Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ & ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

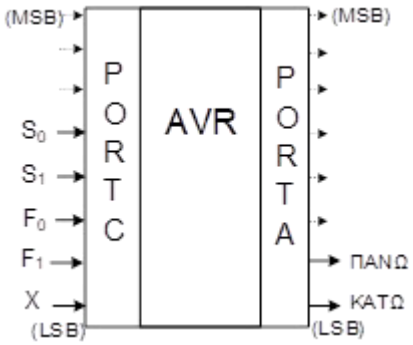
ΑΘΗΝΑ 25. 6. 2021

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "***Συστήματα Μικροϋπολογιστών***"

(ΘΕΜΑ 2ο – ΣΥΝΟΛΟ 4.5 Μονάδες)

**Έναρξη 12:30 - ΔΙΑΡΚΕΙΑ 60' + 10' Παράδοση: 13:40'**

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ:** . . . . [redacted]




**ΘΕΜΑ 2ο:** (4.5 ΜΟΝΑΔΕΣ): Σε ένα μικροελεγκτή AVR Mega16 που αξιοποιεί μία θύρα εισόδου και μία εξόδου, όπως φαίνεται στο διπλανό σχήμα, να υλοποιηθεί ένα σύστημα οδήγησης ενός ανελκυστήρα δυο θέσεων (ισογείου και 1ου ορόφου). Η κίνηση προς το ισόγειο ή τον 1ο όροφο ελέγχεται από τους εξωτερικούς διακόπτες (Push-Buttons) *F*0 και *F*1 αντίστοιχα καθώς και από έναν εσωτερικό διακόπτη (Push-Button) *X*. Για να δοθεί εντολή από τους διακόπτες αυτούς, προϋπόθεση είναι το βαγόνι να είναι σταματημένο στο ισόγειο ή στον 1ο όροφο. Όταν κινείται πρέπει να σταματάει από το πρόγραμμα με βάση τους αισθητήρες *S*0 και *S*1 που είναι τερματικοί διακόπτες και οι οποίοι δίνουν λογικό 1 αυτόματα όταν ο θάλαμος φτάνει στο ισόγειο ή στον 1ο όροφο αντίστοιχα. Υποθέτουμε ότι κατά την εκκίνηση του συστήματος, ο θάλαμος πρέπει να βρίσκεται στο ισόγειο, αλλιώς πριν δεχτεί οποιαδήποτε εντολή να μεταφέρεται σε αυτή τη θέση αυτόματα.

Αναλυτικά, αν ο θάλαμος φτάσει στο ισόγειο, τότε πρέπει να σταματάει η κίνησή του και να ελέγχονται οι διακόπτες *F*1 και *X*. Αν ένας από αυτούς είναι ενεργοποιημένος (=1) τότε έχουμε κίνηση προς τα πάνω. Αντίστοιχα αν ο θάλαμος φτάσει στον 1ο όροφο, τότε πρέπει να σταματάει η κίνησή του και να ελέγχονται οι διακόπτες *F*0 και *X*. Αν ένας από αυτούς είναι ενεργοποιημένος (=1) τότε έχουμε κίνηση προς τα κάτω. Δώστε το αντίστοιχο πρόγραμμα σε assembly και σε C.

(Assembly: 2.5 ΜΟΝΑΔΕΣ και C: 2 ΜΟΝΑΔΕΣ)

**ASSEMBLY:**

```
.DEF S0 = r18
.DEF B = r19
.DEF X = r20
.DEF F0 = r21
.DEF F1 = r22
.DEF S1 = r23
.DEF L = r17
.DEF setter = r16

stack: ldi r24 , low(RAMEND)    ; initialize stack pointer
                out SPL , r24
                ldi r24 , high(RAMEND)
                out SPH , r24

IO_set:
        ser r24
        out DDRA,r24
        clr r24
        OUT DDRC

Main:
    START_INIT:
        IN B,PORTC
        LDI L,0x01
        MOV S0,B
        ANDI S0,0x10; this is  S0
```

```
        LSR S0
        LSR S0
        LSR S0
        LSR S0
        CMP S0,L
        BRNE go_down_setup   l IF S0 = 0 (lvl 1 or MID STUCK go 0)
    Start:
        IN B,PORTC
        MOV X,B
        ANDI X,0x01 ; this is X
        MOV F1,B
        ANDI F1,0x02; this is  F1
        LSR F1
        MOV F0,B
        ANDI F0,0x04; this is  F0
        LSR F0
        LSR F0
        MOV S1,B
        ANDI S1,0x08; this is  S1
        LSR S1
        LSR S1
        LSR S1
        MOV S0,B
        ANDI S0,0x10; this is  S0
        LSR S0
        LSR S0
        LSR S0
        LSR S0

        CP S0, L
        BREQ   CONTINUE ;  if S0 != 1 then go to isogeio.
        CP S1, L
        BREQ  CONTINUE
        JMP START_INIT  ; IF S0 & S1 = 0   then its in transit aka Jump start.
    CONTINUE:
        CMP F0, L
        BREQ go_down
        CMP F1,L
        BREQ go_up
        CMP X,L
        BREQ go_up_down
        rjmp Start
go_down _setup:
        LDI setter,0x01
        out PORTA, setter
        CMP S0, L
        BRNE go_down_setup
        LDI setter,0x00
        out PORTA, setter
        rjmp START_INIT
 go_down:
        LDI setter,0x01;
        out PORTA,setter
        CMP S0, L
        BRNE go_down
        LDI setter,0x00
        out PORTA,setter
        rjmp Start
go_up:
        LDI setter,0x02;
        out PORTA,setter
        CMP S0, L
```

```
        BRNE go_up
        LDI setter,0x00
        out PORTA,setter
        rjmp Start


 go_up_down:
        CMP S0,L
        BREQ go_up_x
        CMP S1,L
        BREQ go_down_x
        rjmp START_INIT
go_up_x:
        LDI setter,0x02;
        out PORTA,setter
        CMP S0, L
        BRNE go_up_x
        LDI setter,0x00
        out PORTA,setter
        rjmp Start
go_down_x:
        LDI setter,0x01
        out PORTA, setter
        CMP S0, L
        BRNE go_down_x
        LDI setter,0x00
        out PORTA, setter
        rjmp START_INIT

C:
#include <avr/io.h>
#include <mega16.h>
unsigned char B,S0,S1,F0,F1,X,setter;

void restarter (){
        IF (S0 == 0){
          WHILE (S0 != 1){
                setter = 0x01;
                PORTC = setter;
          }
         setter = 0x00;
        PORTC = setter;
        }
}
void go_up (){
        IF (S1 == 0){
          WHILE (S1 != 1){
                setter = 0x02;
                PORTC = setter;
          }
         setter = 0x00;
        PORTC = setter;
        }
}

void go_down (){
        IF (S0 == 0){
          WHILE (S0 != 1){
                setter = 0x01;
                PORTC = setter;
          }
         setter = 0x00;
```

```
        PORTC = setter;
        }
}


int main (void){
        DDRC = 0x00 //input
        DDRA = 0xFF //output

         B=PINC
        S0 = B & 0x10;
        S1 = B & 0x08;
        F0 = B & 0x04;
        F1 = B & 0x02;
        X  =  B & 0x01;
        S0 = S0>>4;
        S1 = S1 >>3;
        F0 = F0>>2;
        F1 = F1>>1;
        Restarter();

        WHILE (1){
        B=PINC
        S0 = B & 0x10;
        S1 = B & 0x08;
        F0 = B & 0x04;
        F1 = B & 0x02;
        X  =  B & 0x01;
        S0 = S0>>4;
        S1 = S1 >>3;
        F0 = F0>>2;
        F1 = F1>>1;
                IF(S0 == 1 && S1 == 0){
                        If (F1 == 1 || X == 1){
                                Go_up();
                        }
                }
                IF (S0 == 0&&S1 ==1 ){
                        If(F0 == 1 || X == 1){
                                Go_down();
                        }
                }
                else restarter();
        }
        return 0;
}
```

**θεωρούμε ότι για την έξοδο 01 σημαίνει κάτω 10 σημαίνει πάνω και 00 σημαίνει καμία λειτουργία/σταματάει.**
**Θεωρούμε ότι το X αναφέρεται στην μεταφορά του άλλου ορόφου( εάν είναι στο ισόγειο πάει 1$^{ο}$ και αντίστοιχα).**