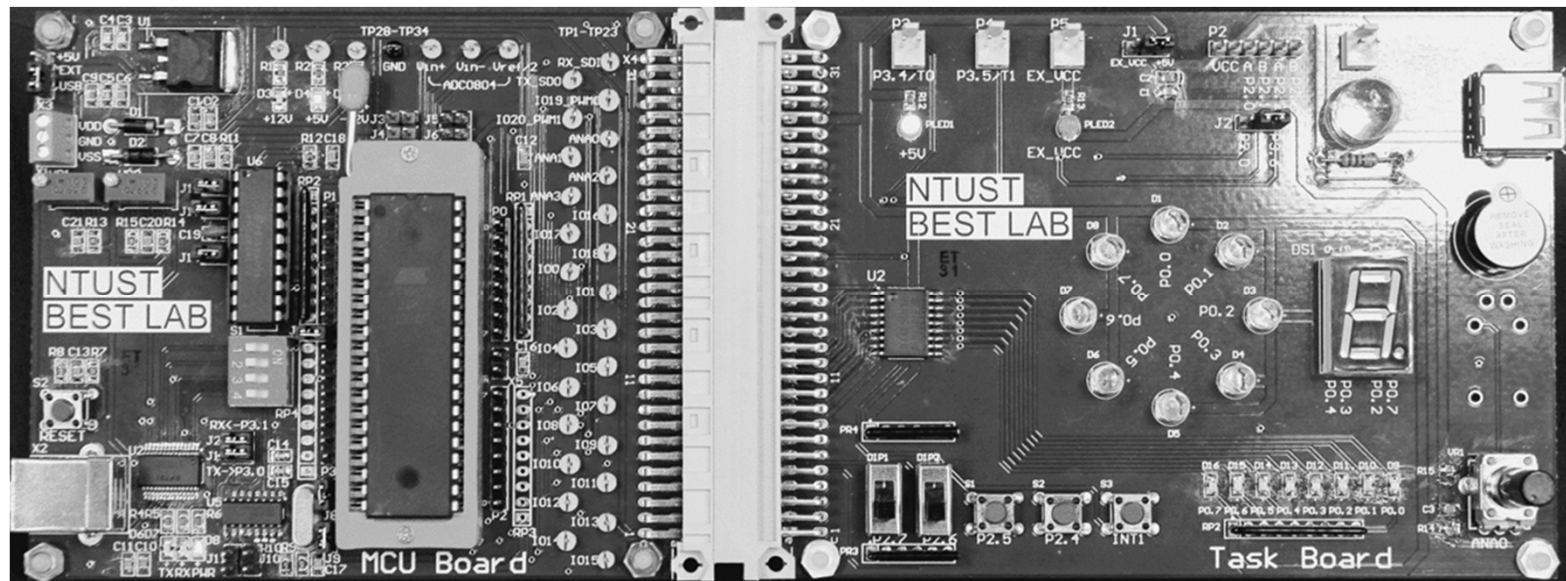
- **功能說明：**

由電腦 **PuTTY** 介面顯示資料，**AT89S51** 傳送 **Hello World!** 字串顯示在 **PuTTY**上。

電路圖



主電路板(MCU Board)

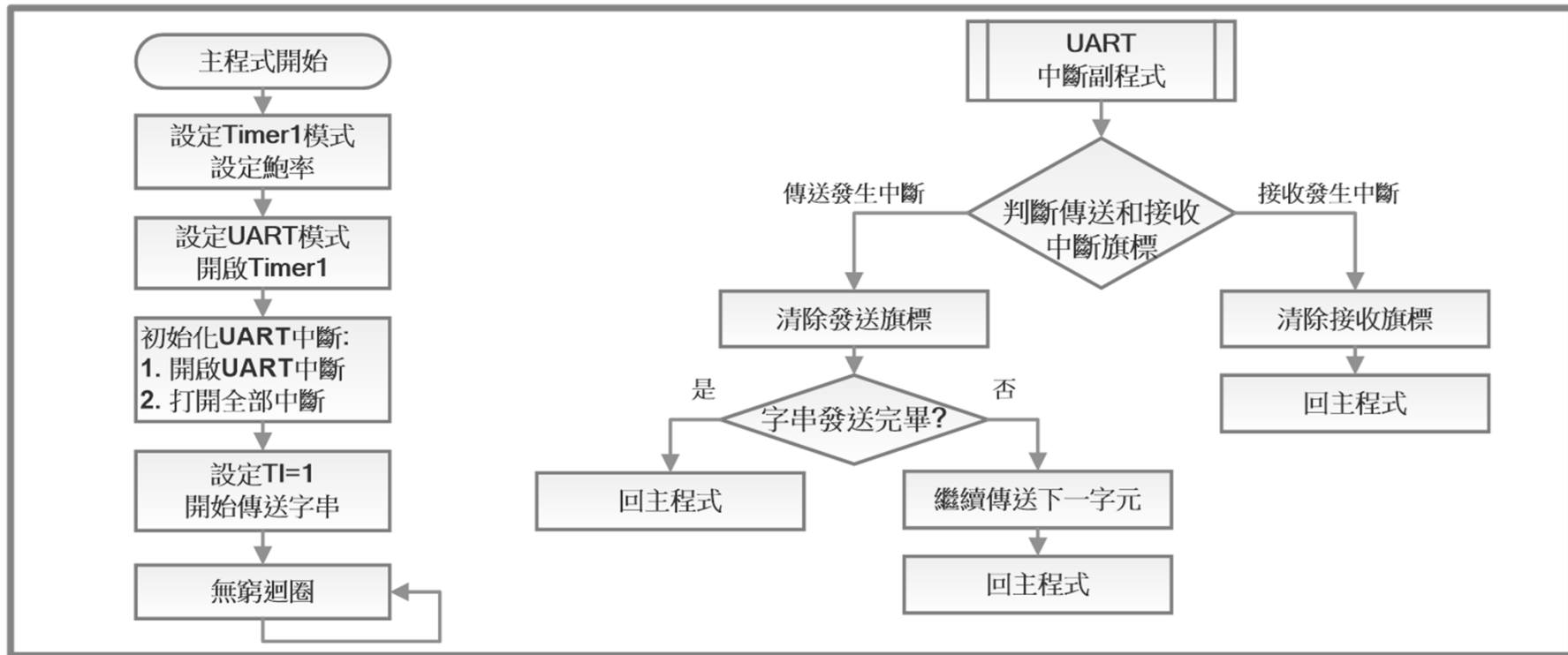


影片

UART中斷設定方式

組合語言	C語言
<pre>;初始化要做的事 MOV TMOD,#20 ;將TIMER1設為模式2, 8bit MOV TH1,#253 ;設定鮑率為9600 MOV SCON,#50H SETB TR1 SETB ES ;打開UART中斷 SETB EA ;打開全部中斷 ;中斷服務要做的事 UART_INT: ;中斷服務程式 JB TI,TI_OK ;判斷發送或接收發生中斷 RI_OK: CLR RI ;清除接收旗標 ;串列接收完成後要做的程式碼 JMP UART_INT_EXIT TI_OK: CLR TI ;清除發送旗標 ;串列發送完成後要做的程式碼 UART_INT_EXIT: RETI</pre>	<pre>//初始化要做的事 TMOD = 20; //將TIMER1設為模式2, 8bit TH1 = 253; //設定鮑率為9600 SMOD = 0x50; TR1 = 1; ES = 1; //打開UART中斷 EA = 1; //打開全部中斷 //中斷服務要做的事 void UART_isr(void) interrupt 4 { if(TI){ TI = 0; //串列發送完成後要做的程式碼 } if(RI){ RI = 0; //串列接收完成後要做的程式碼 } }</pre>

程式流程圖



程式碼(組合語言)

START:	ORG	0		UART_INT:	JB	TI, TI_OK	;判斷中斷旗標
	JMP	START		RI_OK:	CLR	RI	;清除接收中斷旗標
	ORG	23H			JMP	UART_EXIT	
	JMP	UART_INT		TI_OK:	CLR	TI	;清除發送中斷旗標
	MOV	DPTR, #TABLE	;字串查表		MOV	A, R0	;字串若發送完畢
	MOV	R0, #0			CJNE	A, #12, SEND	;不再傳送
	MOV	P0, #0			JMP	UART_EXIT	
	MOV	TMOD, #20H	;設定計時器模式	SEND:	INC	R0	
	MOV	TH1, #253	;設定鮑率為9600bps		MOVC	A, @A+DPTR	;查表字串
	MOV	SCON, #50H	;設定UART模式		MOV	SBUF, A	;發送字串的字元
	SETB	TR1	;啟動Timer1	UART_EXIT:	RETI		
	SETB	ES	;開啟UART中斷開關				
	SETB	EA	;開啟中斷總開關	TABLE:	DB	'H','e','l','l','o',' '	
	SETB	TI	;開始發送字串		DB	'W','o','r','l','d','!'	
	JMP	\$			END		

程式碼(C語言)

```
#include <REGX51.H>
// -----宣告變數與常數-----
code char String[15]="Hello World!" ; //建立字串查表
char i=0;
// -----主程式開始-----
void main(void)
{
    P0=0;
    TMOD=0x20;    //設定計時器T1為模式2
    TH1=0xfd;     //設定鮑率為9600bps
    SCON = 0x50;  //設定UART模式
    TR1=1;        //啟動Timer1
    ES=1;          //開啟Serial中斷
    EA=1;          //開啟中斷總開關
    TI=1;          //開始發送字串
    while(1);
}
```

```
// -----中斷副程式開始-----
void UART_isr(void) interrupt 4
{
    if(TI)                //判斷發送旗標
    {
        TI=0;            //清除發送旗標
        if(i<12)          //判斷字串是否發送完畢
            SBUF=String[i++]; //發送字元後索引值遞增
        else
            i=0;
    }
    if(RI)                //判斷接收旗標
    {
        RI=0;            //清除接收旗標
    }
}
```

單元八：8051的中斷

PART H

微算機原理及應用實習

範例四：串列傳輸(接收)中斷

範例說明

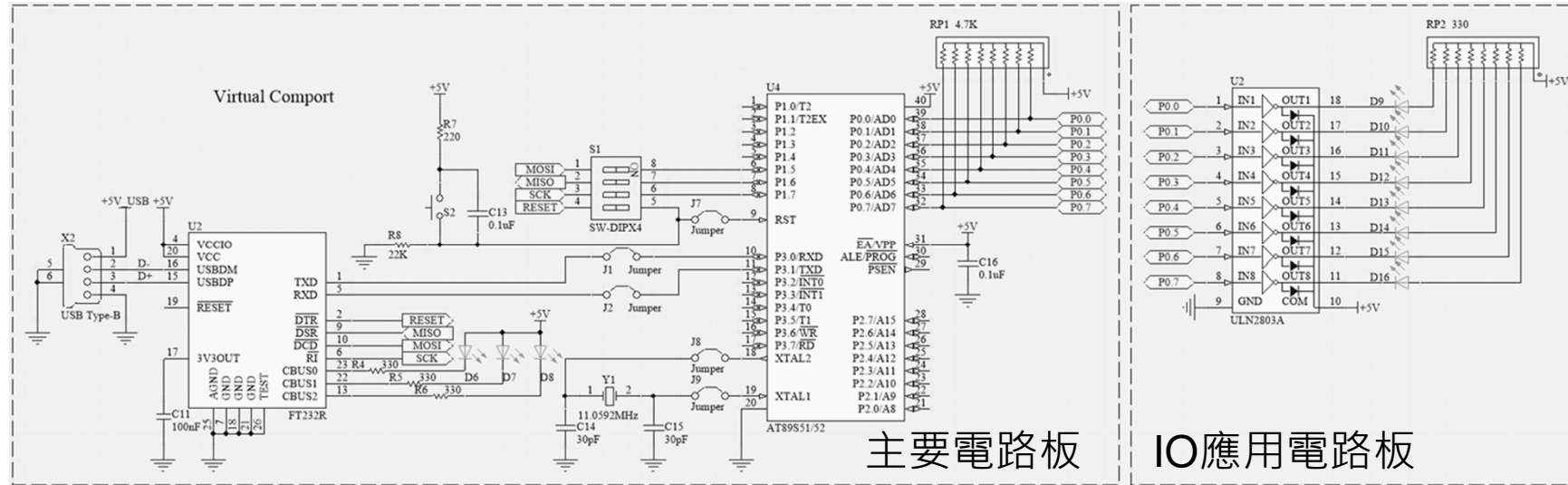
- **實驗目的：**

瞭解 **8051** 之串列傳輸(接收)中斷的控制方法，並練習使用串列傳輸(接收)中斷來接收資料的功能。

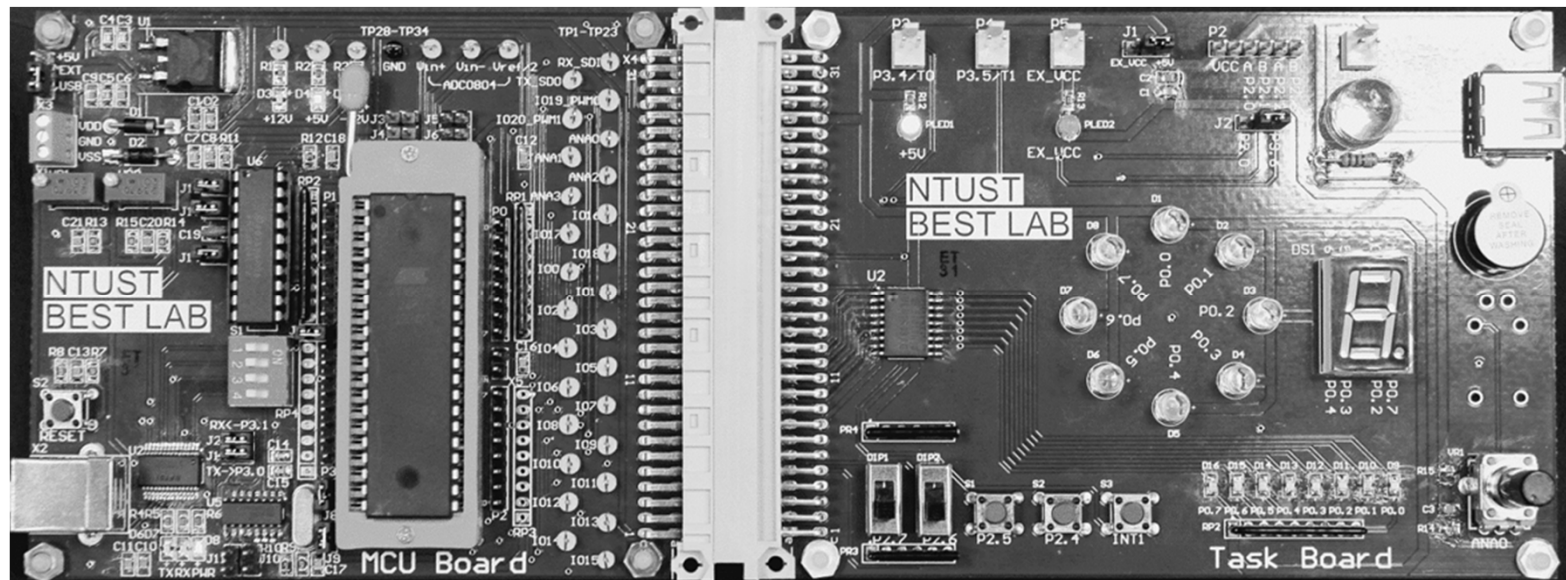
- **功能說明：**

由電腦 **PuTTY** 介面輸入資料，**AT89S51** 接收電腦鍵盤所按下的數字鍵 **ASCII Code** 顯示在 **LED** 上。

電路圖

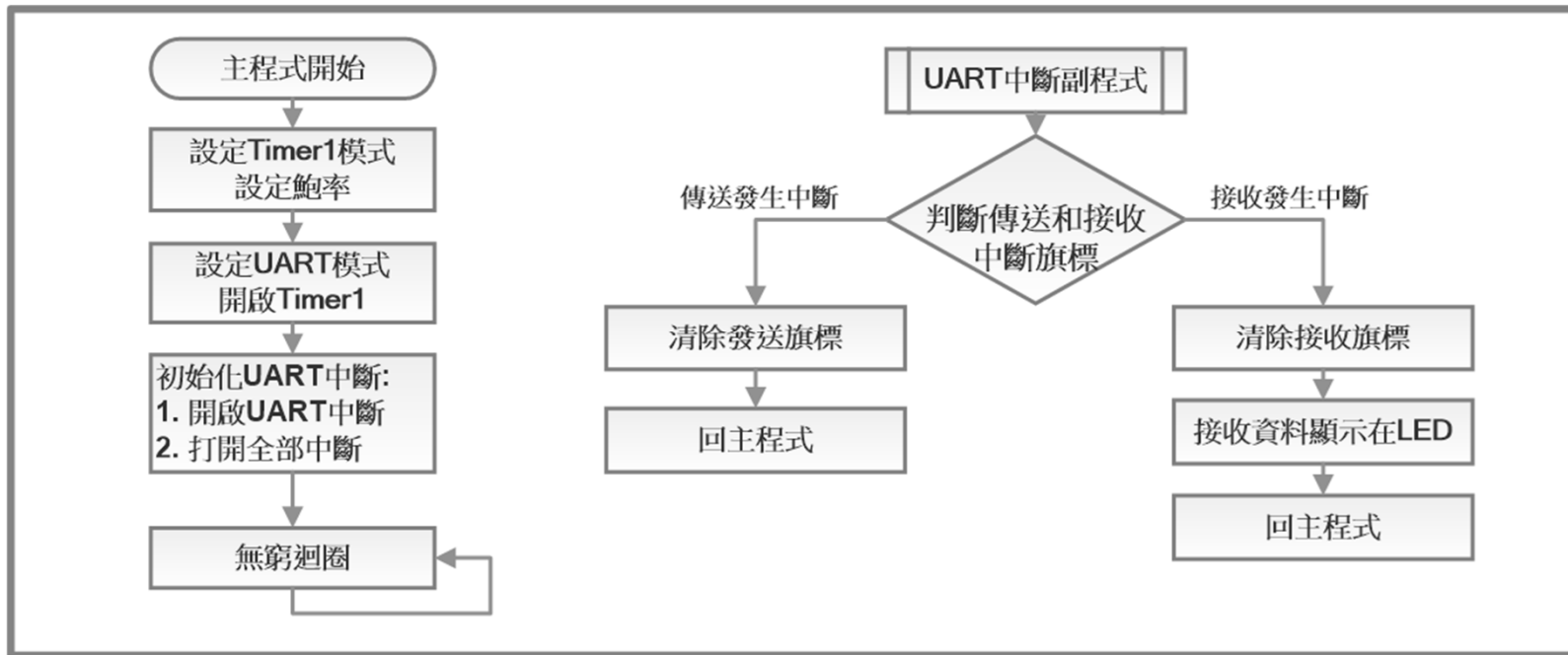


主電路板(MCU Board)與 IO應用電路板(Task board 1)



影片

程式流程圖



程式碼(組合語言)

<pre>ORG 0 JMP START ORG 23H JMP UART_INT START: MOV P0,#0 MOV TMOD,#20H ;設定計時器模式 MOV TH1,#253 ;設定鮑率為9600bps MOV SCON,#50H ;設定UART模式 SETB TR1 ;啟動Timer1 SETB ES ;開啟UART中斷開關 SETB EA ;開啟中斷總開關 JMP \$</pre>	<pre>UART_INT: JB TI,TI_OK RI_OK: CLR RI ;清除接收中斷旗標 MOV P0,SBUF ;電腦鍵盤ASCII Code ;顯示在LED上 JMP UART_EXIT TI_OK: CLR TI ;清除發送中斷旗標 UART_EXIT: RETI END</pre>
---	--

程式碼(C語言)

```
#include <REGX51.H>
// -----主程式開始-----
void main(void)
{
    P0=0;
    TMOD=0x20;    //設定計時器T1模式
    TH1=0xfd;     //設定鮑率為9600bps
    SCON = 0x50;  //設定UART模式
    TR1=1;        //啟動Timer1
    ES=1;         //開啟Serial中斷
    EA=1;         //開啟總中斷
    while(1);
}
```

```
// -----中斷副程式開始-----
void UART_isr(void) interrupt 4
{
    if(TI)        //判斷發送旗標
    {
        TI=0;    //清除發送旗標
    }
    if(RI)        //判斷接收旗標
    {
        RI=0;    //清除接收旗標
        P0=SBUF;  //接收資料並顯示在P0的LED上
    }
}
```

練習題

- **功能說明：**

AT89S51接收由電腦 PuTTY 所按下的按鍵，之後再回傳到 PuTTY 顯示。

RESET 完，PuTTY 先顯示：

“ The keyboard you pressed is ”

例如：按下電腦鍵盤的 “F” ，則 PuTTY 顯示：

“ F ”，然後再顯示 “ The keyboard you pressed is ”

影片

8.6 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.**

