



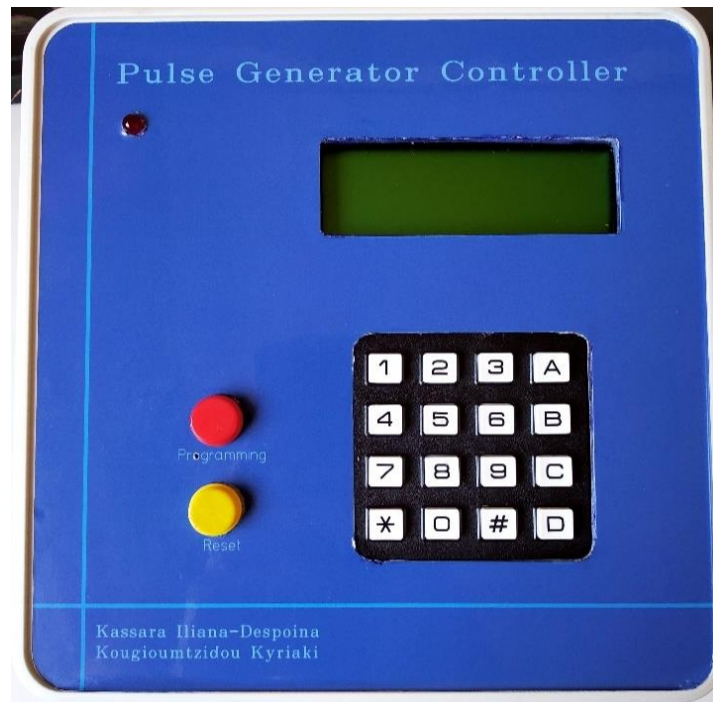
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ

ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σύστημα παραγωγής παλμών για έλεγχο φάσης ηλεκτρονικών ισχύος με τον μικροελεγκτή PIC18F4550



Των φοιτητριών

Κασσάρα Ηλιάννα-Δέσποινα 515052
Κουγιουμτζίδου Κυριακή 515059

Επιβλέπων

Καζακόπουλος Αριστοτέλης
Κοσμήτορας

Θεσσαλονίκη, Σεπτέμβριος 2020

Τίτλος Δ.Ε: Σύστημα παραγωγής παλμών για έλεγχο φάσης ηλεκτρονικών ισχύος με τον μικροελεγκτή PIC18F4550

Κωδικός Δ.Ε: 18156Μ

Όνοματεπώνυμο φοιτητριών: Κασσάρα Ηλιάνα-Δέσποινα
Κουγιουμτζίδου Κυριακή

Όνοματεπώνυμο εισηγητή: Καζακόπουλος Αριστοτέλης

Ημερομηνία ανάληψης Δ.Ε: 13/03/2019

Ημερομηνία περάτωσης Δ.Ε: 30/08/2020

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητριών Κασσάρα Ηλιάνα-Δέσποινα και Κουγιουμτζίδου Κυριακή που την εκπόνησαν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην οικογένειά μας»

Πρόλογος

Η συγκεκριμένη διπλωματική εργασία αποτελείται ουσιαστικά από δύο μέρη. Τον προγραμματισμό του μικροελεγκτή PIC18F4550 και την σχεδίαση των ισχύος για δοκιμή του προγράμματος. Επιλέξαμε, λοιπόν, τη συγκεκριμένη εργασία λόγω της επιθυμίας μας να συνδυάσουμε τα δύο αυτά μέρη, που καθ' όλη τη διάρκεια των σπουδών αποτέλεσαν τα αγαπημένα μας μαθήματα, με τα οποία θα θέλαμε να ασχοληθούμε και πέρα από το ακαδημαϊκό μέρος των σπουδών μας.

Τα οφέλη που αποκομίσαμε ήταν αρκετά. Αρχικά, μάθαμε περισσότερα για τα μικροϋπολογιστικά συστήματα, για τους μικροελεγκτές και τα περιφερειακά τους. Προγραμματίσαμε τον μικροελεγκτή και κατασκευάσαμε μια πλήρη συσκευή, αποτελούμενη από πληκτρολόγιο, οθόνη και κουμπιά. Ασχοληθήκαμε με το λογισμικό Proteus στο οποίο μπορέσαμε να κάνουμε μια προσομοίωση του κώδικά μας πριν τη τελική σχεδίαση της συσκευής. Τέλος, όσον αφορά το κομμάτι των ηλεκτρονικών ισχύος, χρησιμοποιήσαμε κυκλώματα απαραίτητα για μια διάταξη ηλεκτρονικών ισχύος και διαπιστώσαμε πως μπορούμε με πολύ απλό και οικονομικό τρόπο να ελέγχουμε την ροή ισχύος σε ηλεκτρονικές συσκευές.

Περίληψη

Στόχος της διπλωματικής εργασίας ήταν η μελέτη, η σχεδίαση/κατασκευή της πλακέτας, καθώς και ο προγραμματισμός του μικροελεγκτή της σειράς PIC18F, με σκοπό τη δημιουργία μιας ηλεκτρονικής συσκευής για τον έλεγχο ηλεκτρονικών ισχύος. Ο έλεγχος αυτός θα γίνεται από τον χρήστη, ο οποίος με τη χρήση του πληκτρολογίου θα εισάγει τις μοίρες που επιθυμεί και που αντιστοιχούν σε καθυστέρηση του θετικού παλμού εξόδου για τον έλεγχο της ροής ισχύος στο φορτίο. Σαν φορτίο επιλέχθηκε μια ηλεκτρική μηχανή (κινητήρας) καθώς και ένας λαμπτήρας. Περιληπτικά, παρακάτω φαίνεται το περιεχόμενο των κεφαλαίων που ακολουθούν.

Στο κεφάλαιο 1 περιγράφονται τα βασικά χαρακτηριστικά των μικροελεγκτών.

Στο κεφάλαιο 2 αναλύεται η σειρά PIC18F των μικροελεγκτών και περιγράφονται τα χαρακτηριστικά αυτών με περισσότερη λεπτομέρεια.

Στο κεφάλαιο 3 γίνεται η θεωρητική περιγραφή του συστήματος που θέλουμε να υλοποιήσουμε.

Στο κεφάλαιο 4 περιέχεται ο προγραμματισμός της πλακέτας σε γλώσσα C. Επιπλέον, γίνεται η επιλογή των εξαρτημάτων, σχεδιάζονται οι πλακέτες χρησιμοποιώντας το λογισμικό Altium Designer και η ολοκλήρωση της κατασκευής.

Στο κεφάλαιο 5 περιγράφονται τα αποτελέσματα και οι προτάσεις βελτίωσης.

Pulse generator controller

Kassara Iliana-Despoina
Kougioumtzidou Kyriaki

Abstract

The main purpose of this thesis was the investigation, designing/building of a PCB, as well as the programming of the microcontroller of PIC18F series, in order to create an electronic devise for the control of power electronics. This control will be done by the user, who using the keyboard will insert in the degrees that he desires. These degrees are about to correspond to delay of the positive output pulse for the control of the power flow on the load. Such a load was chose an electric machine (motor), as well as an electric light bulb. Concisely, the contents of the chapters are reported below.

Chapter 1 presents the basic features of microcontrollers.

Chapter 2 analyzes the PIC18F series of microcontrollers and describes the features of these microcontrollers thoroughly.

Chapter 3 provides a theoretical description of the system that we want to implement.

Chapter 4 includes the programming of our pcb in C language using software and hardware by Microchip and CCS Compiler (Custom Computer Services, Inc). Furthermore, becomes the selection of the electronics components, the designing of the pcb using the Altium Designer software and the consummation of the project.

Chapter 5 describes the results and the proposals for improvement.

Ευχαριστίες

Τις ευχαριστίες μας επιθυμούμε να εκφράσουμε σε όλους εκείνους που μας βοήθησαν με την άμεση ή έμμεση συμβολή τους στην αποπεράτωση αυτής της διπλωματικής εργασίας.

Αρχικά, θα θέλαμε να ευχαριστήσουμε τον επιβλέποντά μας, κοσμήτορα του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, Αριστοτέλη Καζακόπουλο. Η υποστήριξη και διαθεσιμότητά του καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής, αποτέλεσε σπουδαία βοήθεια, συμβάλλοντας ποικιλοτρόπως στην ολοκλήρωσή της, παρέχοντάς μας συμβουλές και καθοδήγηση όπου κρίθηκε αναγκαίο.

Επιπλέον, θα θέλαμε να ευχαριστούμε θερμά τους καθηγητές μας κ. Άγγελο Γιακουμή και κ. Ιορδάνη Κιοσκερίδη, για την μεγάλη βοήθεια που μας προσέφεραν. Η ανταπόκριση στις ανάγκες που προέκυπταν και η βαθιά γνώση του αντικειμένου και από τους δύο καθηγητές κρίθηκε καθοριστική για την ολοκλήρωση της εργασίας μας.

Αφιερώνουμε αυτήν τη διπλωματική εργασία στις οικογένειες και φίλους μας, που ήταν δίπλα μας σε όλη αυτήν την μεγάλη και δύσκολη προσπάθεια, δίνοντάς μας κουράγιο και δύναμη.

Περιεχόμενα

Πρόλογος	v
Περίληψη	vi
Abstract.....	vii
Ευχαριστίες	viii
Κατάλογος Σχημάτων.....	xiii
Κατάλογος Πινάκων	xiv
Κεφάλαιο 1^ο : Μικροϋπολογιστικά Συστήματα	
1.1 Εισαγωγή.....	1
1.2 Συστήματα Μικροελεγκτών.....	1
1.2.1 Ορισμός	1
1.2.2 Δομή	2
1.2.3 RAM.....	4
1.2.4 ROM.....	5
1.2.5 PROM.....	5
1.2.6 EPROM	5
1.2.7 EEPROM.....	5
1.2.8 Flash EEPROM	6
1.3 Χαρακτηριστικά Μικροελεγκτών	6
1.3.1 Τάση τροφοδοσίας (Supply Voltage)	6
1.3.2 Το Ρολόι (Clock)	6
1.3.3 Χρονιστές (Timers)	7
1.3.4 Watchdog Timer	7
1.3.5 Είσοδος Επανεκκίνησης (Reset Input)	7
1.3.6 Διακοπές (Interrupts).....	7
1.3.7 Brown-Out Detector	8
1.3.8 Μετατροπέας Αναλογικό-Σε-Ψηφιακό (Analog-to-Digital Converter)	8
1.3.9 Σειριακή Είσοδος-Έξοδος	9
1.3.10 EEPROM Μνήμη Δεδομένων (EEPROM Data Memory).....	9
1.3.11 LCD Drivers	9
1.3.12 Αναλογικός Συγκριτής (Analog Comparator)	9
1.3.13 Ρολόι πραγματικού χρόνου (Real-time Clock).....	9

1.3.14 Λειτουργία Ύπνου (Sleep Mode)	10
1.3.15 Power-on Reset.....	10
1.3.16 Λειτουργία Χαμηλής Κατανάλωσης (Low-Power Operation).....	10
1.3.17 Δυνατότητα Πηγής Ρεύματος (Current Sink/Source Capability)	10
1.3.18 Διεπαφή USB (USB Interface).....	10
1.3.19 Διεπαφή Ελέγχου Κινητήρα (Motor Control Interface)	11
1.3.20 Διεπαφή CAN (CAN interface).....	11
1.3.21 Διεπαφή Ethernet (Ethernet interface).....	11
1.3.22 Διεπαφή ZigBee (ZigBee interface)	11
1.4. Η αρχιτεκτονική των μικροελεγκτών PIC.....	12
1.4.1 Η αρχιτεκτονική Von Neumann και Harvard.....	12
1.4.2 Η εσωτερική αρχιτεκτονική των μικροελεγκτών PIC	13
Κεφάλαιο 2^ο : Μικροελεγκτές της σειράς PIC18F	
2.1 Η αρχιτεκτονική του μικροελεγκτή PIC18F4550	15
2.1.1 Ακροδέκτες	15
2.1.2 Εσωτερικό μπλοκ διάγραμμα	15
2.1.3 Διάγραμμα του Clock.....	16
2.1.4 Η Οργάνωση της Μνήμης του Προγράμματος	17
2.1.5 Η Οργάνωση της Μνήμης Δεδομένων	18
2.1.6 Τάση και Ρεύμα Τροφοδοσίας (Supply Voltage/Current).....	21
2.1.7 Επανεκκίνηση (Reset)	22
2.1.8 Clock.....	24
2.1.9 Μετατροπέας από αναλογικό σε ψηφιακό σήμα (A / D Converter).....	24
Κεφάλαιο 3^ο : Σύστημα παραγωγής παλμών για έλεγχο ηλεκτρονικών ισχύος	
3.1 Σύστημα παραγωγής παλμών με τη χρήση μικροελεγκτή	26
3.2 Ανιχνευτής μηδενός.....	27
3.3 Έλεγχος ισχύος με θυρίστορ.....	31
3.4 Προσομοίωση ολόκληρης της κατασκευής στο Matlab.....	35
Κεφάλαιο 4^ο : Κατασκευή και Προγραμματισμός	
4.1 Σχηματικό Κυκλώματος Μικροελεγκτή.....	40
4.2 Κατασκευή σχηματικού και τυπωμένου της πλακέτας.....	41
4.3 Κατασκευή πλακέτας.....	43
4.4 Αυτοκόλλητο κατασκευής.....	45

4.5 CCS C compiler	48
4.6 Διάγραμμα ροής κώδικα	48
4.7 Πηγαίος κώδικας του προγράμματος	49
4.8 Proteus.....	55
4.8.1 Δοκιμή λειτουργίας εισαγωγής παλμών και εμφάνιση στην οθόνη	55
4.8.2 Δοκιμή λειτουργίας διακοπών από RB και timer0	57
Κεφάλαιο 5^ο : Αποτελέσματα και Συμπεράσματα	
5.1 Αποτελέσματα.....	59
5.2 Συμπεράσματα-Προτάσεις βελτιώσεων.....	69
Βιβλιογραφία.....	70
Παράρτημα.....	71
A. main.h	71
B. flex_lcd.h	72
Γ. keypad.h	76

Κατάλογος Σχημάτων

Σχήμα 1.1 Η δομή του μικροελεγκτή	3
Σχήμα 1.2 Αρχιτεκτονική Von Neumann	12
Σχήμα 1.3 Αρχιτεκτονική Harvard	12
Σχήμα 1.4 Η εσωτερική αρχιτεκτονική του μικροελεγκτή PIC18F4550	13
Σχήμα 2.1 Ακροδέκτες PIC18F4550	15
Σχήμα 2.2 Εσωτερικό μπλοκ διάγραμμα	16
Σχήμα 2.3 Διάγραμμα clock	17
Σχήμα 2.4 Μνήμη προγράμματος του μικροελεγκτή PIC18F4550	18
Σχήμα 2.5 Μνήμη δεδομένων του μικροελεγκτή PIC18F4550	19
Σχήμα 2.6 Χάρτης SFR	20
Σχήμα 2.7 Απαιτήσεις για τάση στον PIC18F4550	21
Σχήμα 2.8 Απαιτήσεις για ρεύμα στον PIC18F4550	22
Σχήμα 2.9 Παράδειγμα 1 power-on reset	23
Σχήμα 2.10 Παράδειγμα 2 power-on reset	23
Σχήμα 2.11 Παράδειγμα ρολογιού με κρύσταλλο	24
Σχήμα 2.12 Οι 8 λειτουργίες του ταλαντωτή	24
Σχήμα 3.1 Σχηματική αναπαράσταση του συστήματος	26
Σχήμα 3.2 Η τάση εξόδου ρυθμίζεται από το σήμα οδήγησης της πύλης στο θυρίστορ	27
Σχήμα 3.3 Διάταξη zero crossing	28
Σχήμα 3.4 Κυματομορφή εισόδου και κυματομορφή εξόδου κυκλώματος zero crossing	28
Σχήμα 3.5 Αναπαράσταση λειτουργίας optocoupler	29
Σχήμα 3.6 PNP transistors	29
Σχήμα 3.7 Απλή διάταξη ισχύος με triac	31
Σχήμα 3.8 Σχηματικό τριφασικής γέφυρας	33
Σχήμα 3.9 Παραγόμενη κυματομορφή εξόδου με διαφορά φάσης	33
Σχήμα 3.10 Ανορθωμένη παραγόμενη κυματομορφή εξόδου	34
Σχήμα 3.11 Σχηματικό κυκλώματος οδήγησης	34
Σχήμα 3.12 Προσομοίωση στο Simulink	35
Σχήμα 3.13 Εισαγωγή μοιρών	36
Σχήμα 3.14 Εισαγωγή 0°	36
Σχήμα 3.15 Εισαγωγή 10°	37
Σχήμα 3.16 Εισαγωγή 20°	37
Σχήμα 3.17 Εισαγωγή 30°	37
Σχήμα 3.18 Εισαγωγή 50°	38
Σχήμα 3.19 Εισαγωγή 70°	38
Σχήμα 3.20 Εισαγωγή 90°	39
Σχήμα 4.1 Σχηματικό κυκλώματος μικροελεγκτή	40
Σχήμα 4.2 Σχηματικό σε μορφή pdf	42
Σχήμα 4.3 Σχέδιο για τύπωση της πλακέτας	43
Σχήμα 4.4 Τυπωμένη πλακέτα	43
Σχήμα 4.5 Κολλήσεις και άνοιγμα τρυπών για τα υλικά	44
Σχήμα 4.6 Τελική όψη πλακέτας	44
Σχήμα 4.7 Αυτοκόλλητο	46
Σχήμα 4.8 Τελική κατασκευή	47

Σχήμα 4.9 Αρχική κατάσταση προγράμματος	55
Σχήμα 4.10 Εισαγωγή 25° από το πληκτρολόγιο	56
Σχήμα 4.11 Δοκιμή διακοπών από τον timer0	57
Σχήμα 4.12 Πλακέτα κυκλωμάτων οδήγησης και τριφασικής γέφυρας	58
Σχήμα 5.1 Ρυθμίσεις γεννήτριας	59
Σχήμα 5.2 Εισαγωγή 90°	59
Σχήμα 5.3 Εισαγωγή 70°	59
Σχήμα 5.4 Λήψη αποτελεσμάτων κατασκευής	60
Σχήμα 5.5 Κυματομορφή εξόδου με καθυστέρηση φάσης 80°	60
Σχήμα 5.6 Κυματομορφή εξόδου με καθυστέρηση φάσης 70°	61
Σχήμα 5.7 Κυματομορφή εξόδου με καθυστέρηση φάσης 60°	61
Σχήμα 5.8 Κυματομορφή εξόδου με καθυστέρηση φάσης 50°	62
Σχήμα 5.9 Κυματομορφή εξόδου με καθυστέρηση φάσης 40°	62
Σχήμα 5.10 Κυματομορφή εξόδου με καθυστέρηση φάσης 30°	62
Σχήμα 5.11 Ισοδύναμο κύκλωμα κινητήρα	63
Σχήμα 5.12 Κυματομορφή εξόδου και φωτεινότητα λαμπτήρα με καθυστέρηση φάσης 80°	63
Σχήμα 5.13 Κυματομορφή εξόδου και φωτεινότητα λαμπτήρα με καθυστέρηση φάσης 80°	64
Σχήμα 5.14 Κυματομορφή εξόδου με καθυστέρηση φάσης 70°	64
Σχήμα 5.15 Κυματομορφή εξόδου με καθυστέρηση φάσης 60°	65
Σχήμα 5.16 Κυματομορφή εξόδου με καθυστέρηση φάσης 50°	65
Σχήμα 5.17 Κυματομορφή εξόδου με καθυστέρηση φάσης 40°	66
Σχήμα 5.18 Κυματομορφή εξόδου με καθυστέρηση φάσης 30°	66
Σχήμα 5.19 Κυματομορφή εξόδου με καθυστέρηση φάσης 20°	66
Σχήμα 5.20 Κυματομορφή εξόδου με καθυστέρηση φάσης 10°	67
Σχήμα 5.21 Κυματομορφή εξόδου με καθυστέρηση φάσης 0°	67

Κατάλογος Πινάκων

Πίνακας 1: Υλικά πλακέτας μικροελεγκτή	42
Πίνακας 2: Αποτελέσματα μετρήσεων	67

1.1 Εισαγωγή

Ο όρος *μικροϋπολογιστής* χρησιμοποιείται για να περιγράψει ένα σύστημα που περιλαμβάνει τουλάχιστον ένα μικροεπεξεργαστή, μνήμη προγράμματος, μνήμη δεδομένων και μια συσκευή εισόδου-εξόδου (I / O). Ορισμένα συστήματα μικροϋπολογιστών περιλαμβάνουν επιπρόσθετα στοιχεία όπως, χρονοδιακόπτες (timers), μετρητές (counters) και αναλογικούς σε ψηφιακούς μετατροπείς (ADC). Έτσι, ένα σύστημα μικροϋπολογιστών μπορεί να είναι οτιδήποτε, από έναν μεγάλο υπολογιστή που έχει σκληρούς δίσκους, δισκέτες και εκτυπωτές σε ενσωματωμένο ελεγκτή ενός τσιπ.

Στην παρούσα διπλωματική εργασία θα μελετήσουμε μόνο τον τύπο των μικροϋπολογιστών που αποτελούνται από ένα ενιαίο τσιπ πυριτίου. Τέτοιου είδους μικροϋπολογιστικά συστήματα ονομάζονται *μικροελεγκτές* και χρησιμοποιούνται σε ηλεκτρικές συσκευές, συστήματα αυτοματισμών, κυκλώματα τηλεπικοινωνιών, συστήματα συλλογής δεδομένων (Data Acquisition), εφαρμογές ηλεκτρονικών ισχύος, συστήματα διασύνδεσης, εφαρμογές δικτύων. Γενικότερα, οι μικροελεγκτές χρησιμοποιούνται όπου απαιτείται έλεγχος συστημάτων.

Στην αγορά υπάρχει μεγάλη ποικιλία διαφορετικών μικροελεγκτών από εταιρείες όπως Texas Instruments, Microchip, Silicon Labs, Renesas, Intel, Fujitsu, ST Microelectronics, Zilog, Freescale, Maxim, Toshiba, Hitachi, Atmel και πολλές άλλες. Ο μικροελεγκτής που χρησιμοποιήθηκε ανήκει στη σειράς των PIC (programmable interface controller) μικροελεγκτών, που κατασκευάζονται από την Microchip Technology Inc.

1.2 Συστήματα Μικροελεγκτών

1.2.1 Ορισμός

Ο όρος *μικροελεγκτής* χρησιμοποιείται για να περιγράψει ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα, το οποίο διαθέτει τουλάχιστον έναν επεξεργαστή, μνήμη προγράμματος, μνήμη δεδομένων, διάφορα περιφερειακά κυκλώματα, καθώς επίσης και θύρες εισόδου/εξόδου (I/O) για επικοινωνία με εξωτερικές συσκευές και μάλιστα ολοκληρωμένα σε ένα μόνο τσιπ. Η λέξη -μικρο- υποδεικνύει ότι η συσκευή είναι μικρή, και ο ελεγκτής υποδεικνύει ότι η συσκευή χρησιμοποιείται σε εφαρμογές ελέγχου. Ένας άλλος όρος για τον μικροελεγκτή είναι ενσωματωμένος ελεγκτής, δεδομένου ότι οι περισσότεροι μικροελεγκτές έχουν κατασκευαστεί ή ενσωματωθεί μέσα σε συσκευές τις οποίες ελέγχουν.

1.2.2 Δομή

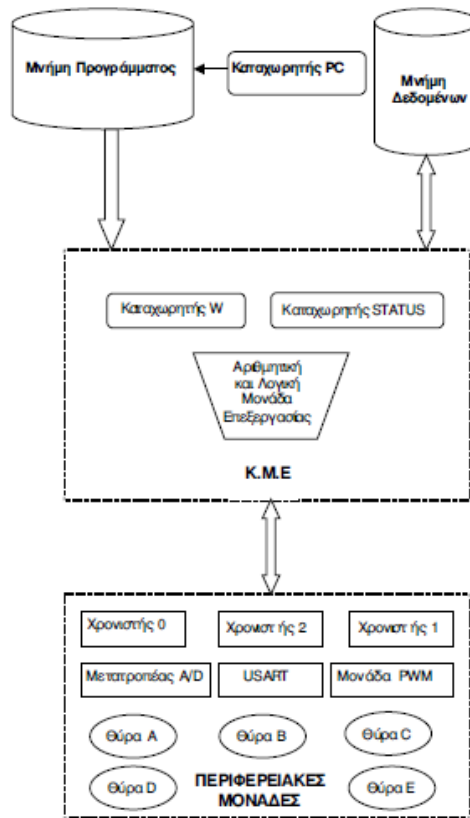
Μπορούμε να διαχωρίσουμε τη δομή του μικροελεγκτή PIC σε δύο μέρη, τον πυρήνα (core) και τις περιφερειακές μονάδες του (peripheral units). Ο πυρήνας αποτελείται από όλα τα στοιχεία, τα οποία είναι απαραίτητα για τη λειτουργία του μικροελεγκτή. Οι περιφερειακές μονάδες είναι αυτές που κάνουν το μικροελεγκτή να διαφέρει από ένα μικροεπεξεργαστή και βρίσκονται ενσωματωμένες σε αυτόν.

Στον πυρήνα PIC του ανήκουν οι:

- ❖ Κεντρική μονάδα επεξεργασίας
- ❖ Μνήμη
- ❖ Εντολές
- ❖ Λειτουργίες διακοπών

Στις περιφερειακές μονάδες ανήκουν οι:

- ❖ Θύρες εισόδου/εξόδου γενικής χρήσης
- ❖ Μετρητές χρόνου (τρεις μονάδες)
- ❖ Μονάδα διαμόρφωσης πλάτους
- ❖ Θύρες σειριακής επικοινωνίας (τρεις θύρες)
- ❖ Θύρα παράλληλης επικοινωνίας
- ❖ Μονάδα παραγωγής τάσης αναφοράς
- ❖ Συγκριτές
- ❖ Μετατροπέας αναλογικού σήματος σε ψηφιακό



Σχήμα 1.1 Η δομή του μικροελεγκτή

Αν και ο μικροελεγκτής είναι ένα είδος επεξεργαστή, αποτελεί ουσιαστικά μια παραλλαγή του μικροεπεξεργαστή. Ένας μικροεπεξεργαστής διαφέρει από έναν μικροελεγκτή με διάφορους τρόπους. Η κύρια διάκρισή τους είναι ότι ένας μικροεπεξεργαστής απαιτεί διάφορα άλλα στοιχεία για τη λειτουργία του, όπως μνήμη προγράμματος και μνήμη δεδομένων, συσκευές εισόδου-εξόδου και ένα εξωτερικό ρολόι κύκλωμα. Από την άλλη πλευρά, ένας μικροελεγκτής, έχει ενσωματωμένα τα περισσότερα υποσυστήματα υποστήριξης, μέσα στο ενιαίο τσιπ του.

Όλοι οι μικροελεγκτές λειτουργούν σε ένα σύνολο οδηγιών (ή του χρήστη προγράμματος) που έχουν αποθηκευτεί στη μνήμη τους. Ένας μικροελεγκτής ανακτά τις οδηγίες από το μνήμη προγράμματος μία προς μία, τις αποκωδικοποιεί και στη συνέχεια εκτελεί τις λειτουργίες που απαιτούνται.

Οι μικροελεγκτές παραδοσιακά έχουν προγραμματιστεί χρησιμοποιώντας τη γλώσσα assembly της συσκευής προορισμού. Παρότι η γλώσσα assembly είναι γρήγορη, παρουσιάζει αρκετά μειονεκτήματα. Ένα πρόγραμμα assembly αποτελείται από μνημονικά (κωδικοί εντολών), τα οποία κάνουν τη μάθηση και τη συντήρηση ενός προγράμματος δύσκολη. Επιπρόσθετα, μικροελεγκτές που έχουν κατασκευαστεί από διαφορετικές εταιρείες, έχουν διαφορετική γλώσσα assembly, συνεπώς ο χρήστης πρέπει να μάθει νέα γλώσσα με κάθε νέο μικροελεγκτή που πρόκειται να χρησιμοποιήσει.

Οι μικροελεγκτές μπορούν επίσης να προγραμματιστούν χρησιμοποιώντας μια γλώσσα υψηλού επιπέδου, όπως η BASIC, PASCAL ή C. Είναι ευκολότερο να μάθει κανείς μια γλώσσα υψηλού συγκριτικά με τη γλώσσα assembly. Επιπλέον, οι γλώσσες υψηλού επιπέδου διευκολύνουν την ανάπτυξη μεγάλων και πολύπλοκων προγραμμάτων. Στη διπλωματική εργασία προγραμματίζουμε το μικροελεγκτή PIC χρησιμοποιώντας τη γλώσσα C.

Θεωρητικά, ένα μόνο τσιπ αρκεί ώστε να λειτουργεί ένα σύστημα μικροελεγκτή. Σε πρακτικές εφαρμογές, ωστόσο, μπορεί να χρειασθούν επιπρόσθετα εξαρτήματα, ώστε ο μικροϋπολογιστής να έχει δυνατότητα διασύνδεσης με το περιβάλλον του. Με την έλευση της οικογένειας των μικροελεγκτών PIC ο χρόνος ανάπτυξης ενός ηλεκτρονικού προγράμματος έχει μειωθεί σε μερικές ώρες.

Ουσιαστικά, ένας μικροϋπολογιστής εκτελεί ένα πρόγραμμα χρήστη το οποίο έχει φορτωθεί στη μνήμη προγράμματος. Υπό τον έλεγχο του προγράμματος αυτού, δεδομένα λαμβάνονται από εξωτερικές συσκευές (είσοδοι), παραποιοούνται, και στη συνέχεια αποστέλλονται σε εξωτερικές συσκευές (έξοδοι).

Ο μικροελεγκτής είναι ένα πολύ ισχυρό εργαλείο που επιτρέπει σε έναν σχεδιαστή να δημιουργήσει εξελιγμένη διαχείριση των δεδομένων εισόδου-εξόδου υπό τον έλεγχο του προγράμματος. Οι μικροελεγκτές ταξινομούνται με βάση τον αριθμό των bits που επεξεργάζονται. Μικροελεγκτές με 8 bits είναι οι πιο δημοφιλείς και χρησιμοποιούνται στις περισσότερες εφαρμογές. Μικροελεγκτές με 16 και 32 bits είναι πολύ πιο ισχυροί, αλλά συνήθως ακριβότεροι και δεν απαιτούνται σε εφαρμογές μικρού ή μεσαίου μεγέθους γενικής χρήσης, οι οποίες απαιτούν μικροελεγκτές.

Η απλούστερη αρχιτεκτονική του μικροελεγκτή αποτελείται από ένα μικροεπεξεργαστή, μνήμη, και εισόδους-εξόδους. Ο μικροεπεξεργαστής αποτελείται από μια κεντρική μονάδα επεξεργασίας (CPU) και μια μονάδα ελέγχου (CU). Η κεντρική μονάδα επεξεργασίας (CPU) είναι ο εγκέφαλος του μικροελεγκτή, το μέρος στο οποίο γίνονται όλες οι αριθμητικές και λογικές πράξεις. Η μονάδα ελέγχου (CU) ελέγχει και συντονίζει τις εσωτερικές λειτουργίες του μικροεπεξεργαστή και στέλνει σήματα σε άλλα μέρη του μικροελεγκτή, ώστε να εκτελεστούν οι απαιτούμενες εντολές (instructions). Η κύρια λειτουργία της μονάδας ελέγχου είναι να διαβάζει και να αποκωδικοποιεί τις εντολές από την μνήμη προγράμματος.

Η μνήμη, ως ένα σημαντικό μέρος του συστήματος του μικροελεγκτή, μπορεί να ταξινομηθεί σε δύο τύπους: τη μνήμη προγράμματος και τη μνήμη δεδομένων. Η μνήμη προγράμματος αποθηκεύει το γραμμένο πρόγραμμα του προγραμματιστή και είναι συνήθως μη πτητική (δηλαδή, τα δεδομένα δεν χάνονται μετά τη διακοπή της τροφοδοσίας ρεύματος). Η μνήμη δεδομένων αποθηκεύει προσωρινά τα δεδομένα που χρησιμοποιούνται σε ένα πρόγραμμα και είναι συνήθως πτητική (τα δεδομένα χάνονται μετά τη διακοπή της τροφοδοσίας ρεύματος).

Υπάρχουν βασικά έξι τύποι μνημών:

1.2.3 RAM

RAM (Random Access Memory), μνήμη τυχαίας προσπέλασης, είναι μια μνήμη γενικής χρήσης που αποθηκεύει συνήθως δεδομένα χρήστη σε ένα πρόγραμμα. Η μνήμη RAM είναι πτητική, αφού με τη διακοπή της παροχής ρεύματος, τα περιεχόμενα της μνήμης διαγράφονται. Χρησιμοποιείται για προσωρινή αποθήκευση δεδομένων και ενδιάμεσων αποτελεσμάτων που δημιουργήθηκαν και χρησιμοποιήθηκαν κατά τη λειτουργία του μικροελεγκτή. Οι περισσότεροι μικροελεγκτές έχουν κάποια εσωτερική μνήμη RAM, συνήθως 256 bytes, ωστόσο η τιμή της μνήμης μπορεί να διαφέρει

ανά μικροελεγκτή. Ο PIC18F4550 που χρησιμοποιούμε, για παράδειγμα, έχει 2048 bytes μνήμη RAM. Η μνήμη μπορεί να επεκταθεί με την προσθήκη εξωτερικών τσιπ μνήμης.

1.2.4 ROM

ROM (Read-Only Memory), μνήμη μόνο ανάγνωσης, συνήθως καταλαμβάνει ένα πρόγραμμα ή σταθερά δεδομένα χρήστη. Η ROM είναι μη πτητική, δηλαδή τα αρχικά δεδομένα δεν χάνονται με τη διακοπή της παροχής ρεύματος. Επιπλέον, μια μνήμη ROM προγραμματίζεται κατά τη διάρκεια της διαδικασίας κατασκευής, και ο χρήστης δεν μπορεί να αλλάξει το περιεχόμενό του. Η μνήμη ROM είναι χρήσιμη μόνο αν έχουμε αναπτύξει ένα πρόγραμμα και θέλουμε να δημιουργήσουμε αρκετές χιλιάδες αντίγραφα αυτού.

1.2.5 PROM

PROM (Programmable Read-Only Memory), προγραμματιζόμενη μνήμη μόνο για ανάγνωση, είναι ένας τύπος ROM που μπορεί να προγραμματιστεί από τον τελικό χρήστη, χρησιμοποιώντας μια συσκευή που ονομάζεται PROM προγραμματιστής. Μόλις η PROM προγραμματιστεί, το περιεχόμενό της δεν μπορεί να αλλάξει. Οι PROM χρησιμοποιούνται συνήθως σε εφαρμογές χαμηλής παραγωγής, όπου απαιτούνται μόνο μερικές από αυτές τις μνήμες.

1.2.6 EPROM

Η EPROM (Erasable Programmable Read-Only Memory), διαγραφόμενη προγραμματιζόμενη μνήμη μόνο για ανάγνωση, είναι παρόμοια με την ROM, αλλά μπορεί να προγραμματιστεί χρησιμοποιώντας μια κατάλληλη συσκευή προγραμματισμού. Μια μνήμη EPROM έχει μικρό διαφανές παράθυρο πάνω από το τσιπ, όπου τα δεδομένα μπορούν να διαγραφούν με ισχυρό υπεριώδες φως. Μόλις η μνήμη προγραμματιστεί, το παράθυρο μπορεί να καλυφθεί με σκοτεινή ταινία για να αποφευχθεί η καταλάθος διαγραφή των δεδομένων. Μια μνήμη EPROM πρέπει να διαγραφεί προτού να επαναπρογραμματιστεί. Πολλές αναπτυξιακές εκδόσεις των μικροελεγκτών κατασκευάζονται με τις μνήμες EPROM, όπου το πρόγραμμα χρήστη μπορεί να αποθηκευτεί. Ορισμένες εκδόσεις των EPROM, γνωστές ως OTP (μία φορά προγραμματιζόμενες), μπορούν να προγραμματιστούν χρησιμοποιώντας μια κατάλληλη συσκευή προγραμματισμού, αλλά δεν μπορούν να διαγραφούν. Οι μνήμες OTP είναι πολύ φθηνότερες από τις EPROM. Οι μνήμες OTP είναι χρήσιμες σε περίπτωση που ένα πρόγραμμα έχει αναπτυχθεί πλήρως και χρειάζονται πολλά αντίγραφα της μνήμης του προγράμματος.

1.2.7 EEPROM

Η EEPROM (Electrically Erasable Programmable Read Only Memory), ηλεκτρικά διαγραφόμενη προγραμματιζόμενη μνήμη μόνο για ανάγνωση, είναι μια μη πτητική μνήμη που μπορεί να διαγραφεί και να επαναπρογραμματιστεί χρησιμοποιώντας μια κατάλληλη συσκευή προγραμματισμού. Οι μνήμες EEPROM χρησιμοποιούνται για την αποθήκευση πληροφοριών ρύθμισης παραμέτρων, μέγιστες και ελάχιστες τιμές, δεδομένα ταυτοποίησης κ.α. Μερικοί μικροελεγκτές έχουν ενσωματωμένη μνήμη EEPROM. Για παράδειγμα, ο PIC18F4550 έχει 256-byte μνήμη EEPROM,

όπου κάθε byte μπορεί να προγραμματιστεί και να διαγραφεί άμεσα από το λογισμικό. Οι μνήμες EEPROM είναι συνήθως πολύ αργές. Επίσης είναι πολύ ακριβότερες από τις μνήμες EPROM.

1.2.8 Flash EEPROM

Η Flash EEPROM, μια έκδοση της μνήμης EEPROM, έχει γίνει δημοφιλής στις εφαρμογές των μικροελεγκτών και χρησιμοποιείται για την αποθήκευση του προγράμματος χρήστη. Η Flash EEPROM είναι μη πτητική και συνήθως πολύ γρήγορη. Τα δεδομένα μπορούν να διαγραφούν και στη συνέχεια να επαναπρογραμματιστεί χρησιμοποιώντας μια κατάλληλη συσκευή προγραμματισμού. Μερικοί μικροελεγκτές έχουν μόνο 1K Flash EEPROM μνήμη, ενώ άλλοι 32K ή περισσότερο. Ο μικροελεγκτής PIC18F4550 έχει 32K bytes μνήμη Flash.

1.3 Χαρακτηριστικά Μικροελεγκτών

Μικροελεγκτές από διαφορετικούς κατασκευαστές έχουν διαφορετική αρχιτεκτονική και διαφορετικές ικανότητες. Κάποιοι μπορεί να ταιριάζουν σε μια συγκεκριμένη εφαρμογή, ενώ άλλοι μπορεί να είναι εντελώς ακατάλληλοι για την ίδια εφαρμογή. Σε αυτήν την ενότητα περιγράφονται τα χαρακτηριστικά υλικού (hardware) που είναι κοινά στους περισσότερους μικροελεγκτές.

1.3.1 Τάση τροφοδοσίας (Supply Voltage)

Οι περισσότεροι μικροελεγκτές λειτουργούν με λογική τάση των +5V. Μερικοί μικροελεγκτές μπορούν να λειτουργούν σε τόσο χαμηλή τάση όσο τα +2.7V, ενώ άλλοι μπορούν να ανεχθούν +6V χωρίς κάποιο πρόβλημα. Τα φύλλα δεδομένων (datasheets) του κατασκευαστή θα έχουν πληροφορίες σχετικά με τα επιτρεπόμενα όρια της τάσης τροφοδοσίας. Η τάση τροφοδοσίας του PIC18F4550 κυμαίνεται από +4.2V έως +5.5V.

1.3.2 Το Ρολόι (Clock)

Όλοι οι μικροελεγκτές απαιτούν ένα ρολόι (ή έναν ταλαντωτή) για να λειτουργήσουν, το οποίο συνήθως παρέχεται από εξωτερικές συσκευές χρονοισμού που συνδέονται με τον μικροελεγκτή. Στις περισσότερες περιπτώσεις οι εξωτερικές συσκευές χρονοισμού είναι ένας κρύσταλλος συν δύο μικροί πυκνωτές. Σε ορισμένες περιπτώσεις είναι αντηχεία ή ένα εξωτερικό ζεύγος αντιστάσεως-πυκνωτή. Μερικοί μικροελεγκτές έχουν ενσωματωμένα τα κυκλώματα χρονοισμού και δεν απαιτούν εξωτερικά εξαρτήματα χρονοισμού. Εάν η εφαρμογή δεν απαιτεί μεγάλη χρονική ακρίβεια, εξωτερικά ή εσωτερικά (αν είναι διαθέσιμο) εξαρτήματα αντιστάσεως-πυκνωτή είναι η καλύτερη επιλογή για την απλότητα τους και το χαμηλό κόστος.

Μια εντολή (instruction) εκτελείται με την ανάκληση (fetching) από την μνήμη και έπειτα αποκωδικοποιώντας την. Αυτό συνήθως διαρκεί αρκετούς κύκλους ρολογιού και είναι γνωστή ως κύκλος εντολής (instruction cycle or fetch–decode–execute cycle). Στους PIC μικροελεγκτές, ένας κύκλος εντολής διαρκεί τέσσερις περιόδους ρολογιού. Συνεπώς, ο μικροελεγκτής λειτουργεί σε ρυθμό

ρολογιού που είναι το ένα τέταρτο (1/4) της συχνότητας του ταλαντωτή. Η PIC18F σειρά των μικροελεγκτών μπορεί να λειτουργήσει με συχνότητα ρολογιού έως 40MHz.

1.3.3 Χρονιστές (Timers)

Οι χρονιστές αποτελούν σημαντικά μέρη του κάθε μικροελεγκτή. Ένας χρονιστής είναι στην ουσία ένας μετρητής που οδηγείται από ένα εξωτερικό παλμό ρολογιού ή εσωτερικό ταλαντωτή του μικροελεγκτή. Ο χρονιστής μπορεί να είναι εύρους 8 bits ή 16 bits. Δεδομένα μπορούν να φορτωθούν στον χρονιστή υπό τον έλεγχο του προγράμματος, και η λειτουργία του χρονιστή μπορεί να σταματήσει ή να ξεκινήσει υπό τον έλεγχο του προγράμματος. Οι περισσότεροι χρονιστές μπορεί να ρυθμιστούν ώστε να παράγουν μια διακοπή (interrupt) όταν φθάσουν σε έναν ορισμένο αριθμό (συνήθως όταν συμβαίνει υπερχείλιση, overflow). Το πρόγραμμα χρήστη μπορεί να χρησιμοποιήσει μια διακοπή για να τακτοποιήσει λειτουργίες με χρονική ακρίβεια στο εσωτερικό του μικροελεγκτή. Οι μικροελεγκτές της σειράς PIC18F έχουν τουλάχιστον τρεις χρονιστές. Ο PIC18F4550 έχει τέσσερις timers. Μερικοί μικροελεγκτές έχουν δυνατότητα κράτησης και σύγκρισης, όπου μια τιμή του χρονιστή διαβάζεται όταν συμβαίνει ένα εξωτερικό συμβάν, ή συγκρίνεται με μια προκαθορισμένη τιμή και προκαλείται διακοπή όταν φτάσει σε αυτή την τιμή. Οι περισσότεροι PIC18F μικροελεγκτές έχουν τουλάχιστον δύο μονάδες κράτησης και σύγκρισης.

1.3.4 Watchdog Timer

Οι περισσότεροι μικροελεγκτές έχουν τουλάχιστον μία μονάδα watchdog (φύλακα). Watchdog είναι ουσιαστικά ένας χρονιστής, ο οποίος ανανεώνεται από το πρόγραμμα χρήστη. Κάθε φορά που το πρόγραμμα αποτυγχάνει να ανανεώσει το watchdog, συμβαίνει επανεκκίνηση (reset). Ο watch dog timer χρησιμοποιείται για να ανιχνεύσει ένα πρόβλημα του συστήματος, όπως για παράδειγμα σε περίπτωση που το πρόγραμμα βρίσκεται σε έναν ατέρμονα βρόχο. Αυτό το χαρακτηριστικό ασφαλείας αποτρέπει την εκτέλεση ανεπιθύμητου και χωρίς νόημα κώδικα από τον μικροελεγκτή. Οι μονάδες watchdog χρησιμοποιούνται συνήθως σε συστήματα πραγματικού χρόνου, όπου ο επιτυχής τερματισμός μίας ή περισσότερων δραστηριοτήτων πρέπει να ελέγχεται τακτικά.

1.3.5 Είσοδος Επανεκκίνησης (Reset Input)

Μία είσοδος επανεκκίνησης (reset input) χρησιμοποιείται για να κάνει reset έναν μικροελεγκτή εξωτερικά. Η επανεκκίνηση βάζει τον μικροελεγκτή σε μία γνωστή κατάσταση, όπως η εκτέλεση του προγράμματος, να ξεκινά από τη διεύθυνση 0 της μνήμης προγράμματος. Η εξωτερική επανεκκίνηση επιτυγχάνεται συνήθως με την σύνδεση ενός μπουτόν διακόπτη (push-button switch) στην είσοδο επανεκκίνησης. Όταν ο διακόπτης είναι πατημένος, ο μικροελεγκτής κάνει reset.

1.3.6 Διακοπές (Interrupts)

Η διακοπή είναι σημαντική έννοια στους μικροελεγκτές. Μια διακοπή αναγκάζει τον μικροελεγκτή να ανταποκριθεί στα εξωτερικά και εσωτερικά (π.χ. αλλαγή μετώπου στο πόδι RB0 ή υπερχείλιση του χρονιστή 0) συμβάντα πολύ γρήγορα. Συγκεκριμένα, διακόπτεται η εκτέλεση του τρέχοντος προγράμματος και γίνεται αλλαγή της ροής του σε συγκεκριμένη διεύθυνση της μνήμης

προγράμματος για την εκτέλεση ειδικού προγράμματος εξυπηρέτησης της διακοπής και έπειτα συμβαίνει επαναφορά της ροής στην αμέσως επόμενη εντολή από όπου έγινε η διακοπή μετά την ολοκλήρωση του προγράμματος εξυπηρέτησης της διακοπής. Το πρόγραμμα εξυπηρέτησης ονομάζεται ρουτίνα εξυπηρέτησης της διακοπής (ISR, interrupt service routine) ή ρουτίνα διαχείρισης της διακοπής (IH, interrupt handler).

Το ISR ξεκινά από μια σταθερή διεύθυνση της μνήμης του προγράμματος, γνωστή ως vector διεύθυνση της διακοπής. Μερικοί μικροελεγκτές με χαρακτηριστικά πολλών διακοπών (multi-interrupt) έχουν μόνο μια vector διεύθυνση διακοπής, ενώ άλλοι έχουν μια μοναδική vector διεύθυνση για κάθε πηγή διακοπής. Οι διακοπές μπορεί να είναι ένθετες, έτσι ώστε μια νέα διακοπή να μπορεί να αναστείλει την εκτέλεση μιας άλλης διακοπής. Ένα άλλο σημαντικό χαρακτηριστικό της δυνατότητας πολυ-διακοπής, είναι ότι σε διαφορετικές πηγές διακοπής μπορούν να εκχωρηθούν διαφορετικά επίπεδα προτεραιότητας. Για παράδειγμα, η σειρά PIC18F μικροελεγκτών έχει τόσο χαμηλής προτεραιότητας όσο και υψηλής προτεραιότητας επίπεδα διακοπών.

1.3.7 Brown-Out Detector

Οι Brown-out ανιχνευτές, οι οποίοι είναι κοινοί σε πολλούς μικροελεγκτές, κάνουν reset τον μικροελεγκτή αν η τάση τροφοδοσίας πέσει κάτω από μια συγκεκριμένη τιμή. Αυτά τα χαρακτηριστικά ασφαλείας μπορούν να χρησιμοποιηθούν για την αποφυγή απρόβλεπτης λειτουργίας σε χαμηλές τάσεις, ειδικότερα για την προστασία του περιεχομένου των μνημών τύπου EEPROM.

1.3.8 Μετατροπέας Αναλογικό-Σε-Ψηφιακό (Analog-to-Digital Converter)

Ένας μετατροπέας αναλογικό-σε-ψηφιακό (A / D) χρησιμοποιείται για τη μετατροπή ενός αναλογικού σήματος, όπως τάσης, σε ψηφιακή μορφή, ώστε ένας μικροελεγκτής να μπορεί να το διαβάσει και να το επεξεργαστεί. Μερικοί μικροελεγκτές έχουν ενσωματωμένους μετατροπείς A / D. Εξωτερικός A / D μετατροπέας μπορεί επίσης να συνδεθεί με οποιοδήποτε τύπο μικροελεγκτή. Οι A / D μετατροπείς είναι συνήθως 8 έως 10 bits, έχοντας 256 με 1024 επίπεδα κβαντοποίησης. Οι περισσότεροι μικροελεγκτές PIC με A / D χαρακτηριστικά έχουν πολυπλεκτικούς μετατροπείς A / D, τα οποία παρέχουν περισσότερα από ένα αναλογικό κανάλι εισόδου. Για παράδειγμα ο PIC18F4550 έχει 13 A / D κανάλια των 10bit.

Η διαδικασία μετατροπής A / D πρέπει να ξεκινήσει από το πρόγραμμα του χρήστη και μπορεί να διαρκέσει μερικές εκατοντάδες μικρο-δευτερόλεπτα για να ολοκληρωθεί. Οι μετατροπείς A / D παράγουν συνήθως διακοπή, όταν μια μετατροπή έχει ολοκληρωθεί, ώστε το πρόγραμμα χρήστη να μπορεί να διαβάσει τα δεδομένα που έχουν μετατραπεί γρήγορα.

Οι μετατροπείς A / D είναι ιδιαίτερα χρήσιμοι στις εφαρμογές ελέγχου και παρακολούθησης, δεδομένου ότι οι περισσότεροι αισθητήρες (π.χ. αισθητήρες θερμοκρασίας, αισθητήρες πίεσης, αισθητήρες δύναμης, κλπ) παράγουν αναλογικές τάσεις εξόδου.

1.3.9 Σειριακή Είσοδος-Έξοδος

Η σειριακή επικοινωνία (που ονομάζεται επίσης επικοινωνία RS232) επιτρέπει σε έναν μικροελεγκτή να συνδεθεί με έναν άλλον μικροελεγκτή ή σε έναν υπολογιστή χρησιμοποιώντας ένα σειριακό καλώδιο. Μερικοί μικροελεγκτές έχουν ενσωματωμένο hardware που ονομάζεται USART (Universal Synchronous/Asynchronous Receiver/Transmitter) για να εφαρμόσουν μια σειριακή επικοινωνία. Το πρόγραμμα χρήστη μπορεί συνήθως να επιλέξει την ταχύτητα μετάδοσης και την μορφή των δεδομένων. Αν δεν υπάρχει σειριακό υλικό εισόδου-εξόδου, τότε είναι εύκολο να αναπτυχθεί λογισμικό για την σειριακή επικοινωνία δεδομένων, χρησιμοποιώντας οποιαδήποτε pin εισόδου-εξόδου (I / O) του μικροελεγκτή. Η σειρά PIC18F των μικροελεγκτών έχει ενσωματωμένες μονάδες USART. Μερικοί μικροελεγκτές (π.χ., η σειρά PIC18F) συμπεριλαμβάνουν SPI (serial peripheral interface) ή I2C (integrated circuit) λογισμικό σειριακών διαύλων. Αυτά επιτρέπουν σε έναν μικροελεγκτή τη διασύνδεση με άλλες συμβατές συσκευές εύκολα.

1.3.10 EEPROM Μνήμη Δεδομένων (EEPROM Data Memory)

Η μνήμη δεδομένων τύπου EEPROM είναι επίσης πολύ συχνή σε πολλούς μικροελεγκτές. Το πλεονέκτημα μιας μνήμης EEPROM είναι ότι ο προγραμματιστής μπορεί να αποθηκεύσει μη-πτητικά δεδομένα εκεί και να τα αλλάξει όποτε αυτό απαιτείται. Για παράδειγμα, σε μια εφαρμογή παρακολούθησης της θερμοκρασίας, οι μέγιστες και οι ελάχιστες ενδείξεις θερμοκρασίας μπορούν να αποθηκευτούν σε μια μνήμη EEPROM. Εάν η παροχή ρεύματος έχει διακοπεί για κάποιο λόγο, οι τιμές των τελευταίων αναγνώσεων είναι διαθέσιμες στη μνήμη EEPROM. Ο PIC18F4550 έχει 256-byte μνήμη EEPROM, ενώ για παράδειγμα ο PIC18F6680 έχει 1024 bytes.

1.3.11 LCD Drivers

Οι οδηγοί LCD επιτρέπουν σε ένα μικροελεγκτή να συνδεθεί απευθείας με μια εξωτερική οθόνη LCD. Αυτοί οι οδηγοί δεν είναι κοινοί, δεδομένου ότι οι περισσότερες από τις λειτουργίες που παρέχουν μπορούν να υλοποιηθούν σε λογισμικό. Για παράδειγμα ο μικροελεγκτής PIC18F6490 έχει ενσωματωμένο LCD driver module.

1.3.12 Αναλογικός Συγκριτής (Analog Comparator)

Οι αναλογικοί συγκριτές χρησιμοποιούνται όταν πρέπει να συγκριθούν δύο αναλογικές τάσεις. Αν και τα κυκλώματα αυτά εφαρμόζονται στους περισσότερους high-end PIC μικροελεγκτές, δεν είναι κοινοί σε άλλους μικροελεγκτές. Η σειρά PIC18F των μικροελεγκτών έχει ενσωματωμένη μονάδα αναλογικού συγκριτή.

1.3.13 Ρολόι πραγματικού χρόνου (Real-time Clock)

Ένα ρολόι πραγματικού χρόνου επιτρέπει σε έναν μικροελεγκτή να λαμβάνει συνεχώς την απόλυτη πληροφορία περί ώρας και ημερομηνίας. Ενσωματωμένοι real-time clocks δεν είναι κοινοί στους

περισσότερους μικροελεγκτές, δεδομένου ότι η ίδια λειτουργία μπορεί εύκολα να υλοποιηθεί για το σκοπό αυτό είτε από αφιερωμένο τσιπ πραγματικού χρόνου είτε από ένα πρόγραμμα.

1.3.14 Λειτουργία Ύπνου (Sleep Mode)

Μερικοί μικροελεγκτές (π.χ. οι PIC) προσφέρουν ενσωματωμένες λειτουργίες «ύπνου»- αναμονής. Όταν εκτελείται η λειτουργία ύπνου, σταματάει ο εσωτερικός ταλαντωτής και μειώνεται η κατανάλωση ρεύματος σε ένα εξαιρετικά χαμηλό επίπεδο. Ο κύριος σκοπός της λειτουργίας ύπνου είναι να διατηρήσει την ισχύ της μπαταρίας, όταν ο μικροελεγκτής δεν κάνει τίποτα χρήσιμο. Ο μικροελεγκτής συνήθως «ξυπνάει» από την κατάσταση αναμονής με μια εξωτερική επανεκκίνηση (reset) ή watch dog time-out.

1.3.15 Power-on Reset

Μερικοί μικροελεγκτές (π.χ. οι PIC), έχουν ενσωματωμένα κυκλώματα power-on reset, τα οποία διατηρούν τον μικροελεγκτή σε κατάσταση επανεκκίνησης (reset) μέχρι να αρχικοποιηθούν όλα τα εσωτερικά κυκλώματα. Αυτό το χαρακτηριστικό είναι πολύ χρήσιμο, καθώς ξεκινά τον μικροελεγκτή από μια γνωστή κατάσταση κατά την εκκίνηση. Μια εξωτερική επανεκκίνηση μπορεί επίσης να παρέχεται, όπου ο μικροελεγκτής θα βρίσκεται σε κατάσταση επανεκκίνησης όταν πατηθεί ένα εξωτερικό κουμπί.

1.3.16 Λειτουργία Χαμηλής Κατανάλωσης (Low-Power Operation)

Η λειτουργία χαμηλής κατανάλωσης είναι ιδιαίτερα σημαντική σε φορητές εφαρμογές όπου ο εξοπλισμός με τον μικροελεγκτή λειτουργεί από μπαταρίες. Μερικοί μικροελεγκτές (π.χ. οι PIC), μπορούν να λειτουργούν με λιγότερο από 2mA από παροχή των 5V και γύρω στα 15μΑ από παροχή των 3V. Άλλοι μικροελεγκτές, ειδικά συστήματα βασισμένα σε μικροεπεξεργαστή με διάφορα τσιπ, μπορεί να καταναλώνουν αρκετές εκατοντάδες mA ή ακόμη περισσότερο.

1.3.17 Δυνατότητα Πηγής Ρεύματος (Current Sink/Source Capability)

Η δυνατότητα πηγής ρεύματος είναι σημαντική σε περίπτωση που ο μικροελεγκτής συνδεθεί σε μια εξωτερική συσκευή, η οποία απορροφά μεγάλη ποσότητα ρεύματος για να λειτουργήσει. Οι PIC μικροελεγκτές μπορούν να προμηθεύσουν 25mA ρεύματος από κάθε ακροδέκτη εξόδου pin. Το ρεύμα αυτό είναι συνήθως επαρκές για να οδηγηθούν LED, μικρές λάμπες, βομβητές, μικρά ρελέ κ.α. Η παροχή ρεύματος μπορεί να αυξηθεί με την σύνδεση εξωτερικών μεταγωγικών κυκλωμάτων τρανζίστορ ή ρελέ στους ακροδέκτες εξόδου pins.

1.3.18 Διεπαφή USB (USB Interface)

Η διεπαφή USB είναι σήμερα πολύ δημοφιλής προδιαγραφή διασύνδεσης υπολογιστών που χρησιμοποιείται για τη σύνδεση διαφόρων περιφερειακών συσκευών με τους υπολογιστές και τους

μικροελεγκτές. Μερικοί PIC μικροελεγκτές παρέχουν ενσωματωμένες μονάδες USB, όπως για παράδειγμα ο PIC18F2x50.

1.3.19 Διεπαφή Ελέγχου Κινητήρα (Motor Control Interface)

Μερικοί PIC μικροελεγκτές, όπως για παράδειγμα ο PIC18F2x31 παρέχουν τη διεπαφή ελέγχου του κινητήρα.

1.3.20 Διεπαφή CAN (CAN interface)

Ο διάλογος CAN είναι ένα πολύ δημοφιλές σύστημα διαύλου που χρησιμοποιείται κυρίως σε εφαρμογές αυτοματισμού. Μερικοί μικροελεγκτές της σειράς PIC18F (π.χ. ο PIC18F4680) παρέχουν την διεπαφή CAN.

1.3.21 Διεπαφή Ethernet (Ethernet interface)

Μερικοί PIC μικροελεγκτές (π.χ. ο PIC18F97J60) παρέχουν τη διεπαφή Ethernet και έτσι χρησιμοποιούνται εύκολα σε εφαρμογές που βασίζονται σε δίκτυο.

1.3.22 Διεπαφή ZigBee (ZigBee interface)

Η διεπαφή ZigBee, μια διεπαφή παρόμοια με Bluetooth, χρησιμοποιείται σε εφαρμογές ασύρματου οικιακού αυτοματισμού χαμηλού κόστους. Κάποιοι μικροελεγκτές της σειράς PIC18F παρέχουν την δυνατότητα ZigBee διασύνδεσης, κάνοντας πολύ εύκολο το σχεδιασμό τέτοιων ασύρματων συστημάτων.

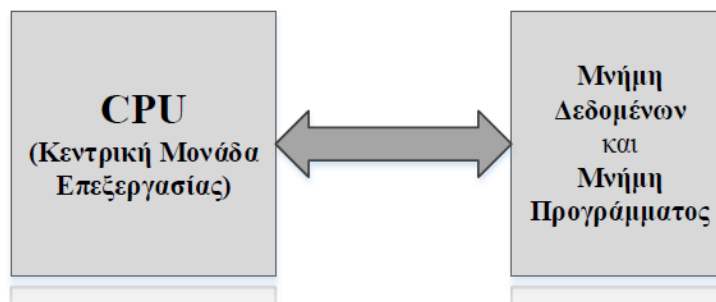
1.4. Η αρχιτεκτονική των μικροελεγκτών PIC

1.4.1 Η αρχιτεκτονική Von Neumann και Harvard

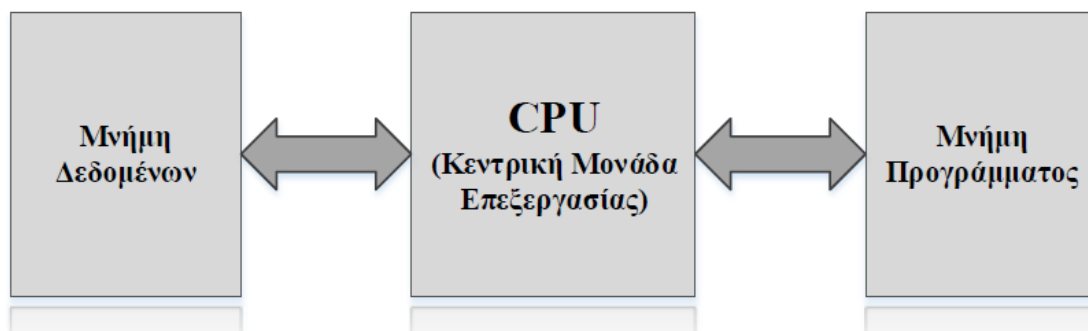
Οι δύο βασικές αρχιτεκτονικές που χρησιμοποιούνται στους μικροελεγκτές είναι η αρχιτεκτονική Von Neumann και η αρχιτεκτονική Harvard.

Η αρχιτεκτονική των περισσότερων μικροελεγκτών βασίζεται στη δομή του Von Neumann. Όπως φαίνεται στο σχήμα 1.2, οι εντολές του προγράμματος καθώς και τα δεδομένα αποθηκεύονται σε μία μνήμη. Αυτό σημαίνει ότι υπάρχει μόνο ένας διάυλος μνήμης, κάτι που αποτελεί ένα βασικό μειονέκτημα της αρχιτεκτονικής αυτής, καθώς ο μικροελεγκτής δεν θα μπορεί κάποια δεδομένη στιγμή να κάνει παραπάνω από μία προσπελάσεις μνήμης, αφού οι εντολές και τα δεδομένα μοιράζονται τον ίδιο διάυλο.

Η αρχιτεκτονική Harvard από την άλλη (σχήμα 1.3), η οποία χρησιμοποιείται από τους PIC μικροελεγκτές, αποτελείται από μία ξεχωριστή μνήμη για τις εντολές του προγράμματος και μια άλλη ξεχωριστή για τα δεδομένα και τους καταχωρητές που ελέγχουν τα περιφερειακά του μικροελεγκτή. Αυτό σημαίνει ότι οι εντολές και τα δεδομένα μπορούν να κινούνται ταυτόχρονα σε ξεχωριστούς διαύλους, εξασφαλίζοντας έτσι μεγαλύτερη ταχύτητα και απόδοση.



Σχήμα 1.2 Αρχιτεκτονική Von Neumann

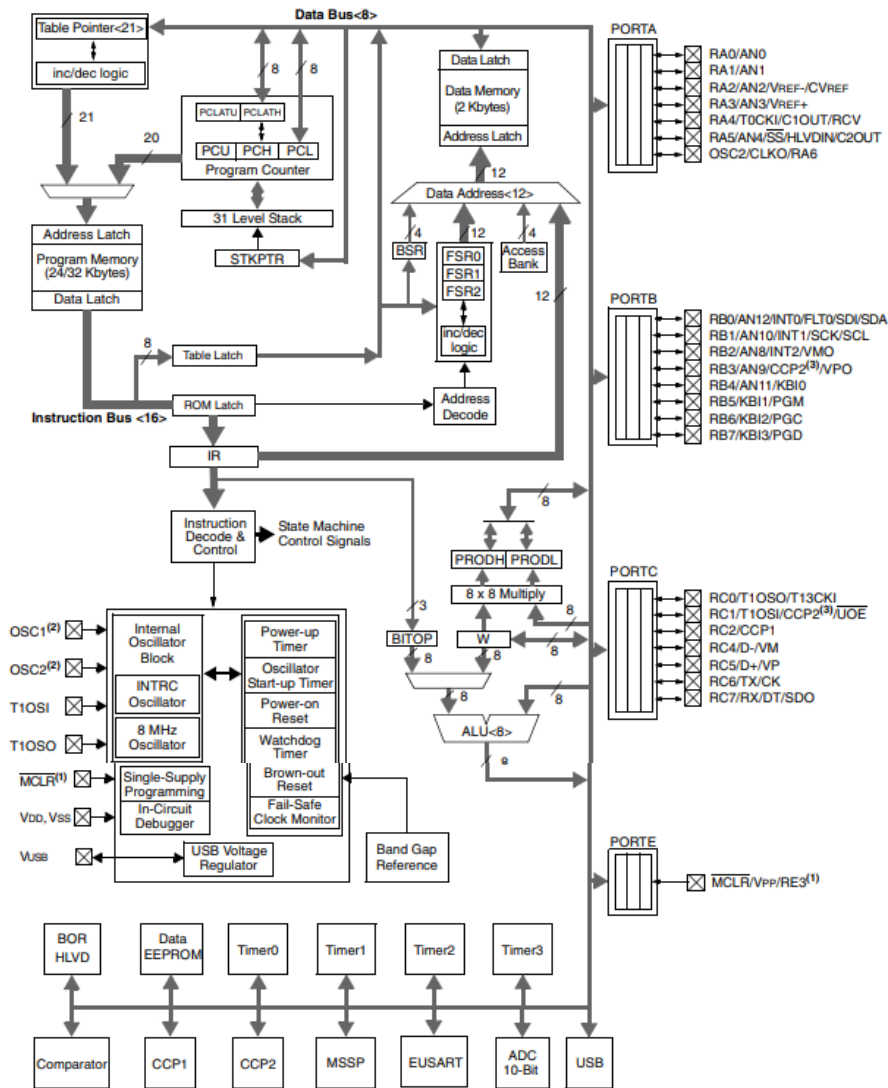


Σχήμα 1.3 Αρχιτεκτονική Harvard

1.4.2 Η εσωτερική αρχιτεκτονική των μικροελεγκτών PIC

Στο σημείο αυτό, μπορούμε να χωρίσουμε τους μικροελεγκτές σε δύο μεγάλες κατηγορίες. Η πρώτη αφορά την αρχιτεκτονική CISC ή διευρυμένου ρεπερτορίου εντολών και η δεύτερη RISC ή αλλιώς μειωμένου ρεπερτορίου εντολών. Κατά την πρώτη αρχιτεκτονική, αρκετές πολύπλοκες εντολές περιλαμβάνονται στο ρεπερτόριο εντολών προκειμένου να εκτελούνται με όσο το δυνατόν λιγότερες εντολές ακόμη και οι πιο πολύπλοκες εντολές. Όσο μεγαλώνει αυτό το ρεπερτόριο όμως, τόσο πιο δύσκολη και σύνθετη γίνεται η σχεδίαση του μικροελεγκτή, γεγονός που οδηγεί στην καθυστέρηση της εκτέλεσης κάθε εντολής. Επίσης, τα δεδομένα και ο κώδικας βρίσκονται στον ίδιο δίαυλο και δεν μπορούν να ανακληθούν ταυτόχρονα.

Έτσι, προέκυψε η δεύτερη αρχιτεκτονική RISC, στην οποία ανήκουν οι μικροελεγκτές PIC και κατασκευάζεται με όσο το δυνατόν λιγότερες εντολές. Αυτό έχει σαν αποτέλεσμα κάθε εντολή να εκτελείται συχνά και γρήγορα. Ακόμη και οι πιο απλές διαδικασίες, εκτελούνται με συνδυασμό των απλών εντολών.



Σχήμα 1.4 Η εσωτερική αρχιτεκτονική του μικροελεγκτή PIC18F4550

Κεφάλαιο 2^ο Μικροελεγκτές της σειράς PIC18F

Οι περιορισμοί στη μνήμη προγράμματος και δεδομένων, καθώς επίσης και στις εντολές της σειράς PIC16F, οδήγησαν στην ανάπτυξη της σειράς PIC18F. Η συγκεκριμένη σειρά περιέχει περισσότερες εντολές, μεγαλύτερη μνήμη προγράμματος-δεδομένων και υποστηρίζει διασυνδέσεις όπως USB, διάυλος CAN κ.α., που η προηγούμενη σειρά δεν υποστήριζε. Η ανάπτυξη αυτής της σειράς είναι χρήσιμη σε πολύπλοκες εφαρμογές, καθώς επίσης και σε εφαρμογές γενικού σκοπού. Κάποια βασικά χαρακτηριστικά της σειράς αυτής είναι:

- ❖ Περιέχει 77 εντολές
- ❖ Μνήμη προγράμματος έως 2Mbytes
- ❖ Μνήμη δεδομένων έως 4Kbytes
- ❖ Λειτουργία DC έως 40MHz
- ❖ Προτεραιότητα διακοπών
- ❖ 8 × 8 hardware πολλαπλασιαστή
- ❖ Έως δύο timers/counters των 8bit
- ❖ Έως τρεις timers/counters των 16bit
- ❖ Έως τέσσερις εξωτερικές διακοπές
- ❖ Παράλληλες πόρτες (PSP)
- ❖ Υψηλό ρεύμα πηγής (25mA)
- ❖ Γρήγορο μετατροπέα αναλογικού σε ψηφιακό (10bit)
- ❖ Προγραμματισμός εντός του κυκλώματος
- ❖ Διαθέτει ενεργοποίηση επαναφοράς (Power-on reset), χρονοδιακόπτη ενεργοποίησης (power-uprtimer) και χρονοδιακόπτη εκκίνησης του ταλαντωτή (oscillator start-uprtimer)
- ❖ Χρονοδιακόπτη WDT με ταλαντωτή RC πάνω στο ολοκληρωμένο
- ❖ Προγραμματιζόμενη μονάδα ανίχνευσης χαμηλής τάσης (LVD)
- ❖ Κύρια συγχρονισμένη μονάδα σειριακής θύρας (λειτουργία SPI και I2C)

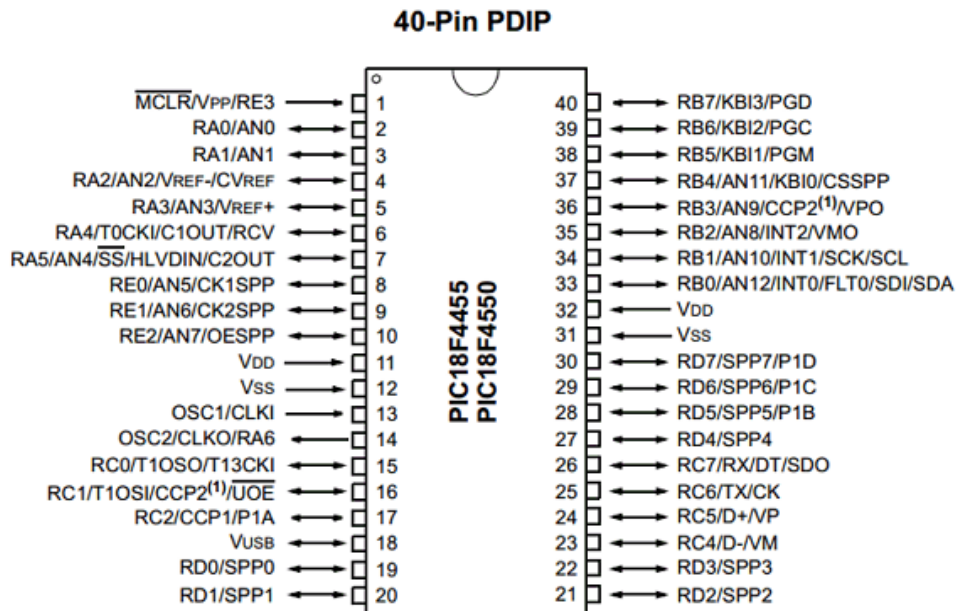
Επιπλέον, ορισμένοι μικροελεγκτές αυτής της σειράς διαθέτουν και τα εξής χαρακτηριστικά:

- ❖ Άμεση διεπαφή διαύλου CAN 2.0
- ❖ Άμεση διεπαφή διαύλου USB 2.0
- ❖ Άμεση διασύνδεση ελέγχου LCD
- ❖ TCP/IP διασύνδεση
- ❖ ZigBee διασύνδεση
- ❖ Άμεση διεπαφή ελέγχου κινητήρα

2.1 Η αρχιτεκτονική του μικροελεγκτή PIC18F4550

2.1.1 Ακροδέκτες

Στο ακόλουθο σχήμα 2.1 φαίνεται πως είναι διαμορφωμένοι οι ακροδέκτες του PIC18F4550 σε DIP μορφή:

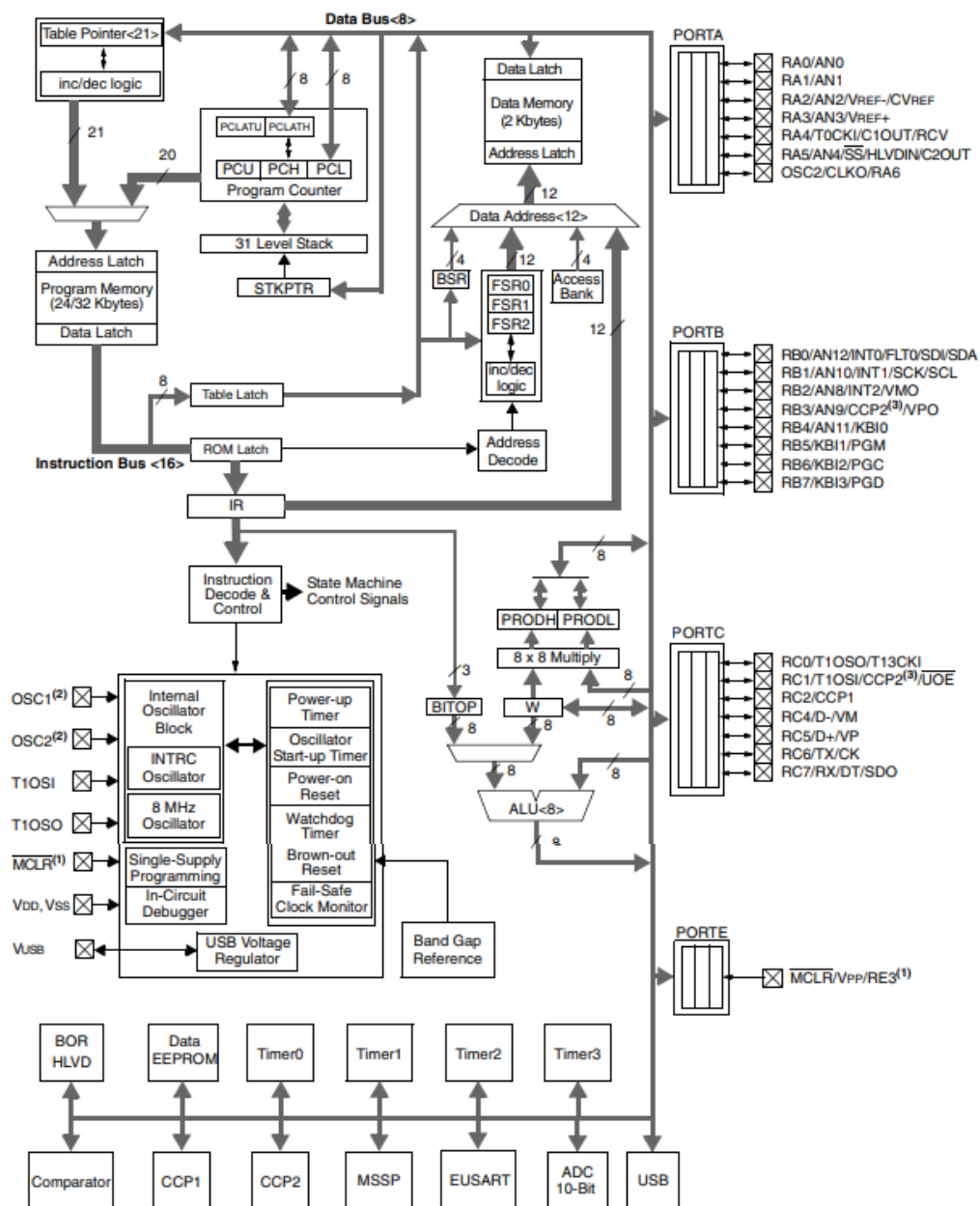


Σχήμα 2.1 Ακροδέκτες PIC18F4550

2.1.2 Εσωτερικό μπλοκ διάγραμμα

Στο σχήμα 2.2 απεικονίζεται το εσωτερικό μπλοκ διάγραμμα του μικροελεγκτή. Στο κέντρο βρίσκεται η κεντρική μονάδα επεξεργασίας (CPU), η οποία αποτελείται από μία λογική και αριθμητική μονάδα των 8-bit (ALU), έναν 8×8 hardware πολλαπλασιαστή και έναν καταχωρητή w (Working Register) των 8 bit. Έστω, ότι θέλουμε να στείλουμε μια τιμή σε μία θέση μνήμης. Θα πρέπει αυτήν την τιμή πρώτα να την φορτώσουμε στον καταχωρητή και μετά να την στείλουμε στην συγκεκριμένη διεύθυνση που θέλουμε. Μία ακόμα χρήση του καταχωρητή w είναι η αποθήκευση κάποιων τιμών κατά την εκτέλεση μιας αριθμητικής και λογικής πράξης.

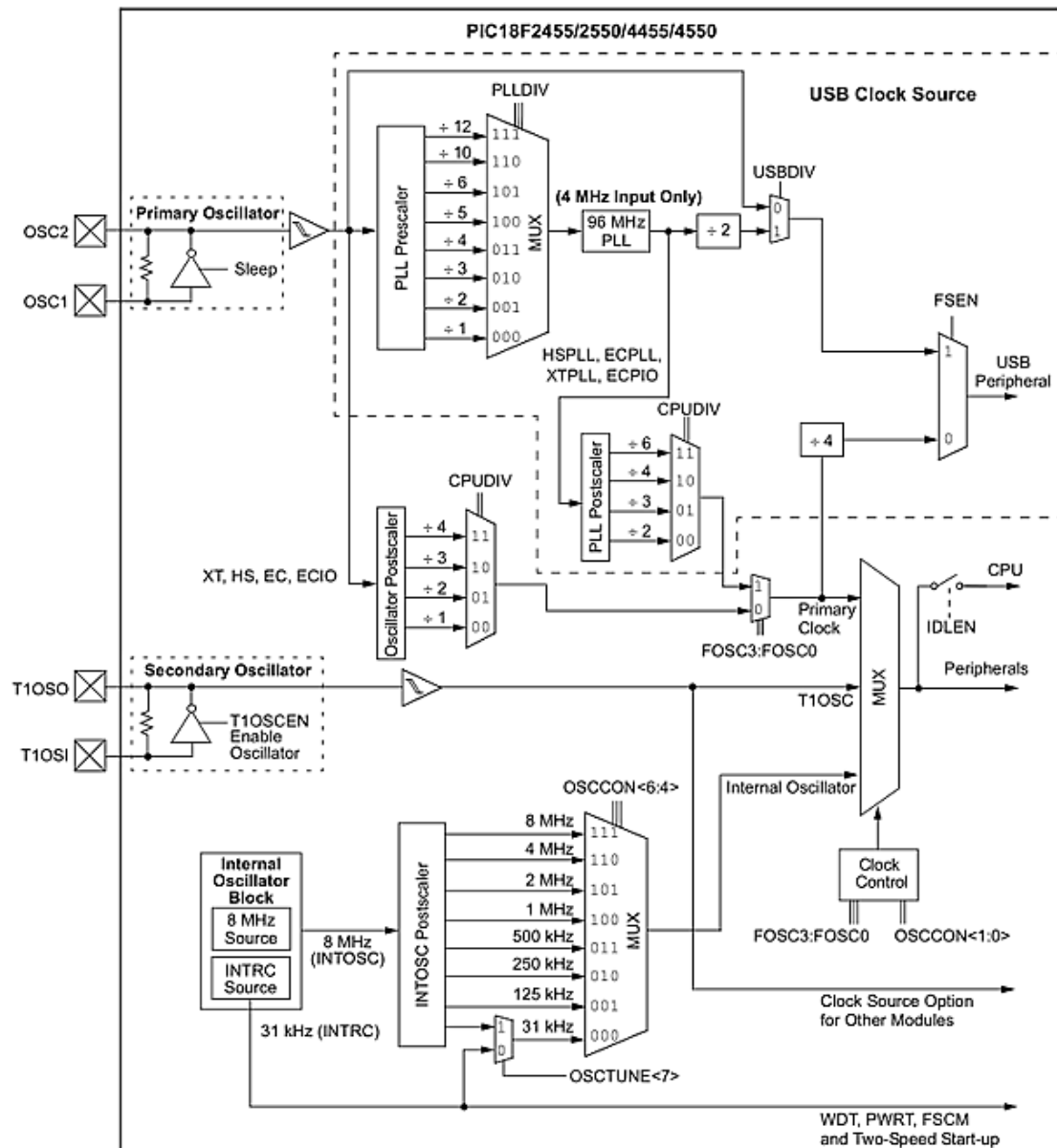
FIGURE 1-1: PIC18F2455/2550 (28-PIN) BLOCK DIAGRAM



Σχήμα 2.2 Εσωτερικό μπλοκ διάγραμμα

2.1.3 Διάγραμμα του Clock

Για την επιτάχυνση της συνολικής λειτουργίας του μικροελεγκτή, οι PIC18F4550 διαθέτουν ένα νέο κύκλωμα PLL, το οποίο δίνει την δυνατότητα πολλαπλασιασμού της συχνότητας του ταλαντωτή. Το κύκλωμα αυτό περιέχει έναν in-circuit debugger (αποσφαλματωτή), ο οποίος χρησιμεύει για την επιστροφή διαγνωστικών δεδομένων και τιμών κατά τη διάρκεια της εκτέλεσης του προγράμματος. Επίσης, σε περίπτωση αιφνίδιας διακοπής του προγράμματος, το κύκλωμα διαθέτει έναν watchdog timer, ο οποίος κάνει επανεκκίνηση του προγράμματος.

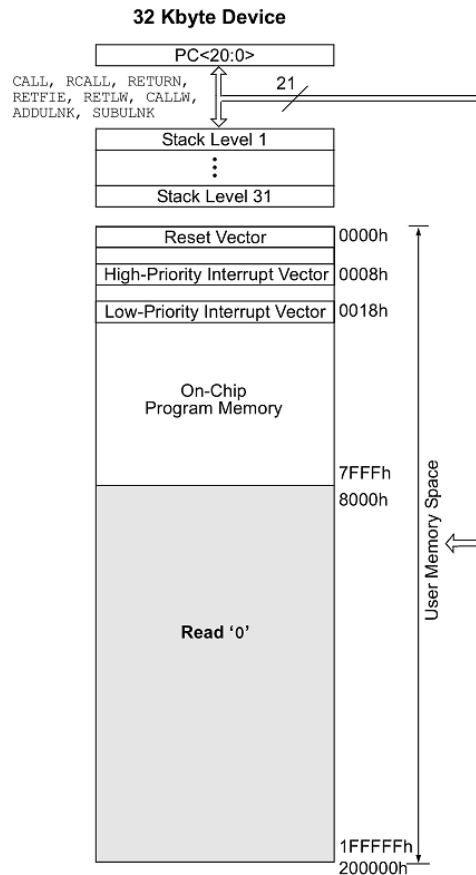


Σχήμα 2.3 Διάγραμμα clock

2.1.4 Η Οργάνωση της Μνήμης του Προγράμματος

Σε όλες τις συσκευές υπάρχει ένας μετρητής προγράμματος 21-bit. Οι διαθέσιμες θέσεις μνήμης στις οποίες μπορεί να απευθυνθεί ο μετρητής είναι από την διεύθυνση 0000h έως τη διεύθυνση 7FFFh, διότι οι θέσεις μνήμης 8000h - 1FFFFFFh δεν υπάρχουν. Συνεπώς, με μέγεθος 15 bit υπάρχουν 215 διαθέσιμες θέσεις μνήμης. Ο PIC18F4550 περιέχει Flash μνήμη 32Kbyte με λέξεις των 2 ή 4 bytes και μπορούν να αποθηκευτούν 16384 μονές εντολές.

Οι διευθύνσεις για κλήσεις υπορουτίνας και επεξεργασία διακοπών αποθηκεύονται σε μια 31-entry stack (στοίβα), η οποία δεν αποτελεί μέρος της μνήμης προγράμματος ή της μνήμης δεδομένων. Ένας δείκτης ελέγχει τη στοίβα. Σε κάθε κλήση της υπορουτίνας, ο δείκτης αυξάνεται και το περιεχόμενο του μετρητή γράφεται στη θέση μνήμης που υποδεικνύεται. Σε κάθε επιστροφή της κλήσης υπορουτίνας, η θέση μνήμης που δείχνει ο δείκτης μειώνεται.



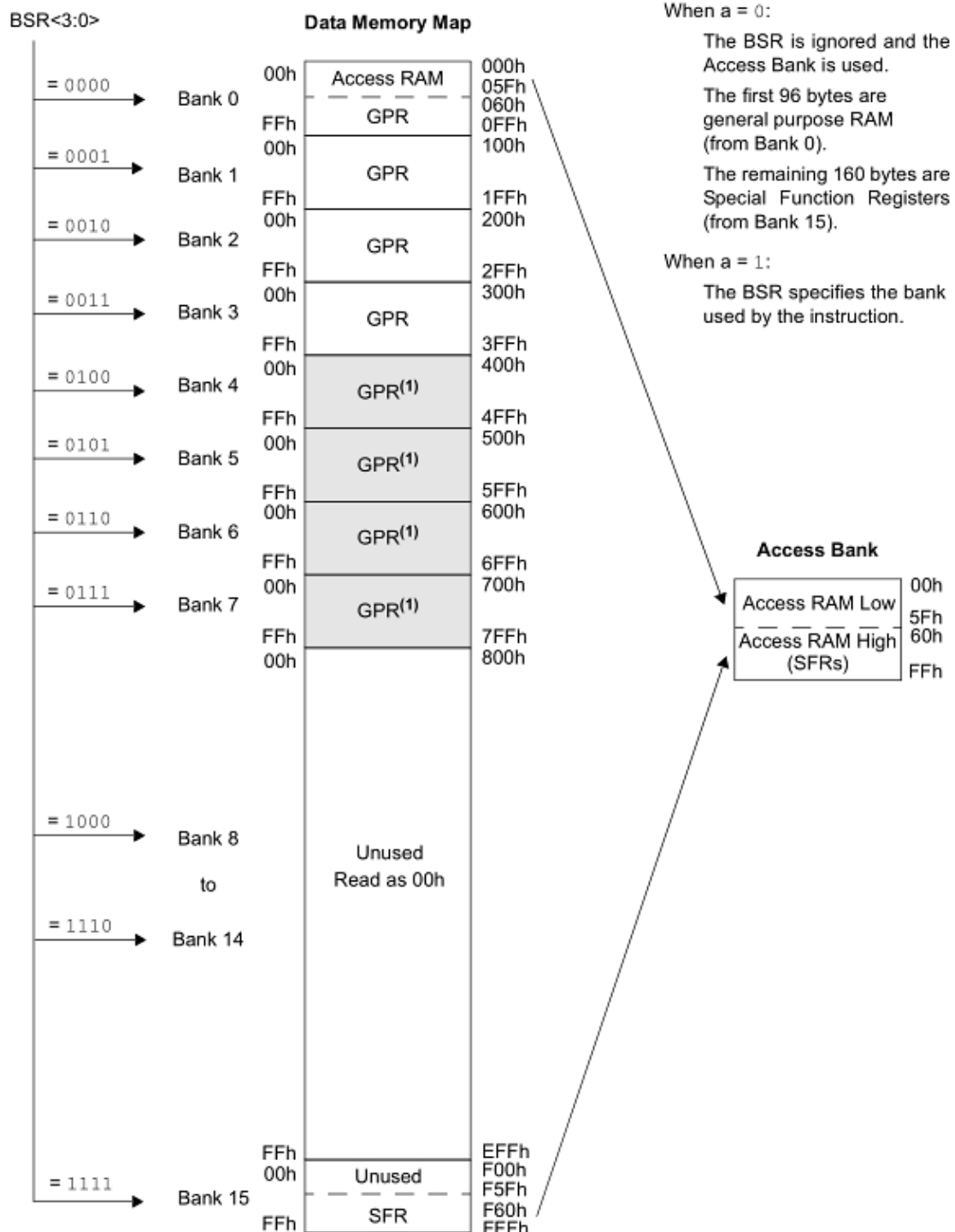
Σχήμα 2.4 Μνήμη προγράμματος του μικροελεγκτή PIC18F4550

2.1.5 Η Οργάνωση της Μνήμης Δεδομένων

Στη μνήμη δεδομένων κάθε καταχωρητής αποτελείται από 12 bits διευθύνσεων, δηλαδή υπάρχουν έως και 4096 bytes δεδομένων. Η μνήμη χωρίζεται σε 16 περιοχές (banks), κάθε μία από τις οποίες περιέχει 256bytes. Οι PIC18F αποτελούνται από 8 πλήρεις περιοχές, με συνολικά 2048 bytes. Η μεταγωγή τους γίνεται όταν χρησιμοποιούνται μεταγωγιστές γλωσσών υψηλού επιπέδου, γίνεται αυτόματα. Οι θέσεις μνήμης ξεκινούν από τη διεύθυνση 000h έως την διεύθυνση FFFh . Περιέχουν 212 θέσεις , δηλαδή 22×210 θέσεις = 4 Kbyte.

Οι SFR (Special Function Registers) είναι καταχωρητές που έχουν μια ειδική λειτουργία για τον μικροελεγκτή και είναι θέσεις μνήμης στη RAM. Οι καταχωρητές δεδομένων των πορτών και οι καταχωρητές που σχετίζονται με τον έλεγχο και τη λειτουργία των χρονιστών (timers) ανήκουν στην κατηγορία των καταχωρητών ειδικής λειτουργίας. Σε αυτή την κατηγορία ανήκουν καταχωρητές που ελέγχουν λειτουργίες όπως περιφερειακές συσκευές, χρονιστές/μετρητές, μετατροπείς A / D, USART. Ακόμη, στην κατηγορία των καταχωρητών ειδικής λειτουργίας ανήκουν και οι καταχωρητές κατεύθυνσης των πορτών. Για παράδειγμα, στη θέση F95h βρίσκεται ο καταχωρητής κατεύθυνσης της πόρτας D. Η τιμή που αποθηκεύεται σε αυτή τη θέση μνήμης καθορίζει ποιοι ακροδέκτες της πόρτας θα είναι είσοδοι και ποιοι θα είναι έξοδοι. Όπως φαίνεται από τον χάρτη μνήμης του σχήματος 2.6, οι καταχωρητές αυτοί ευρίσκονται στην περιοχή από την διεύθυνση F60h έως την διεύθυνση FFFh.

Για παράδειγμα, ο TRISD είναι ο καταχωρητής κατεύθυνσης της πόρτας D και η τιμή του καθορίζει ποιοι ακροδέκτες της πόρτας D θα είναι είσοδοι και ποιοι θα είναι έξοδοι. Όταν σε ένα bit του καταχωρητή υπάρχει 1, ο αντίστοιχος ακροδέκτης της πόρτας D είναι είσοδος. Όταν υπάρχει 0, ο αντίστοιχος ακροδέκτης της πόρτας D είναι έξοδος. Ο καταχωρητής PORTD χρησιμοποιείται για να διαβάσουμε δεδομένα από τους ακροδέκτες της πόρτας D που χρησιμοποιούνται σαν είσοδοι και για να στείλουμε δεδομένα προς τους ακροδέκτες της πόρτας D που χρησιμοποιούνται σαν έξοδοι.



Σχήμα 2.5 Μνήμη δεδομένων του μικροελεγκτή PIC18F4550

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDfH	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCCh	CCPR2H	F9Ch	— ⁽²⁾	F7Ch	UEP12
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	UEP11
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	— ⁽²⁾	F7Ah	UEP10
FF9h	PCL	FD9h	FSR2L	FB9h	— ⁽²⁾	F99h	— ⁽²⁾	F79h	UEP9
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	— ⁽²⁾	F78h	UEP8
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP1DEL	F97h	— ⁽²⁾	F77h	UEP7
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE ⁽³⁾	F76h	UEP6
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD ⁽³⁾	F75h	UEP5
FF4h	PRODH	FD4h	— ⁽²⁾	FB4h	CMCON	F94h	TRISC	F74h	UEP4
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	UEP3
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA	F72h	UEP2
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	— ⁽²⁾	F71h	UEP1
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	— ⁽²⁾	F70h	UEP0
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG	F8Fh	— ⁽²⁾	F6Fh	UCFG
FEh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FAEh	RCREG	F8Eh	— ⁽²⁾	F6Eh	UADDR
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG	F8Dh	LATE ⁽³⁾	F6Dh	UCON
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD ⁽³⁾	F6Ch	USTAT
FEbH	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FEAh	FSR0H	FCAh	T2CON	FAAh	— ⁽²⁾	F8Ah	LATB	F6Ah	UEIR
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA	F69h	UIE
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	— ⁽²⁾	F68h	UIR
FE7h	INDF1 ⁽¹⁾	FC7h	SSPSTAT	FA7h	EECON2 ⁽¹⁾	F87h	— ⁽²⁾	F67h	UFRMH
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSPCON1	FA6h	EECON1	F86h	— ⁽²⁾	F66h	UFRML
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSPCON2	FA5h	— ⁽²⁾	F85h	— ⁽²⁾	F65h	SPPCON ⁽³⁾
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	— ⁽²⁾	F84h	PORTE	F64h	SPPEPS ⁽³⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	— ⁽²⁾	F83h	PORTD ⁽³⁾	F63h	SPPCFG ⁽³⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SPPDATA ⁽³⁾
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	— ⁽²⁾
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	— ⁽²⁾

- Note 1:** Not a physical register.
Note 2: Unimplemented registers are read as '0'.
Note 3: These registers are implemented only on 40/44-pin devices.

Σχήμα 2.6 Χάρτης SFR

Χρονιστής T0CON (timer0)

Ο χρονιστής T0CON (timer0) είναι ένας καταχωρητής των 8 bit ή των 16 bit (επιλέγεται από τον χρήστη) του οποίου η τιμή ανά σταθερό χρονικό διάστημα, αυξάνεται κατά 1. Το χρονικό διάστημα αύξησης κατά 1 εξαρτάται από την συχνότητα του ρολογιού του μικροελεγκτή και από την τιμή ενός προγραμματιζόμενου διαιρέτη συχνότητας. Κάθε φορά που γίνεται μια μετάβαση από οκτώ άσους (ή 16 άσους αν ο χρονιστής λειτουργεί σαν 16 bit) σε 8 μηδενικά (ή 16 μηδενικά αν ο χρονιστής λειτουργεί σαν 16 bit) το bit 2 του καταχωρητή INTCON (Interrupt Control), το οποίο ονομάζεται TMR0IF (Timer0 Interrupt Flag), γίνεται 1 (SET).

2.1.6 Τάση και Ρεύμα Τροφοδοσίας (Supply Voltage/Current)

DC Characteristics: Supply Voltage
PIC18F2455/2550/4455/4550 (Industrial)
PIC18LF2455/2550/4455/4550 (Industrial)

PIC18F2455/2550/4455/4550 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC18F2455/2550/4455/4550 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
D001	VDD	Supply Voltage	2.0 ⁽²⁾	—	5.5	V	EC, HS, XT and Internal Oscillator modes
			3.0 ⁽²⁾	—	5.5	V	HSPLL, XTPLL, ECPIO and ECPLL Oscillator modes
D002	VDR	RAM Data Retention Voltage ⁽¹⁾	1.5	—	—	V	
D003	VPOR	VDD Start Voltage to Ensure Internal Power-on Reset Signal	—	—	0.7	V	See Section 4.3 "Power-on Reset (POR)" for details
D004	SVDD	VDD Rise Rate to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	See Section 4.3 "Power-on Reset (POR)" for details
D005	VBOR	Brown-out Reset Voltage					
		BORV1:BORV0 = 11	2.00	2.05	2.16	V	
		BORV1:BORV0 = 10	2.65	2.79	2.93	V	
		BORV1:BORV0 = 01	4.11	4.33	4.55	V	
		BORV1:BORV0 = 00	4.36	4.59	4.82	V	

Legend: Shading of rows is to assist in readability of the table.

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

Note 2: The stated minimums apply for the PIC18LF products in this device family. PIC18F products in this device family are rated for 4.2V minimum in all oscillator modes.

Σχήμα 2.7 Απαιτήσεις για τάση στον PIC18F4550

DC Characteristics: Power-Down and Supply Current
PIC18F2455/2550/4455/4550 (Industrial)
PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

PIC18LF2455/2550/4455/4550 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial							
PIC18F2455/2550/4455/4550 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial							
Param No.	Symbol	Device	Typ	Max	Units	Conditions			
Supply Current (I_{DD})⁽²⁾									
		PIC18LFX455/X550	250	500	μA	-40°C	$V_{DD} = 2.0\text{V}$	Fosc = 1 MHz (PRI_RUN, EC oscillator)	
			250	500	μA	$+25^{\circ}\text{C}$			
			250	500	μA	$+85^{\circ}\text{C}$			
		PIC18LFX455/X550	550	650	μA	-40°C			$V_{DD} = 3.0\text{V}$
			480	650	μA	$+25^{\circ}\text{C}$			
			460	650	μA	$+85^{\circ}\text{C}$			
		All devices	1.2	1.6	mA	-40°C	$V_{DD} = 5.0\text{V}$		
			1.1	1.5	mA	$+25^{\circ}\text{C}$			
			1.0	1.4	mA	$+85^{\circ}\text{C}$			
		PIC18LFX455/X550	0.74	2.0	mA	-40°C	$V_{DD} = 2.0\text{V}$		Fosc = 4 MHz (PRI_RUN, EC oscillator)
			0.74	2.0	mA	$+25^{\circ}\text{C}$			
			0.74	2.0	mA	$+85^{\circ}\text{C}$			
PIC18LFX455/X550	1.3	3.0	mA	-40°C	$V_{DD} = 3.0\text{V}$				
	1.3	3.0	mA	$+25^{\circ}\text{C}$					
	1.3	3.0	mA	$+85^{\circ}\text{C}$					
All devices	2.7	6.0	mA	-40°C	$V_{DD} = 5.0\text{V}$				
	2.6	6.0	mA	$+25^{\circ}\text{C}$					
	2.5	6.0	mA	$+85^{\circ}\text{C}$					
All devices	15	35	mA	-40°C	$V_{DD} = 4.2\text{V}$	Fosc = 40 MHz (PRI_RUN, EC oscillator)			
	16	35	mA	$+25^{\circ}\text{C}$					
	16	35	mA	$+85^{\circ}\text{C}$					
All devices	21	40	mA	-40°C	$V_{DD} = 5.0\text{V}$				
	21	40	mA	$+25^{\circ}\text{C}$					
	21	40	mA	$+85^{\circ}\text{C}$					
All devices	20	40	mA	-40°C	$V_{DD} = 4.2\text{V}$	Fosc = 48 MHz (PRI_RUN, EC oscillator)			
	20	40	mA	$+25^{\circ}\text{C}$					
	20	40	mA	$+85^{\circ}\text{C}$					
All devices	25	50	mA	-40°C	$V_{DD} = 5.0\text{V}$				
	25	50	mA	$+25^{\circ}\text{C}$					
	25	50	mA	$+85^{\circ}\text{C}$					

Legend: Shading of rows is to assist in readability of the table.

- Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS} and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).
- 2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
 The test conditions for all I_{DD} measurements in active operation mode are:
 OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD} or V_{SS} ;
 MCLR = V_{DD} ; WDT enabled/disabled as specified.
- 3:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to $+70^{\circ}\text{C}$. Extended temperature crystals are available at a much higher cost.
- 4:** BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

Σχήμα 2.8 Απαιτήσεις για ρεύμα στον PIC18F4550

Όπως φαίνεται στα παραπάνω σχήματα 2.7-2.8, ο PIC18F4550 λειτουργεί με τάση τροφοδοσίας από +4.2V έως +5.5V σε όλες τις λειτουργίες του ταλαντωτή.

2.1.7 Επανεκκίνηση (Reset)

Η λειτουργία της επανεκκίνησης αποτελεί έναν τρόπο ο μικροελεγκτής να ξεκινήσει την εκτέλεση του προγράμματος από την διεύθυνση 0000h της μνήμης του προγράμματος. Η επανεκκίνηση μπορεί να πραγματοποιηθεί κατά τη διάρκεια μιας από τις ακόλουθες λειτουργίες:

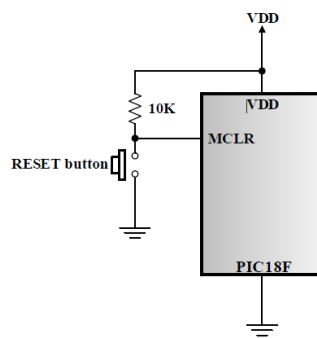
- ❖ Power-on reset (POR)
- ❖ MCLR reset

- ❖ Watchdog timer (WDT) reset
- ❖ Brown-out reset (BOR)
- ❖ RESET instruction
- ❖ Stack full reset
- ❖ Stack underflow reset

Στο σημείο αυτό ακολουθεί περιγραφή των δύο πρώτων τρόπων.

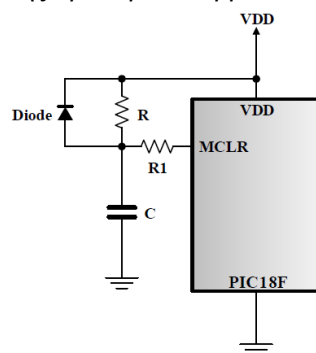
Power-on reset (POR)

Η επανεκκίνηση power-on πραγματοποιείται αυτόματα μόλις εφαρμοστεί η τάση τροφοδοσίας στον μικροελεγκτή. Ο ακροδέκτης MCLR συνδέεται απευθείας ή καλύτερα μέσω μιας pull-up αντίστασης 10K στη τάση τροφοδοσίας (VDD). Στο σχήμα 2.9, η επανεκκίνηση πραγματοποιείται μόλις πατηθεί το reset button.



Σχήμα 2.9 Παράδειγμα 1 power-on reset

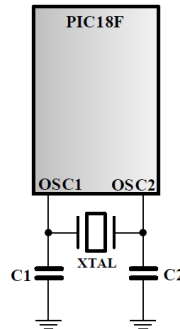
Υπάρχουν περιπτώσεις όμως όπου υπάρχει αργή άνοδος της τάσης τροφοδοσίας. Σε αυτές τις εφαρμογές είναι σωστό να χρησιμοποιείται μία διόδος, ένας πυκνωτής και μια επιπλέον αντίσταση ακριβώς όπως φαίνεται στο σχήμα 2.10. Η αντίσταση R προτείνεται να είναι μικρότερη από 40 kΩ, ώστε να εξασφαλιστεί ότι η πτώση τάσης στα άκρα της R δεν θα παραβιάσει τις προδιαγραφές της συσκευής. Επιπλέον, η αντίσταση R1 προτείνεται να είναι από 100 Ω έως 1 kΩ. Η αντίσταση αυτή θα περιορίσει κάθε ρεύμα προς τον ακροδέκτη MCLR από τον εξωτερικό πυκνωτή σε περίπτωση ηλεκτροστατικής εκκένωσης. Τέλος, η διόδος βοηθάει στην γρήγορη εκφόρτιση του πυκνωτή, μόλις η VDD απενεργοποιηθεί. Το κύκλωμα αυτό απαιτείται μόνο εάν η συσκευή εξέλθει από την επανεκκίνηση πριν η VDD της συσκευής προλάβει να βρεθεί στο έγκυρο εύρος λειτουργίας.



Σχήμα 2.10 Παράδειγμα 2 power-on reset

2.1.8 Clock

Για την εκτέλεση των εντολών, καθώς και για τη λειτουργία των περιφερειακών του μικροελεγκτή είναι απαραίτητο το ρολόι του συστήματος. Κάθε τέσσερις περίοδοι του παράγουν έναν εσωτερικό κύκλο ρολογιού. Το ρολόι αυτό μπορεί να είναι είτε ένας εξωτερικός κρύσταλλος είτε ένα κεραμικό αντηχείο που συνδέεται στους ακροδέκτες OSC1 και OSC2.



Σχήμα 2.11 Παράδειγμα ρολογιού με κρύσταλλο

Επιπλέον, μια εξωτερική αντίσταση μαζί με έναν πυκνωτή, καθώς και εσωτερικοί ταλαντωτές χρησιμοποιούνται για να παρέχουν παλμούς ρολογιού στον μικροελεγκτή.

Το εξωτερικό ρολόι μπορεί να παραχθεί σε μια από τις οκτώ διαφορετικές λειτουργίες του ταλαντωτή. Η επιλογή της λειτουργίας του ταλαντωτή γίνεται μέσω των bits διαμόρφωσης.

EC	External Clock
ECIO	External Clock with IO pin enabled
LP	Low Frequency (Power) Crystal
XT	Crystal/Resonator
HS	High Speed Crystal/Resonator
RC	External Resistor/Capacitor
RCIO	External Resistor/Capacitor with IO pin enabled
HS4	High Speed Crystal/Resonator with 4x frequency PLL multiplier enabled

Σχήμα 2.12 Οι 8 λειτουργίες του ταλαντωτή

2.1.9 Μετατροπές από αναλογικό σε ψηφιακό σήμα (A / D Converter)

Ένα πολύ σημαντικό περιφερειακό ενός μικροελεγκτή είναι ο μετατροπές από αναλογικό σε ψηφιακό σήμα. Μια αναλογική τάση εισόδου μετατρέπεται σε ψηφιακή, ώστε να μπορεί να επεξεργαστεί από έναν μικροελεγκτή ή από οποιοδήποτε άλλο ψηφιακό σύστημα.

Οι μετατροπές A / D μπορούν να ταξινομηθούν σε μονοπολικούς και σε διπολικούς όσον αφορά την τάση εισόδου. Ο μονοπολικός μετατροπές A / D δέχεται στην είσοδο μονοπολική τάση από 0V έως

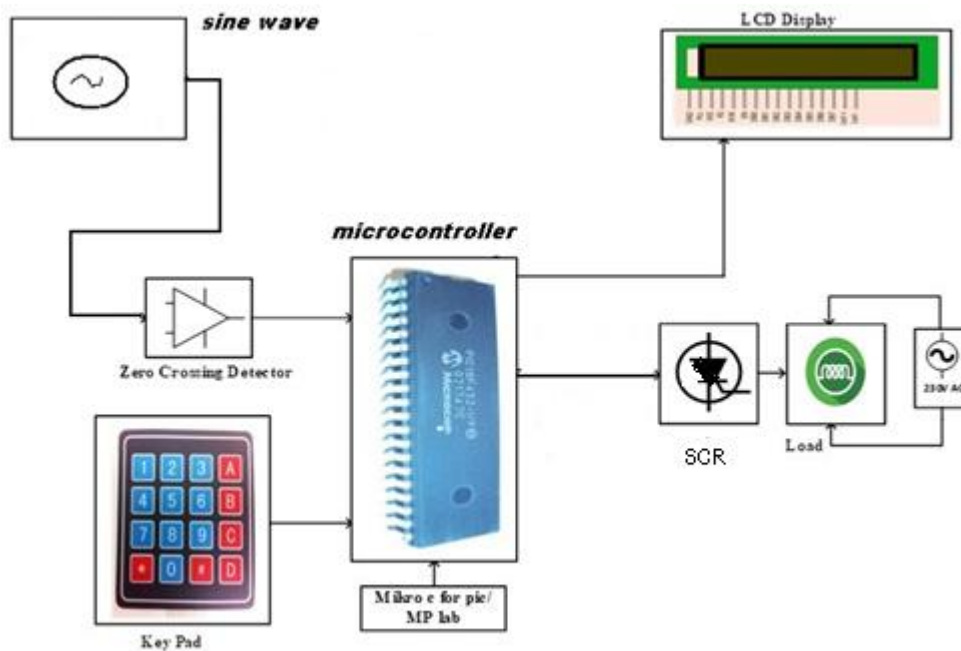
+V , ενώ ο διπολικός A / D μετατροπέας δέχεται διπολική τάση εισόδου από $-V$ έως $+V$. Οι διπολικοί μετατροπείς χρησιμοποιούνται συχνότερα σε εφαρμογές επεξεργασίας σήματος, όπου τα σήματα είναι διπολικά. Οι μονοπολικοί μετατροπείς είναι συνήθως φθηνότεροι, και χρησιμοποιούνται σε πολλές εφαρμογές ελέγχου και σε όργανα. Ο A / D μετατροπέας που περιέχεται στον PIC18F4550 είναι των 10 bits και έχει 13 κανάλια αναλογικής εισόδου.

Κεφάλαιο 3^ο

Σύστημα παραγωγής παλμών για έλεγχο ηλεκτρονικών ισχύος

3.1 Σύστημα παραγωγής παλμών με τη χρήση μικροελεγκτή

Ένα σύστημα παραγωγής παλμών με τη χρήση μικροελεγκτή είναι μια συσκευή που ρυθμίζει την ισχύ ενός εναλλασσόμενου σήματος μέσω ενός ηλεκτρολογίου, που είναι φιλικό στη χρήση για τον καθένα. Κατασκευάζεται με τη βοήθεια ηλεκτρονικών εξαρτημάτων όπως είναι ένα θυρίστορ, ένα κύκλωμα ανιχνευτή μηδενικής διέλευσης, τον μικροελεγκτή PIC18F4550 που ανήκει στην οικογένεια των PIC, μια οθόνη LCD, ένα ηλεκτρολόγιο και ένα φορτίο. Το φορτίο στην παρούσα εργασία επιλέχθηκε να είναι μια ηλεκτρική μηχανή (λειτουργία ως κινητήρας) και ένας λαμπτήρας. Η συσκευή αυτή, λοιπόν, είναι πιο αξιόπιστη, πιο αποτελεσματική και φθηνότερη από άλλες συσκευές ελέγχου ισχύος. Το μπλοκ διάγραμμα της παρούσας εργασίας με τον μικροελεγκτή και όλα τα υπόλοιπα απαραίτητα εξαρτήματα φαίνεται στο σχήμα 3.1.



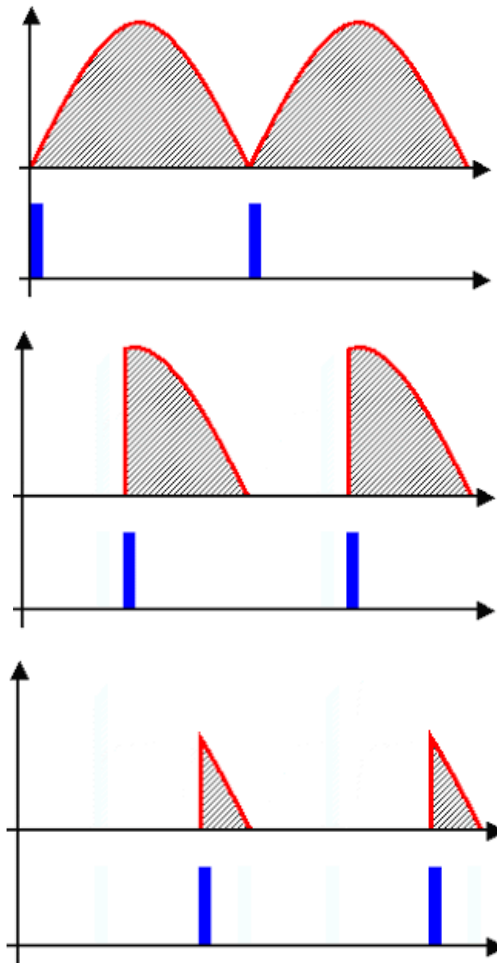
Σχήμα 3.1 Σχηματική αναπαράσταση του συστήματος

Η αρχή λειτουργίας αυτής της συσκευής είναι αρκετά απλή και στηρίζεται στην αρχή ότι ο χρήστης μέσω του ηλεκτρολογίου θα καθορίζει την γωνία έναυσης του παλμού που θα σταλθεί στο θυρίστορ. Η συγκεκριμένη συσκευή που αποτελείται από τον μικροελεγκτή, μια οθόνη LCD κλπ, τροφοδοτείται με 5V συνεχούς τάσης.

Σαν φορτίο συνδέουμε στην έξοδο μια ηλεκτρική μηχανή. Η ισχύς εξόδου θα είναι αυτή που θα καθορίζει το ποσό τάσης που θα «πέφτει» πάνω στον κινητήρα. Στην πραγματικότητα, αυτό γίνεται

όταν ο χρήστης θα εισάγει την γωνία έναυσης, με την οποία θα στέλνεται ο παλμός στη συσκευή ισχύος, ρυθμίζοντας έτσι το ποσοστό της ισχύος. Τότε, ο μικροελεγκτής, ο οποίος είναι προγραμματισμένος σε γλώσσα C, θα δίνει τον παλμό στο θυρίστορ σύμφωνα με αυτή την ισχύ. Όταν αυτή η γωνία θα αλλάζει, θα αλλάζει και η ισχύς εξόδου. Η γωνία αυτή θα απεικονίζεται στην οθόνη LCD, η οποία είναι επίσης συνδεδεμένη στον μικροελεγκτή.

Στο παρακάτω σχήμα 3.2, φαίνονται όσα αναφέρθηκαν παραπάνω για την λειτουργία της συσκευής. Η κόκκινη κυματομορφή αντιπροσωπεύει την τάση εξόδου, ενώ οι μπλε παλμοί είναι τα σήματα οδήγησης του θυρίστορ, όπως τα έχει ορίσει ο χρήστης σε τυχαίες γωνίες.



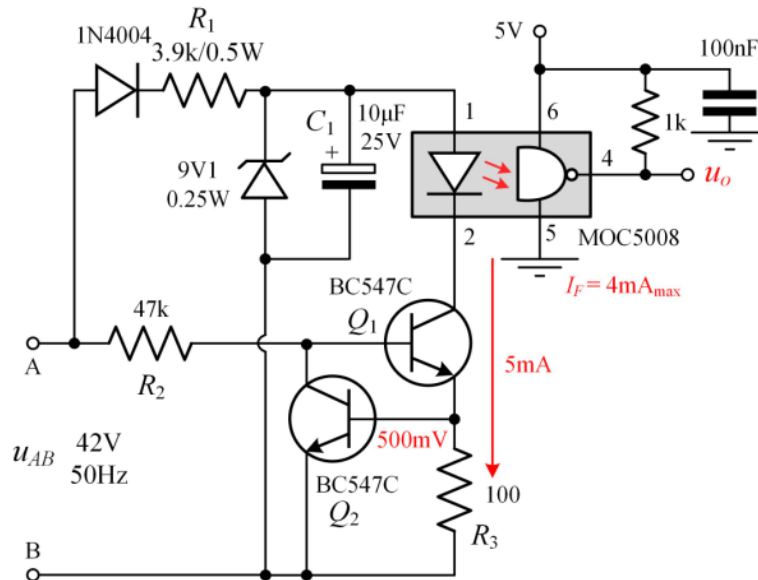
Σχήμα 3.2 Η τάση εξόδου ρυθμίζεται από το σήμα οδήγησης της πύλης στο θυρίστορ.

3.2 Ανιχνευτής μηδενός

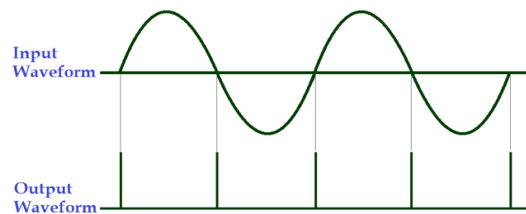
Προκειμένου να μπορέσουμε να ελέγξουμε την κυματομορφή εισόδου, χρειάζεται να χρησιμοποιήσουμε ένα κύκλωμα, το οποίο λειτουργεί σαν ανιχνευτής μηδενικής διέλευσης (σχήμα 3.3). Ο ανιχνευτής μηδενικής διέλευσης είναι ένας τύπος συγκριτή τάσης που χρησιμοποιείται για την ανίχνευση της μετάβασης από την θετική ημιπερίοδο μιας εναλλασσόμενης τάσης στην αρνητική ημιπερίοδο, κάτι που συμπίπτει με την μηδενική κατάσταση τάσης. Είναι χρήσιμο να γίνει μια

ανάλυση στο συγκεκριμένο κύκλωμα, καθώς είναι απαραίτητη διάταξη για την σωστή λειτουργία της κατασκευής μας.

Αρχικά, θα ασχοληθούμε με τις επιμέρους ηλεκτρονικές διατάξεις και εξαρτήματα από τα οποία αποτελείται.



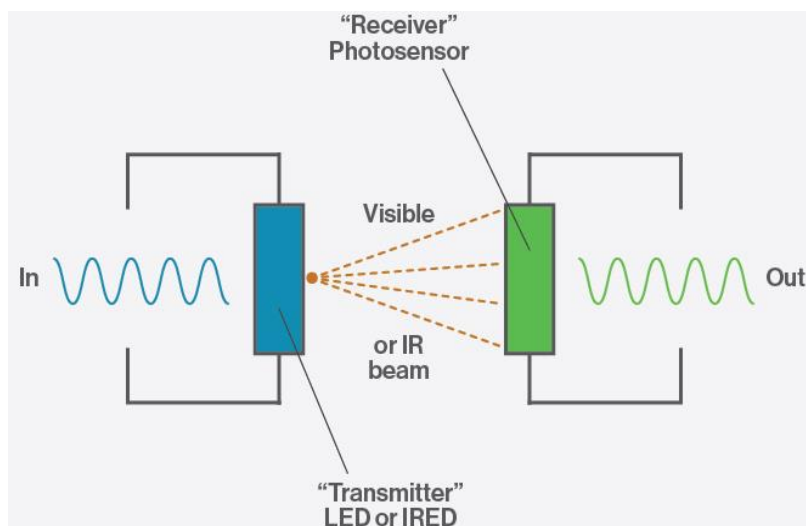
Σχήμα 3.3 Διάταξη zero crossing



Σχήμα 3.4 Κυματομορφή εισόδου και κυματομορφή εξόδου κυκλώματος zero crossing

❖ Optocoupler

Στις περισσότερες εφαρμογές απαιτείται γαλβανική απομόνωση. Τη λύση δίνει η χρήση optocoupler. Όσον αφορά τη λειτουργία του, το ρεύμα που εφαρμόζεται στον optocoupler, είναι αυτό που κάνει το υπέρυθρο LED να εκπέμπει ένα φως, το οποίο είναι ανάλογο του ρεύματος. Όταν το φως «πέφτει» στην φωτοευαίσθητη επιφάνεια, ενεργοποιείται και οδηγεί το ρεύμα όπως θα έκανε ένα τρανζίστορ.



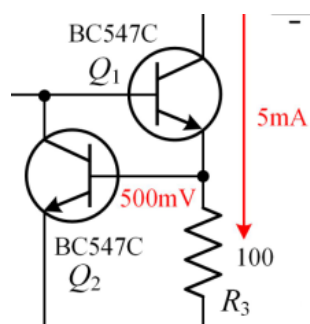
Σχήμα 3.5 Αναπαράσταση λειτουργίας optocoupler

Η φωτοευαίσθητη επιφάνεια είναι τυπικά ασύνδετη από επιλογή για να παρέχει την υψηλότερη ευαισθησία στο υπέρυθρο φως. Μπορεί, επίσης, να συνδεθεί στη γείωση με μία εξωτερική αντίσταση για υψηλότερο βαθμό ελέγχου της ευαισθησίας του διακόπτη.

Το Optocoupler απομονώνει αποτελεσματικά ένα κύκλωμα εισόδου και ένα κύκλωμα εξόδου. Αυτή η συσκευή βασικά λειτουργεί σαν διακόπτης, συνδέοντας δύο απομονωμένα κυκλώματα σε μια πλακέτα PCB. Όταν το ρεύμα σταματάει να «ρέει» στο LED, η φωτοευαίσθητη επιφάνεια σταματάει, επίσης, να άγει και σβήνει.

Το optocoupler μπορεί αποτελεσματικά να:

- ❖ Απομακρύνει τον ηλεκτρικό θόρυβο από τα σήματα.
 - ❖ Απομονώνει κυκλώματα χαμηλής τάσης από κυκλώματα υψηλής τάσης.
 - ❖ Επιτρέπει να χρησιμοποιούνται χαμηλά ψηφιακά σήματα για τον έλεγχο εναλλασσόμενων σημάτων.
- ❖ NPN transistors



Σχήμα 3.6 PNP transistors

Η συνδεσμολογία αυτή αποτελείται από δύο PNP transistors, όπου ο εκπομπός του ενός γίνεται βάση του άλλου. Λειτουργούν σε $V_{ce} < 10\text{ V}$ και $I_c \approx 1\text{ mA}$.

Τα transistors σχηματίζουν μια ροή σταθερού ρεύματος. Η αρνητική ανάδραση που υπάρχει αποδίδει σταθερή τάση στην αντίσταση R3 (αυτή είναι η τάση εκπομπού βάσης του Q2, περίπου 500mV στο σημείο λειτουργίας σχεδίασης) και κατά συνέπεια ένα σταθερό ρεύμα μέσω της αντίστασης R3. Εάν το τρέχον κέρδος και των δύο τρανζίστορ είναι αρκετά υψηλό, αυτό θα είναι περίπου ίσο με το ρεύμα συλλέκτη του Q1 και το ρεύμα LED.

Κατά τη θετική ημιπερίοδο, όταν το ρεύμα της αντίστασης R2 είναι αρκετό, η βάση του Q1 μπαίνει σε κατάσταση αγωγής, ενεργοποιώντας τον μικροελεγκτή. Κατά την αρνητική ημιπερίοδο, η τάση στη βάση του Q1 αναγκάζεται να φτάσει τα -0,5 V περίπου, καθώς η βάση του Q1 και ο συλλέκτης του Q2 είναι βραχυκυκλωμένοι.

❖ Υπόλοιπα ηλεκτρονικά εξαρτήματα

Τα D1, R1, C1 και ZD1 σχηματίζουν ένα απλό τροφοδοτικό. Η φόρτιση του C1 επιτρέπει στο ρεύμα για τη λειτουργία του LED που δημιουργείται κατά τη στιγμή της μηδενικής διέλευσης να ρέει. Πιο συγκεκριμένα, η δίοδος προστασίας αντίστροφης πολικότητας D1 πρέπει να αντέχει στην μέγιστη αντίστροφη τάση που αναμένεται στο δίκτυο. Εκτός από το ρεύμα διαρροής (το οποίο πρέπει να είναι κάτω από 100μΑ στο πλήρες εύρος θερμοκρασίας λειτουργίας) δεν υπάρχουν ειδικές απαιτήσεις για αυτήν τη δίοδο.

Να σημειωθεί πως ο κυματισμός τάσης είναι κάτω από 500mV, κάτι που δεν αποτελεί πρόβλημα για την απόδοση του μικροελεγκτή. Η τρέχουσα κυμάτωση μέσω C1 είναι μικρή (κάτω από 1 mA), οπότε η επιλογή εξαρτήματος είναι απλή. Η ονομαστική τάση πρέπει να υπερβαίνει την τάση zener 8,2V. Για την επιλογή της αντίστασης R1 πρέπει να σκεφτούμε το εξής: εάν το R1 είναι πολύ χαμηλό, η υπερβολική ισχύς διασκορπίζεται στη δίοδο zener, ενώ εάν το R1 είναι πολύ υψηλό, το ρεύμα LED θα περιορίζεται από το R1 και όχι από την τρέχουσα πηγή.

Όσον αφορά την αντίσταση R2 η τιμή της μπορεί να μειωθεί 2 έως 3 φορές εάν επιθυμείται περαιτέρω βελτίωση της καθυστέρησης που οδηγεί σε έλεγχο της ισχύος. Ωστόσο, μια συνεχής καθυστέρηση είναι πιο σημαντική από μια μικρή καθυστέρηση.

Ανακεφαλαιώνοντας, ένα κύκλωμα ανιχνευτή μηδενός ή αλλιώς zero crossing detect, χρησιμοποιείται σε μια κυματομορφή για την ανίχνευση της μετάβασης από την θετική στην αρνητική ημιπερίοδο, κάτι που συμπίπτει με την κατάσταση μηδενικής τάσης. Άλλες εφαρμογές των ανιχνευτών μηδενικής διέλευσης είναι ο έλεγχος της κατάστασης ενός λειτουργικού ενισχυτή, η χρήση ως μετρητές συχνότητας και η μεταγωγή σε κυκλώματα ηλεκτρονικών ισχύος.

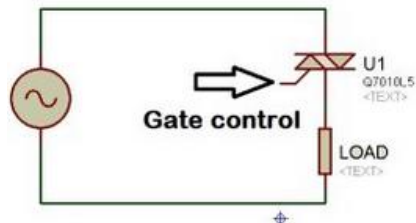
3.3 Έλεγχος ισχύος με θυρίστορ

Όπως αναφέρθηκε, ο έλεγχος AC ισχύος χρησιμοποιώντας μικροελεγκτή είναι μια διάταξη σχεδιασμένη για τον έλεγχο της ροής ισχύος κατά μήκος του φορτίου. Για τον έλεγχο της ροής ισχύος με θυρίστορ χρησιμοποιείται η μέθοδος φασικού ελέγχου, ενώ το θυρίστορ χρησιμοποιείται σαν διακόπτης για να ελέγχει την ροή της ενέργειας. Αρκετές διατάξεις ηλεκτρονικών ισχύος στηρίζονται σε αυτόν τον έλεγχο ισχύος.

Ο έλεγχος γωνίας φάσης είναι μια μέθοδος PWM (Pulse Width Modulation) που εφαρμόζεται σε κυκλώματα με εναλλασσόμενες τάσεις εισόδου. Φυσικά, η εναλλασσόμενη τάση μπορεί να προέρχεται από έναν μετασχηματιστή, αλλά συνήθως είναι αυτή η κύρια πηγή -κάτι που δείχνει την χρησιμότητα της διάταξης που μελετάμε. Η χρησιμότητα της έγκειται στον έλεγχο ή περιορισμό της ισχύος του φορτίου.

Πιο συγκεκριμένα, λοιπόν, η συσκευή ισχύος που χρησιμοποιείται είναι το θυρίστορ- συνήθως triacs ή SCRs. (Υπάρχουν μέθοδοι ελέγχου φάσης για υψηλές συχνότητες, οι οποίες χρησιμοποιούν MOSFET ή IGBT, όμως στην παρούσα εργασία έγινε χρήση των θυρίστορ). Η ισχύς που ρέει στο φορτίο ελέγχεται με την καθυστέρηση της γωνίας πυροδότησης (πυροδότηση σε κάθε ημιπερίοδο) της συσκευής ισχύος.

Αρχικά, ας κατανοήσουμε τα βασικά για τον έλεγχο της ισχύος με θυρίστορ. Για παράδειγμα, έστω ότι ένα φορτίο συνδέεται σε μία τροφοδοσία εναλλασσόμενου ρεύματος, μέσω ενός διακόπτη όπως φαίνεται στο σχήμα παρακάτω. Το triac χρησιμοποιείται σαν διακόπτης και είναι ένα είδος θυρίστορ, (ένα triac αποτελείται από δύο θυρίστορ back to back).



Σχήμα 3.7 Απλή διάταξη ισχύος με triac

Η ροή ισχύος στο φορτίο μπορεί να ρυθμιστεί, ελέγχοντας την rms τιμή τάσης που εμφανίζεται κατά μήκος του φορτίου. Αυτή η μέθοδος καλείται, επίσης, και μέθοδος ελέγχου AC τάσης. Σε έναν έλεγχο AC τάσης, η rms τιμή τάσης κατά μήκος του φορτίου, ρυθμίζεται μεταβάλλοντας την γωνία πυροδότησης του σήματος της πύλης του triac. Όπως είναι γνωστό, η ισχύς σε κυκλώματα ac ισχύος, υπολογίζεται ως εξής:

$$P = V \times I \times \cos(\theta), \quad (3.1)$$

όπου $\cos(\theta)$ είναι ο συντελεστής ισχύος. Για να γίνει πιο εύκολη η κατανόηση του κυκλώματος χρησιμοποιήσαμε μόνο ωμικό φορτίο. Ωστόσο, μπορεί να χρησιμοποιηθεί και σε επαγωγικά φορτία. Στα ωμικά φορτία ο συντελεστής ισχύος πάντα είναι ίσος με τη μονάδα. Οπότε, ο παραπάνω τύπος γίνεται:

$$P = V \cdot I = V \cdot \frac{V}{R} = \frac{V^2}{R} \Rightarrow P = \frac{V^2}{R} \quad (3.2)$$

Συνεπώς, σύμφωνα με τον παραπάνω τύπο, η ισχύς στο φορτίο είναι ανάλογη του τετραγώνου της τάσης. Έτσι, μπορεί να ρυθμιστεί η ισχύς σε κάποια ηλεκτρονική συσκευή, ρυθμίζοντας την τάση στο φορτίο. Οι διατάξεις αυτές χρησιμοποιούνται συχνά στην βιομηχανία σαν μεταγωγείς σε μετασχηματιστές, σε επαγωγική θέρμανση, στον έλεγχο ταχύτητας κινητήρα και έλεγχο φωτισμού.

Όσον αφορά το κύκλωμά μας, το θυρίστορ που χρησιμοποιήσαμε είναι το SCR. Όπως είναι γνωστό το θυρίστορ είναι μια συσκευή μανδάλωσης – ενεργοποιείται από ένα σήμα στην πύλη του SCR και όταν το ρεύμα γίνει μεγαλύτερο από το ρεύμα συγκράτησης (το ελάχιστο ρεύμα ανόδου ώστε το SCR να είναι σε κατάσταση ON και να μην απενεργοποιείται) και το ρεύμα μανδάλωσης (το ελάχιστο ρεύμα ανόδου που απαιτείται για να διατηρηθεί το SCR σε κατάσταση ON ακόμη και μετά την αφαίρεση του παλμού στην πύλη του), το θυρίστορ παραμένει ενεργοποιημένο, μέχρι το ρεύμα που το διαρρέει να γίνει ξαφνικά χαμηλό (πολύ κοντά στο μηδέν), όπως συμβαίνει κατά την διέλευση του εναλλασσόμενου σήματος από το μηδέν.

Οπότε, κατά τον έλεγχο γωνίας φάσης, ένας παλμός στέλνεται στην πύλη του SCR. Αυτός ο παλμός στέλνεται μεταξύ της μίας ανίχνευσης μηδενικής διέλευσης του σήματος και της άλλης. Χωρίς τον παλμό στην πύλη, ακριβώς μετά την ανίχνευση της μηδενικής διέλευσης το SCR είναι απενεργοποιημένο και δεν διαρρέεται από ρεύμα. Μετά από ένα συγκεκριμένο διάστημα, το σήμα δίνεται στην πύλη και το SCR ενεργοποιείται, παραμένοντας έτσι μέχρι το ρεύμα να γίνει μηδέν. Αυτό συμβαίνει στην επόμενη ανίχνευση μηδενικής διέλευσης. Για λόγους απλότητας, υποθέτουμε ότι το ρεύμα στο SCR όσο είναι ενεργοποιημένο είναι μεγαλύτερο από το ρεύμα μανδάλωσης και το ρεύμα συγκράτησης.

Συνεπώς, η τάση που υπάρχει στο φορτίο εξαρτάται από το χρονικό διάστημα που το SCR παραμένει ενεργοποιημένο. Το χρονικό διάστημα που συμβαίνει αυτό καθορίζεται από την χρονική καθυστέρηση μεταξύ την ανίχνευση της μηδενικής διέλευσης και της εφαρμογής σήματος στην πύλη του SCR.

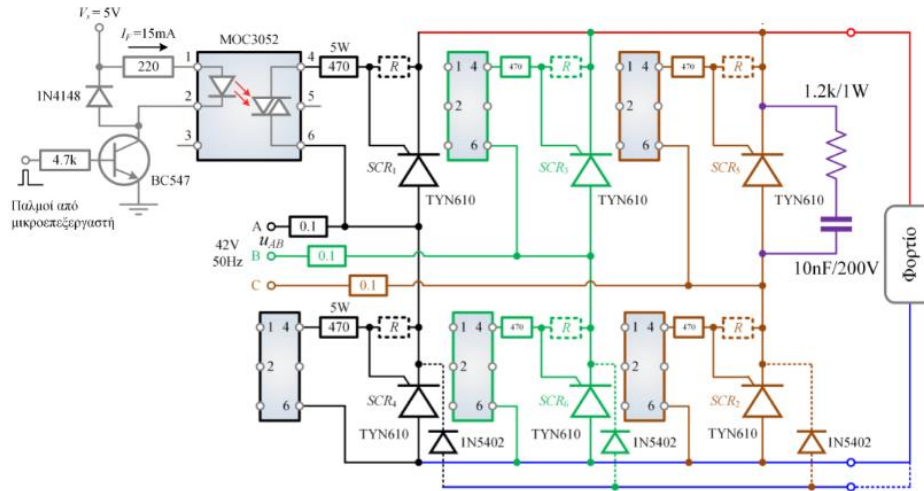
Ανακεφαλαιώνοντας, μπορούμε να ρυθμίσουμε την τάση και κατά συνέπεια την ισχύ που θα φτάσει στο φορτίο καθυστερώντας το σήμα στην πύλη του SCR, με τη βοήθεια του μικροελεγκτή.

Τριφασική πλήρως ελεγχόμενη γέφυρα ανόρθωσης

Η τριφασική ανόρθωση είναι η διαδικασία μετατροπής ενός ισορροπημένου τριφασικού τροφοδοτικού σε σταθερή παροχή DC τάσης, χρησιμοποιώντας διόδους στερεάς κατάστασης ή θυρίστορ, τα οποία και χρησιμοποιήσαμε. Ένας τριφασικός μετατροπέας γέφυρας χρησιμοποιείται για την τριφασική ανόρθωση της τάσης δικτύου. Αποτελείται από δύο τριφασικούς μετατροπείς απλής ανόρθωσης συνδεδεμένους στη σειρά. Οι $SCR_{1,3,5}$ απαρτίζουν το θετικό μετατροπέα απλής ανόρθωσης, ενώ οι $SCR_{2,4,6}$ απαρτίζουν τον αρνητικό μετατροπέα απλής ανόρθωσης.

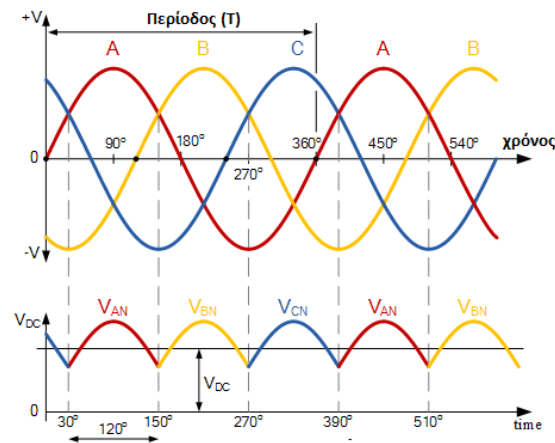
Το πλεονέκτημα των τριφασικών κυκλωμάτων είναι ότι μπορούν να χρησιμοποιηθούν για την τροφοδοσία πολλών βιομηχανικών εφαρμογών, όπως τον έλεγχο κινητήρα, με τον οποίο ασχοληθήκαμε ή φόρτιση μπαταρίας, οι οποίες απαιτούν υψηλότερες απαιτήσεις ισχύος από ό, τι είναι σε θέση να παρέχει ένα μονοφασικό κύκλωμα ανόρθωσης.

Ακολουθεί το σχεδιαστικό σχήμα 3.8 της γέφυρας πριν την τύπωσή της και του κυκλώματος οδήγησης, που θα αναλυθεί παρακάτω για ευκολότερη κατανόηση.



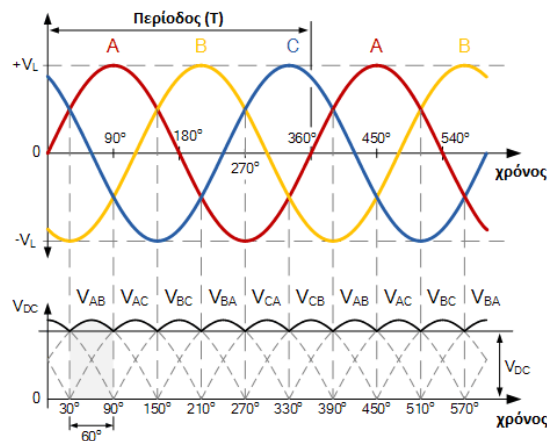
Σχήμα 3.8 Σχηματικό τριφασικής γέφυρας

Συνδυάζοντας τρεις τάσεις AC ίδιας συχνότητας και πλάτους, κάθε τάση AC ονομάζεται «φάση». Αυτές οι τρεις φάσεις είναι 120° εκτός φάσης μεταξύ τους, παράγοντας μια ακολουθία φάσης ή περιστροφή φάσης: $360^\circ \div 3 = 120^\circ$, όπως φαίνεται στο παρακάτω σχήμα 3.9. Κάθε φάση VA-VB-VC ξεκινά από τις 0° .



Σχήμα 3.9 Παραγόμενη κυματομορφή εξόδου με διαφορά φάσης

Σε ανορθωτές ισχύος τριών φάσεων, η αγωγή συμβαίνει πάντα στην πιο θετική δίοδο και στην αντίστοιχη πιο αρνητική δίοδο. Έτσι, καθώς οι τρεις φάσεις περιστρέφονται κατά μήκος των άκρων του ανορθωτή, η αγωγή περνά από θυρίστορ σε θυρίστορ. Στη συνέχεια, κάθε θυρίστορ άγει για 120° (το ένα τρίτο) σε κάθε κύκλο τροφοδοσίας, αλλά καθώς χρειάζονται δύο θυρίστορ να άγουν σε ζεύγη, κάθε θυρίστορ του ζεύγους θα άγει μόνο 60° (ένα έκτο) ενός κύκλου οποιαδήποτε στιγμή, όπως φαίνεται η ανορθωμένη εξόδός τους στο παρακάτω σχήμα 3.10.



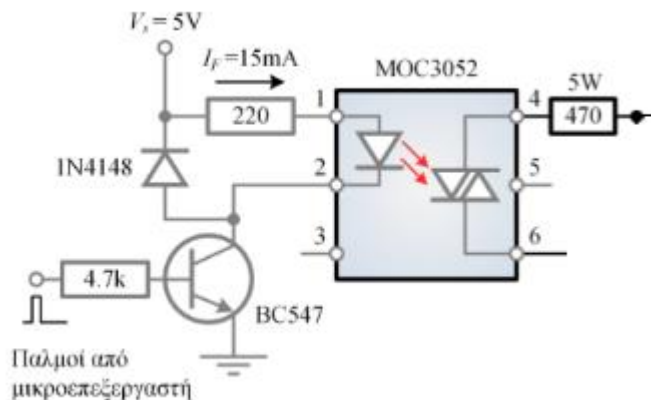
Σχήμα 3.10 Ανορθωμένη παραγόμενη κυματομορφή εξόδου

Γενικότερα, το thyristor είναι ένας ελεγχόμενος ανορθωτής πυριτίου τεσσάρων στρωμάτων (p-n-p-n), το οποίο μπορεί να θεωρηθεί σαν συνδυασμός δύο τρανζίστορ. Όταν στην πύλη του thyristor δεν υπάρχει παλμός έναυσης, τότε δεν υπάρχει ροή ρεύματος μέσω αυτού. Για να μεταβεί το thyristor σε κατάσταση αγωγής, πρέπει στην πύλη του να δοθεί παλμός έναυσης, με την προϋπόθεση ότι το thyristor είναι θετικά πολωμένο.

Έτσι, με ελεγχόμενη ανόρθωση τριών φάσεων χρησιμοποιώντας thyristor αντί για διόδους, μπορούμε να ελέγξουμε την τιμή της μέσης τάσης εξόδου DC ελέγχοντας τη γωνία έναυσης α των ζευγών thyristor. Η γωνία έναυσης λαμβάνει τιμές από 0° έως 100° .

Ένα άλλο πλεονέκτημα ενός ανορθωτή γέφυρας πλήρους κύματος είναι ότι το ρεύμα φορτίου είναι καλά ισορροπημένο κατά μήκος της γέφυρας, βελτιώνοντας την απόδοση (ο λόγος ισχύος εξόδου DC προς την παρεχόμενη ισχύ εισόδου) και μειώνοντας το περιεχόμενο κυματισμού, τόσο σε πλάτος όσο και σε συχνότητα.

Κύκλωμα οδήγησης

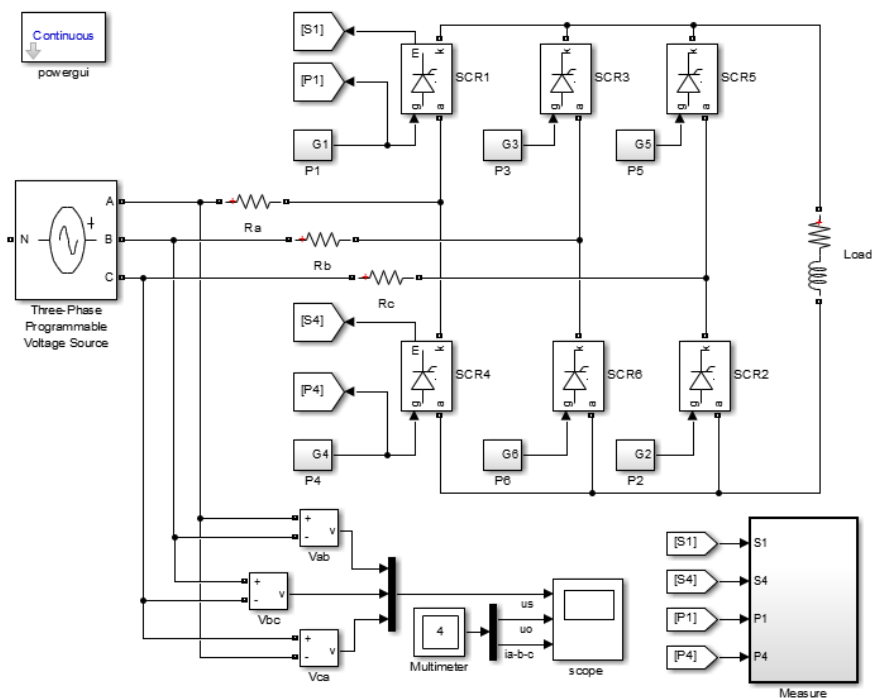


Σχήμα 3.11 Σχηματικό κυκλώματος οδήγησης

Η χρησιμότητα του παραπάνω κυκλώματος εξυπηρετεί στην οδήγηση της τριφασικής πλήρως ελεγχόμενης γέφυρας ανόρθωσης που οδηγεί το φορτίο του συστήματός μας. Με την είσοδο των παλμών από τον μικροελεγκτή στη βάση του τρανζίστορ, το τρανζίστορ-διακόπτης μεταβαίνει σε κατάσταση αγωγής και διαρρέεται από ρεύμα. Στη συνέχεια, το ρεύμα από την τροφοδοσία V_s προχωρά στην αντίσταση των 220Ω για να «ρίξει» την τάση και να μην καεί το led μέσα στο MOC, το οποίο προσφέρει γαλβανική απομόνωση. Η διάδος είναι για προστασία του τρανζίστορ και της εισαγωγής των παλμών, ώστε να μην διαρρεύσει ανάστροφο ρεύμα.

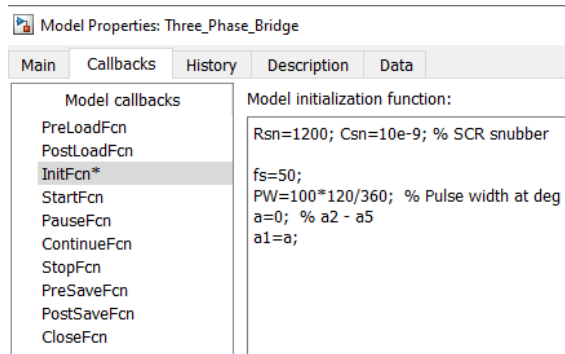
3.4 Προσομοίωση ολόκληρης της κατασκευής στο Matlab

Για να είμαστε σίγουροι ότι το μέρος των ισχύος λειτουργεί κανονικά και ανταποκρίνεται στις απαιτήσεις μας, πραγματοποιήσαμε προσομοίωσή στο πρόγραμμα Matlab. Στην προσομοίωση συμπεριλαμβάνονται όλα τα κυκλώματα που αναφέρθηκαν στο τρέχον κεφάλαιο, καθώς και ο προγραμματισμός του μικροελεγκτή, χωρίς τη χρήση της οθόνης και του ηλεκτρολογίου. Εξάριση αποτελεί επίσης η έξοδος, στην οποία τοποθετήσαμε μια αντίσταση και ένα πηνίο ως φορτίο. Στο παρακάτω σχήμα 3.12 φαίνεται το κύκλωμα της προσομοίωσης.



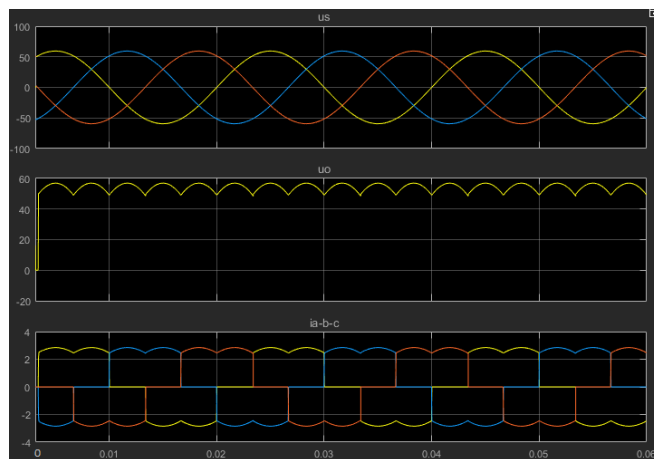
Σχήμα 3.12 Προσομοίωση στο Simulink

Αφού ορίσαμε τις κατάλληλες τιμές στα υλικά και τοποθετήσαμε τα κατάλληλα όργανα μέτρησης, ήμασταν πλέον σε θέση να εισάγουμε τις μοίρες για να λάβουμε τα αποτελέσματα εξόδου. Η εισαγωγή και αλλαγή των μοιρών πραγματοποιήθηκε σε παράθυρο του Matlab και φαίνεται στο παρακάτω σχήμα 3.13. Η αρχική γωνία έναυσης ήταν το $\alpha=0^0$.



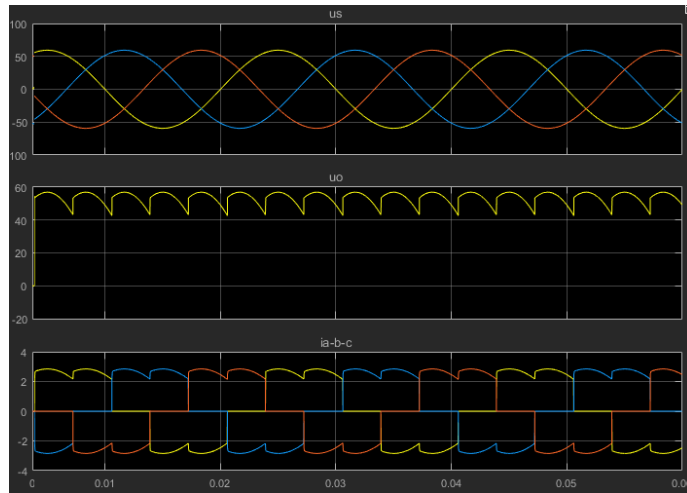
Σχήμα 3.13 Εισαγωγή μοιρών

Αφού ολοκληρώσαμε όλες τις απαραίτητες ρυθμίσεις, λάβαμε τα παρακάτω αποτελέσματα της τάσης και του ρεύματος εξόδου σε συσχέτισμό με την κυματομορφή εισόδου.

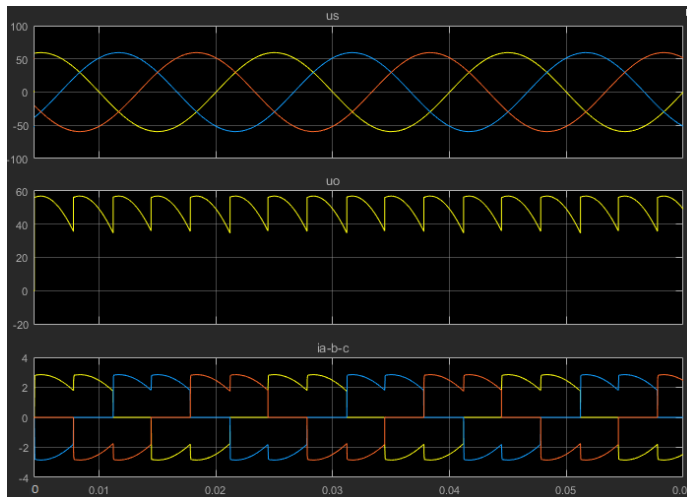


Σχήμα 3.14 Εισαγωγή 0°

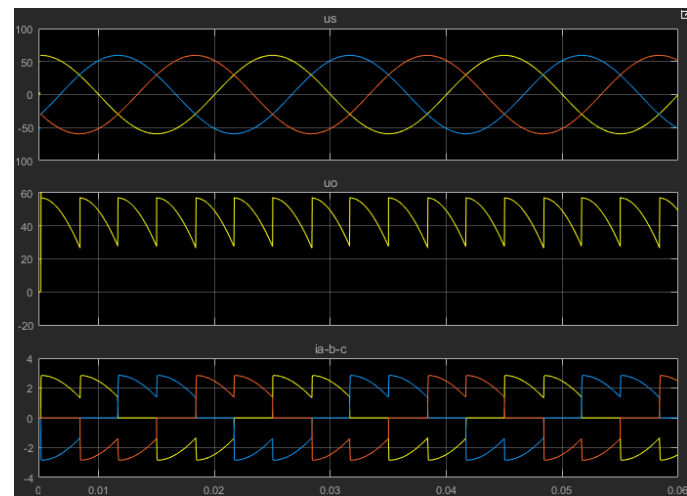
Παρατηρήσαμε πως οι παραγόμενες κυματομορφές εξόδου ήταν τα αναμενόμενα αποτελέσματα, δηλαδή μια πλήρως ανορθωμένη τάση εξόδου σε σχέση με την τριφασική μας τάση εισόδου κι έτσι συνεχίσαμε να αυξάνουμε σιγά σιγά τη γωνία έναυσης. Συνεχίσαμε με εισαγωγή 10°, 20° και 30°. Τα αποτελέσματά τους φαίνονται στα παρακάτω σχήματα.



Σχήμα 3.15 Εισαγωγή 10°

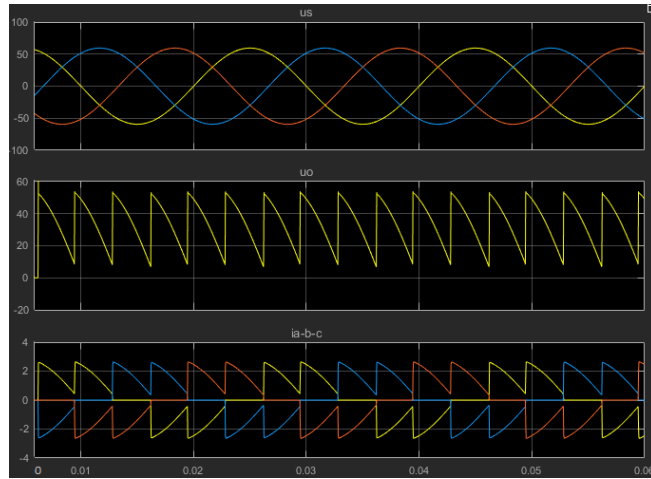


Σχήμα 3.16 Εισαγωγή 20°

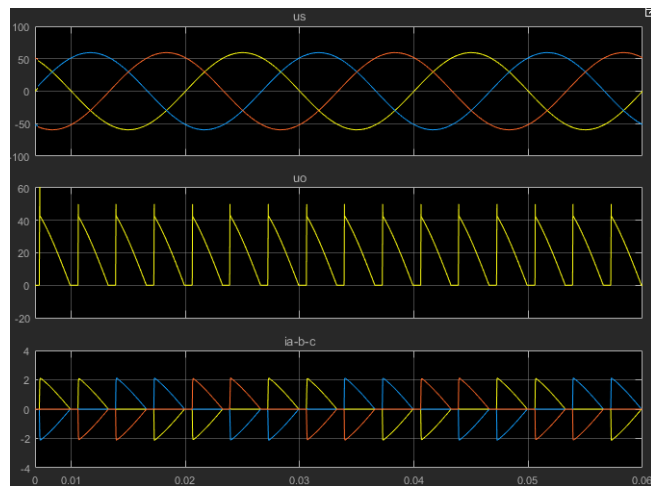


Σχήμα 3.17 Εισαγωγή 30°

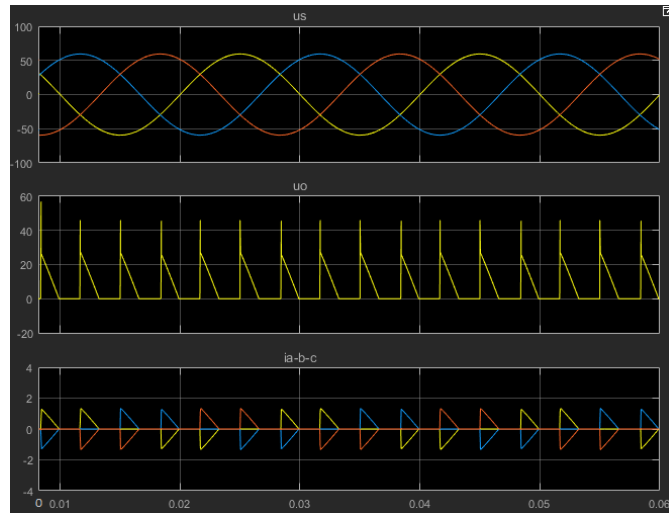
Αφού οι κυματομορφές εξόδου ανταποκρινόταν στην αύξηση των μοιρών και ήταν σύμφωνες με τις αναμενόμενες κυματομορφές εξόδου, συνεχίσαμε με εισαγωγή των 50° , 70° και 90° , με σκοπό να έχουμε περισσότερα αποτελέσματα εξόδου για εξακρίβωση. Τα αποτελέσματα της εισαγωγής φαίνονται στα ακόλουθα σχήματα.



Σχήμα 3.18 Εισαγωγή 50°



Σχήμα 3.19 Εισαγωγή 70°



Σχήμα 3.20 Εισαγωγή 90°

Πραγματοποιώντας διάφορες εισαγωγές γωνιών έναυσης και επαληθεύοντας τις κυματομορφές των τάσεων εξόδου με τις επιθυμητές, ήμασταν πλέον βέβαιοι πως το μέρος των ισχύος ανταποκρίνεται και λειτουργεί κανονικά. Έτσι, μπορούσαμε να συνεχίσουμε την υλοποίηση της κατασκευής.

Το κύκλωμα τροφοδοτείται μέσω του USB. Οι πυκνωτές των 100nF, χρησιμοποιούνται ως πυκνωτές απόξευξης (by pass capacitors) και συνδέονται πάντα μεταξύ της γείωσης και ενός κόμβου μιας ηλεκτρονικής διάταξης (όσο πιο κοντά γίνεται), με σκοπό να μην αφήσουν να αναπτυχθούν πάνω στον κόμβο σύνδεσης του εναλλασσόμενης (AC) συνιστώσες. Το πηνίο λειτουργεί ως φίλτρο καταστολής ρεύματος. Η χρήση του πηνίου ως ηλεκτρονικού φίλτρου οφείλεται στο γεγονός ότι είναι εντελώς αγώγιμο για το συνεχές ρεύμα, ενώ παρουσιάζει σύνθετη αντίσταση στο εναλλασσόμενο που αυξάνει όσο αυξάνεται η συχνότητα του εναλλασσόμενου ρεύματος. Επομένως, το πηνίο επιτρέπει την διέλευση των συνεχών ρευμάτων και των ρευμάτων χαμηλών συχνοτήτων, αλλά αποκόπτει τα ρεύματα πολύ υψηλών συχνοτήτων. Τα υπόλοιπα υλικά χρησιμοποιήθηκαν με παρόμοια λογική και σύμφωνα με τις πληροφορίες στα datasheets και reference manual.

Η επικοινωνία με τον υπολογιστή γίνεται μέσω της USB χρησιμοποιώντας τους ακροδέκτες 2 και 3 (D-, D+), οι οποίοι καταλήγουν στους ακροδέκτες 13 και 14 του μικροελεγκτή.

Σύμφωνα με πληροφορίες στο Datasheet, όταν χρησιμοποιείται η διασύνδεση USB, η συχνότητα του ρολογιού θα πρέπει να είναι είτε 6 MHz είτε 48 MHz, ανάλογα με το σε ποια λειτουργία χρησιμοποιείται (Low-Speed Mode ή High-Speed Mode). Η πλακέτα λειτουργεί σε High-Speed Mode.

Η οθόνη LCD 20×4 (20 χαρακτήρες, 4 γραμμές) έχει ενσωματωμένο τον μικροελεγκτή HD44780 της Hitachi, είναι αυτός που χαρακτηρίζει τον τρόπο επικοινωνίας με τον μικροελεγκτή PIC18F4550 της πλακέτας και για τον σκοπό αυτό χρησιμοποιεί μια σειρά από εντολές.

4.2 Κατασκευή σχηματικού και τυπωμένου της πλακέτας

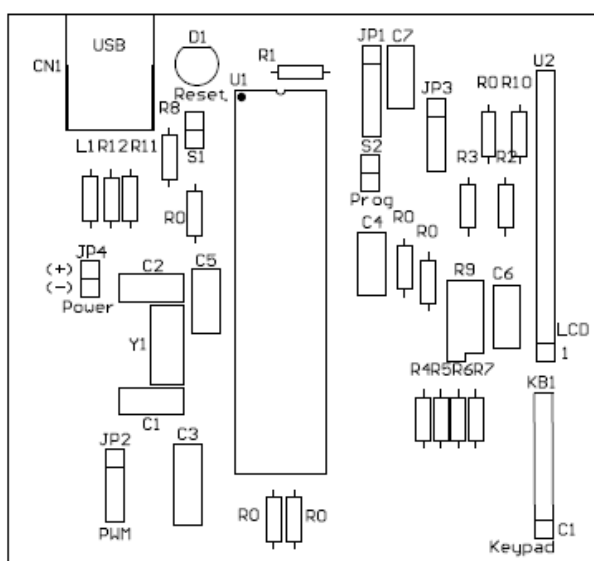
Για τη σχεδίαση του θεωρητικού κυκλώματος-σχηματικού χρησιμοποιήθηκε το πρόγραμμα Altium Designer. Πριν την τοποθέτηση των εξαρτημάτων ορίσαμε τις διαστάσεις του σχηματικού, οι οποίες είναι 8,2cm πλάτος και 7,7cm μήκος. Τα υλικά που επιλέχθηκαν και χρησιμοποιήθηκαν στη συνέχεια αποτυπώνονται στον παρακάτω πίνακα 1, στον οποίο παρουσιάζεται η τιμή, η κατηγορία, ο συμβολισμός, το αποτύπωμα, η βιβλιοθήκη και η ποσότητα του κάθε υλικού.

ΤΙΜΕΣ	ΚΑΤΗΓΟΡΙΑ	ΣΥΜΒΟΛΙΣΜΟΣ	ΑΠΟΤΥΠΩΜΑ	ΒΙΒΛΙΟΘΗΚΗ	ΠΟΣΟΤΗΤΑ
15pF	Capacitor	C1, C2	RAD-0.2	CAPACITOR	2
470nF	Capacitor	C3	RAD-0.3	CAPACITOR POL	1
100nF	Capacitor	C4, C5, C6, C7	RAD-0.2	CAPACITOR	4
CN-USB	USB type B	CN1	USB_S	CN-USB	1
power	Header 2 pins	D1	LED_3_3MM	LED	1
prog	Header 5 pins	JP1	EM_SIP5	HEADER 5	1
4 HEADER	4 Pin Header	JP2, JP3	EM_SIP4	4 HEADER	2
HEADER 2	Header 2 pins	JP4	EM_SIP2	HEADER 2	1
KEYBORD4X4	Header 8 pins	KB1	EM_SIP8	KEYBORD4X4	1
Inductor	Inductor 3.3 uH	L1	AXIAL0.4	Inductor	1
330Ω	Resistor 1/4W	R1	AXIAL0.4	RES1	1
10K	Resistor 1/4W	R2, R8	AXIAL0.4	RES1	2
220Ω	Resistor 1/4W	R3	AXIAL0.4	RES1	1
4.7K	Resistor 1/4W	R4, R5, R6, R7	AXIAL0.4	RES1	4
CONTRAST 1K	Potentiometer 1 K	R9	VRS	POT1	1
22Ω	Resistor 1/4W	R10, R11, R12	AXIAL0.4	RES1	3
RESET	Tag Switch	S1	EM_SIP2	SW-PB	1
Program	Tag Switch	S2	EM_SIP2	SW-PB	1
PIC18F4550-I/P	High Performance, Enhanced Flash, USB Microcontroller with nanoWatt Technology, 32K Flash, 40-Pin PDIP, Industrial Temperature Range	U1	PDIP600-P40	PIC18F4550-I/P	1
LCD2X16	2x16 LCD Display	U2	EM_SIP16	LCD2X16	1
20MHz	Crystal	Y1	crystal_th	CRYSTAL	1
0Ω	Resistor 1/4W	R0	AXIAL0.4	RES1	6

Πίνακας 1 Υλικά πλακέτας μικροελεγκτή

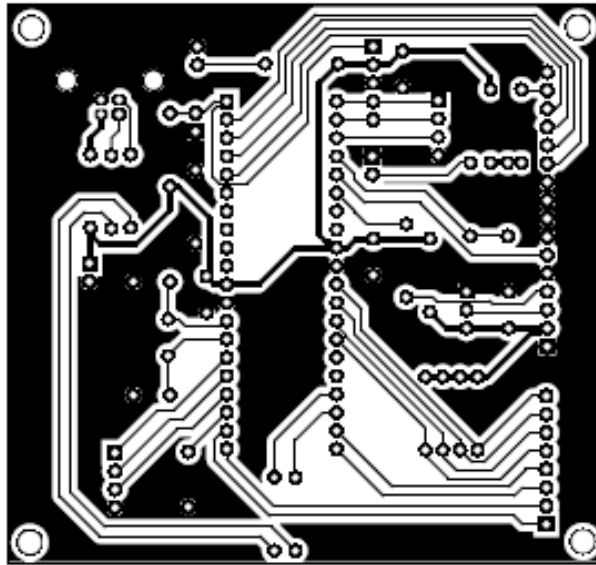
Έπειτα, έγινε η χωροταξική οργάνωση των υλικών όπως φαίνεται και στο σχήμα 4.2, καθώς και η σύνδεσή τους με την εντολή Interactively Route Connections. Στα σημεία που δεν ήταν εφικτή η σύνδεση των υλικών λόγω «σύγκρουσης» των αγωγών, τοποθετήθηκαν μηδενικές αντιστάσεις για αμεσότερη σύνδεση.

Στο σχήμα 4.2 απεικονίζεται το σχηματικό σε μορφή pdf με το συμβολισμό κάθε υλικού, το οποίο αποτέλεσε τον οδηγό μας για το μοντάρισμα της πλακέτας.



Σχήμα 4.2 Σχηματικό σε μορφή pdf

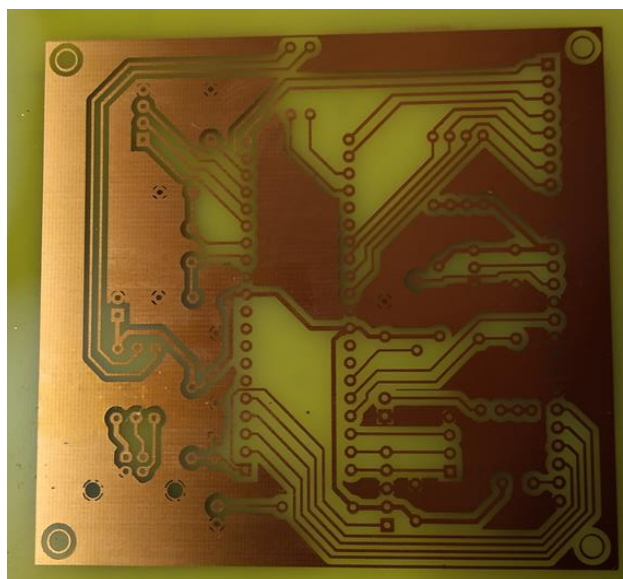
Τέλος, για να εξάγουμε το ασπρόμαυρο σχέδιο του κυκλώματος μας, με το οποίο τυπώσαμε την πλακέτα επιλέξαμε από το μενού File>Smart PDF, στα settings κάναμε δεξί κλικ και διαγραφή σε όλα τα layers εκτός από TopLayer. Επιλέγουμε τα holes>Next> Επιλέγουμε Monochrome, Quality 600dpi. Το σχήμα 4.3 τυπώθηκε πάνω σε διαφάνεια με Laser εκτυπωτή.



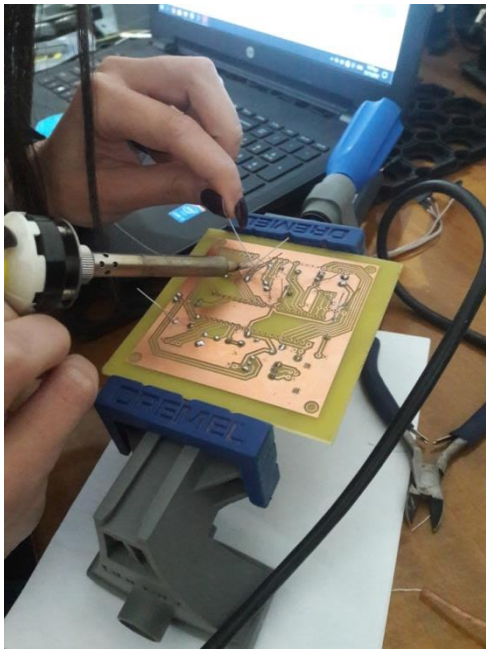
Σχήμα 4.3 Σχέδιο για τύπωση της πλακέτας

4.3 Κατασκευή πλακέτας

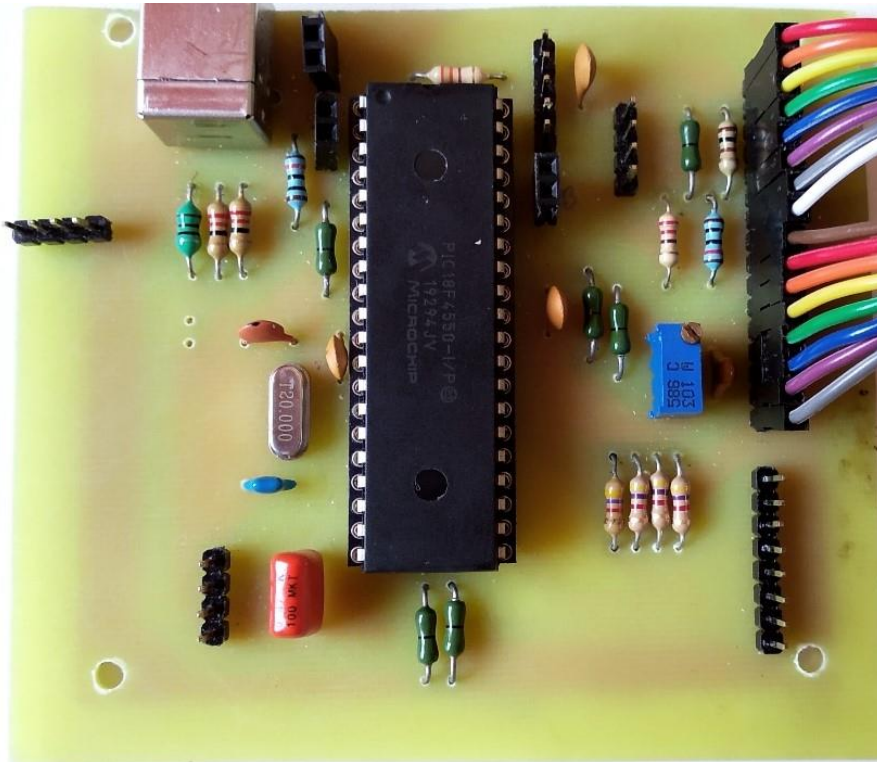
Παρακάτω υπάρχουν φωτογραφίες από την συναρμολόγηση της πλακέτας, καθώς και η τελική της μορφή.



Σχήμα 4.4 Τυπωμένη πλακέτα



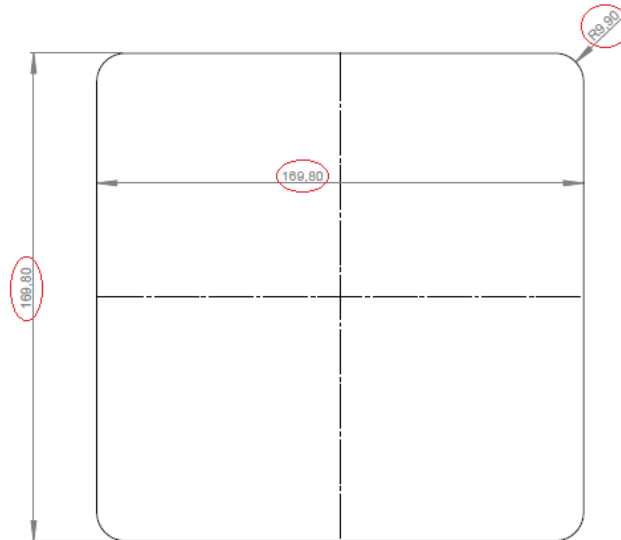
Σχήμα 4.5 Κολλήσεις και άνοιγμα τρυπών για τα υλικά



Σχήμα 4.6 Τελική όψη πλακέτας

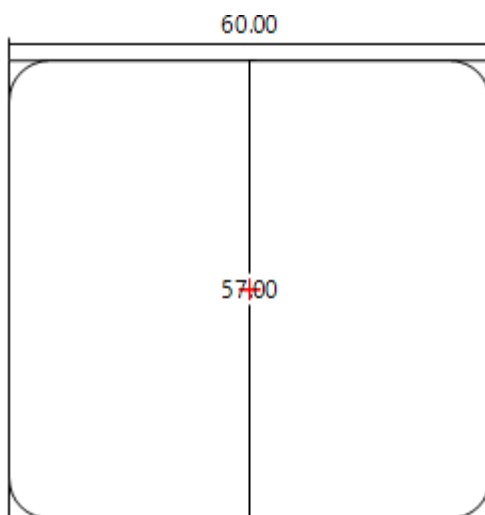
4.4 Αυτοκόλλητο κατασκευής

Για τη δημιουργία του αυτοκόλλητου του κουτιού της κατασκευής μας χρησιμοποιήσαμε το πρόγραμμα Front Designer. Οι διαστάσεις του κουτιού βρέθηκαν από τα datasheets του κατασκευαστή με κωδικό OKW:B4418107 και φαίνονται στο ακόλουθο σχήμα. Το πλάτος και το μήκος του ανέρχονται στα 169.8mm, ενώ η καμπύλη στις γωνίες στα 9.9mm.

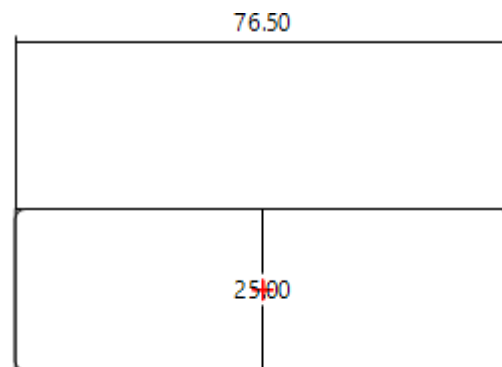


Διαστάσεις κουτιού

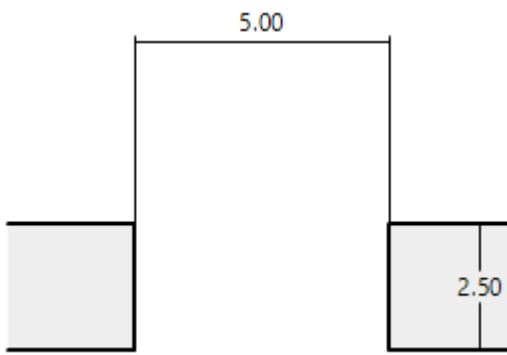
Οι διαστάσεις των υπόλοιπων υλικών μετρήθηκαν με παχύμετρο, με σκοπό την σωστή τοποθέτηση τους στα ανοίγματα που πραγματοποιήθηκαν στο κουτί. Στα παρακάτω σχήματα φαίνονται οι διαστάσεις τους.



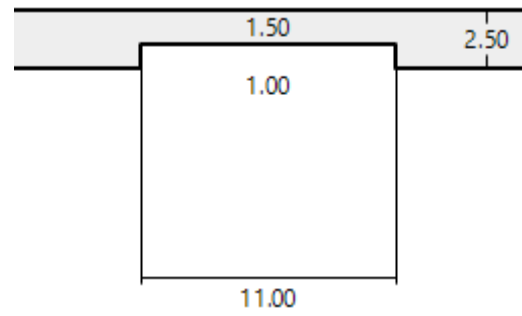
Διαστάσεις ηλεκτρολογίου



Διαστάσεις οθόνης

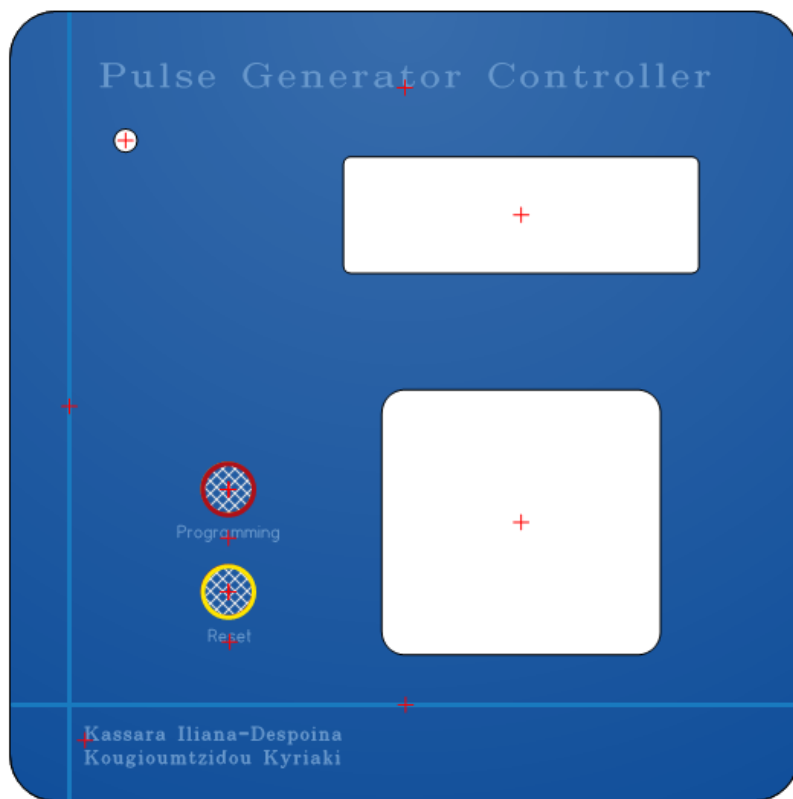


Διαστάσεις LED

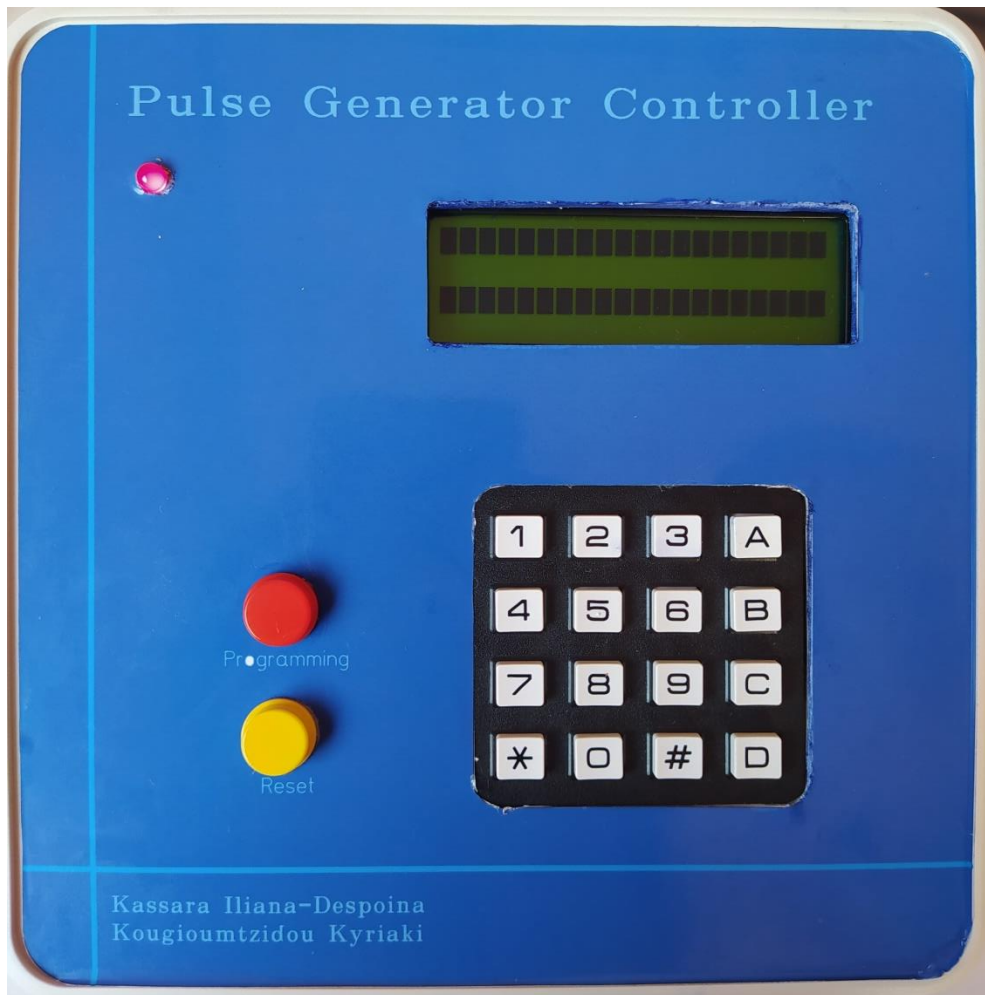


Διαστάσεις Button

Στα παρακάτω σχήμα 4.7 φαίνεται το ολοκληρωμένο και τελικό σχέδιο, ενώ στο σχήμα 4.8 έχουμε την ολοκληρωμένη και τελική μορφή του κουτιού κατασκευής.



Σχήμα 4.7 Αυτοκόλλητο

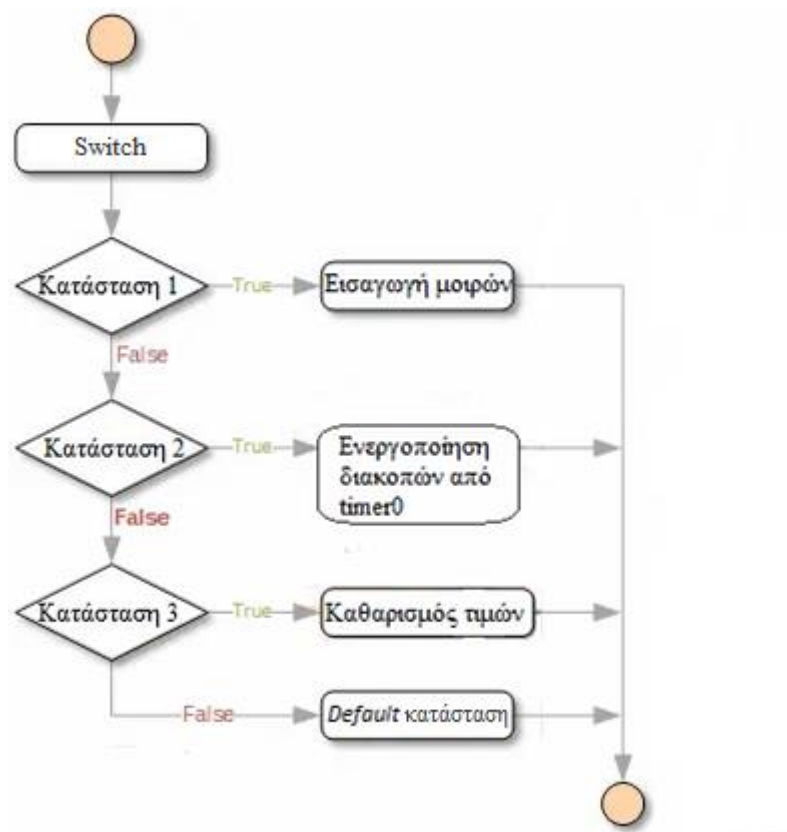


Σχήμα 4.8 Τελική κατασκευή

4.5 CCS C compiler

CCS C compiler είναι ο πιο δημοφιλής μεταγλωττιστής για τους Microchip PIC® μικροελεγκτές. Ο compiler αυτός περιλαμβάνει pro-level optimization, την μεγαλύτερη βιβλιοθήκη από ενσωματωμένες συναρτήσεις και εντολές pre-processor συγκεκριμένα για τους PIC μικροελεγκτές. Επιπλέον, περιλαμβάνει έτοιμους drivers για LCD modules, Keypads, Serial EEPROM, Real-time Clocks, Touch Sensors, Memory Devices, A / D Converters, Temperature sensors, Digital Potentiometers, I / O Expanders και άλλα. Τα Drivers βρίσκονται στο φάκελο εγκατάστασης του compiler: C:\Program Files (x86)\PICC\Drivers. Ο Standard Header file για τον μικροελεγκτή PIC18F4550 βρίσκεται στο φάκελο εγκατάστασης του compiler: C:\Program Files (x86)\PICC\Devices.

4.6 Διάγραμμα ροής κώδικα



4.7 Πηγαίος κώδικας του προγράμματος

```
#include<main.h>
#include<flex_lcd-kassara4.h>
#include< keypad.h>

#byte PORTA=0xF80
#byte PORTB=0xF81
#byte PORTC=0xF82
#byte PORTD=0xF83

char state=1;

char N1=0;
float deg1on=0.0;
float deg1off=120.0;

float SCR1=0.0;
float SCR2=0.0;
float SCR3=0.0;
float SCR4=0.0;
float SCR5=0.0;
float SCR6=0.0;

float SCR1a=60.0;
float SCR2a=120.0;
float SCR3a=180.0;
float SCR4a=240.0;
float SCR5a=300.0;
float SCR6a=0.0;

int8 last_value;
int1 flag1;
```

```
//----- Βιβλιοθήκες προγράμματος -----
//Ρουτίνα main
//Ρουτίνα flex_lcd για εμφάνιση στην οθόνη
//Ρουτίνα ανάγνωσης από το πληκτρολόγιο
//-----
//Ορισμός των θυρών με τη θέση τους στη μνήμη
//Καθορισμός του καταχωρητή δεδομένων της
//πόρτας A
//Καθορισμός του καταχωρητή δεδομένων της
//πόρτας B
//Καθορισμός του καταχωρητή δεδομένων της
// πόρτας C
//Καθορισμός του καταχωρητή δεδομένων της
// πόρτας D
//----- Ορισμός μεταβλητών -----
//Μεταβλητή που δείχνει ποιο ψηφίο πρόκειται
// να εισαχθεί στη δομή switch(state) { }
//Μεταβλητή για αποθήκευση του ASCII ψηφίου
//Γωνία έναυσης που λαμβάνει τιμές από 0-100°
//Γωνία θέσης σε αποκοπή της κυματομορφής στις
//120°
//Μεταβλητές στις οποίες θα αποθηκευτεί η
//μετατροπή της εισαχθείσας γωνίας σε βήμα
//counter

//Μεταβλητές για πρόσθεση των αρχικών γωνιών
//των SCRs και των γωνιών έναυσης

//Η τελευταία τιμή της πόρτας B πριν τη διακοπή
//Χρησιμοποιείται ως σήμα ένδειξης στο
// πρόγραμμα ότι έχει φτάσει στην επιθυμητή
//κατάσταση
```

```

float counter=0.0;
float d1=0;

int16 Pr=1;
int16 timerinitial=64335;

void init (void);
void clearall(void);
void timer0_int(void);
void rb(void);
void initialangles(void);

void transfervalues(void);

void main()
{
  init();
  delay_ms(500);
  lcd_init();
  clearall();
  kbd_init();

  char k;

  while(TRUE){
    switch(state){
    case 1 :
      k=kbd_getc();

      while (k!=0){

        if(k<0x30){state=1;}
        if(k=='A') {initialangles();
          state = 2; lcd_gotoxy(1,1);
          printf(lcd_putc, "%2u",state);
          }
        if(k=='C') {transfervalues();state=1;}

        if(k=='D') {clearall();state=1;}
        break; }

//Αρχικοποίηση του counter
//Μεταβλητή στην οποία καταχωρείται η
//μετατροπή του αριθμού 360 σε βήμα counter
//Η τιμή του Prescaler του Timer0
//Αρχική τιμή για διακοπές κάθε 100μs
//----- Ορισμός συναρτήσεων -----
//Συνάρτηση αρχικοποίησης
//Συνάρτηση καθαρισμού οθόνης
//Συνάρτηση αρχικοποίησης διακοπών από timer
//Συνάρτηση αρχικοποίησης διακοπών από rb
//Συνάρτηση τοποθέτησης και εμφάνισης αρχικών
//γωνιών
//Συνάρτηση μεταφοράς των τιμών που εισάγαμε
//στις μεταβλητές για την παραγωγή των
//κυματομορφών εξόδου
//----- Κύριο πρόγραμμα -----

//Ρουτίνα αρχικοποίησης
//Καθυστέρηση 500ms
//Αρχικοποίηση ρουτίνας εμφάνισης στην οθόνη
//Συνάρτηση καθαρισμού οθόνης και μεταβλητών
//Αρχικοποίηση της συνάρτησης ανάγνωσης από
//το πληκτρολόγιο
//Μεταβλητή στην οποία αποθηκεύεται ο ASCII
//κώδικας του πλήκτρου που πατήθηκε

//Διαβάζει την τιμή που επιστρέφει η συνάρτηση
//ανάγνωσης από το πληκτρολόγιο
//Αν πατήθηκε ένα πλήκτρο εκτελείται το
//περιεχόμενο των αγκυλών του while γιατί
//τότε είναι και διάφορο του 0. Τότε εμφανίζεται
//η τιμή του πλήκτρου που πατήθηκε, το σύμβολο
//χ (πολλαπλασιασμός) και γίνεται μετάβαση
//state=2 για να εισαχθεί ο 2ος αριθμός

//Αν πατήθηκε το 'A' πήγαμε state=2 για
//εισαγωγή των γωνιών έναρξης των παλμών
//Πήγαμε state=1 εισαγωγή ακέραιου μέρους

//Αν πατηθεί το 'C' γίνεται η μεταφορά των τιμών
//για τις κυματομορφές
//Αν πατηθεί το 'D' καθαρίζουν όλα

```

```

break;

case 2:
k=kbd_getc();
while(k!=0){

if(k=='D'){ clearall(); break; } //Αν πατηθεί το D γίνεται καθαρισμός όλων

if(k=='='){ //Αν πατηθεί το = γίνεται η πρόσθεση των
//αρχικών τιμών των SCRs και της εισαχθείσας
//γωνίας

SCR1a=((1*60)+deg1on);
SCR2a=((2*60)+deg1on);
SCR3a=((3*60)+deg1on);
SCR4a=((4*60)+deg1on);
SCR5a=((5*60)+deg1on);
SCR6a=(deg1on);
state=1; lcd_gotoxy(1,1); printf(lcd_putc,"%2u", state); //Επιστροφή στο state=1
break;
}

if(k=='C') { transfervalues(); break; } //Αν πατηθεί το C γίνεται μεταφορά των
//υπολογισμένων τιμών στις κυματομορφές
//εξόδου

if(k<0x30||k>0x39){ state=2; break; } //Αν πατηθεί νούμερο, γίνεται η απεικόνισή του
//στην οθόνη

N1=k & 0b00001111;
deg1on=(float)N1+deg1on*10;
lcd_gotoxy(5,2);
printf(lcd_putc,"%f",deg1on);
break;
}
break;
}
}

//----- Καθαρισμός οθόνης -----

void clearall (void){
timerinitial=64335; //Αρχική τιμή για διακοπές κάθε 100 μs
Pr=1; //Αρχική τιμή prescaler=1
printf(lcd_putc,"\f"); //Καθαρισμός οθόνης και τοποθέτηση του
//κέρσορα στην αρχική θέση
lcd_gotoxy(3,1); //Αρχικές και μόνιμες εμφανίσεις στην οθόνη
lcd_putc("ANGLES IN DEGREES"); //κατά την ενεργοποίησή της

```

```

lcd_gotoxy(1,2); lcd_putc("ON1:"); //Τοποθέτηση του deg1on
lcd_gotoxy(11,2); lcd_putc("OFF1:"); //Τοποθέτηση του deg1off
initialangles(); //Τοποθέτηση και εμφάνιση των αρχικών
//γωνιών
state=1; //Κατάσταση 1 για εισαγωγή νέων ψηφίων και
} //νέου κύκλου εργασίας
//-----
void initialangles(void){ //Ρουτίνα εισαγωγής και εμφάνισης των
//αρχικών γωνιών
deg1on=0; //Αρχική γωνία θέσης σε αγωγή της
//κυματομορφής
deg1off=120; //Αρχική γωνία θέσης σε αποκοπή της
//κυματομορφής
lcd_gotoxy(5,2); printf(lcd_putc,"% .1f",deg1on); //Εμφάνιση εισαγωγής του deg1on
lcd_gotoxy(16,2); printf(lcd_putc,"% .1f",deg1off); //Εμφάνιση διάρκειας θετικού παλμού στις 120°
}
//-----
void transfervalues(void){ //Μεταφορά τιμών στις κυματομορφές εξόδου
deg1off=100.0*(120.0/180.0); //Μετατροπή deg1off σε βήμα του counter
d1=100.0*(360.0/180.0); //Μετατροπή 360° σε βήμα του counter
//Μετατροπή των τελικών υπολογισμένων τιμών
//σε βήμα counter
SCR1=100.0*(SCR1a/180.0);
SCR2=100.0*(SCR2a/180.0);
SCR3=100.0*(SCR3a/180.0);
SCR4=100.0*(SCR4a/180.0);
SCR5=100.0*(SCR5a/180.0);
SCR6=100.0*(SCR6a/180.0);
}
//-----
void init (void){ //Ρουτίνα αρχικοποίησης
set_tris_a(0x00); //Η θύρα A γίνεται έξοδος
set_tris_b(0xE0); //Η θύρα B γίνεται έξοδος και είσοδος
set_tris_c(0x00); //Η θύρα C γίνεται έξοδος
enable_interrupts(INT_RB); //Ενεργοποίηση της διακοπής του RB
enable_interrupts(INT_TIMER0); //Ενεργοποίηση της διακοπής του TIMER0
SETUP_TIMER_0(RTCC_INTERNAL | T0_DIV_1); //Το SETUP του timer0 παίρνει παλμούς από
το εσωτερικό ρολόι.
PORTC=0x00; //Μηδενισμός της πόρτας C
set_timer0(64335); //Αρχική τιμή του timer
enable_interrupts(Global); //Ενεργοποίηση του γενικού διακόπτη των
//διακοπών
deg1on=0; //Αρχική γωνία θέσης σε αγωγή της
//κυματομορφής

```

```

degloff=120; //Αρχική γωνία θέσης σε αποκοπή της
state=1; //κυματομορφής
} //Εμφάνιση της κατάστασης state

// - διακοπή timer0 με μεγάλη προτεραιότητα -

#INT_TIMER0 HIGH
void timer0_int(void){

set_timer0(64335); //Αρχική τιμή του timer0

counter=counter+1.0; //Αύξηση του counter κατά 1

if((counter>=SCR1)&&(counter<=SCR1+degloff)) //Αν η τιμή του counter είναι μεγαλύτερη ή
{ // ίση της τιμής του SCR1 και μικρότερη ή
// ίση από το άθροισμα της τιμής SCR1 και
// τις 120° αποκοπής της κυματομορφής
output_high(PIN_C0);} //κάνε το pin C0 1
else { output_low(PIN_C0);} //αλλιώς κάνε το 0

if((counter>=SCR2)&&(counter<=SCR2+degloff)) //Αν η τιμή του counter είναι μεγαλύτερη ή
{ // ίση της τιμής του SCR2 και μικρότερη ή
// ίση από το άθροισμα της τιμής SCR2 και
// τις 120° αποκοπής της κυματομορφής
output_high(PIN_C1);} //κάνε το pin C1 1
else {output_low(PIN_C1);} //αλλιώς κάνε το 0

if(((counter>=SCR3)&&(counter<=(SCR3+degloff))||//Αν η τιμή του counter είναι μεγαλύτερη
((counter>=(SCR3-d1))&& (counter<(SCR3+degloff-d1)))) //ή ίση της τιμής του SCR3 και μικρότερη ή
{ //ίση από το άθροισμα της τιμής SCR3 και
//τις 120° αποκοπής της κυματομορφής ή
//αν είναι μεγαλύτερη από 360°,
//αφαίρεσε από τον counter τις 360°
output_high(PIN_C2);} //κάνε το pin C2 1
else {output_low(PIN_C2);} //αλλιώς κάνε το 0

if(((counter>=SCR4)&&(counter<=(SCR4+degloff))||//Αν η τιμή του counter είναι μεγαλύτερη
((counter>=(SCR4-d1))&& (counter<(SCR4+degloff-d1)))) //ή ίση της τιμής του SCR4 και μικρότερη ή
{ //ίση από το άθροισμα της τιμής SCR4 και
//τις 120° αποκοπής της κυματομορφής ή
//αν είναι μεγαλύτερη από 360°,
//αφαίρεσε από τον counter τις 360°
output_high(PIN_B0);} //κάνε το pin B0 1
else {output_low(PIN_B0);} //αλλιώς κάνε το 0

```

```

if(((counter>=SCR5)&&(counter<=(SCR5+deg1off)))||//An η τιμή του counter είναι μεγαλύτερη
((counter>=(SCR5-d1))&&(counter<(SCR5+deg1off-d1))))
//ή ίση της τιμής του SCR5 και μικρότερη ή
//ίση από το άθροισμα της τιμής SCR5 και
//τις 120° αποκοπής της κυματομορφής ή
{
//αν είναι μεγαλύτερη από 360°,
//αφαίρεσε από τον counter τις 360°
output_high(PIN_ A4); }
//κάνε το pin A4 1
else {output_low(PIN_ A4);}
//αλλιώς κάνε το 0

if((counter>=SCR1)&&(counter<=SCR1+deg1off)) //An η τιμή του counter είναι μεγαλύτερη ή
// ίση της τιμής του SCR6 και μικρότερη ή
{
// ίση από το άθροισμα της τιμής SCR6 και
// τις 120° αποκοπής της κυματομορφής
output_high(PIN_ A5);}
//κάνε το pin A5 1
else { output_low(PIN_ A5); }
//αλλιώς κάνε το 0
}

#INT_RB //----- διακοπές με RB -----
//Ανιχνεύεται από ποιον ακροδέκτη γίνεται η
//διακοπή(RB5) κι αν η αλλαγή γίνεται από
//HIGH LOW ο counter θα πρέπει να
//μηδενίζεται

void rb(void){

int8 changes;
changes=last_value^PORTB; //Η μεταβλητή changes ισοδυναμεί με την
//τελευταία τιμή της πόρτας B πριν τη
//διακοπή και γίνεται XOR μεταξύ τρέχουσας
//και προηγούμενης κατάστασης

if(bit_test(changes,5)&&bit_test(last_value,5)){ //Εκτελείται κατά την μετάβαση του RB5 από
//1 σε 0
//μηδένισε τον counter
//Η αλλαγή γίνεται από το LOW
}

counter=0;
flag1=0;
}

last_value= PORTB; //Το περιεχόμενο της πόρτας B ισοδυναμεί με
// την τελευταία τιμή της πόρτας B
}

```

4.8 Proteus

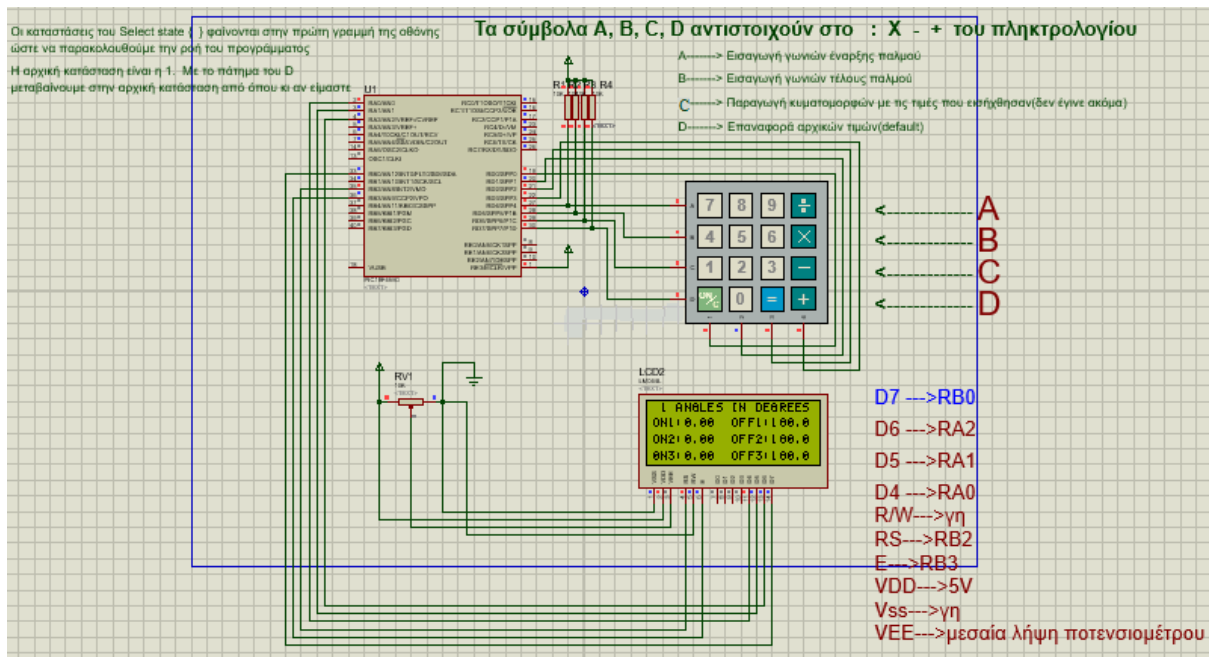
4.8.1 Δοκιμή λειτουργίας εισαγωγής παλμών και εμφάνιση στην οθόνη

Με σκοπό να βεβαιωθούμε ότι ο κώδικας μας «δουλεύει», δηλαδή πραγματοποιείται η εισαγωγή και προβολή των μοιρών στην LCD οθόνη, χρησιμοποιήσαμε το πρόγραμμα Proteus, ώστε να κάνουμε την παραπάνω προσομοίωση. Για να μπορέσουμε να προσομοιώσουμε τον κώδικα από το C Compiler στο Proteus, χρησιμοποιήσαμε τις παρακάτω βιβλιοθήκες, στις οποίες έγιναν οι απαραίτητες αλλαγές που θα τις καθιστούσαν κατάλληλες για το πρόγραμμα προσομοίωσης (θα τοποθετηθούν παρακάτω οι κώδικες) :

```
#include<main-proteus.h>
```

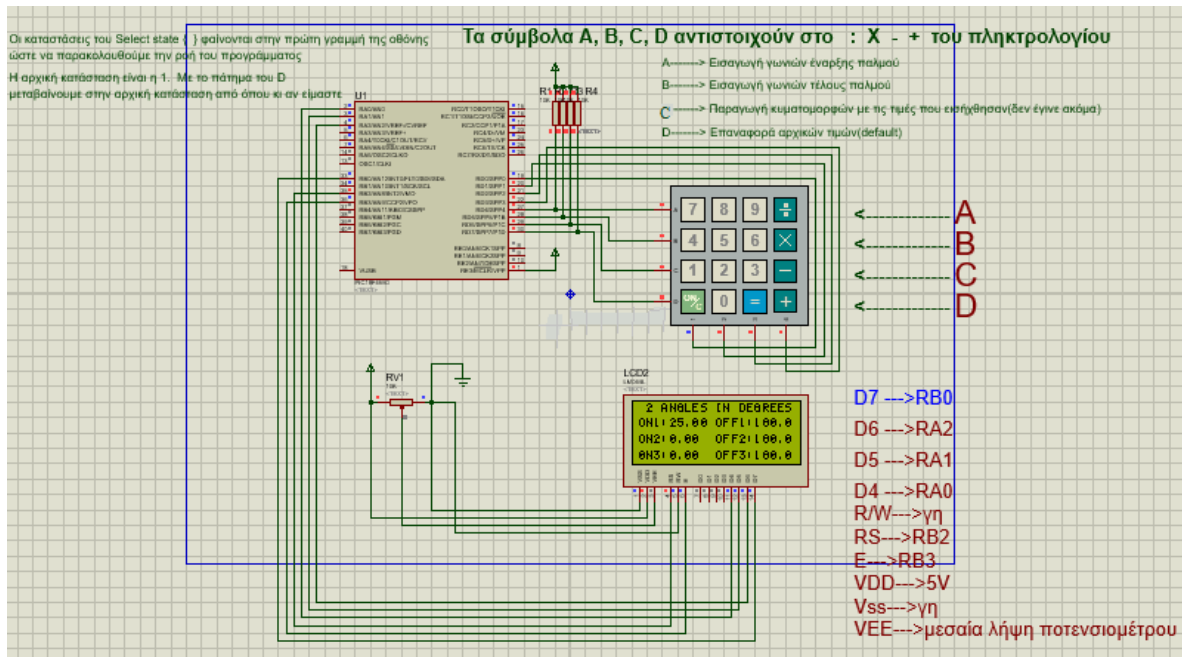
```
#include<flex_lcd-proteus-kassara4.h>
```

```
#include<keypad-proteus.h>
```



Σχήμα 4.9 Αρχική κατάσταση προγράμματος

Αφού «φορτώθηκε» ο κώδικας στο μικροελεγκτή του προγράμματος, βρεθήκαμε στην κατάσταση του σχήματος 4.9 και ήμασταν έτοιμοι για εισαγωγή των μοιρών του πρώτου παλμού πατώντας το σύμβολο της διαίρεσης (A). Έτσι, από την κατάσταση 1 μεταβαίνουμε στην κατάσταση 2, κατάσταση στην οποία μπορεί να γίνει η εισαγωγή των μοιρών.

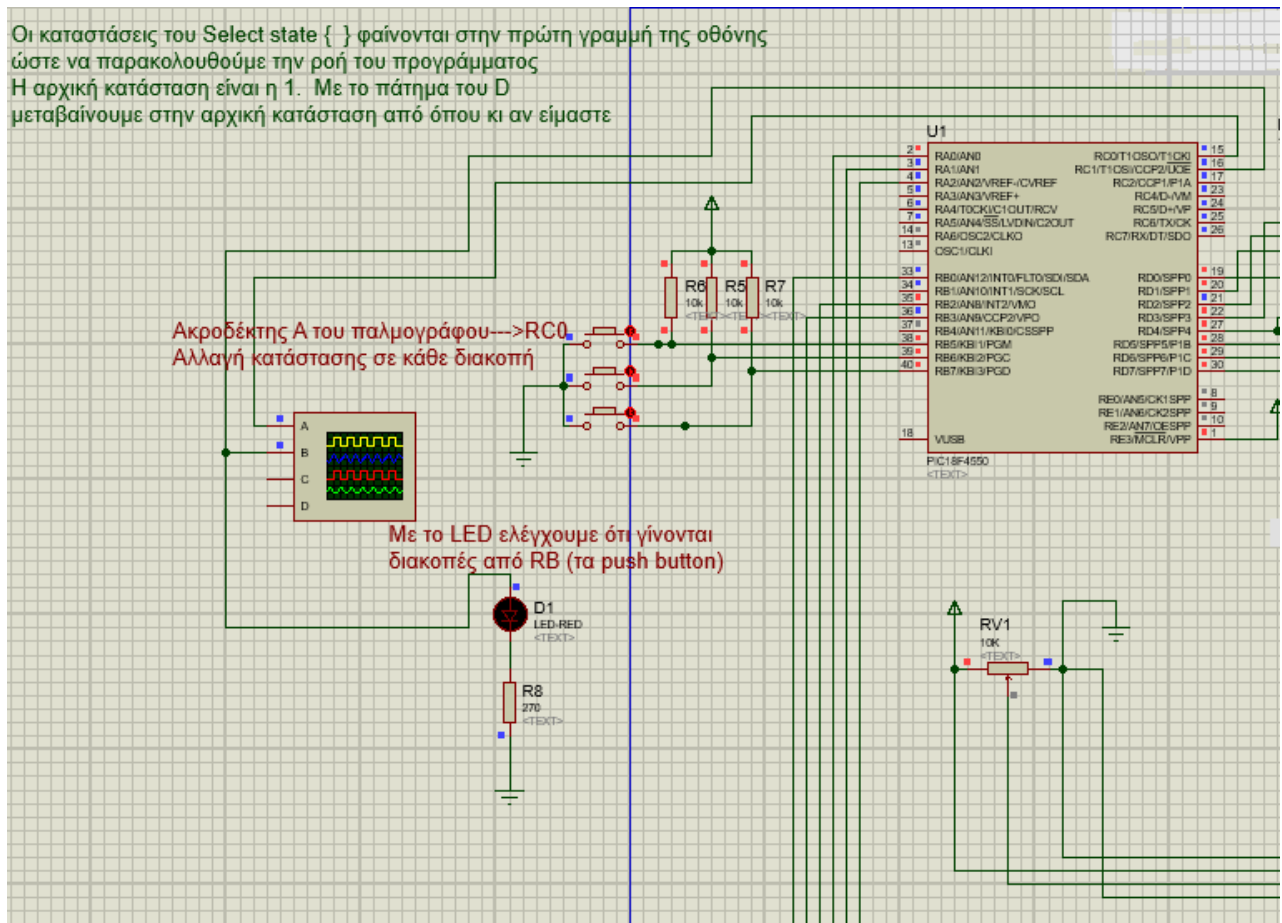


Σχήμα 4.10 Εισαγωγή 25° από το πληκτρολόγιο

Στο σχήμα 4.10 έγινε εισαγωγή 25°. Με το πάτημα του συμβόλου = (Enter), γίνεται ο υπολογισμός της τελικής καθυστέρησης που θα έχουν οι παλμοί εξόδου και με το πάτημα του συμβόλου C γίνεται η μεταφορά των τιμών αυτών στις κυματομορφές εξόδου.

Όταν τελειώσουμε και με αυτήν την εισαγωγή μπορούμε να επιστρέψουμε στην αρχική κατάσταση 1 πατώντας το σύμβολο της πρόσθεσης (D) για μια καινούρια εισαγωγή.

4.8.2 Δοκιμή λειτουργίας διακοπών από RB και timer0



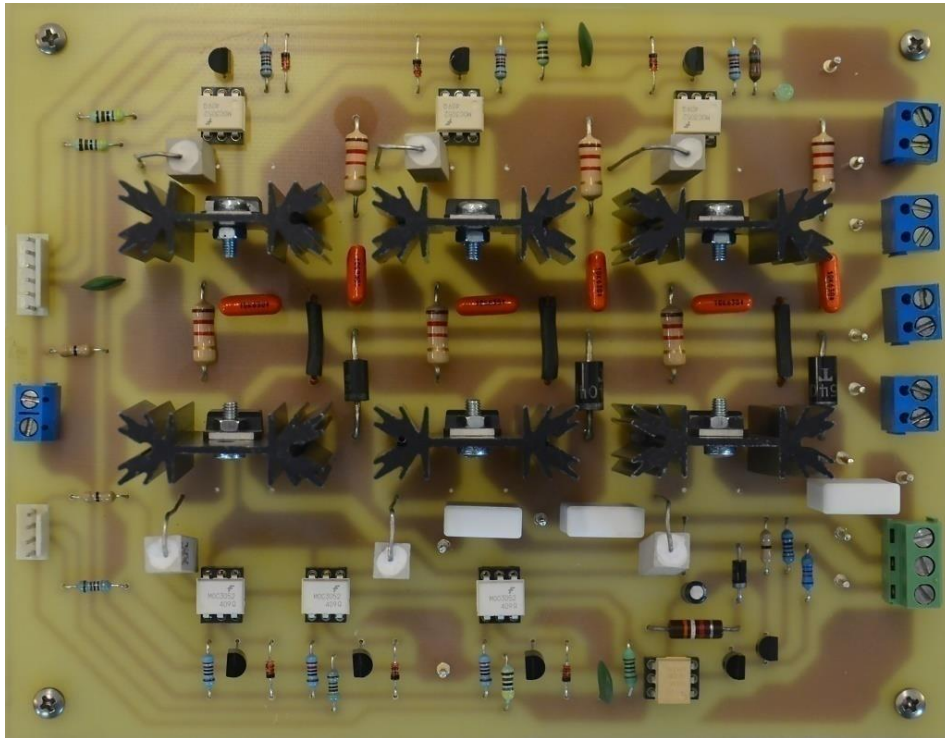
Σχήμα 4.11 Δοκιμή διακοπών από τον timer0

Στη ρουτίνα διακοπών από RB προσθέσαμε μια εντολή $PORTC=PORTC\wedge 00000010$, ώστε να αλλάζει την κατάσταση του RC1 σε κάθε διακοπή που δημιουργούμε από το πάτημα του μπουτόν της προσομοίωσης. Έτσι, θα βεβαιωθούμε πως δουλεύουν οι διακοπές από RB. Με το πάτημα του μπουτόν, παρατηρήσαμε ότι άναβε το LED, που είναι συνδεδεμένο στον ακροδέκτη RC1 του μικροελεγκτή, καθώς επίσης ανεβαίνει και η στάθμη στο κανάλι B του παλμογράφου.

Όσον αφορά τη ρουτίνα διακοπών του timer0 προσθέσαμε μια εντολή $PORTC=PORTC\wedge 0b00000001$, ώστε να αλλάζει την κατάσταση του RC0 σε κάθε διακοπή του timer0. Στην προσομοίωση παρατηρήθηκε πως με κάθε διακοπή του timer0 η κατάσταση του καναλιού A μετέβαινε από τα 0 στα +5Volt.

Η μετάβαση της τάσης στα +5Volt οφείλεται στο ότι έχουμε ορίσει τους ακροδέκτες RC0-RC1-RC2 για τις θετικές ημιπεριόδους και αντίστοιχα για τις αρνητικές ημιπεριόδους έχουμε ορίσει τους ακροδέκτες RA3-RA4-RA5.

Έπειτα, αφού ολοκληρώσαμε το πρώτο μέρος της κατασκευής μας, υλοποιήσαμε και την δεύτερη πλακέτα της κατασκευής που σχετίζεται με τα ηλεκτρονικά ισχύος. Στο σχήμα 4.12 βλέπουμε την πλακέτα της τριφασικής πλήρως ελεγχόμενης γέφυρας ανόρθωσης που κατασκευάσαμε και χρησιμοποιήσαμε για την οδήγηση του κινητήρα και του λαμπτήρα.



Σχήμα 4.12 Πλακέτα κυκλωμάτων οδήγησης και τριφασικής γέφυρας

Κεφάλαιο 5^ο

Αποτελέσματα και Συμπεράσματα

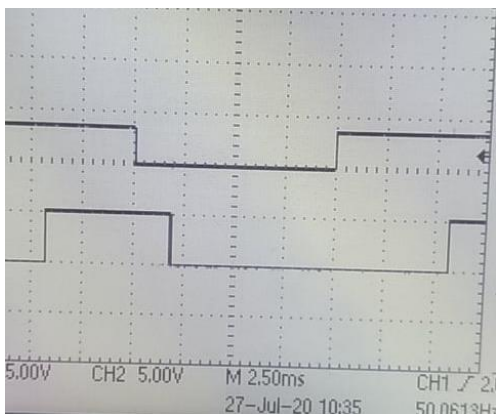
5.1 Αποτελέσματα

Για να ελέγξουμε αν ο κώδικας φέρει τις επιθυμητές κυματομορφές εξόδου και λειτουργεί σωστά προβήκαμε στις παρακάτω ενέργειες. Αρχικά, συνδέσαμε στην RB5 είσοδο του μικροελεγκτή την γεννήτρια, οι ρυθμίσεις της οποίας φαίνονται στο σχήμα 5.1. Έπειτα, συνδέσαμε το πρώτο κανάλι του παλμογράφου στο ίδιο πόδι του μικροελεγκτή για να βλέπουμε τον παλμό εισόδου της γεννήτριας. Στη συνέχεια, για να δούμε το σήμα εξόδου συνδέσαμε το δεύτερο κανάλι του παλμογράφου στο πόδι C2 του μικροελεγκτή, που έχουμε συνδεδεμένο τον SCR3.

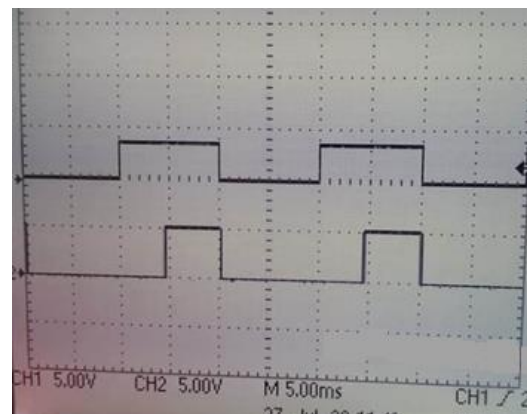


Σχήμα 5.1 Ρυθμίσεις γεννήτριας

Τροφοδοτώντας την συσκευή, η αρχική τιμή του SCR3 είναι $3 \times 60^0 = 270^0$. Εισάγοντας σαν διαφορά φάσης 90^0 παρατηρήσαμε πως ο παλμός δεν ξεκινούσε κατά την μετάβαση του παλμού εισόδου από την άνοδο στην κάθοδο όπως θα έπρεπε, αλλά κάποιες μοίρες μετά. Παρόλα αυτά η διαφορά του αναμενόμενου παλμού από τον παραγόμενο παλμό ήταν οι αναμενόμενες μοίρες, διότι κατά την εισαγωγή 90^0 , γίνεται η αφαίρεση $390^0 - 360^0 = 30^0$ (όπου $390^0 = 270^0 + 120^0$). Μια ακόμη επαλήθευση του παραγόμενου παλμού, ήταν η ταύτιση του μετρούμενου χρόνου μέσω του παλμογράφου με το πηλίκο της διαίρεση των $30^0/180^0$ (για να μετατρέψουμε τις μοίρες σε χρόνο). Το ίδιο συνέβη όταν κάναμε και την εισαγωγή των 70^0 . Η διαφορά του παραγόμενου παλμού ήταν στις 10^0 όπως φαίνεται στα παρακάτω σχήματα.



Σχήμα 5.2 Εισαγωγή 90 μοιρών

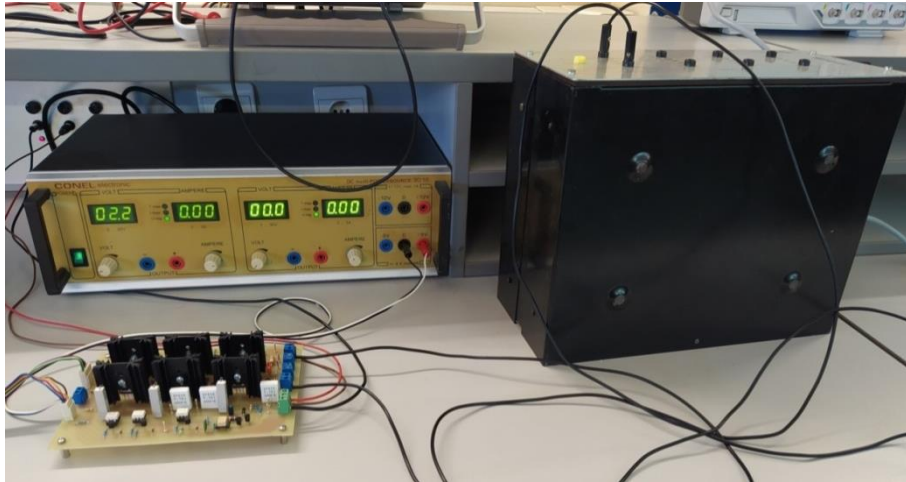


Σχήμα 5.3 Εισαγωγή 70 μοιρών

Τα παραπάνω δοκιμάσαμε και για 0° - 100° . Έτσι, ξέροντας πλέον πως ο κώδικας μας λειτουργεί κανονικά, ήμασταν έτοιμοι να ολοκληρώσουμε το πειραματικό μας μέρος και να καταγράψουμε τα αποτελέσματα.

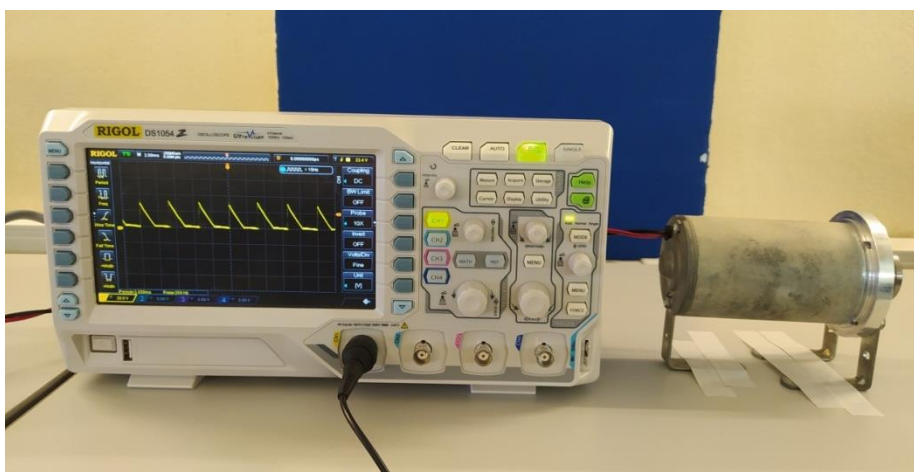
Για τη λήψη των αποτελεσμάτων της κατασκευής μας προβήκαμε στις εξής ενέργειες:

- ❖ Στην έξοδο του κυκλώματος τοποθετήσαμε σε σειρά αντίσταση των 25Ω για οδήγηση των φορτίων.
- ❖ Έπειτα, τροφοδοτήσαμε την είσοδο της γέφυρας από το τριφασικό δίκτυο, και τα κυκλώματα οδήγησης και zero crossing detect από το τροφοδοτικό του εργαστηρίου με $5V$.



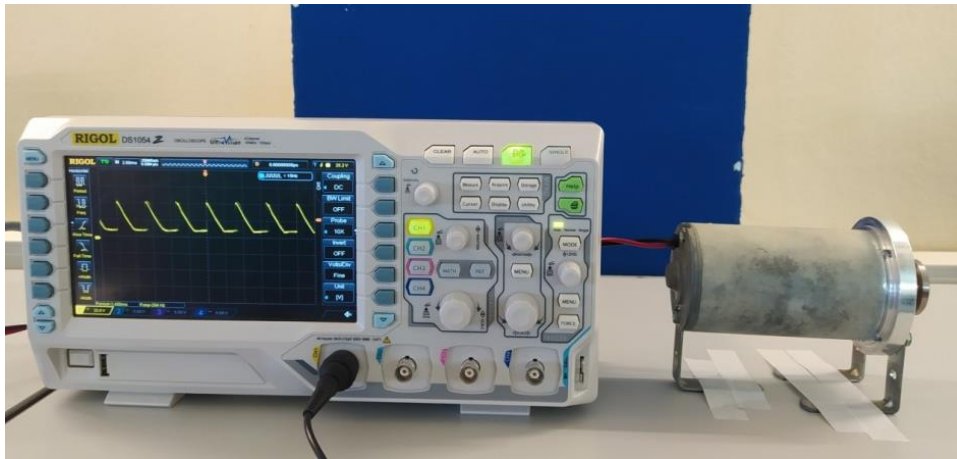
Σχήμα 5.4 Λήψη αποτελεσμάτων κατασκευής

Πρώτα συνδέσαμε στην έξοδο του κυκλώματος τον DC κινητήρα, ξεκινώντας την εισαγωγή από τις 100° . Ωστόσο, δεν παρατηρήσαμε κάποια κίνηση σε αυτόν λόγω της μικρής τάσης που είχε, γεγονός που δεν επέτρεπε την περιστροφή του. Το ίδιο συνέβη και με τις 90° . Κάνοντας εισαγωγή των 80° ξεκινήσαμε να παρατηρούμε κίνηση. Τα αποτελέσματα της εισαγωγής φαίνονται στο σχήμα 5.5.

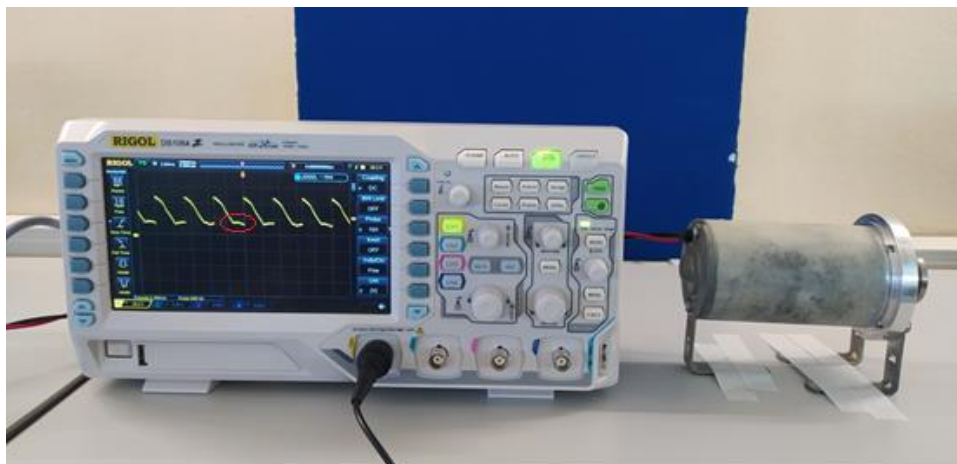


Σχήμα 5.5 Κυματομορφή εξόδου με καθυστέρηση φάσης 80°

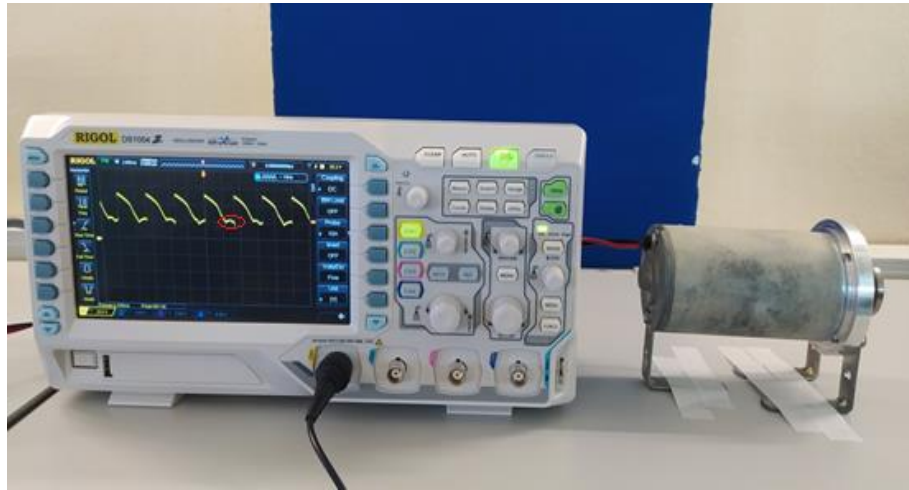
Στη συνέχεια, αρχίσαμε να ελαττώνουμε τις μοίρες. Όσο μικρότερες ήταν οι μοίρες, τόσο γρηγορότερα περιστρεφόταν ο κινητήρας. Συνεπώς, με τη μείωση των μοιρών είχαμε αύξηση της τάσης και κατ' επέκταση αύξηση της ροής ισχύος στον κινητήρα, γεγονός που οδηγούσε σε γρηγορότερη περιστροφή του. Σταματήσαμε την εισαγωγή στις 30° , λόγω της έντονης περιστροφής, διαφορετικά δεν υπήρχε κάποιο πρόβλημα να φτάσουμε στις 0° . Τα αποτελέσματα από τις 70° έως τις 30° φαίνονται στα παρακάτω σχήματα.



Σχήμα 5.6 Κυματομορφή εξόδου με καθυστέρηση φάσης 70°



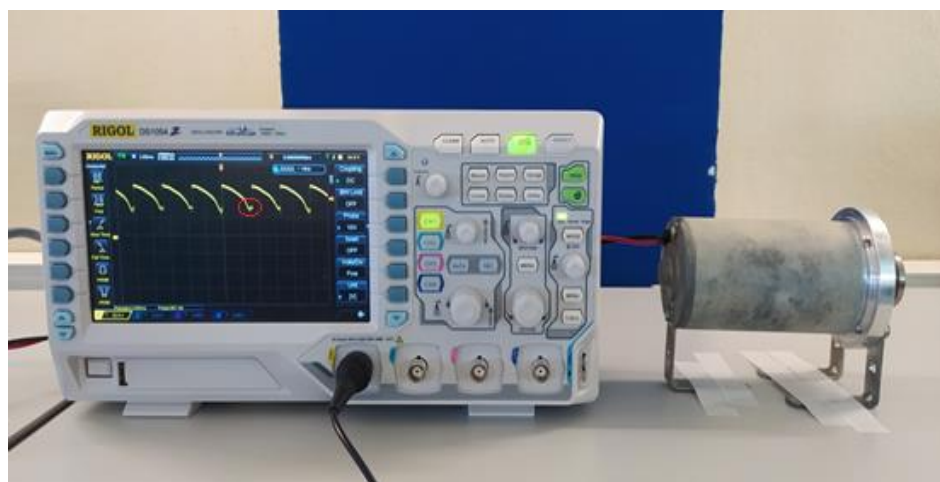
Σχήμα 5.7 Κυματομορφή εξόδου με καθυστέρηση φάσης 60°



Σχήμα 5.8 Κυματομορφή εξόδου με καθυστέρηση φάσης 50°



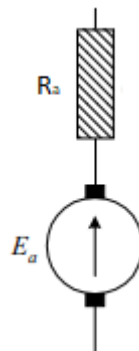
Σχήμα 5.9 Κυματομορφή εξόδου με καθυστέρηση φάσης 40°



Σχήμα 5.10 Κυματομορφή εξόδου με καθυστέρηση φάσης 30°

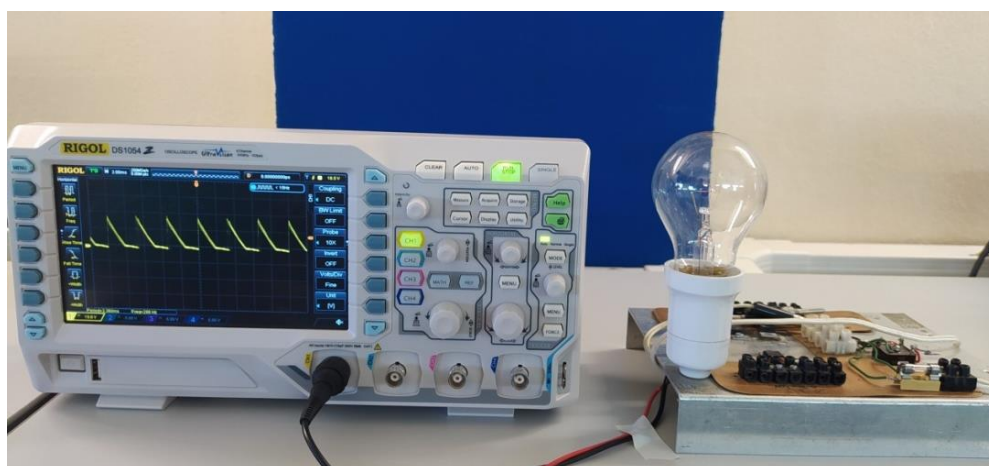
Από τις 60^0 έως τις 30^0 παρατηρούμε στις παραγόμενες κυματομορφές εξόδου ένα σημείο που «τρεμο-παίζει» (κυκλωμένο κόκκινο σημείο). Αυτό οφείλεται στην αντι-ηλεκτρεγερτική δύναμη του κινητήρα, δηλαδή στην εσωτερική τάση που παράγεται στο τύλιγμα του σπλισμού, καθώς αυτός περιστρέφεται μέσα στο κύριο μαγνητικό πεδίο του στάτη. Ο κινητήρας στην ουσία είναι μια αντίσταση σε σειρά με την αντι-ηλεκτρεγερτική δύναμη, η μορφή του οποίου φαίνεται στην παρακάτω σχήμα 5.11. Συνεπώς, εκτός από την αντίσταση των 25Ω , η αντίσταση που έχει ως κινητήρας δημιουργεί μια διαφορά δυναμικού στα άκρα της, για αυτό δεν παρατηρούσαμε κίνηση στις 100 και 90 μοίρες. Η τάση που έφτανε τελικά σε αυτόν ήταν πάρα πολύ μικρή. Η ισχύς στην έξοδο του κινητήρα θα είναι της μορφής:

$$P_{motor} = \frac{V^2}{R_{ολ}} = \frac{V^2}{R_{25\Omega} + (R_{motor} + E_a)} \quad (5.1)$$



Σχήμα 5.11 Ισοδύναμο κύκλωμα του κινητήρα

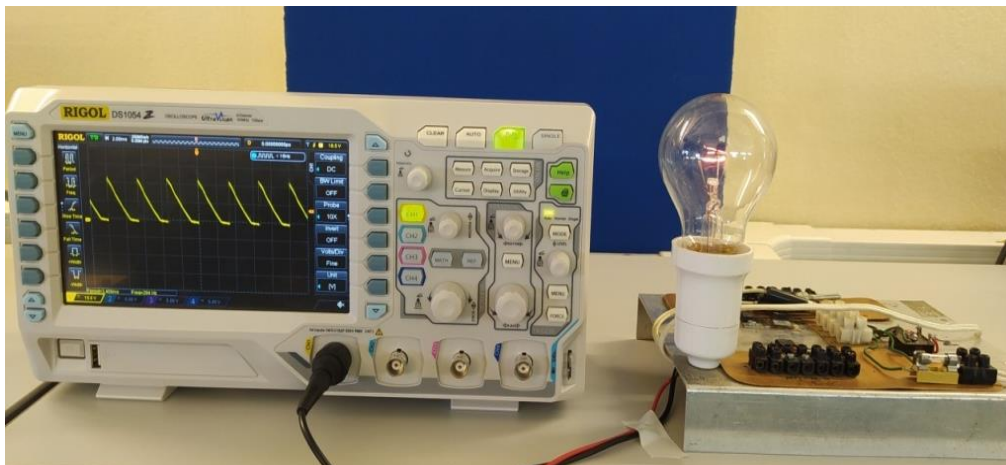
Έπειτα, συνδέσαμε στην έξοδο της γέφυρας έναν λαμπτήρα. Ξεκινήσαμε την εισαγωγή από τις 100^0 , αλλά ίσα που φαινόταν να εκπέμπεται φως από το σύρμα του. Το ίδιο συνέβη και με τις 90^0 . Κάνοντας εισαγωγή των 80^0 , η φωτεινότητα στο σύρμα ήταν εντονότερη και φαίνεται στα παρακάτω σχήματα μαζί με την έξοδο του κυκλώματος.



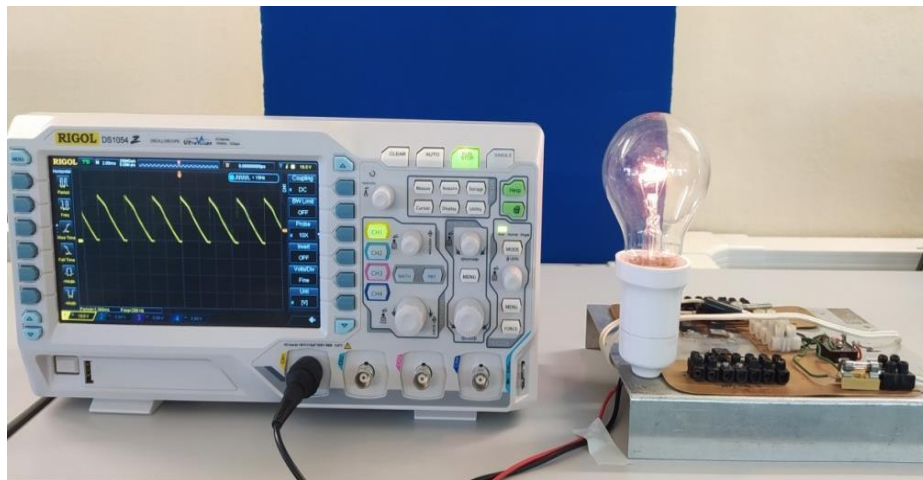


Σχήμα 5.12-5.13 Κυματομορφή εξόδου και φωτεινότητα λαμπτήρα με καθυστέρηση φάσης 80°

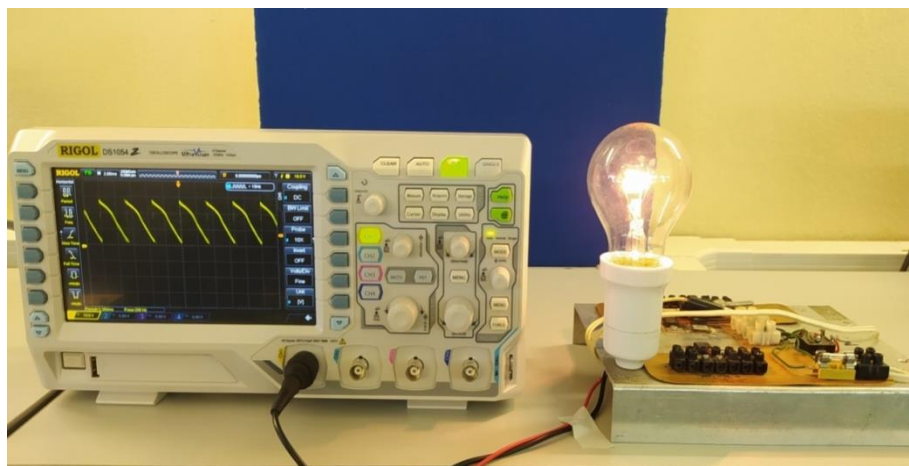
Συνεχίσαμε να ελαττώνουμε τις μοίρες έως στις 0° . Σε αυτό το σημείο παρατηρούμε ότι η μείωση των μοιρών και κατ' επέκταση η αύξηση της τάσης, οδηγεί σε εντονότερη φωτεινότητα του λαμπτήρα. Να σημειωθεί, πως η τάση στις 0° είναι η μέγιστη τάση που μπορώ να «πάρω» και τα αποτελέσματα φαίνονται στα ακόλουθα σχήματα.



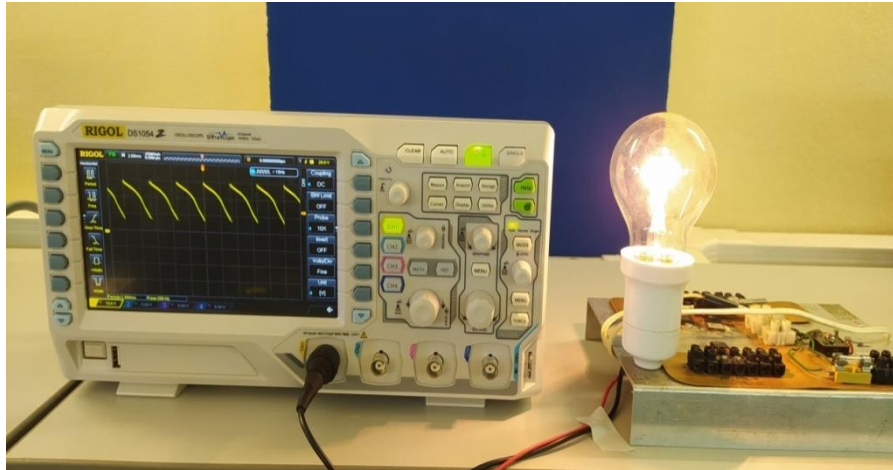
Σχήμα 5.14 Κυματομορφή εξόδου με καθυστέρηση φάσης 70°



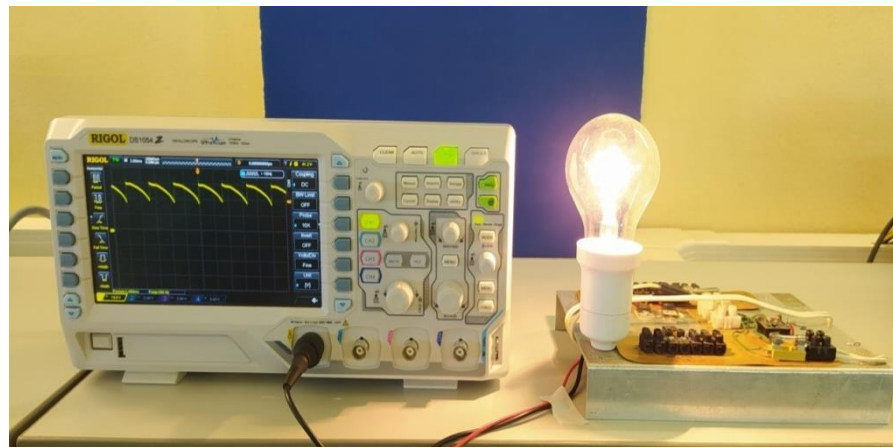
Σχήμα 5.15 Κυματομορφή εξόδου με καθυστέρηση φάσης 60°



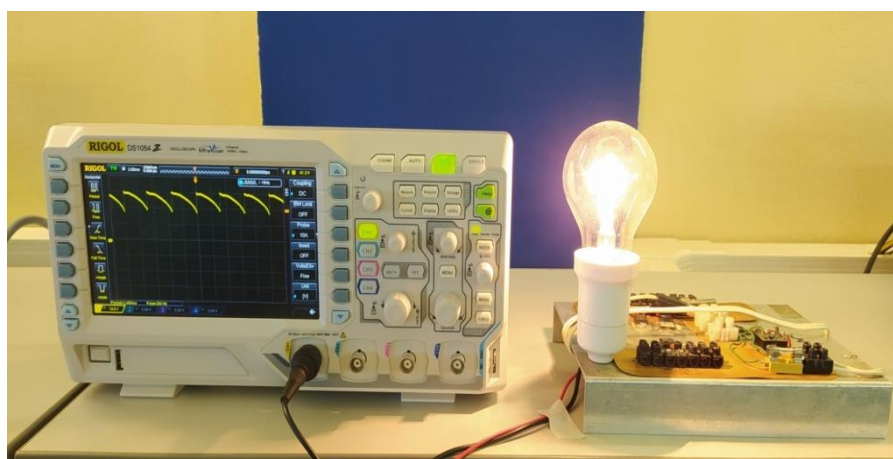
Σχήμα 5.16 Κυματομορφή εξόδου με καθυστέρηση φάσης 50°



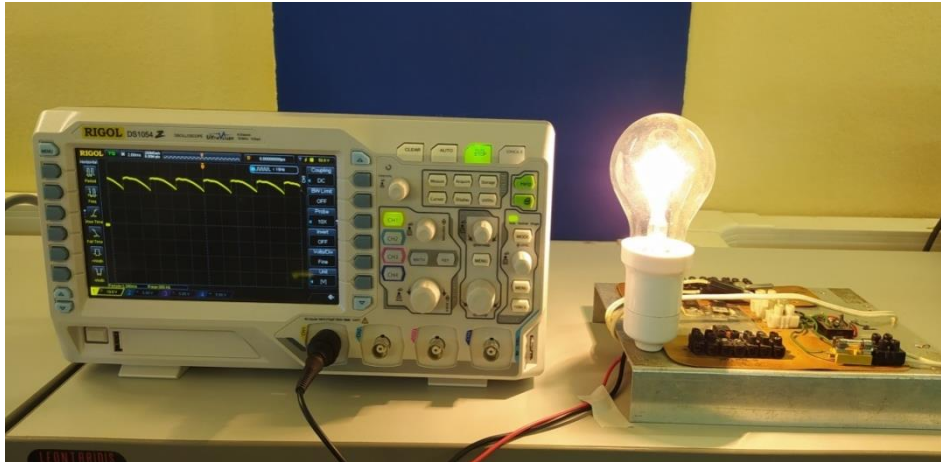
Σχήμα 5.17 Κυματομορφή εξόδου με καθυστέρηση φάσης 40°



Σχήμα 5.18 Κυματομορφή εξόδου με καθυστέρηση φάσης 30°



Σχήμα 5.19 Κυματομορφή εξόδου με καθυστέρηση φάσης 20°



Σχήμα 5.20 Κυματομορφή εξόδου με καθυστέρηση φάσης 10°



Σχήμα 5.21 Κυματομορφή εξόδου με καθυστέρηση φάσης 0°

Σε αυτό το πείραμα πραγματοποιήθηκαν και οι μετρήσεις της τάσης πριν από την αντίσταση των 25Ω , για κάθε μία εισαγωγή μοιρών. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα.

$0^\circ \rightarrow 55,52V$	$50^\circ \rightarrow 32,30V$
$10^\circ \rightarrow 54,50V$	$60^\circ \rightarrow 23,55V$
$20^\circ \rightarrow 51,20V$	$70^\circ \rightarrow 15,42V$
$30^\circ \rightarrow 46,50V$	$80^\circ \rightarrow 08,30V$
$40^\circ \rightarrow 40,20V$	

Πίνακας 2 Αποτελέσματα μετρήσεων

Να σημειωθεί επίσης πως η αντίσταση των 25Ω , που τοποθετήθηκε στην έξοδο της γέφυρας ήταν για λόγους προστασίας. Ο συγκεκριμένος κινητήρας άντεχε έως $24V$ και ο συγκεκριμένος λαμπτήρας έως $42V$. Οι παραπάνω τάσεις που μετρήθηκαν έγιναν πριν την αντίσταση και τον λαμπτήρα. Η ισχύς στην έξοδο του λαμπτήρα θα είναι της μορφής:

$$P_{\lambda\alpha\mu\pi} = \frac{V^2}{R_{o\lambda}} = \frac{V^2}{R_{25\Omega} + R_{\lambda\alpha\mu\pi}} \quad (5.2)$$

5.2 Συμπεράσματα-Προτάσεις βελτιώσεων

Ανακεφαλαιώνοντας, με την συγκεκριμένη διπλωματική εργασία διαπιστώσαμε πως μπορούμε με εύκολο και οικονομικό τρόπο, με τη βοήθεια ενός ελεγκτή, όπως είναι ο PIC18F4550 να ελέγξουμε την ροή ισχύος σε μια συσκευή ισχύος. Ανάλογα με το σήμα που θα στέλνει ο PIC , με την επιθυμητή διαφορά φάσης που θα έχει εισάγει ο χρήστης, θα ελέγχεται το χρονικό διάστημα που το θυρίστορ θα παραμένει ενεργοποιημένο και κατά η συνέπεια η «τάση» που θα στέλνει στο φορτίο. Με αυτόν τον τρόπο θα ελέγχεται και η ροή ισχύος, όπως αναφέρθηκε παραπάνω.

Μια βελτίωση στο σύστημα μας θα ήταν η εισαγωγή και δεκαδικού μέρος στη γωνία που θα εισάγει ο χρήστης για μεγαλύτερη ακρίβεια. Υπενθυμίζεται ότι στο πρόγραμμα τώρα μπορεί να εισάγεται μόνο ακέραιος αριθμός μοιρών.

Βιβλιογραφία

Βιβλία

- [1] Σ.Μπαλτουδάκης, Γ.Πατουλίδης, Ν.Ασημόπουλος, Προγραμματισμός μικροελεγκτών για μηχανικούς, Εκδόσεις Τζιόλα 2017
- [2] Ι.Κιοσκερίδης, Ηλεκτρονικά ισχύος, Εκδόσεις Τζιόλα 2016
- [3] Γιακουμής Άγγελος, Σημειώσεις του Μαθήματος: Μικροελεγκτές ΙΙ στο e-μάθηση, 2017
- [4] Dogan Ibrahim, Advanced PIC microcontroller projects in C, Elsevier Ltd., 2008

Data Sheet

- [4] Microchip, Available: <https://ww1.microchip.com/downloads/en/devicedoc/39632e.pdf>, 2009

Internet Site

- [5] Elprocus, Zero Crossing Detector Circuit and Working, 2013-2020, Available: <https://www.elprocus.com/zero-crossing-detector-circuit-and-working/>
- [6] Microcontroller and Power Electronics, AC Power Control with Thyristor: Phase Angle Control using triac with PIC16F877A, 2013, Available: <http://tahmidmc.blogspot.com/2013/06/power-control-with-thyristor-phase.html>
- [7] Microcontrollerslab.com, AC power control with thyristor using pic microcontroller, 2013-2020, Available: <https://microcontrollerslab.com/ac-power-control-with-thyristor-using-pic/>, <https://microcontrollerslab.com/programmable-ac-power-control-using-pic-microcontroller/>
- [8] Electronics tutorials, Three Phase Rectification, 2020, Available: <https://www.electronics-tutorials.ws/power/three-phase-rectification.html>
- [9] Wikipedia, Θυρίστορ, 2020, Available: <https://el.wikipedia.org/wiki/%CE%98%CF%85%CF%81%CE%AF%CF%83%CF%84%CE%BF%CF%81>

Παράρτημα

A. main.h

```
#include <18F4550.h>
#DEVICE HIGH_INTS=TRUE
#device adc=10

#FUSES NOWDT           //No Watch Dog Timer
#FUSES WDT32768       //Watch Dog Timer uses 1:128 Postscale
#FUSES HSPLL          //High Speed Crystal/Resonator with PLL enabled
#FUSES NOPROTECT     //Code not protected from reading
#FUSES BROWNOUT       //No brownout reset
//#FUSES BORV20       //Brownout reset at 2.0V
#FUSES NOPUT         //No Power Up Timer
#FUSES NOCPD         //No EE protection
#FUSES STVREN        //Stack full/underflow will cause reset
#FUSES NODEBUG       //No Debug mode for ICD
#FUSES NOLVP         //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOWRT        //Program memory not write protected
#FUSES NOWRTD       //Data EEPROM not write protected
#FUSES NOIESO       //Internal External Switch Over mode enabled
#FUSES NOFCMEN      //Fail-safe clock monitor enabled
#FUSES NOPBADEN     //PORTB pins are configured as analog input channels on RESET
#FUSES NOWRTC       //configuration not registers write protected
#FUSES NOWRTB       //Boot block not write protected
#FUSES NOEBTR       //Memory not protected from table reads
#FUSES NOEBTRB      //Boot block not protected from table reads
#FUSES NOCPB        //No Boot Block code protection
#FUSES MCLR         //Master Clear pin enabled
#FUSES NOLPT1OSC    //Timer1 configured for low-power operation
#FUSES NOXINST      //Extended set extension and Indexed Addressing mode disabled (Legacy
mode)
#FUSES PLL5         //Divide By 12(48MHz oscillator input)
#FUSES CPUDIV1      //System Clock by 2
#FUSES USBDIV       //USB clock source comes from PLL divide by 2
#FUSES VREGEN       //USB voltage regulator enabled
#FUSES ICPRT        //ICPRT enabled
#FUSES WRTB
#FUSES NOICPRT
#FUSES NOLPT1OSC
#FUSES CCP2B3
#FUSES NOCPB
#FUSES NOCPD
#FUSES NOWRTC
```

```

#FUSES NOWRTD
#FUSES NOEBTR
#FUSES NOEBTRB

/* systm clock is 48 MHz */
#use delay(clock=4800000,RESTART_WDT)
/* --- BEGIN: changes required for bootloader ----- */

/* ----- */
/* map reset vector and interrupt vector */
/* 0x000-0x7FF is used by the bootloader. The bootloader maps the original */
/* reset vecotr (0x000) to 0x800 and the reset vector (0x008) to 0x800. */
/* ----- */
#build (reset=0x800, interrupt=0x808)

/* ----- */
/* reserve boot block area */
/* This memory range is used by the bootloader, so the application must not */
/* use this area. */
/* ----- */
#org 0, 0x7FF { }

/* --- END: changes required for bootloader ----- */

```

B. flex_lcd.h

```

#define LCD_DB4 PIN_A0
#define LCD_DB5 PIN_A1
#define LCD_DB6 PIN_A2
#define LCD_DB7 PIN_A3

#define LCD_E PIN_B3
#define LCD_RS PIN_B2
#define LCD_RW PIN_B1
#define lcd_type 2 // 0=5x7, 1=5x10, 2=2 lines
#define lcd_line_one 0x00 // LCD RAM address for the 1st line
#define lcd_line_two 0x40 // LCD RAM address for the 2nd line
#define lcd_line_three 0x14 // LCD RAM address for the 3rd line
#define lcd_line_four 0x54 // LCD RAM address for the 4th line
//Προσοχή για theHitachi HD66712U οι LCD RAM address είναι
// διαφορετικές

```

```

int8const LCD_INIT_STRING[4] =
{
    0x20 | (lcd_type<< 2),          // Func set: 4-bit, 2 lines, 5x8 dots
    0xc,                            // Display on
    1,                              // Clear display
    6                               // Increment cursor
};
//=====
voidlcd_send_nibble(int8 nibble);
voidlcd_send_byte(int8 address, int8 n);
voidlcd_init(void);
voidlcd_gotoxy(int8 x, int8 y);
voidlcd_putc(char c);
//=====
//-----
voidlcd_send_nibble(int8 nibble)
{
    // Note: !!converts an integer expression
    // to a boolean (1 or 0).
    output_bit(LCD_DB4, !(nibble& 1));
    output_bit(LCD_DB5, !(nibble& 2));
    output_bit(LCD_DB6, !(nibble& 4));
    output_bit(LCD_DB7, !(nibble& 8));

    delay_cycles(20);
    output_high(LCD_E);
    delay_us(50);
    output_low(LCD_E);
}

//-----
// This sub-routine is only called by lcd_read_byte().
// It's not a stand-alone routine. For example, the
// R/W signal is set high by lcd_read_byte() before
// this routine is called.

//-----
// Send a byte to the LCD.
voidlcd_send_byte(int8 address, int8 n)
{
    output_low(LCD_RS);

    if(address)
        output_high(LCD_RS);
    else
        output_low(LCD_RS);
}

```

```

delay_cycles(10);

output_low(LCD_E);

lcd_send_nibble(n >> 4);
lcd_send_nibble(n & 0xf);
}

//-----
void lcd_init(void)
{
int8 i;

output_low(LCD_RS);

output_low(LCD_E);
output_low(LCD_RW);
delay_ms(200);

for(i=0 ;i < 3; i++)
{
lcd_send_nibble(0x03);
delay_ms(10);
}

lcd_send_nibble(0x02);

for(i=0; i < sizeof(LCD_INIT_STRING); i++)
{
lcd_send_byte(0, LCD_INIT_STRING[i]);

// If the R/W signal is not used, then
// the busy bit can't be polled. One of
// the init commands takes longer than
// the hard-coded delay of 60 us, so in
// that case, lets just do a 5 ms delay
// after all four of them.

delay_ms(10);
}

}

//-----

```

```

voidlcd_gotoxy(int8 x, int8 y)
{
int8 address;

if(y == 2)
    {address = lcd_line_two;}
else if(y==3)
    {address = lcd_line_three;}
else if(y==4)
    {address = lcd_line_four;}
else
address=0;

address += x-1;
lcd_send_byte(0, 0x80 | address);
}

//-----
voidlcd_putc(char c)
{
switch(c)
    {
case 'f':
lcd_send_byte(0,1);
delay_ms(4);
break;

case '\n':
lcd_gotoxy(1,2);
break;

case '\b':
lcd_send_byte(0,0x10);
break;

default:
lcd_send_byte(1,c);
break;
    }
}

//-----

```

Г. keypad.h

```
//Keypad connection:
#define row0 PIN_D4
#define row1 PIN_D5
#define row2 PIN_D6
#define row3 PIN_D7
#define col0 PIN_D0
#define col1 PIN_D1
#define col2 PIN_D2
#define col3 PIN_D3

//----- Variable definition-----
charconst KEYS[4][4] =
{{'1','2','3','A'},
 {'4','5','6','B'},
 {'7','8','9','C'},
 {'.','0','=','D'}};

#define KBD_DEBOUNCE_FACTOR 33 // Set this number to apx n/333 where
// n is the number of times you expect
// to call kbd_getc each second

//-----Function Definition-----
shortint ALL_ROWS (void);
voidkbd_init(void);
charkbd_getc(void);
//-----
voidkbd_init(void)
{
set_tris_b(0xE0);

}
//-----
shortint ALL_ROWS (void)
{
if(input (row0) & input (row1) & input (row2) & input (row3))
return (0);
else
return (1);
}

//-----

charkbd_getc(void)
```

```

{
static byte kbd_call_count;
static short intkbd_down;
static char last_key;
static byte col;

bytekchar;
byte row;

kchar='\0';

if(++kbd_call_count>KBD_DEBOUNCE_FACTOR)
{
switch (col)
{
case 0:
output_low(col0);
output_high(col1);
output_high(col2);
output_high(col3);
break;

case 1:
output_high(col0);
output_low(col1);
output_high(col2);
output_high(col3);
break;

case 2:
output_high(col0);
output_high(col1);
output_low(col2);
output_high(col3);
break;

case 3:
output_high(col0);
output_high(col1);
output_high(col2);
output_low(col3);
break;
}

if(kbd_down)
{

```

```

if(!ALL_ROWS())
    {
kbd_down=false;
kchar=last_key;
last_key='\0';
    }
}
else
    {
if(ALL_ROWS())
    {
if(!input (row0))
row=0;
else if(!input (row1))
row=1;
else if(!input (row2))
row=2;
else if(!input (row3))
row=3;

last_key =KEYS[row][col];
kbd_down = true;
    }
else
    {
++col;
if(col==4)
col=0;
    }
}
kbd_call_count=0;
}
return(kchar);
}

```