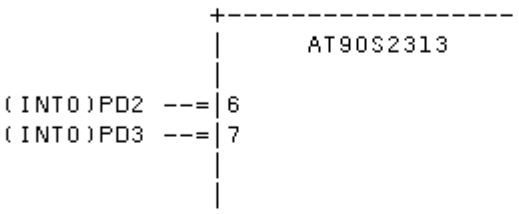
+-----+
| MCU | การอินเตอร์รัพท์จากภายนอก (external interrupt)
| U07 | โดย อ.กำธร เรือนฝายกาศ
+-----+

(1) กล่าวโดยทั่วไป

AT90S2313 จะรับรู้การอินเตอร์รัพท์จากภายนอกได้โดยทางขา 6 และ 7 ซึ่งเป็นขา external interrupt INT0 และ INT1 ตามลำดับ อินเตอร์รัพท์ทั้งสองนี้จะมีอินเตอร์รัพท์เวคเตอร์อยู่ที่ตำแหน่งไบต์ที่ 1 และ 2 ใน sram ตามลำดับ การอินเตอร์รัพท์จากภายนอกสามารถนำไปประยุกต์ใช้ได้หลากหลาย เช่น เรามีสวิตช์สำหรับให้ผู้ใช้ กด แล้วให้ฮาร์ดแวร์ทำงานทันทีที่กด ถ้าใช้วิธี polling คือใช้โปรแกรมมาวนดูสวิตช์ ก็อาจจะพลาดการตรวจสอบไปได้ หากคนที่กดสวิตช์ กดแล้วปล่อยเร็วเกินไป แต่ถ้าใช้การอินเตอร์รัพท์จากภายนอกจะไม่เกิดปัญหานี้ขึ้น เพราะ MCU จะกระโดดไปทำงานที่ต้องการได้ทันทีเมื่อสวิตช์ถูกกด เป็นต้น



ขาสำหรับการอินเตอร์รัพท์จากภายนอก จะเป็นขาเดียวกันกับ PD2 และ PD3 คือขา 6 และ 7 ของไอซี การเขียนโปรแกรมเพื่อสั่งการให้ MCU รองรับงานอินเตอร์รัพท์จากภายนอกมีขั้นตอนดังนี้

1) ให้ pullup ขา INT ที่ต้องการใช้งานด้วยคำสั่ง sbi ดังนี้

```
sbi     PORTD,  2               ; pullup INT0 pin
sbi     PORTD,  3               ; pullup INT1 pin
```

ในตัวอย่างนี้เราใช้งาน INT0 และ INT1 พร้อมกัน จึง pullup ทั้งสองขา แต่ถ้าใช้ขาใดขาหนึ่งก็สั่งเพียงขาเดียว

2) เลือกว่าจะให้อินเตอร์รัพท์ภายนอกเกิดขึ้นที่ขอบขาขึ้น ขอบขาลง หรือลอจิก 0 โดยการตั้งค่ารีจิสเตอร์ MCUCR ซึ่งการตั้งสำหรับ INT0 และ INT1 จะเป็นอิสระจากกัน คือ เราจะให้ INT0 ทำงานที่ขอบขาขึ้น แล้วให้ INT1 ทำงานที่ขอบขาลงก็ได้

+========+
| MCUCR | MCU Control Register
+========+

bit 7	6	5	4	3	2	1	0
-	-	SE	SM	ISC11	ISC10	ISC01	ISC00

การตั้งค่าจะใช้ ISC (Interrupt Sense Control) ดังนี้

ISC11,ISC10 สำหรับควบคุม INT1
ISC01,ISC00 สำหรับควบคุม INT0

โหมดสำหรับการควบคุมมี 3 โหมดคือ ลอจิก 0 ขอบขาลง และขอบขาขึ้น

For INT1

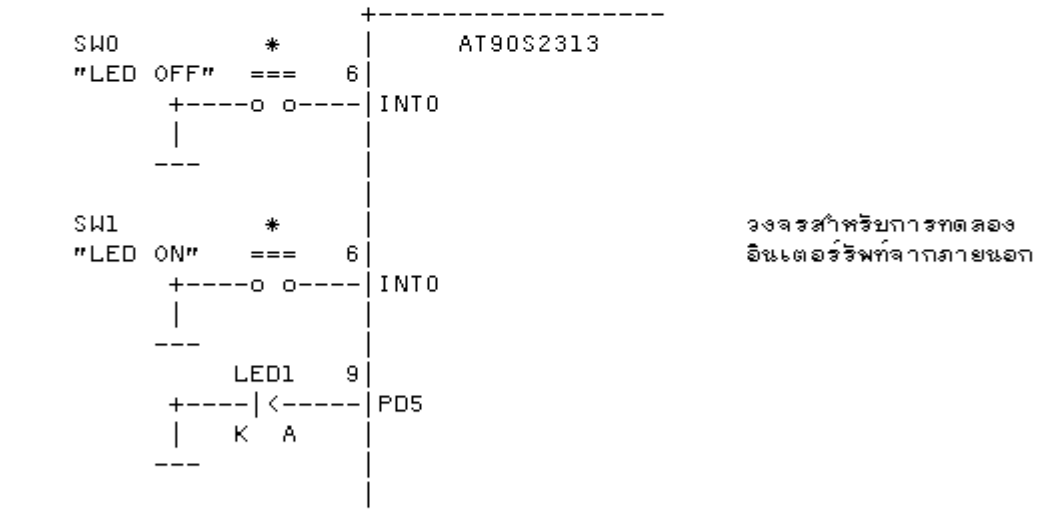
ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request
0	1	Reserved
1	0	The falling edge of INT1 generates an interrupt request
1	1	The rising edge of INT1 generates an interrupt request

For INT0

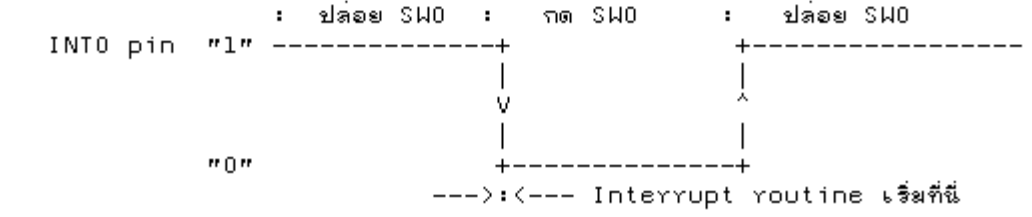
ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request
0	1	Reserved
1	0	The falling edge of INT0 generates an interrupt request
1	1	The rising edge of INT0 generates an interrupt request

ในที่นี้เราจะให้อินเตอร์รัพท์ทั้งสองทำงานที่ขอบขาลงของพัลส์ จึง
ตั้งรีจิสเตอร์ MCUCR ดังนี้

```
ldi     r16,    0b00001010      ; falling edge generates an interrupt
out     MCUCR,  r16             ; -
```



จะทำให้เกิดการอินเตอร์รัพท์ที่ขอบขาลง ในสภาวะปกติถ้าเราไม่กดสวิตช์
SW0 ลอจิกที่ขา INT0 จะเป็นลอจิก 1 ถ้ากดสวิตช์ SW0 ขา INT0 จะเปลี่ยน
เป็นลอจิก 0 ช่วงขอบขาลงของพัลส์ที่ขา INT0 จะทำให้ external
interrupt เริ่มทำงาน ขา INT1 ก็ทำงานในลักษณะเดียวกัน



3) ทำการ enable ให้ INT0 และ INT1 ทำงานโดยตั้งค่ารีจิสเตอร์ GIMSK ดังนี้

```
ldi     r16,    0b11000000      ; enable INT1 and INT0
out     GIMSK,  r16             ; -
```

```
+========+
| GIMSK | General Interrupt Mask Register
+========+
bit    7      6      5      4      3      2      1      0
     +-------------------------------------------------------+
     | INT1 | INT0 |  -   |  -   |  -   |  -   |  -   |  -   |
     +-------------------------------------------------------+
R/W   R/W    R/W     R      R      R      R      R      R
Init   0      0      0      0      0      0      0      0
Value
```

ถ้าต้องการให้อินเตอร์รัพท์ใดทำงานก็ให้บิตนั้นเป็น 1 ในที่นี้เราเซ็ท บิต 7 และ 6 เป็น 1 เพื่อให้ INT1 และ INT0 ทำงาน

4) ให้ global interrupt enable (I-bit) เป็นลอจิก 1 ด้วยคำสั่ง sei

```
sei                             ; enble global interrupt
```

(2) ฟังค์ชั่นรองรับการอินเตอร์รัพท์(Interrupt service routine)
ในการทดลองคราวนี้เราเขียน ฟังค์ชั่นรองรับอินเตอร์รัพท์ทั้งสองเบอร์ ดังนี้

- <> ฟังค์ชั่น I_LED_OFF สำหรับ INT0
- <> ฟังค์ชั่น I_LED_ON สำหรับ INT1

1) เราให้ตารางอินเตอร์รัพท์เบอร์ #1 และ #2 ชี้ไปที่ฟังค์ชั่นทั้งสองนี้ ด้วยคำสั่ง rjmp I_LED_OFF และ rjmp I_LED_ON

```
                                ; ======================
                                ; interrupt vector table
                                ; ======================
rjmp    F_main                  ; #0 RESET
rjmp    I_LED_OFF               ; #1 INT0
rjmp    I_LED_ON                ; #2 INT1
reti                            ; #3 TIMER1 CAPT11
reti                            ; #4 TIMER1 COMP1
reti                            ; #5 TIMER1 OVF1
reti                            ; #6 TIMER0 OVF0
reti                            ; #7 UART,RX
reti                            ; #8 UART,UDRE
reti                            ; #9 UART,TX
reti                            ; #10 ANA_COMP
```

2) เมื่อมี INT0 เข้ามา MCU จะกระโดดไปทำงานที่ฟังค์ชั่น I_LED_OFF คำสั่ง cli จะป้องกันไม่ให้เกิดการอินเตอร์รัพท์ซ้ำเข้ามาในฟังค์ชั่นนี้ จากนั้นสั่งให้ PD5 เป็น output port ด้วยคำสั่ง sbi DDRD,5 และส่งลอจิก 1 ออกไปที่ PD5 ด้วยคำสั่ง sbi PORTD,5 เพื่อให้ LED ติด เมื่อเสร็จแล้วก็ปิดท้ายด้วยคำสั่ง reti (interrupt return)

```
;=============================================================================
; Interrupt routine : LED_ON
;=======================================
I_LED_ON:                               ;
        cli                             ; disable Global interrrupt
        sbi     DDRD,   5               ; PD5 = output
        sbi     PORTD,  5               ; PD5 = 1
        reti                            ;
;.............................................................................
```

3) ฟังค์ชั่น I_LED_OFF ทำงานคล้ายกับ I_LED_ON โดยเมื่อมี INT1 เข้ามา MCU จะกระโดดไปทำงานที่ฟังค์ชั่น I_LED_ON คำสั่ง cli จะป้องกันไม่ให้เกิดการอินเตอร์รัพท์ซ้ำเข้ามาในฟังค์ชั่นนี้ จากนั้นสั่งให้ PD5 เป็น output port ด้วยคำสั่ง sbi DDRD,5 และ ส่งลอจิก 0 ออกไปที่ PD5 เพื่อให้ LED ดับ ด้วยคำสั่ง cbi PORTD,5 เมื่อเสร็จแล้วก็ปิดท้ายด้วยคำสั่ง reti (interrupt return)

```
;=============================================================================
; Interrupt routine : LED_OFF
;=======================================
I_LED_OFF:                              ;
        cli                             ; disable Global interrrupt
        sbi     DDRD,   5               ; PD5 = output
        cbi     PORTD,  5               ; PD5 = 0
        reti                            ;
;.............................................................................
```

(3) โปรแกรม

```
;*****************************************************************************
; AT90S2313
; External Interrupt Experiment
; copyright (c) 2001 by Gumtorn Ruanfaigad.Allright reserved.
; Electronics devision.RIT Chiangmai campus.
; ~ XTAL        = 4.0MHz
; ~ Ucc         = 5v
; ~ MCU         = AT90S2313,Atmel corperation,www.atmel.com
;*****************************************************************************

.include        "2313def.inc"           ;
.DEVICE         AT90S2313               ;
.CSEG                                   ;
.ORG            0x0000                  ;
                                        ; ======================
                                        ; interrupt vector table
                                        ; ======================
        rjmp    F_main                  ; #0 RESET
        rjmp    I_LED_OFF               ; #1 INT0
        rjmp    I_LED_ON                ; #2 INT1
        reti                            ; #3 TIMER1 CAPT11
        reti                            ; #4 TIMER1 COMP1
        reti                            ; #5 TIMER1 OVF1
        reti                            ; #6 TIMER0 OVF0
        reti                            ; #7 UART,RX
        reti                            ; #8 UART,UDRE
        reti                            ; #9 UART,TX
        reti                            ; #10 ANA_COMP
```

```
;***************************************************************************
;void main(void)
;***************************************
F_Main:                                 ; main function start here
        ldi     r16,    low(RAMEND)     ; set stack pointer
        out     SPL,    r16             ; -
        sbi     PORTD,  2               ; pullup INT0 pin
        sbi     PORTD,  3               ; pullup INT1 pin
        ldi     r16,    0b00001010      ; falling edge generates an interrupt
        out     MCUCR,  r16             ; -
        ldi     r16,    0b11000000      ; enable INT1 and INT0
        out     GIMSK,  r16             ; -
        sei                             ; enble global interrupt
L_MainLoop:                             ;
        rjmp            L_MainLoop      ;
;.........................................................................

;***************************************************************************
; Interrupt routine : LED_ON
;***************************************
I_LED_ON:                               ;
        cli                             ; disable Global interrrupt
        sbi     DDRD,   5               ; PD5 = output
        sbi     PORTD,  5               ; PD5 = 1
        reti                            ;
;.........................................................................

;***************************************************************************
; Interrupt routine : LED_OFF
;***************************************
I_LED_OFF:                              ;
        cli                             ; disable Global interrrupt
        sbi     DDRD,   5               ; PD5 = output
        cbi     PORTD,  5               ; PD5 = 0
        reti                            ;
;.........................................................................
```