



**Εθνικό Μετσόβιο Πολυτεχνείο**

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

## **Εργαστήριο Μικροϋπολογιστών**

7<sup>ο</sup> εξάμηνο, Ακαδημαϊκή περίοδος 2025-2026

### **1<sup>η</sup> Εργαστηριακή Άσκηση**

**Ανάπτυξη κώδικα για τον μικροελεγκτή ATmega328PB και προσομοίωση της εκτέλεσης του στο αναπτυξιακό περιβάλλον MPLAB X**

### **Εργαστηριακή Αναφορά**

<b>Φοιτητές:</b>	Παπαδάτος Αναστάσιος	->	<b>ΑΜ:</b> 03122847
	Σέρτζιο Γκούρι	->	<b>ΑΜ:</b> 03122827

**Ομάδα:** 30

**Ημερομηνία:** 19/10/2025

## Άσκηση 1:

Δημιουργία χρονομέτρου x\_msec σε κώδικα assembly

Παρακάτω φαίνεται ο κώδικας assembly για το χρονόμετρο που φτιάξαμε:

```
; 1.1.asm
; Created: 15/10/2025 7:42:11 μμ
; Author : anast

.include "m328PBdef
.inc " .equ D=3000      ;ms
ldi r24,LOW(RAMEND)
out SPL,r24
ldi r25,HIGH(RAMEND)
out SPH,r24
ser r24
out DDRD,r24
clr r29
out PORTD,r29
jmp main

main:
ldi r24,LOW(D)
ldi r25,HIGH(D)
rcall wait_x_msec      ;3 cycles
com r29
out portd,r29
rjmp main

wait_x_msec:
ldi r26,LOW(15988)     ;1 cycle
ldi r27,HIGH(15988)    ;1 cycle
delay:
sbiw r26,4             ;2 cycles
brne delay             ;2 cycles or 1 if last
                        ;delay total 15987 cycles
                        ;at this stage 15989 cycles

sbiw r24,1             ;2 cycles
breq if_last           ;1 cycle or 2 if branch

;at this stage 15992 cycles
nop
nop
nop
nop
nop
nop
nop                    ;6 cycles

;at this stage 15998 cycles
rjmp wait_x_msec       ;2 cycles and total 16000 cycles (1 msec)
```

```

if_last:
;at this stage 15993 cycles
ret      ;4 cycles
         ;rcall is 3 cycles so total at this stage 16000 cycles (1 msec)
         ;so after the completion of the function we have x msec delay

```

### Σχόλια/Περιγραφή του κώδικα:

Αφού το ρολόι του ATmegaPB είναι 16MHz θέλουμε να πετύχουμε ακριβώς 16000 κύκλους για να έχουμε delay 1ms. Αυτό που σκεφτήκαμε είναι πως η συνάρτησή μας (wait\_x\_msec), θα τρέχει ένα βοηθητικό loop μέσα, τόσες φορές όσες και η τιμή του ζεύγους καταχωρητών r24,r25 και αυτό το βοηθητικό loop κάθε φορά που τρέχει θα καταναλώνει 16000 κύκλους, δηλαδή 1ms. Όμως για να πετύχουμε ακριβώς τον χρόνο των r24,r25 σε ms, θα πρέπει να συνυπολογίσουμε τους κύκλους που καταναλώνει η εντολή rcall και η εντολή ret. Έτσι προσέχουμε η τελευταία επανάληψη (τελευταίο ms), να μην τρέχει αυτά τα nop που έτρεχαν οι υπόλοιπες επαναλήψεις και έτσι καταφέρνουμε το επιθυμητό αποτέλεσμα.

## Άσκηση 2:

Λογικές συναρτήσεις σε κώδικα assembly

Παρακάτω φαίνεται ο κώδικας assembly που φτιάξαμε:

```

; 1.2.asm
;
; Created: 16/10/2025 12:36:15 μμ
; Author : anast
;

.include "m328pbdef.inc"
.def A=r16
.def B=r17
.def C=r18
.def D=r19
.def temp=r20
.def counter=r21
.def F0=r22
.def F1=r23

ldi r24,LOW(RAMEND)
out SPL,r24
ldi r24,HIGH(RAMEND)
out SPH,r24
clr r24

```

```

start:
ldi A,0x51
ldi B,0x40
ldi C,0x1F
ldi D,0xFE
ldi counter,0x06
ser r24
out DDRD,r24;

loop:
subi A,-1
subi B,-2
subi C,-3
subi D,-4
mov temp,A
com temp           ;A'
and temp,B         ;A'B
mov F0,temp        ;F0=A'B
mov temp,B
com temp           ;B'
and temp,D         ;B'D
or F0,temp         ; F0=A'B+B'D
com F0             ;F0=(A'B+B'D)'

out PORTD,F0

mov temp,A
or temp,C          ;A+C
mov F1,temp        ;F1=A+C
mov temp,B
or temp,D          ;B+D
and F1,temp        ;F1=(A+C)(B+D)

out PORTD,F1

dec counter
brne loop

rjmp start

```

#### Σχόλια/Περιγραφή του κώδικα:

Χρησιμοποιούμε βοηθητικούς καταχωρητές A,B,C,D όπου φορτώνουμε τις αρχικές τους τιμές έχοντας αφαιρέσει 1 στον A, 2 στον B, 3 στον C και 4 στον D. Έτσι όταν μπαίνει στο loop τους αυξάνει (και έπειτα τους αυξάνει ανάλογα σε κάθε επανάληψη) και άρα καταφέρνουμε να έχουμε τη σωστή τιμή τους στην 1<sup>η</sup> επανάληψη. Υστερα κάνοντας τις σωστές πράξεις μεταξύ των καταχωρητών παίρνουμε τα σωστά αποτελέσματα F0 και F1. Τα εμφανίζουμε στο PORTD της πλακέτας, ώστε να διαβάσουμε τα αποτελέσματα και να συμπληρώσουμε τον πίνακα που μας ζητείται. Είχαμε προσθέσει Breakpoint στην εκτέλεσή

μας μετά από κάθε εμφάνιση ώστε να μπορούμε να διαβάζουμε το αποτέλεσμα και έπειτα να προχωράμε στο επόμενο.

A	B	C	D	F0	F1
0x52	0x42	0x22	0x02	0xFF	0x42
0x53	0x44	0x25	0x06	0xF9	0x46
0x54	0x46	0x28	0x0A	0xF5	0x4C
0x55	0x48	0x2B	0x0E	0xF1	0x4E
0x56	0x4A	0x2E	0x13	0xE7	0x5A
0x57	0x4C	0x32	0x17	0xE5	0x56

### Άσκηση 3:

Βαγόνι που κινείται δεξιά-αριστερά με στάσεις στα άκρα

Παρακάτω φαίνεται ο κώδικας assembly για το βαγόνι που χρησιμοποιεί και τον κώδικα της πρώτης άσκησης:

```
; 1.3.asm
; Created: 16/10/2025 1:32:49 μμ
; Author : anast

.include "m328pbdef.inc"
.def wagon = r16
.def counter = r17

ldi r24,LOW(RAMEND)
out SPL,r24
ldi r24,HIGH(RAMEND)
out SPH,r24
ser r24
out DDRD,r24
clr r24
ldi wagon, 0x01
ldi r18,0x01
out PORTD,wagon

start:
ldi counter, 0x07
BST r18,1           ;to MSB: T flag = 0
```

```

to_MSB:
ldi r24,LOW(2000)
ldi r25,HIGH(2000)
rcall wait_x_msec      ;2 sec delay
lsl wagon              ;move wagon to MSB
out PORTD,wagon
dec counter            ;counter=counter-1
brne to_MSB            ;if not edge continue moving to MSB
ldi r24,LOW(1000)
ldi r25,HIGH(1000)
rcall wait_x_msec      ;another 1 sec delay because wagon is at the edge
BST r18,0              ;to LSB: T flag = 1
ldi counter, 0x07      ;reset counter
to_LSB:
ldi r24,LOW(2000)
ldi r25,HIGH(2000)
rcall wait_x_msec      ;2 sec delay
lsr wagon              ;move wagon to LSB
out PORTD,wagon
dec counter            ;counter=counter-1
brne to_LSB            ;if not edge continue moving to LSB
ldi r24,LOW(1000)
ldi r25,HIGH(1000)     ;1 sec delay
rcall wait_x_msec      ;another 1 sec delay because wagon is at the edge
rjmp start

wait_x_msec:
ldi r26,LOW(15988)     ;1 cycle
ldi r27,HIGH(15988)    ;1 cycle
delay:
sbiw r26,4             ;2 cycles
brne delay             ;2 cycles or 1 if last
                    ;delay total 15987 cycles
;at this stage 15989 cycles

sbiw r24,1             ;2 cycles
breq if_last           ;1 cycle or 2 if branch

;at this stage 15992 cycles
nop
nop
nop
nop
nop
nop                    ;6 cycles
;at this stage 15998 cycles
rjmp wait_x_msec       ;2 cycles and total 16000 cycles (1 msec)

if_last:
;at this stage 15993 cycles
ret                    ;4 cycles
                    ;rcall is 3 cycles so total at this stage 16000 cycles (1 msec)
                    ;so after the completion of the function we have x msec delay

```

Σχόλια/Περιγραφή του κώδικα:

Έχουμε έναν καταχωρητή wagon όπου τον αρχικοποιούμε στην τιμή 0b00000001. Το 1 αντιπροσωπεύει το βαγονέτο (αναμμένο LED) που ξεκινάει από το LSB και με κατάλληλα rotations (έχουμε καθυστέρηση 2 sec μεταξύ κάθε rotation) και ύστερα έξοδο στο PORTD του καταχωρητή wagon, πετυχαίνουμε την κίνηση του βαγονέτου. Με την εντολή BST καταφέρνουμε να αλλάζουμε την τιμή του T flag σε 0 αν το βαγονέτο κινείται προς το MSB και αντίστροφα σε 1 αν κινείται προς το LSB. Έχουμε έναν καταχωρητή counter που μετράει τις 7 επαναλήψεις των rotations του wagon και μέσω αυτού διαπιστώνουμε αν πρέπει να αλλάξουμε κατεύθυνση (πρόσθετη καθυστέρηση 1 sec όταν αλλάζουμε κατεύθυνση). Τις καθυστερήσεις τις πετυχαίνουμε με την κλήση της συνάρτησης wait\_x\_msec (η συνάρτηση της 1<sup>ης</sup> άσκησης) έχοντας φροντίσει να φορτώσουμε το ζεύγος καταχωρητών r24,r25 με τον κατάλληλο αριθμό για την επιθυμητή καθυστέρηση (πχ 1000 για delay 1sec).