

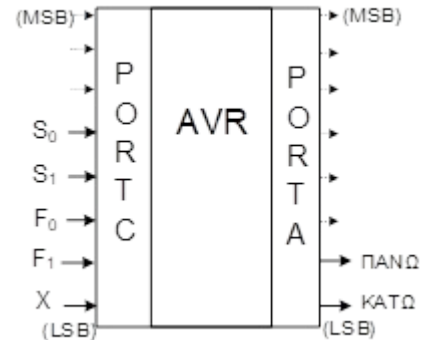
ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 2<sup>ο</sup> – ΣΥΝΟΛΟ 4.5 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΤΕΦΑΝΑΚΗΣ ΓΕΩΡΓΙΟΣ el18436

**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<sup>ο</sup> ορόφου). Η κίνηση προς το ισόγειο ή τον 1<sup>ο</sup> 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 1<sup>ο</sup> 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 1<sup>ο</sup> 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 1<sup>ο</sup> 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)

Υλοποίηση σε AVR

```
.INCLUDE "m16def.inc"
```

```
.def input=r18
```

```
main:
```

```
    clr r28
    out DDRC, r28        ; input on PORTC
    ser r28
    out DDRA, r28        ; output on PORTA
```

```
init:
```

```
    in    input, PINC
    andi  input, 0x10     ; check if S0 is ON on startup
    cpi   input, 0x10
    breq  zeroth_floor
    ldi   r29, 0x01       ; if not on zeroth_floor on startup
    out   PORTA, r29      ; show on output going down
    jmp   down            ; jump to down routine
```

```
zeroth_floor:
```

```
    ldi   r29, 0x00
    out   PORTA, r29
    in    input, PINC
    andi  input, 0x03     ; X or F1
    cpi   input, 0x00     ; if X or F1 != 0 → move
    breq  zeroth_floor
```

```

ldi    r29, 0x02
out    PORTA, r29      ; show on output “elevator coming up”
jmp    up              ; jump to up routine

first_floor:
in     input, PINC
andi   input, 0x05      ; X and F0
cpi    input, 0x00      ; simillar logic, if one of them is 1 → move
breq   first_floor
ldi    r29, 0x01
out    PORTA, r29      ; show on output going down
jmp    down

up:
in     input, PINC
andi   input, 0x08      ; S1
cpi    input, 0x08
brne   up              ; while S1 != 1 → still going up
ldi    r29, 0x00
out    PORTA, r29      ; stop showing on output “elev going up”
jmp    first_floor      ; jump to first floor routine

down:
in     input, PINB
andi   input, 0x10      ; S0, simillar logic with routine “up”
cpi    input, 0x10
brne   down            ;
ldi    r29, 0x02
out    PORTA, r29      ; stop showing on output “elev going down”
jmp    zeroth_floor     ; jump to zeroth floor routine

```

Υλοποίηση σε C

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

```
char X, F0, F1, S1, S0;
```

```
int main() {
```

```
    DDRC = 0x00;
```

```
    DDRA = 0xff;
```

```
    X = PINC & 0x01;
```

```
    F1 = PINC & 0x02;
```

```
    F0 = PINC & 0x04;
```

```
    S1 = PINC & 0x08;
```

```
    S0 = PINC & 0x10;
```

```
    F1 >> 1;
```

```
    F0 >> 2;
```

```
    S1 >> 3;
```

```
    S0 >> 4;
```

```
    // while not on zeroth floor go down
```

```
    while (S0 != 1)
```

```
        PORTA = 0x01;
```

```
    // stop going down
```

```
    PORTA = 0x00;
```

```
    zeroth = 1;
```

```
    while (1) {
```

```
X = PINC & 0x01;
F1 = PINC & 0x02;
F0 = PINC & 0x04;
S1 = PINC & 0x08;
S0 = PINC & 0x10;

F1 >> 1;
F0 >> 2;
S1 >> 3;
S0 >> 4;

if (F1 | X) {
    while (F1 | X);
    while (S1 != 1) {
        PORTA = 0x02;
    }
    PORTA = 0x00;
}

if (F0 | X) {
    while (F0 | X);
    while (S0 != 1) {
        PORTA = 0x01;
    }
    PORTA = 0x00;
}

}
}
```