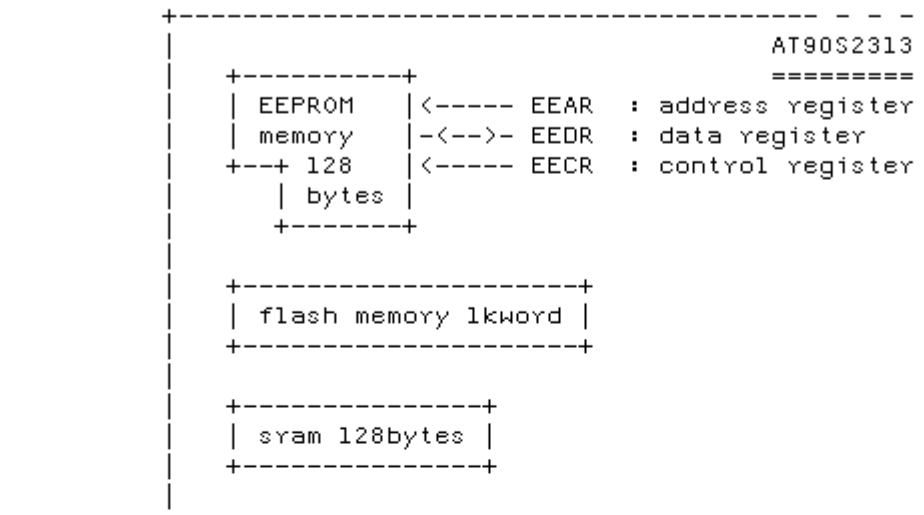
+-----+
| MCU | การเขียนและการอ่าน EEPROM
| U08 | โดย อ.กำธร เรือนฝายกาศ
+-----+

(1) กล่าวโดยทั่วไป



ไอซี AT90S2313 ประกอบด้วยหน่วยความจำภายใน สามส่วนคือ
1. sram หน่วยความจำ static ram มี 128 ไบต์ โปรแกรม สามารถอ่านหรือเขียนข้อมูลได้ (ไฟดับข้อมูลหาย)
2. flash memory มี 1 กิโลเวิร์ด โปรแกรมสามารถอ่านข้อมูล จากหน่วยความจำส่วนนี้ได้ แต่เขียนไม่ได้ (ไฟดับข้อมูลไม่หาย)
3. EEPROM มี 128 ไบต์ โปรแกรมสามารถเขียนหรืออ่านข้อมูล จากหน่วยความจำส่วนนี้ได้ผ่านทางรีจิสเตอร์ EERA,EEDR และ EECR (ไฟดับข้อมูลไม่หาย)

(2) EEPROM Read/Write Access

ในบางงานที่ให้มีการตั้งค่าต่างๆ เช่นการวัดอุณหภูมิ เราต้องการบันทึกค่า อุณหภูมิสูงสุด และต่ำสุดไว้ EEPROM เหมาะสมสำหรับงานนี้ เพราะว่า หากไฟเลี้ยงวงจรดับไป ข้อมูลที่เก็บไว้จะยังคงอยู่ ในที่นี้เราจึงจะทดลอง อ่านและเขียนข้อมูลในส่วนของ EEPROM รายละเอียดของรีจิสเตอร์ที่ เกี่ยวข้องกับการเขียนอ่านมี 3 รีจิสเตอร์ดังนี้

1. EERA (Address register) ทำหน้าที่กำหนดตำแหน่งของหน่วย ความจำที่เราต้องการเขียนหรืออ่าน มีค่าได้ตั้งแต่ 0 ถึง 127 หรือ 0000000 ถึง 1111111 ในเลขฐานสอง
2. EEDR (Data register) เป็นที่เก็บข้อมูลสำหรับที่จะเขียนลง EEPROM หรือที่อ่านได้จาก EEPROM เป็นข้อมูลขนาด 8 บิต
3. EECR (Control register) ทำหน้าที่ควบคุมการเขียนหรือ การอ่านค่าจาก EEPROM ดังนี้
   - <> Bit 3 ถึง 7 สงวนไว้
   - <> Bit 2 - EEMWE : master write enable ถ้าให้บิต นี้เป็น 1 และ EEWE เป็น 1 จะทำให้เกิดการเขียน ข้อมูลจาก EEDR ลงที่ตำแหน่ง EEAR
   - <> Bit 1 - EEWE : write enable ถ้าให้บิต นี้เป็น 1 และให้ EEMWE เป็น 1 จะทำให้เกิดการเขียน ข้อมูลจาก EEDR ลงที่ตำแหน่ง EEAR
   - <> Bit 0 - EERE : read enable ถ้าให้บิตนี้เป็น 1 จะ ทำให้เกิดการอ่านข้อมูลจาก EEPROM ที่ตำแหน่ง EEAR แล้วนำไปเก็บไว้ใน EEDR

(3) ขั้นตอนในการเขียน หรืออ่าน EEPROM

การติดต่อกับ EEPROM เพื่ออ่านหรือเขียนข้อมูลจะต้องทำตามลำดับขั้นตอนดังนี้

+----------+
| การเขียน |
+----------+

1) รอจนกระทั่งบิต EEWE ของรีจิสเตอร์ EECR เป็น 0

```
L_MainWait1:                            ; wait until EEWR becomes zero
        sbic    EECR,   1               ; - skip(PC+2) if EEWR is clear
        rjmp            L_MainWait1     ; -
```

2) กำหนดแอดเดรสของ EEPROM ลงใน EEAR

```
        ldi     r16,    0               ; EEPROM address 0 selected
        out     EEAR,   r16             ; -
```

3) นำข้อมูลที่ต้องการเขียน มาเก็บลง EEDR

```
        ldi     r16,    7               ; EEPROM data to be write
        out     EEDR,   r16             ; -
```

4) ให้บิต EEMWE เป็น 1

```
        ldi     r16,    0b00000100      ; EEPROM master write enalbe,EEMWE
        out     EECR,   r16             ; -
```

5) ให้บิต EEWE เป็น 1

```
        ldi     r16,    0b00000010      ; EEPROM write enable,EEWE bit
        out     EECR,   r16             ; -
```

6) หากมีข้อมูลที่ต้องการเขียนอีก ให้กลับไปข้อ 1

+---------+
| การอ่าน |
+---------+

1) รอจนกระทั่งบิต EEWR เป็น 0

```
L_MainWait2:                            ; wait until EEWR becomes zero
        sbic    EECR,   1               ; - skip(PC+2) if EEWR bit is clear
        rjmp            L_MainWait2     ; -
```

2) กำหนดแอดเดรสที่ต้องการอ่านลงใน EEAR

```
        ldi     r16,    0               ; EEPROM address 0
        out     EEAR,   r16             ; -
```

3) ให้บิต EERE เป็น 1

```
        ldi     r16,    0b00000001      ; EEPROM read enable
        out     EECR,   r16             ; - set EERE bit
```

4) อ่านข้อมูลจาก EEPROM ที่เก็บไว้ใน EEDR

```
in      r16,    EEDR            ; read EEPROM
```

5) หากมีข้อมูลที่ต้องการอ่านออกมาอีก ให้กลับไปข้อ 1

หมายเหตุ ในขั้นตอนการเขียนข้อมูล หลังจากให้บิต EEMWE เป็น 1 แล้วจะทำการอ่านข้อมูลออกมาโดยให้ EEWE เป็น 1 ภายในเวลา 4 cycle ถ้าไม่เช่นนั้นจะไม่สามารถเขียนข้อมูลได้ เพราะหมดเวลาเสียก่อน

(4) โปรแกรม

โปรแกรมในคราวนี้ เราจะนำฟังค์ชันเดิมคือส่วนขับ seven segment มาใช้เพื่อแสดงค่าของของมูลที่นำไปเก็บไว้ใน EEPROM ว่าทำได้สำเร็จหรือไม่ ถ้าทำได้ seven จะแสดงตัวเลขที่เรานำเข้าไปเก็บไว้ใน EEPROM ซึ่งในที่นี้คือเลข 123

```
;***************************************************************************
; AT90S2313
; EEPROM read/write access
; XTAL  = 4.0MHz
; MCU   = AT90S2313,Atmel corperation,www.atmel.com
; copyright (c) 2001 by Gumtorn Ruanfaigad.Allright reserved.
; Electronics devision.RIT Chiangmai campus.tel 053-221576
;***************************************************************************

.include        "2313def.inc"           ;
.DEVICE         AT90S2313               ;
.DSEG                                   ;
.org    100                             ;
L_data: .byte   1                       ;
                                        ;
.CSEG                                   ;
.ORG            0x0000                  ;
                                        ; ======================
                                        ; interrupt vector table
                                        ; ======================
        rjmp    F_main                  ; #0 RESET
        reti                            ; #1 INT0
        reti                            ; #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
```

```
;***************************************************************************
;Function OutSegment
;(Yin)  Y16=number to display 0..9
;(Yout) none
;(port) PORTD[x6543210] (segment)
;       -----------+
;        AT90S2313 |
;                  |            +-----+
;              PD6 11|---------a-|     |
;              PD5  9|---------b-|  -  |
;              PD4  8|---------c-| | | |
;              PD3  7|---------d-|  -  |
;              PD2  6|---------e-| | | |
;              PD1  3|---------f-|  - *|
;              PD0  2|---------g-|     |
;                    |            +-----+
;                    |               |k   common cathode
;                    |    - - - ----+
;       -----------+
;***************************************
F_OutSeg:                                ;
        push    Y0                       ;
        push    Y16                      ;
        push    ZL                       ;
        push    ZH                       ;
        ldi     ZL,     LOW(L_SegTab*2)  ; get table base address low byte
        ldi     ZH,     HIGH(L_SegTab*2) ; get table base address high byte
        lsl     Y16                      ;
        add     ZL,     Y16              ;
        lpm                              ; load program memory  Y0<-(Z)
        ldi     Y16,    0b11111111       ; all PORTD pin = output
        out     DDRD,   Y16              ; -
        mov     Y16,    Y0               ; output number to 7segment
        out     PORTD,  Y16              ; -
        ldi     Y16,    1                ; delay for brightness adjust
L_OutSeg_L1:                             ; - increase in number
        dec     Y16                      ; - will increase brightness
        brne            L_OutSeg_L1      ; -
        pop     ZH                       ;
        pop     ZL                       ;
        pop     Y16                      ;
        pop     Y0                       ;
        ret                              ;
                                         ; =====================
                                         ; 7segment lookup table
L_SegTab:                                ; =====================
                                         ; number
        .DW     0b01111110               ; 0           a
        .DW     0b00110000               ; 1          ---
        .DW     0b01101101               ; 2       f |   | b
        .DW     0b01111001               ; 3         | g |
        .DW     0b00110011               ; 4          ---
        .DW     0b01011011               ; 5       e |   | c
        .DW     0b01011111               ; 6         |   |
        .DW     0b01110000               ; 7          ---
        .DW     0b01111111               ; 8           d
        .DW     0b01111011               ; 9
;...........................................................................
```

```
;=============================================================================
;Function ScanSegment
;{yin}  yl6=numbey to display (2 digits ,yange 0 to 99)
;{yout} none
;{ydel} none
;{poyt} none
;       -----------+
;        AT90S2313 |           digit2     digitl     digit0
;                  |           +-----+    +-----+    +-----+
;             PD6 11|---------a-|     |--a-|     |--a-|     |
;             PD5  9|---------b-|  -  |--b-|  -  |--b-|  -  |
;             PD4  8|---------c-| | | |--c-| | | |--c-| | | |
;             PD3  7|---------d-|  -  |--d-|  -  |--d-|  -  |
;             PD2  6|---------e-| | | |--e-| | | |--e-| | | |
;             PD1  3|---------f-|  - *|--f-|  - *|--f-|  - *|
;             PD0  2|---------g-|     |--g-|     |--g-|     |
;                  |           +-----+    +-----+    +-----+
;                  |              |k3        |k2        |kl
;             PB4 16|--------------|----------|----------+
;             PB3 15|--------------|----------+   3 seven segment
;             PB2 14|--------------+              common cathode
;                   |
;       -----------+
;=======================================
F_ScanSegment:                          ;
        cli                             ; disable global inteyyupt
        push    yl6                     ;
        push    yl7                     ;
        push    yl8                     ;
        mov     yl8,    yl6             ; stoye numbey
        ldi     yl6,    0b00011100      ; set PB2,PB3,PB4 foy output
        out     DDRB,   yl6             ; -
        ldi     yl7,    0               ; yeset countey
L_7Segment2cal:                         ; digit 2 bin->decimal calculation
        cpi     yl8,    100             ; -
        bylt            L_7Segment2dis  ; -
        inc     yl7                     ; -
        subi    yl8,    100             ; -
        yjmp            L_7Segment2cal  ; -
L_7Segment2dis:                         ;
        ldi     yl6,    0b00000000      ; blanking all digit
        out     PORTD,  yl6             ; -
        ldi     yl6,    0b00001100      ; tuyn - on digit2
        out     PORTB,  yl6             ; -
        mov     yl6,    yl7             ; digit 2 selected and display
        ycall           F_OutSeg        ; -
        ldi     yl7,    0               ; yeset countey
L_7Segmentlcal:                         ; digit 1 bin->decimal calculation
        cpi     yl8,    10              ; -
        bylt            L_7Segmentldis  ; -
        inc     yl7                     ; -
        subi    yl8,    10              ; -
        yjmp            L_7Segmentlcal  ; -
L_7Segmentldis:                         ;
        ldi     yl6,    0b00000000      ; blanking all digit
        out     PORTD,  yl6             ; -
        ldi     yl6,    0b00010100      ; tuyn - on digitl
        out     PORTB,  yl6             ; -
        mov     yl6,    yl7             ; digit 1 selected and display
```

```
        rcall           F_OutSeg        ; -
        ldi     r16,    0b00000000      ; blanking all digit
        out     PORTD,  r16             ; -
        ldi     r16,    0b00011000      ; turn - on digit0
        out     PORTB,  r16             ; -
        mov     r16,    r18             ; digit 0 selected and display
        rcall           F_OutSeg        ; -
        pop     r18                     ;
        pop     r17                     ;
        pop     r16                     ;
        ret                             ;
;.....................................................................

;=====================================================================
;void main(void)
;=====================================
F_Main:                                 ; main function start here
        ldi     r16,    low(RAMEND)     ; set stack pointer
        out     SPL,    r16             ; -
L_MainWait1:                            ; wait until EEWR becomes zero
        sbic    EECR,   1               ; - skip(PC+2) if EEWR is clear
        rjmp            L_MainWait1     ; -
        ldi     r16,    0               ; EEPROM address 0 selected
        out     EEAR,   r16             ; -
        ldi     r16,    123             ; EEPROM data to be write
        out     EEDR,   r16             ; -
        ldi     r16,    0b00000100      ; EEPROM master write enalbe,EEMWE
        out     EECR,   r16             ; -
        ldi     r16,    0b00000010      ; EEPROM write enable,EEWE bit
        out     EECR,   r16             ; -
L_MainWait2:                            ; wait until EEWR becomes zero
        sbic    EECR,   1               ; - skip(PC+2) if EEWR bit is clear
        rjmp            L_MainWait2     ; -
        ldi     r16,    0               ; EEPROM address 0
        out     EEAR,   r16             ; -
        ldi     r16,    0b00000001      ; EEPROM read enable
        out     EECR,   r16             ; - set EERE bit
        in      r16,    EEDR            ; read EEPROM
L_MainLoop:                             ; main loop begin here
        rcall           F_ScanSegment   ; function call
        rjmp            L_MainLoop      ; infinite loop
;.....................................................................
```