



Technical  
University  
of Crete

# Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

ΑΝΑΦΟΡΑ ΠΡΩΤΗΣ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

Καλογεράκης Στέφανος | Ζερβάκης Αρης

## Σκοπός εργαστηριακής άσκησης

Σκοπός της πρώτης εργαστηριακής άσκησης ήταν η πρώτη επαφή με τους μικροελεγκτές AVR της οικογένειας ATMEΛ, αλλά και η εξοικείωση με το περιβάλλον ATMEΛ Studio με το οποίο και θα εργαζόμαστε.

## Περιγραφή

Η πρώτη εργαστηριακή άσκηση είχε στόχο την ανάπτυξη ενός απλού χρονομετρητή ενός δευτερολέπτου με δύο τρόπους. Ο πρώτος τρόπος ήταν με την χρήση βρόγχου κατάλληλου αριθμού επαναλήψεων, ενώ ο δεύτερος με την χρήση χρονομετρητών/μετρητών (Timers). Επιτρεπόμενες γλώσσες για ανάπτυξη του κώδικα ήταν C, Assembly.

## Υλοποίηση

Κατά την υλοποίηση μας χρησιμοποιήσαμε τον μικροελεγκτή **ATmega16a** όπως και προτάθηκε από την εκφώνηση της άσκησης. Ακόμη, και στις δύο περιπτώσεις, χρησιμοποιήσαμε LED το οποίο και αναβοσβήναμε με συχνότητα 1sec. Η γλώσσα προγραμματισμού που χρησιμοποιούσαμε ήταν η C, ενώ χρησιμοποιήσαμε και την Assembly για την εξαγωγή αποτελεσμάτων και ανάλυση του κώδικά μας.

### **Βρόγχος επαναλήψεων**

Κατά την περίπτωση του βρόγχου επαναλήψεων έπρεπε να βρούμε πόσες επαναλήψεις έπρεπε να γίνουν για την καθυστέρηση ενός δευτερολέπτου όπως ζητήθηκε. Η σχέση που πρέπει να ικανοποιείται είναι

$$T_{delay} = (N_{mc} * L_{cnt} - 1) * t_{mc} \text{ όπου}$$

$T_{delay}$ : Επιθυμητή καθυστέρηση

$t_{mc}$ : περίοδος ενός κύκλου ρολογίου ( $F_{clock} = 1\text{MHz}$ )

$N_{mc}$ : αριθμός κύκλων ανά επανάληψη

$L_{cnt}$ : αριθμός επαναλήψεων που πρέπει να τρέξουν

Από τα παραπάνω καταλαβαίνουμε ότι για καθυστέρηση 1sec και έχοντας γνώση των αριθμών κύκλων την κάθε επανάληψη μπορούμε να υπολογίσουμε τις ζητούμενες επαναλήψεις. Για τον υπολογισμό των κύκλων σε κάθε επανάληψη γράψαμε κώδικα σε C και παρατηρήσαμε την παραγόμενη Assembly, ενώ συμβουλευτήκαμε και το Set manual του συγκεκριμένου μικροελεγκτή για να διαπιστώσαμε τελικά ότι εκτελούνταν 6 κύκλοι/επανάληψη. Συγκεκριμένα, συναντήσαμε τις εντολές SBIW, SBC και BRNE (βλ. Παράρτημα για κύκλους κάθε

εντολής). Αντικαθιστώντας όλα τα παραπάνω προέκυψε τελικά ότι ο ζητούμενος αριθμός επαναλήψεων ήταν 166.666.

Αξίζει να σημειωθεί ότι ο μετρητής μας δεν είναι απόλυτα ακριβής καθώς δεν λαμβάνουμε υπόψιν μας τους ζητούμενους κύκλους για να ανάψουμε το LED και δεν αφαιρούμε την συγκεκριμένη καθυστέρηση (2 κύκλοι ανά sec) το οποίο όμως δεν επηρεάζει αισθητά την απόδοση του χρονομετρητή μας καθώς έχουμε απώλεια της τάξης μς.

Αναλυτικά στο παράρτημα ακολουθούν όλες οι διαφορετικές εντολές σε Assembly που συναντήθηκαν μαζί με τους κύκλους που απαιτούν.

Η επιβεβαίωση των ισχυρισμών μας επιβεβαιώθηκε κατά το simulation του κώδικα με την χρήση breakpoints. Μετά από reset του Cycle counter και του Stop watch στο πλαίσιο Process status παρατηρούμε τους κύκλους ανά επανάληψη αλλά και τον απαιτούμενο χρόνο επαληθεύοντας όσα προαναφέραμε. Από το πλαίσιο I/O παρατηρούμε την λειτουργία των LED για συγκεκριμένο PORT

**I/O**

Name	Value
CPU Registers (CPU)	
EEPROM (EEPROM)	
External Interrupts (EXINT)	
I/O Port (PORTA)	
<b>I/O Port (PORTB)</b>	0x01
I/O Port (PORTC)	
I/O Port (PORTD)	
JTAG Interface (JTAG)	
Serial Peripheral Interface (...)	
Timer/Counter, 16-bit (TC1)	
Timer/Counter, 8-bit (TC0)	
Timer/Counter, 8-bit Asyn...	

Name	Address	Value	Bits
PINB	0x36	0x01	00000001
DDRB	0x37	0x01	00000001
PORTB	0x38	0x01	00000001

**Processor Status**

Name	Value
Program Counter	0x0000003F
Stack Pointer	0x045B
X Register	0x0000
Y Register	0x045F
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	1000002
Frequency	1.000 MHz
Stop Watch	1,000,002.00 μs

**Registers**

## Χρονομετρητής

Για την υλοποίηση με timer η λογική μας δεν διαφοροποιείται σε σχέση με την προηγούμενη υλοποίηση. Στην συγκεκριμένη περίπτωση έπρεπε να εξοικειωθούμε με τις δυνατότητες των timers έχοντας την ελευθερία να επιλέξουμε όποια υλοποίηση επιθυμούσαμε. Όλες μας οι επιλογές βασίστηκαν στο datasheet του συγκεκριμένου μικροελεγκτή.

Στην υλοποίηση μας επιλέξαμε να χρησιμοποιήσουμε Timer1 των 16 bit ενώ επιλέξαμε τον Timer1B καθώς επιθυμούσαμε και την επιλογή του prescaler. Ο Prescaler είναι κυκλωματικό κομμάτι του timer που επιτρέπει να το σήμα του ρολογιού να διαιρεθεί με μια δύναμη του 2

δίνοντας μας μεγαλύτερο εύρος με αντάλλαγμα την ακρίβεια. Επιλέξαμε τον prescaler 64 ο οποίος για να ενεργοποιηθεί έπρεπε να μετατρέψουμε σε 1 τις τιμές των CS10 και CS11.

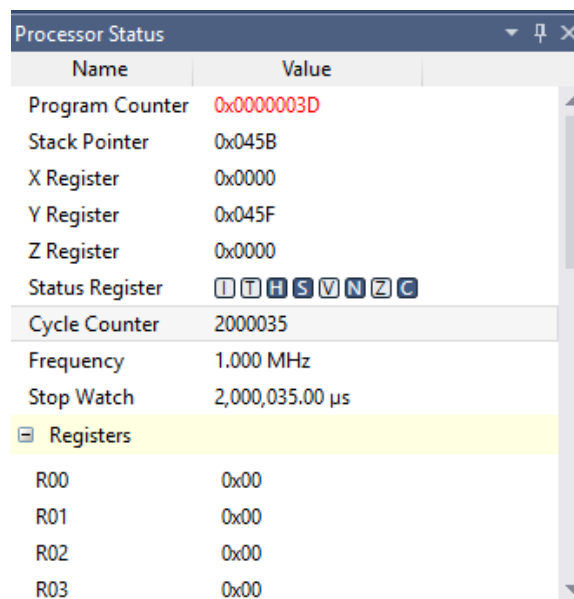
Ακόμη, αξιοποιήσαμε τον TCNT(Timer/Counter Register) που αποτελεί ουσιαστικά τον μετρητή μας. Όπως και την προηγούμενη περίπτωση θα βρούμε ένα όριο εναλλαγής των LED και σε αυτή την περίπτωση δίνεται από την σχέση

$$TimerCount = \frac{Delay}{ClockPeriod} - 1$$

Η διαίρεση για την αξιοποίηση του prescaler γίνεται με την συχνότητα του ρολογιού. Το όριο που προκύπτει μετά την λύση της εξίσωσης είναι 15.624.

Μετά από ανάλυση της Assembly βλέπουμε και πάλι ότι εκτελούνται και πάλι 6 κύκλοι/επανάληψη εντολές IN, CPI, SBCI, BRCS αυτή την φορά όμως για αρκετά μικρότερο αριθμό επαναλήψεων. (βλ. παράσταση για όλες τις εντολές και τους κύκλους που απαιτήθηκαν). Και σε αυτό το ερώτημα ο μετρητής δεν είναι απόλυτα ακριβής για αντίστοιχο λόγο όπως το προηγούμενο ερώτημα.

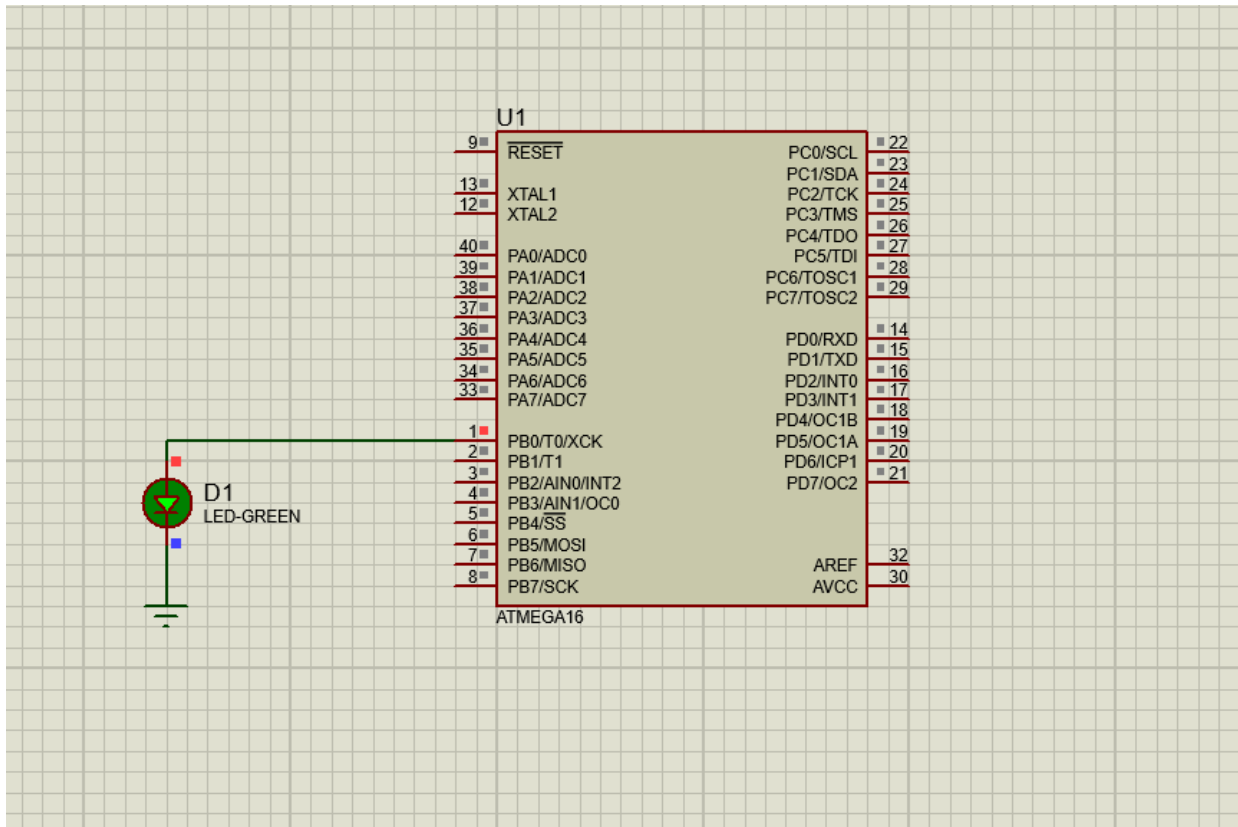
Όλοι οι παραπάνω ισχυρισμοί φαίνονται κατά την προσομοίωση του κώδικα παρακάτω



Name	Value
Program Counter	0x000003D
Stack Pointer	0x045B
X Register	0x0000
Y Register	0x045F
Z Register	0x0000
Status Register	
Cycle Counter	2000035
Frequency	1.000 MHz
Stop Watch	2,000,035.00 μs
Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00

## Προσομοίωση Proteus

Εκτός από την βασική προσομοίωση που επιτρέπει το Atmel Studio πραγματοποιήθηκε και οπτική προσομοίωση στο λογισμικό Proteus 8 Professional. Οι κώδικες και από τα δύο μέρη επιτρέπουν την προσομοίωση τους στην ίδια τρέχοντας κάθε φορά τον αντίστοιχο κώδικα. Ενδεικτικά φαίνεται παρακάτω η συγκεκριμένη υλοποίηση η οποία επισυνάπτεται στα απεσταλμένα αρχεία.



## Σύνοψη

Στο πρώτο εργαστήριο του μαθήματος ήρθαμε για πρώτη φορά σε επαφή με το Atmel Studio ενώ εξοικειωθήκαμε και με τα βασικά χαρακτηριστικά των μικροελεγκτών AVR μελετώντας τις ιδιότητες των timers

## Παράρτημα

SIGNAL	#CYCLES
OUT	1
IN	1
BRNE	1 If false / 2 true
LD1	1
SBC	1
SBWI	2
RET	4 for 16bit devices
RJMP	2
CPI	1
SBCI	1
BRCS	1 If false / 2 true