

微介實驗六

中斷

日期：10月27日

報告者：陳韋翰

Outline

- 實驗內容
- 學習重點
- 實驗器材
- 中斷介紹
- 實驗電路
- 軟體流程
- 程式設計

Outline

- 實驗內容
- 學習重點
- 實驗器材
- 中斷介紹
- 實驗電路
- 軟體流程
- 程式設計

實驗內容

利用一顆外接按鈕實現外部中斷，並藉由LED之閃爍呈現結果。

基礎題有兩小題，分別用兩種方式觸發中斷。

(1.負緣觸發2.低準位觸發)

Outline

- 實驗內容
- 學習重點
- 實驗器材
- 中斷介紹
- 實驗電路
- 軟體流程
- 程式設計

學習重點

- 了解中斷用途。
- 認識8051的中斷源以及與中斷相關之暫存器。
- 熟悉8051中斷流程。

Outline

- 實驗內容
- 學習重點
- 實驗器材
- 中斷介紹
- 實驗電路
- 軟體流程
- 程式設計

實驗器材

| 器材名稱 | | 數量 |
|-------------|--------------|----|
| AT89S51 | | 1 |
| 12MHz 石英震盪器 | | 1 |
| LED二極體 | | 8 |
| 按壓開關 | | 3 |
| 電阻 | 1k Ω | 10 |
| | 10k Ω | 1 |
| 電容 | 20pF | 2 |
| | 10 μ F | 1 |

Outline

- 實驗內容
- 學習重點
- 實驗器材
- 中斷介紹
- 實驗電路
- 軟體流程
- 程式設計

中斷vs.輪詢

MCU服務周邊裝置的方法有兩種:

- 中斷(Interrupt):

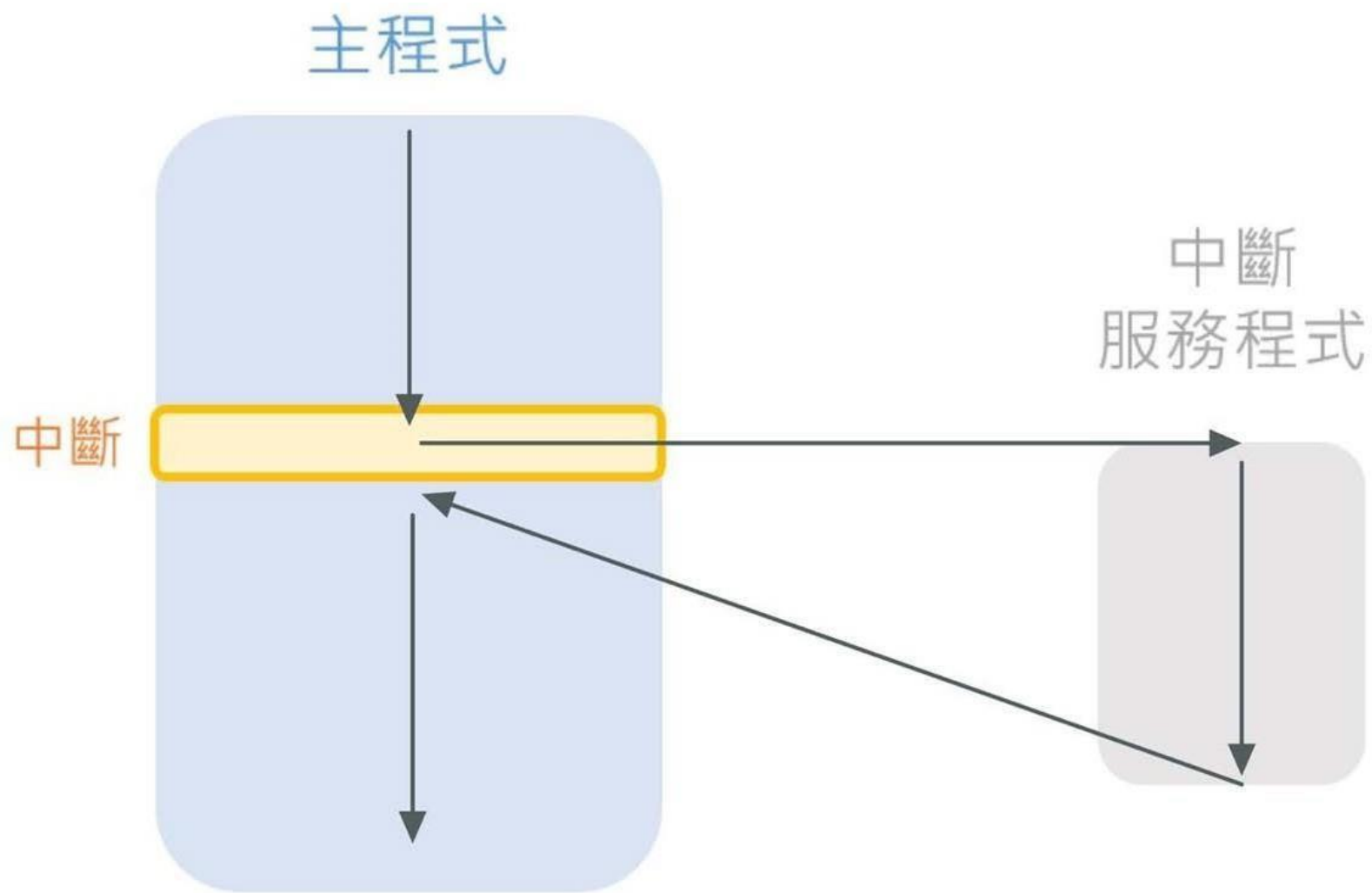
當中斷訊號產生時，主程式會暫停執行，並跳去執行中斷服務程式 (ISR, Interrupt Service Routines)。

若同時有很多中斷訊號產生，則會依照優先權決定順序。

- 輪詢(Polling):

程式不斷詢問某個條件是否達成，當條件達成，就去執行使用者需求。

中斷基本流程



8051的五個中斷源

- INT0：外部中斷
- INT1：外部中斷
- T0：計時/計數器中斷
- T1：計時/計數器中斷
- UART(RI和TI)：串列埠中斷

備註: 8052有六個中斷源(多了T2)

| IC1 | | | |
|-----|-----------|----------|----|
| 1 | P1.0 | VCC | 40 |
| 2 | P1.1 | P0.0_AD0 | 39 |
| 3 | P1.2 | P0.1_AD1 | 38 |
| 4 | P1.3 | P0.2_AD2 | 37 |
| 5 | P1.4 | P0.3_AD3 | 36 |
| 6 | P1.5_MOSI | P0.4_AD4 | 35 |
| 7 | P1.6_MISO | P0.5_AD5 | 34 |
| 8 | P1.7_SCK | P0.6_AD6 | 33 |
| 9 | RST | P0.7_AD7 | 32 |
| 10 | P3.0/RXD | EA/VPP | 31 |
| 11 | P3.1/TXD | ALE/PROG | 30 |
| 12 | P3.2/INT0 | PSEN | 29 |
| 13 | P3.3/INT1 | P2.7_A15 | 28 |
| 14 | P3.4/T0 | P2.6_A14 | 27 |
| 15 | P3.5/T1 | P2.5_A13 | 26 |
| 16 | P3.6/WR | P2.4_A12 | 25 |
| 17 | P3.7/RD | P2.3_A11 | 24 |
| 18 | XTAL2 | P2.2_A10 | 23 |
| 19 | XTAL1 | P2.1_A9 | 22 |
| 20 | GND | P2.0_A8 | 21 |

AT89S51-24P

中斷向量 (interrupt vector)

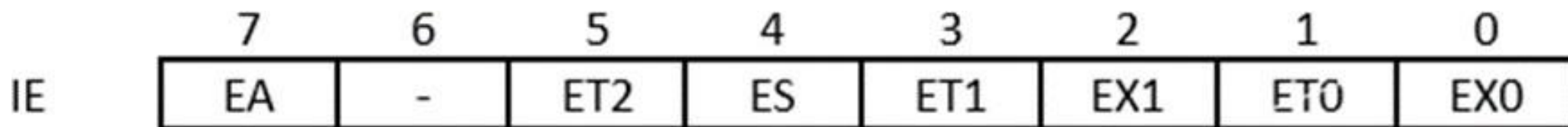
| 中斷源 | 中斷向量(位址值) | 旗標 | 所屬暫存器 |
|-----------|-----------|-----|--------|
| INT0 | 03H | IE0 | TCON.1 |
| T0 | 0BH | TF0 | TCON.5 |
| INT1 | 13H | IE1 | TCON.3 |
| T1 | 1BH | TF1 | TCON.7 |
| UART(RXD) | 23H | RI | SCON.0 |
| UART(TXD) | | TI | SCON.1 |

中斷致能(IE)暫存器

| | | | | | | | | |
|----|----|---|-----|----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IE | EA | - | ET2 | ES | ET1 | EX1 | ET0 | EX0 |

| | | |
|------|-----|------------------|
| IE.7 | EA | 中斷總開關。 |
| IE.6 | - | 未使用。 |
| IE.5 | ET2 | T2之中斷開關(8052才有)。 |
| IE.4 | ES | 串列埠之中斷開關。 |
| IE.3 | ET1 | T1之中斷開關。 |
| IE.2 | EX1 | INT1之中斷開關。 |
| IE.1 | ET0 | T0之中斷開關。 |
| IE.0 | EX0 | INT0之中斷開關。 |

中斷致能(IE)暫存器



- reset後IE值均為0，若設定為1則可啟用該中斷。
- IE暫存器為可位元定址(bit addressable)。

計時/計數器控制(TCON)暫存器

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

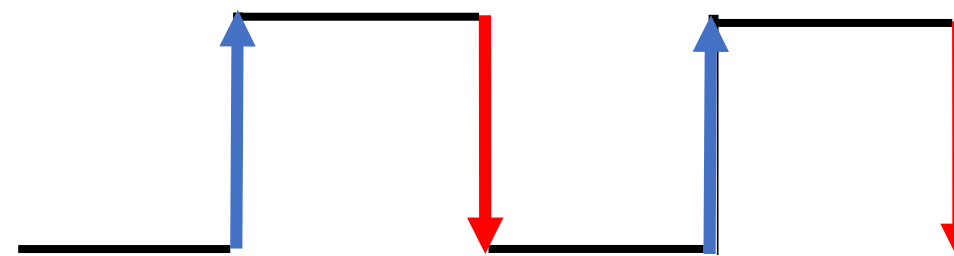
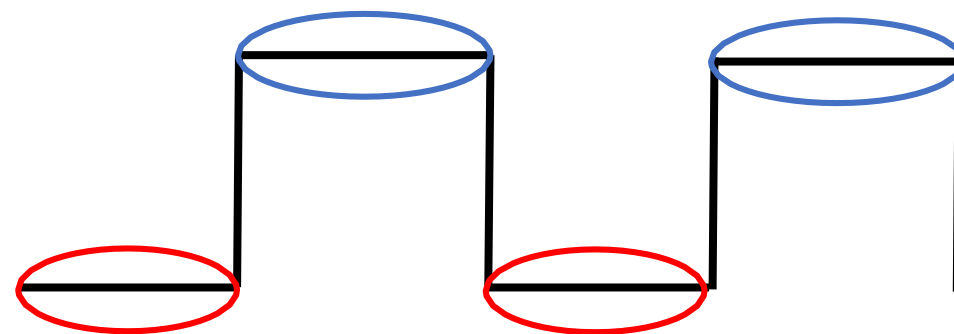
| | | |
|--------|-----|-----------------------------------|
| TCON.7 | TF1 | 計時/計數器功能。 (實驗七-計時/計數器會做介紹) |
| TCON.6 | TR1 | |
| TCON.5 | TF0 | |
| TCON.4 | TR0 | |
| TCON.3 | IE1 | INT1之中斷旗標。(產生中斷時IE變為1) |
| TCON.2 | IT1 | INT1之觸發模式。(預設0 時為低準位觸發，設1 時為負緣觸發) |
| TCON.1 | IE0 | INT0之中斷旗標。(產生中斷時IE變為1) |
| TCON.0 | IT0 | INT0之觸發模式。(預設0 時為低準位觸發，設1 時為負緣觸發) |

外部中斷 (INT0、INT1)

- 8051的訊號的採樣方式可分為低準位觸發以及負緣觸發。(每一個 machine cycle 採樣一次)
- 進入中斷服務程式後，IE0(TCON.1)與IE1(TCON.3)的值 會被CPU清除。

外部中斷 (INT0、INT1)

- 準位觸發(level-triggered)
 - 高準位觸發high level-triggered
 - 低準位觸發low level-triggered
- 邊緣觸發(edge-triggered)
 - 正緣觸發rising edge-triggered
 - 負緣觸發falling edge-triggered



外部中斷 (INT0、INT1)

CPU透過INT0、INT1接腳接受外部中斷需求。

➤INT0接腳(Pin12，與Port3.2共用)

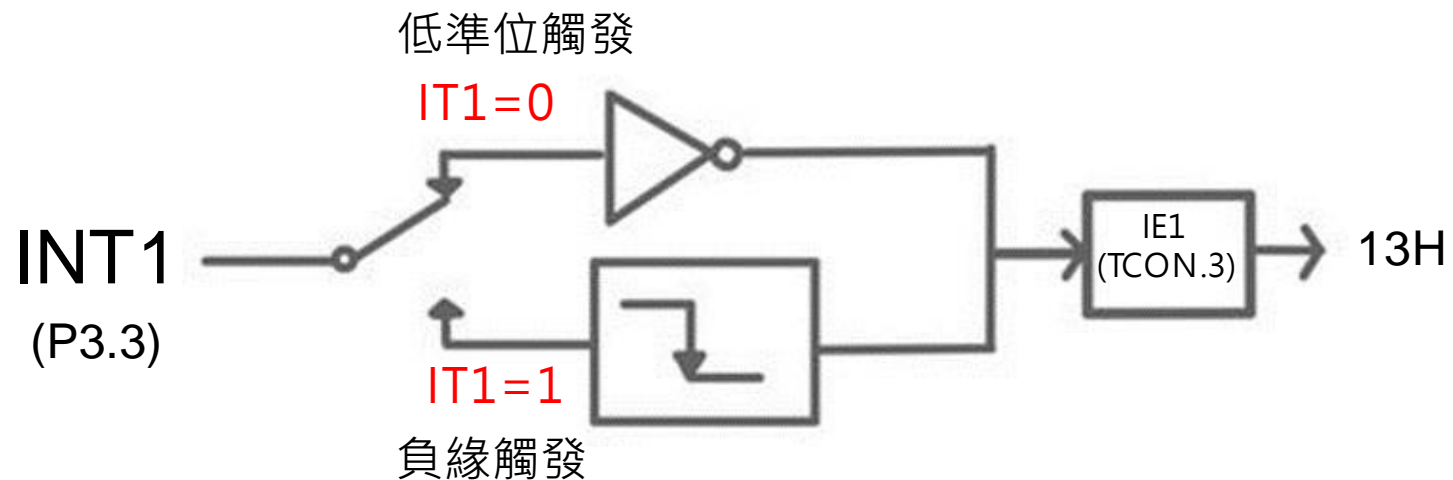
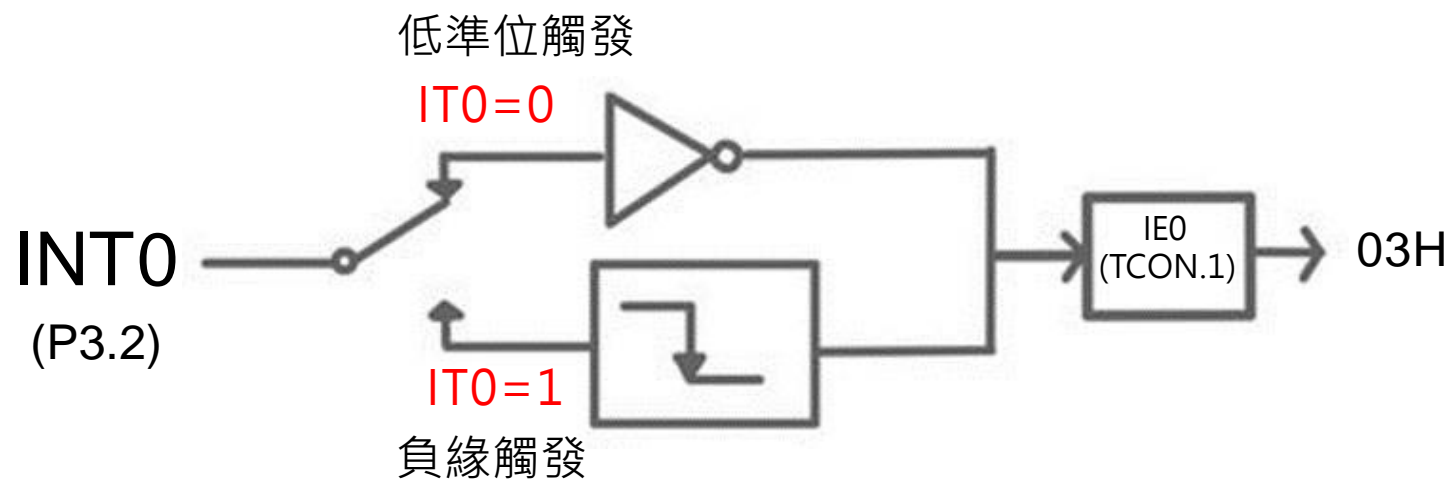
此腳位接收到low level或falling edge訊號時，IE0會被CPU自動設為1，產生外部中斷。

➤INT1接腳(Pin13，與Port3.3共用)

此腳位接收到low level或falling edge訊號時，IE1會被CPU自動設為1，產生外部中斷。

外部中斷 (INT0、INT1)

控制流程



串列埠中斷 (UART)

CPU透過RXD、TXD接腳接受或傳送中斷需求。

➤ RXD接腳(Pin10，與Port3.0共用)

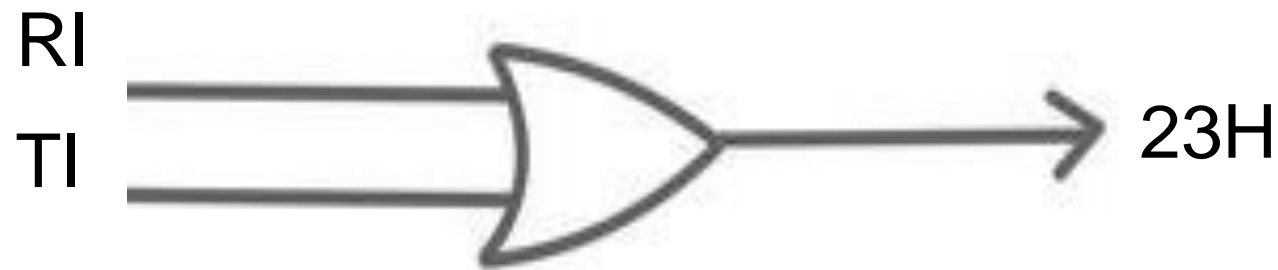
對應旗標為RI(Receive Interrupt)，接受中斷需求。

➤ TXD接腳(Pin11，與Port3.1共用)

對應旗標為TI(Transfer Interrupt)，傳送中斷需求。

串列埠中斷(UART)

- 當其對應旗標TI或RI設定為1後，中斷服務程式將會執行。
- 共用一個向量，ISR必須自己判斷是TI或RI造成中斷。
- TI變成1表示資料傳完;RI變成1表示接收到資料。
- CPU不會自動清除TI與RI。



中斷優先等級(IP)暫存器

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|---|---|-----|----|-----|-----|-----|-----|
| IP | - | - | PT2 | PS | PT1 | PX1 | PT0 | PX0 |

| | | |
|------|-----|---------------------|
| IP.7 | - | 未使用。 |
| IP.6 | - | 未使用。 |
| IP.5 | PT2 | TF2之中斷優先等級。(8052才有) |
| IP.4 | PS | 串列埠之中斷優先等級。 |
| IP.3 | PT1 | TF1之中斷優先等級。 |
| IP.2 | PX1 | INT1之中斷優先等級。 |
| IP.1 | PT0 | TF0之中斷優先等級。 |
| IP.0 | PX0 | INT0之中斷優先等級。 |

IE暫存器與IP暫存器位置相對應

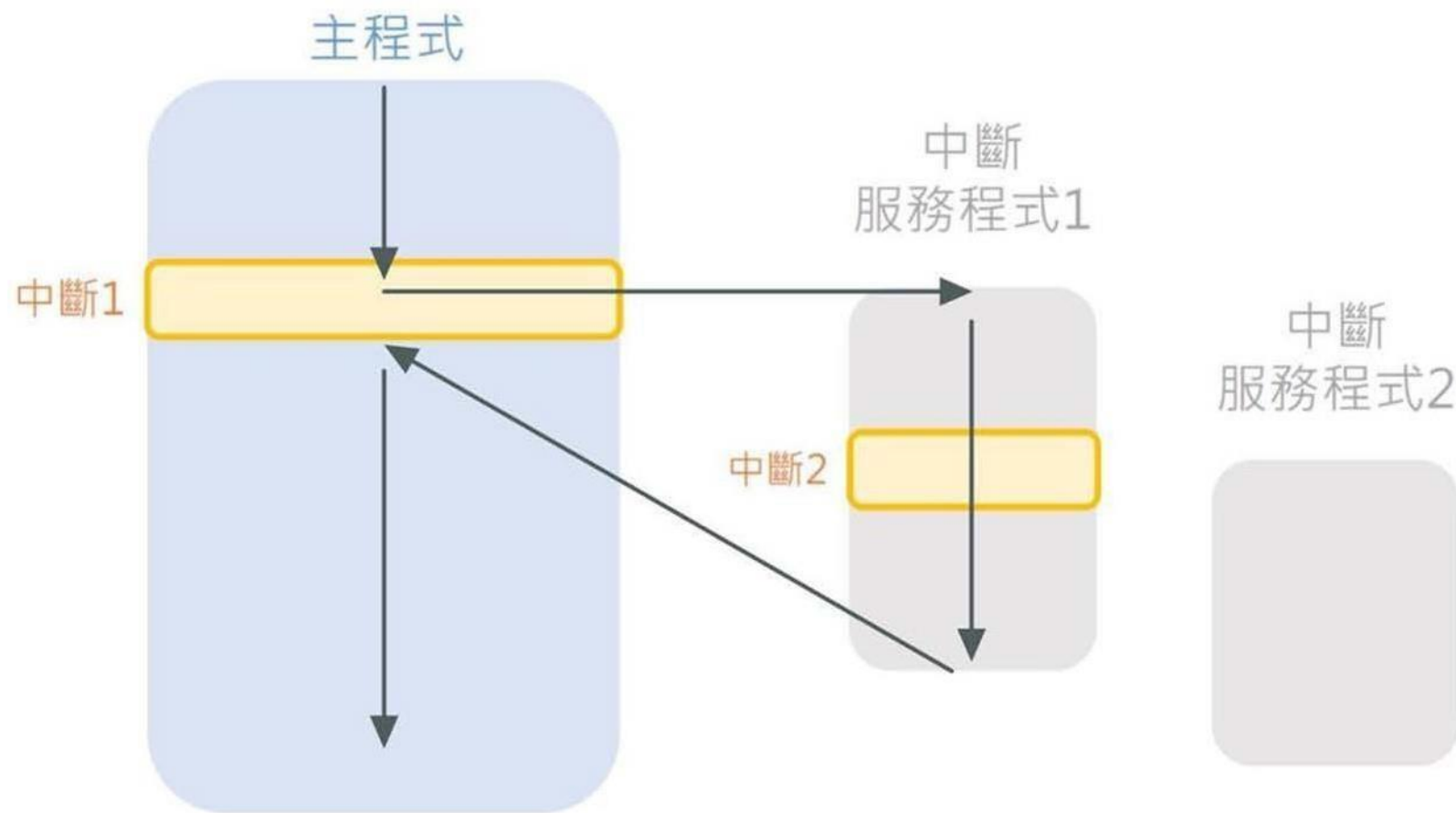
| | | | | | | | | |
|----|----|---|-----|----|-----|-----|-----|-----|
| IE | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EA | - | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| IP | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | - | - | PT2 | PS | PT1 | PX1 | PT0 | PX0 |

中斷優先等級(IP)暫存器

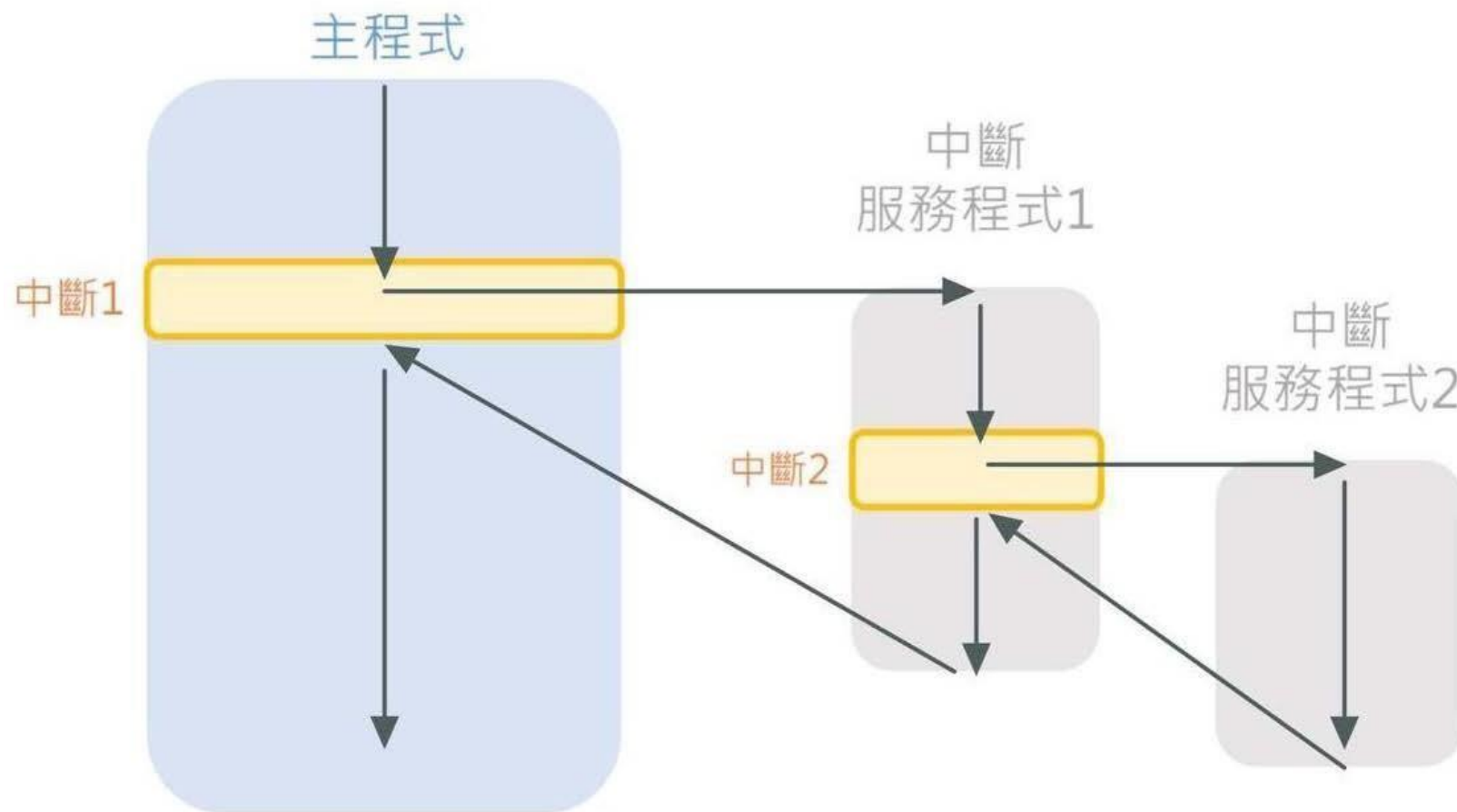
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|---|---|-----|----|-----|-----|-----|-----|
| IP | - | - | PT2 | PS | PT1 | PX1 | PT0 | PX0 |

- reset後IP值均為0，代表為相同的低層次優先權，若設定為1則會變為高層次優先權。
- 相同層次仍有預設優先順序：INT0>TF0>INT1>TF1>RI/TI
- IP暫存器為可位元定址(bit addressable)。

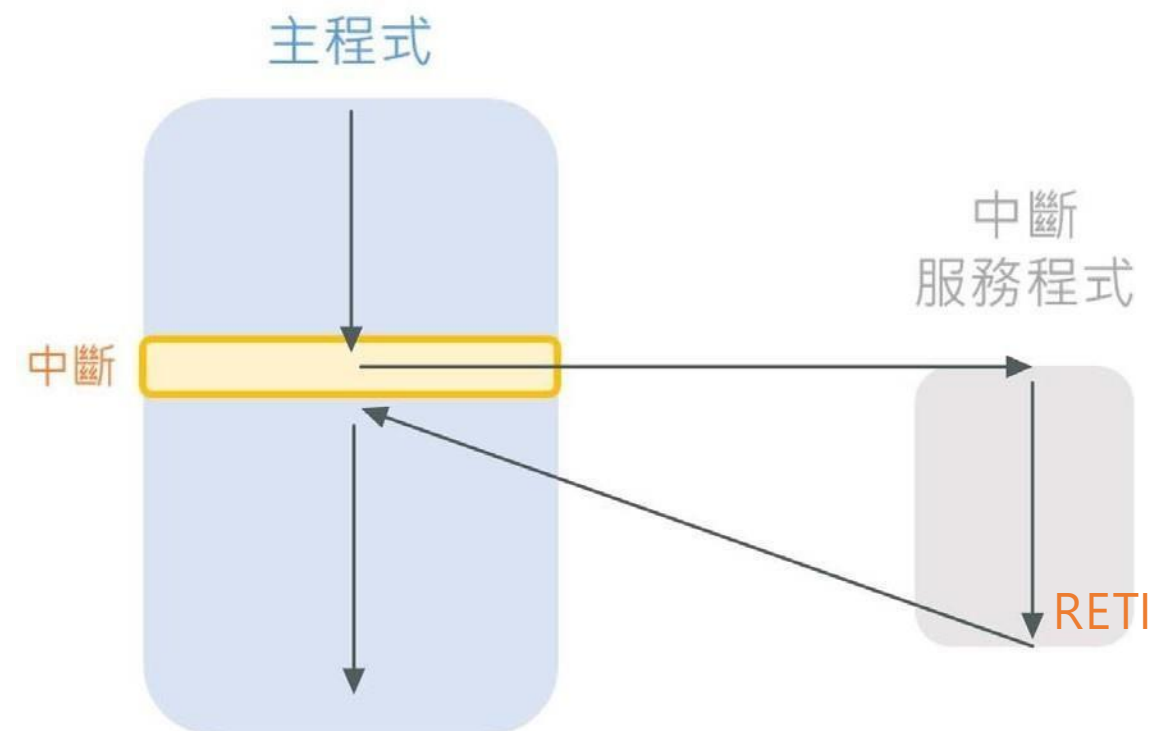
中斷2的優先層次低於或等於中斷1時



中斷2的優先層次高於中斷1時



中斷流程



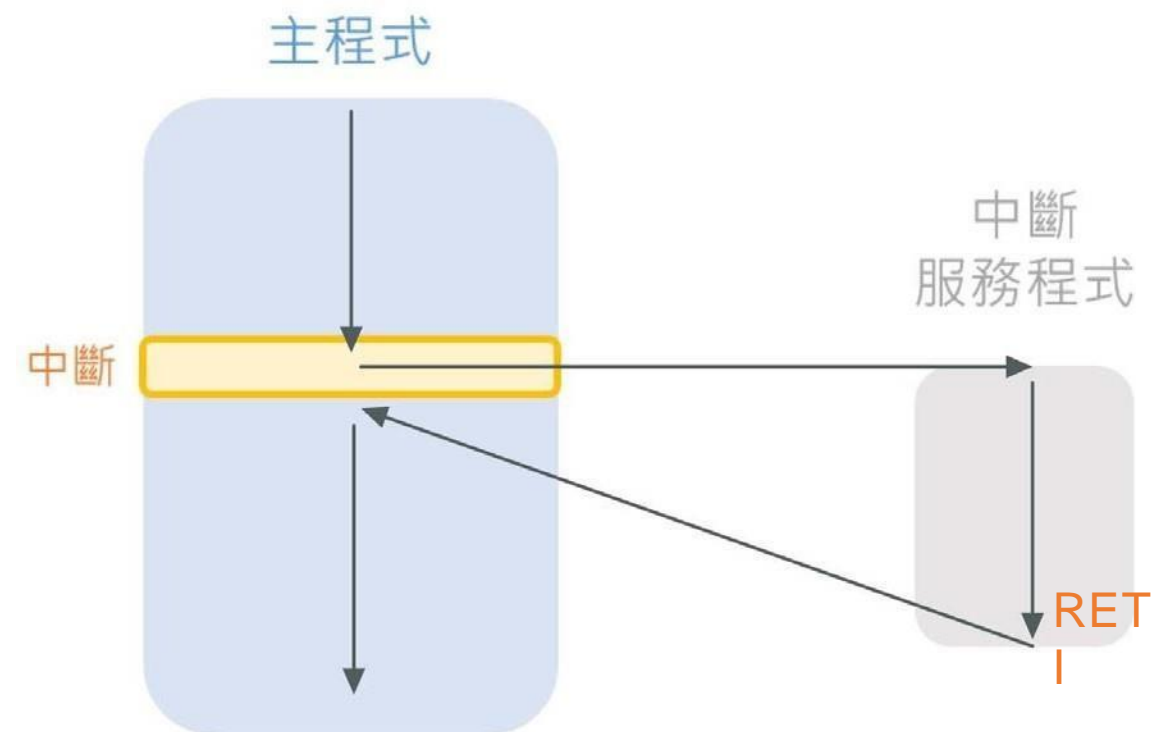
中斷發生時

1. PC值會被push進stack
2. PC值會被設成相對應的中斷向量位址

執行RETI來結束中斷服務程式

1. return address會被從stack中pop到PC
2. 繼續執行中斷前的主程式

中斷流程



進入中斷服務程式後，依照應用情況，必要時會將主程式的資料push進stack(例如:PSW、ACC等)。

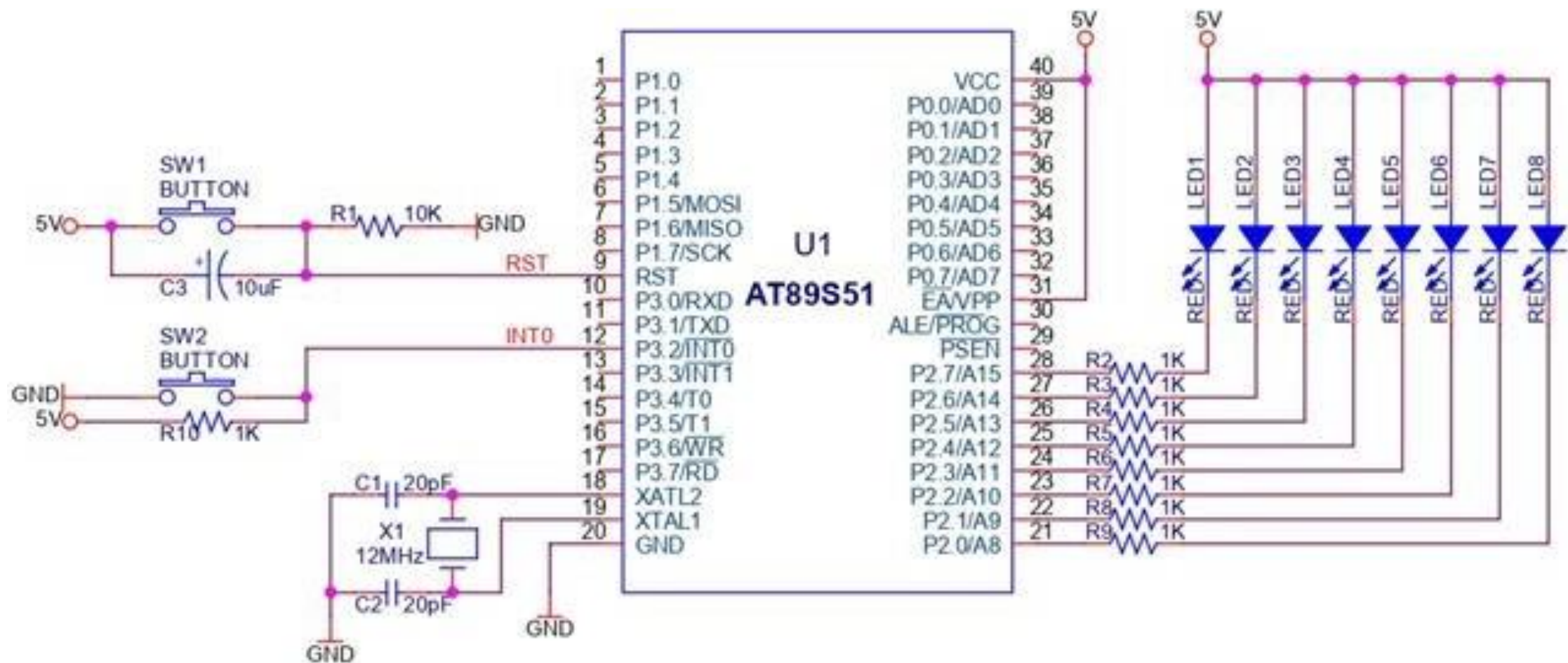
若中斷服務程式中會使用到同樣的暫存器，也會在進入中斷服務程式後切換暫存器庫。

結束中斷服務程式前，須將原本主程式的狀態從stack中pop出來。

Outline

- 實驗內容
- 學習重點
- 實驗器材
- 元件原理
- **實驗電路**
- 軟體流程
- 程式設計

實驗電路

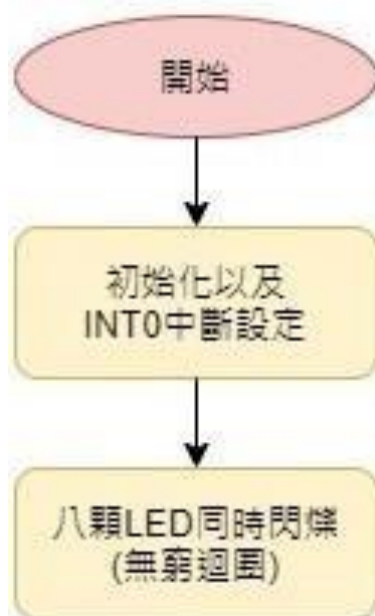


Outline

- 實驗內容
- 學習重點
- 實驗器材
- 中斷介紹
- 實驗電路
- 軟體流程
- 程式設計

軟體流程

主程式:



中斷服務程式:



Outline

- 實驗內容
- 學習重點
- 實驗器材
- 中斷介紹
- 實驗電路
- 軟體流程
- 程式設計

程式設計

| | | |
|-----------|-------------------|---------------------------|
| 1. | ORG 00H | ;code start from 00H |
| 2. | SJMP MAIN | ;jump to MAIN |
| 3. | ORG 03H | ;vector address for INTO |
| 4. | SJMP INTO_ISR | ;jump to INTO_ISR |
| 5. | ORG 030H | ;after vector table space |
| 6. MAIN: | MOV IE,#10000001B | ;enable EA and EX0 |
| 7. | MOV SP,#30H | ;stack start from #30H |
| 8. | SETB IT0 | ;falling edge-triggered |
| 9. | MOV A,#00000000B | ;set ACC as 00000000B |
| 10. LOOP: | MOV P2,A | ;P1 = A (LED output) |
| 11. | CALL DELAY | ;call delay function |
| 12. | CPL A | ;reverse A |
| 13. | SJMP LOOP | ;infinite loop |

程式設計- 中斷服務程式

| | | |
|----------------------|--------------------------|------------------------|
| 14. INT0_ISR: | PUSH PSW | ;push PSW into stack |
| 15. | PUSH ACC | ;push ACC into stack |
| 16. | SETB RS0 | ;switch to RB1 |
| 17. | MOV A, #11111110B | ;set ACC as 11111110B |
| 18. | MOV R0, #24 | ;loop counter = 24 |
| 19. ROTATE_L: | MOV P2, A | ;P2 = A (LED output) |
| 20. | CALL DELAY | ;call delay function |
| 21. | RL A | ;rotate left |
| 22. | DJNZ R0, ROTATE_L | ;loop until R0 is 0 |
| 23. | POP ACC | ;pop out ACC fromstack |
| 24. | POP PSW | ;pop out PSW fromstack |
| 25. | RETI | ;return from ISR |

程式設計

| | | | |
|-----|--------|-------------|---------------|
| 26. | DELAY: | MOV R7,#200 | |
| 27. | D1: | MOV R6,#250 | |
| 28. | | DJNZ R6,\$ | |
| 29. | | DJNZ R7,D1 | |
| 30. | | RET | ;return |
| 31. | | END | ;end the code |

Q&A