

ΕΙΣΑΓΩΓΗ

1. Το λογισμικό MPLAB

Στις εργαστηριακές ασκήσεις που θα ακολουθήσουν για την ανάπτυξη των προγραμμάτων του μικροελεγκτή χρησιμοποιείται το περιβάλλον προγραμματισμού και αποσφαλμάτωσης MPLAB, στην έκδοση 8.0. Το λογισμικό αυτό διανέμεται δωρεάν από τη Microchip και νεότερη έκδοσή του μπορεί να βρεθεί στο δικτυακό τόπο της εταιρείας στην εξής διεύθυνση: <http://www.microchip.com>

Το MPLAB υποστηρίζει όλες τις οικογένειες των μικροελεγκτών της Microchip ενώ επίσης συνεργάζεται με πληθώρα προγραμματιστών για μικροελεγκτές PIC ακόμα και από άλλους κατασκευαστές.

Το περιβάλλον αυτό διαθέτει ενσωματωμένο κειμενογράφο (editor), μεταγλωττιστή (assembler) καθώς και προσομοιωτή (simulator) για την αποσφαλμάτωση των προγραμμάτων. Επιπλέον στο MPLAB μπορεί να εγκατασταθεί μεταγλωττιστής (compiler) γλώσσας υψηλότερου επιπέδου όπως C, Basic κλπ.

Η διαδικασία ανάπτυξης ενός προγράμματος για μικροελεγκτή γενικά περιλαμβάνει τα παρακάτω στάδια:

- Συγγραφή κώδικα στον κειμενογράφο του MPLAB
- Μεταγλώττιση και διόρθωση των συντακτικών λαθών
- Προσομοίωση και έλεγχος σωστής λειτουργίας προγράμματος για τη διόρθωση των λογικών σφαλμάτων
- Φόρτωση του προγράμματος στο μικροελεγκτή
- Έλεγχος σωστής λειτουργίας κυκλώματος

2. Υλικό προγραμματισμού και αποσφαλμάτωσης

Η ανάπτυξη εφαρμογών βασισμένες σε μικροελεγκτές, ειδικά όταν η πολυπλοκότητα του κώδικα φτάνει σε υψηλά επίπεδα, δεν αποτελεί καθόλου απλή υπόθεση. Όπως σε όλες τις γλώσσες προγραμματισμού έτσι και στους μικροελεγκτές η δυσκολία εντοπίζεται στη διόρθωση των λογικών σφαλμάτων. Για τη διευκόλυνση του

προγραμματιστή έχουν αναπτυχθεί ειδικά εργαλεία για αποσφαλμάτωση (debugging) προγραμμάτων τα οποία συνοψίζονται στις παρακάτω κατηγορίες.

- Προσομοιωτές (Simulators). Με τα εργαλεία αυτά προσομοιώνεται η εσωτερική λειτουργία του μικροελεγκτή στον υπολογιστή. Μπορούμε να εκτελέσουμε εντολή προς εντολή το πρόγραμμα και να παρατηρούμε ανα πάσα στιγμή ποια είναι η λογική κατάσταση των καταχωρητών μνήμης του μικροελεγκτή καθώς και την κατάσταση πολλών από τα περιφερειακά του.
- Εξομοιωτές (Emulators). Στα εργαλεία αυτά εξομοιώνεται πλήρως η λειτουργία του μικροελεγκτή εντός του κυκλώματος (In-Circuit). Περιλαμβάνουν υλικό (hardware) το οποίο τοποθετείται στο κύκλωμα αντί του μικροελεγκτή. Το υλικό αυτό συμπεριφέρεται με τον ίδιο ακριβώς τρόπο με τη διαφορά ότι επικοινωνεί ταυτόχρονα με υπολογιστή, στέλνοντας του όλες τις πληροφορίες σχετικά με την τρέχουσα κατάσταση στην οποία θα βρισκόταν ο μικροελεγκτής εάν ήταν εκείνος συνδεδεμένος στο κύκλωμα.

Η δυνατότητα διόρθωσης σφαλμάτων εντός του κυκλώματος που διαθέτουν οι εξομοιωτές, τους καθιστά ισχυρά εργαλεία για τον προγραμματιστή. Τα εργαλεία αυτά όμως αποτελούν ιδιαίτερα ακριβές λύσεις στον τομέα της αποσφαλμάτωσης. Οι μικροελεγκτές PIC διαθέτουν επιπλέον μια δυνατότητα η οποία αποτελεί τη βάση για τη χρησιμοποίηση φθηνότερων εργαλείων ανάπτυξης εφαρμογών. Περιέχουν ενσωματωμένα κυκλώματα τα οποία χρησιμοποιούνται για In-Circuit Debugging (αποσφαλμάτωση εντός κυκλώματος). Για την πραγματοποίηση της αποσφαλμάτωσης δεσμεύονται ορισμένοι από τους πόρους του μικροελεγκτή (οι ακροδέκτες RB6, RB7, ορισμένες από τις διαθέσιμες θέσεις μνήμης προγράμματος και δεδομένων κλπ) και πλέον δεν είναι διαθέσιμοι στον προγραμματιστή.

3. Ο μικροελεγκτής PIC16F887

Πριν γίνει αναφορά συγκεκριμένα στο μικροελεγκτή PIC16F887 κρίνεται σκόπιμο να δοθεί πρώτα ένας ορισμός για το τι είναι ένας μικροελεγκτής. Γενικά μπορούμε να πούμε ότι:

Μικροελεγκτής είναι ένα ολοκληρωμένο κύκλωμα το οποίο περιλαμβάνει μικροεπεξεργαστή, έχει ενσωματωμένη μνήμη και περιφερειακά ενώ επίσης διαθέτει διάφορες θύρες εισόδου/εξόδου για επικοινωνία με εξωτερικές συσκευές.

Θα μπορούσε να παρομοιαστεί με έναν μικροϋπολογιστή. Όπως ακριβώς ένας μικροϋπολογιστής αποτελείται από επεξεργαστή, μνήμη, περιφερειακές συσκευές και εκτελεί προγράμματα, έτσι κι ένας μικροελεγκτής διαθέτει τα παραπάνω χαρακτηριστικά και μάλιστα είναι ενσωματωμένα σε ένα μόνο ολοκληρωμένο κύκλωμα.

Σήμερα οι μικροελεγκτές χρησιμοποιούνται πολύ σε ευρύτατο φάσμα εφαρμογών στη βιομηχανία, στις τηλεπικοινωνίες και γενικότερα όπου απαιτείται ψηφιακός έλεγχος. Υπάρχουν δεκάδες τύποι μικροελεγκτών που κυκλοφορούν στο εμπόριο άλλοι ιδιαίτερα ισχυροί και άλλοι με λιγότερες δυνατότητες. Ένας μικροελεγκτής επιλέγεται για την υλοποίηση μιας εφαρμογής με βάση δύο κριτήρια:

- Τις απαιτούμενες δυνατότητες ώστε να καλύπτονται οι ανάγκες της εφαρμογής
- Το κόστος

Ο μικροελεγκτής PIC16F887 συνδυάζει το χαμηλό κόστος με μια πληθώρα δυνατοτήτων όπως:

- Flash μνήμη προγράμματος
- Πέντε θύρες εισόδου/εξόδου
- Μνήμη δεδομένων EEPROM
- Ενσωματωμένα κυκλώματα για σειριακή επικοινωνία (SPI, I²C, EUSART)
- Μετατροπέας αναλογικού σήματος σε ψηφιακό (A/D Converter)
- Ενσωματωμένα κυκλώματα Capture/Compare/PWM
- Αναλογικοί συγκριτές (Analog Comparators)
- Κύκλωμα παραγωγής τάσης αναφοράς (Voltage Reference)

Επιπλέον ο μικροεπεξεργαστής που περιλαμβάνει ο PIC16F887 είναι αρχιτεκτονικής RISC (Reduced Instruction Set Computing), ένα χαρακτηριστικό που του προσδίδει εκτέλεση εντολών με μεγάλη ταχύτητα. Κάθε εντολή του μικροελεγκτή εκτελείται σε ένα κύκλο ρολογιού, εκτός των εντολών διακλάδωσης που εκτελούνται σε δύο κύκλους.

2. Οργάνωση μνήμης του μικροελεγκτή PIC16F887

Ο μικροελεγκτής PIC16F887 περιλαμβάνει τις παρακάτω περιοχές μνήμης:

- Μνήμη προγράμματος
- Μνήμη δεδομένων
- Στοίβα (stack)

α) Μνήμη προγράμματος

Η μνήμη προγράμματος είναι τύπου flash. Αυτό σημαίνει ότι η μνήμη διατηρεί τα δεδομένα της όταν διακοπεί η παροχή τροφοδοσίας αλλά επίσης μπορεί να ξαναγράφεται πολλές φορές χωρίς να χρειάζεται προηγουμένως η διαγραφής της. Η μνήμη 8K ($=2^{13}$) θέσεων των 14 bit που διαθέτει ο PIC16F887 χωρίζεται σε τέσσερις υποπεριοχές που ονομάζονται σελίδες μνήμης. Αυτές είναι:

- Page 0 (από h'0000' έως h'1FFF')
- Page 1 (από h'2000' έως h'3FFF')
- Page 2 (από h'4000' έως h'5FFF')
- Page 3 (από h'6000' έως h'7FFF')

β) Μνήμη δεδομένων

Ο μικροελεγκτής PIC16F887 διαθέτει δύο τύπους μνήμης δεδομένων. Τη μνήμη SRAM και τη μνήμη EEPROM.

Η μνήμη Static RAM (Random Access Memory) διατηρεί τα δεδομένα της όσο διαρκεί η παροχή τροφοδοσίας. Χωρίζεται στις παρακάτω τέσσερις υποπεριοχές:

- Bank0 (από h'00 έως h'7F')
- Bank1 (από h'80 έως h'FF')
- Bank2 (από h'100' έως h'17F')
- Bank3 (από h'180' έως h'1FF')

Όπως είναι αντιληπτό κάθε υποπεριοχή μνήμης περιλαμβάνει έως 128 καταχωρητές. Ορισμένοι από αυτούς τους καταχωρητές είναι στη διάθεση του προγραμματιστή για γενική χρήση ενώ κάποιοι άλλοι χρησιμοποιούνται για να καθορίσουν ειδικές λειτουργίες. Επιπλέον κάποιοι καταχωρητές είναι δεσμευμένοι από την αρχιτεκτονική του μικροελεγκτή για μελλοντική χρήση. Οι καταχωρητές μνήμης του μικροελεγκτή PIC16F887 παρουσιάζονται στο χάρτη μνήμης του Πίνακα 1.

Η μνήμη EEPROM (Electrically Erasable Programmable Read Only Memory) διατηρεί τα δεδομένα της ακόμα και μετά από διακοπή της παροχής τροφοδοσίας. Ο μικροελεγκτής PIC16F887 διαθέτει 256 bytes μνήμης EEPROM.

γ) Στοιίβα

Ο μικροελεγκτής PIC16F887 διαθέτει στοίβα 8 θέσεων. Η στοίβα όμως χρησιμοποιείται για να αποθηκεύονται μόνο πληροφορίες της μνήμης προγράμματος και όχι της μνήμης δεδομένων. Επομένως η στοίβα δεν είναι διαθέσιμη στον προγραμματιστή.

BANK0	ΔΙΕΥΘ.	BANK1	ΔΙΕΥΘ.	BANK2	ΔΙΕΥΘ.	BANK3	ΔΙΕΥΘ.
INDF	h'00'	INDF	h'80'	INDF	h'100'	INDF	h'180'
TMR0	h'01'	OPTION_REG	h'81'	TMR0	h'101'	OPTION_REG	h'181'
PCL	h'02'	PCL	h'82'	PCL	h'102'	PCL	h'182'
STATUS	h'03'	STATUS	h'83'	STATUS	h'103'	STATUS	h'183'
FSR	h'04'	FSR	h'84'	FSR	h'104'	FSR	h'184'
PORTA	h'05'	TRISA	h'85'	WDTCON	h'105'	SRCON	h'185'
PORTB	h'06'	TRISB	h'86'	PORTB	h'106'	TRISB	h'186'
PORTC	h'07'	TRISC	h'87'	CM1CON0	h'107'	BAUDCTL	h'187'
PORTD	h'08'	TRISD	h'88'	CM2CON0	h'108'	ANSEL	h'188'
ORTE	h'09'	TRISE	h'89'	CM2CON1	h'109'	ANSELH	h'189'
PCLATH	h'0A'	PCLATH	h'8A'	PCLATH	h'10A'	PCLATH	h'18A'
INTCON	h'0B'	INTCON	h'8B'	INTCON	h'10B'	INTCON	h'18B'
PIR1	h'0C'	PIE1	h'8C'	EEDAT	h'10C'	EECON1	h'18C'
PIR2	h'0D'	PIE2	h'8D'	EEADR	h'10D'	EECON2	h'18D'
TMR1L	h'0E'	PCON	h'8E'	EEDATH	h'10E'	δεσμευμένη θέση	h'18E'
TMR1H	h'0F'	OSCCON	h'8F'	EEADRH	h'10F'	δεσμευμένη θέση	h'18F'
T1CON	h'10'	OSCTUNE	h'90'		h'110'		h'190'
TMR2	h'11'	SSPCON2	h'91'		h'111'		h'191'
T2CON	h'12'	PR2	h'92'		h'112'		h'192'
SSPBUF	h'13'	SSPADD	h'93'		h'113'		h'193'
SSPCON	h'14'	SSPSTAT	h'94'		h'114'		h'194'
CCPR1L	h'15'	WPUB	h'95'	Μνήμη	h'115'	Μνήμη	h'195'
CCPR1H	h'16'	IOCB	h'96'	γενικής	h'116'	γενικής	h'196'
CCP1CON	h'17'	VRCON	h'97'	χρήσης	h'117'	χρήσης	h'197'
RCSTA	h'18'	TXSTA	h'98'		h'118'		h'198'
TXREG	h'19'	SPBRG	h'99'		h'119'		h'199'
RCREG	h'1A'	SPBRGH	h'9A'	16 Bytes	h'11A'	16 Bytes	h'19A'
CCPR2L	h'1B'	PWM1CON	h'9B'		h'11B'		h'19B'
CCPR2H	h'1C'	ECCPAS	h'9C'		h'11C'		h'19C'
CCP2CON	h'1D'	PSTRCON	h'9D'		h'11D'		h'19D'
ADRESH	h'1E'	ADRESL	h'9E'		h'11E'		h'19E'
ADCON0	h'1F'	ADCON1	h'9F'		h'11F'		h'19F'
	h'20'	Μνήμη	h'A0'	Μνήμη	h'120'	Μνήμη	h'1A0'
Μνήμη		γενικής		γενικής		γενικής	
χρήσης		Χρήσης		Χρήσης		Χρήσης	
96 Bytes		80 Bytes	h'EF'	80 Bytes	h'16F'	80 Bytes	h'1EF'
		Μη	h'F0'	Μη	h'170'	Μη	h'1F0'
		διαθέσιμες		διαθέσιμες		διαθέσιμες	
	h'7F'	θέσεις	h'FF'	θέσεις	h'17F'	θέσεις	h'1FF'

Πίνακας 1. Ο χάρτης μνήμης στατικής RAM του μικροελεγκτή PIC16F887.

4. Το σύνολο εντολών του μικροελεγκτή PIC16F887

Το σύνολο εντολών (instruction set) του μικροελεγκτή αποτελείται από 35 εντολές και παρουσιάζεται στον πίνακα 2.

A/A	ΕΝΤΟΛΗ	ΠΕΡΙΓΡΑΦΗ	ΚΥΚΛΟΙ	ΣΗΜΑΙΕΣ
1	addwf f,d	Πρόσθεση ανάμεσα στους καταχωρητές w και f	1	C, DC, Z
2	andwf f,d	Λογικό AND ανάμεσα στους καταχωρητές w και f	1	Z
3	clrf f	Μηδενισμός του καταχωρητή f	1	Z
4	clrw	Μηδενισμός του καταχωρητή w	1	Z

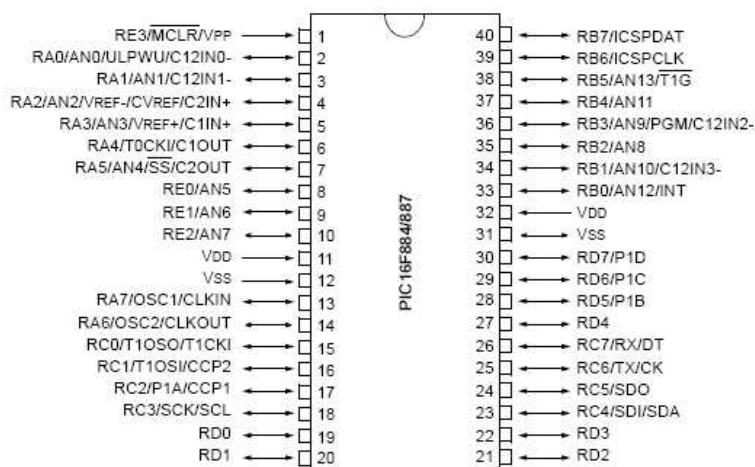
5	comf	f, d	Συμπλήρωμα ως προς '1' του καταχωρητή f	1	Z
6	decf	f, d	Μείωση κατά ένα του καταχωρητή f	1	Z
7	decfsz	f, d	Μείωση κατά ένα του καταχωρητή f και παράκαμψη της επόμενης εντολής αν το αποτέλεσμα είναι μηδέν	1 ή 2	Z
8	incf	f, d	Αύξηση κατά ένα του καταχωρητή f	1	Z
9	incfsz	f, d	Αύξηση κατά ένα του καταχωρητή f και παράκαμψη της επόμενης εντολής αν το αποτέλεσμα είναι μηδέν	1 ή 2	
10	iorwf	f, d	Λογικό OR ανάμεσα στους καταχωρητές f και w	1	Z
11	movf	f, d	Ανάγνωση του καταχωρητή f	1	
12	movwf	f	Αποθήκευση του καταχωρητή f στον w	1	
13	nop		Καμία λειτουργία	1	
14	rlf	f, d	Περιστροφή προς τα αριστερά μέσω της σημαίας κρατουμένου του καταχωρητή f	1	C
15	rrf	f, d	Περιστροφή προς τα δεξιά μέσω της σημαίας κρατουμένου του καταχωρητή f	1	C
16	subwf	f, d	Αφαίρεση του w από τον καταχωρητή f	1	C, DC, Z
17	swapf	f, d	Αμοιβαία αλλαγή θέσης των τεσσάρων λιγότερων σημαντικών bit με τα τέσσερα περισσότερα σημαντικά bit του καταχωρητή f	1	
18	xorwf	f, d	Λογικό XOR ανάμεσα στους καταχωρητές w και f	1	Z
19	bcf	f, b	Μηδενισμός του bit θέσης b του καταχωρητή f	1	
20	bsf	f, b	Ανάθεση στο λογικό '1' του bit θέσης b του καταχωρητή f	1	
21	btfsf	f, b	Έλεγχος του bit θέσης b του καταχωρητή f και παράκαμψη της επόμενης εντολής αν είναι στο λογικό '0'	1 ή 2	
22	btssf	f, b	Έλεγχος του bit θέσης b του καταχωρητή f και παράκαμψη της επόμενης εντολής αν είναι στο λογικό '1'	1 ή 2	
23	addlw	k	Πρόσθεση ανάμεσα στον καταχωρητή w και το αριθμητικό k	1	C, DC, Z
24	andlw	k	Λογικό AND ανάμεσα στον καταχωρητή w και το αριθμητικό k	1	Z
25	call	k	Κλήση υπορουτίνας στη διεύθυνση που δείχνει το αριθμητικό k	2	
26	clrwdt		Μηδενισμός του Watchdog Timer	1	$\overline{TO}, \overline{PD}$
27	goto	k	Άλμα του προγράμματος στη θέση μνήμης που δείχνει το αριθμητικό k	2	
28	iorlw	k	Λογικό OR ανάμεσα στον καταχωρητή w και το αριθμητικό k	1	Z
29	movlw	l	Φόρτωση του αριθμητικού k στον καταχωρητή w	1	
30	retfie		Επιστροφή από διακοπή	2	
31	retlw	k	Επιστροφή από κλήση σε υπορουτίνα φορτώνοντας τον καταχωρητή w με το αριθμητικό k	2	
32	return		Επιστροφή από κλήση σε υπορουτίνα	2	
33	sleep		Είσοδος σε λειτουργία χαμηλής κατανάλωσης ισχύος	1	$\overline{TO}, \overline{PD}$
34	sublw	k	Αφαίρεση του καταχωρητή w από το αριθμητικό k	1	C, DC, Z
35	xorlw	k	Λογικό XOR ανάμεσα στον καταχωρητή w και το αριθμητικό k	1	Z
	d=0 ή w		Το αποτέλεσμα αποθηκεύεται στον w		
	d=1 ή f		Το αποτέλεσμα αποθηκεύεται στον f		

Πίνακα 2. Το σετ εντολών του μικροελεγκτή PIC16F887.

3. Το διάγραμμα ακροδεκτών του μικροελεγκτή PIC16F887

Ο μικροελεγκτής PIC16F887 περιλαμβάνει πέντε θύρες εισόδου/εξόδου. Οι θύρες αυτές είναι:

- Θύρα PORTA (οκτώ ακροδέκτες)
- Θύρα PORTB (οκτώ ακροδέκτες)
- Θύρα PORTC (οκτώ ακροδέκτες)
- Θύρα PORTD (οκτώ ακροδέκτες)
- Θύρα PORTE (τέσσερις ακροδέκτες)



Εικόνα 1. Το διάγραμμα ακροδεκτών του PIC16F887.

Το διάγραμμα ακροδεκτών του μικροελεγκτή σε συσκευασία DIP (Dual-In line-Parallel) παρουσιάζεται στην εικόνα 1. Κάθε ακροδέκτης μπορεί να πολυπλέκεται με κάποια είσοδο ή έξοδο των διάφορων ενσωματωμένων περιφερειακών κυκλωμάτων.

5. Ο καταχωρητής STATUS

Ένας από τους σπουδαιότερους καταχωρητές ειδικής χρήσης του PIC16F887 είναι ο STATUS. Ο καταχωρητής STATUS περιέχει τα flag bit (σημείες) που προκύπτουν από τις πράξεις της αριθμητικής λογικής μονάδας (ALU) του επεξεργαστή, τα Bank bit τα οποία δείχνουν ποια περιοχή της μνήμης (Bank) διευθυνσιοδοτείται καθώς και κάποια bit που παρέχουν ενδείξεις σχετικά με την κατάσταση στην οποία βρίσκεται ο μικροελεγκτής. Αναλυτικότερα ο καταχωρητής STATUS αναλύεται στον πίνακα που ακολουθεί.

Καταχωρητής STATUS

bit 7	IRP: Bit επιλογής περιοχής μνήμης (Bank) για την έμμεση διευθυνσιοδότηση 1= Bank2, Bank3 0= Bank0, Bank1
bit 6-5	RP1:RP0: Bit επιλογής περιοχής μνήμης (Bank) για την άμεση διευθυνσιοδότηση 11= Bank3 10= Bank2

	01= Bank1
	00= Bank0
bit 4	TO: Bit ένδειξης υπερχείλισης WDT 1= Ένδειξη εφαρμογής τροφοδοσίας, εκτέλεσης εντολής clrwdt ή sleep 0= Υπερχείλιση Watchdog Timer
bit 3	PD: Bit ένδειξης κατάστασης χαμηλής κατανάλωσης ισχύος 1= Μετά από εφαρμογή τροφοδοσίας ή από εκτέλεση εντολής clrwdt 0= Ένδειξη εκτέλεσης εντολής sleep
bit 2	Z: Σημαία μηδενισμού 1= Το αποτέλεσμα μιας αριθμητικής ή λογικής λειτουργίας είναι μηδέν 0= Το αποτέλεσμα μιας αριθμητικής ή λογικής λειτουργίας δεν είναι μηδέν
bit 1	DC: Κρατούμενο ή δανεικό κρατούμενο από τα 4 bits χαμηλής τάξης 1= Υπάρχει κρατούμενο από τα 4 bits χαμηλής τάξης 0= Δεν υπάρχει κρατούμενο από τα 4 bits χαμηλής τάξης
bit 0	C: Κρατούμενο 1= Κρατούμενο που προκύπτει από το bit υψηλότερης τάξης 0= Δεν προκύπτει κρατούμενο
	Σημ.: Στην αφαίρεση ισχύει ακριβώς το αντίθετο. Όταν υπάρχει δανεικό κρατούμενο τότε C=0 ενώ όταν δεν υπάρχει τότε C=1

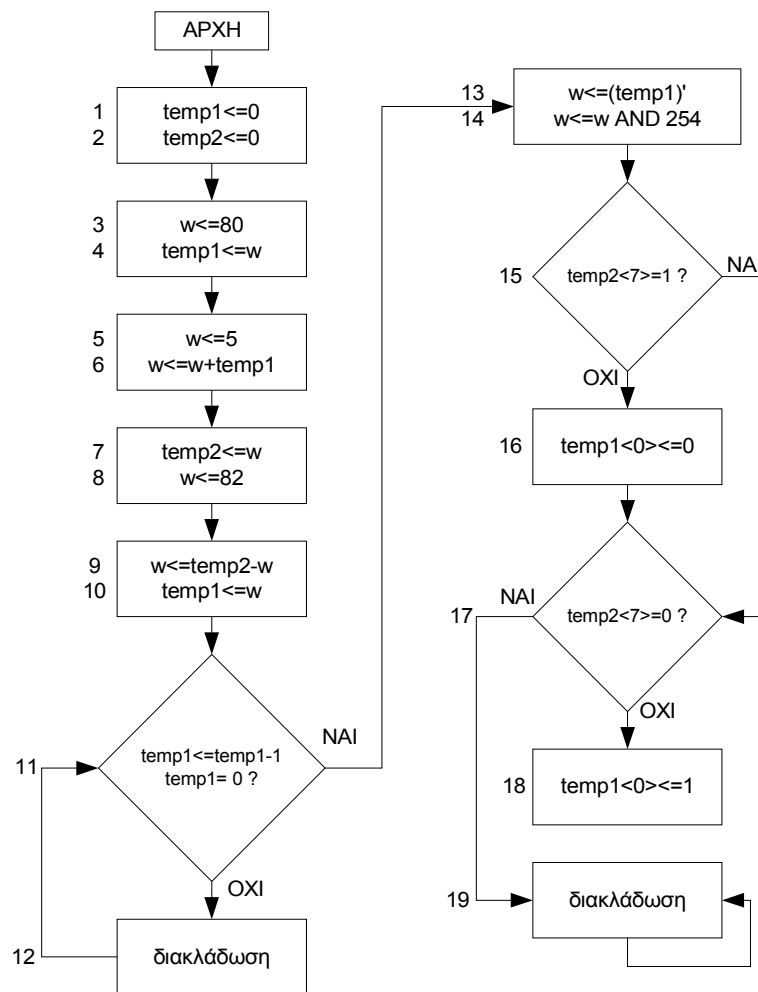
Πίνακας 3. Ο καταχωρητής STATUS.

ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΙΣ ΕΝΤΟΛΕΣ ΤΟΥ PIC16F887

1. Θεωρία

Στην πρώτη άσκηση το αντικείμενο είναι η εξοικείωση με τις βασικότερες εντολές του μικροελεγκτή PIC16F887. Για το λόγο αυτό το πρόγραμμα που θα φορτώσουμε στον μικροελεγκτή δεν πραγματοποιεί κάποια συγκεκριμένη λειτουργία αλλά εκτελεί ορισμένες πράξεις με στόχο την κατανόηση της κάθε εντολής.

Το λογικό διάγραμμα του προγράμματος που θα χρησιμοποιηθεί παρουσιάζεται στην εικόνα 1.



Εικόνα 1. Λογικό διάγραμμα.

Στον πίνακα 1 παρουσιάζεται η αναλυτική περιγραφή για κάθε σημείο του λογικού διαγράμματος.

A/A

ΛΕΙΤΟΥΡΓΙΑ

- 1 Μηδενισμός του καταχωρητή temp1
- 2 Μηδενισμός του καταχωρητή temp2
- 3 Φόρτωση στον καταχωρητή w του αριθμού 80
- 4 Αποθήκευση των περιεχομένων του καταχωρητή w στον καταχωρητή temp1
- 5 Φόρτωση στον καταχωρητή w του αριθμού 5
- 6 Πρόσθεση ανάμεσα στα περιεχόμενα των καταχωρητών w και temp1 και αποθήκευση του αποτελέσματος στον καταχωρητή w
- 7 Αποθήκευση των περιεχομένων του καταχωρητή w στον καταχωρητή temp2
- 8 Φόρτωση στον καταχωρητή w του αριθμού 82
- 9 Αφαίρεση από τα περιεχόμενα του καταχωρητή temp2 τα περιεχόμενα του καταχωρητή w και αποθήκευση του αποτελέσματος στον καταχωρητή w
- 10 Αποθήκευση των περιεχομένων του καταχωρητή w στον καταχωρητή temp1
- 11 Μείωση κατά ένα του καταχωρητή temp1, αποθήκευση του αποτελέσματος στον καταχωρητή temp1 και παράκαμψη της επόμενης εντολής αν το αποτέλεσμα είναι μηδέν
- 12 Διακλάδωση στην προηγούμενη εντολή
- 13 Υπολογισμός συμπληρώματος ως προς ένα του καταχωρητή temp1 και αποθήκευση του αποτελέσματος στον καταχωρητή w
- 14 Λογικό AND ανάμεσα στα bit του καταχωρητή w και στα bit του αριθμού 254 και αποθήκευση του αποτελέσματος στον καταχωρητή w
- 15 Έλεγχος του bit 7 του καταχωρητή temp2 και παράκαμψη της επόμενης εντολής αν είναι στο λογικό '1'
- 16 Μηδενισμός του bit 0 του καταχωρητή temp1
- 17 Έλεγχος του bit 7 του καταχωρητή temp2 και παράκαμψη της επόμενης εντολής αν είναι στο λογικό '0'
- 18 Ανάθεση στο λογικό '1' του bit 0 του καταχωρητή temp1
- 19 Διακλάδωση στην ίδια εντολή

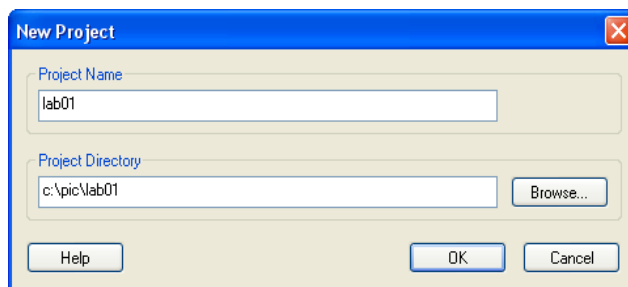
Πίνακας 1. Περιγραφή λογικού διαγράμματος.

2. Προσομοίωση

α) Να εκκινήσετε το λογισμικό ανάπτυξης εφαρμογών MPLAB IDE.

β) Με την επιλογή *Project>New...* να δημιουργήσετε μια νέα εφαρμογή (project file).

γ) Στο παράθυρο που θα εμφανιστεί (εικόνα 2) να δώσετε ένα όνομα για την εφαρμογή καθώς και για τον κατάλογο στον οποίο θα αποθηκευτούν τα αρχεία της εφαρμογής. Στη συνέχεια πατήστε το κουμπί «OK».



Εικόνα 2. Η επιλογή Project>New...

δ) Με την επιλογή Configure>Select Device... να εμφανίσετε το παράθυρο της εικόνας 3 όπου θα πρέπει να καθοριστεί ποιος μικροελεγκτής θα χρησιμοποιηθεί στην τρέχουσα εφαρμογή. Εδώ επιλέξτε τον μικροελεγκτή PIC16F887 και πατήστε το κουμπί «OK».

ε) Να δημιουργήσετε ένα αρχείο πηγαίου κώδικα πηγαίνοντας στην επιλογή File>New

στ) Να αντιγράψετε στον κειμενογράφο τον κώδικα που ακολουθεί:

```
include <p16f887.inc> ; οδηγία στον assembler να χρησιμοποιήσει το αρχείο
                      ; στο οποίο δηλώνονται τα συμβολικά ονόματα
                      ; καταχωρητών ειδικής χρήσης

temp1 equ h'20'      ; Δήλωση συμβολικών ονομάτων
temp2 equ h'21'      ; καταχωρητών γενικής χρήσης

org h'0000'          ; Οδηγία προς τον assembler να τοποθετήσει το
                      ; πρόγραμμα από τη θέση μνήμης h'0000' και μετά

nop                  ; Καμία λειτουργία (no operation)

clrf temp1           ; Μηδενισμός του καταχωρητή temp1

clrf temp2           ; Μηδενισμός του καταχωρητή temp2

movlw h'50'          ; Φόρτωση του αριθμού h'50'=d'80' στον καταχωρητή w

movwf temp1          ; Αποθήκευση των περιεχομένων του w στον καταχωρητή temp1

movlw h'05'          ; Φόρτωση του αριθμού h'05'=d'05' στον καταχωρητή w

addwf temp1,w        ; Πρόσθεση ανάμεσα στους καταχωρητές w και temp1
                      ; και αποθήκευση του αποτελέσματος στον w

movwf temp2          ; Αποθήκευση των περιεχομένων του w στον καταχωρητή temp2

movlw h'52'          ; Φόρτωση του αριθμού h'52'=d'82' στον καταχωρητή w
```

```
subwf temp2,w      ; Αφαίρεση του καταχωρητή w από τον καταχωρητή temp2 και
                   ; αποθήκευση του αποτελέσματος στον w

movwf temp1        ; Αποθήκευση των περιεχομένων του w στον καταχωρητή temp1

LABEL1             ; Καθορισμός ετικέτας μνήμης
decfsz temp1,f     ; Μείωση κατά ένα του καταχωρητή temp1 και παράκαμψη της
                   ; επόμενης εντολής εάν το αποτέλεσμα είναι μηδέν

goto LABEL1        ; Άλμα στη διεύθυνση μνήμης με ετικέτα LABEL1

comf temp1,w       ; Συμπλήρωμα ως προς '1' του καταχωρητή temp1
                   ; και αποθήκευση του αποτελέσματος στον w

andlw b'11111110'  ; AND ανάμεσα στον w και στον αριθμό b'11111110'=d'254'

rlf temp1,f        ; Ολίσθηση προς τα αριστερά του καταχωρητή temp1

btfss temp2,7      ; Έλεγχος του bit 7 του καταχωρητή temp2 και
                   ; παράκαμψη της επόμενης εντολής αν είναι '1'

bcf temp1,0        ; Μηδενισμός του bit 0 του καταχωρητή temp1

btfsc temp2,7      ; Έλεγχος του bit 7 του καταχωρητή temp2 και
                   ; παράκαμψη της επόμενης εντολής αν είναι '0'

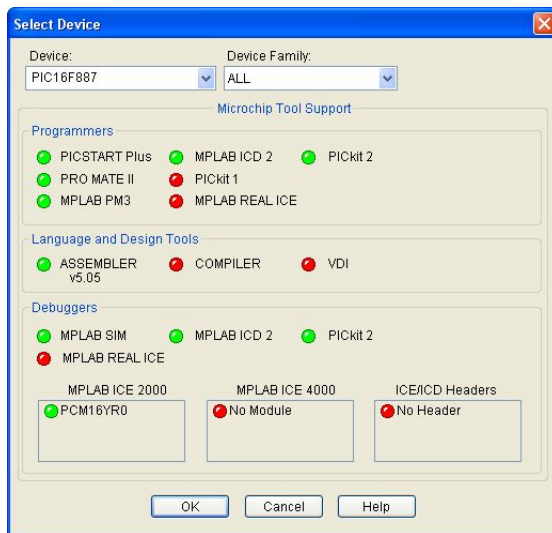
bsf temp1,0        ; Ανάθεση στο λογικό '1' του bit 0 του καταχωρητή temp1

goto $             ; Άλμα στην ίδια εντολή

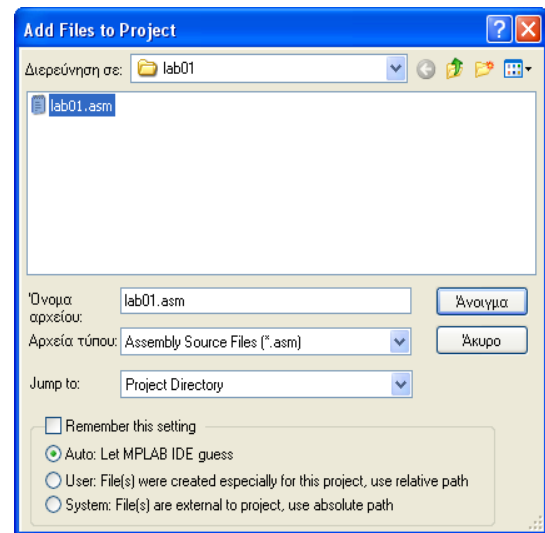
end
```

ζ) Με την επιλογή File>Save As... αποθηκεύστε το αρχείο πηγαίου κώδικα με την επέκταση .asm στον κατάλογο που ορίσατε για την εφαρμογή.

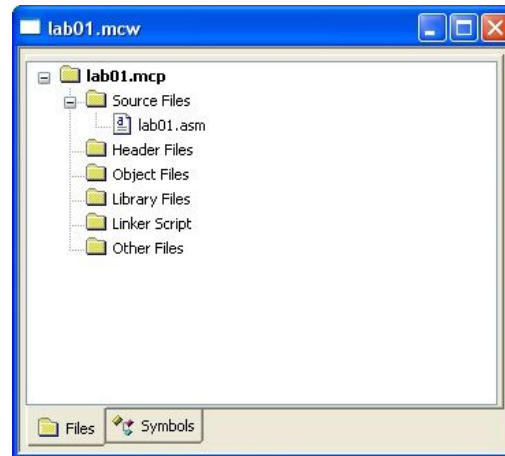
η) Προσθέστε το αρχείο πηγαίου κώδικα στην εφαρμογή πηγαίνοντας στην επιλογή Project>Add Files to Project.... Θα εμφανιστεί το παράθυρο της εικόνας 4. Επιλέξτε το αρχείο πηγαίου κώδικα που μόλις αποθηκεύσατε και πατήστε το κουμπί «Άνοιγμα». Μόλις το αρχείο προστεθεί θα πρέπει να εμφανίζεται στον «χώρο εργασίας» της εφαρμογής (workspace) όπως φαίνεται στην εικόνα 5. Αν ο «χώρος εργασίας» δεν είναι ορατός τότε εμφανίστε το παράθυρο με την επιλογή View>Project.



Εικόνα 3. Επιλογή μικροελεγκτή.

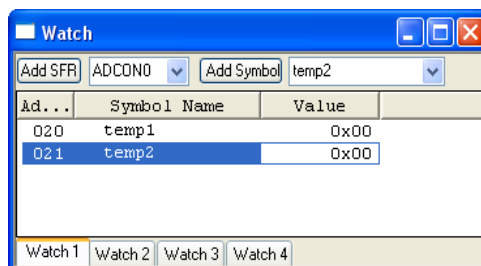


Εικόνα 4. Προσθήκη αρχείου πηγαίου κώδικα.



Εικόνα 5. Ο «χώρος εργασίας» (Workspace).

ι) Να μεταγλωττίσετε το πρόγραμμα με την επιλογή *Project>Build All* και διορθώστε, εφόσον υπάρχουν, τα συντακτικά λάθη. Να εμφανίσετε το παράθυρο καταχωρητών ειδικής χρήσης και το παράθυρο παρακολούθησης μεταβλητών (watch window) με τις επιλογές *View>Special Function Registers* και *View/Watch* αντίστοιχα. Στο παράθυρο Watch εντοπίστε τα συμβολικά ονόματα των καταχωρητών temp1 και temp2 στη λίστα και πατήστε το κουμπί «Add Symbol» ώστε να τους προσθέσετε μέσα στο παράθυρο όπως δείχνει η εικόνα 6.



Εικόνα 6. Το παράθυρο παρακολούθησης μεταβλητών (Watch Window).

ια) Στο σημείο αυτό όλα είναι έτοιμα να προσομοιώσουμε στον simulator που διαθέτει το MPLAB IDE, το πρόγραμμα που έχουμε γράψει για το μικροελεγκτή.

Να ενεργοποιήσετε τον simulator με την επιλογή Debugger>Select Tool>MPLAB SIM.

ιβ) Να εκτελέσετε εντολή προς εντολή το πρόγραμμα με την επιλογή Debugger>Step Into ή με το βοηθητικό πλήκτρο συντόμευσης F7. Για να εκτελέσετε το πρόγραμμα από την αρχή πραγματοποιήστε RESET στο μικροελεγκτή με την επιλογή Debugger>Reset>Processor Reset ή με το βοηθητικό πλήκτρο συντόμευσης F6.

Παρατηρήστε τον τρόπο με τον οποίο μεταβάλλονται οι τιμές των καταχωρητών κατά την εκτέλεση κάθε εντολής.

Να συμπληρώσετε στον παρακάτω πίνακα ποιος καταχωρητής μεταβάλλεται μετά από κάθε εντολή και ποια η τιμή του μετά την εκτέλεση κάθε εντολής.

A/A	ΕΝΤΟΛΗ	W	STATUS	temp1	temp2
1	nop				
2	clrf temp1				
3	clrf temp2				
4	movlw h'50'				
5	movwf temp1				
6	movlw h'05'				
7	addwf temp1,w				
8	movwf temp2				
9	movlw h'52'				
10	subwf temp2,w				
11	movwf temp1				
12	decfsz temp1,f				
13	goto LABEL1				
14	decfsz temp1,f				
15	goto LABEL1				
16	decfsz temp1,f				
17	comf temp1,w				
18	andlw b'11111110'				
19	btfss temp2,7				
20	bcf temp1,0				
21	btfsc temp2,7				
22	bsf temp1,0				
23	goto \$				

Πίνακας 2

3. Εργασίες

α) Γράψτε ένα πρόγραμμα το οποίο πραγματοποιεί πρόσθεση ανάμεσα σε δύο 16-bit αριθμούς.

β) Γράψτε ένα πρόγραμμα το οποίο πραγματοποιεί αφαίρεση ανάμεσα σε δύο 16-bit αριθμούς.

Υπόδειξη: Να χρησιμοποιηθούν δύο 8-bit καταχωρητές για την αποθήκευση κάθε 16-bit αριθμού. Ο ένας καταχωρητής θα περιέχει το byte υψηλής τάξης και ο άλλος το byte χαμηλής τάξης του αριθμού. Το αποτέλεσμα κάθε πράξης να αποθηκεύεται σε δύο νέους καταχωρητές.

4. Προχωρημένα θέματα

α) Να γράψετε ένα πρόγραμμα το οποίο μετατρέπει έναν μονοψήφιο δεκαδικό αριθμό στον αντίστοιχο αριθμό του κώδικα ASCII και το αντίστροφο.

β) Γράψτε ένα πρόγραμμα το οποίο πραγματοποιεί πολλαπλασιασμό ανάμεσα σε δύο 8-bit αριθμούς.

γ) Να υλοποιήσετε μια ρουτίνα που πραγματοποιεί ακέραια διαίρεση με διαιρετέο εύρους 16-bit και διαιρέτη 8-bit.

δ) Να γράψετε ένα πρόγραμμα το οποίο μετατρέπει έναν διψήφιο δεκαδικό αριθμό (αποθηκευμένο σε δυαδική μορφή) σε μορφή BCD.

ε) Να γράψετε ένα πρόγραμμα που να υπολογίζει το ακέραιο μέρος της τετραγωνικής ρίζας ενός 16-bit αριθμού.

ΔΙΑΧΕΙΡΙΣΗ ΨΗΦΙΑΚΩΝ ΕΙΣΟΔΩΝ/ΕΞΟΔΩΝ (I/O)

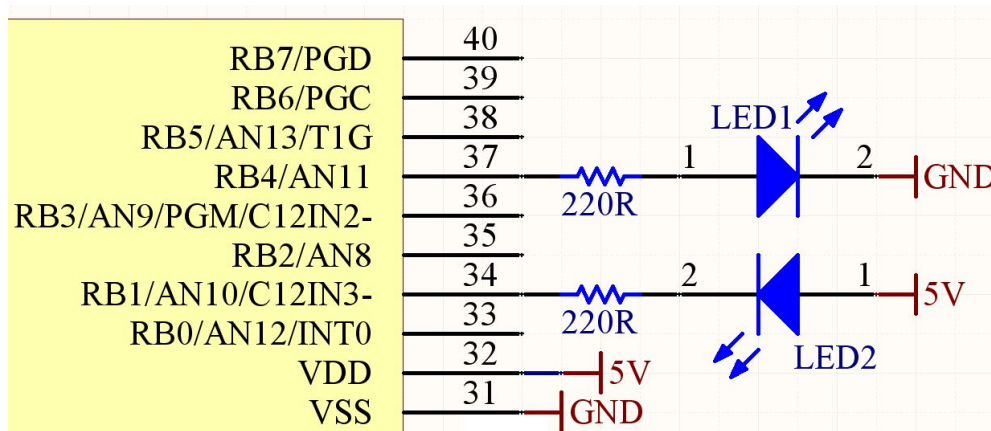
1. Θεωρία

Στην παρούσα άσκηση θα πραγματοποιηθεί η πρώτη εργαστηριακή εφαρμογή με τον PIC16F887. Σκοπός της άσκησης αυτής είναι η εξοικείωση με τη διαδικασία προγραμματισμού του μικροελεγκτή καθώς και με τη διαχείριση των ψηφιακών εισόδων/εξόδων.

Ο PIC16F887 διαθέτει έξι θύρες εισόδου/εξόδου. Οι θύρες PORTA, PORTB, PORTC και PORTD περιλαμβάνουν από οκτώ ακροδέκτες η κάθε μία ενώ η θύρα PORTE περιλαμβάνει τέσσερις. Ο προγραμματιστής μπορεί να διαχειριστεί την κάθε θύρα με τη χρήση δύο καταχωρητών εύρους τόσων bit όσος είναι ο αριθμός των ακροδεκτών που διαθέτει. Η θύρα PORTA διαχειρίζεται από τους καταχωρητές TRISA και PORTA οι οποίοι είναι εύρους 8 bit. Η θύρα PORTB διαχειρίζεται από τους καταχωρητές PORTB και TRISB εύρους επίσης 8 bit, η θύρα PORTE από τους καταχωρητές TRISE και PORTE που είναι εύρους τεσσάρων bit κ.ο.κ.. Κάθε bit των καταχωρητών TRISx, PORTx (x=A, B, C, D, E) αντιστοιχεί σε έναν ακροδέκτη του PIC16F887. Το χαμηλότερης τάξης bit του καταχωρητή TRISA (bit 0 ή αλλιώς TRISA<0>) αντιστοιχεί στο RA0 (ακροδέκτης 2 του PIC16F887 συσκευασίας DIP), το αμέσως πιο σημαντικό bit (bit 1 ή αλλιώς TRISA<1>) αντιστοιχεί στο RA1 (ακροδέκτης 3 του PIC16F887 συσκευασίας DIP) κλπ. Αν κάποιο bit του καταχωρητή TRISx βρίσκεται στο λογικό '1' τότε ο ακροδέκτης στον οποίο αντιστοιχεί αποτελεί ψηφιακή είσοδο για τον μικροελεγκτή ενώ αν βρίσκεται στο λογικό '0' τότε αποτελεί ψηφιακή έξοδο. Στην περίπτωση που κάποιο bit του καταχωρητή TRISx είναι στο λογικό '0' (έξοδος) τότε η κατάσταση στην έξοδο για τον ακροδέκτη που αντιστοιχεί, καθορίζεται από το αντίστοιχο bit του καταχωρητή PORTx. Αν το bit του PORTx είναι στο λογικό '0' τότε στην έξοδο θα έχουμε χαμηλό επίπεδο τάσης (συνήθως 0V) ενώ αν το bit αυτό είναι στο λογικό '1' τότε θα έχουμε υψηλό επίπεδο τάσης (συνήθως 5V).

Οι έξοδοι του μικροελεγκτή μπορούν να χρησιμοποιηθούν για να δώσουμε εντολές εκκίνησης/παύσης, για να πραγματοποιήσουμε επικοινωνία με άλλες συσκευές, για να δημιουργήσουμε σήματα ελέγχου κλπ. Για να παρατηρήσουμε οπτικά την κατάσταση ενός ακροδέκτη εξόδου μπορούμε να συνδέσουμε μια δίοδο LED. Μια δίοδος LED όταν διαρρέεται από ρεύμα ακτινοβολεί φως. Μια τυπική τιμή έντασης ρεύματος για

να εκπέμπει φως μια δίοδος LED είναι 10mA. Οι δύο πιθανές συνδεσμολογίες παρουσιάζονται στην εικόνα 1.



Εικόνα 1. Συνδεσμολογία διόδων LED σε ακροδέκτες του PIC16F887.

Η LED1 άγει όταν ο ακροδέκτης RB4 βρίσκεται στο λογικό '1'. Η LED2 άγει όταν ο ακροδέκτης RB1 βρίσκεται στο λογικό '0'. Ιδανικά η ένταση του ρεύματος που θα διαρρέει τις LED σε κάθε περίπτωση θα είναι:

$$I_{LED} = \frac{5V - V_{f,LED}}{R} = \frac{5V - 1.2V}{220\Omega} \cong 17mA$$

, όπου $V_{f,LED}$ η πρόσω τάση πόλωσης της διόδου LED.

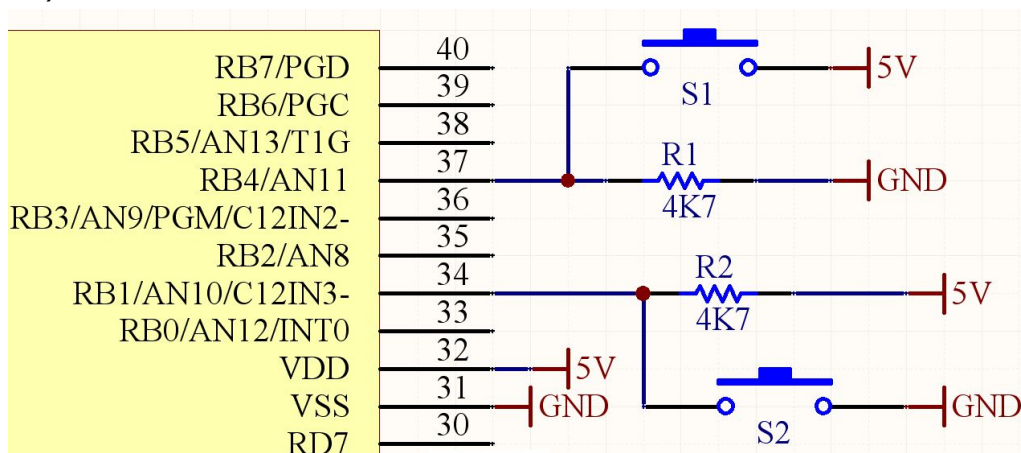
Πρακτικά το ρεύμα θα είναι λίγο μικρότερο καθώς δεν συνυπολογίζεται η εσωτερική αντίσταση εξόδου του μικροελεγκτή. Προσοχή πρέπει να δοθεί κατά το σχεδιασμό του κυκλώματος ώστε το ρεύμα που παρέχεται ή επάγεται να ΜΗΝ υπερβαίνει ποτέ τα 25mA καθώς στην συγκεκριμένη τιμή καθορίζεται η ικανότητα οδήγησης κάθε ακροδέκτη από τον κατασκευαστή του μικροελεγκτή. Μια τυπική τιμή αντίστασης σε σειρά με τη δίοδο LED είναι 220Ω.

Για να δώσουμε εξωτερικά κάποιες εντολές στο μικροελεγκτή θα πρέπει να χρησιμοποιήσουμε ορισμένους ακροδέκτες ως ψηφιακές εισόδους. Μια εντολή στον PIC μπορεί να δοθεί με τη χρησιμοποίηση ενός διακόπτη τύπου push-button. Οι δύο πιθανές συνδεσμολογίες παρουσιάζονται στην εικόνα 2.

Στην περίπτωση του διακόπτη S1, λόγω της αντίστασης R1 που είναι συνδεδεμένη στη γείωση (pull-down resistor), στην είσοδο RB4 θα έχουμε λογικό '0'. Όταν πατηθεί ο διακόπτης push-button S1 τότε στον ακροδέκτη RB4 εφαρμόζονται 5V άρα μια ανάγνωση θα δίνει λογικό '1'.

Στην περίπτωση του διακόπτη S2 παρουσιάζεται ακριβώς αντίστροφη λογική. Ο ακροδέκτης RB1 θα βρίσκεται σε λογικό '1' λόγω της αντίστασης R2 που είναι συνδεδεμένη στην τροφοδοσία (pull-up resistor). Όταν πατηθεί ο διακόπτης push-button S2 τότε στον ακροδέκτη RB1 εφαρμόζεται το σήμα της γείωσης οπότε μια ανάγνωση θα δίνει λογικό '0'.

Από τις παραπάνω περιπτώσεις πιο συνηθισμένη είναι η συνδεσμολογία με pull-up αντιστάσεις.



Εικόνα 2. Συνδεσμολογία ψηφιακών εισόδων.

Ο προγραμματισμός του μικροελεγκτή μπορεί να επιτευχθεί με διάφορα αναπτυξιακά εργαλεία. Για τον προγραμματισμό του PIC16F887 χρησιμοποιούνται οι ακροδέκτες 40 (RB7), 39 (RB6), 11 και 32 (VDD), 12 και 31 (VSS) καθώς και ο ακροδέκτης 1 (MCLR). Σημασία πρέπει επίσης να δοθεί στα bit ρυθμίσεων (Configuration Bits). Με τα bit αυτά καθορίζονται ειδικές λειτουργίες του μικροελεγκτή.

α) Oscillator: Καθορισμός ταλαντωτή χρονισμού του μικροελεγκτή.

- External RC Clockout. Επιλογή ταλαντωτή χρονισμού βασισμένου σε εξωτερικό δίκτυο RC. Το ρολόι είναι διαθέσιμο εξωτερικά στον ακροδέκτη 14 του PIC16F887
- External RC No Clockout. Επιλογή ταλαντωτή χρονισμού βασισμένου σε εξωτερικό δίκτυο RC. Το ρολόι είναι δεν διαθέσιμο εξωτερικά του μικροελεγκτή
- Internal RC Clockout. Επιλογή ταλαντωτή χρονισμού βασισμένου στον εσωτερικό ταλαντωτή του μικροελεγκτή. Το ρολόι είναι διαθέσιμο εξωτερικά στον ακροδέκτη 14 του PIC16F887
- Internal RC No Clockout. Επιλογή ταλαντωτή χρονισμού βασισμένου στον εσωτερικό ταλαντωτή του μικροελεγκτή. Το ρολόι είναι δεν διαθέσιμο εξωτερικά στο μικροελεγκτή
- EC. Επιλογή ταλαντωτή χρονισμού από εξωτερικό ρολόι (External Clock)
- HS. Επιλογή κρυσταλικού ταλαντωτή υψηλής ταχύτητας (High Speed) $\geq 8\text{MHz}$

- XT. Επιλογή κρυσταλικού ταλαντωτή συχνότητας < 8MHz
- LP. Επιλογή κεραμικού ταλαντωτή χαμηλής ισχύος (Low Power)

β) WatchDog Timer: Ο WatchDog timer αποτελεί ένα κύκλωμα χρονισμού επιτήρησης. Σε κάθε υπερχείλιση προκαλεί RESET και χρησιμοποιείται για αποφυγή «κολλημάτων» του προγράμματος του μικροελεγκτή.

- On. Ενεργοποίηση
- Off. Απενεργοποίηση

γ) Power Up Timer: Ο Power Up Timer αποτελεί ένα κύκλωμα που διατηρεί σε RESET το μικροελεγκτή για μερικά msec κατά την εφαρμογή της τροφοδοσίας μέχρι να σταθεροποιηθεί.

- On. Ενεργοποίηση
- Off. Απενεργοποίηση

δ) Master Clear Enable: Καθορισμός του ακροδέκτη 1 του PIC16F887 (MCLR) ως Master Clear ή ως ψηφιακή είσοδος.

- RE3 is digital input. Ο ακροδέκτης 1 του PIC16F887 αποτελεί ψηφιακή είσοδο
- /Master Clear is external. Ο ακροδέκτης 1 του PIC16F887 αποτελεί το Master Clear.

ε) Code Protect: Με το bit αυτό καθορίζεται αν θα υπάρχει προστασία της μνήμης προγράμματος από ανάγνωση.

- Off. Δεν υπάρχει προστασία από ανάγνωση
- On. Υπάρχει προστασία από ανάγνωση

στ) Data EE Read Protect: Με το bit αυτό καθορίζεται αν θα υπάρχει προστασία της μνήμης δεδομένων EEPROM από ανάγνωση.

- Off. Δεν υπάρχει προστασία από ανάγνωση
- On. Υπάρχει προστασία από ανάγνωση

ζ) Brown Out Detect: Με το χαρακτηριστικό Brown out Detect προκαλείται RESET στο μικροελεγκτή όταν η τροφοδοσία πέσει κάτω από κάποια τάση αναφοράς. Με την παραπάνω επιλογή καθορίζεται η ενεργοποίηση/απενεργοποίηση αυτού του χαρακτηριστικού του PIC16F887

- BOD and SBOREN disabled. Απενεργοποίηση
- SBOREN controls BOR function. Το bit SBOREN του καταχωρητή PCON ελέγχει την ενεργοποίηση/απενεργοποίηση
- BOD enabled in run, disabled in sleep, SBOREN disabled. Εδώ το χαρακτηριστικό αυτό είναι ενεργοποιημένο κατά την κανονική λειτουργία και απενεργοποιημένο κατά τη λειτουργία χαμηλής κατανάλωσης ισχύος
- BOD enabled, SBOREN disabled. Ενεργοποίηση

η) Internal External Switch Over Mode: Με το bit αυτό ο ενεργοποιείται/απενεργοποιείται η αλλαγή του ταλαντωτή χρονισμού από τον εσωτερικό σε εξωτερικό και το αντίστροφο.

- Enabled. Ενεργοποίηση
- Disabled. Απενεργοποίηση

θ) Monitor Clock Fail-safe: Με το bit αυτό ενεργοποιείται/απενεργοποιείται το χαρακτηριστικό του μικροελεγκτή να μπορεί να μεταβεί στον εσωτερικό ταλαντωτή χρονισμού σε περίπτωση αποτυχίας του εξωτερικού ταλαντωτή.

- Enabled. Ενεργοποίηση
- Disabled. Απενεργοποίηση

ι) Low Voltage Program: Με το bit αυτό ενεργοποιείται/απενεργοποιείται η δυνατότητα για προγραμματισμό εντός του κυκλώματος με χαμηλή τάση. Όταν είναι ενεργοποιημένη η δυνατότητα αυτή τότε ο ακροδέκτης RB3/PGM παύει να είναι διαθέσιμος στον προγραμματιστή και χρησιμοποιείται μόνο για τον προγραμματισμό του μικροελεγκτή.

- Enabled. Ενεργοποίηση
- Disabled. Απενεργοποίηση

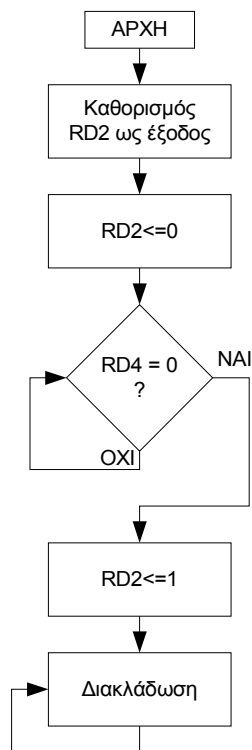
ια) Brown Out Reset Sel Bit: Με το bit αυτό επιλέγεται η τάση αναφοράς για το Brown-out Reset.

- Brown out at 2.1V. Θα πραγματοποιηθεί RESET όταν η τάση γίνει 2.1V.
- Brown out at 4.0V. Θα πραγματοποιηθεί RESET όταν η τάση γίνει 4.0V.

ιβ) Self Write Enable: Με την επιλογή αυτή καθορίζουμε αν η μνήμη προγράμματος θα έχει προστασία εγγραφής από το firmware το μικροελεγκτή.

- 0000-0fff prot. Προστασία εγγραφής στις διευθύνσεις μνήμης από h'0000' έως h'0FFF'.
- 0000-7fff prot. Προστασία εγγραφής στις διευθύνσεις μνήμης από h'0000' έως h'7FFF'.
- 0000-00ff prot. Προστασία εγγραφής στις διευθύνσεις μνήμης από h'0000' έως h'00FF'.
- No protection. Προστασία εγγραφής απενεργοποιημένη.

Στην παρούσα εργαστηριακή άσκηση θα γράψουμε ένα πρόγραμμα κατά το οποίο κατά την εκκίνηση θα διατηρεί μια δίοδο LED σβηστή. Η δίοδος LED θα παίρνει εντολή από το μικροελεγκτή να ανάψει μόλις πατηθεί ένας διακόπτης τύπου push-button. Η δίοδος συνδέεται στον ακροδέκτη RD2 ενώ ο διακόπτης στον ακροδέκτη RD4 μέσω μιας pull-up αντίστασης. Το λογικό διάγραμμα του προγράμματος παρουσιάζεται στην εικόνα 3.



Εικόνα 3. Το λογικό διάγραμμα.

2. Προσομοίωση

β) Δημιουργείστε μια νέα εφαρμογή στο MPLAB βασισμένη στον PIC16F887 και φορτώστε στον κειμενογράφο τον παρακάτω κώδικα:

```

include <p16f887.inc>

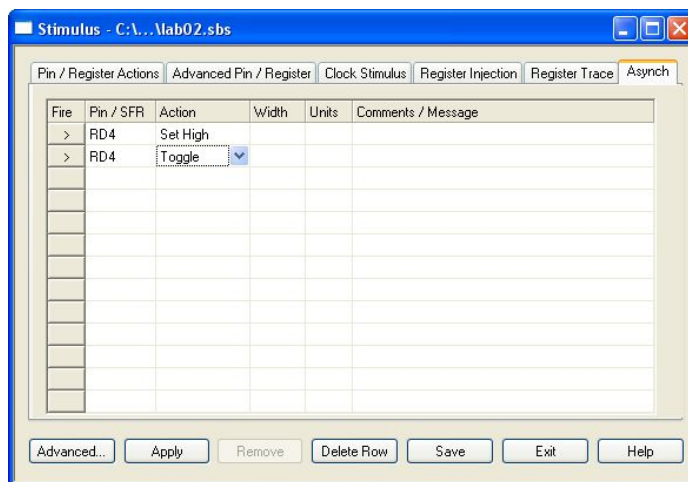
org h'0000'
nop

bsf STATUS,RP0 ; Διευθυνσιοδότηση της Bank1
bcf TRISD,2    ; Ρύθμιση του ακροδέκτη RD2 ως έξοδος
bcf STATUS,RP0 ; Διευθυνσιοδότηση της Bank0
bcf PORTD,2    ; Καθορισμός της εξόδου RD2 σε λογικό '0'
btfsc PORTD,4  ; Παράκαμψη της επόμενης εντολής αν το RD4 είναι στο '0'
goto $-1       ; Άλλα στην προηγούμενη εντολή
bsf PORTD,2    ; Η έξοδος RD2 οδηγείται στο λογικό '1'
goto $         ; Άλλα στην ίδια εντολή

end
    
```

γ) Μεταγλωττίστε το πρόγραμμα, διορθώστε αν υπάρχουν συντακτικά λάθη και ενεργοποιήστε τον προσομοιωτή (MPLAB SIM) του MPLAB.

δ) Για να προσομοιωθεί η λειτουργία του διακόπτη θα πρέπει να εφαρμοστούν λογικά σήματα εισόδου στον simulator. Αυτό επιτυγχάνεται με το παράθυρο διέγερσης του MPLAB. Για να εμφανίσετε το παράθυρο διέγερσης του MPLAB επιλέξτε Debugger>Stimulus>New Workbook. Στο παράθυρο αυτό επιλέξτε την κάρτα Asynch. Στην πρώτη γραμμή στη λίστα που βρίσκεται κάτω από τη στήλη Pin/SFR επιλέξτε τον ακροδέκτη RD4 ενώ κάτω από τη στήλη Action επιλέξτε την ενέργεια «Set High» (Καθορισμός στο λογικό '1'). Στη δεύτερη γραμμή στη λίστα που βρίσκεται κάτω από τη στήλη Pin/SFR επιλέξτε ξανά τον ακροδέκτη RD4 ενώ κάτω από τη στήλη Action επιλέξτε την ενέργεια «Toggle» (Αντιστροφή της λογικής κατάστασης). Στο σημείο αυτό το παράθυρο διέγερσης πρέπει να είναι όπως παρουσιάζεται στην εικόνα 4.



Εικόνα 4. Το παράθυρο διέγερσης.

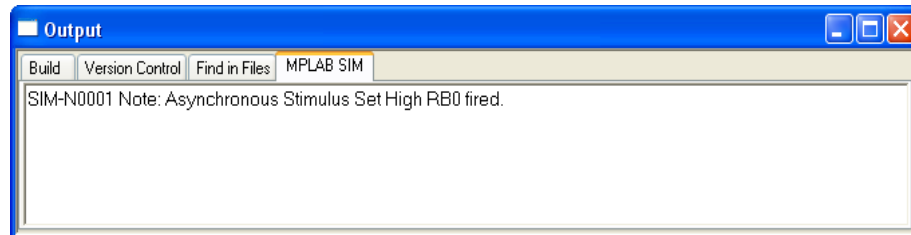
Κάθε φορά που πατάμε το κουμπί « > » (Fire) από το παράθυρο διέγερσης εκτελείται η ενέργεια που έχουμε προκαθορίσει στη στήλη Action.

ε) Πατήστε μία φορά το κουμπί « > » για να εκτελεστεί η ενέργεια «Set High». Στο παράθυρο εξόδου θα πρέπει να εμφανιστεί το μήνυμα που δείχνει η εικόνα 5. Αν το παράθυρο εξόδου δεν είναι ορατό τότε εμφανίστε το με την επιλογή View>Output.

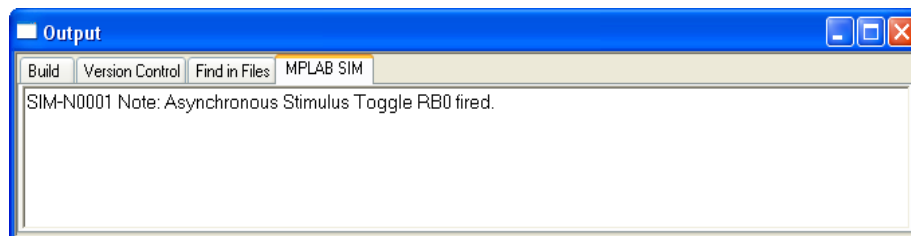
στ) Εκτελέστε εντολή προς εντολή το πρόγραμμα παρατηρώντας την τιμή που δείχνει ο ακροδέκτης εισόδου RD4. Σε ποιο σημείο κολλάει το πρόγραμμα; Ποια είναι η λογική κατάσταση του ακροδέκτη RD2;

ζ) Πατήστε μία φορά το κουμπί « > » για να εκτελεστεί η ενέργεια «Toggle». Το παράθυρο εξόδου θα πρέπει να έχει εμφανίσει το μήνυμα της εικόνας 6. Να συνεχίσετε την εκτέλεση του προγράμματος και να αναφέρετε σε ποια εντολή καταλήγει. Τι συνέβει

στον ακροδέκτη RD4 μετά την ενέργεια «Toggle»; Ποια είναι τώρα η λογική κατάσταση του ακροδέκτη RD2;



Εικόνα 5. Το παράθυρο εξόδου μετά την εφαρμογή λογικού '1' στο RD4 (Set High RD4).



Εικόνα 6. Το παράθυρο εξόδου μετά την εφαρμογή αντίστροφου σήματος στο RD4 (Toggle RD4).

3. Υλοποίηση

Για την υλοποίηση του κυκλώματος της εργαστηριακής άσκησης απαιτούνται τα παρακάτω υλικά:

IC1	=	PIC16F887
S1	=	διακόπτης τύπου push button
R1	=	4K7, $\geq 1/8W$
R2	=	220R, $\geq 1/8W$
LED1	=	Δίοδος LED
C1	=	15pF, $\geq 6V$
C2	=	15pF, $\geq 6V$
X1	=	Κρύσταλος 20MHz

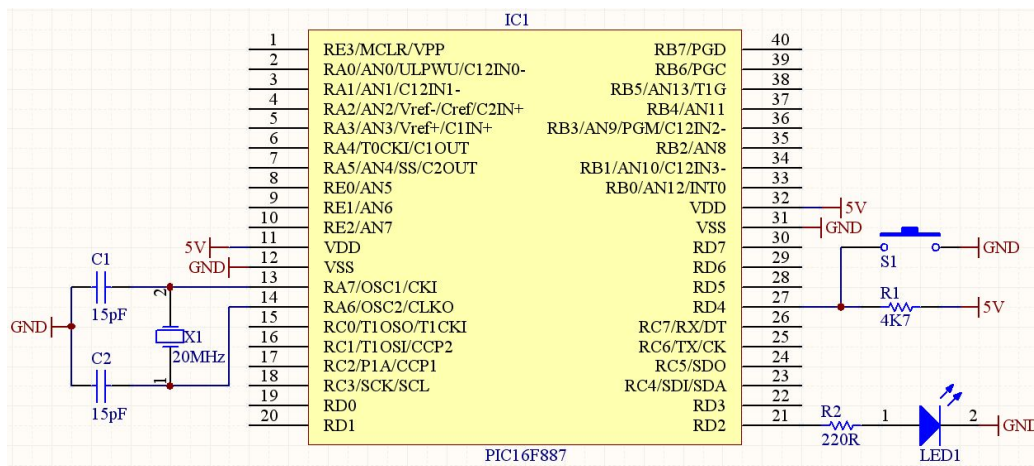
α) Υλοποιήστε σε πειραματική πλακέτα (breadboard) το κύκλωμα της εικόνας 7.

β) Με την επιλογή *Configure>Configuration Bits* πραγματοποιήστε τις ρυθμίσεις που παρουσιάζονται στο παράθυρο της εικόνας 8. Μεταγλωττίστε την εφαρμογή.

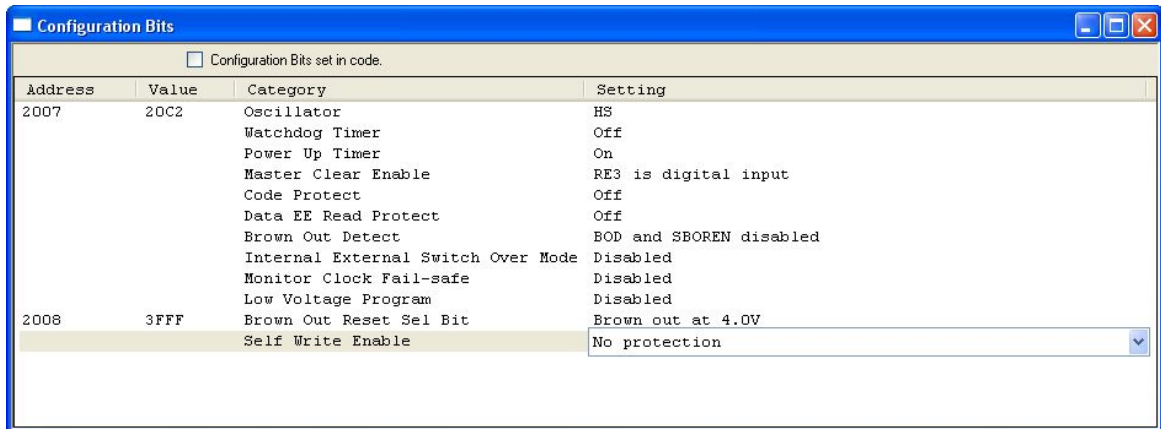
γ) Προγραμματίστε το μικροελεγκτή με τον κώδικα που υπάρχει στο τμήμα της προσομοίωσης. Ο προγραμματισμός μπορεί να πραγματοποιηθεί με οποιοδήποτε

εργαλείο είναι διαθέσιμο. Συνίσταται ο προγραμματισμός εντός του κυκλώματος ώστε να μην είναι απαραίτητη η απομάκρυνση του μικροελεγκτή από το κύκλωμα.

δ) Εφαρμόστε την τροφοδοσία στο κύκλωμα και παρατηρήστε ποια είναι η κατάσταση της διόδου LED; Στη συνέχεια πατήστε το διακόπτη S1. Ποια είναι τώρα η κατάσταση της διόδου;



Εικόνα 7. Το κύκλωμα της εργαστηριακής άσκησης 2.



Εικόνα 8. Configuration Bits.

4. Εργασίες

α) Τροποποιήστε το παραπάνω πρόγραμμα ώστε η διόδος LED να ακτινοβολεί για όσο κρατάμε πατημένο το διακόπτη S1.

β) Τροποποιήστε το παραπάνω πρόγραμμα ώστε η διόδος LED να μην ακτινοβολεί όσο κρατάμε πατημένο το διακόπτη S1.

γ) Τροποποιήστε κατάλληλα το πρόγραμμα ώστε η δίοδος LED να αναβοσβήνει.

5. Προχωρημένα θέματα

α) Να αναπτύξετε μια εφαρμογή με την οποία ο μικροελεγκτής συνδέεται και διαβάζει πατήματα πλήκτρων από ένα πληκτρολόγιο δώδεκα πλήκτρων.

β) Να αναπτύξετε μια εφαρμογή με την οποία ο μικροελεγκτής συνδέεται και διαβάζει πατήματα πλήκτρων από ένα κωδικοποιημένο πληκτρολόγιο δεκαέξι πλήκτρων.

ΧΡΟΝΙΚΕΣ ΚΑΘΥΣΤΕΡΗΣΕΙΣ

1. Θεωρία

Κατά την ανάπτυξη εφαρμογών βασισμένες σε μικροελεγκτές ένα από τα κρίσιμότερα ζητήματα που θα κληθεί να δώσει λύση ο προγραμματιστής είναι η δημιουργία χρονικών καθυστερήσεων. Ο επεξεργαστής του μικροελεγκτή πολλές φορές θα χρειαστεί να «περιμένει», μέχρι να ολοκληρωθούν πρώτα κάποιες άλλες πιο αργές λειτουργίες ή να εκτελεί συγκεκριμένα τμήματα κώδικα σε καθορισμένα χρονικά διαστήματα. Οι καθυστερήσεις που μπορεί να χρειαστούν ποικίλουν, καθώς μπορεί να είναι της τάξεως των μsec ή msec αλλά μπορεί να είναι και της τάξεως των λεπτών. Συνήθως βασίζονται στον ταλαντωτή χρονισμού ο οποίος καθορίζει τον κύκλο εντολής. Οι μικροελεγκτές PIC πραγματοποιούν εσωτερικά διαίρεση συχνότητας με το 4 για να προκύψει ο χρόνος που αντιστοιχεί σε ένα κύκλο εντολής. Έτσι ένας ταλαντωτής χρονισμού 20MHz δημιουργεί στον επεξεργαστή του PIC ένα ρολόι 5MHz(=200nsec κύκλος εντολής), ένα ταλαντωτής 4MHz δημιουργεί ένα ρολόι 1MHz(=1 μsec κύκλος εντολής) κ.ο.κ..

Υπάρχουν δύο βασικοί τρόποι με τους οποίους μπορούμε να δημιουργήσουμε χρονικές καθυστερήσεις. Ο ένας τρόπος συνοψίζεται στο να παγιδεύουμε τον έλεγχο του προγράμματος σε ένα συγκεκριμένο τμήμα κώδικα μειώνοντας κατά ένα έναν καταχωρητή μνήμης έως να μηδενιστεί. Ο δεύτερος τρόπος περιλαμβάνει τη χρησιμοποίηση ενός χρονιστή (timer), κυκλώματος που περιέχει εσωτερικά ο μικροελεγκτής. Ο timer ρυθμίζεται να μετράει σύμφωνα με κάποιο ρολόι, που συνήθως είναι το ρολόι του επεξεργαστή, και όταν υπερχειλίζει καθορίζει στο λογικό '1' την κατάσταση του bit TMR0IF του καταχωρητή INTCON.

Ο PIC16F887 διαθέτει τρεις timers. Στο σημείο αυτό θα μελετήσουμε τον Timer0. Ο Timer0 είναι 8 bit και έχει δύο τρόπους λειτουργίας. Μπορεί να λειτουργήσει είτε ως χρονιστής είτε μετρητής. Όταν λειτουργεί ως μετρητής, μετράει παλμούς που εφαρμόζονται στον ακροδέκτη RA4. Όταν λειτουργεί ως χρονιστής, μετράει σύμφωνα με τον ταλαντωτή χρονισμού. Επιπλέον ο Timer0 διαθέτει έναν προ-μετρητή (prescaler), δηλαδή μια ακόμα βαθμίδα μέτρησης κάτι που τον μετατρέπει σε μετρητή εύρους 16 bit. Η ρύθμιση του Timer0 πραγματοποιείται από τον καταχωρητή ειδικής χρήσης OPTION_REG ο οποίος παρουσιάζεται στον πίνακα 1. Για τη ρύθμιση του Timer0 ως χρονιστή ακολουθούμε τα παρακάτω βήματα:

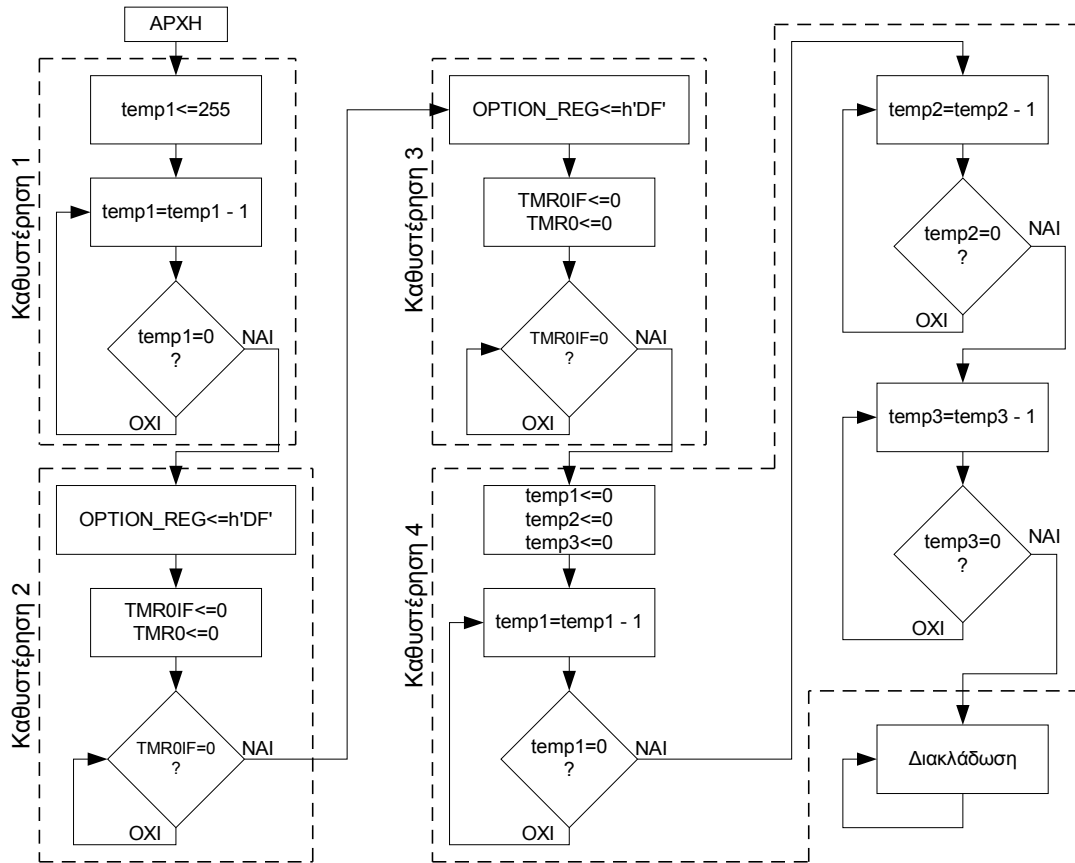
- Θέτουμε στο λογικό '1' το bit T0CS του καταχωρητή OPTION_REG
- Αν θέλουμε να ρυθμίσουμε τον prescaler στον Timer0 τότε μηδενίζουμε το bit PSA του καταχωρητή OPTION_REG
- Αν χρησιμοποιούμε τον prescaler ρυθμίζουμε κατάλληλα τα bits PS2, PS1 και PS0 ώστε να καθορίσουμε τον επιθυμητό ρυθμό μέτρησης
- Μηδενίζουμε τον καταχωρητή TMR0
- Μηδενίζουμε τη σημαία υπερχείλισης T0IF του καταχωρητή INTCON

Καταχωρητής OPTION_REG (διεύθυνση h'81' της Bank1 ή h'181' της Bank3)

bit 7	RBPV: Bit ενεργοποίησης pull-up αντιστάσεων της θύρας PORTB 1= Οι εσωτερικές pull-up αντιστάσεις της θύρας PORTB είναι απενεργοποιημένες 0= Οι εσωτερικές pull-up αντιστάσεις της θύρας PORTB είναι ενεργοποιημένες		
bit 6	INTEDG: Bit επιλογής ανερχόμενου ή κατερχόμενου μετώπου για τη σηματοδότηση διακοπής 1= Σηματοδότηση διακοπής στο ανερχόμενο μέτωπο του ακροδέκτη RB0 0= Σηματοδότηση διακοπής στο κατερχόμενο μέτωπο του ακροδέκτη RB0		
bit 5	T0CS: Bit επιλογής πηγής ρολογιού για τον Timer0 1= Μέτρηση σε κάθε μετάβαση του ακροδέκτη RA4 0= Μέτρηση με πηγή το ρολόι του επεξεργαστή		
bit 4	T0SE: Bit επιλογής μετώπου μέτρησης για τον Timer0 1= Μέτρηση κατά τη μετάβαση από υψηλό σε χαμηλό επίπεδο τάσης στον ακροδέκτη RA4 0= Μέτρηση κατά τη μετάβαση από χαμηλό σε υψηλό επίπεδο τάσης στον ακροδέκτη RA4		
bit 3	PSA: Bit ανάθεσης προμετρητή (prescaler) 1= O prescaler έχει ανατεθεί στον WatchDog Timer 0= O prescaler έχει ανατεθεί στον Timer0		
bit 2-0	PS2:PS0: Bits επιλογής εύρους prescaler		
	<u>PS2:PS1:PS0</u>	<u>Ρυθμός μέτρησης TMR0</u>	<u>Ρυθμός μέτρησης WDT</u>
	000	1:2	1:1
	001	1:4	1:2
	010	1:8	1:4
	011	1:16	1:8
	100	1:32	1:16
	101	1:64	1:32
	110	1:128	1:64
	111	1:256	1:128

Πίνακας 1

Στην παρούσα εργαστηριακή άσκηση θα γράψουμε κώδικα που δημιουργεί διαδοχικά τέσσερις διαφορετικές χρονικές καθυστερήσεις. Η πρώτη και η τέταρτη βασίζονται στο να αναθέτουν στον επεξεργαστή του μικροελεγκτή να μειώνει κατά ένα την τιμή ενός ή περισσότερων καταχωρητών μνήμης μέχρι να μηδενιστούν. Η δεύτερη και η τρίτη καθυστέρηση βασίζονται στην υπερχείλιση του Timer0. Το λογικό διάγραμμα του λογισμικού του PIC16F887 παρουσιάζεται στην εικόνα 1.



Εικόνα 1. Το λογικό διάγραμμα του προγράμματος.

2. Προσομοίωση

- Εκκινείτε το λογισμικό ανάπτυξης εφαρμογών MPLAB IDE. Δημιουργείτε μια νέα εφαρμογή βασισμένη στον PIC16F887 και αποθηκεύστε τη στο δίσκο του υπολογιστή.
- Δημιουργείτε ένα νέο αρχείο στον κειμενογράφο και αντιγράψτε τον κώδικα που ακολουθεί:

```

include <p16f887.inc>

temp1 equ h'20'
temp2 equ h'21'
temp3 equ h'22'

org h'0000'
nop

```

; Αρχή καθυστέρησης Νο 1

```

movlw d'255'      ; Φόρτωση του αριθμού 255 στον καταχωρητή w
movwf temp1       ; Αποθήκευση του w στον καταχωρητή temp1
decfsz temp1,f     ; Μείωση κατά ένα του temp1 και παράκαμψη της
                  ; επόμενης εντολής αν είναι μηδέν
goto $-1           ; Άλλα στην προηγούμενη εντολή
; Τέλος καθυστέρησης No 1

; Αρχή καθυστέρησης No 2
bsf STATUS,RP0     ; Ρύθμιση των Bank bits να δείχνουν στη Bank1
movlw b'11011111'  ; Αποθήκευση του αριθμού b'11011111' (=h'DF') στον w
movwf OPTION_REG    ; Αποθήκευση του w στον Timer0 (Ρύθμιση χρονιστή)
bcf STATUS,RP0     ; Ρύθμιση των Bank bits να δείχνουν στη Bank0
bcf INTCON,TMR0IF   ; Μηδενισμός του flag bit του Timer0
clrf TMR0           ; Μηδενισμός του Timer0
btfss INTCON,TMR0IF ; Παράκαμψη της επόμενης εντολής αν TMR0IF='1'
goto $-1           ; Άλλα στην προηγούμενη εντολή
bcf INTCON,TMR0IF   ; Μηδενισμός του bit TMR0IF
; Τέλος καθυστέρησης No 2

; Αρχή καθυστέρησης No 3
bsf STATUS,RP0     ; Ρύθμιση των Bank bits να δείχνουν στη Bank1
bcf OPTION_REG,PSA  ; Ανάθεση prescaler στον Timer0
bcf OPTION_REG,PS2  ; Ρύθμιση στον Timer0 να μετρά με prescaler 1:4
bcf OPTION_REG,PS1  ; Ρύθμιση στον Timer0 να μετρά με prescaler 1:4
bcf STATUS,RP0     ; Ρύθμιση των Bank bits να δείχνουν στη Bank0
clrf TMR0           ; Μηδενισμός του Timer0
btfss INTCON,TMR0IF ; Παράκαμψη της επόμενης εντολής αν TMR0IF='1'
goto $-1           ; Άλλα στην προηγούμενη εντολή
; Τέλος καθυστέρησης No 3

; Αρχή καθυστέρησης No 4
clrf temp1          ; Μηδενισμός καταχωρητή temp1
clrf temp2          ; Μηδενισμός καταχωρητή temp2
clrf temp3          ; Μηδενισμός καταχωρητή temp3
LONG_DELAY          ; Καθορισμός ετικέτας
decfsz temp1,f      ; Μείωση κατά ένα του temp1 και παράκαμψη της
                  ; επόμενης εντολής αν είναι μηδέν
goto LONG_DELAY     ; Άλλα στην εντολή με ετικέτα LONG_DELAY
decfsz temp2,f      ; Μείωση κατά ένα του temp2 και παράκαμψη της
                  ; επόμενης εντολής αν είναι μηδέν
goto LONG_DELAY     ; Άλλα στην εντολή με ετικέτα LONG_DELAY
decfsz temp3,f      ; Μείωση κατά ένα του temp3 και παράκαμψη της
                  ; επόμενης εντολής αν είναι μηδέν
goto LONG_DELAY     ; Άλλα στην εντολή με ετικέτα LONG_DELAY
; Τέλος καθυστέρησης No 4
goto $              ; Άλλα στην ίδια εντολή

```

end

γ) Να μεταγλωττίσετε την εφαρμογή και να διορθώσετε όποια συντακτικά λάθη υπάρχουν. Να εμφανίσετε το παράθυρο παρακολούθησης μεταβλητών (Επιλογή View/Watch). Στο watch window να προσθέσετε τους καταχωρητές temp1, temp2 και temp3.

δ) Ενεργοποιείτε τον simulator του MPLAB (Debugger>Select Tool>MPLAB SIM) και με την επιλογή Debugger>Stopwatch να εμφανίσετε το παράθυρο Stopwatch. Το παράθυρο αυτό παρουσιάζει πληροφορίες σχετικά με το χρόνο εκτέλεσης των διάφορων εντολών.

ε) Με την επιλογή Debugger>Settings εμφανίστε το παράθυρο ρυθμίσεων του simulator. Επιλέξτε την καρτέλα Osc/Trace. Στο πεδίο κειμένου Processor Frequency πληκτρολογήστε τον αριθμό 20. Στα radio buttons Units επιλέξτε τα MHz. Οι παραπάνω ρυθμίσεις καθορίζουν την προσομοίωση του μικροελεγκτή με ταλαντωτή χρονισμού στα 20MHz άρα με ένα κύκλο εντολής στα 200nsec.

στ) Εκτελέστε το πρόγραμμα στον simulator και με βάση το παράθυρο stopwatch συμπληρώστε στον πίνακα που ακολουθεί το χρονικό διάστημα στο οποίο διαρκεί η κάθε καθυστέρηση.

ΚΑΘΥΣΤΕΡΗΣΗ	ΧΡΟΝΙΚΟ ΔΙΑΣΤΗΜΑ
Καθυστέρηση No 1	
Καθυστέρηση No 2	
Καθυστέρηση No 3	
Καθυστέρηση No 4	

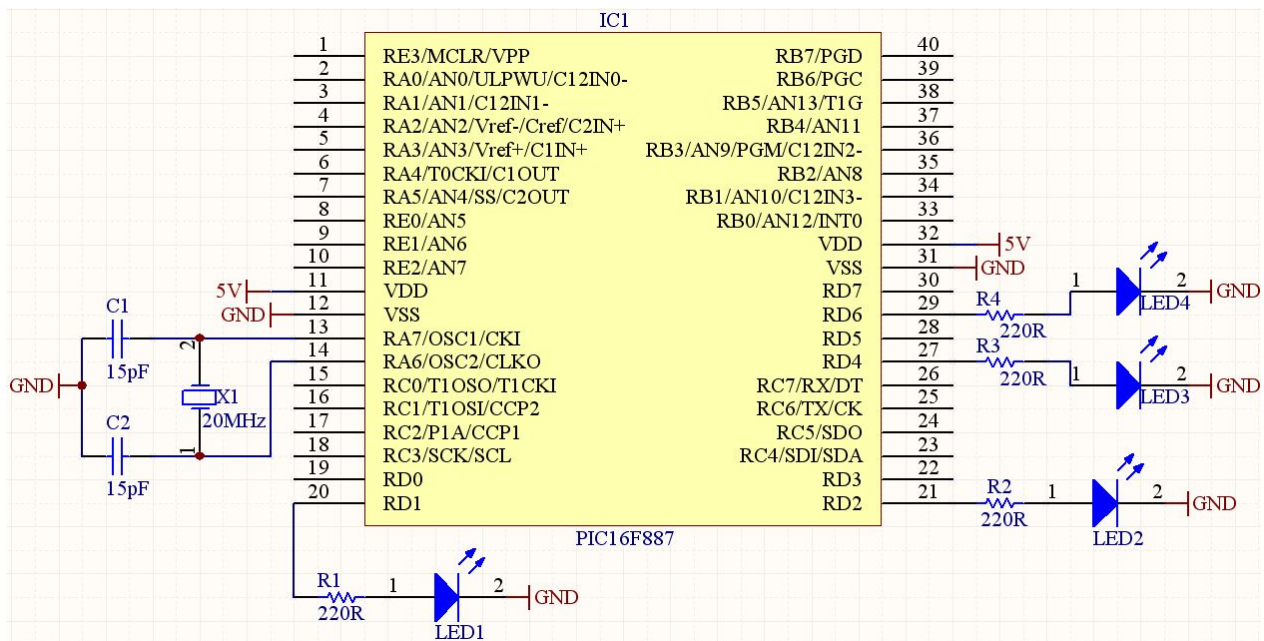
Πίνακας 2

3.Υλοποίηση

Για την υλοποίηση του κυκλώματος της εργαστηριακής άσκησης απαιτούνται τα παρακάτω υλικά:

IC1	=	PIC16F887
R1	=	220R, $\geq 1/8W$
R2	=	220R, $\geq 1/8W$
R3	=	220R, $\geq 1/8W$
R4	=	220R, $\geq 1/8W$
LED1	=	Δίοδος LED
LED2	=	Δίοδος LED

LED3 = Δίοδος LED
 LED4 = Δίοδος LED
 C1 = 15pF, ≥6V
 C2 = 15pF, ≥6V
 X1 = Κρύσταλος 20MHz



Εικόνα 2. Το κύκλωμα της εργαστηριακής άσκησης 3.

α) Υλοποιήστε σε πειραματική πλακέτα το κύκλωμα της εικόνας 2.

β) Προσθέστε στο σημείο πριν την πρώτη καθυστέρηση τον παρακάτω κώδικα:

```
bsf STATUS,RP0
movlw b'10101001'
andwf TRISD,f
bcf STATUS,RP0
clrf PORTD
bsf PORTD,1
bcf PORTD,1
```

Προσθέστε ανάμεσα στο τέλος της πρώτης καθυστέρησης και στην αρχή της δεύτερης την παρακάτω εντολή:

```
bsf PORTD,1
```

Προσθέστε στο σημείο ανάμεσα στο τέλος της δεύτερης καθυστέρησης και στην αρχή της τρίτης την παρακάτω εντολή:

```
bsf PORTD,2
```

Προσθέστε στο σημείο ανάμεσα στο τέλος της τρίτης καθυστέρησης και στην αρχή της τέταρτης την παρακάτω εντολή:

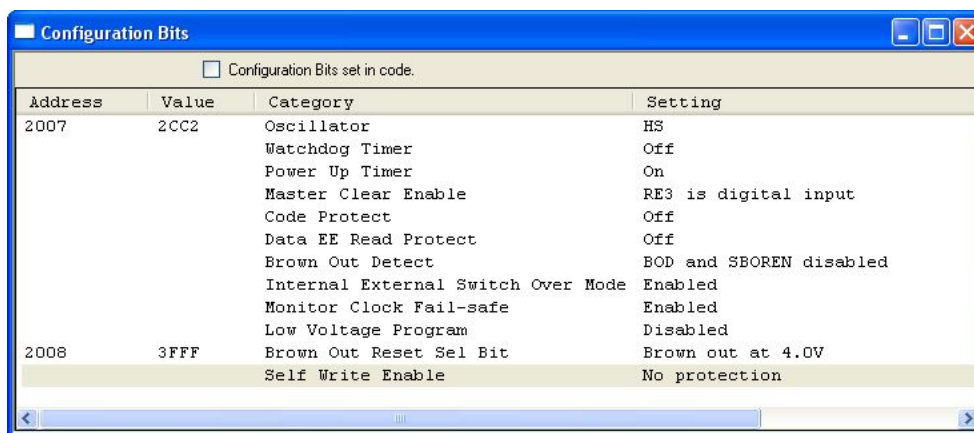
```
bsf PORTD, 4
```

Προσθέστε στο σημείο μετά το τέλος της τέταρτης καθυστέρησης την παρακάτω εντολή:

```
bsf PORTD, 6
```

γ) Μεταγλωττίστε και διορθώστε τα όποια συντακτικά λάθη υπάρχουν.

δ) Καθορίστε τα Configuration Bits όπως φαίνονται στην εικόνα 3.



Address	Value	Category	Setting
2007	2CC2	Oscillator	HS
		Watchdog Timer	Off
		Power Up Timer	On
		Master Clear Enable	RE3 is digital input
		Code Protect	Off
		Data EE Read Protect	Off
		Brown Out Detect	BOD and SBOREN disabled
		Internal External Switch Over Mode	Enabled
		Monitor Clock Fail-safe	Enabled
		Low Voltage Program	Disabled
		Brown Out Reset Sel Bit	Brown out at 4.0V
		Self Write Enable	No protection
2008	3FFF		

Εικόνα 3. Configuration bits.

ε) Φορτώστε τον κώδικα στο μικροελεγκτή και εφαρμόστε την τροφοδοσία. Να αναφέρετε τι παρατηρείτε. Ακτινοβολούν όλες οι LED; Περιγράψτε τι συμβαίνει στο κύκλωμα.

στ) Συνδέστε παλμογράφο στους ακροδέκτες RD1, RD2 και RD4. Ελέγξτε με τον παλμογράφο ότι η μέτρηση των χρονικών καθυστερήσεων κατά την προσομοίωση ανταποκρίνεται στην πραγματικότητα. Να σχεδιάσετε το παλμογράφημα.

4. Εργασίες

α) Να δημιουργήσετε μια καθυστέρηση περίπου στο 1msec.

β) Να δημιουργήσετε μια καθυστέρηση περίπου στα 10msec.

γ) Τροποποιήστε το πρόγραμμα ώστε μια LED να αναβοσβήνει.

5. Προχωρημένα θέματα

- α) Να γράψετε ένα πρόγραμμα στο οποίο ο μικροελεγκτής δημιουργεί σε έναν από τους ακροδέκτες του μια τετραγωνική κυματομορφή συχνότητας 100Hz.
- β) Να δημιουργήσετε μια καθυστέρηση 5 δευτερόλεπτα.

ΜΝΗΜΗ ΔΕΔΟΜΕΝΩΝ EEPROM

1. Θεωρία

Ο μικροελεγκτής PIC16F887 διαθέτει μνήμη δεδομένων EEPROM (Electrically Erasable Programmable Read Only Memory) εύρους 256 bytes. Η μνήμη αυτή διατηρεί τα περιεχόμενά της ακόμα και μετά τη διακοπή της παροχής τροφοδοσίας. Για τη διαχείριση στις θέσεις μνήμης της EEPROM ο PIC16F887 χρησιμοποιεί τους παρακάτω καταχωρητές ειδικής χρήσης:

- EEDAT
- EEADR
- EECON1
- EECON2

Ο καταχωρητής EEDAT χρησιμοποιείται για την αποθήκευση των δεδομένων προς ανάγνωση ή εγγραφή ενώ ο EEADR για χρησιμοποιείται για τη διευθυνσιοδότηση των θέσεων μνήμης της EEPROM. Η διαδικασία ανάγνωσης και εγγραφής ελέγχεται από τους καταχωρητές ειδικής χρήσης EECON1 και EECON2. Στον πίνακα 1 παρουσιάζεται ο καταχωρητής ειδικής χρήσης EECON1.

Καταχωρητής EECON1 (διεύθυνση h'18C' της Bank3)

bit 7	EEPGD: Bit επιλογής για flash μνήμη προγράμματος/μνήμης δεδομένων EEPROM 1= Διαχείριση της flash μνήμης προγράμματος 0= Διαχείριση της μνήμης δεδομένων EEPROM
bit 6-4	Δεν είναι υλοποιημένα
bit 3	WRERR: Σημαία σφάλματος της μνήμης EEPROM 1= Η λειτουργία εγγραφής τερματίστηκε πρόωρα 0= Η λειτουργία εγγραφής ολοκληρώθηκε επιτυχώς
bit 2	WREN: Bit επίτρεψης διαδικασίας εγγραφής 1= Επίτρεψη έναρξης κύκλου εγγραφής 0= Απαγόρευση εγγραφών στην EEPROM
bit 1	WR: Bit ελέγχου εγγραφής 1= Εκκίνηση κύκλου εγγραφής. Το bit μηδενίζεται αυτόματα μόλις ολοκληρωθεί η εγγραφή 0= Ο κύκλος εγγραφής στην EEPROM έχει ολοκληρωθεί
bit 0	RD: Bit ελέγχου ανάγνωσης 1= Εκκίνηση της διαδικασίας ανάγνωσης. Το bit μηδενίζεται αυτόματα μόλις ολοκληρωθεί η ανάγνωση 0= Δεν βρίσκεται σε εξέλιξη διαδικασία ανάγνωσης

Πίνακας 1

Η διαδικασία ανάγνωσης και εγγραφής από και προς τη μνήμη EEPROM περιλαμβάνει κάποια συγκεκριμένα βήματα που θα πρέπει να ακολουθήσει ο προγραμματιστής του μικροελεγκτή και συνοψίζονται στα εξής:

Διαδικασία ανάγνωσης από τη μνήμη EEPROM:

- α) Αποθήκευση στον καταχωρητή EEADR της διεύθυνσης μνήμης EEPROM που πρέπει να διαβαστεί.
- β) Μηδενισμός του bit 7 (EEPGD) του καταχωρητή EECON1.
- γ) Καθορισμός στο λογικό '1' του bit 0 (RD) του καταχωρητή EECON1.
- δ) Ανάγνωση του καταχωρητή EEDAT στον οποίο θα έχουν αποθηκευτεί τα δεδομένα της μνήμης EEPROM.

Διαδικασία εγγραφής στη μνήμη EEPROM:

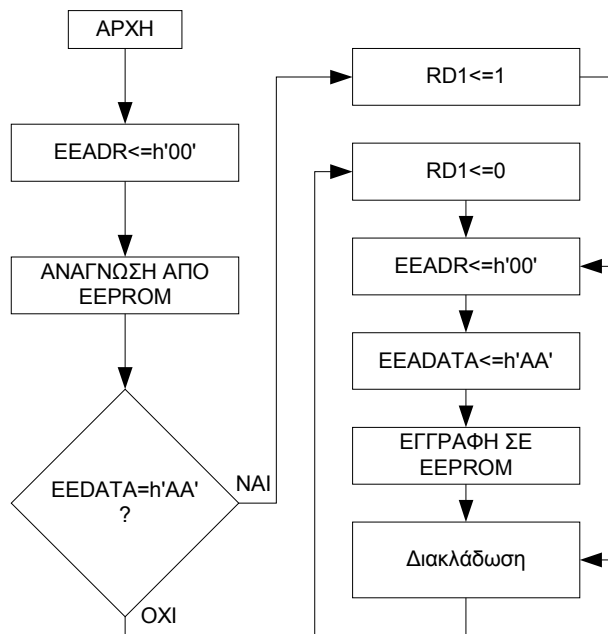
- α) Αποθήκευση στον καταχωρητή EEADR της διεύθυνσης μνήμης της EEPROM στην οποία θα γίνει η εγγραφή.
- β) Αποθήκευση στον καταχωρητή EEDAT των δεδομένων προς εγγραφή.
- γ) Μηδενισμός του bit 7 (EEPGD) του καταχωρητή EECON1.
- δ) Καθορισμός στο λογικό '1' του bit WREN του καταχωρητή EECON1.
- ε) Εκτέλεση της παρακάτω ακολουθίας εντολών:

```
movlw h'55'  
movwf EECON2  
movlw h'AA'  
movwf EECON2
```

- στ) Καθορισμός στο λογικό '1' του bit 1 (WR) του καταχωρητή EECON1.
 - ζ) Έλεγχος μηδενισμού του bit WR του καταχωρητή EECON1 για την επιτυχή ολοκλήρωση της εγγραφής.
- Ο κύκλος εγγραφής αρχίζει όταν ο προγραμματιστής καθορίσει στο λογικό '1' το bit WR του EECON1. Όταν ολοκληρωθεί εσωτερικά ο κύκλος εγγραφής τότε το bit αυτό μηδενίζεται αυτόματα. Έτσι, ελέγχοντας τότε αυτό το bit θα γίνει μηδέν εξακριβώνουμε τότε ολοκληρώθηκε η εγγραφή στην EEPROM.

Στην παρούσα εργαστηριακή άσκηση θα αναπτυχθεί μια εφαρμογή κατά την οποία ο μικροελεγκτής κατά την εκκίνηση θα διαβάσει τα περιεχόμενα της EEPROM από τη θέση h'00' και εάν τα περιεχόμενα αυτά είναι ο αριθμός h'AA' τότε θα οδηγή στο λογικό '1' τον ακροδέκτη RD1 του μικροελεγκτή. Εάν τα περιεχόμενα της θέσης h'00' είναι ένας αριθμός διάφορος του h'AA' τότε θα οδηγή στο λογικό '0' τον ακροδέκτη RD1 του μικροελεγκτή. Στη συνέχεια θα πραγματοποιεί μια εγγραφή του αριθμού h'AA' στη

θέση h'00' της EEPROM. Το λογικό διάγραμμα του λογισμικού του μικροελεγκτή παρουσιάζεται στην εικόνα 1.



Εικόνα 1. Το λογικό διάγραμμα του προγράμματος.

2. Προσομοίωση

- Εκκινείτε το λογισμικό ανάπτυξης εφαρμογών MPLAB IDE. Δημιουργείτε μια νέα εφαρμογή βασισμένη στον PIC16F887 και αποθηκεύστε τη στο δίσκο του υπολογιστή.
- Δημιουργείτε ένα νέο αρχείο στον κειμενογράφο και αντιγράψτε τον κώδικα που ακολουθεί:

```
include <p16f887.inc>

org h'0000'
nop

bcf PORTD,1      ; Καθορισμός του RD1 στο λογικό '0'
bsf STATUS,RP0   ; Ρύθμιση των Bank bits να δείχνουν στη Bank1
bcf TRISD,1      ; Καθορισμός του RD1 ως έξοδος

; Εντολές ανάγνωσης από τη θέση h'00' της EEPROM
bcf STATUS,RP0   ; Ρύθμιση των Bank bits να δείχνουν
bsf STATUS,RP1   ; στη Bank2
```

```

clrf EEADR          ; Καθορισμός του EEADR να δείχνει στη θέση μνήμης 0
bsf STATUS,RP0      ; Ρύθμιση των Bank bits να δείχνουν στη Bank3
bcf EECON1,EEPGD     ; Μηδενισμός του bit EEPGD του καταχωρητή EECON1
bsf EECON1,RD        ; Εκκίνηση του κύκλου ανάγνωσης από την EEPROM
bcf STATUS,RP0      ; Ρύθμιση των Bank bits να δείχνουν στη Bank2

movlw h'AA'          ; Φόρτωση του αριθμού h'AA' στον καταχωρητή w
subwf EEDAT,w        ; Αφαίρεση του καταχωρητή w από τον καταχωρητή EEDAT
bcf STATUS,RP1      ; Ρύθμιση των Bank bits να δείχνουν στη Bank0
btfsc STATUS,Z       ; Παράκαμψη της επόμενης εντολής αν Z=0
    bsf PORTD,1      ; Καθορισμός του RD1 στο λογικό '1'
btfss STATUS,Z       ; Παράκαμψη της επόμενης εντολής αν Z=1
    bcf PORTD,1      ; Καθορισμός του RD1 στο λογικό '0'

; Εντολές εγγραφής στη θέση h'00' του αριθμού h'AA'
bsf STATUS,RP1      ; Ρύθμιση των Bank bits να δείχνουν στη Bank2
clrf EEADR          ; Καθορισμός του EEADR να δείχνει στη θέση μνήμης 0
movlw h'AA'          ; Φόρτωση του αριθμού h'44'=d'68' στον w
movwf EEDAT          ; Αποθήκευση των περιεχομένων του w στον EEDAT
bsf STATUS,RP0      ; Ρύθμιση των Bank bits να δείχνουν στη Bank3
bcf EECON1,EEPGD     ; Μηδενισμός του bit EEPGD του καταχωρητή EECON1
bsf EECON1,WREN      ; Καθορισμός στο λογικό '1' του bit WREN του EECON1

movlw h'55'          ;
movwf EECON2         ; Απαραίτητη ακολουθία εντολών για την
movlw h'AA'          ; πραγματοποίηση εγγραφής στην EEPROM
movwf EECON2         ;

bsf EECON1,WR        ; Εκκίνηση του κύκλου εγγραφής στην EEPROM
btfsc EECON1,WR      ; Παράκαμψη της επόμενης εντολής αν WR=0
    goto $-1         ; Πήγαινε στην προηγούμενη εντολή

goto $               ; Άλλα στην ίδια εντολή

end

```

γ) Μεταγλωττίστε την εφαρμογή και διορθώστε τυχόν συντακτικά λάθη που υπάρχουν. Ενεργοποιήστε τον simulator και εμφανίστε με την επιλογή View>EEPROM το παράθυρο της μνήμης EEPROM το οποίο θα πρέπει να φαίνεται όπως δείχνει η εικόνα 2.

δ) Ποια είναι τα περιεχόμενα της EEPROM στη διεύθυνση h'00'; Να εκτελεστεί εντολή προς εντολή το τμήμα του προγράμματος που περιλαμβάνει τις εντολές ανάγνωσης από την EEPROM. Πώς καθορίζεται στη συνέχεια η έξοδος RD1 και γιατί;

ε) Προσθέστε ένα breakpoint στην τελευταία εντολή του προγράμματος (`goto $`) τοποθετώντας το δρομέα επάνω στην γραμμή της συγκεκριμένης εντολής και κάνοντας

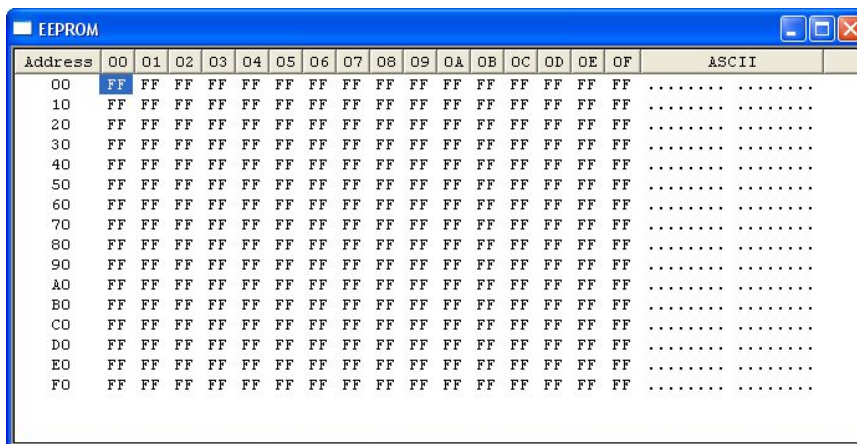
διπλό κλικ. Εάν τοποθετηθεί σωστά το breakpoint τότε θα εμφανιστεί το γράμμα B στην αρχή της γραμμής μέσα σε κόκκινο κυκλικό πλαίσιο.

ε) Να εκτελέσετε εντολή προς εντολή το τμήμα του προγράμματος που περιλαμβάνει τις εντολές εγγραφής στην EEPROM. Παρατηρείστε ότι ο έλεγχος του προγράμματος «κολλάει» στο σημείο που βρίσκονται οι παρακάτω εντολές:

```
btfsc EECON1,WR    ; Παράκαμψη της επόμενης εντολής αν WR=0  
goto $-1            ; Πήγαινε στην προηγούμενη εντολή
```

Στο σημείο αυτό πατήστε το πλήκτρο F9 ώστε να εκτελεστεί κανονικά η ροή του προγράμματος. Η εκτέλεση θα σταματήσει στην εντολή που έχει προστεθεί το breakpoint. Το σημείο στο οποίο κολλάει το πρόγραμμα αναπαριστάνει το χρόνο που χρειάζεται εσωτερικά ο μικροελεγκτής για να πραγματοποιήσει την εγγραφή στην EEPROM. Όταν ολοκληρωθεί ο κύκλος εγγραφής και το πρόγραμμα «ξεκολλήσει» ποια θα πρέπει να είναι τα περιεχόμενα της EEPROM στη διεύθυνση h'00';

στ) Πραγματοποιείτε RESET στο μικροελεγκτή και επαναλάβετε την παραπάνω διαδικασία. Εκτελώντας ξανά το πρόγραμμα να αναφέρετε ποια θα είναι η λογική κατάσταση της εξόδου RD1 και γιατί;



Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
40	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
50	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
60	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
70	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
90	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

Εικόνα 2. Το παράθυρο της EEPROM του PIC16F887.

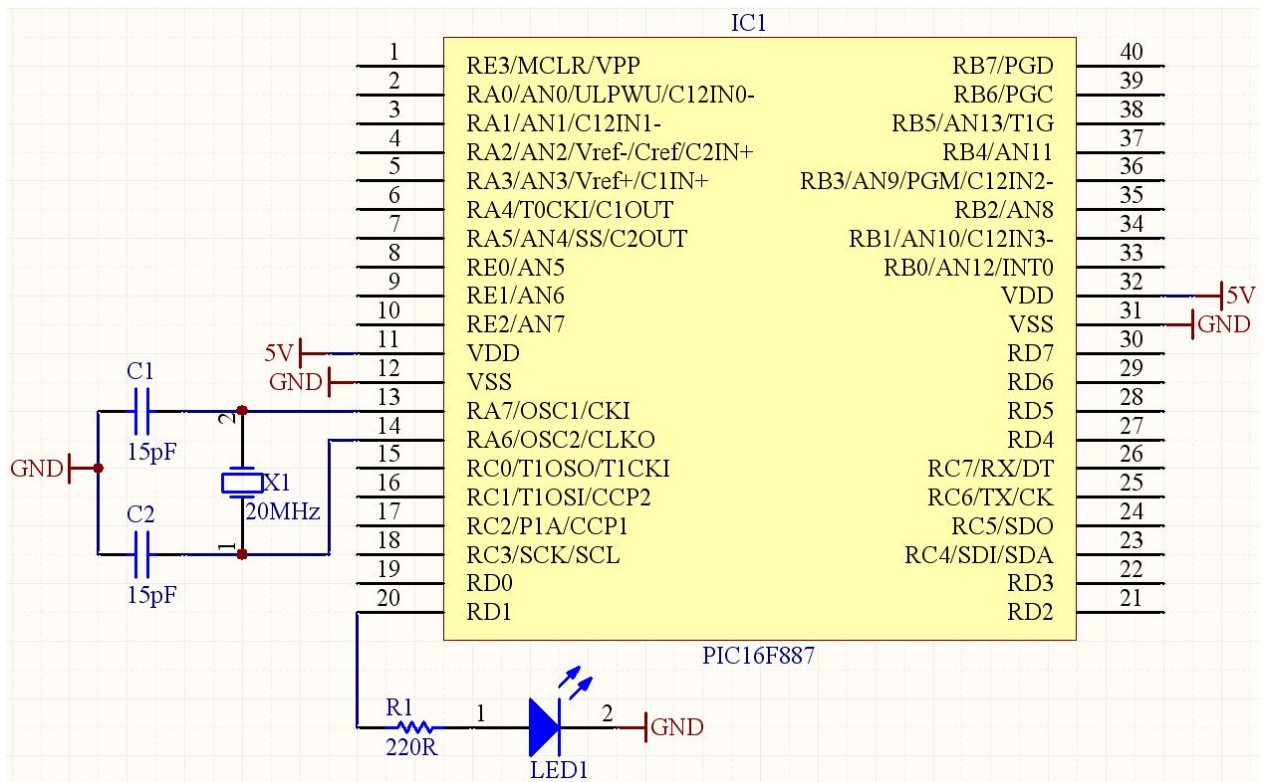
3. Υλοποίηση

Για την υλοποίηση του κυκλώματος της εργαστηριακής άσκησης απαιτούνται τα παρακάτω υλικά:

IC1 = PIC16F887

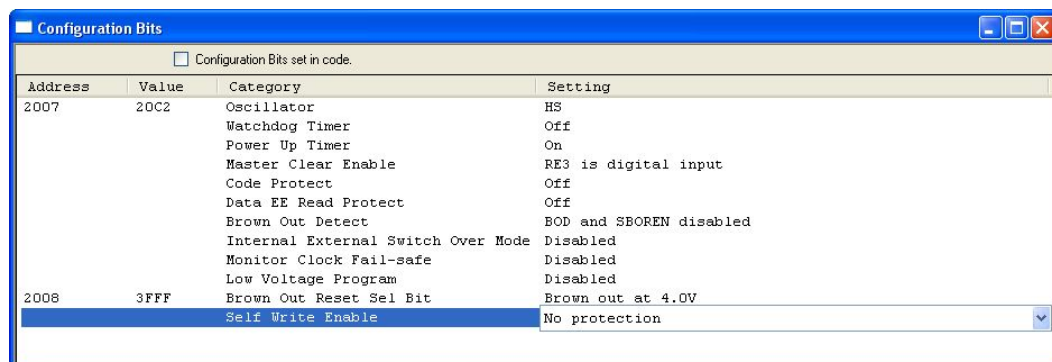
R1 = 220R, $\geq 1/8W$
 LED1 = Δίοδος LED
 C1 = 15pF, $\geq 6V$
 C2 = 15pF, $\geq 6V$
 X1 = Κρύσταλος 20MHz

α) Υλοποιήστε σε πειραματική πλακέτα το κύκλωμα της εικόνας 3.



Εικόνα 3. Το κύκλωμα της εργαστηριακής άσκησης 4.

β) Διαμορφώστε τα Configuration Bits όπως φαίνονται στην εικόνα 4.



Εικόνα 4. Configuration Bits.

γ) Μεταγλωττίστε, διορθώστε συντακτικά λάθη και προγραμματίστε το μικροελεγκτή με το πρόγραμμα που προσομοιώθηκε στο προηγούμενο τμήμα της άσκησης.

δ) Εφαρμόστε την τροφοδοσία στο κύκλωμα. Ποια είναι η κατάσταση της LED; Είναι αναμμένη; Σε ποιο συμπέρασμα οδηγούμαστε για τα περιεχόμενα της EEPROM στη θέση h'00' πριν την εφαρμογή της τροφοδοσίας; Μετά την εκτέλεση του προγράμματος ποια δεδομένα αποθηκεύτηκαν σε αυτή τη θέση μνήμης.

ε) Αποσυνδέστε την τροφοδοσία. Κατά την αποσύνδεση τα δεδομένα στην EEPROM πρέπει να διατηρηθούν. Εφαρμόστε την τροφοδοσία και πάλι. Αυτή τη φορά ποια είναι η κατάσταση της LED και γιατί;

στ) Τροποποιήστε το πρόγραμμα ώστε να αποθηκεύεται στην EEPROM ο αριθμός h'FF'. Μεταγλωττίστε και προγραμματίστε το μικροελεγκτή. Εφαρμόστε την τροφοδοσία και αναφέρετε ποια είναι κατάσταση της LED.

ζ) Προσθέστε στο σημείο αμέσως μετά την ολοκλήρωση του κύκλου εγγραφής τις παρακάτω εντολές:

```
bsf STATUS,RP0  
bcf STATUS,RP1  
bcf TRISD,2  
bcf STATUS,RP0  
bsf PORTD,2
```

Προγραμματίστε το μικροελεγκτή με το τροποποιημένο πρόγραμμα. Συνδέστε έναν παλμογράφο στους ακροδέκτες RD1 και RD2. Να υπολογίσετε με τη χρήση του παλμογράφου πόσο διαρκεί ο κύκλος εγγραφής στην EEPROM του PIC16F887. Να σχεδιάσετε το παλμογράφημα.

4. Εργασίες

α) Να γράψετε πρόγραμμα το οποίο αποθηκεύει τους αριθμούς 150 και 250 στις θέσεις μνήμης h'10' και h'20' της EEPROM του PIC16F887.

β) Να διαβάσετε το περιεχόμενο της θέσης μνήμης h'15' και να το αποθηκεύσετε στη θέση μνήμης h'25'.

4. Προχωρημένα θέματα

- α) Να γράψετε ένα πρόγραμμα το οποίο θα αποθηκεύει σε κάθε θέση της EEPROM τη διεύθυνσή της. Στη θέση μηδέν να αποθηκευτεί ο αριθμός 0, στη θέση ένα ο αριθμός 1, ...στη θέση h'FF' να αποθηκευτεί ο αριθμός 255 (=h'FF').
- β) Να γράψετε ένα πρόγραμμα το οποίο πραγματοποιεί ταξινόμηση των αριθμών που βρίσκονται αποθηκευμένοι στις θέσεις μνήμης της EEPROM από την h'10' έως τη θέση h'17'. Να τρέξετε το πρόγραμμα αφού αποθηκεύσετε πρώτα στις οκτώ θέσεις τις EEPROM τους εξής αριθμούς: 84, 14, 240, 158, 100, 200, 22, 50.

ΔΙΑΚΟΠΕΣ

1. Θεωρία

Οι διακοπές (interrupts) αποτελούν ένα πολύ βασικό αλλά και ισχυρό χαρακτηριστικό των μικροελεγκτών. Δίνοντας έναν ορισμό για το τι είναι οι διακοπές θα μπορούσαμε να πούμε ότι:

Διακοπή είναι το χαρακτηριστικό του μικροελεγκτή ή του επεξεργαστή κατά το οποίο όταν ικανοποιηθούν κάποιες συνθήκες διακόπτεται η κανονική ροή του προγράμματος και εκτελείται ένα συγκεκριμένο τμήμα κώδικα, γνωστό ως ρουτίνα διαχείρισης διακοπής, ενώ μετά την εξυπηρέτηση το πρόγραμμα επιστρέφει στο σημείο που συνέβει η διακοπή.

Όταν σηματοδοτηθεί μια διακοπή στον PIC16F887, ο έλεγχος του προγράμματος μεταφέρεται στο τμήμα κώδικα που βρίσκεται στη διεύθυνση μνήμης h'0004' η οποία ονομάζεται και **διάνυσμα διακοπής** (interrupt vector). Η επιστροφή στο σημείο που βρισκόταν ο μικροελεγκτής πριν τη διακοπή πραγματοποιείται με την εκτέλεση της εντολής retfie (return from interrupt).

Ο PIC16F887 χρησιμοποιεί πέντε καταχωρητές για τη ρύθμιση των διακοπών. Ο καταχωρητής INTCON είναι υπεύθυνος για τα εσωτερικά interrupt του PIC. Οι καταχωρητές PIE1 και PIE2 (Peripheral Interrupt Enable) είναι υπεύθυνοι για την ενεργοποίηση ή την απενεργοποίηση των διακοπών που αιτούνται τα περιφερειακά. Οι καταχωρητές PIR1 και PIR2 (Peripheral Interrupt Request) περιέχουν τις σημαίες (flag bits) των διακοπών. Στο τέλος της παραγράφου εξηγούνται αναλυτικά οι πέντε καταχωρητές του PIC16F887 που διαχειρίζονται τις διακοπές.

Στον PIC16F887 υπάρχουν διάφορες πηγές που μπορούν να προκαλέσουν interrupt. Οι διακοπές μπορούν να απενεργοποιηθούν συνολικά μηδενίζοντας το bit GIE (Global Interrupt Enable) του καταχωρητή INTCON. Με το bit PEIE (Peripheral Interrupt Enable) απενεργοποιούνται συνολικά οι διακοπές που ζητούν τα περιφερειακά του PIC. Διακοπή γενικά μπορεί να προκληθεί στις παρακάτω περιπτώσεις:

- Σε κατερχόμενα ή ανερχόμενα μέτωπα παλμών στον ακροδέκτη RB0
- Σε κάθε αλλαγή της λογικής κατάστασης στους ακροδέκτες της θύρας PORTB
- Κατά τη λήψη ή την αποστολή δεδομένων από τα ενσωματωμένα κυκλώματα σειριακής επικοινωνίας

- Κατά την ταυτόχρονη αποστολή και σύγκρουση (collision) δεδομένων πάνω στο δίαυλο επικοινωνίας I²C
- Κατά την υπερχείλιση κάποιου timer από τους τρεις που διαθέτει ο PIC16F887
- Κατά την ολοκλήρωση μετατροπής αναλογικού σήματος σε ψηφιακό
- Από το κύκλωμα Capture/Compare/PWM κατά τη διαδικασία σύλληψης ενός εξωτερικού συμβάντος (Capture mode) ή κατά τη διαδικασία σύγκρισης εσωτερικών καταστάσεων (Compare mode)
- Κατά την αλλαγή στην έξοδο κάποιου από τους δύο συγκριτές του PIC16F887
- Κατά την ολοκλήρωση της διαδικασίας εγγραφής στην EEPROM
- Κατά την αποτυχία λειτουργίας εξωτερικού ταλαντωτή
- Κατά την επαναφορά σε κανονική λειτουργία από λειτουργία χαμηλής κατανάλωσης ισχύος με τη χρήση του χαρακτηριστικού Ultra Low Power Wake Up.

Κατά την εξυπηρέτηση μιας διακοπής, πριν πραγματοποιηθεί οποιαδήποτε άλλη λειτουργία, πρέπει στο πρόγραμμα του μικροελεγκτή να αποθηκευτούν οι τιμές του συσσωρευτή W (accumulator) και του καταχωρητή STATUS. Οι τιμές αυτές θα πρέπει να επαναφερθούν στους καταχωρητές αυτούς πριν την επιστροφή από τη διακοπή. Όταν το λογισμικό του μικροελεγκτή εκτείνεται σε πολλές σελίδες μνήμης τότε πρέπει να αποθηκευτεί και ο καταχωρητής PCLATH. Οι καταχωρητές αυτοί χαρακτηρίζονται ιδιαίτερα κρίσιμοι για τη σωστή εκτέλεση των λειτουργιών του μικροελεγκτή καθώς μια απώλεια των δεδομένων τους θα μπορούσε να προκαλέσει από λογικά σφάλματα έως και κατάρρευση του προγράμματος. Για την επίλυση αυτού του ζητήματος ένας τρόπος είναι να τοποθετηθεί ο παρακάτω κώδικας στο άνωσμα διακοπής:

```
org h'0004'          ; Οδηγία προς τον assembler ανύσματος διακοπής
movwf W_TEMP         ; Αποθήκευση του w στον προσωρινό καταχωρητή W_TEMP
swapf STATUS,w       ; Αποθήκευση του STATUS στον w με αντιστροφή του nibble
clrf STATUS          ; Μηδενισμός του STATUS ώστε να δείχνει στη Bank0
movwf STATUS_TEMP    ; Αποθήκευση του w στον STATUS_TEMP της Bank0
movf PCLATH,w         ; Αποθήκευση του PCLATH στον w
movwf PCLATH_TEMP    ; Αποθήκευση του w στον PCLATH_TEMP της Bank0
;
; Interrupt Service Routine
;
movf PCLATH_TEMP,w    ; Αποθήκευση του PCLATH_TEMP στον w
movwf PCLATH          ; Αποθήκευση του w στον PCLATH
swapf STATUS_TEMP,w  ; Επαναφορά της τιμής του STATUS_TEMP στον w
movwf STATUS          ; Αποθήκευση στον STATUS
swapf W_TEMP,f        ; Αντιστροφή των bits χαμηλής τάξης με τα bits υψηλής
swapf W_TEMP,w        ; τάξης του W_TEMP και αποθήκευση στον w
retfie               ; Επιστροφή από τη διακοπή
```

Ο παραπάνω κώδικας κάνει χρήση των καταχωρητών W_TEMP, STATUS_TEMP και PCLATH_TEMP στους οποίους αποθηκεύει τις τιμές των καταχωρητών W, STATUS και PCLATH. Για τον καταχωρητή W_TEMP θα πρέπει να είναι δεσμευμένες τέσσερις επικαλυπτόμενες θέσεις μνήμης στις τέσσερις Bank του μικροελεγκτή. Ως επικαλυπτόμενες θέσεις μνήμης στις Bank του PIC ορίζουμε τις θέσεις εκείνες που τα επτά λιγότερο σημαντικά ψηφία της διεύθυνσής τους είναι ίδια. Για παράδειγμα επικαλυπτόμενες είναι οι θέσεις με διευθύνσεις h'20' (Bank0), h'40' (Bank1), h'120' (Bank2), h'140' (Bank3). Ο STATUS_TEMP και ο PCLATH_TEMP θα πρέπει να είναι δηλωμένοι στη Bank0.

Στην παρούσα εργαστηριακή άσκηση θα εξεταστεί η διακοπή που σηματοδοτείται από τον ακροδέκτη RB0. Ο ακροδέκτης RB0 χρησιμοποιείται για να ανιχνεύει όλα τα ανερχόμενα μέτωπα (ή αν έχει ρυθμιστεί διαφορετικά όλα τα κατερχόμενα μέτωπα) σηματοδοτώντας μια διακοπή. Για τη ρύθμιση της διακοπής του RB0 ακολουθούμε τα παρακάτω βήματα:

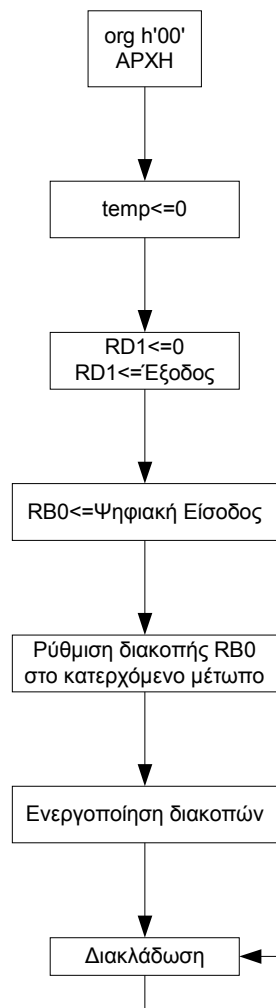
- Ρυθμίζουμε τον ακροδέκτη RB0 ως ψηφιακή είσοδο θέτοντας στο λογικό '0' το bit 4 του καταχωρητή ANSELH και στο λογικό '1' το bit 0 του καταχωρητή TRISB
- Καθορίζουμε την επιθυμητή λογική κατάσταση στο bit INTEDG του καταχωρητή OPTION_REG. Εάν είναι στο λογικό '0' τότε η διακοπή σηματοδοτείται σε κάθε κατερχόμενο μέτωπο που εντοπίζεται στον ακροδέκτη RB0 ενώ εάν είναι στο λογικό '1' σε κάθε ανερχόμενο
- Θέτουμε στο λογικό '1' το bit INTE του καταχωρητή INTCON για να ενεργοποιήσουμε τις διακοπές που σηματοδοτούνται από τον ακροδέκτη RB0
- Μηδενίζουμε το bit INTF για την αποφυγή διακοπής σε περίπτωση που ήδη έχει σηματοδοτηθεί
- Θέτουμε στο λογικό '1' το bit GIE του καταχωρητή INTCON για την ενεργοποίηση των διακοπών

Από το σημείο αυτό και μετά οποιοδήποτε κατερχόμενο ή ανερχόμενο μέτωπο (ανάλογα πώς έχει ρυθμιστεί) εμφανιστεί στον ακροδέκτη RB0 θα σηματοδοτήσει μια διακοπή και θα προκαλέσει τη μετάβαση του προγράμματος στο διάνυσμα διακοπής.

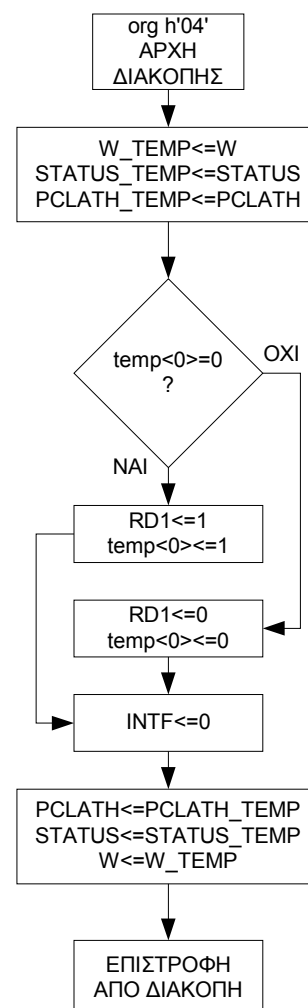
Στην παρούσα εργαστηριακή άσκηση θα γράψουμε ένα πρόγραμμα το οποίο ρυθμίζει τη διακοπή του RB0 να σηματοδοτείται σε κάθε κατερχόμενο μέτωπο παλμού. Η ρουτίνα διαχείρισης της διακοπής αλλάζει τη λογική κατάσταση της εξόδου RD1. Εάν είναι στο λογικό '1' τότε μετά τη διακοπή θα βρίσκεται στο λογικό '0' ενώ εάν είναι στο λογικό '0' τότε μετά τη διακοπή θα είναι στο λογικό '1'. Για την πραγματοποίηση της λειτουργίας αυτής χρησιμοποιείται ο καταχωρητής temp. Το λιγότερο σημαντικό του bit συγκρατεί την λογική κατάσταση του ακροδέκτη RD1. Έτσι για τον έλεγχο της

κατάστασης του ακροδέκτη RD1 ελέγχεται το bit 0 του καταχωρητή temp. Αυτό μας εξασφαλίζει ότι θα γίνει σωστή ανάγνωση της τιμής που ήδη έχουμε δώσει στο latch εξόδου του συγκεκριμένου ακροδέκτη. Γενικά κάθε εγγραφή σε κάποια θύρα του PIC συνοδεύεται από μια ακολουθία ανάγνωσης – τροποποίησης – εγγραφής των δεδομένων στο latch εξόδου. Αντίθετα μια ανάγνωση από έναν ακροδέκτη του PIC16F887 εξετάζει τη λογική κατάσταση που υπάρχει τη συγκεκριμένη χρονική στιγμή.

Στην εικόνα 1 παρουσιάζεται το λογικό διάγραμμα της παρούσας εργατηριακής άσκησης. Το διάγραμμα α απεικονίζει το λογικό διάγραμμα του κυρίως προγράμματος της εφαρμογής ενώ το διάγραμμα β απεικονίζει το λογικό διάγραμμα του κώδικα που εκτελείται κατά τη σηματοδότηση της διακοπής.



Εικόνα 1α. Λογικό διάγραμμα προγράμματος.



Εικόνα 1β. Λογικό διάγραμμα κώδικα διακοπής.

Στο σημείο αυτό παρουσιάζονται και εξηγούνται πλήρως οι καταχωρητές που χρησιμοποιούνται για τη διαχείριση των διακοπών στον PIC16F887. Στον πίνακα 1

παρουσιάζεται ο καταχωρητής INTCON ενώ στους πίνακες 2 και 3 παρουσιάζονται οι καταχωρητές PIE1 και PIE2 οι οποίοι ενεργοποιούν οι απενεργοποιούν διακοπές από τα περιφερειακά του μικροελεγκτή. Τέλος στους πίνακες 4 και 5 παρουσιάζονται οι καταχωρητές PIR1 και PIR2 οι οποίοι συγκρατούν τις σημαίες των διακοπών.

Καταχωρητής INTCON

- bit 7 **GIE**: Bit ενεργοποίησης διακοπών
1= Ενεργοποίηση διακοπών
0= Απενεργοποίηση διακοπών
- bit 6 **PEIE**: Bit ενεργοποίησης διακοπών από τα περιφερειακά κυκλώματα του PIC16F887
1= Ενεργοποίηση διακοπών που αιτούνται τα περιφερειακά
0= Απενεργοποίηση διακοπών που αιτούνται τα περιφερειακά
- bit 5 **TMR0IE**: Bit ενεργοποίησης διακοπών κατά την υπερχείλιση του TMR0
1= Ενεργοποίηση διακοπών TMR0
0= Απενεργοποίηση διακοπών TMR0
- bit 4 **INTE**: Bit ενεργοποίησης διακοπών κατά την εφαρμογή ανερχόμενου (ή κατερχόμενου εάν έχει ρυθμιστεί κατάλληλα) μετώπου στον ακροδέκτη RB0
1= Ενεργοποίηση διακοπών RB0
0= Απενεργοποίηση διακοπών RB0
- bit 3 **RBIE**: Bit ενεργοποίησης διακοπών κατά τη μετάβαση της λογικής κατάστασης των τεσσάρων ακροδεκτών υψηλής τάξης της θύρα PORTB
1= Ενεργοποίηση διακοπών
0= Απενεργοποίηση διακοπών
- bit 2 **TMR0IF**: Σημαία σηματοδότησης διακοπών υπερχείλισης TMR0
1= Ο TMR0 έχει υπερχείλσει (πρέπει να μηδενιστεί στο λογισμικό)
0= Ο TMR0 δεν έχει υπερχείλσει
- bit 1 **INTF**: Σημαία σηματοδότησης διακοπών ανερχόμενου (ή κατερχόμενου εάν έχει ρυθμιστεί κατάλληλα) μετώπου στον ακροδέκτη RB0
1= Ανερχόμενο (ή κατερχόμενο) μέτωπο εντοπίστηκε στον ακροδέκτη RB0 (πρέπει να μηδενιστεί στο λογισμικό)
0= Δεν εντοπίστηκε ανερχόμενο (ή κατερχόμενο) μέτωπο στον ακροδέκτη RB0
- bit 0 **RBIF**: Σημαία σηματοδότησης διακοπών κατά τη μετάβαση της λογικής κατάστασης των τεσσάρων ακροδεκτών υψηλής τάξης της θύρα PORTB
1= Τουλάχιστον ένας από τους ακροδέκτες RB7, RB6, RB5, RB4 έχει αλλάξει λογική κατάσταση. Το bit αυτό πρέπει να μηδενιστεί στο λογισμικό αφού πρώτα γίνει ανάγνωση της θύρας PORTB
0= Κανένας από τους ακροδέκτες RB7, RB6, RB5 και RB4 δεν έχει αλλάξει λογική κατάσταση

Πίνακας 1

Καταχωρητής PIE1 (Διεύθυνση h'8C' της Bank1)

- bit 7 **PSPIE**: Bit ενεργοποίησης διακοπών παράλληλης θύρας
1= Ενεργοποίηση διακοπών παράλληλης θύρας
0= Απενεργοποίηση διακοπών παράλληλης θύρας
- bit 6 **ADIE**: Bit ενεργοποίησης διακοπών από τον μετατροπέα αναλογικού σήματος σε ψηφιακό (Analog to Digital Converter)
1= Ενεργοποίηση διακοπών ADC
0= Απενεργοποίηση διακοπών ADC
- bit 5 **RCIE**: Bit ενεργοποίησης διακοπών κατά τη λήψη δεδομένων από τη μονάδα USART
1= Ενεργοποίηση διακοπών κατά τη λήψη δεδομένων από τη μονάδα USART
0= Απενεργοποίηση διακοπών κατά τη λήψη δεδομένων από τη μονάδα USART
- bit 4 **TXIE**: Bit ενεργοποίησης διακοπών κατά την αποστολή δεδομένων από τη μονάδα USART
1= Ενεργοποίηση διακοπών κατά την αποστολή δεδομένων από τη μονάδα USART
0= Απενεργοποίηση διακοπών κατά την αποστολή δεδομένων από τη μονάδα USART
- bit 3 **SSPIE**: Bit ενεργοποίησης διακοπών της σύγχρονης σειριακής θύρας (SSP)
1= Ενεργοποίηση διακοπών της σύγχρονης σειριακής θύρας
0= Απενεργοποίηση διακοπών της σύγχρονης σειριακής θύρας
- bit 2 **CCP1IE**: Bit ενεργοποίησης διακοπών του περιφερειακού κυκλώματος Capture/Compare/PWM1
1= Ενεργοποίηση διακοπών CCP1
0= Απενεργοποίηση διακοπών CCP1
- bit 1 **TMR2IE**: Bit ενεργοποίησης διακοπών υπερχείλισης του TMR2
1= Ενεργοποίηση διακοπών υπερχείλισης του TMR2
0= Απενεργοποίηση διακοπών υπερχείλισης του TMR2
- bit 0 **TMR1IE**: Bit ενεργοποίησης διακοπών υπερχείλισης του TMR1
1= Ενεργοποίηση διακοπών υπερχείλισης του TMR1
0= Απενεργοποίηση διακοπών υπερχείλισης του TMR1

Πίνακας 2

Καταχωρητής PIR1 (Διεύθυνση h'0C' της Bank0)

- bit 7 **PSPIF**: Σημαία σηματοδότησης διακοπών παράλληλης θύρας
1= Μια λειτουργία ανάγνωσης ή εγγραφής έχει πραγματοποιηθεί στην παράλληλη θύρα
0= Δεν έχει συμβεί ανάγνωση ή εγγραφή στην παράλληλη θύρα
- bit 6 **ADIF**: Σημαία σηματοδότησης διακοπών από τον μετατροπέα αναλογικού σήματος σε ψηφιακό (Analog to Digital Converter)
1= Μια μετατροπή αναλογικού σήματος σε ψηφιακό έχει ολοκληρωθεί
0= Η μετατροπή αναλογικού σήματος σε ψηφιακό δεν έχει ολοκληρωθεί
- bit 5 **RCIF**: Σημαία σηματοδότησης διακοπών κατά τη λήψη δεδομένων από τη μονάδα USART
1= Ο καταχωρητής λήψης δεδομένων από τη μονάδα USART περιέχει δεδομένα προς ανάγνωση
0= Ο καταχωρητής λήψης δεδομένων από τη μονάδα USART είναι άδειος
- bit 4 **TXIE**: Σημαία σηματοδότησης διακοπών κατά την αποστολή δεδομένων από τη μονάδα USART
1= Ο καταχωρητής αποστολής δεδομένων της μονάδας USART είναι άδειος
0= Ο καταχωρητής αποστολής δεδομένων της μονάδας USART περιέχει δεδομένα προς αποστολή
- bit 3 **SSPIF**: Σημαία σηματοδότησης διακοπών της σύγχρονης σειριακής θύρας (SSP)
1= Οι συνθήκες που θέτουν στο λογικό '1' τη σημαία της σύγχρονης σειριακής θύρας είναι:
- SPI
 - Πραγματοποιήθηκε αποστολή ή λήψη δεδομένων
 - slave I2C
 - Πραγματοποιήθηκε αποστολή ή λήψη δεδομένων
 - master I2C
 - Πραγματοποιήθηκε αποστολή ή λήψη δεδομένων
 - Η διαδικασία START ολοκληρώθηκε από το κύκλωμα SSP
 - Η διαδικασία STOP ολοκληρώθηκε από το κύκλωμα SSP
 - Η διαδικασία Restart ολοκληρώθηκε από το κύκλωμα SSP
 - Η διαδικασία επιβεβαίωσης λήψης (Acknowledge) ολοκληρώθηκε από το κύκλωμα
- 0= Δεν λαμβάνει χώρα συνθήκη που να θέτει στο λογικό '1' τη σημαία της σύγχρονης σειριακής θύρας
- bit 2 **CCP1IF**: Σημαία σηματοδότησης διακοπών περιφερειακού κυκλώματος Capture/Compare/PWM1
- Capture Mode:
1= Μια σύλληψη της τιμής του TMR1 έχει συμβεί (πρέπει να μηδενιστεί στο λογισμικό)
0= Δεν έχει συμβεί σύλληψη της τιμής του TMR1
- Compare Mode:
1= Μια επιτυχής σύγκριση ισότητας έχει συμβεί με τον TMR1 (πρέπει να μηδενιστεί στο λογισμικό)
0= Δεν έχει συμβεί επιτυχής σύγκριση ισότητας με τον TMR1
- PWM Mode:
Δεν χρησιμοποιείται σε αυτόν τον τρόπο λειτουργίας
- bit 1 **TMR2IF**: Σημαία σηματοδότησης διακοπών του TMR2
1= Ο TMR2 συγκρινόμενος με τον καταχωρητή PR2 έχουν προκύψει ίσοι (πρέπει να μηδενιστεί στο λογισμικό)
0= Δεν έχει προκύψει ισότητα ανάμεσα στον TMR2 και τον PR2
- bit 0 **TMR1IF**: Σημαία σηματοδότησης διακοπών υπερχείλισης του TMR1
1= Ο TMR1 έχει υπερχείλισει (πρέπει να μηδενιστεί στο λογισμικό)
0= Ο TMR1 δεν έχει υπερχείλισει

Πίνακας 3

Καταχωρητής PIE2 (Διεύθυνση h'8D' της Bank1)

bit 7	Δεν είναι υλοποιημένο
bit 6	CMIE : Bit ενεργοποίησης διακοπών που αιτούνται οι συγκριτές 1= Ενεργοποίηση διακοπών των συγκριτών 0= Απενεργοποίηση διακοπών των συγκριτών
bit 5	Δεν είναι υλοποιημένο
bit 4	EEIE : Bit ενεργοποίησης διακοπών κατά την εγγραφή στην EEPROM 1= Ενεργοποίηση διακοπών 0= Απενεργοποίηση διακοπών
bit 3	BCLIE : Bit ενεργοποίησης διακοπών κατά την ταυτόχρονη αποστολή δεδομένων σε δίαυλο I ² C 1= Ενεργοποίηση διακοπών 0= Απενεργοποίηση διακοπών
bit 2-1	Δεν είναι υλοποιημένο
bit 0	CCP2IE : Bit ενεργοποίησης διακοπών περιφερειακού κυκλώματος Capture/Compare/PWM2 1= Ενεργοποίηση διακοπών 0= Απενεργοποίηση διακοπών

Πίνακας 4

Καταχωρητής PIR2 (Διεύθυνση h'0D' της Bank0)

bit 7	Δεν είναι υλοποιημένο
bit 6	CMIF : Σημαία σηματοδότησης διακοπών των συγκριτές 1= Έχει συμβεί αλλαγή στην έξοδο κάποιου συγκριτή 0= Δεν έχει συμβεί αλλαγή στην έξοδο όλων των συγκριτών
bit 5	Δεν είναι υλοποιημένο
bit 4	EEIF : Σημαία σηματοδότησης διακοπών κατά την εγγραφή στην EEPROM 1= Η διαδικασία εγγραφής έχει ολοκληρωθεί (πρέπει να μηδενιστεί στο λογισμικό) 0= Η διαδικασία εγγραφής δεν έχει ολοκληρωθεί ή δεν έχει αρχίσει ακόμα
bit 3	BCLIF : Σημαία σηματοδότησης διακοπών κατά την ταυτόχρονη αποστολή δεδομένων σε δίαυλο I ² C 1= Συνέβει ταυτόχρονη αποστολή δεδομένων πάνω στο δίαυλο I ² C 0= Δεν έχει συμβεί ταυτόχρονη αποστολή δεδομένων πάνω στο δίαυλο I ² C
bit 2-1	Δεν είναι υλοποιημένα
bit 0	CCP2IF : Σημαία σηματοδότησης διακοπών του περιφερειακού κυκλώματος Capture/Compare/PWM2 <u>Capture Mode</u> : 1= Μια σύλληψη της τιμής του TMR1 έχει συμβεί (πρέπει να μηδενιστεί στο λογισμικό) 0= Δεν έχει συμβεί σύλληψη της τιμής του TMR1 <u>Compare Mode</u> : 1= Μια επιτυχής σύγκριση ισότητας έχει συμβεί με τον TMR1 (πρέπει να μηδενιστεί στο λογισμικό) 0= Δεν έχει συμβεί επιτυχής σύγκριση ισότητας με τον TMR1 <u>PWM Mode</u> : Δεν χρησιμοποιείται σε αυτόν τον τρόπο λειτουργίας

Πίνακας 5

2. Προσομοίωση

- α) Εκκινείτε το λογισμικό ανάπτυξης εφαρμογών MPLAB IDE. Δημιουργείτε μια νέα εφαρμογή βασισμένη στον PIC16F887 και αποθηκεύστε τη στο δίσκο του υπολογιστή.
β) Δημιουργείτε ένα νέο αρχείο στον κειμενογράφο και αντιγράψτε τον κώδικα που ακολουθεί:

```
include <p16f887.inc>

W_TEMP      equ h'20'
STATUS_TEMP equ h'21'
PCLATH_TEMP equ h'22'
W_TEMP1     equ h'A0'
W_TEMP2     equ h'120'
W_TEMP3     equ h'1A0'
temp        equ h'23'

org h'0000'      ; Οδηγία προς τον assembler ανύσματος RESET
nop
goto START      ; Άλμα στη διεύθυνση με ετικέτα START

org h'0004'      ; Οδηγία προς τον assembler ανύσματος διακοπής
movwf W_TEMP     ; Αποθήκευση του w στον προσωρινό καταχωρητή W_TEMP
swapf STATUS,w   ; Αποθήκευση του STATUS στον w με αντιστροφή του nibble
clrf STATUS      ; Μηδενισμός του STATUS ώστε να δείχνει στη Bank0
movwf STATUS_TEMP ; Αποθήκευση του w στον STATUS_TEMP της Bank0
movf PCLATH,w     ; Αποθήκευση του PCLATH στον w
movwf PCLATH_TEMP ; Αποθήκευση του w στον PCLATH_TEMP της Bank0

; Interrupt Service Routine from here
btfsc temp,0     ; Παράκαμψη επόμενης εντολής αν το bit 0 του temp είναι '0'
goto CLEAR_IT    ; Άλμα στη διεύθυνση με ετικέτα CLEAR_IT
SET_IT           ; Ετικέτα SET_IT
bsf PORTD,1      ; Καθορισμός της εξόδου RD1 στο λογικό '1'
bsf temp,0       ; Καθορισμός του bit 0 του καταχωρητή temp στο '1'
goto CONTINUE_ISR; Άλμα στην εντολή με ετικέτα CONTINUE_ISR
CLEAR_IT         ; Διεύθυνση με ετικέτα CLEAR_IT
bcf PORTD,1      ; Καθορισμός της εξόδου RD1 στο λογικό '0'
bcf temp,0       ; Καθορισμός του bit 0 του καταχωρητή temp στο '0'
CONTINUE_ISR     ; Ετικέτα CONTINUE_ISR
bcf INTCON,INTF  ; Μηδενισμός σημαίας διακοπής RB0

movf PCLATH_TEMP,w ; Αποθήκευση του PCLATH_TEMP στον w
movwf PCLATH      ; Αποθήκευση του w στον PCLATH
swapf STATUS_TEMP,w ; Επαναφορά της τιμής του STATUS_TEMP στον w
movwf STATUS      ; Αποθήκευση στον STATUS
swapf W_TEMP,f    ; Αντιστροφή των bits χαμηλής τάξης με τα bits υψηλής
swapf W_TEMP,w    ; τάξης του W_TEMP και αποθήκευση στον w
```

```
retfie          ; Επιστροφή από τη διακοπή
; Interrupt Service Routine till here

START           ; Διεύθυνση με ετικέτα START
  clrf temp     ; Μηδενισμός του καταχωρητή temp
  bcf PORTD,1   ; Καθορισμός της εξόδου RD1 στο λογικό '0'
  bsf STATUS,RP0 ; Ρύθμιση τα Bank bits να δείχνουν στην bank1
  bcf TRISD,1   ; Καθορισμός του RD1 ως ψηφιακή έξοδος
  bsf STATUS,RP1 ; Ρύθμιση στα Bank bits να δείχνουν στη Bank3
  bcf ANSELH,4  ; Καθορισμός του RB0 ως ψηφιακή είσοδος
  bcf STATUS,RP1 ; Ρύθμιση στα Bank bits να δείχνουν στη Bank1
  bsf TRISB,0   ; Καθορισμός ακροδέκτη RB0 ως ψηφιακή είσοδος
  bcf OPTION_REG,INTEDG ; Καθορισμός διακοπής RB0 σε κάθε κατερχόμενο μέτωπο
  bcf STATUS,RP0 ; Ρύθμιση στα Bank bits να δείχνουν στη Bank0
  bsf INTCON,INTE ; Ενεργοποίηση διακοπών του ακροδέκτη RB0
  bcf INTCON,INTF ; Μηδενισμός σημαίας διακοπών του RB0
  bsf INTCON,GIE ; Ενεργοποίηση διακοπών

goto $          ; Άλμα στην ίδια εντολή

end
```

γ) Να μεταγλωττίσετε την εφαρμογή και να διορθώσετε όποια συντακτικά λάθη υπάρχουν. Να εμφανίσετε το παράθυρο παρακολούθησης μεταβλητών και να προσθέσετε τους καταχωρητές W_TEMP, STATUS_TEMP, PCLATH_TEMP και temp.

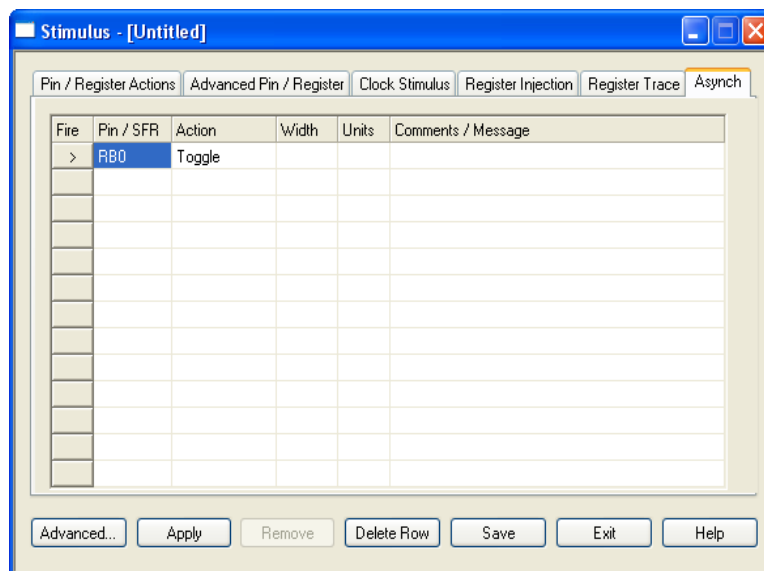
δ) Ενεργοποιείτε τον simulator του MPLAB και με την επιλογή Debugger>Stimulus>New Workbook να εμφανίσετε το παράθυρο διέγερσης. Στο παράθυρο αυτό μπορούν να προσομοιωθούν ψηφιακά σήματα εισόδου στους ακροδέκτες του μικροελεγκτή.

ε) Στο παραπάνω παράθυρο (Stimulus) ανοίξετε την καρτέλα Asynch. Στο drop down μενού της στήλης PIN/SFR επιλέξτε τον ακροδέκτη RB0. Στο drop down μενού της στήλης Action επιλέξτε Toggle. Το παράθυρο Stimulus θα πρέπει να είναι όπως παρουσιάζεται στην εικόνα 2.

στ) Εκτελέστε το πρόγραμμα στον simulator μέχρι ο έλεγχος του προγράμματος να φτάσει στο τέλος (εντολή goto \$). Σε ποια λογική κατάσταση βρίσκεται ο ακροδέκτης RD1 του μικροελεγκτή;

ζ) Παρατηρείστε την τιμή του ακροδέκτη RB0 όσο ο κώδικας του προγράμματος βρίσκεται στο σημείο αυτό. Θα πρέπει να βρίσκεται στη λογική κατάσταση '0'. Στη συνέχεια πατήστε ΜΙΑ φορά πάνω στο σύμβολο '>'. Εξακολουθήστε να εκτελείτε το πρόγραμμα εντολή προς εντολή πατώντας το πλήκτρο F7. Παρατηρείστε την τιμή του ακροδέκτη RB0. Τώρα θα πρέπει να δείχνει πως βρίσκεται στο λογικό '1'.

η) Ξαναπατήστε το σύμβολο '>' στο παράθυρο διέγερσης. Τώρα θα έχουμε μια αντιστροφή του ψηφιακού σήματος εισόδου στον ακροδέκτη RB0 από το λογικό '1' στο λογικό '0'. Αυτό σημαίνει πως υπάρχει κατερχόμενο μέτωπο άρα και σηματοδότηση διακοπής. Συνεχίστε την εκτέλεση του προγράμματος πατώντας το πλήκτρο F7. Παρατηρήστε ότι αρχίζουν να εκτελούνται οι εντολές που βρίσκονται στο τμήμα κώδικα από τη διεύθυνση h'0004' και μετά.



Εικόνα 2. Το παράθυρο διέγερσης.

θ) Μετά την εκτέλεση της εντολής `retfie` σε ποιο σημείο του προγράμματος επιστρέφει ο μικροελεγκτής; Ποια είναι η λογική κατάσταση τώρα στον ακροδέκτη RD1;
 ι) Πατήστε στο παράθυρο διέγερσης δύο συνεχόμενες φορές το πλήκτρο '>' ώστε να προσομοιωθεί ένα ανερχόμενο και στη συνέχεια ένα κατερχόμενο μέτωπο παλμού. Συνεχίστε την εκτέλεση του προγράμματος εντολή προς εντολή με το πλήκτρο F7. Ποια θα είναι η νέα λογική κατάσταση του ακροδέκτη RD1;

Καταχωρητής	Πριν τη διακοπή	Όταν εκτελεστεί η εντολή <code>bcf INTCON,INTF</code>	Μετά τη διακοπή
W			
STATUS			
PCLATH			
W_TEMP			
STATUS_TEMP			
PCLATH_TEMP			

Πίνακας 6

ια) Πραγματοποιήστε RESET στο μικροελεγκτή. Επαναλάβετε την προσομοίωση από την αρχή και συμπληρώστε τις τιμές των καταχωρητών W, STATUS, W_TEMP και

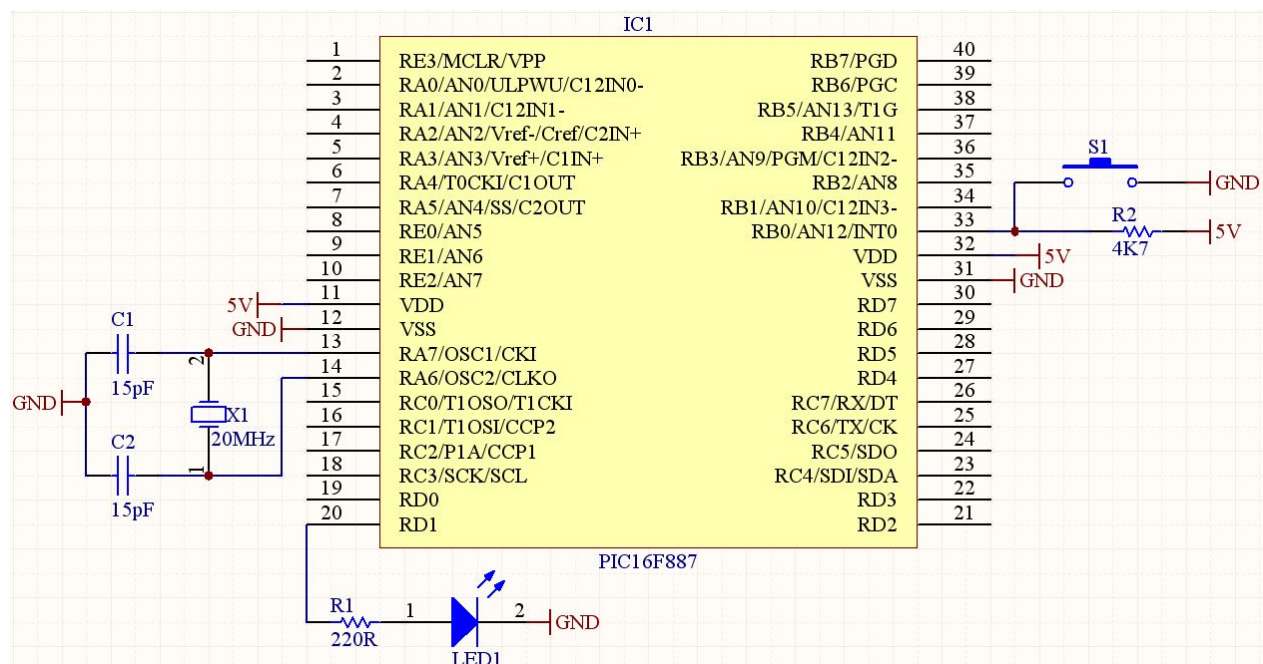
STATUS_TEMP στον πίνακα 6 για τα 3 χρονικά σημεία που αναφέρεται (πριν τη διακοπή, μετά την ολοκλήρωση του κώδικα της διακοπής, μετά τη διακοπή). Ποιες στήλες πρέπει να είναι απολύτως ίδιες;

3. Υλοποίηση

Για την υλοποίηση του κυκλώματος της εργαστηριακής άσκησης απαιτούνται τα παρακάτω υλικά:

IC1	=	PIC16F887
R1	=	220R, $\geq 1/8W$
LED1	=	Δίοδος LED
C1	=	15pF, $\geq 6V$
C2	=	15pF, $\geq 6V$
X1	=	Κρύσταλος 20MHz
S1	=	Διακόπτης τύπου push button
R2	=	4K7, $\geq 1/8W$

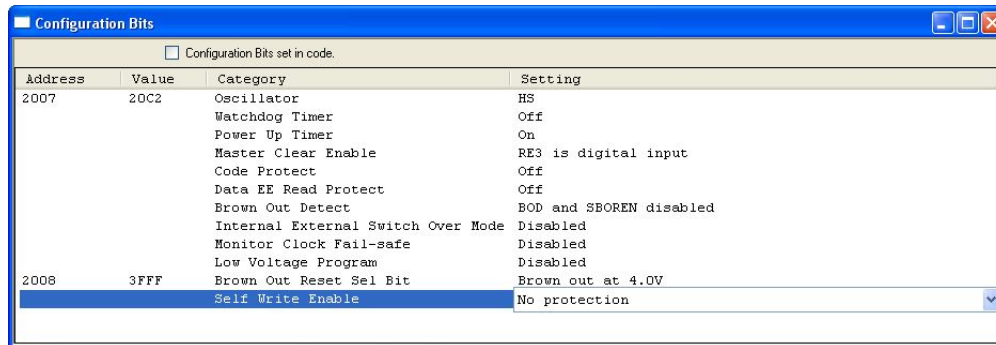
α) Υλοποιήστε σε πειραματική πλακέτα το κύκλωμα της εικόνας 3.



Εικόνα 3. Το κύκλωμα της εργαστηριακής άσκησης 5.

β) Διαμορφώστε τα Configuration Bits όπως φαίνονται στην εικόνα 4.

γ) Μεταγλωττίστε, διορθώστε συντακτικά λάθη και προγραμματίστε το μικροελεγκτή με το πρόγραμμα που προσομοιώθηκε στο προηγούμενο τμήμα της άσκησης.



Εικόνα 4. Configuration Bits.

δ) Εφαρμόστε την τροφοδοσία και παρατηρήστε την κατάσταση της LED. Είναι αναμμένη; Πατώντας μία φορά το διακόπτη τύπου push button S1 θα πρέπει να έχουμε αντιστροφή της κατάστασης της LED. Ανταποκρίνεται αυτό και στην πραγματικότητα; Δοκιμάστε αρκετές φορές να πατήσετε το διακόπτη S1. Θα πρέπει να παρατηρήσετε ότι μερικές φορές έχουμε αντιστροφή ενώ κάποιες άλλες όχι. Δώστε μια λογική εξήγηση γιατί συμβαίνει αυτό.

ε) Συνδέστε ένα παλμογράφο στο RB0 και στο RD1 του PIC16F887. Πατήστε το διακόπτη S1 και σχεδιάστε το παλμογράφημα κατά την αλλαγή κατάστασης στην είσοδο RB0. Θα πρέπει να παρατηρήσετε ότι κάθε φορά που πατάμε το διακόπτη (ή που τον αφήνουμε) στον ακροδέκτη RB0 αντί να έχουμε ένα κατερχόμενο ή ανερχόμενο μέτωπο, έχουμε δεκάδες ανερχόμενα και κατερχόμενα. Αυτό οφείλεται στο ότι ο διακόπτης που χρησιμοποιούμε λόγω του είναι είναι μηχανικός, παρουσιάζει αναπηδήσεις. Έτσι σε λίγα msec μπορεί να δώσει δεκάδες ή και εκατοντάδες ανερχόμενα και κατερχόμενα μέτωπα με ένα μόνο πάτημα έως να σταθεροποιηθεί σε χαμηλό ή υψηλό επίπεδο τάσης.

στ) Να δηλώσετε στη Bank0 τους καταχωρητές DelayReg1 και DelayReg2 όπως φαίνεται παρακάτω:

```
DelayReg1    equ h'24'  
DelayReg2    equ h'25'
```

Να προσθέσετε τον παρακάτω κώδικα στην αρχή της ρουτίνας εξυπηρέτησης διακοπής:

```
clrf DelayReg1    ; Μηδενισμός του καταχωρητή DelayReg1  
clrf DelayReg2    ; Μηδενισμός του καταχωρητή DelayReg2  
DELAY             ; Διεύθυνση με ετικέτα DELAY  
decfsz DelayReg1,f ; Μείωση DelayReg1 κατά 1, παράκαμψη εντολής αν 0  
goto $-1          ; Άλλα στην προηγούμενη εντολή  
decfsz DelayReg2,f ; Μείωση DelayReg2 κατά 1, παράκαμψη εντολής αν 0  
goto DELAY        ; Άλλα στην εντολή με ετικέτα DELAY  
btfsc PORTB,0     ; Παράκαμψη επόμενης εντολής αν RB0=0
```

`goto CONTINUE_ISR;` Άλλα στην εντολή με ετικέτα `CONTINUE_ISR`

Το παραπάνω τμήμα κώδικα δημιουργεί μια καθυστέρηση μέσα στη ρουτίνα διαχείρισης διακοπής του μικροελεγκτή για 40msec. Μετά την καθυστέρηση ελέγχεται ο ακροδέκτης RB0. Εάν βρίσκεται στο λογικό '1' τότε επιστρέφουμε από τη διακοπή χωρίς να εκτελεστεί καμία άλλη λειτουργία. Εάν βρίσκεται στο λογικό '0' τότε αντιστρέφεται η λογική κατάσταση της εξόδου RD1.

ζ) Μεταγλωττίστε και φορτώστε ξανά το πρόγραμμα στο μικροελεγκτή. Εφαρμόστε την τροφοδοσία. Πατήστε το διακόπτη S1 διαδοχικά αρκετές φορές. Τώρα σε κάθε πάτημα έχουμε αντιστροφή της κατάστασης του RD1; Γιατί η προσθήκη της παραπάνω καθυστέρησης διόρθωσε το πρόβλημα των αναπηδήσεων; Δώστε την απάντησή σας.

4. Εργασίες

α) Να γράψετε ένα πρόγραμμα για τον PIC16F887 και χρησιμοποιώντας τον Timer0 να δημιουργήσετε μια διακοπή στον μικροελεγκτή η οποία θα σηματοδοτείται περίπου κάθε 1msec. Να θεωρηθεί ταλαντωτής χρονισμού στα 20MHz.

4. Προχωρημένα θέματα

α) Χρησιμοποιώντας τον Timer2 να γράψετε ένα πρόγραμμα στο οποίο ο μικροελεγκτής δημιουργεί ένα σήμα χρονισμού ακριβείας ενός δευτερολέπτου. Να θεωρήσετε ταλαντωτή χρονισμού στα 160KHz.

β) Χρησιμοποιώντας τον Timer1 και έναν κρύσταλλο 32.768KHz να γράψετε ένα πρόγραμμα στο οποίο ο μικροελεγκτής δημιουργεί με χρήση διακοπών ένα σήμα ακριβείας ενός δευτερολέπτου.

Ο ΜΕΤΑΤΡΟΠΕΑΣ ΑΝΑΛΟΓΙΚΟΥ ΣΗΜΑΤΟΣ ΣΕ ΨΗΦΙΑΚΟ

1. Θεωρία

Για την επεξεργασία σημάτων που προέρχονται από αναλογικές πηγές ο PIC16F887 διαθέτει έναν 10-bit μετατροπέα αναλογικού σήματος σε ψηφιακό (Analog to Digital Converter) με δεκατέσσερα κανάλια. Τα επίπεδα τάσεων που μπορεί να διαχειριστεί ρυθμίζονται μέσα από το λογισμικό του μικροελεγκτή και κυμαίνονται από 0V έως 5V. Επιπλέον υπάρχει η δυνατότητα να χρησιμοποιηθούν επίπεδα τάσεων που καθορίζονται από τους ακροδέκτες RA2 και RA3.

Τα δεκατέσσερα κανάλια του PIC16F887 πολυπλέκονται με πέντε από τους ακροδέκτες της θύρας PORTA (RA0, RA1, RA2, RA3 και RA5), έξι από τους ακροδέκτες της θύρας PORTB (RB0, RB1, RB2, RB3, RB4 και RB5) και τρεις από τους ακροδέκτες της θύρας PORTE (RE0, RE1 και RE2). Το αποτέλεσμα της μετατροπής αποθηκεύεται στους καταχωρητές ADRESH:ADRESL. Ο μικροελεγκτής χρησιμοποιεί δύο καταχωρητές για τη ρύθμιση του ADC. Ο ADCON0 ρυθμίζει τη λειτουργία του ADC και ο ADCON1 καθορίζει τη λειτουργία των ακροδεκτών του PIC που αποτελούν αναλογικές εισόδους. Επιπλέον χρησιμοποιούνται άλλοι δύο καταχωρητές, οι οποίοι είναι ο ANSEL και ο ANSELH, για τον καθορισμό ενός ακροδέκτη ως αναλογική είσοδος.

Ο χρόνος μετατροπής για κάθε bit ονομάζεται T_{AD} . Ο μετατροπέας του μικροελεγκτή απαιτεί για τη σωστή λειτουργία του ρολόι το οποίο καθορίζει τιμή του χρόνου T_{AD} μεγαλύτερη του 1.6μsec.

Μετά τον καθορισμό του αναλογικού καναλιού που θα χρησιμοποιηθεί για μια μετατροπή θα πρέπει να παρέλθει ένα χρονικό διάστημα (acquisition time) πριν ξεκινήσει η μετατροπή. Το χρονικό διάστημα απαιτείται ώστε να αποκτήσει την εφαρμοζόμενη τάση προς μετατροπή το εσωτερικό κύκλωμα δειγματοληψίας και συγκράτησης του μικροελεγκτή. Μια τυπική τιμή του διαστήματος αυτού για τον PIC16F887 είναι 4.67μsec.

Κάθε αναλογική τάση που εφαρμόζεται στο μικροελεγκτή για μετατροπή συνδέεται στο αναλογικό κανάλι μέσω μιας αντίστασης μέγιστης τιμής 10K.

Αναλυτικά τα βήματα που πρέπει να ακολουθηθούν για τη ρύθμιση και τη χρησιμοποίηση του A/D Converter του PIC16F887 είναι:

- Ρύθμιση των καναλιών που θα χρησιμοποιηθούν καθορίζοντας στο λογικό '1' τα αντίστοιχα bit των καταχωρητών ANSEL και ANSELH καθώς επίσης και τα αντίστοιχα bit των καταχωρητών TRISA, TRISB και TRISE.
- Επιλογή ρολογιού για τον A/D ρυθμίζοντας κατάλληλα τα bit ADCS2, ADCS1, ADCS0 του καταχωρητή ADCON0
- Καθορισμός των τάσεων αναφοράς ρυθμίζοντας κατάλληλα τα bit VCFG1 και VCFG0 του καταχωρητή ADCON1
- Καθορισμός του καναλιού το οποίο θα εφαρμοστεί στο εσωτερικό κύκλωμα δειγματοληψίας και συγκράτησης του μικροελεγκτή ρυθμίζοντας κατάλληλα τα bit CHS3, CHS2, CHS1 και CHS0 του καταχωρητή ADCON0
- Ρύθμιση του bit ADFM του καταχωρητή ADCON1 για δεξιά ή αριστερή στοίχιση του αποτελέσματος της μετατροπής στους καταχωρητές ADRESH:ADRESL
- Ενεργοποίηση του κυκλώματος του μετατροπέα θέτοντας στο λογικό '1' το bit ADON του καταχωρητή ADCON0
- Ενεργοποίηση των διακοπών εάν είναι επιθυμητό
- Αναμονή μέχρι να παρέλθει το χρονικό διάστημα acquisition time
- Εκκίνηση της μετατροπής θέτοντας στο λογικό '1' το bit GO/DONE του καταχωρητή ADCON0
- Αναμονή μέχρι να ολοκληρωθεί η μετατροπή εξετάζοντας πότε θα μηδενιστεί το bit GO/DONE
- Ανάγνωση των καταχωρητών ADRESH:ADRESL για επεξεργασία του αποτελέσματος

Ο PIC16F887 μπορεί να ψηφιοποιήσει τάσεις σε περιοχές από 0V έως 5V. Αν οι τάσεις που θέλουμε να διαβάσουμε ανήκουν σε διαφορετικό εύρος τιμών τότε είναι απαραίτητη η προσαρμογή τους στα αποδεκτά επίπεδα με τη χρήση τελεστικών ενισχυτών.

Το αποτέλεσμα μιας 10-bit μετατροπής είναι ένας αριθμός από το 0 έως το 1023 ($=2^{10}-1$). Χρησιμοποιώντας ως τάσεις αναφοράς 0V και 5V ο ADC θα μπορεί να ξεχωρίσει διαφορετικές τάσεις κατά 4.8mV ($=5V/1024$). Στη συντριπτική πλειοψηφία των εφαρμογών με ADC αυτό χαρακτηριστικό καλύπτει τους σχεδιαστές. Άλλωστε κοντά σε αυτή την τιμή καθορίζονται και τα επίπεδα του ηλεκτρονικού θορύβου. Η σχέση που δίνει το αποτέλεσμα της μετατροπής είναι η εξής:

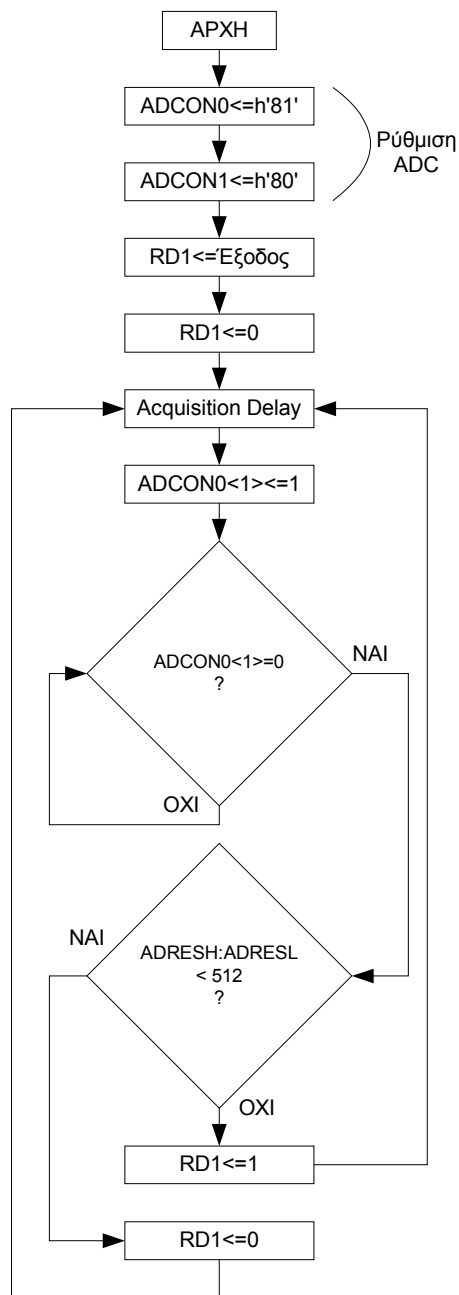
$$[ADRESH]:[ADRESL] = \frac{AV_{in}}{5} \cdot 1023 \quad (1)$$

, όπου AV_{in} είναι η αναλογική τάση που εφαρμόζεται στον ADC του μικροελεγκτή.

Το ακέραιο μέρος του αποτελέσματος αποθηκεύεται στους καταχωρητές ADRESH και ADRESL του μικροελεγκτή. Για τον προγραμματιστή όμως ο άγνωστος όρος είναι η αναλογική τάση που εφαρμόζεται και αυτή θα πρέπει να υπολογίσει. Έτσι σε μια μετατροπή ο προγραμματιστής καλείται συνήθως να πραγματοποιήσει τις πράξεις της παρακάτω σχέσης:

$$AV_{in} = \frac{[ADRESH]:[ADRESL]}{1023} \cdot 5 \quad (2)$$

Σε αρκετές περιπτώσεις η διαίρεση με το 1023 αντικαθίσταται προσεγγιστικά με το 1024 η οποία μπορεί εύκολα να πραγματοποιηθεί με μετατοπίσεις του διαιρέτη προς τα δεξιά.



Εικόνα 1. Το λογικό διάγραμμα του προγράμματος.

Στην παρούσα εργαστηριακή άσκηση θα αναπτύξουμε μια εφαρμογή η οποία χρησιμοποιεί τον ADC του μικροελεγκτή, ψηφιοποιεί μια αναλογική τάση που εφαρμόζεται στον ακροδέκτη RA0 και εάν η τάση αυτή είναι πάνω από 2.5V οδηγεί στο λογικό '1' τον ακροδέκτη RD1 του PIC16F887 ενώ αν είναι κάτω από 2.5V τον οδηγεί στο λογικό '0'. Το λογικό διάγραμμα του προγράμματος παρουσιάζεται στην εικόνα 1.

Στο σημείο αυτό παρουσιάζουμε τους καταχωρητές ειδικής χρήσης του PIC16F887 με τους οποίους διαχειριζόμαστε το μετατροπέα αναλογικού σήματος σε ψηφιακό.

Καταχωρητής ADCON0 (διεύθυνση h'1F' της Bank0)

- bit 7-6 **ADCS1:ADCS0**: Bits επιλογής ρολογιού μετατροπέα A/D
00 = $F_{osc}/2$
01 = $F_{osc}/8$
10 = $F_{osc}/32$
11 = FRC (το ρολόι λαμβάνεται από εσωτερικό RC ταλαντωτή μέγιστης συχνότητας 500KHz)
- bit 5-2 **CHS2:CHS0**: Bits επιλογής αναλογικού καναλιού
0000 = Κανάλι 0 (AN0/RA0)
0001 = Κανάλι 1 (AN1/RA1)
0010 = Κανάλι 2 (AN2/RA2)
0011 = Κανάλι 3 (AN3/RA3)
0100 = Κανάλι 4 (AN4/RA5)
0101 = Κανάλι 5 (AN5/RE0)
0110 = Κανάλι 6 (AN6/RE1)
0111 = Κανάλι 7 (AN7/RE2)
1000 = Κανάλι 8 (AN8/RB2)
1001 = Κανάλι 9 (AN9/RB3)
1010 = Κανάλι 10 (AN10/RB1)
1011 = Κανάλι 11 (AN11/RB4)
1100 = Κανάλι 12 (AN12/RB0)
1101 = Κανάλι 13 (AN13/RB5)
1110 = Τάση αναφοράς συγκριτή
1111 = Τάση αναφοράς 0.6V
- bit 2 **GO/DONE**: Bit κατάσταση A/D μετατροπής
1= A/D μετατροπή σε εξέλιξη (Θέτοντας το bit αυτό στο λογικό '1' ξεκινά η διαδικασία μετατροπής και όταν ολοκληρωθεί τότε μηδενίζεται αυτόματα)
0= Δεν υπάρχει μετατροπή A/D σε εξέλιξη
- bit 1 Δεν είναι υλοποιημένο
- bit 0 **ADON**: Bit ενεργοποίησης μετατροπέα A/D
1= Ο μετατροπέας A/D είναι ενεργοποιημένος
0= Ο μετατροπέας A/D είναι απενεργοποιημένος

Πίνακας 1

Καταχωρητής ADCON1 (διεύθυνση h'9F' της Bank1)

bit 7	ADFM: Bit επιλογής μορφής αποτελέσματος 1 = Το 10-bit αποτέλεσμα της μετατροπής θα είναι στοιχισμένο δεξιά στο ζεύγος καταχωρητών ADRESH:ADRESL. Τα πιο σημαντικά bits του ADRESH διαβάζονται ως '0' 0 = Το 10-bit αποτέλεσμα της μετατροπής θα είναι στοιχισμένο αριστερά στο ζεύγος καταχωρητών ADRESH:ADRESL. Τα λιγότερο σημαντικά bits του ADRESL διαβάζονται ως '0'
bit 6	Δεν είναι υλοποιημένο
bit 5	VCFG1: Bit επιλογής χαμηλής τάσης αναφοράς 1 = Η χαμηλή τάση αναφοράς λαμβάνεται από τον ακροδέκτη V_{REF-} 0 = Η χαμηλή τάση αναφοράς λαμβάνεται από τη γείωση V_{SS}
bit 4	VCFG1: Bit επιλογής υψηλής τάσης αναφοράς 1 = Η υψηλή τάση αναφοράς λαμβάνεται από τον ακροδέκτη V_{REF+} 0 = Η υψηλή τάση αναφοράς λαμβάνεται από την τροφοδοσία V_{DD}
bit 3-0	Δεν είναι υλοποιημένα

Πίνακας 2

Καταχωρητής ANSEL (διεύθυνση h'188' της Bank3)

bit 7	ANS7: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RE2/AN7 1 = Ο ακροδέκτης RE2/AN7 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RE2/AN7 είναι ψηφιακή είσοδος/έξοδος
bit 6	ANS6: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RE1/AN6 1 = Ο ακροδέκτης RE1/AN6 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RE1/AN6 είναι ψηφιακή είσοδος/έξοδος
bit 5	ANS5: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RE0/AN5 1 = Ο ακροδέκτης RE0/AN5 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RE0/AN5 είναι ψηφιακή είσοδος/έξοδος
bit 4	ANS4: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RA5/AN4 1 = Ο ακροδέκτης RA5/AN4 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RA5/AN4 είναι ψηφιακή είσοδος/έξοδος
bit 3	ANS3: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RA3/AN3 1 = Ο ακροδέκτης RA3/AN3 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RA3/AN3 είναι ψηφιακή είσοδος/έξοδος
bit 2	ANS2: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RA2/AN2 1 = Ο ακροδέκτης RA2/AN2 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RA2/AN2 είναι ψηφιακή είσοδος/έξοδος
bit 1	ANS1: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RA1/AN1 1 = Ο ακροδέκτης RA1/AN1 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RA1/AN1 είναι ψηφιακή είσοδος/έξοδος
bit 0	ANS0: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RA0/AN0 1 = Ο ακροδέκτης RA0/AN0 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RA0/AN0 είναι ψηφιακή είσοδος/έξοδος

Πίνακας 3

Καταχωρητής ANSELH (διεύθυνση h'189' της Bank3)

bit 7-6	Δεν είναι υλοποιημένα
bit 5	ANS13: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RB5/AN13

	1 = Ο ακροδέκτης RB5/AN13 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RB5/AN13 είναι ψηφιακή είσοδος/έξοδος
bit 4	ANS12: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RB0/AN12 1 = Ο ακροδέκτης RB0/AN12 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RB0/AN12 είναι ψηφιακή είσοδος/έξοδος
bit 3	ANS11: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RB4/AN11 1 = Ο ακροδέκτης RB4/AN11 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RB4/AN11 είναι ψηφιακή είσοδος/έξοδος
bit 2	ANS10: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RB1/AN10 1 = Ο ακροδέκτης RB1/AN10 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RB1/AN10 είναι ψηφιακή είσοδος/έξοδος
bit 1	ANS9: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RB3/AN9 1 = Ο ακροδέκτης RB3/AN9 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RB3/AN9 είναι ψηφιακή είσοδος/έξοδος
bit 0	ANS8: Bits επιλογής αναλογικής ή ψηφιακής λειτουργίας στον ακροδέκτη RB2/AN8 1 = Ο ακροδέκτης RB2/AN8 είναι αναλογική είσοδος 0 = Ο ακροδέκτης RB2/AN8 είναι ψηφιακή είσοδος/έξοδος

Πίνακας 4

2. Προσομοίωση

α) Δημιουργήστε μια νέα εφαρμογή στο MPLAB βασισμένη στον PIC16F887 και αποθηκεύστε τη στο δίσκο του υπολογιστή.

β) Δημιουργήστε ένα νέο αρχείο πηγαίου κώδικα στον κειμενογράφο του MPLAB και αντιγράψτε τον κώδικα που ακολουθεί:

```
include <p16f887.inc>

DelayReg equ h'20'

org h'0000'
nop
movlw b'10000001' ; Φόρτωση αριθμητικού h'81' στον w
movwf ADCON0      ; Αποθήκευση στον ADCON0
bsf STATUS,RP0    ; Bank1
movlw b'10000000' ; Φόρτωση αριθμητικού h'80' στον w
movwf ADCON1      ; Αποθήκευση στον ADCON1
bcf TRISD,1       ; Καθορισμός RD1 ως έξοδος
bcf STATUS,RP0    ; Bank0
bcf PORTD,1       ; Καθορισμός RD1 στο λογικό '0'
LOOP              ; Διεύθυνση μνήμης με ετικέτα LOOP
movlw d'20'       ;
movwf DelayReg    ;
decfsz DelayReg,f ; Acquisition Delay = 12usec > 4,67usec
```

```
goto $-1 ;  
  
bsf ADCON0,GO ; Έναρξη ADC μετατροπής  
btfsc ADCON0,GO ; Έλεγχος εάν ολοκληρώθηκε η μετατροπή  
goto $-1 ; Άλλα στην προηγούμενη εντολή  
; Στο σημείο αυτό οι καταχωρητές ADRESH:ADRESL θα πρέπει να έχουν  
; το αποτέλεσμα της μετατροπής αναλογικού σήματος σε ψηφιακό  
  
movlw d'2' ; Φόρτωση αριθμητικού d'2' στον w  
subwf ADRESH,w ; Αφαίρεση του w από τον καταχωρητή ADRESH  
btfss STATUS,C ; Παράκαμψη επόμενης εντολής αν C=1  
bcf PORTD,1 ; Καθορισμός RD1 στο λογικό '0'  
btfsc STATUS,C ; Παράκαμψη επόμενης εντολής αν C=0  
bsf PORTD,1 ; Καθορισμός RD1 στο λογικό '1'  
goto LOOP ; Άλλα στη διεύθυνση με ετικέτα LOOP  
  
end
```

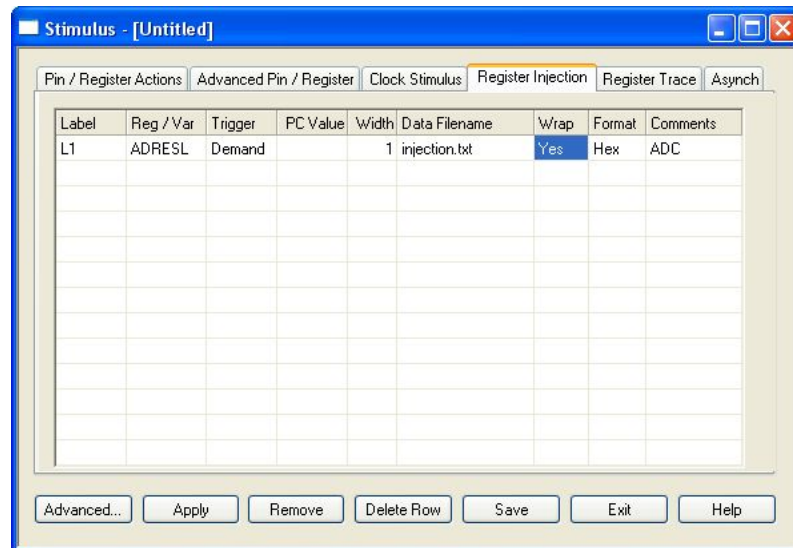
γ) Μεταγλωττίστε το πρόγραμμα, διορθώστε εάν υπάρχουν τα συντακτικά λάθη και ενεργοποιήστε τον simulator.

δ) Στο σημείο αυτό θα πρέπει να προσομοιώσουμε την μετατροπή αναλογικού σήματος σε ψηφιακό με τον simulator του MPLAB. Η προσομοίωση μπορεί να γίνει τοποθετώντας εμβόλιμα τιμές στους καταχωρητές ADRESH και ADREL τη χρονική στιγμή κατά την οποία θεωρητικά θα έχει ολοκληρωθεί μια μετατροπή. Αυτό συμβαίνει κάθε φορά που μεταβαίνει στο λογικό '1' το bit ADIF του καταχωρητή PIR1. Για να το επιτύχουμε αυτό εργαζόμαστε ως εξής:

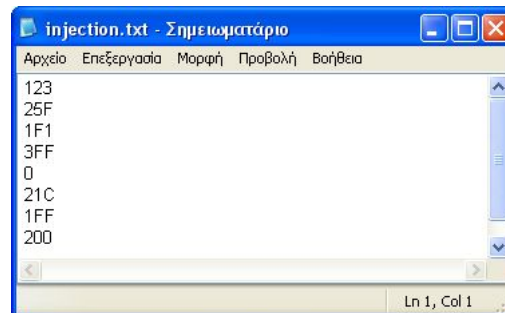
- Με την επιλογή *Debugger>Stimulus>New Workbook* εμφανίζουμε ένα νέο παράθυρο διέγερσης. Το παράθυρο αυτό παρουσιάζεται στην εικόνα 2.
- Μεταβαίνουμε στην καρτέλα *Register Injection*
- Κάνουμε κλικ στην πρώτη γραμμή του παραθύρου ώστε να εμφανιστούν οι αντίστοιχες επιλογές σε κάθε στήλη
- Στη στήλη *Label* αναγράφουμε ένα όνομα της επιλογής μας ώστε να ξεχωρίζουμε κάθε περίπτωση που πραγματοποιούμε φόρτωση τιμών εμβόλιμα σε κάποιο καταχωρητή (προαιρετικό)
- Στην αναδιπλούμενη λίστα της στήλης *Reg/Var* επιλέγουμε τον καταχωρητή ADRESL
- Στη συνέχεια πηγαίνουμε στη στήλη *Data Filename*. Εδώ θα πρέπει να εισάγουμε το όνομα αρχείου το οποίο περιλαμβάνει τις τιμές που θα φορτώνονται στους καταχωρητές ADRESH και ADRESL μετά από κάθε μετατροπή. Να δημιουργήσετε ένα αρχείο κειμένου με τις τιμές όπως αυτές παρουσιάζονται στην εικόνα 3.

Αποθηκεύστε αυτό το αρχείο με το όνομα *injection.txt*. Δηλώστε στη στήλη *Data Filename* το αρχείο με τις τιμές

- Βεβαιωθείτε ότι στις στήλες *Wrap* και *Format* είναι επιλεγμένες οι επιλογές *Yes* και *Hex* αντίστοιχα
- Γράψτε εάν είναι επιθυμητό κάποια σχόλια στη στήλη *Comments*
- Πατήστε το κουμπί «Apply» ώστε να εφαρμοστεί το παράθυρο διέγερσης στο τρέχον πρόγραμμα



Εικόνα 2. Η καρτέλα Register Injection στο παράθυρο διέγερσης.



Εικόνα 3. Το αρχείο με τις τιμές για τους καταχωρητές ADRESH:ADRESL.

Στο σημείο αυτό, εκτελώντας την προσομοίωση, κάθε φορά που πραγματοποιείται μια μετατροπή αναλογικού σήματος σε ψηφιακό, θα πρέπει οι καταχωρητές ADRESH και ADRESL να φορτώνονται διαδοχικά με τις τιμές που υπάρχουν στο αρχείο *injection.txt*.

ε) Εκτελέστε το πρόγραμμα μέχρι να πραγματοποιηθεί η πρώτη μετατροπή. Ελέγξτε ότι όντως φορτώνεται η πρώτη τιμή από το αρχείο διέγερσης στους καταχωρητές του μικροελεγκτή.

στ) Συνεχίστε την εκτέλεση του προγράμματος ώστε να εκτελεστεί οκτώ φορές το τμήμα κώδικα που βρίσκεται στην ανακύκλωση. Να συμπληρωθεί ο πίνακας 5.

Ανακύκλωση (A/A)	10-bit τιμή ADRESH:ADRESL	RD1 (1 ή 0)
1η		
2η		
3η		
4η		
5η		
6η		
7η		
8η		

Πίνακας 5

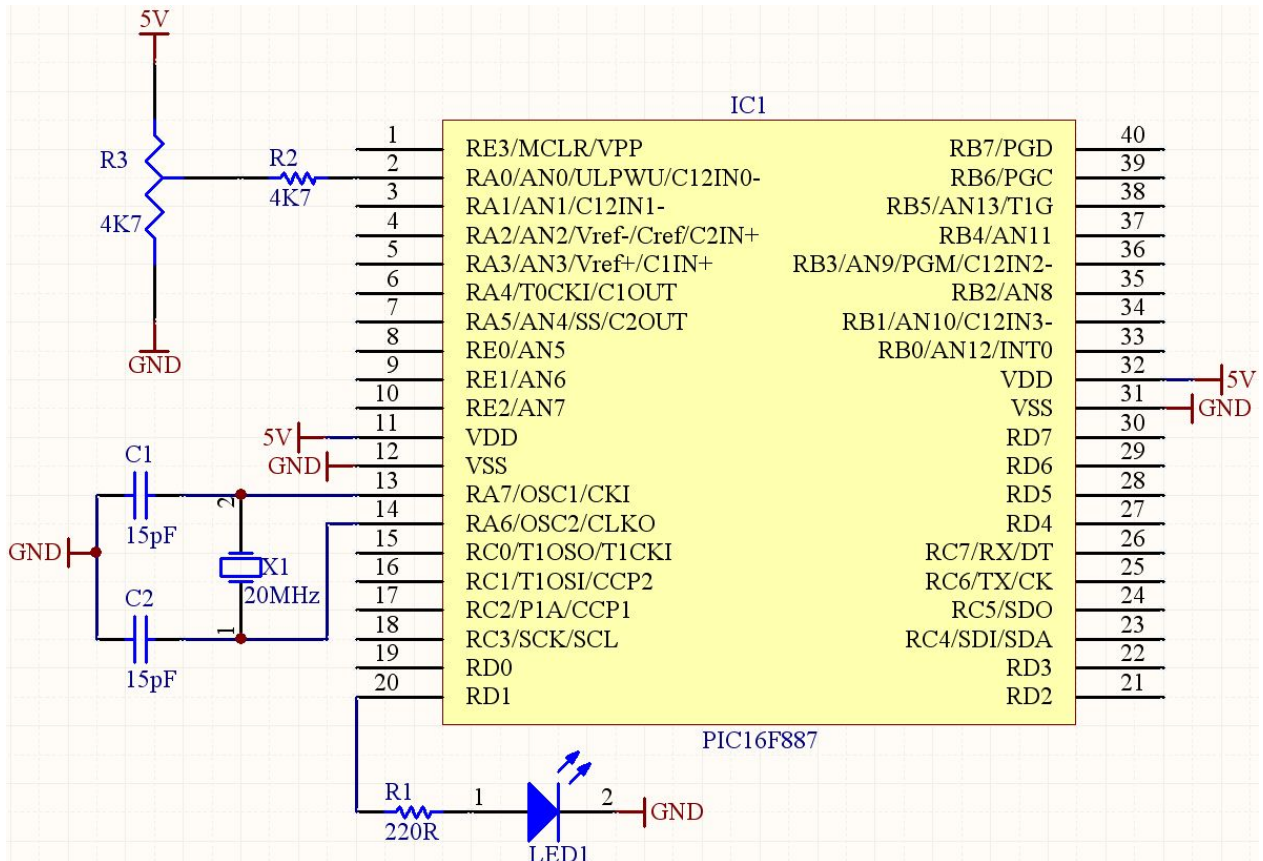
3. Υλοποίηση

Για την περάτωση της παρακάτω εργαστηριακής άσκησης θα χρειαστούν τα παρακάτω υλικά:

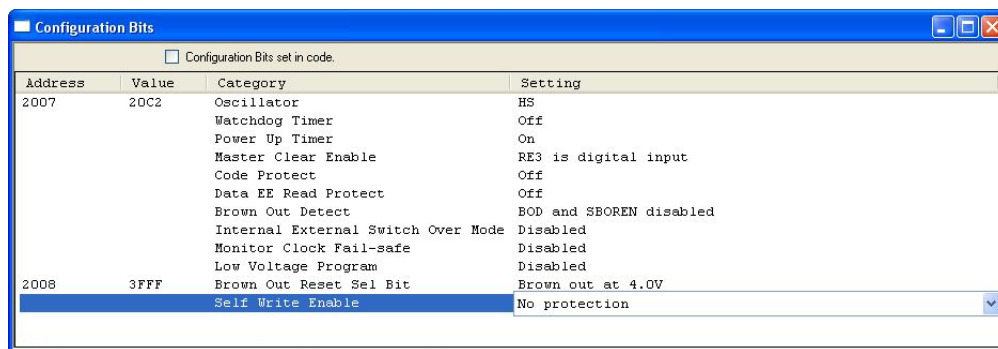
IC1	=	PIC16F887
R1	=	220R, 1/4W
R2	=	4K7, 1/4W
R3	=	τρίμερ 4K7, 1/4W
LED1	=	Δίοδος LED
C1	=	15pF
C2	=	15pF
X1	=	Κρύσταλος 20MHz

- α) Να υλοποιήσετε σε πειραματική πλακέτα το κύκλωμα της εικόνας 4.
- β) Να ρυθμίσετε τα Configuration bits όπως φαίνοντας στην εικόνα 5.
- γ) Προγραμματίστε τον PIC16F887 με το πρόγραμμα της προσομοίωσης και εφαρμόστε την τροφοδοσία.
- δ) Ρυθμίστε το τρίμερ ώστε να δείχνει τάση γύρω στο 1V. Σε ποια κατάσταση βρίσκεται η LED; Είναι αναμμένη; Είναι λογικό το αποτέλεσμα;
- ε) Ρυθμίστε το τρίμερ ώστε να δείχνει τάση γύρω στα 4V. Σε ποια κατάσταση βρίσκεται η LED; Είναι αναμμένη; Είναι λογικό το αποτέλεσμα;
- στ) Ρυθμίστε το τρίμερ σε μια τάση γύρω στα 2.5V ώστε να παρατηρηθεί το φαινόμενο να μειωθεί η ένταση του φωτός που εκπέμπει η LED. Συνδέστε έναν παλμογράφο στον

ακροδέκτη RD1. Σε ποια κατάσταση βρίσκεται; Τι παρατηρείτε; Είναι λογικό αυτό; Δώστε την απάντησή σας.



Εικόνα 4. Το κύκλωμα της εργαστηριακής άσκησης 6.



Εικόνα 5. Configuration Bits.

ζ) Αντικαταστήστε στον κώδικα του προγράμματος την εντολή `movlw d'2'` που είναι με γκριζο χρώμα με την εντολή `movlw d'3'`. Επαναπρογραμματίστε το μικροελεγκτή και εφαρμόστε την τροφοδοσία. Να μεταβάλλεται την τάση του τρίμερ που οδηγείται στον μικροελεγκτή από την ελάχιστη στη μέγιστη τιμή της. Σε ποια τιμή ανάβει η LED;

4. Εργασίες

- α) Τροποποιήστε το παραπάνω πρόγραμμα ώστε να ανάβει η δίοδος LED σε περίπτωση που η τάση στο τρίμερ υπερβεί τα 2V.
- β) Τροποποιήστε το παραπάνω πρόγραμμα ώστε η δίοδος να ανάβει όταν η τάση στο τρίμερ ανεβαίνει πάνω από 3V και να σβήνει όταν πέφτει κάτω από 2V.

4. Προχωρημένα θέματα

- α) Να υλοποιήσετε μια εφαρμογή στην οποία εφαρμόζονται δύο αναλογικές τάσεις στις εισόδους AN0 και AN1 του PIC16F887. Όταν η αναλογική τάση που εφαρμόζεται στην είσοδο AN0 είναι μεγαλύτερη από αυτή που εφαρμόζεται στην AN1 να ανάβει μια δίοδος LED. Όταν ισχύει το αντίστροφο η LED να είναι σβηστή.
- β) Να υλοποιήσετε μια εφαρμογή η οποία διαβάζει την ένδειξη από ένα αισθητήριο θερμοκρασίας LM35.

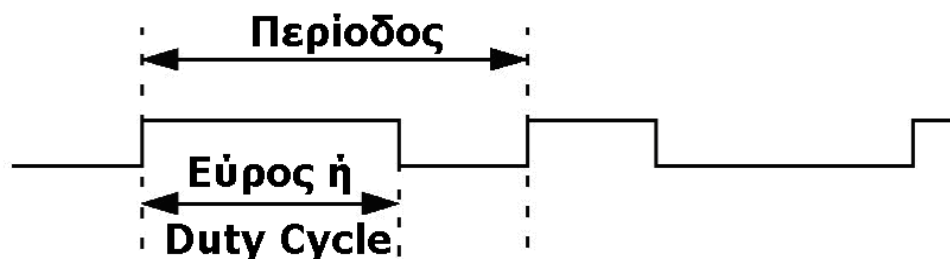
Η ΜΟΝΑΔΑ CAPTURE/COMPARE/PWM (CCP)

1. Θεωρία

Ο μικροελεγκτής PIC16F887 διαθέτει δύο ανεξάρτητες μονάδες Capture/Compare/PWM (CCP1 και CCP2). Οι μονάδες αυτές μπορούν να χρησιμοποιηθούν σε ένα από τα παρακάτω mode λειτουργίας:

- Capture mode
- Compare mode
- PWM mode

Στην παρούσα εργαστηριακή άσκηση θα ασχοληθούμε με την λειτουργία PWM οπότε και θα εξετάσουμε μόνο αυτή την περίπτωση. Στη λειτουργία PWM ο μικροελεγκτής παράγει στην έξοδο κυματομορφές διαμορφωμένες κατά εύρος (Pulse Width Modulation). Οι PWM κυματομορφές βρίσκουν εφαρμογή σε κυκλώματα τηλεπικοινωνιών, στην οδήγηση κινητήρων, στα παλμοτροφοδοτικά κλπ. Επιπλέον με τη μονάδα PWM και την εξωτερική χρήση ενός χαμηλοπερατού φίλτρου ο μικροελεγκτής αποκτά δυνατότητα μετατροπής ψηφιακού σήματος σε αναλογικό χωρίς να διαθέτει το ανάλογο κύκλωμα.



Εικόνα 1. Παράδειγμα PWM κυματομορφής.

Μια PWM κυματομορφή παρουσιάζεται στην εικόνα 1. Το εύρος του παλμού της κυματομορφής ονομάζεται Duty Cycle και μετριέται σε μονάδες χρόνου. Έχει επικρατήσει όμως να ορίζεται ως ο λόγος του χρονικού διαστήματος στο οποίο η κυματομορφή είναι σε υψηλό λογικό επίπεδο προς την περίοδο και συνεπώς αναφέρεται ως καθαρός αριθμός ή σε ποσοστό %.

Η μονάδα CCP1 μπορεί να δημιουργήσει PWM κυματομορφές στον ακροδέκτη RC2 ενώ η μονάδα CCP2 στον ακροδέκτη RC1. Η διαχείριση των μονάδων CCP

πραγματοποιείται από τους καταχωρητές ειδικής χρήσης CCPxCON και CCPRxL (x=1 ή 2). Η περίοδος της PWM κυματομορφής καθορίζεται από την παρακάτω εξίσωση:

$$T_{PWM} = (PR2 + 1) \cdot 4 \cdot T_{OSC} \cdot TMR2_{prescale_value} \quad (1)$$

Όπως είναι φανερό από την περίοδο, εκτός από τους καταχωρητές ειδικής χρήσης CCP1CON και CCP2CON η μονάδα PWM χρησιμοποιεί για τη λειτουργία της και τον Timer2. Γενικά να αναφέρουμε ότι ο Timer2 είναι ένας 8-bit timer ο οποίος μηδενίζεται όχι όταν υπερχειλίσει αλλά όταν γίνει ταύτιση της τιμής του με την τιμή του καταχωρητή PR2. Διαθέτει προηγούμενη βαθμίδα μετρητή (prescaler) και επόμενη βαθμίδα μετρητή (postscaler) και η ρύθμισή του πραγματοποιείται από τον καταχωρητή T2CON.

Η διακριτικότητα της PWM κυματομορφής είναι 10-bit που σημαίνει ότι το Duty Cycle μπορεί να αλλάζει περίπου κάθε 0.1% ανάμεσα σε δύο διαδοχικά βήματα. Το Duty Cycle εξαρτάται από την τιμή του καταχωρητή CCPRxL και τα bit DCxB1 και DCxB0 του καταχωρητή CCPxCON. Πιο συγκεκριμένα καθορίζεται από την παρακάτω σχέση:

$$CCPRxL : CCPxCON < 5,4 > \cdot T_{OSC} \cdot TMR2_{prescale_value} \quad (2)$$

x=1 ή x=2

Ως λόγος του εύρους παλμού προς την περίοδο της κυματομορφής το Duty Cycle ορίζεται ως εξής:

$$Duty_Cycle\% = 100 \cdot \frac{t_{duty_cycle}}{T_{PWM}} \% = 100 \cdot \frac{CCPRxL : CCPxCON < 5,4 >}{(PR2 + 1) \cdot 4} \% \quad (3)$$

x=1 ή x=2

Για τη ρύθμιση των μονάδων CCP του PIC16F887 ακολουθούμε τα εξής βήματα:

- Ρυθμίζουμε τα bit T2CKPS1 και T2CKPS0 του καταχωρητή T2CON και φορτώνουμε τον καταχωρητή PR2 με την κατάλληλη τιμή για τον καθορισμό της περιόδου της PWM κυματομορφής
- Ρυθμίζουμε κατάλληλα τα bit DCxB1 και DCxB0 του καταχωρητή CCPxCON και φορτώνουμε στον καταχωρητή CCPRxL την κατάλληλη τιμή για τον καθορισμό του Duty Cycle της PWM κυματομορφής
- Ρυθμίζουμε τους ακροδέκτες RC2 (για το CCP1) και RC1 (για το CCP2) ως εξόδους μηδενίζοντας τα bit 1 και 2 του καταχωρητή TRISC
- Ενεργοποιούμε τον Timer2 καθορίζοντας στο λογικό '1' το bit TMR2ON

- Ρυθμίζουμε τον καταχωρητή CCPxCON για λειτουργία PWM καθορίζοντας τα bit CCPxM3 και CCPx2 στο λογικό '1'
- Ειδικά για τη μονάδα CCP1 οδηγούμε στο λογικό '0' τα bit P1M1 και P1M0 του καταχωρητή CCP1CON

(x=1 ή 2)

Το τελευταίο βήμα ειδικά για τη μονάδα CCP1 το ακολουθούμε λόγω του ότι η μονάδα αυτή έχει την δυνατότητα για μία λειτουργία πιο εξελιγμένης μορφής. Συγκεκριμένα μπορεί να δημιουργήσει κυματομορφές διαμορφωμένες κατά εύρος χρησιμοποιώντας περισσότερους από έναν ακροδέκτες με τέτοιο τρόπο ώστε να οδηγεί κινητήρες σε συνδεσμολογία μισής γέφυρας (half bridge) ή πλήρους γέφυρας (full bridge).

Στο σημείο αυτό παρουσιάζουμε τους καταχωρητές που χρησιμοποιούνται για τη ρύθμιση των μονάδων CCP. Στον πίνακα 1 παρουσιάζεται ο καταχωρητής T2CON ενώ στους πίνακες 2 και 3 παρουσιάζονται οι καταχωρητές CCP1CON και CCP2CON.

Καταχωρητής T2CON (διεύθυνση h'12' της Bank0)

bit 7	Δεν είναι υλοποιημένο
bit 6-3	TOUTPS<3:0> : Bit επιλογής κλίμακας επόμενης βαθμίδας μέτρησης (Postscaler) 0000 = 1:1 Postscaler 0001 = 1:2 Postscaler 0010 = 1:3 Postscaler 0011 = 1:4 Postscaler 0100 = 1:5 Postscaler 0101 = 1:6 Postscaler 0110 = 1:7 Postscaler 0111 = 1:8 Postscaler 1000 = 1:9 Postscaler 1001 = 1:10 Postscaler 1010 = 1:11 Postscaler 1011 = 1:12 Postscaler 1100 = 1:13 Postscaler 1101 = 1:14 Postscaler 1110 = 1:15 Postscaler 1111 = 1:16 Postscaler
bit 2	TMR2ON : Bit ενεργοποίησης Timer2 1 = Ενεργοποίηση Timer2 0 = Απενεργοποίηση Timer2
bit 1-0	T2CKPS<1:0> : Bit επιλογής κλίμακας προηγούμενης βαθμίδας μέτρησης (Prescaler) 00 = 1:1 Prescaler 01 = 1:4 Prescaler 1x = 1:16 Prescaler

Πίνακας 1

Καταχωρητής CCP1CON (διεύθυνση h'17' της Bank0)

- bit 7-6 **P1M<1:0>**: Bit ρυθμίσεων της PWM εξόδου
Εάν CCP1M<3:2> = 00, 01, 10:
xx = Ο ακροδέκτης P1A ορίζεται ως είσοδος Capture/Compare. Οι ακροδέκτες P1B, P1C και P1D ορίζονται ως είσοδοι/έξοδοι των θυρών στις οποίες ανήκουν
Εάν CCP1M<3:2> = 11:
00 = Ο ακροδέκτης P1A ορίζεται ως απλή έξοδος PWM. Οι ακροδέκτες P1B, P1C και P1D ορίζονται ως είσοδοι/έξοδοι των θυρών στις οποίες ανήκουν
01 = Οδήγηση εμπρόσθιας κατεύθυνσης κινητήρα συνδεσμολογίας πλήρους γέφυρας. Έξοδος P1D διαμορφωμένη κατά εύρος. Έξοδος P1A ενεργός. Έξοδος P1B ανενεργός. Έξοδος P1C ανενεργός
10 = Οδήγηση κινητήρα συνδεσμολογίας μισής γέφυρας. Έξοδοι P1A και P1B διαμορφωμένοι κατά εύρος. Έξοδοι P1C και P1D ορίζονται ως είσοδοι/έξοδοι των θυρών στις οποίες ανήκουν
11 = Οδήγηση αντίστροφης κατεύθυνσης κινητήρα συνδεσμολογίας πλήρους γέφυρας. Έξοδος P1B διαμορφωμένη κατά εύρος. Έξοδος P1C ενεργός. Έξοδος P1A ανενεργός. Έξοδος P1D ανενεργός
- bit 5-4 **DC1B<1:0>**: Τα λιγότερο σημαντικά bit του Duty Cycle της PWM κυματομορφής
Capture mode:
Δεν χρησιμοποιείται σε αυτό το mode λειτουργίας
Compare mode:
Δεν χρησιμοποιείται σε αυτό το mode λειτουργίας
PWM mode:
Τα δύο λιγότερο σημαντικά bits του Duty Cycle της PWM κυματομορφής
- bit 3-0 **CCP1M<3:0>**: Bit επιλογής mode λειτουργίας
0000 = Απενεργοποίηση μονάδας CCP
0001 = Δε χρησιμοποιείται (δεσμευμένη)
0010 = Compare mode, αντιστροφή εξόδου σε κάθε ταύτιση
0011 = Δε χρησιμοποιείται (δεσμευμένη)
0100 = Capture mode, σύλληψη σε κάθε κατερχόμενο μέτωπο
0101 = Capture mode, σύλληψη σε κάθε ανερχόμενο μέτωπο
0110 = Capture mode, σύλληψη σε κάθε 4ο ανερχόμενο μέτωπο
0111 = Capture mode, σύλληψη σε κάθε 16ο ανερχόμενο μέτωπο
1000 = Compare mode, έξοδος στο λογικό '1' σε κάθε ταύτιση
1001 = Compare mode, έξοδος στο λογικό '0' σε κάθε ταύτιση
1010 = Compare mode, έξοδος ανεπηρέαστη, σηματοδότηση διακοπής σε κάθε ταύτιση
1011 = Compare mode, σκανδαλισμός ειδικού συμβάντος προκαλώντας RESET στον Timer1 και κάνοντας εκκίνηση μιας A/D μετατροπής
1100 = PWM mode, για τα P1A, P1C η ενεργός κατάσταση περιγράφεται με υψηλό επίπεδο, για τα P1B, P1D η ενεργός κατάσταση περιγράφεται με υψηλό επίπεδο.
1101 = PWM mode, για τα P1A, P1C η ενεργός κατάσταση περιγράφεται με υψηλό επίπεδο, για τα P1B, P1D η ενεργός κατάσταση περιγράφεται με χαμηλό επίπεδο.
1110 = PWM mode, για τα P1A, P1C η ενεργός κατάσταση περιγράφεται με χαμηλό επίπεδο, για τα P1B, P1D η ενεργός κατάσταση περιγράφεται με υψηλό επίπεδο.
1101 = PWM mode, για τα P1A, P1C η ενεργός κατάσταση περιγράφεται με χαμηλό επίπεδο, για τα P1B, P1D η ενεργός κατάσταση περιγράφεται με χαμηλό επίπεδο.

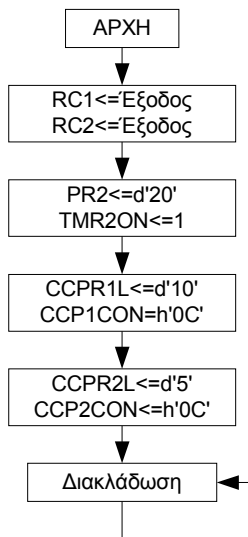
Πίνακας 2

Καταχωρητής CCP2CON (διεύθυνση h'1D' της Bank0)

bit 7-6	Δεν είναι υλοποιημένα
bit 5-4	DC2B<1:0> : Τα λιγότερο σημαντικά bit του Duty Cycle της PWM κυματομορφής <u>Capture mode</u> : Δεν χρησιμοποιείται σε αυτό το mode λειτουργίας <u>Compare mode</u> : Δεν χρησιμοποιείται σε αυτό το mode λειτουργίας <u>PWM mode</u> : Τα δύο λιγότερο σημαντικά bits του Duty Cycle της PWM κυματομορφής
bit 3-0	CCP2M<3:0> : Bit επιλογής mode λειτουργίας 0000 = Απενεργοποίηση μονάδας CCP 0001 = Δε χρησιμοποιείται (δεσμευμένη) 0010 = Δε χρησιμοποιείται (δεσμευμένη) 0011 = Δε χρησιμοποιείται (δεσμευμένη) 0100 = Capture mode, σύλληψη σε κάθε κατερχόμενο μέτωπο 0101 = Capture mode, σύλληψη σε κάθε ανερχόμενο μέτωπο 0110 = Capture mode, σύλληψη σε κάθε 4ο ανερχόμενο μέτωπο 0111 = Capture mode, σύλληψη σε κάθε 16ο ανερχόμενο μέτωπο 1000 = Compare mode, έξοδος στο λογικό '1' σε κάθε ταύτιση 1001 = Compare mode, έξοδος στο λογικό '0' σε κάθε ταύτιση 1010 = Compare mode, έξοδος ανεπηρέαστη, σηματοδότηση διακοπής σε κάθε ταύτιση 1011 = Compare mode, σκανδαλισμός ειδικού συμβάντος προκαλώντας RESET στον Timer1 και κάνοντας εκκίνηση μιας A/D μετατροπής 11xx = PWM mode

Πίνακας 3

Στην παρούσα εργαστηριακή άσκηση θα δημιουργήσουμε PWM κυματομορφές στους ακροδέκτες RC1 και RC2 του PIC16F887. Το λογικό διάγραμμα του προγράμματος παρουσιάζεται στην εικόνα 2.



Εικόνα 2. Το λογικό διάγραμμα του προγράμματος.

2. Προσομοίωση

- α) Δημιουργήστε μια νέα εφαρμογή στο MPLAB βασισμένη στον PIC16F887 και αποθηκεύστε τη στο δίσκο του υπολογιστή.
- β) Δημιουργήστε ένα νέο αρχείο πηγαίου κώδικα στον κειμενογράφο του MPLAB και αντιγράψτε τον κώδικα που ακολουθεί:

```
include <p16f887.inc>

org h'0000'
nop

bsf STATUS,RP0      ; Bank1
bcf TRISC,2         ; Το RC2 καθορίζεται ως έξοδος
bcf TRISC,1         ; Το RC1 καθορίζεται ως έξοδος
movlw d'20'
movwf PR2           ; Φόρτιση στον PR2 το 20
bcf STATUS,RP0      ; Bank0
bsf T2CON,TMR2ON    ; Ενεργοποίηση Timer2
movlw d'10'
movwf CCPR1L        ; Καθορισμός του Duty Cycle στο RC2
movlw b'00001100'
movwf CCP1CON       ; Ρύθμιση CCP1 για PWM
movlw d'5'
movwf CCPR2L        ; Καθορισμός του Duty Cycle στο RC1
movlw b'00001100'
movwf CCP2CON       ; Ρύθμιση CCP2 για PWM
```

```
goto $
```

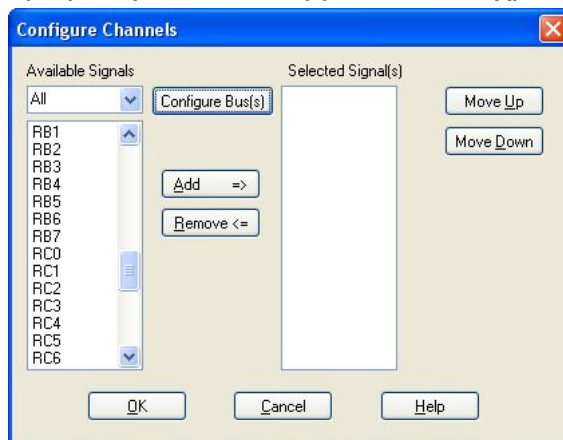
```
end
```

γ) Μεταγλωττίστε το πρόγραμμα, διορθώστε, εάν υπάρχουν, συντακτικά λάθη και ενεργοποιήστε τον simulator.

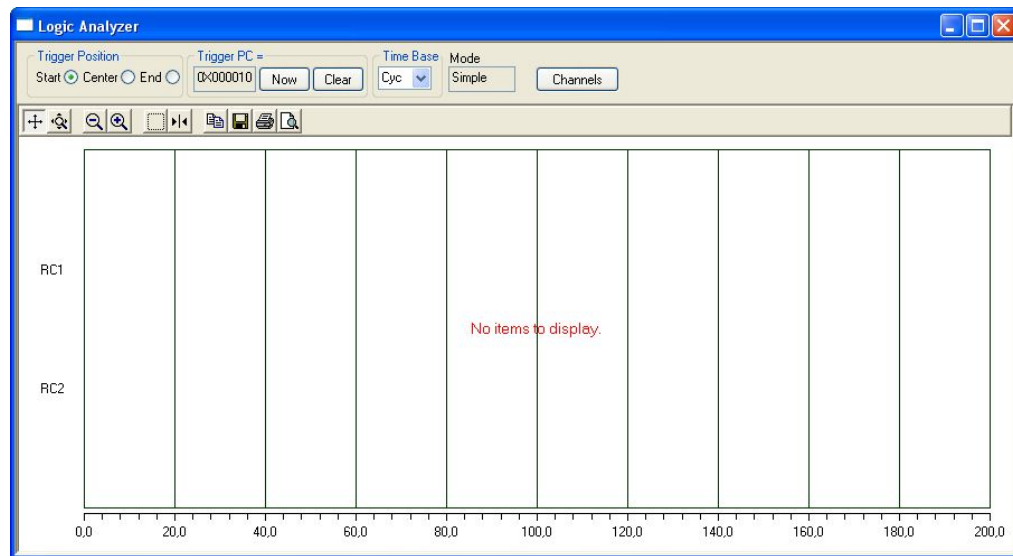
δ) Με την επιλογή View>Simulator Logic Analyzer εμφανίστε το παράθυρο του Λογικού Αναλυτή που διαθέτει ο simulator του MPLAB.

ε) Στο παράθυρο του Λογικού Αναλυτή πατήστε το κουμπί *Channels*. Θα εμφανιστεί το παράθυρο της εικόνας 3. Από τη λίστα στο αριστερό τμήμα του παραθύρου επιλέξτε τους ακροδέκτες RC1 και RC2 και πατήστε το κουμπί Add =>. Τέλος πατήστε το κουμπί OK.

στ) Το παράθυρο του Λογικού αναλυτή θα πρέπει να είναι αυτή τη στιγμή όπως παρουσιάζεται στην εικόνα 4. Στον κατακόρυφο άξονα περιλαμβάνει τους ακροδέκτες RC1 και RC2. Ο οριζόντιος άξονας αποτελεί τη μονάδα του χρόνου σε κύκλους ρολογιού.



Εικόνα 3. Προσθήκη καναλιών στο Λογικό Αναλυτή.



Εικόνα 4. Ο Λογικός Αναλυτής του MPLAB.

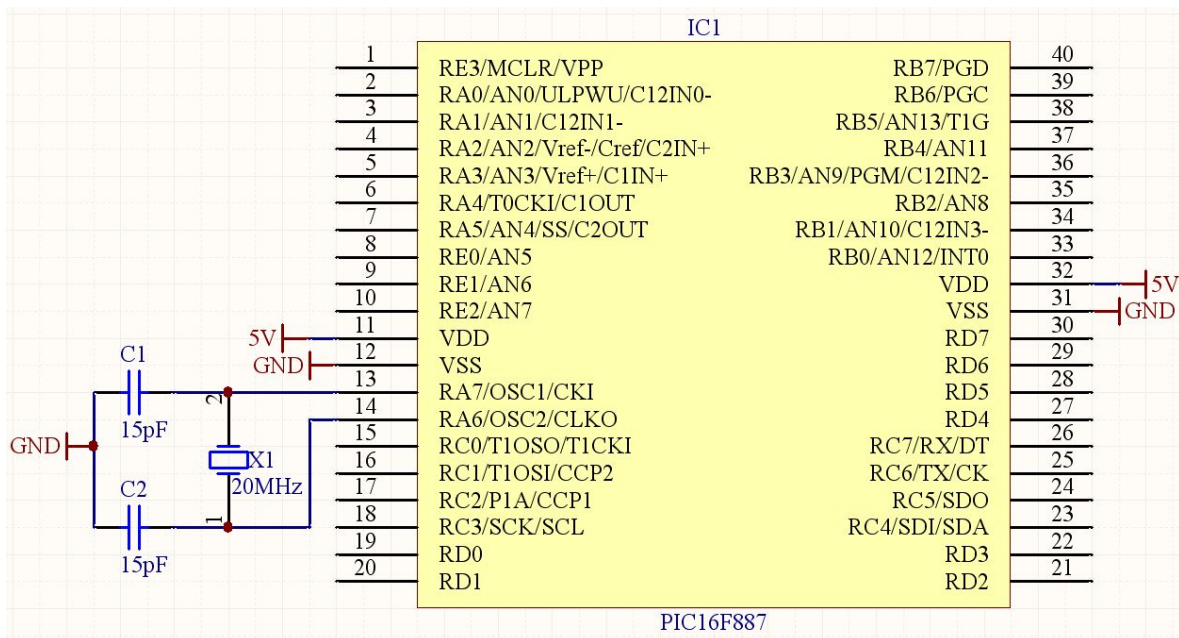
- ζ) Εκτελέστε εντολή προς εντολή το πρόγραμμα και παρατηρήστε από το Λογικό Αναλυτή τη λογική κατάσταση στους ακροδέκτες RC1 και RC2.
- στ) Υπολογίστε κατά προσέγγιση από το Λογικό Αναλυτή το Duty Cycle των δύο PWM κυματομορφών.
- η) Τροποποιήστε κατάλληλα τον κώδικα και δημιουργήστε PWM κυματομορφές ίδιας συχνότητας αλλά με Duty Cycle 75% και 87%. Εκτελέστε από την αρχή το πρόγραμμα και επιβεβαιώστε με το Λογικό Αναλυτή το αποτέλεσμα.
- θ) Πώς μπορούμε να επιτύχουμε, στην παραπάνω συχνότητα των PWM κυματομορφών, Duty Cycle 100%;

3. Υλοποίηση

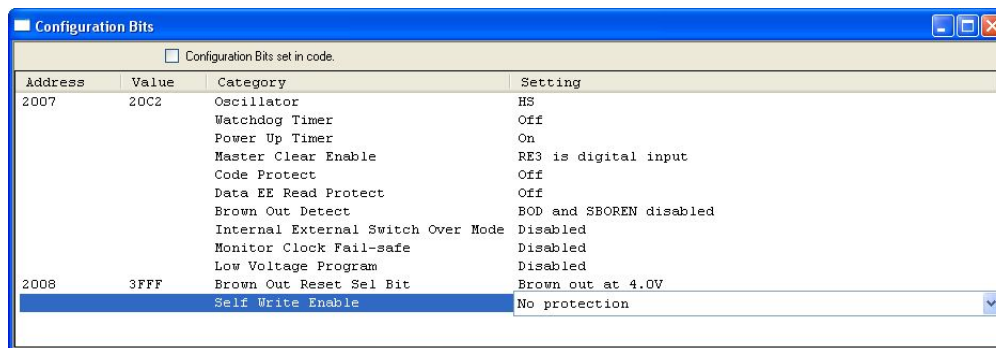
Για την περάτωση της παρούσας εργαστηριακής άσκησης θα χρειαστούν τα παρακάτω υλικά:

IC1 = PIC16F887
C1 = 15pF
C2 = 15pF
X1 = Κρύσταλος 20MHz

- α) Υλοποιήστε σε πειραματική πλακέτα το κύκλωμα της εικόνας 5.

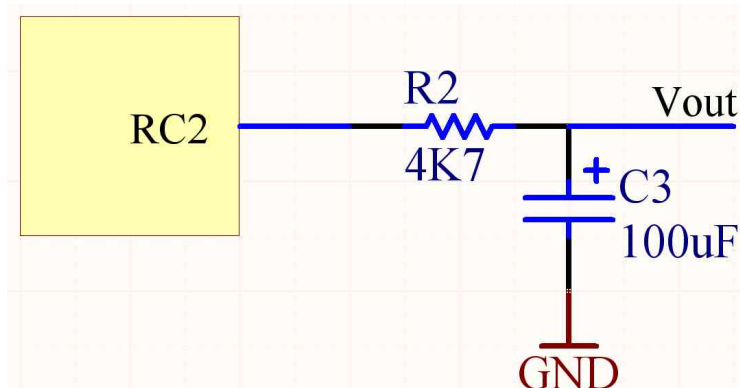


Εικόνα 5. Το κύκλωμα της εργαστηριακής άσκησης.



Εικόνα 6. Τα Configuration Bits.

- β) Ρυθμίστε τα Configuration Bits όπως φαίνονται στην εικόνα 6, προγραμματίστε το μικροελεγκτή με το πρόγραμμα της προσομοίωσης και εφαρμόστε την τροφοδοσία.
- γ) Να συνδέσετε ένα παλμογράφο στους ακροδέκτες RC1 και RC2 του μικροελεγκτή και να σχεδιάσετε το παλμογράφημα. Μετρήστε με τον παλμογράφο την περίοδο και το Duty Cycle των PWM κυματομορφών.
- δ) Τροποποιήστε το πρόγραμμα ώστε να φορτώνονται οι αριθμοί $h'C0'$ και $h'E0'$ στους καταχωρητές CCPR1L και CCPR2L αντίστοιχα. Επαναλάβετε τις μετρήσεις του προηγούμενου ερωτήματος.
- ε) Συνδέστε στον ακροδέκτη RC2 του μικροελεγκτή ένα χαμηλοπερατό φίλτρο RC όπως παρουσιάζεται στην εικόνα 7. Να συνδέσετε τον παλμογράφο και να σχεδιάσετε την κυματομορφή στο V_{out} . Τι παρατηρείτε για την τάση V_{out} ; Πώς επιδρά το φίλτρο RC στην PWM κυματομορφή; Δώστε την απάντησή σας.



Εικόνα 7. Σύνδεση RC φίλτρου στον ακροδέκτη RC2 του PIC16F887.

A/A	CCPR1L	Duty Cycle	Vout
1	h'00'		
2	h'40'		
3	h'60'		
4	h'80'		
5	h'A0'		
6	h'C0'		
7	h'E0'		
8	h'FF'		

Πίνακας 4

στ) Εκτελέστε πολλές φορές το πρόγραμμα φορτώνοντας κάθε φορά και διαφορετική τιμή στον καταχωρητή CCPR1L. Να συμπληρώσετε τον Πίνακα 4 για τις διάφορες τιμές του CCPR1L.

3. Εργασίες

α) Τροποποιήστε το πρόγραμμα ώστε να προκύπτει PWM κυματομορφή:

- με περίοδο 20KHz και Duty Cycle 20%
- με περίοδο 40KHz και Duty Cycle 40%

β) Τροποποιήστε το πρόγραμμα ώστε να παράγεται σταθερή τάση 2V στην έξοδο του φίλτρου.

4. Προχωρημένα θέματα

- α) Να υλοποιήσετε μια εφαρμογή η οποία παράγει μια PWM κυματομορφή με Duty Cycle ελεγχόμενο από ένα ποτενσιόμετρο.
- β) Να υλοποιήσετε μια εφαρμογή η οποία υλοποιεί έναν D/A Converter. Να εφαρμόσετε στην PORTD του μικροελεγκτή μια οκτάμπιτη λέξη και να την μετατρέψετε σε αναλογική (με χρήση PWM – χαμηλοπερατού φίλτρου) σε κλίμακα από 0V έως 5V.

ΑΝΑΛΟΓΙΚΟΙ ΣΥΓΚΡΙΤΕΣ ΚΑΙ ΤΑΣΗ ΑΝΑΦΟΡΑΣ

1. Θεωρία

Ο μικροελεγκτής PIC16F887 διαθέτει δύο αναλογικούς συγκριτές (Analog Comparators) και ενσωματωμένο κύκλωμα δημιουργίας τάσης αναφοράς (Voltage Reference). Με τους συγκριτές μπορούμε να εφαρμόσουμε αναλογικές τάσεις στο μικροελεγκτή και να παρατηρήσουμε τη σχέση μεταξύ τους. Διαθέτουν υστέρηση, ένα χαρακτηριστικό σύμφωνα με το οποίο η τάσεις κατωφλίου για την εναλλαγή στην έξοδο των συγκριτών από χαμηλή σε υψηλή στάθμη και το αντίστροφο είναι διαφορετικές μεταξύ τους. Αυτό προσδίδει μεγαλύτερη ανοχή στο θόρυβο. Με το κύκλωμα τάσης αναφοράς μπορούμε να δημιουργήσουμε διάφορες τάσεις αναλογιομετρικές της τάσης τροφοδοσίας. Τα δύο αυτά κυκλώματα μπορούν να χρησιμοποιηθούν σε συνδυασμό μεταξύ τους καθώς η τάση αναφοράς μπορεί να εφαρμοστεί απευθείας στη μη αναστρέφουσα είσοδο των συγκριτών.

Η ρύθμιση των συγκριτών πραγματοποιείται με τους καταχωρητές ειδικής χρήσης CM1CON0, CM2CON0 και CM2CON1 οι οποίοι βρίσκονται στη Bank2 του PIC16F887. Οι δύο πρώτοι χρησιμοποιούνται για τη ρύθμιση βασικών λειτουργιών των συγκριτών ενώ ο τρίτος για πιο εξελιγμένες λειτουργίες όπως η χρονομέτρηση αναλογικών ή ψηφιακών συμβάντων.

Με τους δύο συγκριτές μπορούν να χρησιμοποιηθούν διάφοροι ακροδέκτες του μικροελεγκτή. Στον συγκριτή C1 στην αναστρέφουσα είσοδο μπορεί να συνδεθεί οποιοσδήποτε από τους ακροδέκτες C12IN0- (RA0), C12IN1- (RA1), C12IN2- (RB3) και C12IN3- (RB1) καθώς πολυπλέκονται μεταξύ τους. Ο καθορισμός της αναστρέφουσας εισόδου πραγματοποιείται από τα bit C1CH1 και C1CH0 του καταχωρητή CM1CON0. Στη μη αναστρέφουσα είσοδο μπορεί να συνδεθεί είτε ο ακροδέκτης C1IN+ (RA3) είτε η έξοδος από το κύκλωμα δημιουργίας τάσης αναφοράς. Αυτό καθορίζεται από το bit C1R του καταχωρητή CM1CON0.

Στον συγκριτή C2 στην αναστρέφουσα είσοδο μπορεί να συνδεθεί επίσης οποιοσδήποτε ακροδέκτης από τους C12IN0-, C12IN1-, C12IN2- και C12IN3-. Αυτό καθορίζεται από τα bit C2CH1 και C2CH0 του καταχωρητή CM2CON0. Στη μη αναστρέφουσα είσοδο μπορεί να συνδεθεί είτε ο ακροδέκτης C2IN+ (RA2) είτε η έξοδος από το κύκλωμα δημιουργίας τάσης αναφοράς. Αυτό καθορίζεται από το bit C2R του καταχωρητή CM2CON0.

Η εφαρμογή αναλογικών τάσεων σε οποιονδήποτε ακροδέκτη του PIC16F887 πραγματοποιείται μέσω μιας ωμικής αντίστασης με τιμή μικρότερη των 10K.

Οι έξοδοι κάθε συγκριτή μπορούν να είναι διαθέσιμοι εξωτερικά του μικροελεγκτή εφόσον ρυθμιστεί κατάλληλα. Η έξοδος του συγκριτή C1 είναι διαθέσιμη στον ακροδέκτη C1OUT (RA4) ενώ η έξοδος του συγκριτή C2 είναι διαθέσιμη στον ακροδέκτη C2OUT (RA5). Αυτό πραγματοποιείται με τον καθορισμό στο λογικό '1' των bit C1OE και C2OE των καταχωρητών CM1CON0 και CM2CON0 αντίστοιχα. Επιπλέον τα αντίστοιχα bit του καταχωρητή TRISA (4 και 5) θα πρέπει να βρίσκονται στο λογικό '0' ώστε οι ακροδέκτες RA4 και RA5 είναι αποτελούν εξόδους.

Τα κυκλώματα των δύο συγκριτών μπορούν να προκαλέσουν διακοπή στο μικροεπεξεργαστή του PIC16F887. Η διακοπή σηματοδοτείται κάθε φορά που υπάρχει αλλαγή στην έξοδο κάποιου συγκριτή οδηγώντας στο λογικό '1' τα bit C1IF και C2IF αντίστοιχα. Τα bit αυτά θα πρέπει να μηδενιστούν στο λογισμικό του μικροελεγκτή αφού πρώτα γίνει ανάγνωση της κατάστασης εξόδου του συγκριτή διαβάζοντας τα bit C1OUT και C2OUT των καταχωρητών CM1CON0 και CM2CON0 αντίστοιχα. Για να εξυπηρετηθεί μια διακοπή που προέρχεται από τους συγκριτές θα πρέπει να έχουν καθοριστεί στο λογικό '1' τα bit C1IE και C2IE του καταχωρητή PIE2 όπως και τα bit PEIE και GIE του καταχωρητή INTCON.

Το κύκλωμα δημιουργίας τάσης αναφοράς ρυθμίζεται από τον καταχωρητή ειδικής χρήσης VRCON. Διαθέτει 16 επίπεδα τάσεων και μπορεί να λειτουργήσει σε δύο κλίμακες. Η κλίμακα καθορίζεται από το bit VRR του καταχωρητή VRCON. Η ακριβής τιμή καθορίζεται από τα bit VR3, VR2, VR1 και VR0.

Όταν VRR=1 τότε η παραγόμενη τάση αναφοράς είναι:

$$CV_{REF} = \frac{VRCON < 3:0 >}{24} \cdot V_{LADDER} \quad (1)$$

Όταν VRR=0 τότε η παραγόμενη τάση αναφοράς είναι:

$$CV_{REF} = \frac{V_{LADDER}}{4} + \frac{VRCON < 3:0 >}{32} \cdot V_{LADDER} \quad (2)$$

Η τάση V_{LADDER} καθορίζεται από το bit VRSS.

Όταν VRSS=1 τότε $V_{LADDER}=V_{REF+}-V_{REF-}$ που σημαίνει ότι η V_{LADDER} προκύπτει από δύο τιμές τάσεων που θα πρέπει να έχουμε εφαρμόσει εξωτερικά στους ακροδέκτες V_{REF+} (RA3) και V_{REF-} (RA2) του μικροελεγκτή.

Όταν VRSS=0 τότε $V_{LADDER}=V_{DD}$

Η παραγόμενη τάση αναφοράς είναι διαθέσιμη εξωτερικά στον ακροδέκτη CV_{REF} (RA2) του PIC16F887. Αυτό πραγματοποιείται κάνοντας '1' το bit VROE του καταχωρητή VRCON.

Τα βήματα που θα πρέπει να ακολουθηθούν για τη ρύθμιση των αναλογικών συγκριτών του PIC16F887 είναι:

- Ρύθμιση των bit CxCH<3:0> για τον καθορισμό του ακροδέκτη που θα εφαρμοστεί στην αναστρέφουσα είσοδο του συγκριτή
- Ρύθμιση του bit CxR για τον καθορισμό του σήματος που θα εφαρμοστεί στη μη αναστρέφουσα είσοδο του συγκριτή. Εάν επιλεγεί να εφαρμοστεί η τάση αναφοράς τότε στη συνέχεια πρέπει να ρυθμίζουμε το ενσωματωμένο κύκλωμα του μικροελεγκτή στην επιθυμητή τάση
- Καθορισμός στο λογικό '1' των κατάλληλων bit του καταχωρητή ANSEL και ANSELH ώστε τα κανάλια που χρησιμοποιούμε να είναι ρυθμισμένα ως αναλογικές εισοδοί
- Ρυθμίζουμε κατάλληλα το bit CxPOL αν θέλουμε να έχουμε αντίστροφη λογική στην έξοδο του συγκριτή
- Ρυθμίζουμε κατάλληλα το bit CxOE αν θέλουμε η έξοδος του συγκριτή να εμφανίζεται στους ακροδέκτες CxOUT του PIC16F887. Εάν επιλεγεί η έξοδος να οδηγείται εξωτερικά στο μικροελεγκτή τότε θα πρέπει να καθοριστούν στο λογικό '0' τα αντίστοιχα bit (4 ή/και 5) του καταχωρητή TRISA
- Καθορίζουμε στο λογικό '1' το bit CxON για την ενεργοποίηση του συγκριτή
- Εφόσον είναι επιθυμητό να χρησιμοποιήσουμε διακοπές καθορίζουμε στο λογικό '1' τα bit CxIE του καταχωρητή PIE2 και τα bit PEIE και GIE του καταχωρητή INTCON

(x=1 ή 2)

Τα βήματα που θα πρέπει να ακολουθηθούν για τη ρύθμιση του κυκλώματος δημιουργίας τάσης αναφοράς του PIC16F887 είναι:

- Ρύθμιση του bit VRSS για τον καθορισμό της τάσης V_{LADDER}
- Ρύθμιση του bit VRR για τον καθορισμό της κλίμακας
- Ρύθμιση των bit VR3, VR2, VR1 και VR0 για τον καθορισμό της ακριβούς τάσης αναφοράς σύμφωνα με τις σχέσεις (1) και (2)
- Ρύθμιση του bit VROE σχετικά με το αν θέλουμε να είναι διαθέσιμη η έξοδος της τάσης αναφοράς στον ακροδέκτη RA2 του PIC16F887.
- Καθορίζουμε στο λογικό '1' το bit VREN για την ενεργοποίηση του κυκλώματος

Στους πίνακες 1, 2, 3 και 4 περιγράφονται αναλυτικά οι καταχωρητές CM1CON0, CM2CON0, CM2CON1 και VRCON αντίστοιχα.

Καταχωρητής CM1CON0 (διεύθυνση h'107' της Bank2)

- bit 7 **C1ON**: Bit ενεργοποίησης συγκριτή C1
1 = Ο συγκριτής C1 είναι ενεργοποιημένος
0 = Ο συγκριτής C1 είναι απενεργοποιημένος
- bit 6 **C1OUT**: Bit εξόδου συγκριτή C1
Εάν C1POL=1 (αντίστροφη πολικότητα) τότε:
C1OUT = 0 όταν $C1V_{IN+} > C1V_{IN-}$
C1OUT = 1 όταν $C1V_{IN+} < C1V_{IN-}$
Εάν C1POL=1 (κανονική πολικότητα) τότε:
C1OUT = 1 όταν $C1V_{IN+} > C1V_{IN-}$
C1OUT = 0 όταν $C1V_{IN+} < C1V_{IN-}$
- bit 5 **C1OE**: Bit ενεργοποίησης εξόδου συγκριτή C1 στον ακροδέκτη C1OUT (RA4)
1 = Η έξοδος C1OUT του συγκριτή είναι διαθέσιμη στον ακροδέκτη C1OUT (RA4) ^(*)
0 = Η έξοδος C1OUT του συγκριτή δεν είναι διαθέσιμη εξωτερικά στο μικροελεγκτή
- bit 4 **C1POL**: Bit επιλογής πολικότητας εξόδου συγκριτή C1
1 = Η έξοδος C1OUT είναι αντεστραμμένη
0 = Η έξοδος C1OUT είναι κανονική
- bit 3 Δεν είναι υλοποιημένο
- bit 2 **C1R**: Bit επιλογής σήματος για τη μη αναστρέφουσα είσοδο του συγκριτή C1
1 = Η μη αναστρέφουσα είσοδος του συγκριτή C1 συνδέεται με την έξοδο του κυκλώματος τάσης αναφοράς
0 = Η μη αναστρέφουσα είσοδος του συγκριτή C1 συνδέεται με τον ακροδέκτη C1VIN+ (RA3) του PIC16F887
- bit 1-0 **C1CH<1:0>**: Bit επιλογής καναλιού για την αναστρέφουσα είσοδο του συγκριτή C1
00 = Ο ακροδέκτης C12VIN0- (RA0) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C1
01 = Ο ακροδέκτης C12VIN1- (RA1) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C1
10 = Ο ακροδέκτης C12VIN2- (RB3) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C1
11 = Ο ακροδέκτης C12VIN3- (RB1) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C1
- *1. Για να είναι διαθέσιμη η έξοδος του συγκριτή C1 στον ακροδέκτη C1OUT (RA4) απαιτούνται οι εξής συνθήκες: C1OE=1, C1ON=1 και TRISA<4>=0

Πίνακας 1

Καταχωρητής CM2CON0 (διεύθυνση h'108' της Bank2)

- bit 7 **C2ON**: Bit ενεργοποίησης συγκριτή C2
1 = Ο συγκριτής C2 είναι ενεργοποιημένος
0 = Ο συγκριτής C2 είναι απενεργοποιημένος
- bit 6 **C2OUT**: Bit εξόδου συγκριτή C2
Εάν C2POL=1 (αντίστροφη πολικότητα) τότε:
C2OUT = 0 όταν $C2V_{IN+} > C2V_{IN-}$
C2OUT = 1 όταν $C2V_{IN+} < C2V_{IN-}$
Εάν C2POL=1 (κανονική πολικότητα) τότε:
C2OUT = 1 όταν $C2V_{IN+} > C2V_{IN-}$
C2OUT = 0 όταν $C2V_{IN+} < C2V_{IN-}$
- bit 5 **C2OE**: Bit ενεργοποίησης εξόδου συγκριτή C2 στον ακροδέκτη C2OUT (RA5)
1 = Η έξοδος C2OUT του συγκριτή είναι διαθέσιμη στον ακροδέκτη C2OUT (RA5) ^(*)
0 = Η έξοδος C2OUT του συγκριτή δεν είναι διαθέσιμη εξωτερικά στο μικροελεγκτή
- bit 4 **C2POL**: Bit επιλογής πολικότητας εξόδου συγκριτή C2

- 1 = Η έξοδος C2OUT είναι αντεστραμμένη
 0 = Η έξοδος C2OUT είναι κανονική
- bit 3 Δεν είναι υλοποιημένο
- bit 2 **C2R**: Bit επιλογής σήματος για τη μη αναστρέφουσα είσοδο του συγκριτή C2
 1 = Η μη αναστρέφουσα είσοδος του συγκριτή C2 συνδέεται με την έξοδο του κυκλώματος τάσης αναφοράς
 0 = Η μη αναστρέφουσα είσοδος του συγκριτή C2 συνδέεται με τον ακροδέκτη C2V_{IN+} (RA2) του PIC16F887
- bit 1-0 **C2CH<1:0>**: Bit επιλογής καναλιού για την αναστρέφουσα είσοδο του συγκριτή C2
 00 = Ο ακροδέκτης C12VIN0- (RA0) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C2
 01 = Ο ακροδέκτης C12VIN1- (RA1) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C2
 10 = Ο ακροδέκτης C12VIN2- (RB3) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C2
 11 = Ο ακροδέκτης C12VIN3- (RB1) συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C2
- *2. Για να είναι διαθέσιμη η έξοδος του συγκριτή C2 στον ακροδέκτη C2OUT (RA5) απαιτούνται οι εξής συνθήκες: C2OE=1, C2ON=1 και TRISA<5>=0

Πίνακας 2

Καταχωρητής CM2CON1 (διεύθυνση h'109' της Bank2)

- bit 7 **MC1OUT**: Αντίγραφο του bit C1OUT (έξοδος συγκριτή C1)
- bit 6 **MC2OUT**: Αντίγραφο του bit C2OUT (έξοδος συγκριτή C2)
- bit 5 **C1RSEL**: Bit επιλογής τάσης αναφοράς για τη μη αναστρέφουσα είσοδο του συγκριτή C1
 1 = Ως τάση αναφοράς για τη μη αναστρέφουσα είσοδο του συγκριτή C1 επιλέγεται η έξοδος του κυκλώματος δημιουργίας τάσης αναφοράς
 0 = Ως τάση αναφοράς για τη μη αναστρέφουσα είσοδο του συγκριτή C1 επιλέγεται η τάση 0.6V
- bit 4 **C2RSEL**: Bit επιλογής τάσης αναφοράς για τη μη αναστρέφουσα είσοδο του συγκριτή C2
 1 = Ως τάση αναφοράς για τη μη αναστρέφουσα είσοδο του συγκριτή C2 επιλέγεται η έξοδος του κυκλώματος δημιουργίας τάσης αναφοράς
 0 = Ως τάση αναφοράς για τη μη αναστρέφουσα είσοδο του συγκριτή C2 επιλέγεται η τάση 0.6V
- bit 3-2 Δεν είναι υλοποιημένα
- bit 1 **T1GSS**: Bit επιλογής πηγής χρονομέτρησης για τον Timer1
 1 = Η πηγή χρονομέτρησης του Timer1 είναι ο ακροδέκτης T1G
 0 = Η πηγή χρονομέτρησης του Timer1 είναι η έξοδος του συγκριτή C2
- bit 1 **C2SYNC**: Bit επιλογής συγχρονισμού της εξόδου του συγκριτή C2
 1 = Η έξοδος του συγκριτή C2 συγχρονίζεται με το κατερχόμενο μέτωπο του ρολογιού του Timer1
 0 = Η έξοδος του συγκριτή C2 δεν συγχρονίζεται με το ρολόι του Timer1

Πίνακας 3

Καταχωρητής VRCON (διεύθυνση h'97' της Bank1)

- bit 7 **VREN**: Bit ενεργοποίησης κυκλώματος δημιουργίας τάσης αναφοράς
 1 = Το κύκλωμα δημιουργίας τάσης αναφοράς είναι ενεργοποιημένο
 0 = Το κύκλωμα δημιουργίας τάσης αναφοράς είναι απενεργοποιημένο ^{(*)3}
- bit 6 **VROE**: Bit ενεργοποίησης εξόδου κυκλώματος τάσης αναφοράς στον ακροδέκτη RA2
 1 = Η τάση αναφοράς είναι διαθέσιμη εξωτερικά στον ακροδέκτη RA2
 0 = Η τάση αναφοράς είναι δεν είναι διαθέσιμη εξωτερικά στον μικροελεγκτή
- bit 5 **VRR**: Bit επιλογής κλίμακας για το κύκλωμα δημιουργίας τάσης αναφοράς
 1 = Μικρή κλίμακα
 0 = Μεγάλη κλίμακα
- bit 4 **VRSS**: Bit επιλογής της τάσης VLADDER

1 = Η τάση V_{LADDER} καθορίζεται ως $V_{REF+} - V_{REF-}$

0 = Η τάση V_{LADDER} καθορίζεται ως VDD

bit 3-0 VR<3:0>: Bit καθορισμού τάσης αναφοράς, $0 \leq VR<3:0> \leq 15$

$$\text{Όταν VRR=1 τότε: } CV_{REF} = \frac{VR < 3:0 >}{24} \cdot V_{LADDER}$$

$$\text{Όταν VRR=0 τότε: } CV_{REF} = \frac{V_{LADDER}}{4} + \frac{VR < 3:0 >}{32} \cdot V_{LADDER}$$

*3. Το κύκλωμα δημιουργίας τάσης αναφοράς μπορεί να ενεργοποιηθεί και με το bit VREN στο λογικό '0'. Αυτό πραγματοποιείται όταν τουλάχιστον ένα από τα bit C1RSEL και C2RSEL του καταχωρητή CM2CON1 βρίσκονται στο λογικό '1'.

Πίνακας 4

Στην παρούσα εργαστηριακή άσκηση δημιουργούμε δύο αναλογικές τάσεις και τις οδηγούμε στο μικροελεγκτή για να τις συγκρίνουμε μεταξύ τους. Επιπλέον τη μία εκ των δύο τη συγκρίνουμε με μια τάση αναφοράς 2.5V την οποία δημιουργούμε από το ενσωματωμένο κύκλωμα του μικροελεγκτή. Πιο συγκεκριμένα:

α) Στον ακροδέκτη C12_{IN0-} (RA0) εφαρμόζουμε την πρώτη αναλογική τάση.

β) Στον ακροδέκτη C1_{IN+} (RA3) εφαρμόζουμε τη δεύτερη αναλογική τάση.

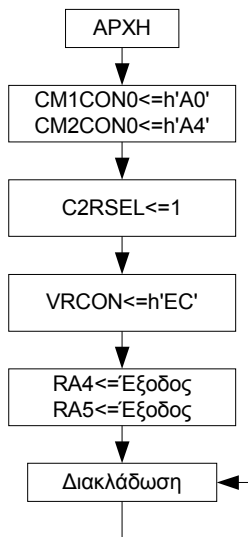
γ) Ρυθμίζουμε το μικροελεγκτή ώστε ο ακροδέκτης C12_{IN0-} να συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C1 και ο ακροδέκτης C1_{IN+} στη μη αναστρέφουσα είσοδο του C1.

δ) Ρυθμίζουμε το μικροελεγκτή ώστε ο ακροδέκτης C12_{IN0-} να συνδέεται στην αναστρέφουσα είσοδο του συγκριτή C2 και η τάση αναφοράς στη μη αναστρέφουσα είσοδο του C2.

ε) Ρυθμίζουμε την τάση αναφοράς στα 2.5V και την καθιστούμε διαθέσιμη στον ακροδέκτη RA2 του μικροελεγκτή.

στ) Το αποτέλεσμα των συγκριτών το οδηγούμε εξωτερικά στους ακροδέκτες C1OUT και C2OUT του μικροελεγκτή.

Το λογικό διάγραμμα του προγράμματος παρουσιάζεται στην εικόνα 1.



Εικόνα 1. Το λογικό διάγραμμα του προγράμματος.

2. Προσομοίωση

- α) Δημιουργήστε μια νέα εφαρμογή στο MPLAB βασισμένη στον PIC16F887 και αποθηκεύστε τη στο δίσκο του υπολογιστή.
- β) Δημιουργήστε ένα νέο αρχείο πηγαίου κώδικα στον κειμενογράφο του MPLAB και αντιγράψτε τον κώδικα που ακολουθεί:

```
include <p16f887.inc>

org h'0000'
nop

START
    bsf STATUS,RP1        ; Bank2
    movlw b'10100000'     ; Φόρτωση στον w το αριθμητικό h'A0'
    movwf CM1CON0         ; Αποθήκευση του w στον CM1CON0
    movlw b'10100100'     ; Φόρτωση στον w το αριθμητικό h'A4'
    movwf CM2CON0         ; Αποθήκευση του w στον CM2CON0
    bsf CM2CON1,C2RSEL    ; Καθορισμός στο '1' του bit C2RSEL του CM2CON1
    bcf STATUS,RP1        ; Bank0
    bsf STATUS,RP0        ; Bank1
    movlw b'11101100'     ; Φόρτωση στον w το αριθμητικό h'EC'
    movwf VRCON           ; Αποθήκευση του w στον VRCON
    bcf TRISA,4           ; Καθορισμός στο '1' του bit 4 του TRISA
    bcf TRISA,5           ; Καθορισμός στο '1' του bit 5 του TRISA
```

```
goto $ ; Άλλα στην ίδια εντολή  
  
end
```

γ) Μεταγλωττίστε το πρόγραμμα, διορθώστε, εάν υπάρχουν, συντακτικά λάθη και ενεργοποιήστε τον simulator.

δ) Να εμφανίσετε το παράθυρο διέγερσης. Μεταβείτε στην καρτέλα *Pin/Register Actions* και προσθέστε ένα σήμα διέγερσης κάνοντας κλικ στην ετικέτα *«Click here to Add Signals»*. Στο παράθυρο που θα εμφανιστεί επιλέξτε από τη λίστα τα σήματα AN0 και AN3 και πατήστε το κουμπί *«Add =>»*. Το παράθυρο θα πρέπει να φαίνεται όπως δείχνει η εικόνα 2. Πατήστε το κουμπί *«OK»*.

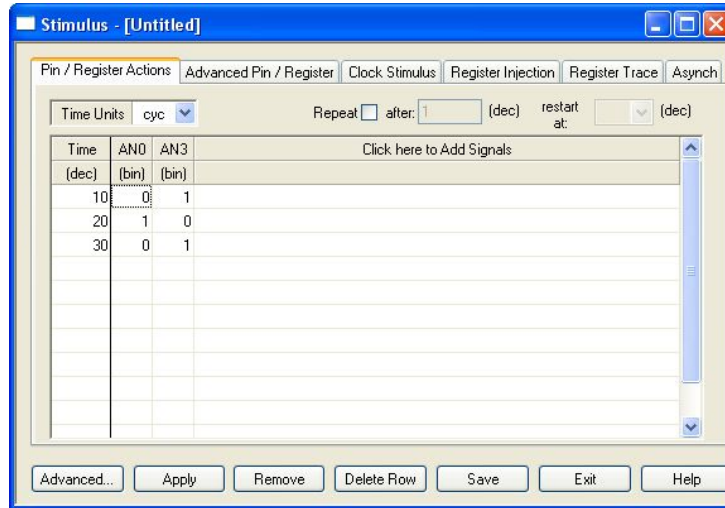
ε) Στο παράθυρο διέγερσης θα πρέπει τώρα δίπλα από τη στήλη *«Time»* να έχουν εμφανιστεί οι δύο στήλες *«AN0»* και *«AN3»*. Στις τρεις πρώτες γραμμές της στήλης *«Time»* γράψτε διαδοχικά τους αριθμούς 10, 20, 30. Στις τρεις πρώτες γραμμές της στήλης *«AN0»* γράψτε διαδοχικά τις τιμές 0, 1, 0. Ομοίως στη στήλη *«AN3»* γράψτε τις τιμές 1, 0, 1. Το παράθυρο διέγερσης θα πρέπει να είναι συμπληρωμένο όπως παρουσιάζεται στην εικόνα 3. Τέλος να διαλέξετε στη λίστα *«Time Units»* την επιλογή *«cyc / Instruction Cycles»*. Τα παραπάνω βήματα ακολουθήθηκαν για να καθορίσουμε σήματα διέγερσης στους ακροδέκτες AN0 και AN3 του PIC16F887. Επειδή δεν είναι δυνατόν να προσομοιωθούν αναλογικά σήματα για να ελέγξουμε τη λειτουργία των συγκριτών εφαρμόζουμε ψηφιακά σήματα. Ο συγκριτής θα δώσει σωστό αποτέλεσμα σύγκρισης και στα ψηφιακά σήματα.



Εικόνα 2. Προσθήκη σημάτων στο παράθυρο διέγερσης.

Πιο συγκεκριμένα μετά τους πρώτους 10 παλμούς ρολογιού εφαρμόζεται στον ακροδέκτη AN0 το λογικό '0' (χαμηλή στάθμη) και στον AN3 το λογικό '1' (υψηλή στάθμη). Επειδή AN0<AN3 ο συγκριτής θα πρέπει να δείχνει αρχικά λογικό '1'. Μετά το πέρας 20 παλμών ρολογιού αντιστρέφεται η κατάσταση στις εισόδους του συγκριτή οπότε θα πρέπει να έχουμε στην έξοδο του αλλαγή σε χαμηλή στάθμη. Μετά από 30

παλμούς έχουμε ξανά επαναφορά στην αρχική κατάσταση AN0=0 και AN3=1, πάντα σύμφωνα με το παράθυρο διέγερσης της προσομοίωσης, οπότε η έξοδος του συγκριτή θα πρέπει να επανέλθει στο λογικό '1'.



Εικόνα 3. Το παράθυρο διέγερσης.

στ) Πατήστε το κουμπί «Apply» για να εφαρμοστεί το παράθυρο διέγερσης στον προσομοιωτή. Επίσης ανοίξτε και το παράθυρο «StopWatch».

ζ) Εκτελέστε εντολή προς εντολής το πρόγραμμα. Παρατηρήστε σε κάθε κύκλο εντολής την τιμή στον ακροδέκτη εξόδου C1OUT. Να συμπληρώσετε τον πίνακα 5.

Κύκλος ρολογιού C1OUT

10^{ος}

15^{ος}

25^{ος}

35^{ος}

Πίνακας 5

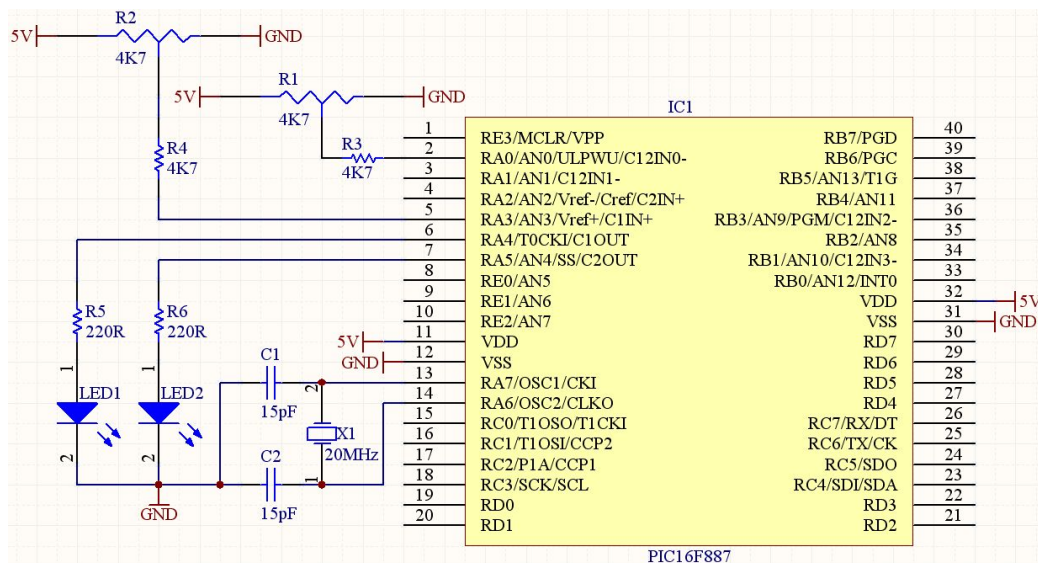
Να σημειώσουμε ότι δεν υπάρχει τρόπος να πραγματοποιήσουμε προσομοίωση στο κύκλωμα δημιουργίας τάσης αναφοράς.

3. Υλοποίηση

Για την περάτωση της παρούσας εργαστηριακής άσκησης θα χρειαστούν τα παρακάτω υλικά:

IC1 = PIC16F887
C1 = 15pF
C2 = 15pF
X1 = Κρύσταλος 20MHz
R1 = Τρίμερ 4K7, 1/4 W
R2 = Τρίμερ 4K7, 1/4 W
R3 = 4K7, 1/4W
R4 = 4K7, 1/4W
R5 = 220R, 1/4W
R6 = 220R, 1/4W
LED1 = Δίοδος LED
LED2 = Δίοδος LED

α) Υλοποιήστε σε πειραματική πλακέτα το κύκλωμα της εικόνας 4.

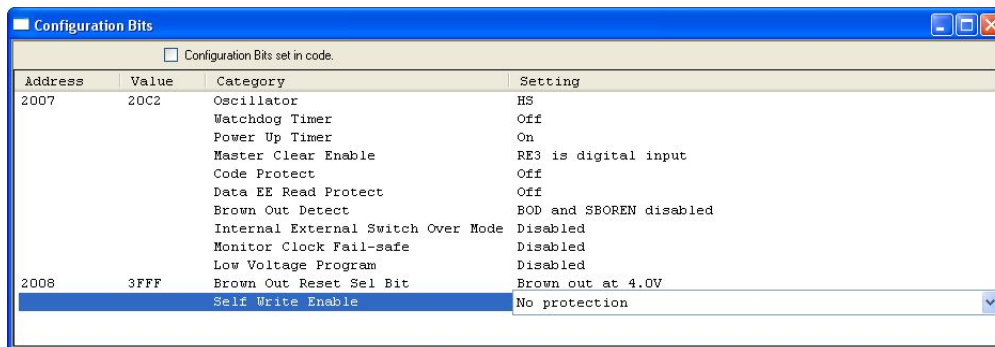


Εικόνα 4. Το κύκλωμα της εργαστηριακής άσκησης.

β) Ρυθμίστε τα Configuration Bits όπως φαίνονται στην εικόνα 5, προγραμματίστε το μικροελεγκτή με το πρόγραμμα της προσομοίωσης και εφαρμόστε την τροφοδοσία.
γ) Ρυθμίστε το τρίμερ R2 ώστε να δίνει στο μεσαίο ακροδέκτη αναλογική τάση $V_{R2}=1.5V$. Αυξομοιώστε την τάση στο τρίμερ R1 και παρατηρήστε την έξοδο στον ακροδέκτη C1OUT.

δ) Να μετρήσετε την τάση αναφοράς στον ακροδέκτη RA2. Ποια είναι η τιμή της; Επαληθεύστε την τιμή της τάσης αναφοράς με τη σχέση (1).

ε) Μεταβάλλετε την τάση στο τρίμερ R1 από 0V έως 5V. Για τις διάφορες τιμές τάσης να συμπληρώσετε τον πίνακα 2.



Εικόνα 5. Configuration Bits.

A/A	Τάση VR1	C1OUT	C2OUT
1	0V		
2	1V		
3	2V		
4	3V		
5	4V		
6	5V		

Πίνακας 2

στ) Τροποποιήστε το πρόγραμμα φορτώνοντας στα bit VR<3:0> την τιμή 15 ώστε να πάρουμε τη μέγιστη τάση αναφοράς που μπορεί να παραχθεί. Προγραμματίστε ξανά το μικροελεγκτή και επαληθεύστε πειραματικά το αποτέλεσμα μετρώντας την πραγματική τιμή της τάσης αναφοράς που προκύπτει.

ζ) Να συνδέσετε ένα παλμογράφο στους ακροδέκτες C1OUT και C2OUT του μικροελεγκτή. Αυξομοιώστε σιγά σιγά την τάση στο τρίμερ R1 ώστε να έχουμε εναλλαγές στην κατάσταση του συγκριτή. Όταν οι τάσεις προς σύγκριση βρίσκονται σε πολύ κοντινές τιμές μεταξύ τους, η παρουσία θορύβου επηρεάζει τη λειτουργία του συγκριτή δίνοντας πότε υψηλό λογικό επίπεδο και πότε χαμηλό; Πώς επιδρά το χαρακτηριστικό της υστέρησης στη λειτουργία των συγκριτών. Δώστε την απάντησή σας.

3. Εργασίες

- α) Τροποποιήστε κατάλληλα το πρόγραμμα ώστε να δημιουργείται τάση αναφοράς στο 1V.
- β) Τροποποιήστε κατάλληλα το πρόγραμμα ώστε οι έξοδοι C1OUT και C2OUT να βρίσκονται στο λογικό '1' όταν η τάση του τρίμερ R1 είναι μεγαλύτερη από την τάση του τρίμερ R2 και την τάση αναφοράς αντίστοιχα.

4. Προχωρημένα θέματα

- α) Να αναπτύξετε μια εφαρμογή η οποία ενεργοποιεί το συναγερμό όταν η έξοδος ενός αισθητήρα που μετράει κάποιο φυσικό μέγεθος βρίσκεται ανάμεσα σε 2V και 3V. Να χρησιμοποιηθούν συγκριτές και τάση αναφοράς.
- β) Να αναπτύξετε μια εφαρμογή η οποία ενεργοποιεί το συναγερμό όταν η έξοδος ενός αισθητήρα που μετράει κάποιο φυσικό μέγεθος δείχνει κάτω από 2V ή πάνω από 3V. Να χρησιμοποιηθούν συγκριτές και τάση αναφοράς.
- γ) Αναπτύξτε μια εφαρμογή η οποία δημιουργεί μια τάση αναφοράς 2.5V. Εφαρμόστε μια αναλογική τάση σε κάποιο κανάλι του μικροελεγκτή. Χρονομετρήστε με τη βοήθεια του Timer1, του συγκριτή C2 και καταχωρητών γενικής χρήσης τα διαστήματα που η αναλογική τάση είναι μεγαλύτερη από την τάση αναφοράς.