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

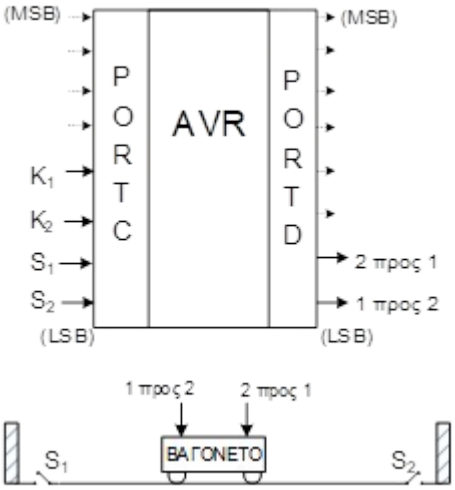
ΑΘΗΝΑ 22. 9. 2020

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

(ΘΕΜΑ 2ο – ΣΥΝΟΛΟ 4.5 Μονάδες)

Έναρξη 12:50 - ΔΙΑΡΚΕΙΑ 40' + 10' Παράδοση: 13:40'

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: . . . . . .



ΘΕΜΑ 2ο: (4.5 ΜΟΝΑΔΕΣ): Σε ένα μικροελεγκτή AVR Mega16 που διαθέτει από μία θύρα εισόδου και εξόδου, όπως φαίνεται στο διπλανό σχήμα, να υλοποιηθεί ένα σύστημα οδήγησης ενός βαγονέτου δυο θέσεων (1 και 2). Η κίνηση προς τη θέση 1 ή 2 ενεργοποιείται από τους διακόπτες (Push-Buttons) K_1 και K_2 αντίστοιχα, με την προϋπόθεση ότι το βαγονέτο είναι σταματημένο στην αντίθετη θέση. Όταν το βαγονέτο κινείται και φτάνει στη θέση 1 ή 2, οι αισθητήρες S_1 και S_2 που είναι τερματικοί διακόπτες δίνουν λογικό 1, αντίστοιχα. Η κίνηση του βαγονέτου ελέγχεται από τα αντίστοιχα σήματα της θύρας εξόδου. Υποθέτουμε ότι κατά την εκκίνηση του συστήματος το βαγονέτο πρέπει να βρίσκεται στη θέση 1 αλλιώς πριν δεχτεί οποιαδήποτε εντολή να μεταφέρεται σε αυτή τη θέση αυτόματα. Δώστε το αντίστοιχο πρόγραμμα σε assembly και σε C.
(Assembly: 2.5 ΜΟΝΑΔΕΣ και C: 2 ΜΟΝΑΔΕΣ)

ΑΠΑΝΤΗΣΗ

κώδικας c

```
#include <avr/io.h>

unsigned char s1, s2, k1, k2;
unsigned char out;
unsigned char x;


int main(void){

DDRD = 0xFF;                    //portD εξοδος
DDRC = 0x00;                    //portC εισοδος


s2 = PINC & 0x01;     //παρε μια αρχικη μετηση
s1 = PINC & 0x02;
k2= PINC & 0x04;
k1= PINC & 0x08;

while(1){    // βαλτο στο αισθητηρα S1
if(s1 == 0){ x = 0x01 ; }

else break;
}

while(1){

s2 = PINC & 0x01;
s1 = PINC & 0x02;
k2= PINC & 0x04;
k1= PINC & 0x08;
```

```
if( s1==1 && k2==1){ // αμα θελει να παει απο το 1 στο 2 και ειναι πανω στο 1 κανε τοχ=1
	x = 0x01;
}

if( s2==1 && k1 ==1){ // αμα θελει να παει απο το 2 στο 1 και ειναι πανω στο 2 κανε τοχ=2
	x = 0x02;
}

out = x; // αλλιως παρε την προηγουμενη τιμη του χ και πηγαινε μεχρι να φτασεις
portD = out;
}

return 0;}
```

Κώδικας assembly

```
.INCLUDE "m16def.inc"
.def EIS=r22
.def temp=r23
.def s1=r24
.def s2=r25
.def k1=r26
.def k2=r27
.def EJO=r28
.def result=r29

main:

clr temp
out DDRC ,temp
ser temp
out PORTD, temp
out DDRD,temp

in EIS, PINC
mov s2, EIS
and s2, 0x01
mov s1, EIS
and s1, 0x02
mov k2, EIS
and k2, 0x04
mov k1, EIS
and k1, 0x08

reset: // βαζουμε το βαγονι στην θεση 1  κανοντας το να κινηθει προς την θεση αυτη μεχρι να φτασει

cpi s1, 0x02
breq run
mov result, 0x02
out PORTD, result
jmp reset


run:
cpi s1, 0x02 // αμα ειναι στο σ1 παμε στην πρωτη περιπτωση
breq case1
cpi  s2, 0x01//αμα ειναι στο σ2 παμε στην δευτερη
breq case2
```

```
jmp run // αλλιως τρεχουμε παλι το προγραμμα



case1:
cpi k2, 0x04// αμα το k2=1 δηλαδη θελουμε να παμε στο 2 τοτε βαζουμε σαν εξοδο το 1 kai
brne run   // παμε παλι στην αρχη αν δεν θελουμε να παρουμε το 2
mov EJO, 0x01
out PORTD, EJO
jmp run

case2:
cpi k1, 0x08 //αντισοιχο με πανω
brne run
mov EJO, 0x02
out PORTD, EJO
jmp run
```