



### Ομάδα 65

Χρίστος Χατζηχριστοφή(03117711)

Δημήτρης Λάμπρος(03117070)

Γιώργος Αντωνίου(03117715)

**Μάθημα:** Συστήματα Μικροϋπολογιστών

**Σχολή – Εξάμηνο:** ΗΜΜΥ 6<sup>ο</sup>

Τέταρτο σετ ασκήσεων

## Άσκηση 1:

**Περιγραφή:** Στην άσκηση αυτή έχουμε το PORT B που είναι τα pins εξόδου μας (DDRB=1) και PORT C που είναι τα pins εισόδου (DDRC=0). Η διαδικασία που ακολουθήσαμε είναι:

1. Αρχικοποίηση.
2. Ολίσθηση μέχρι να φτάσουμε το άκρο.
3. Έλεγχος για input από pinc (input κουμπιού).
4. Εγγραφή στο PORTBx, άρα και στο αντίστοιχο PIN, αφού το IO B έχει οριστεί ως έξοδος.
5. Επανάληψη 2-3-4.

### Ακολουθεί ο κώδικας με σχόλια:

```
; Exercise 1, Set 4 microprocessors
; Created: 22/05/2020 16:22:05
; Author : Team65 (03117711,03117715,03117070)
; A program that makes the led move from lsb to msb, and
; then make the reverse move, from msb to lsb.
; When PC2 is pressed the cyclic move stops until released.

.include "ml6def.inc"
.def setter = r20
.def LEDB = r21
.def push_button = r25

; register to determine I/O
; define r21 as LedB
; define push_button as r25

main:
    ser setter
    out DDRB,setter
    LDI LEDB,0x01
    out PORTB,LEDB
    clr setter
    out DDRC,setter
left_loop:
    lsl LEDB
stop_button1:
    IN push_button,PINC
    cpi push_button,0x04
    breq stop_button1
    out PORTB,LEDB
    cpi LEDB,128
    breq right_loop
    rjmp left_loop
right_loop:
    lsr LEDB
stop_button2:
    IN push_button,PINC
    cpi push_button,0x04
    breq stop_button2
    out PORTB,LEDB
    cpi LEDB,1
    breq left_loop
    rjmp right_loop

; init to 0B11111111
; set port B as output
; init LEDB to 1
; set first led (msb) as on.
; clear setter to init PORTC
; Now DDRC is set with 0B00000000
; left shift, to turn on next led
; read from PINC
; compare PINC with 4, to check if PC2
; is pressed.
; If yes then loop to stop_button1
; until the button is released.
; If no print the output
; Check if LEDB is equal to 128. If yes
; then
; we need to move from MSB to LSB so
; jump to right_loop
; else continue looping until LEDB has
; value 128.
; Right shift to make the opposite move
; read from PINC
; compare PINC with 4, to check if PC2
; is pressed.
; If yes then loop to stop_button2
; until the button is released.
; If no print the output
; Check if LEDB is equal to 1. If yes
; then
; we need to move from LSB to MSB so
; jump to left_loop
; else continue looping until LEDB has
; value 1.
```

## Άσκηση 2:

**Περιγραφή:** Αρχικά λαμβάνουμε τις μεταβλητές εισόδου από την θύρα PORT B, η οποία είναι ορισμένη σαν είσοδος. Η κάθε μεταβλητή αντιστοιχεί σε ένα bit εισόδου οπότε για να μπορούν να εκτελεστούν έπειτα οι λογικές πράξεις στις συναρτήσεις μεταφέρουμε τα bit εισόδου κάθε μεταβλητής στο LSB. Τέλος εκτελούνται οι πράξεις  $F_0$  και  $F_1$  και εμφανίζονται στα δύο τελευταία bits της PORTA.

### Ακολουθεί ο κώδικας με σχόλια:

```
/* Second Exercise Set 4 microprocessors
 * Created: 23/5/2020 5:30:14
 * Author: Team65 (03117711,03117715,03117070)
 * A program that makes the following two logical functions:
 * F0 = (AB' + BC'D)'
 * F1 = (A + C)(B + D)
 * and prints the output to the 2 LSBs of PORTA
 * the inputs A,B,C,D are read from the PORTB(0-3).
 * -----
 * OUTPUT With A = 1,B = 0,C = 1,D = 1
 * F0 = 0 (shown in BIT0 of PortA)
 * F1 = 1 (shown in BIT1 of PortA)
 * -----
 * -----
 * OUTPUT With A = 0,B = 0,C = 0,D = 0
 * F0 = 1 (shown in BIT0 of PortA)
 * F1 = 0 (shown in BIT1 of PortA)
 * -----
 * -----
 * OUTPUT With A = 0,B = 1,C = 1,D = 0
 * F0 = 1 (shown in BIT0 of PortA)
 * F1 = 1 (shown in BIT1 of PortA)
 * -----
 */

#include <avr/io.h>
char A,B,C,D,F0,F1;

int main(void) {
    DDRA = 0xFF; //A as output
    DDRB = 0x00; //B as input

    while (1) {
        A = PINB & 0x01; //A is the first bit of PINB
        B = PINB & 0x02; //B is the second bit of PINB
        B = B >> 1; //Right shift B, so its on bit0 as B
        C = PINB & 0x04; //C is the third bit of PINB
        C = C >> 2; //Right shift C, so its on bit0 as C
        D = PINB & 0x08; //D is the fourth bit of PINB
        D = D >> 3; //Right shift D, so its on bit0 as D
        F0 = !((A & (!B)) | (B & (!C) & D)); //Definition of F0 and F1, using not
                                                //operator(!),
                                                //bitwise or(|), and bitwise and(&)
        F1 = ((A | C) & (B | D)); //bitwise or(|), and bitwise and(&)
        F1 = F1 << 1; //Left shift F1 so as the answer is on
                        //bit1
        PORTA = F0 + F1; //Add both results to PORTA(F0+F1), to
                        //print the output
    }
}
```

### Άσκηση 3:

**Περιγραφή:** Το κουμπί που πατιέται κάθε φορά θα "γράφεται" στα "Physical"(Στο Atmel Studio Virtual) Pins του IO port A. Έτσι ορίζουμε τα pins του PORT A ως inputs και περιμένουμε να "εγκλωβίσουμε" την εγγραφή που γίνεται σε αυτά, ώστε να περιμένουμε έπειτα την επαναφορά του αντίστοιχου push button και άρα την επαναφορά του λογικού 0 στα pins. Αυτό στο atmel studio θα γίνει "χειροκίνητα". Η θύρα B θα οριστεί ως θύρα εξόδου. Έτσι, όπως γράφουμε τις τιμές του LED στα PORT B, θα δούμε το ίδιο "αποτέλεσμα" και στα PINS.

#### Ακολουθεί ο κώδικας με σχόλια:

```
/* Third Exercise Set 4 microprocessors
 * Created: 23/5/2020 6:25:11
 * Author: Team65 (03117711,03117715,03117070)
 * A program that makes some operations when SW0-3 are pressed.
 * SW0 - SW3 are read from PINA
 * SW0: right cyclic move
 * SW1: left cyclic move
 * SW2: move the ON led to LSB
 * SW3: move the ON led to MSB
 */
#include <avr/io.h>

int main(void)
{
    // Init PORTB as output
    DDRB = 0xFF;
    // Init PORTA as input
    DDRA = 0x00;
    //Init PORTB to 1, so as the led0 of portb is turned on.
    PORTB = 0x01;
    while (1) {
        //If SW0 is on then make cyclic move(right)
        if ((PINA & 0x01) == 1) {
            //while SW0 is pressed do nothing.
            while ((PINA & 0x01) == 1);
            //if PORTB is equal to 1 then turn on the Led7 else do right shift
            (PORTB == 1) ? (PORTB = 128) : (PORTB = PORTB >> 1);
        }
        //If SW1 is on then make cyclic move(left)
        if ((PINA & 0x02) == 2) {
            //while SW1 is pressed do nothing.
            while ((PINA & 0x02) == 2);
            //if PORTB is equal to 128 then turn on the Led0 else do left shift
            (PORTB == 128) ? (PORTB = 1) : (PORTB = PORTB << 1);
        }
        //If SW2 is on then turn on LSB Led (led0)
        if ((PINA & 0x04) == 4) {
            //while SW2 is pressed do nothing.
            while ((PINA & 0x04) == 4);
            //make PORTB equal to 1, so as led0 is turned on
            PORTB = 0x01;
        }
        //If SW3 is on then turn on MSB Led (led7)
        if ((PINA & 0x08) == 8) {
            //while SW3 is pressed do nothing.
            while ((PINA & 0x08) == 8);
            //make PORTB equal to 128(0x80), so as led7 is turned on
            PORTB = 0x80;
        }
    }
}
```