



ΜΑΘΗΜΑ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ (ΡΟΗ Υ)

ΕΡΓΑΣΙΑ

3^Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΣΠΟΥΔΑΣΤΕΣ

ΚΑΡΑΜ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΚΟΛΙΟΣ ΑΠΟΣΤΟΛΟΣ

2022-23

Οι κώδικες όλων των ασκήσεων περιέχονται στο zippered αρχείο.

Ζήτημα 3.1

Η υλοποίηση του προγράμματος είναι παρόμοια με αυτή της προηγούμενης εργαστηριακής άσκησης. Αυτή τη φορά όμως για να δημιουργήσουμε την απαιτούμενη καθυστέρηση των τεσσάρων και του μισού δευτερολέπτου, δεν χρησιμοποιούμε την κλήση της συνάρτησης `delay`, αλλά το υλοποιούμε μέσω των χρονιστών και συγκεκριμένα μέσω του χρονιστή `Timer1`. Αφού ενεργοποιήσουμε την ενεργοποίηση διακοπής υπερχειλίσας του μετρητή `TCNT1`, επιλέγουμε την συχνότητα αύξησης του χρονιστή ίση με $1/1024$ την συχνότητας ρολογιού του μικροελεγκτή. Στην εκπαιδευτική μας κάρτα που η συχνότητα ρολογιού είναι 16MHz , η επιλογή αυτή ισοδυναμεί με συχνότητα αύξησης του μετρητή ίση με $16\text{MHz}/1024=15625\text{Hz}$. Αφού λοιπόν θέλουμε να δημιουργήσουμε σήμα διακοπής υπερχειλίσας μετά από 4 δευτερόλεπτα, πρέπει ο χρονιστής να ρυθμιστεί για μέτρημα $4 \times 15625=62500$ κύκλων και $0,5 \times 15625=7812,5$ κύκλων για το μισό δευτερόλεπτο αντίστοιχα. Επειδή η υπερχειλίση γίνεται μετά από 65536 κύκλους, αρχικοποιούμε τον μετρητή με την τιμή $65536-62500=3036$ για τα 4 δευτερόλεπτα και $65536-7812,5=57723,5$ για το μισό δευτερόλεπτο.

Θα εξηγήσουμε απλά την λειτουργία του προγράμματος. Αρχικά το πρόγραμμα ελέγχει εάν έχει πατηθεί το κουμπί `PC5` και αν ναι αφού εφαρμόσουμε μία μικρή καθυστέρηση για τον σπινθηρισμό, αρχικοποιούμε τον χρονιστή για υπερχειλίση σε 4 δευτερόλεπτα και έπειτα ανάβουμε το `Led`. Την πρώτη φορά που θα πατήσει το κουμπί θα αυξηθεί κατά ένα ο καταχωρητής `r18` έτσι ώστε εάν ξανά πατηθεί το κουμπί να κάνει άλμα και να εκτελέσει την ρουτίνα `cont1` η οποία ανάβει για μισό δευτερόλεπτο όλα τα `Led` και έπειτα ξανά ανανεώνει τον χρονιστή για υπερχειλίση σε 4 δευτερόλεπτα. Για να ανάψουν όλα τα `Led` για μισό δευτερόλεπτο και έπειτα να ανανεωθεί ο χρόνος των 4 δευτερολέπτων, αυξάνουμε τον καταχωρητή `r20` κατά ένα και περιμένουμε εωσότου τον μηδενίσει η ρουτίνα εξυπηρέτησης του χρονιστή. Τέλος, η ρουτίνα εξυπηρέτησης του χρονιστή είναι αυτή η οποία θα αναλάβει να σβήσει τα `Led`.

Ζήτημα 3.2

Ο πίνακας με τις διάφορες τιμές του Duty Cycle που έχει υπολογιστεί φαίνεται παρακάτω:

% Duty Cycle	Decimal	Hexadecimal
2	5	05
10	26	1A
18	46	2E
26	67	43
34	87	57
42	108	6C
50	128	80
58	148	94
66	167	A7
74	189	BD
82	210	D2
90	230	E6
98	251	FB

Οι τιμές αυτές αποθηκεύονται εκ των προτέρων σε έναν πίνακα σε αύξουσα σειρά, στη μνήμη του μικροελεγκτή ώστε να μην υπολογίζονται κατά τη διάρκεια εκτέλεσης του κώδικα. Το μέγιστο Duty Cycle (100%) είναι ίσο με 256, επειδή ο TMR1A έχει ρυθμιστεί σε λειτουργία 8-bit. Η λειτουργία των προγραμμάτων και στις δύο γλώσσες είναι αρκετά απλή. Αφού ορίσουμε τον πίνακα με τις τιμές του Duty Cycle, αρχικοποιούμε στο 50% (6^η θέση στον πίνακα) και ελέγχουμε εάν πατιέται κάποιο από τα δύο κουμπιά αύξησης ή μείωσής του. Ο πίνακας στην C ορίζεται ως unsigned char για να μην καταλαμβάνει πολύ χώρο και η αρίθμηση της θέσης των στοιχείων του ξεκινάει από το 0 έως το 12. Στην Assembly, οι θέσεις μνήμης στις οποίες βρίσκονται αποθηκευμένες οι τιμές του Duty Cycle ορίζονται μέσω της εντολής .DW και προσπελάζονται μέσω του διπλού καταχωρητή ZH και ZL, στον οποίο φορτώνεται κάθε φορά η θέση του στοιχείου που θέλουμε να επιλέξουμε, όμως διπλασιασμένη μέσω της εντολής lsl και τελικά το στοιχείο αντιγράφεται στον καταχωρητή r0 μέσω της εντολής lpm.

Ζήτημα 3.3

Η ακραία τιμή TOP του καταχωρητή ICR1 υπολογίζεται μέσω της σχέσης:

$$f_{PWM} = \frac{f_{clk}}{N(1 + TOP)}$$

Όπου f_{PWM} η συχνότητα των παλμών της κυματομορφής εξόδου, ίση με 125, 250, 500 ή 1000Hz ανάλογα με το κουμπί που είναι πατημένο, f_{clk} η συχνότητα του ρολογιού του συστήματος, ίση με 16MHz και N η μεταβλητή που αντιπροσωπεύει την τιμή του prescaler ίση με 8.

Παρακάτω φαίνεται ο πίνακας με τους υπολογισμούς για κάθε μία από τις τέσσερις συχνότητες:

Συχνότητα	TOP(=ICR1)
125 Hz	0x3E7F
250 Hz	0x1F3F
500 Hz	0x0F9F
1000 Hz	0x07CF

Για να είναι το Duty Cycle ίσο με 50% για κάθε PWM κυματομορφή, πριν θέσουμε το ICR1 στην κατάλληλη τιμή για την κάθε περίπτωση, θέτουμε τον καταχωρητή OCR1A ίσο με το μισό του ICR1, επειδή ισχύει ότι $DC = \frac{OCRA}{ICR1}$ και εμείς το θέλουμε ίσο με το μισό (50%).