Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΑΚΑΔ. ΕΤΟΣ 2022-2023

ΑΘΗΝΑ, 3 Νοεμβρίου 2022

3η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"
Χρήση εξωτερικών διακοπών στον Μικροελεγκτή AVR

Εξέταση – Επίδειξη: Τετάρτη 9/11/2022
Προθεσμία για παράδοση Έκθεσης: Κυριακή 13/11/2022 (23:59

Εισαγωγή

Χρονιστής *Timer*1

Ένα πλεονέκτημα των σύγχρονων μικροελεγκτών είναι η ενσωμάτωση στην ίδια ψηφίδα χρήσιμων περιφερειακών συσκευών, με σύνδεση στο σύστημα διακοπών του Μικροελεγκτή. Για παράδειγμα, οι χρονιστές είναι καταχωρητές που μπορούν να προγραμματιστούν να προκαλέσουν διακοπή στον μικροελεγκτή μετά την πάροδο συγκεκριμένου χρόνου.

Ο μικροελεγκτής ATmega328PB διαθέτει 8-ψήφιους 16-ψήφιους χρονιστές. Στους χρονιστές αυτούς μπορεί να τοποθετηθεί μια αρχική τιμή η οποία αυξάνεται από το μικροελεγκτή με επιλεγμένη συχνότητα. Όταν ένας χρονιστής υπερχειλίσει, μπορεί να δημιουργήσει κατάλληλο σήμα διακοπής εφόσον τροποποιηθεί το περιεχόμενο του καταχωρητή TIMSK1 σύμφωνα με το παρακάτω σχήμα. Γράφοντας 1 στο ψηφίο TOIE1 επιτρέπονται διακοπές υπερχείλισης του χρονιστή TCNT1 (εφόσον επιτραπούν γενικά οι διακοπές με την εντολή sei).

TIMSK1 (offset =0x6F)

Bit	7	6	5	4	3	2	1	0
			ICIE1			OCIE1B	OCIE1A	TOIE1
Access			R/W			R/W	R/W	R/W
Reset			0			0	0	0

Για παράδειγμα, με τις παρακάτω εντολές επιτρέπεται η διακοπή υπερχείλισης του μετρητή TCNT1.

```
ldi r24, (1<<TOIE1)      ; ενεργοποίηση διακοπής υπερχείλισης του μετρητή TCNT1
sts TIMSK1, r24          ; για τον timer1
```

Υπενθύμιση: Όποιος καταχωρητής έχει διεύθυνση μεγαλύτερη από 0x60 χρησιμοποιείται η εντολή STS και ό χι η εντολή OUT αφού πρέπει να γίνει αποθήκευση στην SRAM

Ο χρονιστής TCNT1 (TCNT1L, offset = 0x84 και TCNT1H, offset = 0x85) είναι 16 bit. Για την επιλογή της συχνότητας αύξησης του χρησιμοποιείται ο καταχωρητής TCCR1B σύμφωνα με τα δυο παρακάτω σχήματα:

TCCR1B (offset =0x81)

Bit	7	6	5	4	3	2	1	0
	ICNC1	ICES1		WGM13	WGM12	CS12	CS11	CS10
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

CS12	CS11	CS10	**Περιγραφή σήματος εισόδου χρονιστή**
0	0	0	Κανένα. Χρονιστής σταματημένος
0	0	1	CLK
0	1	0	CLK/8
0	1	1	CLK/64
1	0	0	CLK/256
1	0	1	CLK/1024
1	1	0	Κατερχόμενη ακμή εξωτερικού σήματος ακροδέκτη T1
1	1	1	Ανερχόμενη ακμή εξωτερικού σήματος ακροδέκτη T1

Για παράδειγμα, με τις παρακάτω εντολές επιλέγεται συχνότητα αύξησης του χρονιστή TCNT1 ίση με τη 1/1024 της συχνότητας ρολογιού του μικροελεγκτή (δηλαδή, κάθε 1024 κύκλους ρολογιού αυξάνεται κατά 1 ο χρονιστής TCNT1).

```
ldi r24 ,(1<<CS12) | (0<<CS11) | (1<<CS10)      ; CK/1024
sts TCCR1B, r24
```

Στην εκπαιδευτική κάρτα ntuAboard_G1 με συχνότητα ρολογιού 16MHz, η επιλογή αυτή ισοδυναμεί με συχνότητα αύξησης του TCNT1 ίση με 16MHz/1024=15625Hz. Αν με αυτές τις επιλογές θέλουμε ο TCNT1 να δημιουργήσει σήμα διακοπής υπερχείλισης μετά από 3sec, πρέπει να ρυθμιστεί για μέτρημα 3×15625=46875 κύκλων. Επειδή η υπερχείλιση γίνεται μετά από 65536 κύκλους (16 ψηφία), θα πρέπει η αρχική τιμή που θα του δοθεί πριν αρχίσει τη μέτρηση προς τα πάνω να είναι 65535-46875=18660. Αυτό γίνεται με τον παρακάτω κώδικα:

```
ldi r24, HIGH(18660)      ; αρχικοποίηση του TCNT1
out TCNT1H, r24           ; για υπερχείλιση μετά από 3 sec
ldi r24, LOW(18660)
out TCNT1L, r24
```

Προσοχή: Τώρα έχουμε out και όχι sts
Γιατί όμως; Τλκ έχει λάθος. Είναι sts

Για μέγιστη ασφάλεια, τα δύο τμήματα του μετρητή TCNT1, TCNT1H και TCNT1L πρέπει να διαβάζονται αδιαίρετα, χωρίς να μεσολαβήσει για παράδειγμα κάποια άλλη διακοπή. Για το λόγο αυτό ο μικροελεγκτής κάνει μια ειδική διαδικασία. Όταν μια τιμή γράφεται στον TCNT1H, αυτή τοποθετείται στον προσωρινό καταχωρητή TEMP. Στη συνέχεια, όταν γραφεί τιμή στον TCNT1L η τιμή που υπάρχει στον TEMP συνδυάζεται με αυτή και τα 16 ψηφία γράφονται ταυτόχρονα σε όλο το μήκος του TCNT1. Κατά την ανάγνωση, όταν διαβάζεται μια τιμή από τον TCNT1L αυτή τοποθετείται στον επιλεγμένο καταχωρητή του μικροελεγκτή και ταυτόχρονα η τιμή του TCNT1H μεταφέρεται στον καταχωρητή TEMP. Όταν στην συνέχεια διαβαστεί και ο TCNT1H, μεταφέρεται στον επιλεγμένο καταχωρητή η τιμή που έχει τοποθετηθεί στον TEMP.

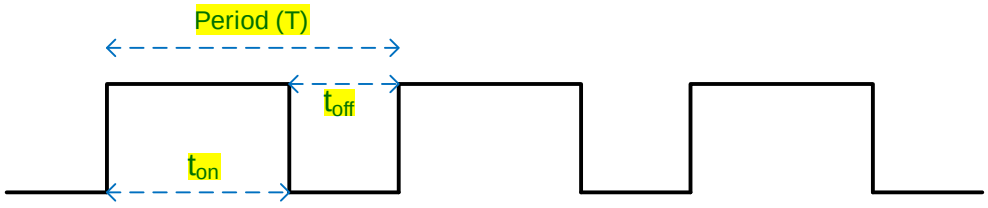
Με βάση αυτή τη διαδικασία κατά την εγγραφή πρέπει πάντα να γράφεται πρώτα η τιμή στον TCNT1H και μετά στον TCNT1L ενώ κατά την ανάγνωση πρέπει πάντα να διαβάζεται πρώτα ο TCNT1L και μετά ο TCNT1H. Σε πολύπλοκες εφαρμογές πριν την πρόσβαση στον TCNT1 μπορούν να απενεργοποιηθούν και οι διακοπές.
Παρόμοια τεχνική μπορεί πρόσβασης μπορεί να χρησιμοποιηθεί και για άλλους 16-bit καταχωρητές όπως οι OCR1A, OCR1B και ICR1.

Η θέση μνήμης του διανύσματος διακοπής του Χρονιστή TCNT1 φαίνεται στον παρακάτω κώδικα.

```
.org 0x1A
rjmp ISR_TIMER1_OVF          ; ρουτίνα εξυπηρέτησης της διακοπής υπερχείλισης του timer1
```

Διαμόρφωση εύρους παλμών (Pulse Width Modulation)

Μία PWM (Pulse Width Modulation) κυματομορφή είναι μία τετραγωνική περιοδική κυματομορφή η οποία έχει δύο τμήματα. Το τμήμα ON στο οποίο η κυματομορφή μία μέγιστη τιμή και το τμήμα OFF στο οποίο έχει μία ελάχιστη τιμή. Η περίοδος της κυματομορφής είναι σταθερή ενώ οι χρόνοι t_{on} και t_{off} μεταβάλλονται.



Ο βαθμός χρησιμοποίησης (Duty Cycle) συμβολίζεται με DC και ορίζεται σύμφωνα με τον τύπο:

$$DC = \frac{t_{on}}{T}$$

Διαμόρφωση εύρους παλμών με τον ATmega328PB

Ο ATmega328PB διαθέτει διάφορους μετρητές (Timer/Counters) οι οποίοι μπορούν να χρησιμοποιηθούν για την παραγωγή PWM κυματομορφών τάσης, με μεταβαλλόμενη συχνότητα και Duty Cycle. Από αυτούς ο Timer/Counter1(TCNT1) έχει δύο εξόδους και μπορεί να ρυθμιστεί ως 8-bit ή ως 16-bit. Οι διάφοροι Timer/Counters έχουν παρόμοιο τρόπο λειτουργίας PWM. Εδώ θα εξεταστεί ένας τρόπος λειτουργίας με τον οποίο παράγεται μια PWM κυματομορφή τάσης υψηλής συχνότητας. Στη λειτουργία αυτή (Fast PWM Mode) ο TCNT1 αυξάνεται ξεκινώντας από την τιμή BOTTOM και όταν φτάσει την τιμή TOP τότε παίρνει ξανά την τιμή BOTTOM και η διαδικασία επαναλαμβάνεται.

Από (**) μπορούμε να έχουμε τον τρόπο λειτουργία PWM όπως περιγράφεται παραπάνω, δηλαδή ξεκινάμε από μια bottom τιμή, αυξάνουμε κατά ένα μέχρι μια top τιμή και μετά ξαναξεκινάει η μέτρηση από το bottom. Άρα Tpwm = (top - bottom + 1)*Τχρονιστή. Προσθέτουμε +1 γιατί εμπεριέχουμε το ακραίο σημείο overflow στην περίοδο. Το Τχρονιστή είναι η περίοδος αύξησης του χρονιστή. Άρα fpwm = ftimer / (top-bottom+1)

Η κυματομορφή εξόδου θα είναι παλμοί με σταθερή συχνότητα (f_{PWM}) η τιμή της οποίας εξαρτάται από τη συχνότητα του ρολογιού του συστήματος (f_{clk}) και την αρχικοποίηση του prescaler όπως προκύπτει από τον παρακάτω τύπο:

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

Μέσω του TCCR1B μπορούμε να ορίσουμε την συχνότητα του χρονιστή ως κλάσμα της συχνότητας του ρολογιού. Δηλαδή ftimer = fclock / N
Άρα fpwm = fclock / (N*(top-bottom+1)
Αν bottom = 0 έχουμε την (*)

Η μεταβλητή N αντιπροσωπεύει την τιμή του prescaler (1, 8, 64, 256 ή 1024).

Το Duty Cycle ρυθμίζεται, μέσω του καταχωρητή OCR1A. Το Low byte OCR1AL έχει Offset 0x88 και το Hi byte OCR1AH έχει Offset 0x89. Οι ακραίες τιμές για τον καταχωρητή OCR1A είναι η BOTTOM, όπου η έξοδος θα είναι ένας εξαιρετικά μικρός παλμός στην αρχή κάθε περιόδου και η TOP όπου η έξοδος θα είναι σταθερά υψηλή. Στην ανάστροφη λειτουργία, οι κυματομορφές εξόδου είναι αντεστραμμένες.

TCCR1A (offset=0x80)

Bit	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	COM1B1	COM1B0			WGM11	WGM10
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

TCCR1B (offset=0x81)

Bit	7	6	5	4	3	2	1	0
	ICNC1	ICES1		WGM13	WGM12	CS12	CS11	CS10
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Παράδειγμα 3.1

Στη συνέχεια παρουσιάζεται ένα παράδειγμα, σε γλώσσα προγραμματισμού C, παραγωγής παλμών PWM στον ακροδέκτης PB1 για την εκπαιδευτική κάρτα ntuAboard_G1. Η ρουτίνα PWM_init() αρχικοποιεί κατάλληλα τον χρονιστή TMR1A για να λειτουργεί σε Fast PWM Mode, με μη ανάστροφη λειτουργία και με τιμή 8 για τον prescaler. Ο TMR1B δεν χρησιμοποιείται. Στον του ακροδέκτη PB1 είναι συνδεδεμένο ένα LED. Η ρουτίνα main() αυξομειώνει συνεχώς το Duty Cycle της PWM κυματομορφής αυξομειώνοντας αντίστοιχα την φωτεινότητα του LED που είναι συνδεμένο στον ακροδέκτη PB1.

```c
#define F_CPU 16000000UL
#include "avr/io.h"
#include <util/delay.h>

int main ()
{
    unsigned char duty;

    //set TMR1A in fast PWM 8 bit mode with non-inverted output
    //prescale=8
    TCCR1A = (1<<WGM10) | (1<<COM1A1);
    TCCR1B = (1<<WGM12) | (1<<CS11);

    DDRB |= 0b00111111;     //set PB5-PB0 pins as output

    while (1)
    {
        for(duty=0; duty<255; duty++)
        {
            OCR1AL=duty;  //increase the LED2 light intensity
            _delay_ms(10);
        }
        for(duty=255; duty>1; duty--)
        {
            OCR1AL=duty;  //increase the LED2 light intensity
            _delay_ms(10);
        }
    }
}
```

8 bit PWM, max FF, fpwm = fclock/510

κανονική PWM

επιλογή σήματος χρονισμού σε fclock/8

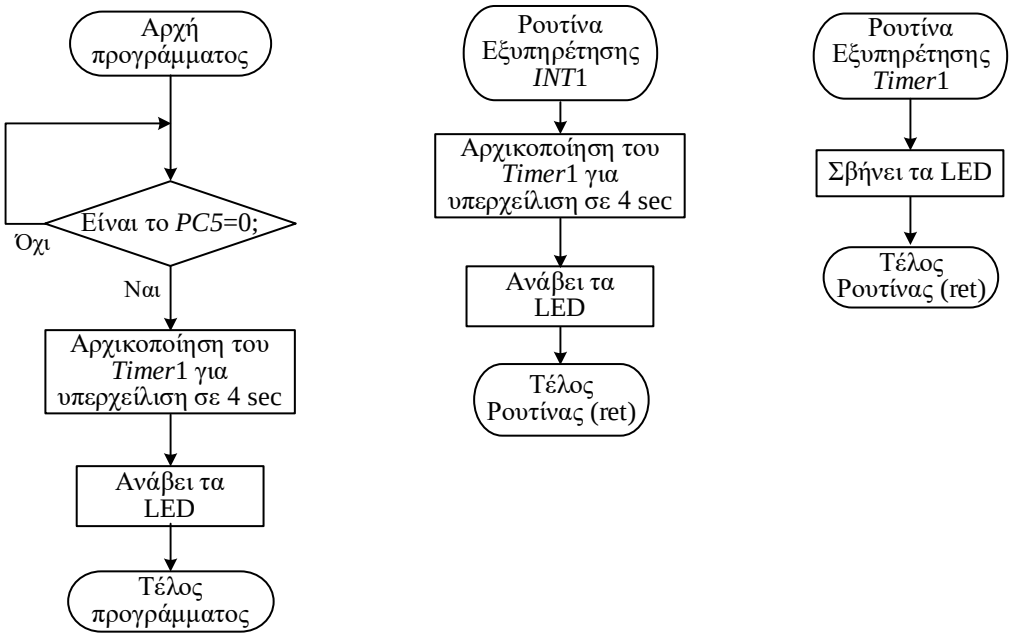
Δες βιβλίο σελίδα 96, 97 και 106

ο μετρητής μηδενίζεται αν η σύγκριση είναι επιτυχής (**)

Γιατί μέχρι 255; Γιατί το μέγιστο για WGM10 είναι FF
Δες βιβλίο σελίδα 106

Ζήτημα 3.1

Να υλοποιηθεί αυτοματισμός που να ελέγχει το άναμμα και το σβήσιμο ενός φωτιστικού σώματος. Όταν πατάμε το push button **PD3** (δηλαδή με την ενεργοποίηση της **INT1**) ή το **PC5** (που υποθέτουμε ότι αντιστοιχεί σε ένα αισθητήρα κίνησης) να ανάβει το **led PB0** της θύρας **PORTB** (που υποθέτουμε ότι αντιπροσωπεύει το φωτιστικό σώμα). Το led θα σβήνει μετά από 4 sec, εκτός και αν ενδιάμεσα υπάρξει νέο πάτημα του PD3 ή PC5, οπότε και ο χρόνος των 4 sec θα ανανεώνεται. Κάθε φορά που γίνεται ανανέωση να ανάβουν τα led της θύρας **PORTB** (PB5-PB0) για 0.5 sec, μετά να σβήνουν εκτός από το led **PB0** που παραμένει συνολικά για 4 sec εκτός και αν ανανεωθεί. Να γίνει χρήση του Χρονιστή *Timer*1. Μπορείτε να βασιστείτε στο Διάγραμμα ροής που δίνεται στο Σχ. 3.1. Περιγράψτε επίσης την συνολική λειτουργία. Το πρόγραμμα να δοθεί σε **assembly** και σε **C**.



Σχήμα 3.1 Λογικό διάγραμμα αυτοματισμού που ανάβει για 4 sec το led PB0.

Ζήτημα 3.2

Να δημιουργηθεί κώδικας σε γλώσσα assembly και γλώσσα C, ο οποίος να αρχικοποιεί τον TMR1A σε λειτουργία 8-bit και να παράγει μία PWM κυματομορφή στον ακροδέκτη PB1. Θεωρήστε ότι BOTTOM=0. Στον ακροδέκτη PB1 είναι συνδεδεμένο ένα LED και η φωτεινότητα του μεταβάλλεται εάν μεταβληθεί το Duty Cycle της PWM κυματομορφής. Αρχικά το Duty Cycle ρυθμίζεται σε 50%. Στη συνέχεια κάθε φορά που πιέζεται το μπουτόν PD1 το Duty Cycle αυξάνεται κατά 8%. Όταν το Duty Cycle φτάσει τη μέγιστη τιμή 98% η πίεση του μπουτόν PD1 δεν το αυξάνει περεταίρω. Κάθε φορά που πιέζεται το μπουτόν PD2 το Duty Cycle μειώνεται κατά 8%. Όταν το Duty Cycle φτάσει τη ελάχιστη τιμή 2% η πίεση του μπουτόν PD1 δεν το μειώνει περεταίρω. Οι τιμές που πρέπει να πάρει ο καταχωρητής OCR1A, για τις διάφορες τιμές του Duty Cycle, να έχουν υπολογιστεί εκ των προτέρων και να έχουν τοποθετηθεί σε ένα πίνακα στη μνήμη του μικροελεγκτή, έτσι ώστε να μη χρειάζεται να υπολογιστούν κατά τη διάρκεια εκτέλεσης του κώδικα.

Ζήτημα 3.3

Να δημιουργηθεί κώδικας σε γλώσσα assembly και γλώσσα C, οποίος να παράγει μία PWM κυματομορφή στον ακροδέκτη PB1 με duty cycle 50%. Όταν δεν είναι πατημένο κάποιο από τα πλήκτρα PD3 – PD0 τότε δεν παράγεται κυματομορφή στον ακροδέκτη PB1.

Στους καταχωρητές TCCR1A και TCCR1B τα WGM11, WGM12, WGM13 να ρυθμιστούν σε λογικό 1 και το WGM10 σε λογικό 0, έτσι ώστε ο TIM1A να λειτουργεί σε fast PWM mode και TOP=ICR1 (ICR1L offset = 0x86, ICR1H offset = 0x87).

Πόσα bit είναι αυτού του είδους η PWM?

Οι συχνότητες των κυματομορφών που παράγονται, εξαρτώνται από το πλήκτρο που είναι πατημένο και πρέπει να είναι όσο ποιο κοντά γίνεται στις τιμές του παρακάτω πίνακα:

ΠΛΗΚΤΡΟ	ΣΥΧΝΟΤΗΤΑ
PD.0	125
PD.1	250
PD.2	500
PD.3	1000

top = 125, N = 64

top = 500, N = 8

top = 250, N = 8

top = 125, N = 8

top = 1000, N=8

top = 62.5, N = 64

Στην εκπαιδευτική κάρτα ntuAboard_G1,συνδέστε τους βραχυκυκλωτήρες μεταξύ PB1 και J16_1 και μεταξύ J15_5V και J15_BUZZ+ έτσι ώστε να παράγονται στο buzzer ηχητικά σήματα ανάλογα, με τις κυματομορφές στον ακροδέκτη PB1.

Από (*) έχουμε top = fclk/(N*fpwm)-1 με fclk =16MHz για την πλακέτα