

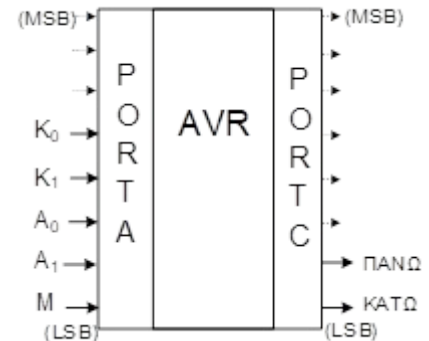
ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 2^ο – ΣΥΝΟΛΟ 4.5 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΓΕΩΡΓΙΟΣ ΚΥΡΙΑΚΟΠΟΥΛΟΣ - el18153

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 1^{ου} ορόφου). Η κίνηση προς το ισόγειο ή τον 1^ο όροφο ελέγχεται από τους εξωτερικούς διακόπτες (Push-Buttons) K0 και K1 αντίστοιχα καθώς και από έναν εσωτερικό διακόπτη (Push-Button) M. Για να δοθεί εντολή από τους διακόπτες αυτούς, προϋπόθεση είναι το βαγόνι να είναι σταματημένο στο ισόγειο ή στον 1^ο όροφο. Όταν κινείται πρέπει να σταματάει από το πρόγραμμα με βάση τους αισθητήρες A0 και A1 που είναι τερματικοί διακόπτες και οι οποίοι δίνουν λογικό 1 αυτόματα όταν ο θάλαμος φτάνει στο ισόγειο ή στον 1^ο 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 K1 και M. Αν ένας από αυτούς είναι ενεργοποιημένος (=1) τότε έχουμε κίνηση προς τα πάνω. Αντίστοιχα αν ο θάλαμος φτάσει στον 1^ο όροφο, τότε πρέπει να σταματάει η κίνησή του και να ελέγχονται οι διακόπτες K0 και M. 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)

C:

```
int main(void){
    DDRC=0xFF;    // output on PORTC
    DDRA=0x00;    // input on PORTA

    while((PINA & 0x04) == 0){    // if not on floor go there (output down = 1)
        PORTC = 1;
    }
    while(1) {
        if ((PINA & 0x01) == 1){    // if M is pressed
            if ((PINA & 0x10) == 16){    // must move to floor
                while ((PINA & 0x04) != 4){
                    PORTC = 1;    // if not on floor move till you are there
                }
            }
            if ((PINA & 0x08) == 8){    // must move to 1st floor
                while ((PINA & 0x02) != 2){
                    PORTC = 2;    // if not on 1st floor move till you are there
                }
            }
        }
    }
}
```

Assembly:

include "m16def.inc"

reset:

```
ldi r24, low(RAMEND)
out SPL, r24
ldi r24, high(RAMEND)
out SPH, r24
ser r24
out DDRC, r24
clr r24
out DDRA, r24
ldi r24, 0b11111111
out PORTA, r24
```

```
in r24, PINA
and r24, 0b00000100 ; check a0
cpi r24, 0 ; if not on ground
jnz mov_ground ; go to ground
```

main:

```
ldi r22, 0b00000000
out PORTC, r22 ; stop moving
in r24, PINA
mov r23, r24
and r24, 0b00000001 ; if moving
cpi r24, 0
jnz moving ; check where to
jmp main ; else loop
```

moving:

```
mov r24, r23
and r24, 0b00001000 ; move to 1st floor
cpi r24, 0
jnz mov_floor
mov r24, r23
and r24, 0b00010000 ; move to ground
cpi r24, 0
jnz mov_ground
jmp main
```

mov_ground:

```
in r24, PINA
and r24, 0b00000100 ; if on ground loop main
jnz main
ldi r22, 0b00000001
out PORTC, r22 ; else move to ground and check again
jmp mov_ground
```

mov_floor:

```
in r24, PINA
and r24, 0b00000010 ; if on floor loop main
jnz main
ldi r22, 0b00000010 ; else move to floor and check again
out PORTC, r22
jmp mov_floor
```