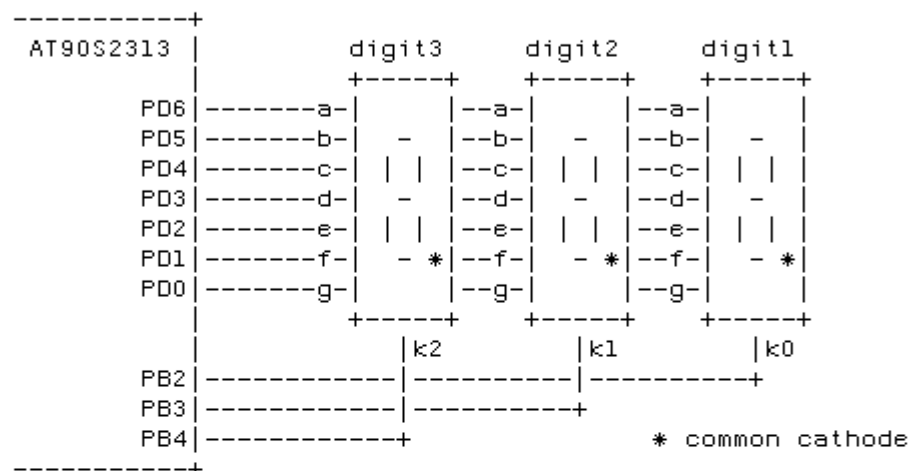


```
+-----+
| MCU | การทำ multiplex สำหรับ seven segment
| U05 | โดย อ.กำธร เรือนฝายกาฬ
+-----+
```

(1) กล่าวโดยทั่วไป

<> ในการทดลองที่ผ่านมาเราได้ฟังก์ชันเพื่อแสดงตัวเลขออก seven segment จำนวน 1 หลัก ถ้าเราต้องการแสดงผลให้มีจำนวนหลักมากกว่านี้จะทำอย่างไร เนื่องจากจำนวนพอร์ตรองไอซี AT90S2313 มีเพียง 15 ขา ดังนั้นถ้าเราต่อ seven segment เข้ากับไอซีโดยตรงจะได้เพียง 2 หลัก เพราะ 7 ขาต่อตัว \* 2 หลัก = 14ขา ถ้าต้องการแสดงผล 3 หลักจะต้องใช้ถึง 21 ขา ซึ่งมากกว่าจำนวนขาทั้งหมดของ AT90S2313 จึงเห็นว่าการริบบแบบนี้ทำไม่ได้ ทางแก้ก็คือใช้วิธีการริบบชนิดมัลติเพล็กซ์(multiplex)



ในช่วงระยะเวลาหนึ่ง seven segment จะติดเพียงหลักเดียวเท่านั้น

หลักการทำ multiplex				
digits		1	2	3
PD6	a	0	1	1
PD5	b	1	1	1
PD4	c	1	0	1
PD3	d	0	1	1
PD2	e	0	1	0
PD1	f	0	0	0
PD0	g	0	1	1
PB4	k2	0	1	1
PB3	k1	1	0	1
PB2	k0	1	1	0
time		t0	t1	t2
display		1__	_2_	--3

<> ในช่วงเวลา t0 ให้ PORTD ส่งลอจิกออกมาเป็น 0110000 ส่งไปยัง segment a ถึง g ของ seven segment ทุกตัว แต่จะยังไม่สว่าง เราจึงส่งลอจิก 00001100 ออกมาทาง PORTB เพื่อให้ seven segment ที่ต่ออยู่กับ PB0 คือ digit2 สว่าง แสดงตัวเลข 1

<> ในช่วงเวลา t1 ให้ PORTD ส่งลอจิก 1101101 และให้ PB3 เป็นลอจิก 0 ดังนั้น ในตอนนี้ digit1 จะสว่างเป็นเลข 2

<> ในช่วงเวลา t2,t3 ก็ทำงานในลักษณะคล้ายกัน ถ้าเราให้การทำงานอย่างนี้วนซ้ำกัน และให้เร็ว ก็จะมองเห็นเลข 123 เหมือนว่าติดสว่างพร้อมกัน

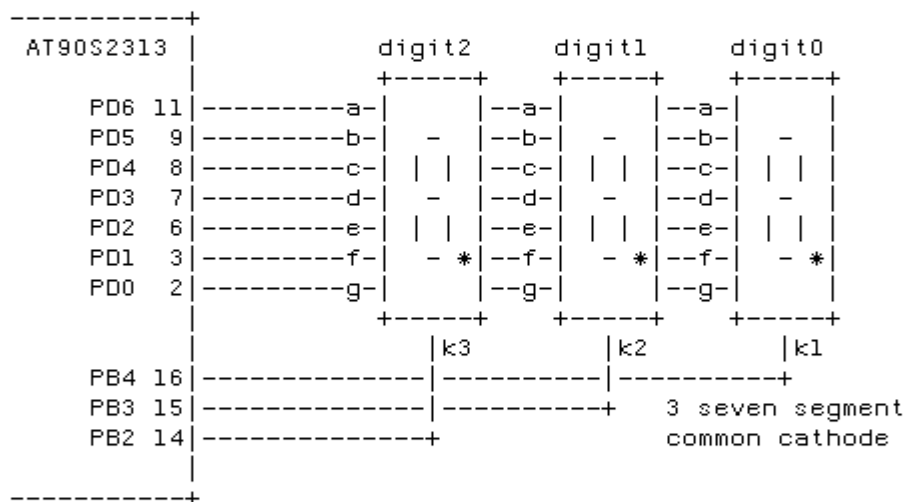
- <> ลำดับขั้นตอนในการควบคุม seven segment
1. สั่งให้ PORTD เป็น 0 ทุกบิตเพื่อตัดการแสดงผล(blanking)
  2. ส่งสัญญาณเลือก digit ออกมาจาก PORTB
  3. ส่งสัญญาณรับ segment ออกมาจาก PORTD (ใช้ฟังก์ชัน OutSeg)
  4. เลื่อนไป digit ถัดไป
  5. กลับไปทำข้อ 1

ลำดับขั้นนี้สำคัญมาก เพราะหากจัดลำดับผิดการแสดงผลจะไม่ถูกต้องเช่น ต้องการแสดงเลข 123 แต่ seven segment แสดงเลข 231 หรือไขว้ 312 ออกมา

(2) วงจร

จากหลักการที่ได้กล่าวมา เราจะออกแบบหน่วยแสดงผลขนาด 2 หลัก โดยจะเขียนโปรแกรมเพิ่มขึ้นมาอีก 1 ฟังก์ชัน เรียกว่าฟังก์ชัน ScanSegment แล้วให้ทำงานร่วมกับฟังก์ชัน OutSegment ที่เรามีอยู่แล้ว โดยฟังก์ชันใหม่นี้ จะมีคุณสมบัติดังนี้

1. รับตัวเลขเข้ามาทางรีจิสเตอร์ r16
2. แสดงผลได้ตั้งแต่ 0 ถึง 255
3. หากตัวเลขเกิน 255 จะไม่แสดงผล



<3> รายการอุปกรณ์

seven segment common cathode	3	ตัว
AT90S2313	1	ตัว

<4> โปรแกรม

```

;*****
; AT90S2313
; 1 seven segment III
; XTAL = 4.0MHz
; MCU = AT90S2313,Atmel corperation,www.atmel.com
; copyright (c) 2001 by Gumtorn Ruanfaigad.Allright reserved.
;*****

.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 INTO
reti ; #2 INT1
reti ; #3 TIMER1 CAPT11
reti ; #4 TIMER1 COMP1
reti ; #5 TIMER1 OVFL
reti ; #6 TIMERO OVFO
reti ; #7 UART,RX
reti ; #8 UART,UDRE
reti ; #9 UART,TX
reti ; #10 ANA_COMP

;=====
;Function OutSegment
;{rin} r16=number to display 0..9
;{rout} 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 r0 ;
push r16 ;
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 r16 ;
add ZL, r16 ;
lpm ; load program memory r0<-(Z)
ldi r16, 0b11111111 ; all PORTD pin = output
out DDRD, r16 ; -
mov r16, r0 ; output number to 7segment
out PORTD, r16 ; -
ldi r16, 1 ; delay for brightness adjust
L_OutSeg_L1:
; - increase in number

dec r16 ; - will increase brightness
brne L_OutSeg_L1 ; -
pop ZH ;

```

```

        pop     ZL           ;
        pop     r16         ;
        pop     r0          ;
        ret                ;
; =====
; 7segment lookup table
; =====
; 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
;{rin} r16=number to display (2 digits ,range 0 to 99)
;{rout} none
;{rdel} none
;{port} none
;
; -----+
;      AT90S2313 |          digit2      digit1      digit0
;
;      PD6 11 |-----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-  -
;
;      +-----+  +-----+  +-----+
;      PB4 16 |-----|-----|-----+
;      PB3 15 |-----|-----+ 3 seven segment
;      PB2 14 |-----+  common cathode
;
; -----+
; =====
F_ScanSegment:
        cli                ; disable global interrupt
        push     r16         ;
        push     r17         ;
        push     r18         ;
        mov     r18,    r16   ; store number
        ldi     r16,    0b00011100 ; set PB2,PB3,PB4 for output
        out     DDRB,    r16   ; -
        ldi     r17,    0     ; reset counter
L_7Segment2cal:
; digit 2 bin->decimal calculation

        cpi     r18,    100   ; -
        brlt   L_7Segment2dis ; -
        inc    r17           ; -
        subi   r18,    100   ; -

```

```

        rjmp          L_7Segment2cal ; -
L_7Segment2dis:
        ldi          r16, 0b00000000 ; blanking all digit
        out          PORTD, r16      ; -
        ldi          r16, 0b00001100 ; turn - on digit2

        out          PORTB, r16      ; -
        mov          r16, r17        ; digit 2 selected and display
        rcall        F_OutSeg        ; -
        ldi          r17, 0          ; reset counter
L_7Segment1cal:
        ; digit 1 bin->decimal calculation
        cpi          r18, 10        ; -
        brlt         L_7Segment1dis ; -
        inc          r17            ; -
        subi         r18, 10        ; -
        rjmp         L_7Segment1cal ; -
L_7Segment1dis:
        ;
        ldi          r16, 0b00000000 ; blanking all digit
        out          PORTD, r16      ; -
        ldi          r16, 0b00010100 ; turn - on digit1
        out          PORTB, r16      ; -
        mov          r16, r17        ; 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_MainLoop:
        ; main loop
        ldi          r16, 123        ; display number "123"
        rcall        F_ScanSegment   ; function call
        rjmp         L_MainLoop     ; infinite loop
;.....

```