

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ

Συστήματα Μικροϋπολογιστών  
Εργαστήριο AVR

Ομάδα: ΜΥ\_16  
Αναστάσιος Λαγός - el13531  
Κωνσταντίνος Βασιλάκης - el16504

# Ασκηση 1

Ο κώδικας είναι ο παρακάτω. Σχόλια για την λειτουργία δίνονται μέσα στον κώδικα.

```
1 .include "m16def.inc"
2
3 reset:
4     ; Initialize Stack Pointer
5     ldi r24 , low(RAMEND)
6     out SPL , r24
7     ldi r24 , high(RAMEND)
8     out SPH , r24
9     ; Set Port A as Output
10    ser r24 ; initialize PORTA for output
11    out DDRA , r24
12    ; Set Port B as Input
13    clr r24
14    out DDRB , r24
15    ; Switch on first led
16    ldi r20 , 0x01
17    out PORTA , r20
18    ; Set direction to left
19    ldi r21 , 0x00
20
21
22    ;r20 holds the current switched on led
23    ;r21 holds the direction
24    ;r21 = 0x00 -> left
25    ;r21 = 0x01 -> right
26    ;r22 holds the current input from PB0
27 main:
28    ; Check if input from button in PB0 is 0 to switch on the next
29    ; led else branch back to main
30    in r22 , PINB
31    andi r22 , 0x01
32    cpi r22 , 0x00
33    brne main
34
35    ; If r21 (direction) is 0x00 shift output to the left
36    cpi r21 , 0x00
37    brne rotate_led_right
38 rotate_led_left:
39    rol r20
40    rjmp switch_new_led_on
41    ; If r21 (direction) is 0x01 shift output to the right
42 rotate_led_right:
43    ror r20
44 switch_new_led_on:
45    ;Switch on new led
46    out PORTA , r20
47    ; Switch direction if needed
48    rcall update_direction
49    rjmp main
```

```

50
51
52 update_direction:
53     ; If the first led is switched on change direction
54     ; to left
55 if_first_led_on:
56     cpi r20 , 0x01
57     brne if_last_led_on
58     ldi r21 , 0x00;
59     rjmp endif
60     ; If the last led is switched on change direction
61     ; to right
62 if_last_led_on:
63     cpi r20 , 0x80
64     brne endif
65     ldi r21 , 0x01
66     ; else keep the same direction
67 endif:
68     ret

```

## Άσκηση 2

Ο κώδικας είναι ο παρακάτω. Για τη άσκηση αυτή χρησιμοποιήθηκε C.

```

1  #include <avr/io.h>
2
3  //Initialize variables
4  char F0,F1,A,B,C,D;
5
6  int main(void)
7  {
8      //Set Port A as Input
9      DDRA = 0X00;
10     //Set Port B as Output
11     DDRB = 0XFF;
12
13     while (1)
14     {
15         //We get the input variables by taking into
16         //account only the required bit and shifting
17         //as necessary
18         A = PINA & 0x01;      //Get A = PA0
19         B = (PINA & 0x02) >> 1;    //Get B = PA1
20         C = (PINA & 0x04) >> 2;    //Get C = PA2
21         D = (PINA & 0x08) >> 3;    //Get D = PA3
22
23         //We do the bitwise operation for F0
24         F0 = ~(A & B & (~C) | C & D);
25         //We only keep the first bit and ignore the rest

```

```

26     F0 = F0 & 0x01;
27
28     //We do the bitwise operation for F1
29     F1 = (A | B) & (C | D);
30     //We keep the first bit and we shift it to
31     //the 2nd position in the byte
32     F1 = F1 & 0x01;
33     F1 = F1 << 1;
34
35     //We take the combination of F0 and F1. F0 has the
36     //result stored in the first position of the byte and
37     //F1 in the second from the previous operations. Bytes
38     //2-7 will all be 0. We output the result in Port B
39     PORTB = F0 | F1;
40 }
41 }

```

## Ασκηση 3

Ο κώδικας δίνεται παρακάτω.

```

1  #include <avr/io.h>
2
3  char x;
4
5  int main() {
6      DDRA = 0xFF; //A as output
7      DDRC = 0x00; //C as input
8
9      x = 1;
10
11     while(1) {
12         if((PINC & 0x01) == 1) { //Push SW0
13             while((PINC & 0x01) == 1); //Release SW0
14             if(x == 128) { //If overflow
15                 x = 1; //go back to LSB
16             }
17             else {
18                 x = x << 1; //else rotate left
19             }
20         }
21         if((PINC & 0x02) == 2) { //Push SW1
22             while((PINC & 0x02) == 2); //Release SW1
23             if(x == 1) {
24                 x = 128; //go back to MSB
25             }
26             else {
27                 x = x >> 1; // else rotate right
28             }
29         }
30     }
31 }

```

```
29     }
30     if((PINC & 0x04) == 4) { //Push SW2
31         while((PINC & 0x04) == 4); //Release SW2
32         x = 128;
33     }
34     if((PINC & 0x08) == 8) { //Push SW3
35         while((PINC & 0x08) == 8); //Release SW3
36         x = 1;
37     }
38     PORTA = x;
39 }
40 return 0;
41 }
```