

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Κατασκευή Διάταξης Μέτρησης Περιστροφής Άξονα»



Του φοιτητή  
*Μάρκου Παρδαλιού*  
Αρ. Μητρώου: 500585

Επιβλέπων  
*Καζακόπουλος Αριστοτέλης*  
Βαθμίδα: Καθηγητής

Τίτλος Δ.Ε.: Κατασκευή Διάταξης Μέτρησης Περιστροφής Άξονα

Κωδικός Π.Ε. 20106

Όνοματεπώνυμο φοιτητή: Μάρκος Παρδαλιός

Όνοματεπώνυμο εισηγητή: Καζακόπουλος Αριστοτέλης

Ημερομηνία ανάληψης 2/2021

Ημερομηνία περάτωσης Π.Ε. Σεπτέμβριος 2022

*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 κειμένου, 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 του φοιτητή Μάρκου Παρδαλιού που την εκπόνησε. 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 πτυχιακής 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.

## Ευχαριστίες

Πρωτίστως, θα ήθελα να ευχαριστήσω την σύζυγό μου, Βασιλική καθώς και την οικογένειά μου που στήριξαν αυτή την προσπάθεια. Επίσης, θα ήθελα να ευχαριστήσω τον καθηγητή μου, κο Καζακόπουλο για την ευκαιρία και την ουσιαστική καθοδήγηση που μου πρόσφερε, προκειμένου να ολοκληρωθεί αυτή η εργασία.



## Πρόλογος

Οι λόγοι, οι οποίοι με έπεισαν να αναλάβω την συγκεκριμένη πτυχιακή εργασία, οφείλονται κατά κύριο λόγο στα ενσωματωμένα συστήματα, και την επίδραση που έχουν στην καθημερινή ζωή του ανθρώπου με ποικίλους τρόπους, και κατά δεύτερον την εις βάθος ανάλυση λειτουργίας ενός ψηφιακού στροφόμετρου, καθώς και την δυνατότητα ακρίβειας μετρήσεων, με το κόστος των υλικών να είναι κατά πολύ μειωμένο σε σχέση με τα εμπορικά στροφόμετρα που κυκλοφορούν στην αγορά.

Στόχος της πτυχιακής είναι η απόδειξη πως με γνώσεις προγραμματισμού, ηλεκτρονικών και τον πολυμερισμό αυτών, ο γνώστης αυτών μπορεί να αναπτύξει και να σχεδιάσει διατάξεις ικανές να ανταγωνιστούν αυτές που είναι διαθέσιμες στην αγορά, και να τις ξεπεράσει σε ακρίβεια, με χαμηλό κόστος.

## Περίληψη

Σκοπός της πτυχιακής εργασίας είναι η μελέτη, σχεδίαση και κατασκευή μέτρησης περιστροφής άξονα. Στο πρώτο κεφάλαιο, περιγράφεται η σειρά PIC18F μικροελεγκτών, και συγκεκριμένα του PIC18F4550, καθώς είναι και ο μικροελεγκτής που επιλέχθηκε για την υλοποίηση της διάταξης. Στο δεύτερο κεφάλαιο, θα μελετηθεί εις βάθος η αρχή λειτουργίας του στροφόμετρου, για την κατανόηση του προβλήματος. Στο τρίτο κεφάλαιο, θα υπάρξει αναφορά στην μοντελοποίηση και λύση του προβλήματος, μέσω των χαρακτηριστικών που προσφέρει ο μικροελεγκτής PIC18F4550, μαζί με το διάγραμμα ροής, την προσομοίωση της διάταξης μέσω του σχεδιαστικού πακέτου Proteus, τον κώδικα που γράφτηκε σε γλώσσα C, με την βοήθεια του MPLab, καθώς και το σχηματικό και η σχεδίαση της πλακέτας με την βοήθεια του σχεδιαστικού πακέτου Eagle. Στο τέταρτο και τελευταίο κεφάλαιο, αναφέρονται τα συμπεράσματα, και μελλοντικές προτάσεις για την βελτιστοποίηση του ψηφιακού στροφόμετρου.

# «Construction of Shaft Rotation Measurement Device»

«Markos Pardalios»

## Abstract

The purpose of this thesis is the study, design and construction of a digital stroboscope. The first chapter describes the fundamentals of PIC18F microcontrollers, specifically the PIC18F4550, because it's the microcontroller selected for the implementation of this design. The second chapter contains the principals of stroboscopes, in order to understand the problem that needs a solution. The third chapter there is going to be a report regarding the modeling of the problem, throughout the features the PIC18F4550 has to offer, together with the flowchart, the schematic simulation with Proteus design package, the code written in C language with the help of MPLab, as well as the schematic layout with the help of Eagle design package. The fourth and last chapter, contains the conclusions and future proposals for the optimization of the digital stroboscope, in ways of packaging, and pcb layout.

# Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract .....	vii
Κατάλογος Σχημάτων .....	xi
Κατάλογος Εικόνων .....	xii
Συντομογραφίες.....	xiii
1. Μικροελεγκτές PIC .....	1
1.1 Εισαγωγή.....	1
1.2 Πλεονεκτήματα των μικροελεγκτών PIC.....	1
1.3 Οι οικογένειες των μικροελεγκτών PIC .....	2
1.3.1 Η οικογένεια PIC12CXXX/PIC12FXXX .....	2
1.3.2 Η οικογένεια PIC16C5X .....	2
1.3.3 Η οικογένεια PIC16CXXX/PIC16FXXX .....	2
1.3.4 Η οικογένεια PIC17CXXX .....	3
1.3.5 Η οικογένεια PIC18CXXXX/PIC18FXXXX.....	3
1.4 Εσωτερική λειτουργία PIC.....	3
1.5 Οι θύρες εισόδου και εξόδου(I/O).....	3
1.6 CPU .....	3
1.7 Clock .....	4
1.8 Memory.....	4
1.9 IO Ports .....	4
1.10 Λογισμικό (Software).....	5
1.11 Programmer.....	6
1.12 Υλοποίηση εφαρμογής PIC.....	8
1.12.1 Διατύπωση.....	9
1.12.2 Επιλογή κατάλληλου PIC.....	9
1.12.3 Ανάπτυξη πρωτότυπου .....	9
1.12.4 Αποσφαλματοποίηση (Debugging).....	9
1.12 Τελική κατασκευή.....	10
1.13 ΠΕΡΙΦΕΡΕΙΑΚΑ PIC18F.....	10
1.13.1 Τάση Τροφοδοσίας.....	10
1.13.2 Ρολόι.....	10
1.13.3 Timers.....	11

1.13.4	Watchdog Timer.....	11
1.13.5	Κουμπί επανεκκίνησης.....	11
1.13.6	Διακοπές.....	12
1.13.7	Brown Out Detector .....	12
1.13.8	ADC module.....	12
1.13.9	UART module .....	12
1.13.10	EEPROM.....	13
1.13.11	LCD Driver.....	13
1.13.12	Συγκριτής.....	13
1.13.13	Λειτουργία Ύπνου.....	13
2.	Διαδικασίες ανάπτυξης λογισμικού.....	14
2.1	Μοντέλο καταρράκτη.....	14
2.2	Μοντέλο Προτυποποίησης.....	16
2.3	Μοντέλο λειτουργικής επαύξεσης.....	17
2.4	Το σπειροειδές μοντέλο.....	18
3.	Σχεδιασμός και κατασκευή πλακέτας (PCB) .....	21
3.1	Εισαγωγή στις πλακέτες PCB .....	21
3.2	Ταξινόμηση των PCBs .....	22
3.3	Τυπωμένη πλακέτα μονής ή απλής όψης(Single-sided PCB) .....	22
3.4	Τυπωμένη πλακέτα διπλής όψης(double-sided PCB) .....	22
3.5	Τυπωμένη πλακέτα πολλαπλών στρωμάτων( multi-layer PCB).....	22
3.6	Παραλλαγές σχεδίασης PCB.....	22
3.7	Ο Σχεδιασμός, η τύπωση και η τελική μορφή του PCB.....	22
3.8	Το σχεδιαστικό πακέτο EAGLE.....	23
3.8.1	Το σχηματικό διάγραμμα της πλακέτας της εφαρμογής .....	23
3.8.2	MCLR.....	24
3.8.3	Πυκνωτής αποθρομβοποίησης .....	25
3.8.4	Speed Sensor Connector.....	25
3.8.5	ICSP Διάταξη .....	26
3.8.6	Σύνδεση Τροφοδοσίας.....	26
3.8.7	Οθόνη LCD 16x2 .....	27
3.8.8	Ποτενσιόμετρο για την οθόνη .....	27
3.8.9	Μικροελεγκτής PIC18F4550.....	28
3.9	PCB Layout.....	28
4.	Μελέτη Στροφόμετρου.....	30

4.1	Εισαγωγή.....	30
4.2	Αρχικά Βήματα .....	31
4.3	Μοντέρνες Τεχνολογίες.....	31
4.4	Γεννήτρια .....	31
4.5	Το φαινόμενο Hall.....	34
4.6	Οπτικός Κωδικοποιητής (Optical Encoder).....	35
4.7	Laser και φωτοδίοδος.....	37
4.8	Laser .....	38
4.9	Φωτοδίοδος .....	40
5.	Στήσιμο εφαρμογής.....	42
5.1	Εισαγωγή.....	42
5.2	Υλοποίηση στην C .....	42
5.3	Ο προγραμματισμός του μικροελεγκτή.....	43
5.4	Το περιβάλλον προγραμματισμού MPLAB .....	43
5.5	Capture Mode.....	43
5.6	LCD Module.....	46
5.7	Αισθητήρας Απόστασης.....	46
5.8	Προσομοίωση με το Proteus .....	47
5.9	Διάγραμμα Ροής.....	48
5.10	Προσομοίωση στο ράστερ.....	49
5.11	Σχεδίαση πλακέτας.....	50
5.12	Χαρακτηριστικά Ανεμιστήρα για την εφαρμογή .....	52
5.13	Τύπωση Πλακέτας και τεστ σε πραγματικές συνθήκες.....	54
6	Επίλογος και Συμπεράσματα.....	58
6.1	Επίλογος.....	58
6.2	Συμπεράσματα ή/και προτάσεις βελτίωσης .....	59
ΠΑΡΑΡΤΗΜΑ .....		61
Κώδικας προγράμματος σε γλώσσα C .....		61
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		66

## Κατάλογος Σχημάτων

Σχήμα 1.1 Σχηματικό διάγραμμα μικροελεγκτή PIC.....	17
Σχήμα 1.2 Διαδικασία προγραμματισμού μικροελεγκτή PIC.....	20
Σχήμα 1.3 Εσωτερικό διάγραμμα ταλαντωτή.....	24
Σχήμα 2.1 Μοντέλο Καταρράκτη.....	28
Σχήμα 2.2 Μοντέλο προτυποποίησης.....	30
Σχήμα 2.3 Μοντέλο λειτουργικής επαύξησης.....	31
Σχήμα 2.4 Σπειροειδές Μοντέλο.....	32
Σχήμα 3.1 Το σχηματικό της εφαρμογής.....	36
Σχήμα 3.2 Η διάταξη MCLR.....	37
Σχήμα 3.3: Πυκνωτής αποθρομβοποίησης.....	38
Σχήμα 3.4 Διάταξη αισθητήρα ταχύτητας.....	38
Σχήμα 3.5 ICSP διάταξη προγραμματισμού του μικροελεγκτή.....	39
Σχήμα 3.6 Διάταξη Τροφοδοσίας USB.....	39
Σχήμα 3.7 Διάταξη οθόνης LCD 16x2.....	40
Σχήμα 3.8 Ποτενσιόμετρο 10κ.....	40
Σχήμα 3.9 PIC18F4550.....	41
Σχήμα 4.1 Διάγραμμα τάσης- RPM.....	45
Σχήμα 4.2 Αντιπαραβολή EMF και μαγνητική ροή με την πάροδο του χρόνου.....	47
Σχήμα 4.3 Ακολουθία του τροχού του οπτικού κωδικοποιητή.....	49
Σχήμα 4.4 Αλληλουχία δεδομένων.....	50
Σχήμα 4.5 Αρχή λειτουργίας λέιζερ και ανακλαστικής ταινίας.....	51
Σχήμα 4.6 Απόκριση φωτοδιόδου στο φάσμα του μήκους κύματος.....	53
Σχήμα 5.1 Διάγραμμα ακίδων του μικροελεγκτή PIC18F4550.....	57
Σχήμα 5.2 Κύκλωμα αισθητήρα απόστασης.....	60

## Κατάλογος Εικόνων

Εικόνα 1.1 Pickit 3.....	21
Εικόνα 1.2 Διάγραμμα διασύνδεσης του Pickit 3.....	21
Εικόνα 3.1 Η δομή μιας PCB πλακέτας.....	34
Εικόνα 3.2 PCB Layout.....	42
Εικόνα 4.1: Το πρώτο στροφόμετρο.....	43
Εικόνα 4.2 Στροφόμετρο επαφής.....	46
Εικόνα 4.3 Δύναμη που ασκείται στην κίνηση Εικόνα.....	48
Εικόνα 4.4 Ο νόμος του δεξιού χεριού.....	48
Εικόνα 4.5 Η τάση που προκάλεσε το φαινόμενο του Hall.....	48
Εικόνα 4.6 Optical Encoder.....	49
Εικόνα 4.7 Περιγραφή διέγερσης αερίου laser.....	52
Εικόνα 4.8 Δίοδος LED εκπέμπει φως.....	53
Εικόνα 4.9 Δίοδος PIN απορροφάει φως.....	53
Εικόνα 4.10 Κοινό στροφόμετρο του εμπορίου.....	54
Εικόνα 5.1 Ο καταχωρητής CCP1CON.....	58
Εικόνα 5.2 Καταχωρητής ελέγχου T1CON.....	58
Εικόνα 5.3 LCD 16x2 Module.....	59
Εικόνα 5.4 Αισθητήρας Απόστασης.....	60
Εικόνα 5.5 Προσομοίωση στο Proteus.....	61
Εικόνα 5.6 Διάγραμμα ροής.....	62
Εικόνα 5.7 Μέτρηση στροφών σε ράστερ.....	63
Εικόνα 5.8 Πλακέτα ψηφιακού στροφόμετρου.....	64
Εικόνα 5.9 Σχηματικό ψηφιακού στροφόμετρου.....	65
Εικόνα 5.10 Προδιαγραφές Ανεμιστήρα.....	66
Εικόνα 5.11 Τυπωμένο κύκλωμα pcb.....	68
Εικόνα 5.12 Ολοκληρωμένη πλακέτα.....	69
Εικόνα 5.13 Ψηφιακό Στροφόμετρο.....	70

## Συντομογραφίες

ΔΠΠΑΕ.....	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.....	Πτυχιακή Εργασία
PIC.....	Peripheral Interface Controller
USB.....	Universal Serial Bus
RISC.....	Reduced Instruction Set Code
RAM.....	Random Access Memory
EEPROM.....	Electrically Erasable Programmable Read-only Memory
ROM.....	Read-only Memory
OTP.....	One-Time Programmable
MIPS.....	Million Instructions Per Second
PLL.....	Phase-locked Loop
ALU.....	Arithmetic Logic Unit
ICSP.....	In Circuit Serial Programming
ADC.....	Analog to Digital Converter
UART.....	Universal Asynchronous Receiver-transmitter
PCB.....	Printed Circuit Board
PWB.....	Printed Wiring Board
RPM.....	Revolutions per Minute
LED.....	Light Emitting Diode
LCD.....	Liquid-Crystal Display



## 1. Μικροελεγκτές PIC

### 1.1 Εισαγωγή

Όλοι οι μικροελεγκτές PIC είναι αρχιτεκτονικής τύπου RISC, με ξεχωριστή μνήμη εντολών και δεδομένων και ξεχωριστούς διαύλους (αρχιτεκτονική Harvard) προσπέλασης των μνημών αυτών. Όλοι οι μικροελεγκτές PIC διαθέτουν ενσωματωμένες (1) μνήμη δεδομένων RAM, (2) μνήμη EEPROM, (3) μνήμη ROM και (4) πόρτες I/O. Η Microchip διαθέτει μεγάλη ποικιλία μικροελεγκτών (περισσότερους από 200) που περιέχουν εκτός από τις προαναφερόμενες μνήμες και πόρτες I/O πολλά ακόμη περιφερειακά.[1]

Πολλοί από τους PIC διαθέτουν ενσωματωμένα περισσότερα περιφερειακά και έχουν περισσότερες δυνατότητες συγκρινόμενοι με άλλους ανταγωνιστικούς μικροελεγκτές, πράγμα που οδηγεί στην μείωση κόστους της τελικής εφαρμογής. Τέτοια περιφερειακά είναι π.χ. ο ενσωματωμένος κρύσταλλος 4MHz, το ενσωματωμένο κύκλωμα reset, η δυνατότητα οδήγησης εξωτερικών συσκευών από τις πόρτες με ρεύμα τάξεως των 20mA, ο προγραμματισμός και σβήσιμο επί τόπου ICSP της ενσωματωμένης μνήμης Flash και πολλά άλλα.

Οι μικροελεγκτές PIC υποστηρίζονται από την εταιρεία κατασκευής τους με πολύ πλούσια βιβλιογραφία και τεκμηρίωση αποτελούμενη από πολλά παραδείγματα εφαρμογής. Σύμφωνα με την Microchip, οι πωλήσεις των μικροελεγκτών PIC ξεπερνούν τις πωλήσεις όλων των άλλων κατασκευαστών.

Ορισμένα από τα λειτουργικά χαρακτηριστικά που θα πρέπει να ληφθούν υπόψη κατά την επιλογή του μικροελεγκτή είναι:

- Τεχνολογία μνήμης (Flash, EEPROM, κτλ)
- Τάση τροφοδοσίας
- Περιοχή θερμοκρασίας λειτουργίας
- Συχνότητα λειτουργίας
- Συσκευασία

### 1.2 Πλεονεκτήματα των μικροελεγκτών PIC

Η οικογένεια μικροελεγκτών PIC, φέρει κάποια χαρακτηριστικά που την επιλέγουν πολλοί μηχανικοί έναντι άλλων οικογενειών μικροελεγκτών. Μερικά από αυτά τα χαρακτηριστικά που απαρτίζουν τους μικροελεγκτές PIC είναι:

- **Χαμηλό Κόστος:** Είναι από τα βασικότερα χαρακτηριστικά που επιλέγει κάποιος σχεδιαστής. Η συνεχής παραγωγή μικροελεγκτών στην αγορά από διάφορες εταιρείες, ύψωσε την ποιότητα στα ύψη και το κόστος έπεσε λόγω του ανταγωνισμού
- **Μικρό μέγεθος:** Η ολοκλήρωση των βασικών στοιχείων από τα οποία απαρτίζεται μείωσε τις διαστάσεις σε σχέση με τη χρήση των επιμέρους στοιχείων του ως σύνολο
- **Χαμηλή κατανάλωση ισχύος:** Το γεγονός ότι οι μικροελεγκτές λειτουργούν σε συγκριτικά χαμηλές συχνότητες που φτάνουν τα 32KHz, οδηγεί στην κατανάλωση μικρών ποσών ισχύος της τάξης των mW ακόμα και  $\mu$ W. Επιπλέον έχουν την δυνατότητα να εισέρχονται σε κατάσταση idle-sleep mode, καταστέλλοντας προσωρινά την λειτουργία της κεντρικής μονάδας επεξεργασίας και των περιφερειακών. Έτσι μπορούν να χρησιμοποιηθούν σε εφαρμογές με αυστηρές απαιτήσεις ως προς την κατανάλωση.

- **Αυτονομία:** Αυτό επιτυγχάνεται μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν
- **Επίτευξη ελέγχου ή μετρήσεων σε πραγματικό χρόνο:** Ενώ οι ηλεκτρονικοί υπολογιστές πρέπει να τρέχουν λειτουργικά συστήματα πραγματικού χρόνου για να το επιτύχουν, οι μικροελεγκτές δεν απαιτούν επιπλέον λογισμικό
- **Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές:** Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και των χαμηλότερων ταχυτήτων λειτουργίας
- **Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους:** Λόγω της μη δέσμευσης τους για τη σύνδεση εξωτερικών περιφερειακών
- **Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών:** Παρόλο που στους πρώτους συναντάται συχνά η αρχιτεκτονική τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων (π.χ. οι σειρές από την Microchip), στους κοινούς μικροεπεξεργαστές συνήθίζεται η ενιαία διάταξη Von Neumann.

### 1.3 Οι οικογένειες των μικροελεγκτών PIC

Οι μικροελεγκτές μπορούν να διακριθούν σε 5 βασικές κατηγορίες

- PIC12CXXX/PIC12FXXX
- PIC16C5X με 12-ψήφιο μήκος εντολής
- PIC16CXXX/PIC16FXXX με 14-ψήφιο μήκος εντολής
- PIC17CXXX με 16-ψήφιο μήκος εντολής
- PIC18CXXX/PIC18FXXX με βελτιωμένο 16-ψήφιο μήκος εντολής

#### 1.3.1 Η οικογένεια PIC12CXXX/PIC12FXXX

Η οικογένεια των μικροελεγκτών PIC12CXXX/PIC12FXXX ιδιαίτερα χαμηλού κόστους. Η οικογένεια αυτή περιλαμβάνει τους μικροελεγκτές των 8 ακίδων. Το ρεπερτόριο των RISC εντολών της οικογένειας περιλαμβάνει 35 εντολές οι οποίες αποτελούνται από 12 ή 14 ψηφία. Τα ολοκληρωμένα μπορούν να λειτουργούν ακόμη και με τάση 2,5v. Κατασκευάζονται με μνήμες προγράμματος FLASH, OTP ή ROM. Περιέχουν πολλαπλά κανάλια A/D καθώς και πλήθος ακίδων διακοπών. Κατασκευάζονται με μνήμες δεδομένων RAM 64ων bytes και EEPROM των 128 bytes.

#### 1.3.2 Η οικογένεια PIC16C5X

Οι εντολές της οικογένειας των μικροελεγκτών 16C5X έχουν μήκος 12 ψηφίων. Διατίθενται στην αγορά σε συσκευασίες SOIC και SSOP των 14, 18, 20 και 28 ακίδων. Μπορούν να λειτουργήσουν ακόμη και με τάση 2V, όταν διαθέτουν μνήμη προγραμματιζόμενη μια φορά (OTP). Επιπρόσθετα, ο PIC16HV5XX μπορεί να λειτουργεί με τάση τροφοδοσίας μέχρι 15V για την σύνδεση του απευθείας με μπαταρία.

#### 1.3.3 Η οικογένεια PIC16CXXX/PIC16FXXX

Οι μικροελεγκτές αυτοί διαθέτουν 16-ψήφιο μήκος εντολής, βελτιωμένο ρεπερτόριο εντολών και βελτιωμένες ικανότητες διαχείρισης διακοπών.

### 1.3.4 Η οικογένεια PIC17CXXX

Οι μικροελεγκτές αυτοί διαθέτουν 16-ψήφιο μήκος εντολής, βελτιωμένο ρεπερτόριο εντολών και βελτιωμένες ικανότητες διαχείρισης διακοπών

### 1.3.5 Η οικογένεια PIC18CXXXX/PIC18FXXXX

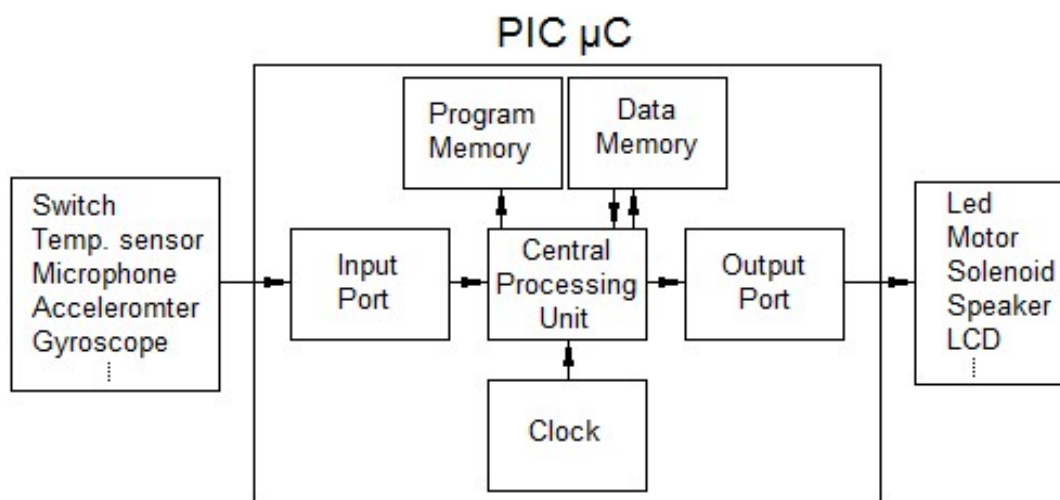
Οι μικροελεγκτές PIC18CXXXX/PIC18FXXXX αποτελούν μια οικογένεια υψηλής απόδοσης CMOS με ενσωματωμένο μετατροπέα A/D και 16-ψήφιο μήκος εντολής. Οι μικροελεγκτές PIC18CXXXX/PIC18FXXXX διαθέτουν 32 επιπέδων σωρό (σε αντίθεση με τους άλλους μικρότερους, που διαθέτουν 8 επιπέδων), καθώς και πολλαπλές πηγές πρόκλησης εσωτερικών και εξωτερικών διακοπών. Διαθέτουν συνολικά 77 εντολές. Λόγω των επιπρόσθετων καταχωρητών που διαθέτουν φθάνουν την ταχύτητα εκτέλεσης εντολών των 10 MIPS.

## 1.4 Εσωτερική λειτουργία PIC

Οι μικροελεγκτές PIC είναι ψηφιακά IC's που χρησιμοποιούνται για τον έλεγχο περιφερειακών ηλεκτρονικών διατάξεων. Παράγονται από την αμερικάνικη εταιρεία Microchip και απαντώνται σε πολλές σύνθετες αλλά και απλές εφαρμογές όπως έλεγχο robot, βιομηχανικές διατάξεις, συστήματα ECU και ESP αυτοκινήτων, οικιακές συσκευές. Τα κύρια τμήματα τους είναι (σχήμα 1.1):

- Η κεντρική μονάδα επεξεργασίας CPU
- Το ρολόι (Clock)
- Η μνήμη (Memory)

### 1.5 Οι θύρες εισόδου και εξόδου(I/O)



Σχήμα 1.1 Σχηματικό διάγραμμα μικροελεγκτή PIC

### 1.6 CPU

Η λειτουργία του PIC καθορίζεται από το πρόγραμμα, που είναι μία σειρά εντολών, που εγγράφεται στη μνήμη του κατά τη διαδικασία του προγραμματισμού του. Η εκτέλεση των εντολών είναι σειριακή

και λαμβάνει χώρα στην CPU. Η CPU περιέχει τη μονάδα αριθμητικών και λογικών πράξεων ALU καθώς και μονάδες επικοινωνίας και ελέγχου.

Μία πολύ σημαντική λειτουργία που υποστηρίζει η CPU είναι οι διακοπές κατά τις οποίες διακόπτεται η κανονική ροή του προγράμματος και εκτελείται μια συγκεκριμένη υπορουτίνα. Μετά το πέρας της υπορουτίνας συνεχίζει η εκτέλεση του προγράμματος από το σημείο που είχε σταματήσει. Η ενεργοποίηση των διακοπών μπορεί να γίνει είτε εσωτερικά είτε εξωτερικά. Εξωτερικά γίνεται μεταβάλλοντας τη λογική κατάσταση ενός ακροδέκτη ή εφαρμόζοντας σε αυτόν δεδομένη αλληλουχία ψηφιακών παλμών. Εσωτερικά γίνεται με την χρήση των χρονιστών (Timers).

Οι Timers είναι διατάξεις της CPU και χρησιμοποιούνται για την ακριβή μέτρηση χρόνου. Ο κάθε Timer αντιπροσωπεύεται από μια μεταβλητή TMR που αυξάνει κατά ένα κάθε φορά που δέχεται ένα συγκεκριμένο αριθμό παλμών, ο οποίος αριθμός ονομάζεται Prescaler. Όταν η TMR φτάσει την τιμή 256 μηδενίζεται και ο Timer ενεργοποιεί το αντίστοιχο Interrupt. Η συχνότητα των διακοπών δίδεται από τις παρακάτω σχέσεις, ανάλογα με τον τρόπο χρονισμού των Timers.

- Από το ρολόι της CPU:  $F_{cpu} = \frac{F_{clk}}{4}, F_{int} = F_{cpu}(\text{Prescaler} * 256)^{-1}$
- εξωτερικοί παλμοί σε ακροδέκτη:  $F_{int} = F_p(\text{Prescaler} * 256)^{-1}$

## 1.7 Clock

Οι λειτουργίες της CPU συγχρονίζονται από το ρολόι (Clock). Το ρολόι στέλνει μία συνεχόμενη σειρά παλμών στη CPU ελέγχοντας έτσι την εκτέλεση των εντολών του προγράμματος και τη μετακίνηση των δεδομένων μέσα στο σύστημα. Η εκτέλεση μιας εντολής από την ALU χρειάζεται τέσσερις παλμούς CK σύμφωνα με το σχήμα 1.1 και επομένως όσο μεγαλύτερη είναι η συχνότητα του ρολογιού τόσο μικρότερος είναι ο χρόνος εκτέλεσης του προγράμματος από τον μικροελεγκτή.

Το ρολόι μπορεί να είναι εσωτερικό ή εξωτερικό. Στο εσωτερικό η συχνότητά του καθορίζεται από το πρόγραμμα. Το εξωτερικό ρολόι αποτελείται από κρύσταλλο RC ή άλλο κύκλωμα χρονισμού συνδεδεμένο στους αντίστοιχους ακροδέκτες του PIC.

## 1.8 Memory

Η μνήμη του PIC χρησιμοποιείται για την αποθήκευση του προγράμματος και για αποθήκευση δεδομένων. Αποτελείται από τρεις διακριτές μονάδες καταχώρησης. Η πρώτη είναι η μνήμη που αποθηκεύεται το πρόγραμμα Program Memory (EPROM). Η δεύτερη είναι η μνήμη προσωρινής αποθήκευσης δεδομένων RAM και η τρίτη είναι η μνήμη μόνιμης αποθήκευσης δεδομένων EEPROM.

Οι PIC που χαρακτηρίζονται ως Flash-Based, διαθέτουν Program Memory η οποία δύναται να σβηστεί και να προγραμματιστεί εκ νέου χωρίς κάποια ειδική διαδικασία πέραν αυτής που θα παρουσιαστεί παρακάτω.

## 1.9 IO Ports

Η επικοινωνία του PIC με τα περιφερειακά του εξαρτήματα γίνεται μέσω των Ports που αποτελούνται από ένα συγκεκριμένο αριθμό ακροδεκτών, κάθε ένας από τους οποίους αντιστοιχεί σε ένα bit. Η λειτουργία των Ports είναι αμφίδρομη και για αυτό και στην αρχή κάθε προγράμματος προσδιορίζεται ποια θα χρησιμοποιηθεί ως είσοδος και ποια ως έξοδος.

Μέσω των Ports ο PIC μπορεί πρώτον να διαχειριστεί απλά ψηφιακά σήματα ON/OFF, όπως η ανάγνωση της λογικής κατάστασης ενός διακόπτη ή της εξόδου ενός φωτοκύτταρου (Input), αλλά και να ελέγξει τη λειτουργία ή μη μίας συσκευής όπως πχ ενός σωληνοειδούς (Output).

Επίσης, ανάλογα με τα ποια εξειδικευμένα κυκλώματα διαθέτει, μπορεί να διαχειριστεί σύνθετα ψηφιακά σήματα για να επικοινωνήσει με άλλα ηλεκτρονικά κυκλώματα όπως μια οθόνη LCD ή ένα card reader μέσω σειριακών πρωτοκόλλων όπως το SPI και το I2C. Με H/Y μέσω RS232 και USB. Ακόμη μπορεί να ελέγξει με PWM ένα κινητήρα και ένα σύστημα φωτισμού.

Ορισμένοι PIC είναι εφοδιασμένοι με A/D Converter και επομένως μπορούν να λάβουν και να επεξεργαστούν αναλογικά σήματα. Οι ακροδέκτες που δέχονται αναλογικά σήματα είναι συνήθως αρκετοί, ωστόσο ένας PIC σπάνια διαθέτει δύο A/D. Αυτό σημαίνει ότι η ψηφιοποίηση των αναλογικών σημάτων δεν είναι ταυτόχρονη. Ακόμα ένας PIC μπορεί να διαθέτει και αναλογικό συγκριτή, η έξοδος του οποίου χρησιμοποιείται συνήθως ως interrupt.

Τέλος, παρατίθεται σαν παράδειγμα η επεξήγηση των λειτουργιών του Pin 26 από τον PIC18F4550.

RC7/RX/DT/SDO→Register Port C bit 7/ Ασύγχρονο bit λήψης/ Σύγχρονο bit λήψης/ SPI έξοδος δεδομένων.

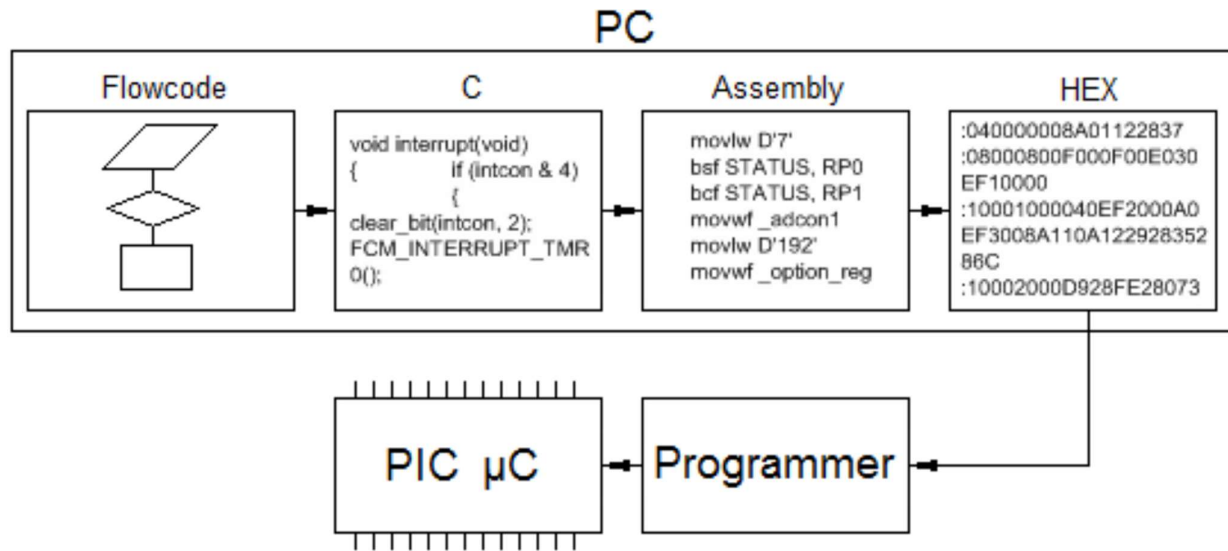
Όπως αναφέρθηκε οι PIC είναι ψηφιακά ολοκληρωμένα κυκλώματα προγραμματιζόμενης λειτουργίας. Η λειτουργία τους καθορίζεται από ένα πρόγραμμα το οποίο αποτελείται από μία καθορισμένη σειρά εντολών και αριθμητικά δεδομένα. Το πρόγραμμα σχ. 1.2 δημιουργείται σε H/Y μέσω μίας γλώσσας προγραμματισμού, μετατρέπεται σε 16αδική μορφή (Hex) και μεταφέρεται στον PIC μέσω μίας διάταξης που ονομάζεται Programmer. Εγγράφεται στην program memory και εκτελείται σειριακά.

### 1.10 Λογισμικό (Software)

Υπάρχουν δύο μέθοδοι δημιουργίας ενός προγράμματος. Η πρώτη είναι μέσω μίας χαμηλού επιπέδου γλώσσας (low-level language) που ονομάζεται Assembly. Είναι δύσκολη στη χρήση αλλά έχει πιο αποδοτική εκτέλεση των εντολών των αλγορίθμων της. Η δεύτερη είναι μέσω μιας υψηλού επιπέδου γλώσσας (high-level language) όπως η C. Είναι ευκολότερη στη χρήση της αλλά παράγει συνήθως μακροσκελείς και επομένως πιο αργούς κώδικες καθώς το πρόγραμμα μεταφράζεται πρώτα σε assembly και κατόπιν σε Hex.

Εκτός από τις παραπάνω γλώσσες προγραμματισμού, υπάρχουν λογισμικά όπως το Flow Code στο οποίο το πρόγραμμα δομείται σε μορφή λογικού διαγράμματος. Δεν διαχωρίζεται από τις high-level γλώσσες καθώς τα λογικά blocks (if,do,macro) είναι πακέτα οπτικοποίησης των αντίστοιχων εντολών της C και ο χρήστης απλά ορίζει τις παραμέτρους. Το επίπεδο δυσκολίας είναι ακόμη μικρότερο καθώς δίνεται η δυνατότητα στο χρήστη να κάνει προσημείωση της λειτουργίας του κώδικα στον PIC, αλλά η αποδοτικότητα του παραγόμενου κώδικα είναι αντίστοιχα μικρότερη.

Η ανάπτυξη της εφαρμογής που θα παρουσιαστεί στην συνέχεια έγινε με τη χρήση του προγράμματος MPLab. Επιλέχθηκε λόγω της εκμάθησης της στο μάθημα των Μικροελεγκτών, καθώς και των ετοιμών ρουτινών, που παρέχονται από την Microchip, για την ταχύτερη υλοποίηση της εφαρμογής, αλλά και την λιγότερο αποσφραλισμοποίηση της.



Σχήμα 1.2 Διαδικασία προγραμματισμού μικροελεγκτή PIC

### 1.11 Programmer

Ο Programmer όπως αναφέρθηκε, είναι η διάταξη μέσω της οποίας μεταφέρεται το πρόγραμμα από τον H/Y στον PIC. Στο εμπόριο διατίθενται μεγάλη ποικιλία τέτοιων διατάξεων με τα κύρια χαρακτηριστικά να είναι:

- Το πρωτόκολλο επικοινωνίας με τον H/Y, RS232 ή USB
- Η θέση προγραμματισμού του PIC
- Η δυνατότητα λειτουργίας του ως Debugger.

Το πρωτόκολλο επικοινωνίας καθορίζει τη θύρα σύνδεσης του Programmer στον H/Y, για το RS232 σύνδεση σε σειριακή θύρα COM ενώ για το USB στην ομώνυμη.

Η θέση προσδιορίζει που τοποθετείται ο PIC για να προγραμματιστεί. Παλαιότερα, οι προγραμματιστές απαιτούσαν την αφαίρεση του PIC από το κύκλωμα στο οποίο βρίσκεται και την τοποθέτηση του επάνω τους, γεγονός που δυσχεραίνει πολύ την διαδικασία ανάπτυξης μιας νέας εφαρμογής, όπου ο PIC προγραμματίζεται ξανά και ξανά, δίχως την αφαίρεση του μικροελεγκτή σε άλλη πλακέτα για την τοποθέτηση του κώδικα. Οι programmers, που υποστηρίζουν την λειτουργία ICSP προγραμματίζουν τον PIC χωρίς την ανάγκη μετακίνησης του, με την προϋπόθεση βέβαια ότι στο κύκλωμα του έχει διαμορφωθεί η απαιτούμενη καλωδίωση προγραμματισμού.

Η υποστήριξη λειτουργίας του Programmer και ως Debugger είναι πολύ σημαντική, καθώς δίνει την δυνατότητα εκτέλεσης του προγράμματος στον PIC step by step, για την ταυτόχρονη παρακολούθηση των μεταβλητών και της λειτουργίας του ελεγχόμενου περιφερειακού hardware.

Ο κάθε Programmer συνοδεύεται από ένα συγκεκριμένο πρόγραμμα H/Y, μέσω του οποίου επιλέγεται το HEX, που θα μεταφερθεί στον PIC και κατόπιν ελέγχεται η λειτουργία του (Read-Write-Verify program, Clean memory...)

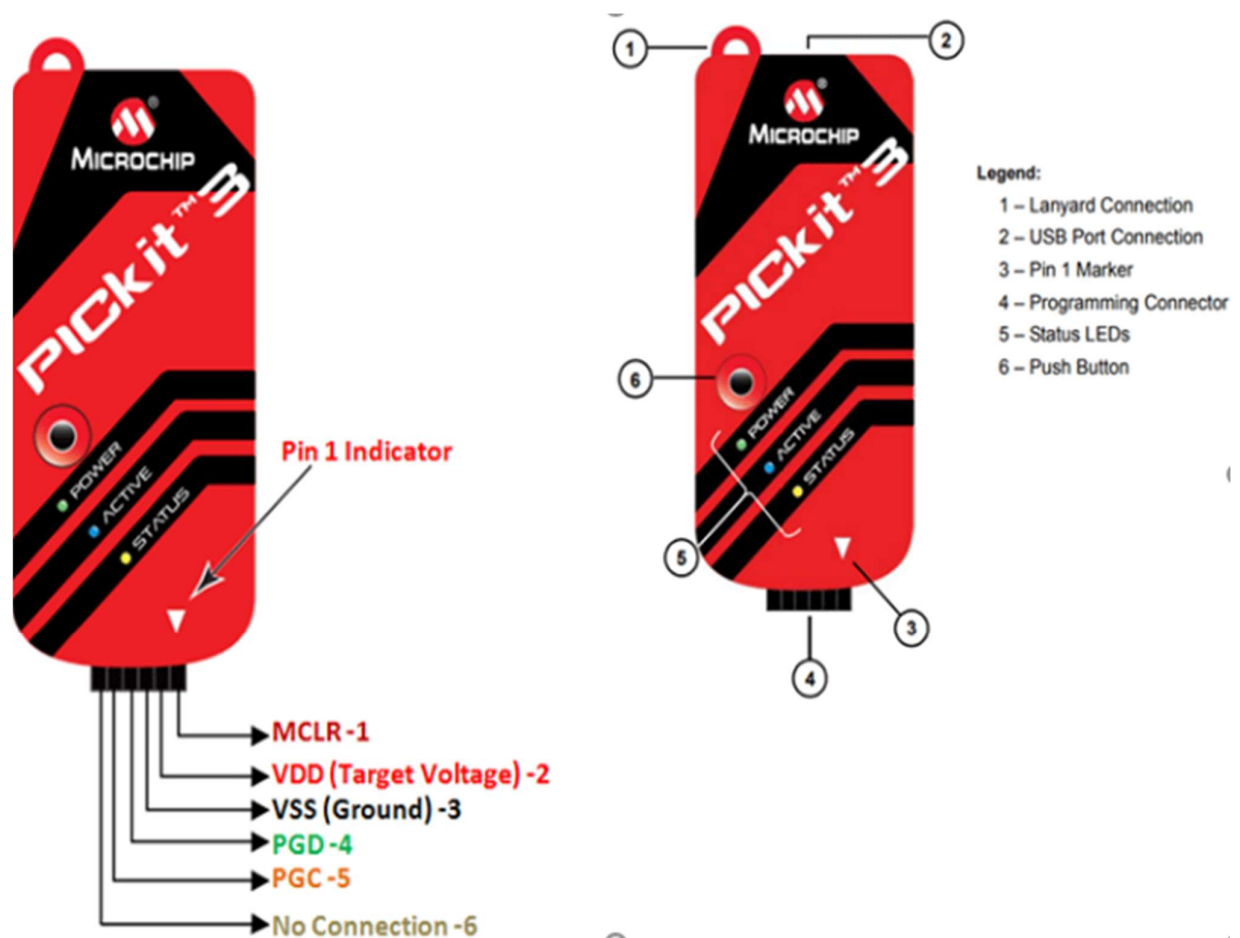
Για την ανάπτυξη της εφαρμογής, που θα παρουσιαστεί στην συνέχεια, επιλέχθηκε να χρησιμοποιηθεί ο Pickit 3, εργαλείο της Microchip, για τον προγραμματισμό αλλά και την αποσφαλματοποίηση (debugging) του κώδικα. Είναι συμβατός με το πρόγραμμα MPLab και μέσω του Pickit 3, αφού

## Μικροελεγκτές PIC

ολοκληρωθεί η συγγραφή του κώδικα, ο προγραμματιστής μπορεί να εκχωρήσει στον μικροελεγκτή το πρόγραμμα απευθείας μέσω της λειτουργίας ICSP του προγραμματιστή Pickit 3.



Εικόνα 1.1 *Pickit 3*



Εικόνα 1.2 Διάγραμμα διασύνδεσης του Pickit 3

Η φόρτωση του προγράμματος στον μικροελεγκτή μπορεί να γίνει και μέσω ενός προγράμματος που έχει δημιουργηθεί αποκλειστικά για το PICKit (Εικόνα 1.2). Η διαδικασία είναι απλή, μόλις γίνουν οι απαραίτητες συνδέσεις, συνδέουμε το PICKit στον υπολογιστή και ανοίγουμε το πρόγραμμα. Το πρόγραμμα θα εμφανίσει τα στοιχεία του μικροελεγκτή. Στη συνέχεια πατάμε το κουμπί Read για να διαβάσουμε το περιεχόμενο της μνήμης του μικροελεγκτή και έπειτα το Erase για να το διαγράψουμε. Το επόμενο βήμα είναι να φορτώσουμε στο πρόγραμμα το αρχείο .hex στο οποίο βρίσκεται ο κώδικας σε δεκαεξαδική μορφή. Αυτό γίνεται από την επιλογή File> Import Hex. Το πρόγραμμα θα μας ειδοποιήσει ότι φορτώθηκε το αρχείο και το τελικό βήμα είναι να πατηθεί το κουμπί Write για να φορτωθεί το αρχείο στον μικροελεγκτή. Μετά το τέλος της διαδικασίας μας ειδοποιεί το πρόγραμμα και το αποσυνδέουμε από τον μικροελεγκτή.

### 1.12 Υλοποίηση εφαρμογής PIC

Η διαδικασία ανάπτυξης μίας νέας εφαρμογής με μικροελεγκτή PIC χωρίζεται στα παρακάτω επιμέρους στάδια:

- Διατύπωση λειτουργιών και επιλογή κατάλληλων περιφερειακών

- Επιλογή κατάλληλου μικροελεγκτή PIC
- Ανάπτυξη πρωτοτύπου και ταυτόχρονη σύνθεση προγράμματος
- Αποσφαλματοποίηση μέσω σχεδιαστικού πακέτου προσομοίωσης για ομαλό πολυμερισμό λογισμικού-υλικού (Debugging)
- Υλοποίηση τελικής κατασκευής

#### 1.12.1 Διατύπωση

Πριν από κάθε ανάπτυξη εφαρμογής, ο μηχανικός πρέπει να λαμβάνει υπόψη του πολλές παραμέτρους, όσο αναφορά την σχεδίαση, τον τρόπο σχεδίασης, τον τρόπο ανάπτυξης της εφαρμογής, την αποσφαλμάτωση του λογισμικού, την σχεδίαση πλακέτας, την επιλογή κουτιού, την εκτίμηση του προϊόντος από το κοινό, αλλά και την διάρκεια ζωής του προϊόντος. Το βασικότερο από όλα είναι η επιλογή των περιφερειακών με σκοπό να γίνει η ανάπτυξη της εφαρμογής του πιο αποδοτική και πιο γρήγορη ώστε να μπει στην παραγωγή και έπειτα στην αγορά.

#### 1.12.2 Επιλογή κατάλληλου PIC

Όταν ο μηχανικός κρίνει ποια περιφερειακά θα του είναι χρήσιμα για την υλοποίηση της εφαρμογής, τότε κάνει μελέτη για τον μικροελεγκτή που θα περιλαμβάνει όλα αυτά, και σε τι κλίμακα (πχ 10bit adc, 12 bit adc) έτσι ώστε να μπορέσει να προσδιορίσει και το κόστος της εφαρμογής, το εύρος της παραγωγής αλλά και την διαθεσιμότητα του μικροελεγκτή.

#### 1.12.3 Ανάπτυξη πρωτότυπου

Αφού ο μηχανικός καταλήξει σε τι μικροελεγκτή θα αναπτύξει την εφαρμογή του, ξεκινάει την αντιστοίχιση των προδιαγραφών του πελάτη, σε ένα διάγραμμα ροής δεδομένων, για να δει τι διαδρομή θα ακολουθήσουν τα δεδομένα, την ανάπτυξη διαδικασιών που θα υπάρξει μεταξύ υλικού και λογισμικού, και των τελικών δεδομένων που θα προκύψουν. Η συγγραφή του κώδικα, σε συνεργασία με το φύλλο δεδομένων του μικροελεγκτή και τις προδιαγραφές του πελάτη, βοηθάνε τον μηχανικό στο να αναπτύξει το λογισμικό, να το εξετάσει στην αρχή σε μορφή προσομοίωσης και έπειτα σε κάποια δοκιμαστική πλακέτα, ή ράστερ. Στη συνέχεια προχωράει στον σχεδιασμό της πλακέτας, ανάλογα με τον χρόνο που θα θέλει να δει το πρωτότυπο και να το δοκιμάσει ο πελάτης. Επίσης μπορεί να προβεί στον σχεδιασμό μιας δεύτερης έκδοσης πλακέτας που θα περιλαμβάνει έναν πολυμερισμό υλικού και λογισμικού με τον στόχο του μηχανικού να είναι πλέον η αρμονική λειτουργία όλων αυτών. Έτσι επιτυγχάνετε η υλοποίηση των προδιαγραφών του πελάτη από τον μηχανικό που συμφωνεί να αναπτύξει μια εφαρμογή βασισμένη στις εκάστοτε προδιαγραφές του πελάτη.

#### 1.12.4 Αποσφαλματοποίηση (Debugging)

Η διαδικασία της αποσφαλμάτωσης στα ενσωματωμένα συστήματα, ανάλογα με την παραγωγή που περιμένει να έχει και με τον χρόνο χρήσης στην αγορά, αυξάνεται και το επίπεδο αποσφαλμάτωσης. Για παράδειγμα, άλλο επίπεδο αποσφαλμάτωσης θα έχει ένας φακός μέχρι να βγει στην αγορά και άλλο επίπεδο αποσφαλμάτωσης θα έχει ένας inverter ηλεκτροκινητήρα αυτοκινήτου. Όπως και αντίστοιχα άλλο επίπεδο αποσφαλμάτωσης θα έχει ένα προϊόν το οποίο θα παραχθεί σε 100 κομμάτια παραγωγής και άλλο επίπεδο αποσφαλμάτωσης θα έχει ένα προϊόν που θα έχει ένα εκατομμύριο κομμάτια παραγωγής. Γενικά, στον κόσμο των ενσωματωμένων συστημάτων, υπάρχουν διάφορα μοντέλα ανάπτυξης έργων. Η κάθε ιδέα, από την σύλληψη της, μέχρι την απόσυρση της, διέρχεται από διάφορες φάσεις, σε καθεμιά εκ των οποίων πρέπει να γίνονται ορισμένες εργασίες ώστε να επιτυγχάνεται το επιθυμητό αποτέλεσμα. Σε μακροσκοπικό επίπεδο οι πολύ γενικές φάσεις είναι:

- Σύλληψη
- Κατασκευή

- Χρήση-Συντήρηση
- Απόσυρση

## 1.12 Τελική κατασκευή

Αφού επιτεύχθηκε ο πολυμερισμός μεταξύ υλικού και λογισμικού, τότε ο μηχανικός είναι έτοιμος να δώσει εντολή για μαζική παραγωγή του προϊόντος που κατασκεύασε, δοκίμασε και αξιολόγησε.

## 1.13 ΠΕΡΙΦΕΡΕΙΑΚΑ PIC18F

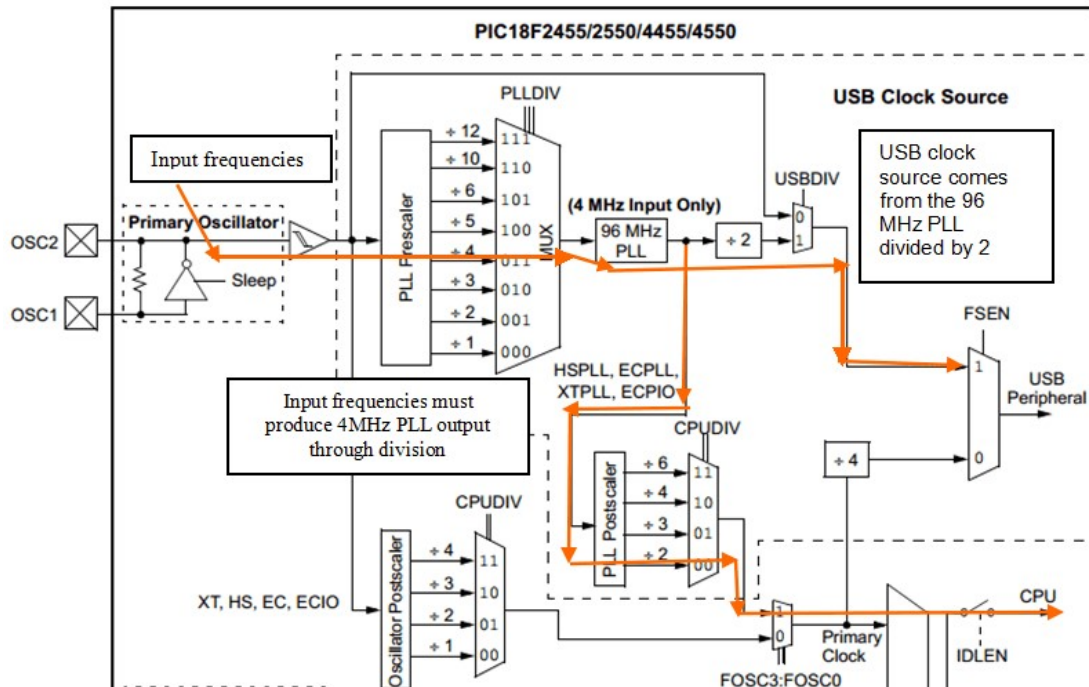
Δεν υπάρχει καλύτερος μικροελεγκτής. Ωστόσο, υπάρχει ο κατάλληλος μικροελεγκτής ανάλογα με την εφαρμογή που καλείται ο εκάστοτε μηχανικός να υλοποιήσει. Σε αυτή την ενότητα περιγράφονται περιληπτικά τα περιφερειακά των μικροελεγκτών της σειράς PIC18F, που μέχρι και σήμερα χρησιμοποιούνται σε πλήθώρα εφαρμογών δίχως την ανάγκη αναβάθμισης τους σε PIC24F ή PIC32F.

### 1.13.1 Τάση Τροφοδοσίας

Οι περισσότεροι μικροελεγκτές λειτουργούν στα 5v. Ωστόσο, αν παρατηρήσει κανείς τα φύλλα δεδομένων των μικροελεγκτών, κάποιοι μικροελεγκτές λειτουργούν με 3,3v, ενώ κάποιοι άλλοι λειτουργούν και με 1,8v. Η τάση τροφοδοσίας που θα επιλεγεί για την εφαρμογή είναι 5volt και αυτό θα επιτευχθεί με την βοήθεια ενός τροφοδοτικού 5volt, το οποίο θα συνδεθεί στις ακίδες του μικροελεγκτή που ενεργοποιεί την τροφοδοσία μέσω USB.

### 1.13.2 Ρολόι

Ο μικροελεγκτής, για να λειτουργήσει, χρειάζεται το λεγόμενο ρολόι (ταλαντωτής), το οποίο παρέχεται είτε από εξωτερικό κύκλωμα κρυστάλλου, είτε από το εσωτερικό ρολόι του μικροελεγκτή. Το σημαντικό με το ρολόι είναι πως, όσο μεγαλύτερη η συχνότητα που λειτουργεί το ρολόι, τόσο περισσότερο ρεύμα χρειάζεται ο μικροελεγκτής. Στην περίπτωση του PIC18F4550, σύμφωνα με το φύλλο δεδομένων στα ηλεκτρικά χαρακτηριστικά, με το PLL συνδεδεμένο και με εξωτερικό κύκλωμα κρυστάλλου, το ρεύμα κυμαίνεται από τα 20mA μέχρι τα 40mA. Αυτό καταλαβαίνει κανείς πως σε εφαρμογές όπου η κατανάλωση ρεύματος αποτελεί κρίσιμο παράγοντα, δεν μπορεί ο μικροελεγκτής να δουλεύει σε τέτοια συχνότητα, καθώς η πηγή (μπαταρία) θα εξαντλείται με συνέπεια να δημιουργείται πρόβλημα στην ομαλή λειτουργία της εφαρμογής, καθώς όσο χαμηλώνει η τάση στην πηγή, μειώνεται η απόδοση του μικροελεγκτή. Για τον σκοπό της εφαρμογής, καθώς δεν κρίνεται κρίσιμη ως προς τον χρόνο εκτέλεσης εντολής, θα προτιμηθεί ο εσωτερικός ταλαντωτής του μικροελεγκτή, ο οποίος ελέγχεται μέσω του καταχωρητή OSCCON, και θα χρησιμοποιηθεί η τιμή του ταλαντωτή στα 8MHz, κάνοντας τον μικροελεγκτή ικανό να ανταποκριθεί στις απαιτήσεις της εφαρμογής.



Σχήμα 1.3 Εσωτερικό διάγραμμα ταλαντωτή

### 1.13.3 Timers

Ο Timer, αποτελεί αναπόσπαστο κομμάτι λειτουργίας του κάθε μικροελεγκτή. Ο χρονιστής, όπως λέγεται αλλιώς, είναι μετρητής που οδηγείται από ένα εξωτερικό παλμό ρολογιού (κρύσταλλος) ή εσωτερικό ταλαντωτή του μικροελεγκτή. Ο χρονιστής μπορεί να είναι μήκους 8 ή 16bit. Ο χρήστης μπορεί να ρυθμίσει τον χρονιστή μέσω του λογισμικού, είτε την ενεργοποίηση, αλλά και την απενεργοποίηση του από κάποιους καταχωρητές που αφορούν την προτεραιότητα, η οποία χωρίζεται σε υψηλή και χαμηλή. Το προαναφερθέν είναι πολύ σημαντικό χαρακτηριστικό των μικροελεγκτών PIC18F. Ο συγκεκριμένος μικροελεγκτή, έχει τέσσερις χρονιστές, οι οποίοι μπορούν να είναι είτε χαμηλής είτε υψηλής προτεραιότητας, είτε 8 είτε 16 bit. Πιο συγκεκριμένα ο Timer2 είναι άμεσα συνδεδεμένος με το PWM περιφερειακό του μικροελεγκτή ενώ οι Timers 1 και 3 με τα περιφερειακά capture&compare. Συγκεκριμένα για την εφαρμογή, θα επιλεγθεί το περιφερειακό capture, το οποίο εντοπίζει ένα γεγονός που συμβαίνει σε μια ακίδα του μικροελεγκτή. Το capture βλέπει την αλλαγή σε μια ακίδα του μικροελεγκτή (από high σε low και το αντίστροφο) και την τοποθετεί σε έναν καταχωρητή.

### 1.13.4 Watchdog Timer

Η πληθώρα των μικροελεγκτών έχει μια μονάδα Watchdog Timer. Watchdog είναι ένας χρονιστής, ο οποίος ανανεώνεται από το πρόγραμμα χρήστη. Κάθε φορά που το πρόγραμμα αποτυγχάνει να ανανεώσει το watchdog, ο program counter επιστρέφει στην θέση 1 του προγράμματος, δηλαδή κάνει reset. Χρησιμοποιείται σε πολλές εφαρμογές σε περίπτωση που κολλήσει ο program counter και συγκεκριμένα σε εφαρμογές πραγματικού χρόνου (Real time).

### 1.13.5 Κουμπί επανεκκίνησης

Σε πολλές εφαρμογές, είτε γιατί ο program counter έχει κολλήσει σε κάποια καθυστέρηση, είτε γιατί η εφαρμογή έχει κάποιο θέμα, τοποθετείται συνήθως στην ακίδα του μικροελεγκτή η οποία όταν

γειώνεται, ο program counter επιστρέφει πάλι στην αρχή και ξεκινάει η εφαρμογή να εκτελείται από την πρώτη εντολή. Συγκεκριμένα, στον μικροελεγκτή PIC18F4550, πρέπει ο καταχωρητής MCLR να είναι ενεργοποιημένος, και να τοποθετηθεί σε σειρά με την αντίσταση ένας διακόπτης ο οποίος όταν πατιέται ο μικροελεγκτής να κάνει επανεκκίνηση του προγράμματος.

#### 1.13.6 Διακοπές

Η διακοπή είναι από τα στοιχεία του μικροελεγκτή που το κάνουν να ξεχωρίζει, να μπορεί να εκτελεί αλληλουχίες εντολών σε πραγματικό χρόνο με ακρίβεια χιλιοστού του δευτερολέπτου. Μία διακοπή ενεργοποιεί συγκεκριμένο bit, που όταν ενεργοποιείται, ενεργοποιείται μια ρουτίνα η οποία περιλαμβάνει ένα σετ εντολών που εκτελούνται μέσω της ρουτίνας διακοπής. Έπειτα, ο program counter επιστρέφει στην βασική ρουτίνα εκτέλεσης του προγράμματος της εφαρμογής.

Η σειρά μικροελεγκτών PIC18F περιλαμβάνει τόσο χαμηλής όσο και υψηλής προτεραιότητας επίπεδα διακοπών. Το χαρακτηριστικό αυτό επιτρέπει να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου, όπου ανάλογα με συγκεκριμένη εξωτερική διακοπή, μπορεί να ενεργοποιηθεί η χαμηλή ή η υψηλή προτεραιότητα. Αυτό, όπως αντιλαμβάνεται κάποιος, μπορεί να χρησιμοποιηθεί σε κάποια εφαρμογή ασφαλείας. Για παράδειγμα σε συναγερμό η αλλαγή σε κάποια από τις ακίδες του μικροελεγκτή αλλάζει την κατάσταση (ο διαρρήκτης ανιχνεύεται από την παγίδα) ενεργοποιείται την διακοπή υψηλής προτεραιότητας που ενεργοποιεί την σειρήνα.

#### 1.13.7 Brown Out Detector

Ο ανιχνευτής Brown-Out είναι κοινό στοιχείο πολλών μικροελεγκτών. Ελέγχει την τάση της μπαταρίας που διοχετεύεται στον μικροελεγκτή και αν πέσει κάτω από το όριο που έχει θέσει ο μηχανικός, αυτό έχει σαν αποτέλεσμα τον program vector να κάνει επανεκκίνηση συνέχεια (reset) και να μην εκτελείται το πρόγραμμα. Έτσι ο χρήστης καταλάβει πως υπάρχει κάποιο πρόβλημα και να απευθυνθεί στον υπεύθυνο για την λύση του προβλήματος.

#### 1.13.8 ADC module

Το ADC module (Μετατροπέας από αναλογικό σε ψηφιακό) χρησιμοποιείται για την μετατροπή μιας αναλογικής μέτρησης σε ψηφιακή. Αυτή η λειτουργία του μικροελεγκτή είναι χρήσιμη κυρίως για έλεγχο. Γενικά το adc module του μικροελεγκτή μπορεί να κάνει ποικίλες λειτουργίες. Μπορεί να ελέγξει την τάση της μπαταρίας που τροφοδοτεί το κύκλωμα και να μετρήσει φυσικά μεγέθη (θερμοκρασία, υγρασία κτλ). Επίσης μερικοί μηχανικοί για να γλυτώσουν ακίδες, τοποθετούν τις αντιστάσεις μαζί με μπουτόν και ανάλογα με το μπουτόν που πατιέται το adc module βγάζει διαφορετική τιμή. Με αυτόν τον τρόπο μπορεί να φτιάξει και ένα μικρό πληκτρολόγιο με απόλυτη ακρίβεια, καθώς το εσωτερικό κύκλωμα του μικροελεγκτή (στην περίπτωση του PIC18F4550) περιλαμβάνει κύκλωμα RC, δηλαδή έχει και ενσωματωμένο hardware debounce. Αυτό έχει ως αποτέλεσμα ο μηχανικός να γλυτώσει κόστος και να φτιάξει μια συσκευή με προχωρημένη εφαρμογή του adc module. Σε πολλές εφαρμογές, επίσης χρησιμοποιείται και για τον έλεγχο εξαρτημάτων, για παράδειγμα, αν ένα τρανζίστορ είναι ενεργό και επιστρέφει μια τιμή ή είναι καμένο και δεν επιστρέφει καμία τιμή.

#### 1.13.9 UART module

Ο PIC18F4550 υποστηρίζει τα σειριακά πρωτόκολλα ασύγχρονης επικοινωνίας UART, αλλά και σύγχρονης επικοινωνίας, τα οποία επιτρέπουν τη σύνδεση είτε με εξωτερικά περιφερειακά επί της πλακέτας του συστήματος (π.χ. ADC, EEPROM, ψηφιακά αισθητήρια με σειριακή σύνδεση κτλ) είτε με εξωτερικές συσκευές (π.χ. PC κτλ).

Στη UART η σειριακή επικοινωνία γίνεται ασύγχρονα, το σήμα δηλαδή δεν είναι συγχρονισμένο με κάποιο κοινό σήμα χρονισμού μεταξύ πομπού και δέκτη και ο συγχρονισμός γίνεται από το ίδιο το σήμα δεδομένων. Τα δεδομένα μεταδίδονται σε πακέτα των 8bit .

Οι ταχύτητες επικοινωνίας μπορεί να είναι από 300bps μέχρι 115Kbps, με συνηθισμένες τιμές τα 9600bps και τα 19200bps.

Για την ρύθμιση του ρυθμού επικοινωνίας διαθέτει κύκλωμα παραγωγής ρολογιού συγχρονισμού και η επιθυμητή ταχύτητα επικοινωνίας ρυθμίζεται από το πρόγραμμα. Όσο αναφορά τους ακροδέκτες επικοινωνίας για την μετάδοση των δεδομένων αυτοί είναι οι RX και TX που αντιστοιχούν στις θύρες RC6 και RC7.

#### 1.13.10 EEPROM

Η μνήμη EEPROM είναι συχνή σε πολλούς μικροελεγκτές. Το πλεονέκτημα της μνήμης EEPROM είναι πως ο μηχανικός μπορεί να αποθηκεύσει δεδομένα τα οποία μένουν στην μνήμη και δεν επηρεάζονται από εξωτερικούς παράγοντες (άνοιγμα- κλείσιμο εφαρμογής, αλλαγή μπαταρίας). Στην συγκεκριμένη εφαρμογή, ο μικροελεγκτής PIC18F4550 έχει 256 θέσεις στην μνήμη EEPROM.

#### 1.13.11 LCD Driver

Μερικοί μικροελεγκτές, έχουν ενσωματωμένα lcd drivers, για την οδήγηση απευθείας lcd οθονών δίχως την χρήση ρουτινών για οδήγηση lcd. Μόνο με τις κατάλληλες τιμές που θέτει ο μηχανικός στους καταχωρητές, που μέσω κάποιων ακίδων που είναι συνδεδεμένες με την οθόνη, στέλνει τα δεδομένα. Αυτό έχει ως αποτέλεσμα την αποδοτικότερη χρήση οθονών lcd, καθώς σε πολλές εφαρμογές, η μεθοδολογία που χρησιμοποιείται με τις οθόνες έχει ως αποτέλεσμα την καθυστέρηση πολλών msec και κατ' επέκταση την εν δυνάμει καθυστέρηση της εφαρμογής

#### 1.13.12 Συγκριτής

Ο PIC18F4550 έχει ενσωματωμένο αναλογικό συγκριτή, ο οποίος μπορεί να χρησιμοποιηθεί ανάλογα με το πως ορίζει ο μηχανικός τον καταχωρητή CMCON. Ο συγκριτής, όπως είναι γνωστό, συγκρίνει δύο τάσεις και ανάλογα με την συνδεσμολογία, τις τιμές του καταχωρητή CMCON και CVRCON. Ρυθμίζει την συνδεσμολογία του συγκριτή και την τάση αναφοράς αν οριστεί η ανάλογη συνδεσμολογία από τον μηχανικό.

#### 1.13.13 Λειτουργία Ύπνου

Η πλειοψηφία των μικροελεγκτών, έχουν την λειτουργία ύπνου, δηλαδή μια κατάσταση στην οποία ο μικροελεγκτής μπορεί να απενεργοποιήσει το εσωτερικό ρολόι και τα περιφερειακά του και να περάσει στην λεγόμενη κατάσταση ύπνου (Sleep Mode). Ο μηχανικός, για να περάσει τον μικροελεγκτή στην κατάσταση αυτή και σε αυτήν την περίπτωση περιλαμβάνετε και ο μικροελεγκτής PIC18F4550, αρκεί να ρυθμίσει τον καταχωρητή OSCCON. Ο καταχωρητής αυτός είναι υπεύθυνος στην περίπτωση όπου ο μηχανικός επιλέξει τον εσωτερικό ταλαντωτή στα πόσα MHz θα τρέχει ο program vector. Για την λειτουργία ύπνου, αρκεί να θέσει ο χρήστης το bit OSCCON.IDLEN=0 μέσα στο πρόγραμμα με την εντολή Sleep

## 2. Διαδικασίες ανάπτυξης λογισμικού

Όσο για τις διαδικασίες ανάπτυξης λογισμικού, μπορούν να διακριθούν ως ακολούθως:

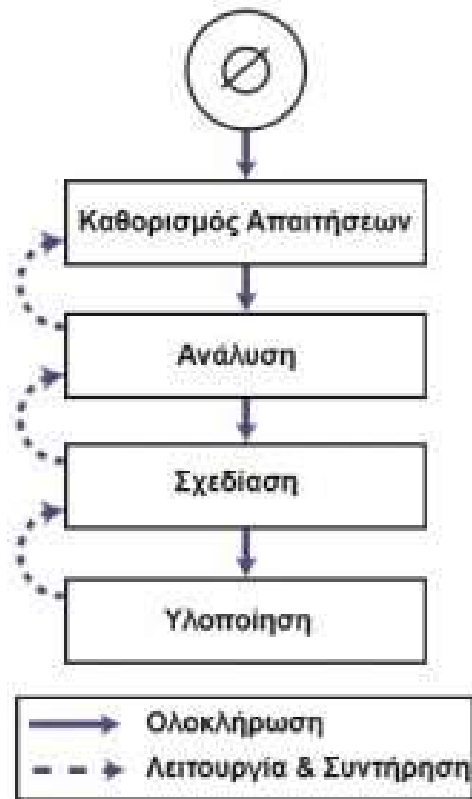
- Προδιαγραφή: Καθορισμός των εργασιών που θα επιτελεί το λογισμικό, καθώς και των περιορισμών και των παραδοχών που ισχύουν.
- Ανάπτυξη: Κατασκευή του λογισμικού. Εδώ, σε όλα τα μοντέλα κύκλου ζωής μπορούμε να διακρίνουμε τρεις επιμέρους φάσεις: Την ανάλυση, τη σχεδίαση και τη συγγραφή του πηγαίου κώδικα (source code) την οποία στην συνέχεια θα ονομάζουμε και κωδικοποίηση.
- Επαλήθευση: Επιβεβαίωση της ικανοποίησης των προδιαγραφών και της μη ύπαρξης σφαλμάτων.
- Εξέλιξη: Επαύξηση των λειτουργικών χαρακτηριστικών του λογισμικού ή τροποποίηση υφιστάμενων, προκειμένου να ικανοποιούνται οι μεταβαλλόμενες ανάγκες.

Ένα μοντέλο κύκλου ζωής λογισμικού στοχεύει στην καθοδήγηση του κατασκευαστή προκειμένου αυτός να επιτύχει την καλύτερη δυνατή υλοποίηση των διαδικασιών ανάπτυξης λογισμικού. Λέγοντας «καλύτερη δυνατή» εννοούμε περισσότερο παραγωγική, με τα λιγότερα δυνατά σφάλματα και το μικρότερο δυνατό ρίσκο στις εκάστοτε συνθήκες. Τα παραπάνω μπορούν να διαφοροποιούνται ανάλογα με το μέγεθος και το θεματικό πεδίο κάθε εφαρμογής λογισμικού, με την εμπειρία και τα ιδιαίτερα χαρακτηριστικά του κάθε κατασκευαστή και με το εκάστοτε περιβάλλον ανάπτυξης.

Μια σημαντική παράμετρος που καταδεικνύει τη σημασία των μοντέλων κύκλου ζωής είναι το κόστος, ιδωμένο με την ευρύτερη σημασία του. Το κόστος αναθεώρησης αποφάσεων ή/και διόρθωσης σφαλμάτων αυξάνετε όσο μεγαλύτερη είναι και η απαιτούμενη οπισθοδρόμηση της διαδικασίας που αυτή συνεπάγεται. Το κόστος αυτό δεν αφορά μόνο οικονομικούς πόρους που αποδίδονται στο έργο, αλλά και χρόνο καθυστέρησης, που δεν είναι πάντα διαθέσιμος σε πραγματικές συνθήκες. Επίσης, είναι συχνό φαινόμενο οι παρενέργειες στο υπόλοιπο σύστημα λογισμικού (side-effects), οι οποίες μπορούν να μεταβάλλουν προς το χειρότερο τα ποιοτικά του χαρακτηριστικά και δεν είναι εύκολο να εντοπιστούν από την αρχή.

### 2.1 Μοντέλο καταρράκτη

Ένα από τα πιο διαδεδομένα μοντέλα κύκλου ζωής είναι αυτό του καταρράκτη (Waterfall), το οποίο φαίνεται στο Σχήμα 2.1. Η κεντρική ιδέα του μοντέλου του καταρράκτη είναι ότι το σύστημα λογισμικού αναπτύσσεται περνώντας ολόκληρο από διαδοχικές επιμέρους φάσεις, καθεμία από τις οποίες θεωρείται περατωμένη με την παραγωγή ορισμένων συστατικών λογισμικού. Κάθε επιμέρους φάση ολοκληρώνεται με μια εργασία επαλήθευσης/επικύρωσης των προϊόντων της, κατά την οποία αποφασίζεται η μετάβαση ή όχι στην επόμενη. Το λογισμικό εμφανίζεται πλήρες, δηλαδή με όλα τα λειτουργικά του χαρακτηριστικά, από την επιμέρους φάση της συνένωσης και μετά. Χαρακτηριστικό του μοντέλου του καταρράκτη είναι ότι, για να ξεκινήσει μια φάση πρέπει να έχει ολοκληρωθεί πλήρως η προηγούμενη. Η ανάπτυξη με τον τρόπο αυτό χαρακτηρίζεται ακολουθιακή, διότι οι επιμέρους φάσεις από τις οποίες διέρχεται είναι διακριτές και ακολουθούν η μία την άλλη.



Σχήμα 2.1 Μοντέλο Καταρράκτη

Αρχικά καθορίζονται οι απαιτήσεις από το σύστημα και το λογισμικό, αντίστοιχα. Όπως αναφέρθηκε στο Κεφάλαιο 1, το λογισμικό είναι μια μόνο από τις συνιστώσες του συστήματος το οποίο μπορεί να περιλαμβάνει και άλλες ειδικές συσκευές, άλλες εφαρμογές λογισμικού κ.ά.

Ακολουθώς γίνεται η προκαταρκτική και η λεπτομερής σχεδίαση του λογισμικού αντίστοιχα. Κατά την προκαταρκτική σχεδίαση καθορίζονται οι μονάδες που θα αποτελούν το λογισμικό, καθώς και οι συσχετίσεις μεταξύ τους. Ο καθορισμός αυτός μπορεί να γίνει σε περισσότερα από ένα επίπεδα λεπτομέρειας, ανάλογα με το μέγεθός και την πολυπλοκότητα του λογισμικού. Το πρώτο επίπεδο (αυτό με τη μικρότερη λεπτομέρεια) περιέχει τα υποσυστήματα, το δεύτερο περιέχει τις μονάδες μέσα σε κάθε υποσύστημά κ.ο.κ. Κατά τη λεπτομερή σχεδίαση καθορίζεται η εσωτερική δομή κάθε μονάδας λογισμικού, η οποία αντιστοιχεί πρακτικά σε μονάδες πηγαίου κώδικα προγράμματος. Ο καθορισμός αυτός περιλαμβάνει όλα τα απαραίτητα στοιχεία (αλγόριθμοι, δομές δεδομένων κτλ.), ώστε η συγγραφή του πηγαίου κώδικα, που ακολουθεί, να είναι μια διαδικασία διεκπεραίωσης και μόνο. Ακολουθεί η συνένωση των μονάδων σε σύστημα και ο έλεγχος του συστήματος, η ολοκλήρωση του οποίου, επιτρέπει την παράδοση ολόκληρου του προϊόντος στον πελάτη και το πέρασμα στη φάση της λειτουργίας και συντήρησης. Το μοντέλο του καταρράκτη υπήρξε για μεγάλο διάστημα το πιο διαδεδομένο μοντέλο κύκλου ζωής λογισμικού. Είναι ιδιαίτερα χρήσιμο σε περιπτώσεις όπου οι απαιτήσεις από το λογισμικό είναι από την αρχή γνωστές και δε μεταβάλλονται κατά την ανάπτυξη του λογισμικού και μπορεί να χρησιμοποιηθεί αποτελεσματικά για τη βιομηχανοποίηση της ανάπτυξης τέτοιων εφαρμογών. Για παράδειγμα, τέτοιες είναι οι εφαρμογές που επιλύουν μεγάλα προβλήματα χρησιμοποιώντας μαθηματικούς υπολογισμούς. Σε πολλές, όμως, περιπτώσεις εφαρμογών οι

απαιτήσεις είτε δεν είναι από την αρχή και με σαφήνεια γνωστές, είτε ενδέχεται να μεταβληθούν κατά τη διάρκεια της ανάπτυξης.

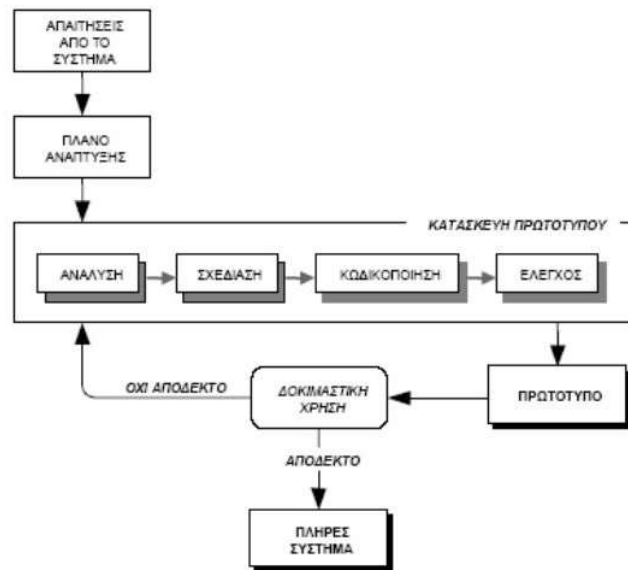
## 2.2 Μοντέλο Προτυποποίησης

Η κεντρική ιδέα του μοντέλου προτυποποίησης (prototyping model) είναι η ανάπτυξη του λογισμικού όχι εξολοκλήρου, αλλά σε τμήματα, που ονομάζονται «πρωτότυπα». Οι διαδικασίες ανάπτυξης επαναλαμβάνονται για ένα τμήμα του συστήματος κάθε φορά και, για το λόγο αυτό, το μοντέλο χαρακτηρίζεται ως επαναληπτικό. Κάθε πρωτότυπο περιλαμβάνει τις βασικές από τις λειτουργίες που προορίζεται να εκτελεί το λογισμικό και τίθεται σε δοκιμασία από τον πελάτη. Από εκεί συλλέγονται παρατηρήσεις και η διαδικασία κατασκευής νέου πρωτοτύπου επαναλαμβάνεται μέχρις ότου ένα πρωτότυπο να ικανοποιεί τις απαιτήσεις, δηλαδή να εκτελεί τις επιθυμητές λειτουργίες του λογισμικού με τρόπο ικανοποιητικό και να γίνεται αποδεκτό από τον πελάτη (Σχήμα 2.2). Από το σημείο αυτό και μετά μπορούν να προστεθούν και οι υπόλοιπες λειτουργίες, ώστε το λογισμικό να ολοκληρωθεί. Ένα σημαντικό πλεονέκτημά του μοντέλου αυτού είναι η δυνατότητα απόκτησης άποψης για την εφαρμογή λογισμικού νωρίτερα απ' ό,τι στο μοντέλο του καταρράκτη. Αυτό μπορεί να γλιτώσει την ανάπτυξη από καθυστερήσεις (και συνεπαγόμενα κόστη) ή ακόμη και από ολική αποτυχία, τα οποία θα επέρχονταν, αν ο κατασκευαστής αναγκαζόταν να οπισθοδρομήσει την ανάπτυξη, ενώ αυτή είχε προχωρήσει πολύ. Παράλληλα, ιδιαίτερη σημασία αποκτά η διοίκηση του έργου, η οποία πρέπει να εξασφαλίζει την υλοποιησιμότητα του πρωτοτύπου και την εύκολη τροποποίησή του. Κάθε κατασκευή πρωτοτύπου μπορεί να θεωρηθεί ως ένα μικρό έργο λογισμικού το οποίο κατασκευάζεται με διαδικασίες που μπορούν να ακολουθούν άλλα μοντέλα κύκλου ζωής, όπως αυτό του καταρράκτη.

Με βάση τις παραπάνω παρατηρήσεις, το μοντέλο προτυποποίησης χρησιμοποιείται στην ανάπτυξη εφαρμογών λογισμικού για τις απαιτήσεις από τις οποίες δεν υπάρχει βεβαιότητα στην αρχή της ανάπτυξης, οπότε δεν μπορούν να συμφωνηθούν και να παγιωποιηθούν. Τέτοιες είναι εφαρμογές που κατασκευάζονται για πρώτη φορά ή που είναι στενά εξαρτημένες από τον πελάτη, χωρίς να υπάρχει αποδεκτό προηγούμενο παράδειγμα. Ωστόσο, το μέγεθος των εφαρμογών αυτών δεν μπορεί να είναι ιδιαίτερα μεγάλο, διότι ο χρόνος ανάπτυξης κάθε πρωτοτύπου μεγαλώνει και η απαιτούμενη ευελιξία μειώνεται.

Για την ανάπτυξη της εφαρμογής, επιλέχθηκε η μέθοδος του μοντέλου προτυποποίησης, γιατί το μέγεθος της εφαρμογής δεν μπορεί να είναι ιδιαίτερα μεγάλο, και ο χρόνος υλοποίησης του είναι σχετικά μικρός, σε σχέση με τα άλλα μοντέλα ανάπτυξης ενσωματωμένων συστημάτων.

## Το μοντέλο Πρωτοτυποποίησης



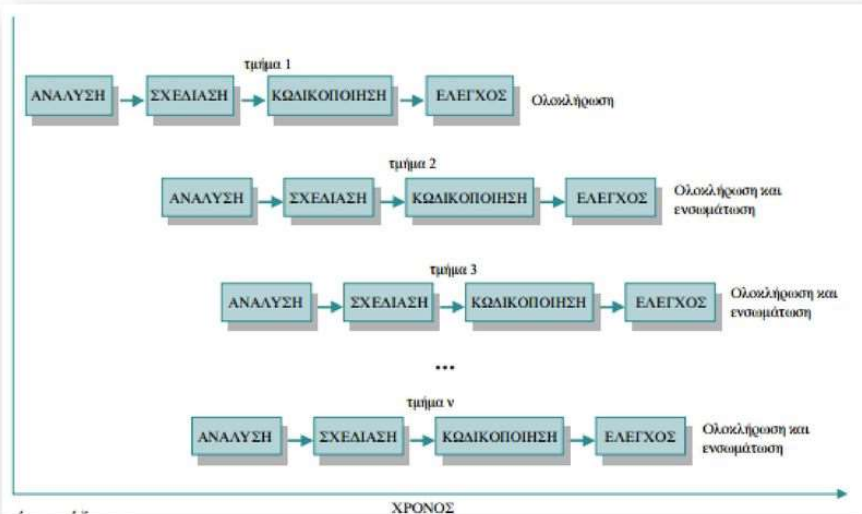
Σχήμα 2.2 Μοντέλο πρωτοτυποποίησης

### 2.3 Μοντέλο λειτουργικής επαύξησης

Το μοντέλο της λειτουργικής επαύξησης (incremental model) συνδυάζει την ακολουθιακή ανάπτυξη του μοντέλου του καταρράκτη με την τμηματική ανάπτυξη του μοντέλου της πρωτοτυποποίησης. Κεντρική ιδέα είναι η κατάτμηση του υπό κατασκευή λογισμικού σε τμήματα που αναπτύσσονται ανεξάρτητα, ακολουθώντας το καθένα ακολουθιακή ανάπτυξη σύμφωνα με το μοντέλο του καταρράκτη, όπως φαίνεται στην Σχήμα 2.3. Κατά την αρχική φάση ανάλυσης και σχεδίασης αποφασίζονται τα τμήματα στα οποία θα κατακτηθεί η εφαρμογή, η ανάπτυξη των οποίων γίνεται στη συνέχεια ανεξάρτητα και παράλληλα. Όταν ολοκληρώνεται η ανάπτυξη κάθε τμήματος, αυτό ενσωματώνεται στο σύνολο της εφαρμογής, διαδικασία η οποία δικαιολογεί και τον τίτλο «λειτουργική επαύξηση».

## Τεχνολογία Λογισμικού

- Μοντέλο λειτουργικής επαύξησης



Σχήμα 2.3 Μοντέλο λειτουργικής επαύξησης

Πλεονεκτήματα της ιδέας είναι η δυνατότητα παράλληλης ανάπτυξης, η οποία τελικά καταλαμβάνει μικρότερο χρόνο, καθώς και ο διαδοχικός εμπλουτισμός των λειτουργικών χαρακτηριστικών του λογισμικού. Τα βασικά μειονεκτήματα του μοντέλου είναι τα ακόλουθα: Η αρχική κατάτμηση και γενική σχεδίαση του συστήματος αποκτά ιδιαίτερη βαρύτητα. Σφάλματα σε αυτή μπορούν να έχουν σημαντικές επιπτώσεις στο λογισμικό που θα κατασκευαστεί στη συνέχεια. Σε περίπτωση μεταβολής των λειτουργικών απαιτήσεων κατά τη χρήση του ημιτελούς συστήματος, μπορεί η αρχιτεκτονική αυτού να μεταβληθεί σε βαθμό που να κλονιστεί η ανάπτυξη των υπόλοιπων τμημάτων αυτού. Το μοντέλο της λειτουργικής επαύξησης χρησιμοποιείται στην ανάπτυξη μεγάλων εφαρμογών λογισμικού για τις οποίες ισχύουν οι απαιτήσεις του μοντέλου του καταρράκτη, δηλαδή σαφής γνώση και μικρή ή καθόλου μεταβλητότητα των απαιτήσεων κατά την ανάπτυξη.

### 2.4 Το σπειροειδές μοντέλο

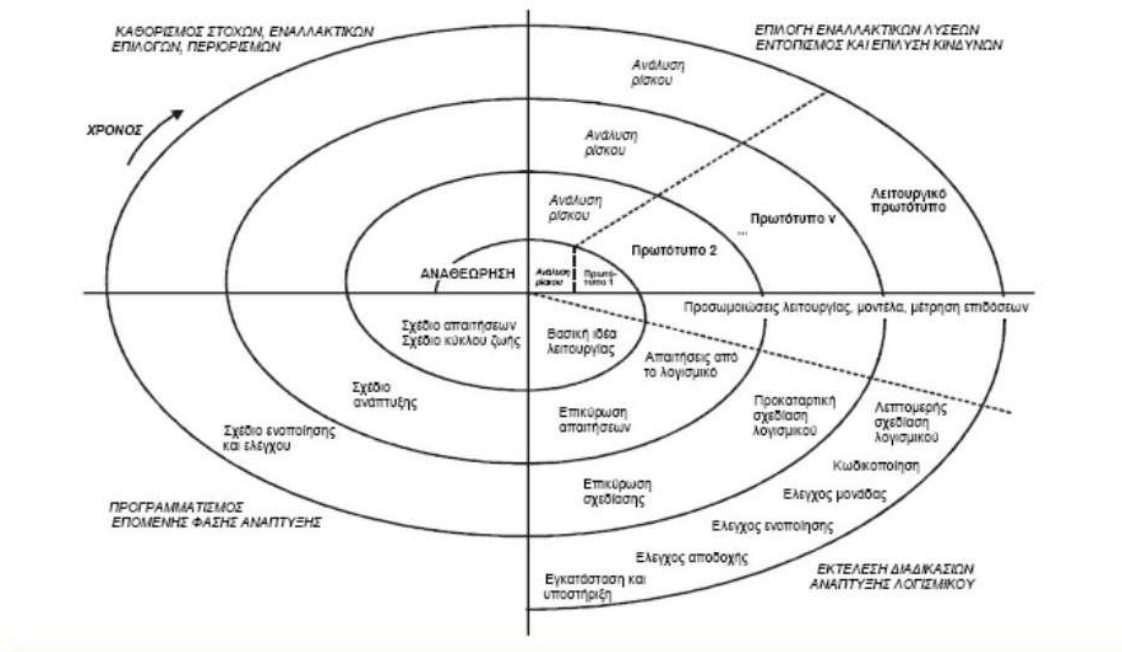
Τα μοντέλα κύκλου ζωής που παρουσιάστηκαν μέχρι τώρα αποτελούν παραλλαγές της βασικής ιδέας του μοντέλου του καταρράκτη. Η ανάπτυξη παραμένει, επί της ουσίας, μια ακολουθιακή διαδικασία, η οποία εφαρμόζεται είτε σε ολόκληρο είτε σε ένα μέρος του συστήματος. Από ότι φαίνεται, δεν είναι η σύλληψη των διαδικασιών ανάπτυξης λογισμικού που διαφοροποιεί τα μοντέλα κύκλου ζωής, αλλά η διάταξή τους. Στο μοντέλο της προτυποποίησης, καθώς και σε αυτό της λειτουργικής επαύξησης, η κατάτμηση είναι λίγο ως πολύ αυθαίρετη. Το ρίσκο δεν αποτιμάται, με αποτέλεσμα κάθε οπισθοδρόμηση ή ανατροπή να κοστίζει σε χρόνο και σε οικονομικούς πόρους και συχνά δε σε συνολική αποτυχία των έργων. Από την άλλη, η κάθετη αποδοχή των αυστηρών φάσεων που

προτείνονται από το μοντέλο του καταρράκτη, δεν είναι εφικτή να ακολουθείται σε όλες τις περιπτώσεις και από όλους τους κατασκευαστές, με αποτέλεσμα η ανάπτυξη λογισμικού είτε να γίνεται άναρχη, με βάση τη διαίσθηση των κατασκευαστών, είτε να είναι μια δαπανηρή και στρυφνή διαδικασία, στην οποία «πρέπει» να ακολουθηθούν κάποια συγκεκριμένα βήματα, ανεξάρτητα από τις εκάστοτε συνθήκες. Απάντηση στα παραπάνω έρχεται να δώσει το σπειροειδές μοντέλο κύκλου ζωής, το οποίο πήρε το όνομα του από την απεικόνιση σε διάγραμμα, όπως φαίνεται στο Σχήμα 2.4. Πρόκειται για μια γενίκευση των μοντέλων της λειτουργικής επαύξησης και της πρωτοτυποποίησης, με σημαντικά νέα στοιχεία:

- Οι φάσεις και οι διαδικασίες ανάπτυξης λογισμικού δεν είναι προκαθορισμένες από το μοντέλο, αλλά εξειδικεύονται στο χώρο της εφαρμογής του.
- Η ανάπτυξη ολόκληρου του συστήματος χωρίζεται σε πολλούς κύκλους, σε καθέναν από τους οποίους προστίθενται νέα λειτουργικά χαρακτηριστικά στο σύστημα.
- Πριν από την έναρξη κάθε κύκλου γίνεται μια μελέτη σκοπιμότητας και ανάλυση κινδύνων, από την οποία προκύπτουν, αφενός οι συγκεκριμένες εργασίες που θα εκτελεστούν μέσα στον κύκλο και αφετέρου η ίδια η εφικτότητα εκτέλεσης του κύκλου αυτού.

Το σπειροειδές μοντέλο δεν καθορίζει εκ των προτέρων ποιες ακριβώς είναι οι εργασίες ανάπτυξης λογισμικού που πρέπει να γίνουν, ούτε σε ποια έκταση του συστήματος αυτές θα εφαρμοστούν. Διαφορετικές διαδικασίες ανάπτυξης μπορεί να επιλεγούν για διαφορετικά τμήματα του λογισμικού. Αυτό που προτείνει είναι ότι ο καθορισμός των λεπτομερειών υλοποίησης πρέπει να γίνεται συνεχώς κατά την ανάπτυξη με ευθύνη και τεκμηρίωση από πλευράς του ίδιου του κατασκευαστή.

# Το Σπειροειδές μοντέλο



Σχήμα 2.4 Σπειροειδές Μοντέλο

Η εφαρμογή του σπειροειδούς μοντέλου δεν είναι πάντα εύκολη υπόθεση. Εισάγονται νέες εργασίες, που δεν ανήκουν καθαρά στις εργασίες ανάπτυξης λογισμικού, αλλά αφορούν την τεκμηρίωση της σκοπιμότητας και τον τμηματικό προγραμματισμό της ανάπτυξης. Οι εργασίες αυτές επιφέρουν ασφαλώς κάποιο κόστος, το οποίο όμως μπορεί να αποσβεστεί από τον έγκαιρο εντοπισμό προβλημάτων και την αποφυγή πιθανού ναυαγίου, κάτι που έχει συμβεί σε αρκετές περιπτώσεις.

Μετά την τύπωση του τυπωμένου ηλεκτρονικού κυκλώματος, ο μηχανικός ξεκινάει να κάνει διάφορα τεστ της εφαρμογής του για να δει αν λειτουργεί ο κώδικας όπως στην προσομοίωση, αλλά αυτή την φορά σε πραγματικές συνθήκες με πλακέτα, με φαινόμενα θορύβου, ακραίων θερμοκρασιών, κακής κόλλησης υλικών στην πλακέτα κ.α. τα οποία επηρεάζουν την λειτουργία του μικροελεγκτή. Αν όλα προχωρήσουν κανονικά σε αυτό το στάδιο, τότε η εφαρμογή είναι έτοιμη για παραγωγή

### 3. Σχεδιασμός και κατασκευή πλακέτας (PCB)

Σε αυτό το κεφάλαιο θα αναλύσουμε το τι είναι μια πλακέτα PCB και στη συνέχεια θα δούμε πως έγινε η σχεδίαση και η υλοποίηση της συγκεκριμένης πλακέτας. Θα δούμε το σχηματικό της πλακέτας αλλά και την τελική τυπωμένη πλακέτα.

#### 3.1 Εισαγωγή στις πλακέτες PCB

Μια πλακέτα τυπωμένου κυκλώματος (PCB ή PWB ή Perforated Circuit Board), είναι μια πλακέτα η οποία χρησιμοποιείται για την υποστήριξη και διασύνδεση ηλεκτρονικών στοιχείων μέσω αγωγίων μονοπατιών τα οποία τυπώνονται πάνω στην πλακέτα. Τα ηλεκτρονικά αυτά στοιχεία συγκολλούνται στις νησίδες ή πίστες (pads), οι οποίες είναι τμήματα των αγωγίων μονοπατιών με αρκετό χώρο για την πραγματοποίηση μιας επαφής κόλλησης (soldered joint) μεταξύ του στοιχείου και του αγωγίου μονοπατιού. Οι πίστες μπορεί να έχουν τρύπες οι οποίες διαπερνούν την πλακέτα για να επιτυγχάνεται η στήριξη των στοιχείων (through-hole technology) ή το στοιχείο τοποθετείται και γίνεται η κόλλησή του απευθείας στην πίστα (τεχνολογία επιφανειακής στήριξης surface-mount technology). Ο πιο συνηθισμένος τρόπος κατασκευής πλακετών βασίζεται στην φωτολιθογραφία (photolithography). Πιο συγκεκριμένα, οι ανεπιθύμητες περιοχές χαλκού ορίζονται μέσω στρώσεων φωτοευαίσθητης ουσίας όπου με την έκθεσή της σε υπεριώδη ακτινοβολία είναι δυνατή η απομάκρυνσή της. Με την απομάκρυνση της φωτοευαίσθητης ουσίας ο ανεπιθύμητος χαλκός είναι δυνατόν να αφαιρεθεί μέσω ειδικών χημικών διεργασιών, συνήθως εφαρμόζοντας ένα μίγμα υπεροξειδίου του υδρογόνου και υδροχλωρικού οξέος. Με αυτόν τον τρόπο παραμένουν μόνο οι επιθυμητές περιοχές χαλκού που ορίζουν και το ηλεκτρονικό μας κύκλωμα.

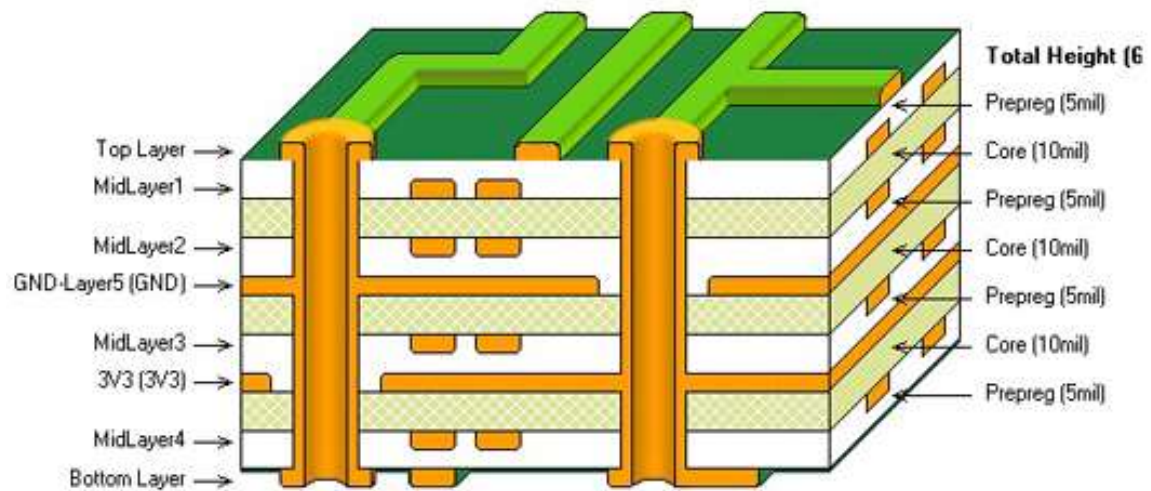


Image courtesy Terry Kozlyk

(c) VentureOutsource.com

Εικόνα 3.1 Η δομή μιας PCB πλακέτας

### 3.2 Ταξινόμηση των PCBs

Οι συνεχείς βελτιώσεις στην τεχνολογία κατασκευής PCB έχουν σαν αποτέλεσμα μια ποικιλία στους τύπους των PCB που ταιριάζουν καλύτερα με τα νέα σχέδια των ηλεκτρονικών εξαρτημάτων. Έτσι μπορούμε να ταξινομήσουμε τις PCB σε τρεις κατηγορίες[30]:

- Τυπωμένη πλακέτα μονής ή απλής όψης (single-sided PCB)
- Τυπωμένη πλακέτα διπλής όψης (double-sided PCB)
- Τυπωμένη πλακέτα πολλαπλών στρωμάτων (multi-layer PCB)

### 3.3 Τυπωμένη πλακέτα μονής ή απλής όψης(Single-sided PCB)

Σε μία τυπωμένη πλακέτα μονής όψης συνήθως όλες οι αγωγίμες διαδρομές (tracks) βρίσκονται σε μία πλευρά ενώ τα ηλεκτρονικά εξαρτήματα βρίσκονται στην άλλη πλευρά. Σε κάποιες περιπτώσεις τα ηλεκτρονικά εξαρτήματα μπορούν να βρίσκονται στην ίδια πλευρά με τις διαδρομές ή και στις δύο πλευρές ανάλογα με το πόσο σύνθετο είναι το σχέδιο.

### 3.4 Τυπωμένη πλακέτα διπλής όψης(double-sided PCB)

Τα PCBs αυτής της κατηγορίας χρησιμοποιούν αγωγίμες διαδρομές σε κάθε πλευρά της πλακέτας. Και σε αυτή την κατηγορία τα εξαρτήματα μπορούν να τοποθετηθούν είτε στη μία είτε στην άλλη πλευρά είτε και στις δύο. Το συνδυασμένο αυτό αποτέλεσμα εξασφαλίζει μεγαλύτερη πυκνότητα εξαρτημάτων συγκριτικά με τις PCBs μονής όψης. Οι διασυνδέσεις μεταξύ των αγωγίμων διαδρομών στα δύο επίπεδα επιτυγχάνεται με έναν αριθμό από μεθόδους όπως, ακίδες μεταξύ των επιπέδων, καλώδια, τους ακροδέκτες των εξαρτημάτων και επιμεταλλωμένες τρύπες (plated through holes). Η τελευταία μέθοδος είναι σήμερα η συνηθέστερη και υλοποιείται με την επικάλυψη του τοιχώματος της οπής με αγωγίμο υλικό. Οι επιμεταλλωμένες τρύπες οι οποίες χρησιμοποιούνται αποκλειστικά για την διασύνδεση αγωγίμων διαδρομών μεταξύ των δύο αντίθετων επιπέδων της πλακέτας είναι γνωστές σαν περάσματα (vias).

### 3.5 Τυπωμένη πλακέτα πολλαπλών στρωμάτων( multi-layer PCB)

Όπως προκύπτει από την ονομασία οι πλακέτες αυτές έχουν πολλαπλά επίπεδα με αγωγίμες διαδρομές, δύο από τις οποίες βρίσκονται στις επιφάνειες της πλακέτας. Τα υπόλοιπα επίπεδα βρίσκονται στο εσωτερικό της πλακέτας η οποία κατασκευάζεται με έναν αριθμό αγωγίμων διαδρομών που συνδυάζονται με μονωτικά επίπεδα. Οι συνδέσεις μεταξύ των αγωγίμων διαδρομών υλοποιείται με περάσματα μεταξύ των επιπέδων (through via) είτε με κρυφά περάσματα (buried vias). Ένα πέραςμα μεταξύ των επιπέδων απλά διαπερνά την πλακέτα ενώ ένα κρυφό πέραςμα διασυνδέει εσωτερικές αγωγίμες διαδρομές.

### 3.6 Παραλλαγές σχεδίασης PCB

Οι παραλλαγές στην σχεδίαση των PCBs σύμφωνα με τα παραπάνω είναι οι ακόλουθες:

- Κυκλώματα με οπές μεταξύ των επιπέδων (through-hole circuits)
- Κυκλώματα με εξαρτήματα επιφανειακής στήριξης (surface mounted circuits)
- Υβριδικά κυκλώματα (hybrid circuits)

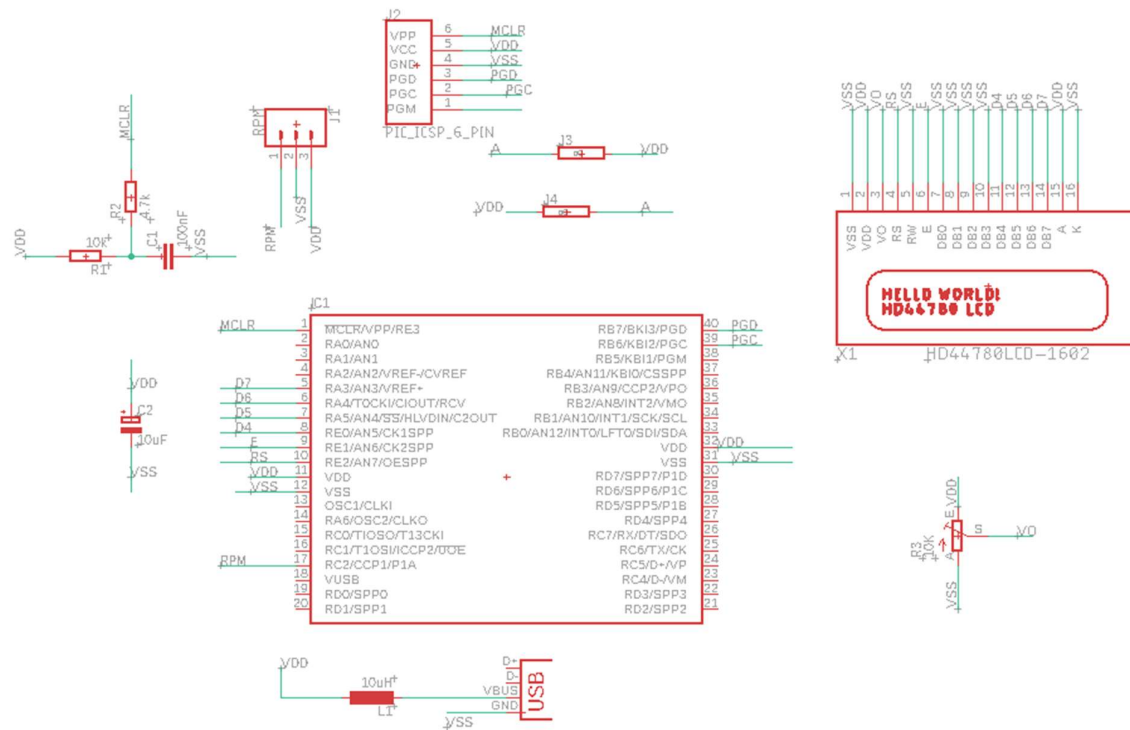
### 3.7 Ο Σχεδιασμός, η τύπωση και η τελική μορφή του PCB

Παρακάτω θα δούμε πως έγινε το ηλεκτρονικό σχέδιο της πλακέτας με τη χρήση ενός προγράμματος σχεδίασης. Επίσης θα δούμε την τελική τυπωμένη πλακέτα αλλά και την πλακέτα με την τοποθέτηση όλων των εξαρτημάτων.

### 3.8 Το σχεδιαστικό πακέτο EAGLE

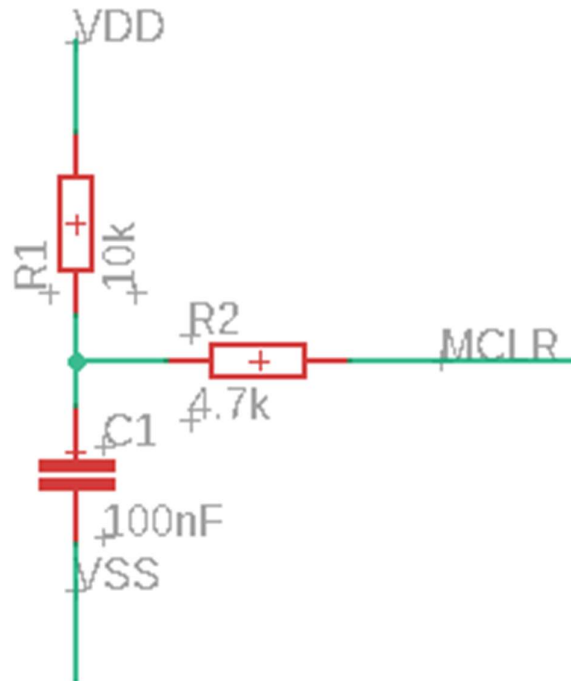
Το πρόγραμμα Eagle της εταιρείας CadSoft είναι ένα περιβάλλον σχεδίασης σχηματικών ηλεκτρονικών διαγραμμάτων και πλακετών. Μεταξύ άλλων περιλαμβάνει ένα σχεδιαστή διαγραμμάτων, ένα σχεδιαστή πλακέτας και τον αυτόματο δρομολογητή λωρίδων χαλκού. Η κατασκευή μιας πλακέτας ξεκινάει από τη δημιουργία του σχηματικού διαγράμματος του κυκλώματος, του οποίου θέλουμε να φτιάξουμε την πλακέτα, με τη βοήθεια του σχεδιαστή διαγραμμάτων. Στη συνέχεια δημιουργούμε το σχέδιο των λωρίδων χαλκού με τη βοήθεια του σχεδιαστή πλακέτας. Κάποιος μπορεί να ορίσει μόνος του τις διαδρομές του χαλκού ή να χρησιμοποιήσει τον αυτόματο δρομολογητή χαλκινων λωρίδων. Εάν το κύκλωμα είναι σχετικά μικρό, τότε κάποιος μπορεί να δημιουργήσει το σχέδιο της πλακέτας απευθείας στο σχεδιαστή πλακέτας χωρίς την δημιουργία σχηματικού διαγράμματος. Τέλος το σχέδιο αποτυπώνεται πάνω σε μια πλακέτα με διάφορες μεθόδους και ακολουθεί η αποχάλκωση της.

#### 3.8.1 Το σχηματικό διάγραμμα της πλακέτας της εφαρμογής



Σχήμα 3.1 Το σχηματικό της εφαρμογής

## 3.8.2 MCLR



Σχήμα 3.2 Η διάταξη MCLR

Η διάταξη του MCLR, σχεδιάστηκε σύμφωνα με το φύλλο δεδομένων (Datasheet) του μικροελεγκτή PIC18F4550, για τους παρακάτω λόγους:

- Ένας παλμός έναυσης παράγεται, όταν το Vdd φτάνει ένα συγκεκριμένο κατώφλι. Αυτό επιτρέπει στον μικροελεγκτή να οδηγήσει τον program counter στην πρώτη γραμμή του κώδικα. Μόνο τότε ο μικροελεγκτής μας έχει τις σωστές συνθήκες έναυσης και εκτέλεσης του κώδικα.
- Όταν η συσκευή ξεκινάει, δηλαδή φεύγει από την κατάσταση επανεκκίνησης (RESET), οι παράμετροι λειτουργίας (τάση, θερμοκρασία, συχνότητα) πρέπει να έχουν την σωστή τιμή, αλλιώς η συσκευή θα παραμείνει σε λειτουργία επανεκκίνησης, μέχρις ότου υλοποιηθούν οι συνθήκες όπου η συσκευή (μικροελεγκτής) λειτουργεί
- Σε πολλές εφαρμογές, ο πελάτης θέτει ως προδιαγραφή την δυνατότητα επανεκκίνησης του ηλεκτρικού κυκλώματος και της εκτέλεσης του πηγαίου κώδικα, με υλικό (hardware) τρόπο. Η ενεργοποίηση της ακίδας 1 (PIN 1) από τον καταχωρητή MCLRE, έχει σαν αποτέλεσμα την υλοποίηση της επανεκκίνησης του προγράμματος από εξωτερικό παράγοντα, όταν η ακίδα καταλήγει στην γείωση.

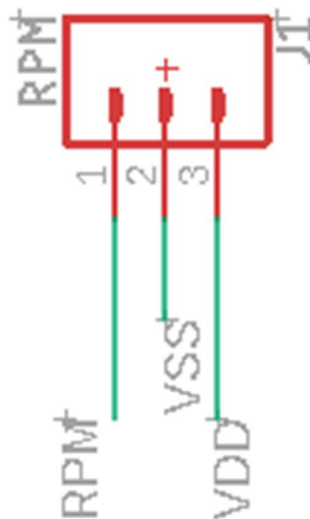
### 3.8.3 Πυκνωτής αποθρομβοποίησης



Σχήμα 3.3: Πυκνωτής αποθρομβοποίησης

Ο πυκνωτής αποθρομβοποίησης τοποθετείται κοντά στον μικροελεγκτή ώστε τα σήματα τροφοδοσίας που καταλήγουν στις ακίδες τροφοδοσίας να έχουν το απαιτούμενο πλάτος, δίχως την παρουσία θορύβου. Αυτό γίνεται διότι η απουσία του πυκνωτή αποθρομβοποίησης οδηγεί την λειτουργία του μικροελεγκτή σε κατάσταση επανεκκίνησης, λόγω του θορύβου στις ακίδες τροφοδοσίας του.

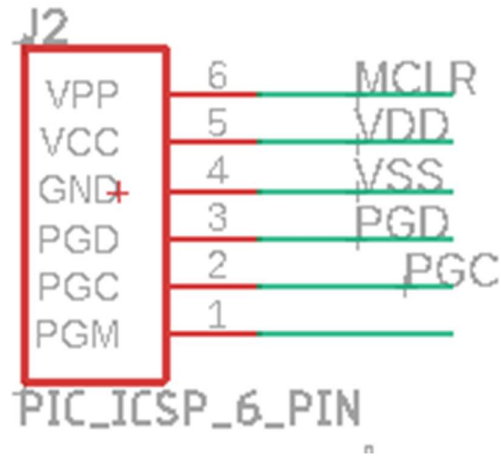
### 3.8.4 Speed Sensor Connector



Σχήμα 3.4 Διάταξη αισθητήρα ταχύτητας

Η διάταξη του αισθητήρα ταχύτητας πραγματοποιείται με την σύνδεση του RPM με την ακίδα του μικροελεγκτή PIC18F4550, που είναι υπεύθυνη για την λειτουργία Capture. Οι άλλες δύο συνδέσεις αντιστοιχούν στην τροφοδοσία του αισθητήρα

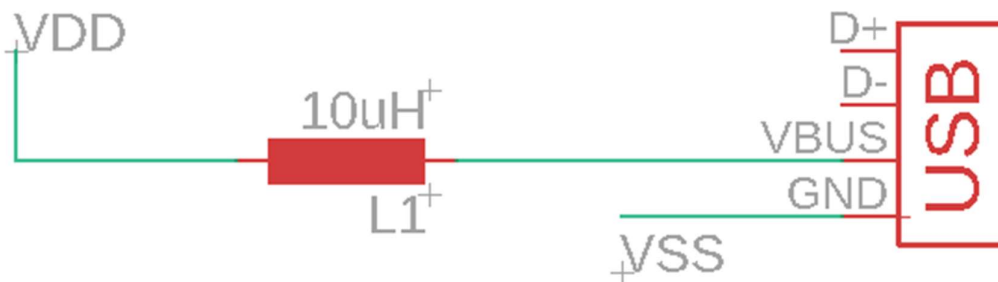
### 3.8.5 ICSP Διάταξη



Σχήμα 3.5 ICSP διάταξη προγραμματισμού του μικροελεγκτή

Ίσως η σημαντικότερη διάταξη που υπάρχει στο σχηματικό, είναι αυτή του κυκλώματος προγραμματιστή, δηλαδή αυτή που συνδέεται το Pickit 3 και γράφει στην μνήμη του μικροελεγκτή τον πηγαίο κώδικα, για να εκτελεστεί η εφαρμογή η οποία γράφτηκε στο περιβάλλον MPLAB IDE.

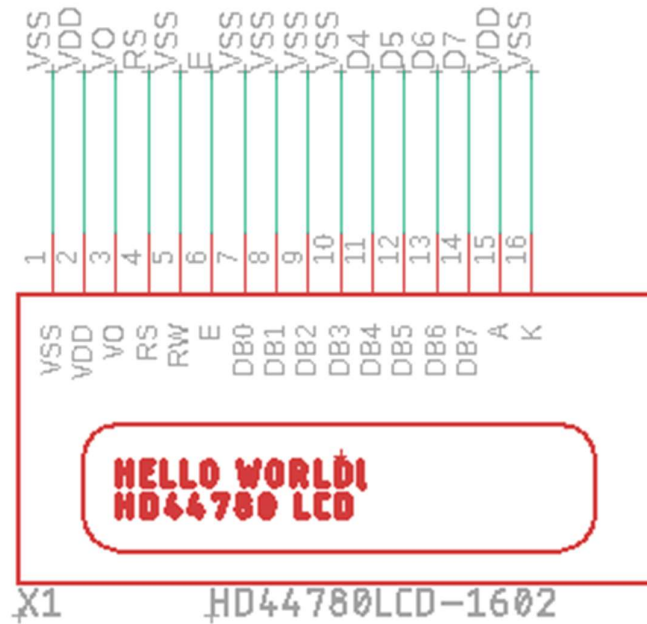
### 3.8.6 Σύνδεση Τροφοδοσίας



Σχήμα 3.6 Διάταξη Τροφοδοσίας USB

Η διάταξη τροφοδοσίας της πλακέτας, γίνεται με την σύνδεση ενός USB 2.0 καλωδίου, ικανό να παρέχει την τροφοδοσία που χρειάζεται το κύκλωμα για να λειτουργεί.

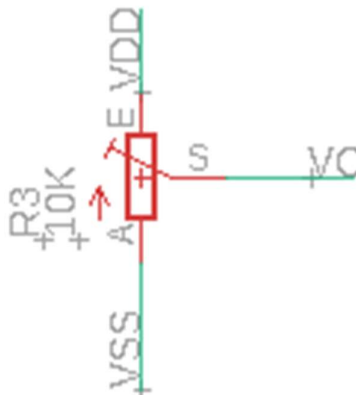
### 3.8.7 Οθόνη LCD 16x2



Σχήμα 3.7 Διάταξη οθόνης LCD 16x2

Η συνδεσμολογία της οθόνης, είναι με την μέθοδο των 4-bit, D4, D5, D6, D7 και τα υπόλοιπα στην γείωση. Με αυτή την μέθοδο, ο μικροελεγκτής έχει διαθέσιμες περισσότερες ακίδες για την ενσωμάτωση περαιτέρω λειτουργιών (Leds, Motors)

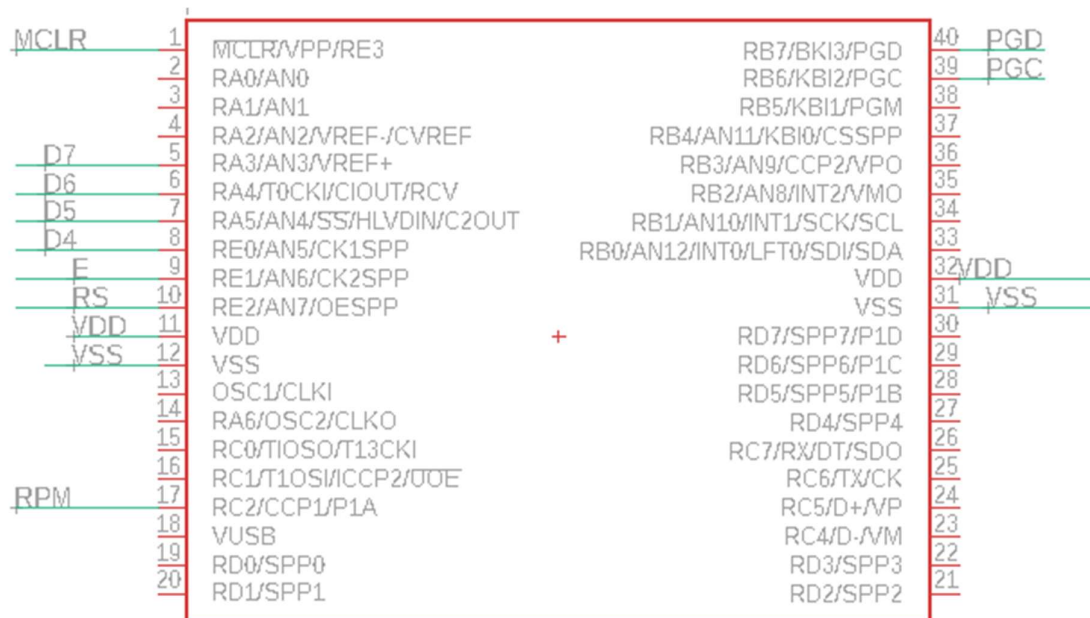
### 3.8.8 Ποτενσιόμετρο για την οθόνη



Σχήμα 3.8 Ποτενσιόμετρο 10κ

Για την ομαλή λειτουργία της οθόνης, απαιτείται ένα ποτενσιόμετρο για το contrast της, το οποίο τοποθετείται κοντά στην V<sub>o</sub> ακίδα της οθόνης.

### 3.8.9 Μικροελεγκτής PIC18F4550

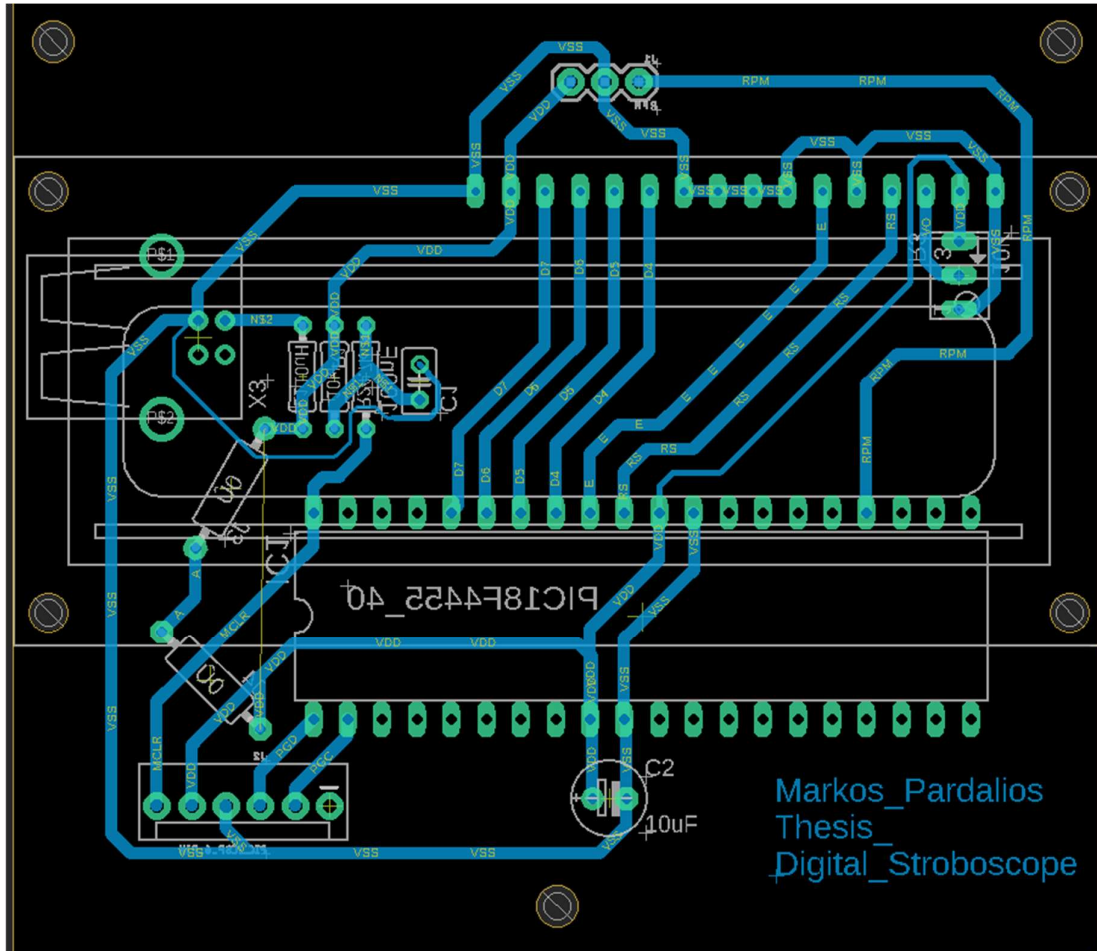


Σχήμα 3.9 PIC18F4550

Είναι ο εγκέφαλος του συστήματος, ο μικροελεγκτής υπεύθυνος για την λειτουργία της εφαρμογής, δηλαδή παίρνει σαν είσοδο το σήμα του αισθητήρα, υλοποιεί πράξεις στο εσωτερικό του και η έξοδος που παράγεται από τον μικροελεγκτή καταλήγει στην οθόνη LCD 16x2.

### 3.9 PCB Layout

Αφού αναλύθηκε εις βάθος το σχηματικό και όλα τα κομμάτια που το απαρτίζουν, το σχεδιαστικό πακέτο eagle παρέχει την δυνατότητα και σχεδιασμού του PCB layout, δηλαδή την πλακέτα που περιέχει όλες τις συνδέσεις, ενωμένες μεταξύ τους με τέτοιο τρόπο έτσι ώστε να μπορεί να περνάει τάση από όλα τα σημεία και να λειτουργεί η εφαρμογή. Παρακάτω (Εικόνα 3.2) φαίνεται το τελικό pcb layout, αυτό που χρησιμοποιήθηκε για την τύπωση του PCB. Τύπωμα του pcb είναι η ενσωμάτωση του διαγράμματος που σχεδιάστηκε στο eagle επάνω στην πλακέτα.



Εικόνα 3.2 PCB Layout

Το layout στήθηκε με τέτοιο τρόπο, έτσι ώστε να μην υπάρχει καθόλου παρεμβολή από κάποιο άλλο σήμα στην είσοδο της ακίδας RPM, όπως φαίνεται και στο σχήμα. Η είσοδος της ακίδας RPM, παίρνει το σήμα από τον αισθητήρα ταχύτητας, το οποίο μετά υπολογίζεται μέσα στον μικροελεγκτή με πράξεις σε πραγματικό χρόνο και βγαίνει το αποτέλεσμα των μετρήσεων στην οθόνη lcd 16x2. Το συνολικό μέγεθος της πλακέτας είναι 8x7 εκατοστά, δίνοντας την δυνατότητα στον χρήστη να μπορέσει να το μεταφέρει με σχετική ευκολία.

## 4. Μελέτη Στροφόμετρου

### 4.1 Εισαγωγή

Το στροφόμετρο, είναι μια συσκευή που μετράει την ταχύτητα ενός μηχανήματος. Αυτή η συσκευή συνήθως απεικονίζει την περιστροφική ταχύτητα σε περιστροφές ανά λεπτό (RPM) είτε σε ψηφιακή είτε σε αναλογική ένδειξη. Η πρώτη καταγεγραμμένη χρήση του ταχογράφου, έγινε από ένα μηχανικό που ονομαζόταν Dietrich Uhlhorn το 1817, όπου είναι ο υποτιθέμενος εφευρέτης του στροφόμετρου. Κάνεις δεν είναι σίγουρος για το ποιος έφτιαξε το στροφόμετρο, όμως βασίστηκε στην κεντροφόρο δύναμη, συγκρινόμενη με έναν φυγοκεντρικό ρυθμιστή, όπως φαίνεται στην Εικόνα 4.1. Στην εικόνα δείχνει μια αυτορρυθμιζόμενη συσκευή για να ελέγχει την ροή ενός υγρού. Το πρώτο στροφόμετρο χρησιμοποίησε την κεντροφόρος δύναμη από τις μπάλες οι οποίες κινούνταν και ήταν πιασμένες από έναν ενδείκτη ή από μία βελόνα για να δείξουν την ταχύτητα με την οποία κινούνταν. Αργότερα χρησιμοποιήθηκε για να μετράει την ταχύτητα των ατμοκινητήρων το 1840.

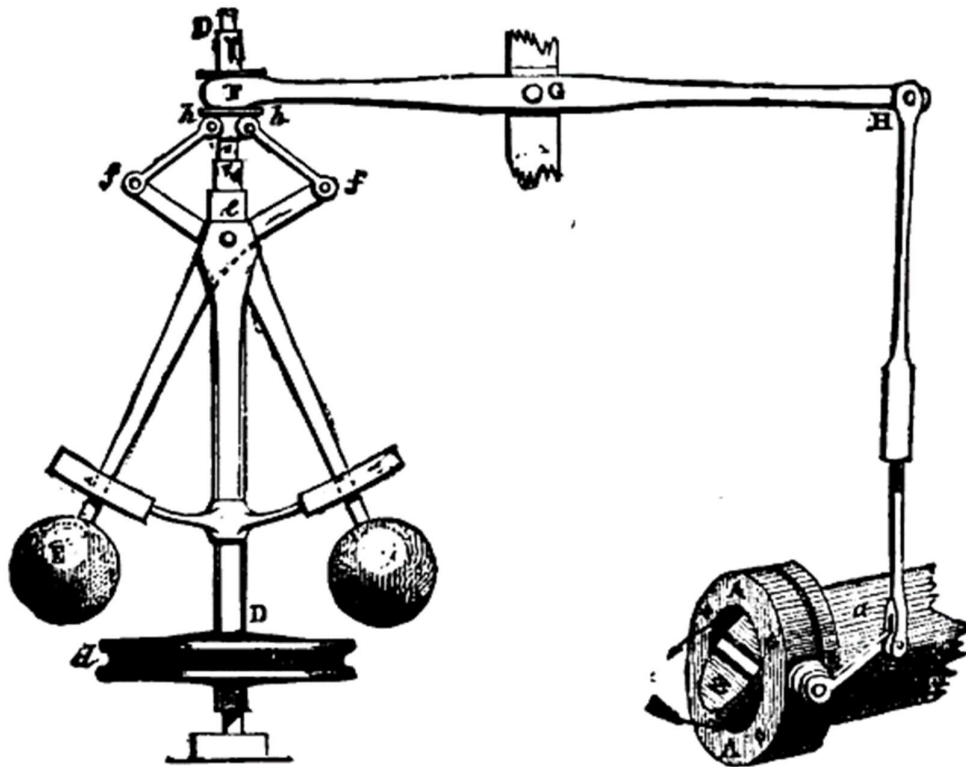


FIG. 4.—Governor and Throttle-Valve.

Εικόνα 4.1 Το πρώτο στροφόμετρο

Τα στροφόμετρα χρησιμοποιούνται στην σύγχρονη εποχή, για την μέτρηση της περιστροφικής ταχύτητας πολλών πραγμάτων. Είναι πλεονέκτημα να γνωρίζει κάποιος την διάρκεια χρόνου και την ταχύτητα ενός κινούμενου αντικειμένου για τον υπολογισμό της απόστασης ώστε να μπορεί να προγραμματιστεί η συντήρηση του για την αποφυγή φθοράς. Τα πιο κοινά στροφόμετρα,

χρησιμοποιούνται στα οχήματα για τον υπολογισμό της ταχύτητας και της απόστασης. Υπάρχουν πολλοί τρόποι για την μέτρηση της περιστροφικής ταχύτητας και σε αυτή την πτυχιακή εργασία θα παρουσιαστεί την μέθοδο ανάπτυξης ενός στροφόμετρου, και τα προβλήματα που αντιμετώπισε ο συγγραφέας για να το αναπτύξει. Η ταχύτητα της αντίστροφης της κίνησης θα ληφθεί υπόψη κατά την κατασκευή ενός ταχύμετρου.

## 4.2 Αρχικά Βήματα

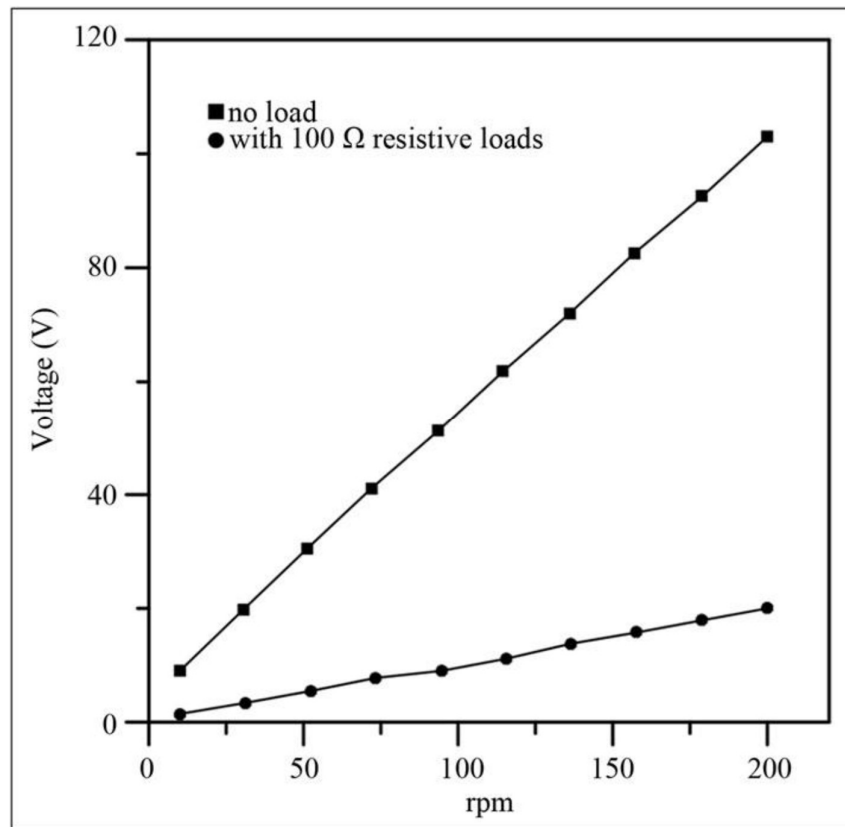
Πρωτίστως, πρέπει να απαντηθεί το ερώτημα που λέει «Τι είναι ένα στροφόμετρο;». Πως λειτουργεί; Τι μπορεί να πετύχει κάποιος με την ανάπτυξη του; Οι ιδέες και οι καλλιγραφικές νότες θα αναφερθούν αναλυτικά παρακάτω. Στην αγορά, υπάρχουν φαινομενικά ατελείωτα μοντέλα με λειτουργίες που μπορούν να αναπαραχθούν σε αυτό το έργο μαζί με πράγματα που μπορούν να προστεθούν για μια μοναδική πινελιά. Το πρότυπο μεταξύ των στροφόμετρων, μετά από μια γρήγορη αναζήτηση ως προς το τι κυκλοφορεί στην αγορά, είναι πως το 90% περιλαμβάνει οθόνη. Η οθόνη αυτή δείχνει στον χρήστη τα RPM μέγεθος που δείχνει πόσες φορές το κινούμενο αντικείμενο περιστρέφεται σε μια καθορισμένη χρονική περίοδο. Μερικά από τα στροφόμετρα περιλαμβάνουν έναν συνδυασμό μέτρησης επαφής και μη επαφής.

## 4.3 Μοντέρνες Τεχνολογίες

Υπάρχουν πολλοί τρόποι για να μετρήσει κάποιος την ταχύτητα. Κάποιες τεχνολογίες περιλαμβάνουν, όπως το στροφόμετρο που αναφέρθηκε παραπάνω, την μέτρηση της φυγόκεντρου δύναμης. Κάποια από τα μοντέρνα στροφόμετρα χρησιμοποιούν την δέσμη όπως laser και αισθητήρα, παρομοίως έναν οπτικό κωδικοποιητή, κάποια στροφόμετρα θα χρησιμοποιούν την επαφή και θα προκαλούν κίνηση σε έναν κινητήρα για μέτρηση, και υπάρχουν πολλοί τρόποι να μετρήσει κάποιος, όπως η σύνδεση μιας συσκευής μέτρησης απευθείας στο μπουζί ενός αυτοκινήτου για την μέτρηση του εμβόλου σε RPM.

## 4.4 Γεννήτρια

Είναι κοινό ότι, όταν εφαρμόζεται τάση στους ακροδέκτες ενός κινητήρα, ρέει ρεύμα και ο κινητήρας περιστρέφεται. Διάφοροι κινητήρες θα παράγουν διαφορετικά αποτελέσματα, για την απεικόνιση της τάσης σε συνάρτηση με το χρόνο σύμφωνα με την παρακάτω εικόνα. Από τα συμπεράσματα που βγάζει κάποιος, μελετώντας την Σχήμα 4.1 θα παρατηρήσει πως όσο μεγαλύτερη τάση έχει ο κινητήρας, τόσο μεγαλύτερη θα είναι και η ταχύτητα του.



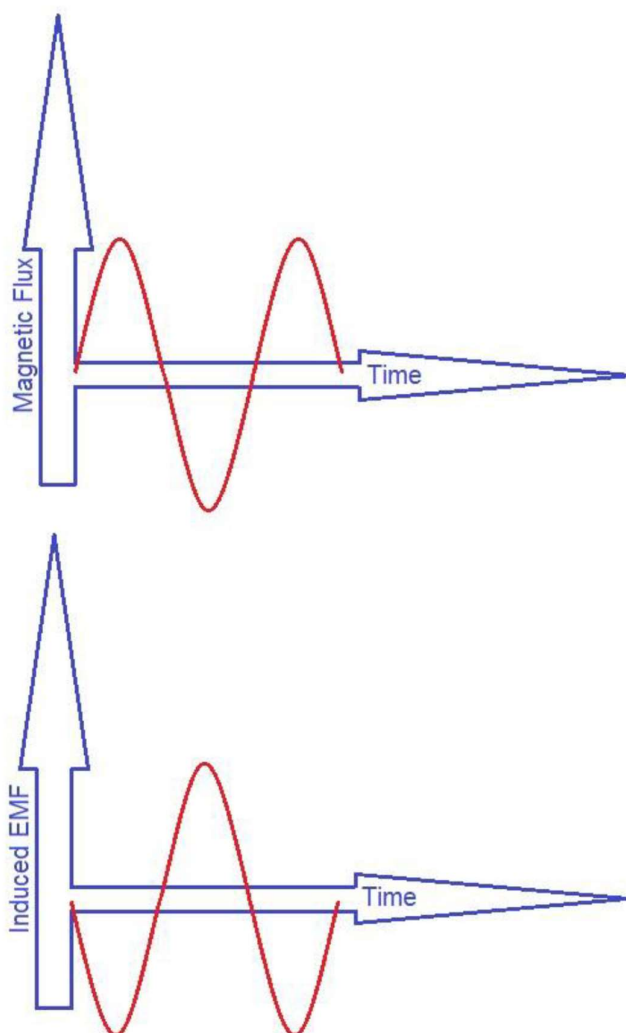
Σχήμα 4.1 Διάγραμμα τάσης- RPM

Για την δημιουργία μιας γεννήτριας τάσης ισχύει το αντίθετο: Όταν εφαρμόζεται μια δύναμη σε έναν κινητήρα, παράγεται μια τάση στα άκρα του. Αυτό είναι στην ουσία ο νόμος του Faraday. Όταν υπάρχει ένα πηνίο γύρω από ένα μεταβαλλόμενο μαγνητικό πεδίο, το πηνίο παράγει ρεύμα, το οποίο ονομάζεται ηλεκτρομαγνητική επαγωγή και υπάρχουν πολλοί τρόποι για να παραχθεί αυτό το αποτέλεσμα. Η τάση μπορεί να δημιουργηθεί αλλάζοντας την ροή μέσα σε ένα πηνίο, μετακινώντας έναν μαγνήτη προς ένα πηνίο ή περιστρέφοντας ένα πηνίο μέσα σε ένα μαγνητικό πεδίο. Ένα παράδειγμα ενός στροφόμετρου φαίνεται στην παρακάτω εικόνα (Εικόνα 4.2), με το σημείο επαφής στην κορυφή του.



Εικόνα 4.2 Στροφόμετρο επαφής

Ενώ αυτή η μέθοδος είναι πολύ εύκολη για τον χρήστη και σχεδόν πάντα βγάξει αποτελέσματα υπάρχουν μερικά μειονεκτήματα με την μέθοδο υλοποίησής του. Ένα από τα μειονεκτήματα είναι πως η έξοδος της γεννήτριας (DC ή AC), όταν χρησιμοποιείται σε έναν απλό κινητήρα, σε κάποιο σημείο το πηνίο θα είναι παράλληλο με τον μαγνήτη ο οποίος δεν θα παράγει τάση όπως φαίνεται στο παρακάτω σχήμα (Σχήμα 4.2). Αυτό μπορεί να λυθεί με την χρήση ενός πυκνωτή για να κατεβάσει το κατώφλι της τάσης, ώστε να την κάνει να φαίνεται σαν DC. Δυστυχώς όμως σε χαμηλές ταχύτητες δεν είναι εφικτό, καθώς δεν είναι αποτελεσματική λόγω της απώλειας φόρτισης του πυκνωτή. Υπάρχει μεγάλο σφάλμα που σχετίζεται με αυτό τον τύπο μέτρησης. Μόνο εάν είναι καλά βαθμονομημένο ή εάν η ακριβής εξίσωση του ηλεκτροκινητήρα είναι γνωστή με τις σταθερές (ισχύς μαγνητικού πεδίου κτλ), θα είναι δυνατά ακριβή αποτελέσματα. Ένα άλλο μειονέκτημα είναι ότι αυτοί οι κινητήρες ή γεννήτριες θα έχουν μεγαλύτερη φθορά στα εξαρτήματα σε σύγκριση με άλλες τεχνολογίες είτε λόγω των φυσικών μερών επαφής, είτε των μεγαλύτερων τάσεων και ρευμάτων που προκαλούνται στην γεννήτρια. Υπάρχουν επίσης ανησυχίες για την ασφάλεια κατά την αντιμετώπιση πολύ μεγάλων μηχανημάτων καθώς μπορεί να είναι επικίνδυνο να διαβαστεί μια τάση από επαφή.



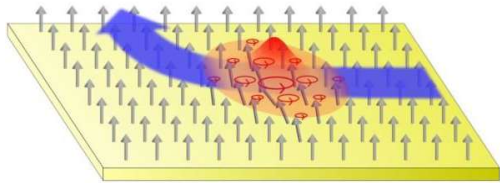
Σχήμα 4.2 Αντιπαράβολή EMF και μαγνητική ροή με την πάροδο του χρόνου

#### 4.5 Το φαινόμενο Hall

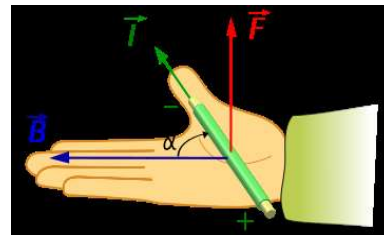
Το φαινόμενο του Hall ανακαλύφθηκε από τον Edwin Hall το 1879 και ήταν ένα μικρό εφέ που παρήγαγε η συσκευή που χρησιμοποιούσε 18 χρόνια πριν ανακαλυφθεί το ηλεκτρόνιο. Ο αισθητήρας Hall λειτουργεί περνώντας ρεύμα μέσω μιας διηλεκτρικής πλάκας και εν παρουσία μαγνητικού πεδίου θα παράγει τάση φαινομένου Hall. Τα ηλεκτρόνια θα ρέουν από την μία πλευρά της πλάκας στην άλλη όταν υπάρχει διαφορά δυναμικού στα άκρα της. Με την παρουσία μαγνητικού πεδίου, τα ηλεκτρόνια θα εκτραπούν στη μία πλευρά της πλάκας κάθετα προς την αρχική ροή τους. Στην Εικόνα 4.3 φαίνεται η ροή ηλεκτρονίων που εμφανίζεται ως το παχύ μπλε βέλος που ρέει κανονικά προς τα αριστερά, τα μικρότερα βέλη που βγαίνουν από την πλάκα είναι το μαγνητικό πεδίο και η δύναμη είναι προς το υλικό. Εξηγείται ξανά με τον κανόνα του δεξιού χεριού (Εικόνα 4.4) εάν ο αναγνώστης αυτής της διατριβής κρατά το χέρι τους με τη δεξιά παλάμη προς το χαρτί, τα δάχτυλα είναι ίσια και ο αντίχειρας είναι κάθετος στα δάχτυλα: τα δάχτυλα είναι το μαγνητικό πεδίο, ο αντίχειρας είναι το ρεύμα και έξω από την παλάμη είναι η δύναμη. Αυτό περιγράφει τι συμβαίνει στα ηλεκτρόνια που ρέουν στην πλάκα. Καθώς τα ηλεκτρόνια ρέουν προς τη μία πλευρά της πλάκας και οι σπές ηλεκτρονίων ρέουν προς την

άλλη πλευρά, δημιουργώντας έτσι δυναμικό τάσης από την μία πλευρά της πλάκας στην άλλη και αυτό μπορεί να μετρηθεί ως τάση Hall.

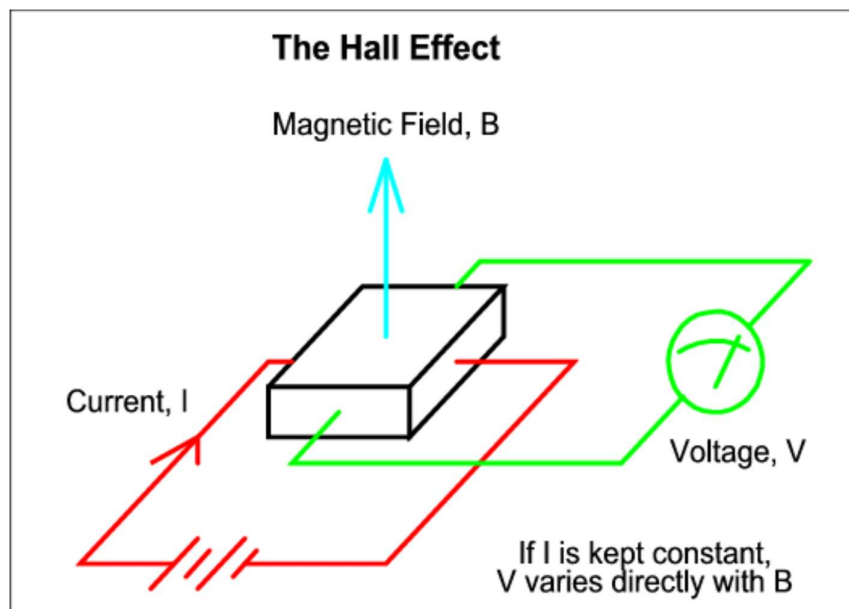
Η τάση Hall προκαλείται μόνο όταν υπάρχει μαγνητικό πεδίο και εάν για παράδειγμα ένας μαγνήτης ήταν συνδεδεμένος με έναν κινητήρα, τότε η περιστροφή του θα μπορούσε να υπολογιστεί ως ταχύτητα με τις κορυφές της τάσης Hall. Ένα απλό κύκλωμα του φαινομένου Hall φαίνεται στην Εικόνα 4.5. Ομοίως με την μέθοδο της γεννήτριας απαιτείται ένας αισθητήρας Hall πολύ κοντά στο μηχανήμα και εάν είναι επικίνδυνο, τότε είναι μειονέκτημα. Σε αντίθεση με την γεννήτρια, η ακίδα τάσης που παράγεται είναι ισχυρή( κατά την χρήση ορισμένων μεθόδων που αναφέρονται παρακάτω) και δεν απαιτούνται σταθερές επειδή το μαγνητικό πεδίο είτε θα είναι παρών είτε όχι.



Εικόνα 4.3 Δύναμη που ασκείται στην κίνηση Εικόνα



Εικόνα 4.4 Ο νόμος του δεξιού χεριού

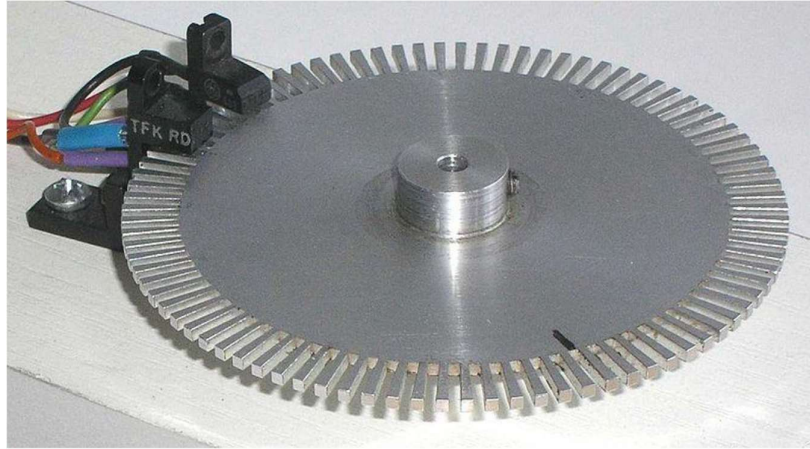


Εικόνα 4.5 Η τάση που προκάλεσε το φαινόμενο του Hall

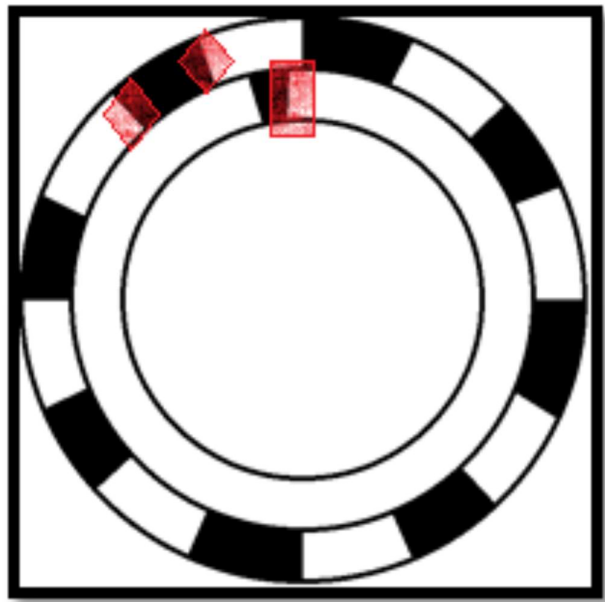
#### 4.6 Οπτικός Κωδικοποιητής (Optical Encoder)

Ένας οπτικός κωδικοποιητής είναι μία από τις καλύτερες μεθόδους για τον υπολογισμό της ταχύτητας ενός κινητήρα. Μπορεί να τοποθετηθεί σε περιστρεφόμενο μηχανήμα μακριά από τους χρήστες και επομένως είναι ασφαλέστερο. Δυστυχώς, επειδή χτίστηκε πάνω ή δίπλα στο μηχανήμα δεν είναι φορητό. Μπορούν να χρησιμοποιηθούν εγκοπές σε μία πλάκα που περιστρέφεται με τον κινητήρα (στον

άξονα) για μέτρηση της ταχύτητας. Κάθε οπτικός κωδικοποιητής έχει μια πηγή φωτός και ένα μέσο ανίχνευσης όπως φαίνεται στην Εικόνα 4.6.



Εικόνα 4.6 *Optical Encoder*



Σχήμα 4.3 *Ακολουθία του τροχού του οπτικού κωδικοποιητή*

Στο Σχήμα 4.3 απεικονίζεται ο δίσκος κωδικοποιητής που έχει εγκοπές που του επιτρέπουν να περάσει το φως και όπου το υλικό δεν είναι κομμένο, θα εμποδίσει τη ροή του φωτός. Το φως που διέρχεται θα ανιχνευθεί από έναν αισθητήρα και μπορεί να θεωρηθεί δυαδικό ως 1 και όταν το φως δεν περνάει μπορεί να θεωρηθεί ως 0. Όσο περισσότερες εγκοπές στο δίσκο τόσο πιο πολλά τα αποτελέσματα ακόμη και σε μία υψηλή ταχύτητα. Από ένα σημείο όμως και μετά στις υψηλότερες ταχύτητες ο ανιχνευτής μπορεί να μην πάρει το φως αρκετά γρήγορα για να δώσει μια καλή άκρη ανόδου, καθώς

όλοι οι αισθητήρες έχουν ελάχιστο χρόνο ανόδου. Επιπλέον, οι οπτικοί κωδικοποιητές είναι ευαίσθητοι σε βρωμιά ή λάδια που εμποδίζουν το φως που πηγαίνει στον ανιχνευτή καθώς επίσης και το φως του περιβάλλοντος χώρου. Απαιτούνται περισσότεροι από ένας ανιχνευτές για τον υπολογισμό της κατεύθυνσης περιστροφής και εκτός εάν βαθμονομηθούν προγραμματιστούν για να μετρήσουν τις εγκοπές για μια πλήρη στροφή απαιτείται ένας πρόσθετος αισθητήρας ως δείκτης ( που είναι μια εγκοπή για τη μέτρηση μιας πλήρους περιστροφής). Για έναν πολύ λειτουργικό κωδικοποιητή, 3 αισθητήρες μπορούν να χρησιμοποιηθούν για την μέτρηση των στροφών και της κατεύθυνσης. Ο τρόπος ρύθμισης τους φαίνεται στο ίδιο σχήμα με τα κόκκινα τετράγωνα όπου θα τοποθετηθούν οι αισθητήρες και όπου το φως και οι αισθητήρες αλληλοσυμπληρώνονται. Η είσοδος και η έξοδος σε δυαδικό ή υψηλό και χαμηλό εμφανίζεται στο Σχήμα 4.4.

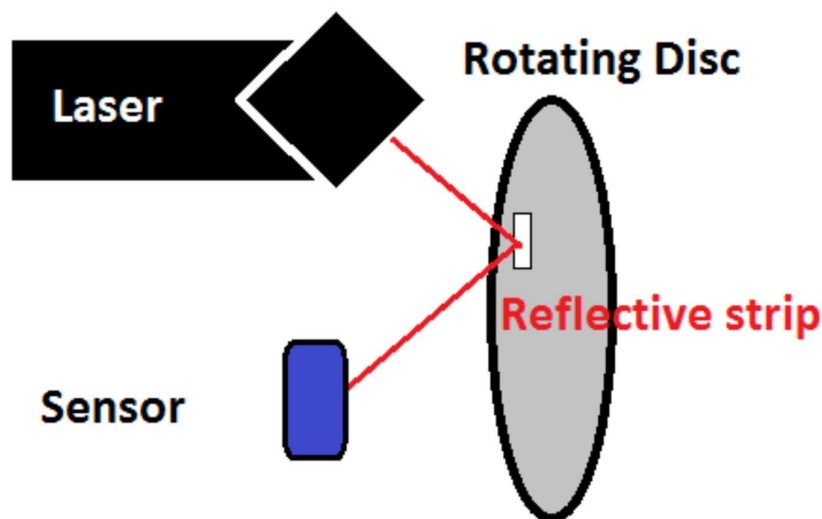


Σχήμα 4.4 Αλληλουχία δεδομένων

Όπως φαίνεται στο παραπάνω σχήμα, όταν ανιχνεύεται το φως, θα καταγραφεί πρώτα είτε στο A είτε στο B. Στο απεικονιζόμενο παράδειγμα, εάν ο κινητήρας υποτίθεται ότι περιστρέφει τον δίσκο κωδικοποίησης προς τη φορά των δεικτών του ρολογιού, ο αισθητήρας A θα καταγράψει πρώτα και στη συνέχεια ο B θα υποδεικνύει την δεξιόστροφη κατεύθυνση. Αντίστροφα όταν ο B θα καταγράψει πρώτα και μετά ο A, τότε θεωρείται ότι πηγαίνει στην αντίστροφη κατεύθυνση η αριστερόστροφη. Ο τρίτος ανιχνευτής ή δείκτης σε αυτό το παράδειγμα δεν θα κάνει διάκριση μεταξύ εμπρός ή αντίστροφης κατεύθυνσης, καθώς θα λάβει έναν παλμό μόλις ο δίσκος κάνει πλήρη περιστροφή.

#### 4.7 Laser και φωτοδίοδος

Όπως ο οπτικός κωδικοποιητής, αυτή η μέθοδος χρησιμοποιεί το φως. Χρησιμοποιείται ένα λέιζερ για την παροχή μιας συγκεντρωμένης δέσμης φωτός που θα αντανάκλατε πάνω σε μια ανακλαστική λωρίδα ή έναν καθρέφτη, σε έναν αισθητήρα που θα παρέχει ένα άκρο ανόδου για τη μέτρηση της ταχύτητας ενός αντικειμένου. Η αρχή της λειτουργίας φαίνεται στο Σχήμα 4.5. Σε κάποιο σημείο καθώς η συσκευή περιστρέφεται, που μπορεί να είναι είτε ένας δίσκος στο άκρο ενός άξονα, είτε ο ίδιος ο άξονας, μια ακτίνα θα αντανάκλα στον αισθητήρα.

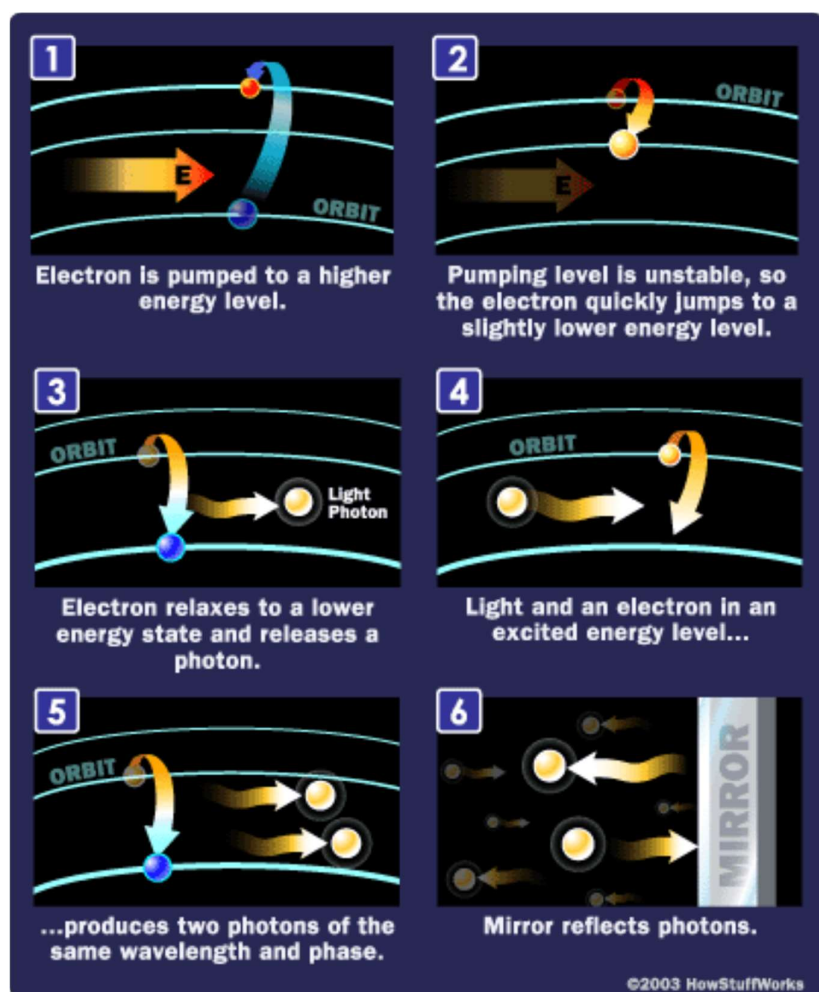


Σχήμα 4.5 Αρχή λειτουργίας λέιζερ και ανακλαστικής ταινίας

Αυτή η μέθοδος υπόκειται επίσης (όπως ο οπτικός κωδικοποιητής) σε απώλειες φωτός που φτάνουν στον αισθητήρα λόγω βρώμικου φακού ή φωτοδιόδου, αλλά προσφέρει ασφάλεια καθώς μπορεί να μετρηθεί από απόσταση. Δυστυχώς η μέτρηση της μεθόδου αυτής, μπορεί να είναι πολύ δαπανηρή και υπάρχει ανάγκη για περισσότερες από μία φωτοδιόδους. Και πάλι, όπως ο οπτικός κωδικοποιητής, το φως περιβάλλοντος μπορεί να αποδειχθεί πρόβλημα καθώς μπορεί να ανιχνεύσει ψευδείς ή περιττές αναγνώσεις.

#### 4.8 Laser

Η εξήγηση του πως λειτουργούν πλήρως τα laser είναι πέρα από το πεδίο αυτής της πτυχιακής, άλλα θα εξηγηθεί εν συντομία. Τα λέιζερ διαφέρουν από άλλες πηγές φωτός λόγω της συνοχής τους (πόσο στενή είναι η εκπεμπόμενη δέσμη φωτός). Το φως μπορεί να δημιουργηθεί από ένα laser με μερικούς τρόπους, όπως η συγκέντρωση μιας πηγής φωτός με ένα φακό όπως ένα LED ή η μεταφορά ενέργειας σε άτομα, όπως ένα αέριο που εκπέμπει φως. Η διαδικασία φαίνεται στην Εικόνα 4.7. Η ενέργεια θα προστεθεί σε ένα ηλεκτρόνιο αυξάνοντας το επίπεδο ενέργειας των ηλεκτρονίων του (διεγερμένο), γίνετε ασταθές στο υψηλότερο επίπεδο ενέργειας και για να επανέλθει στο αρχικό του επίπεδο, απελευθερώνει ενέργεια με τη μορφή φωτός (φωτόνιο). Όταν προστίθεται φως σε ένα διεγερμένο ηλεκτρόνιο παράγει δύο φωτόνια του ίδιου μήκους κύματος φωτός και φάσης. Αυτή η διαδικασία περιέχεται μέσα σε ένα σωλήνα με έναν καθρέφτη στο ένα άκρο για τη δέσμευση των ηλεκτρονίων σε μια συγκεκριμένη κατεύθυνση και θα ρέουν μέσω ενός φακού ή μιας σειράς φακών για τη συγκέντρωση του φωτός.



Εικόνα 4.7 Περιγραφή διέγερσης αερίου laser

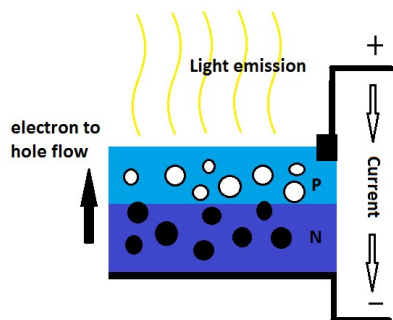
Λόγω της φύσης των laser μπορεί να είναι επικίνδυνα και πρέπει να ληφθούν υπόψη κατά τον σχεδιασμό αυτού του έργου για να διασφαλιστεί η ασφάλεια εάν επιλεγεί αυτή η μέθοδος. Τα laser ταξινομούνται σε τέσσερις γενικές περιοχές και είναι το δυναμικό τους για ζημία που τα διαχωρίζει. Ένα laser πρέπει πάντα να φέρει ετικέτα στο χώρο εργασίας με μία από τις τέσσερις κατηγορίες.

- Class I: Τα laser κατηγορίας I δεν έχουν την δυνατότητα να εκπέμπουν ακτινοβολία laser, επομένως δεν πρέπει να αποδειχθούν επικίνδυνα.
- Class IA: Αυτή η κατηγορία είναι ένα ειδικό laser χαμηλής τάξης που δεν έχει σχεδιαστεί, αλλά μπορεί ακόμα να χρησιμοποιηθεί σε μέρη όπως οι σαρωτές σούπερ μάρκετ, το μέγιστο όριο ισχύος που ορίζεται στα 4mW
- Class II: Laser χαμηλής ισχύος που είναι πάνω από την κατηγορία II και εκτός αν τα κοιτάζει ένα άτομο μπορεί να πάθει ζημιά
- Class IIIA: Laser μεγέθους πέννας που μπορεί να είναι επικίνδυνα και κυμαίνονται μεταξύ 1-5mW
- Class IIIB: Μεσαίας ισχύος laser

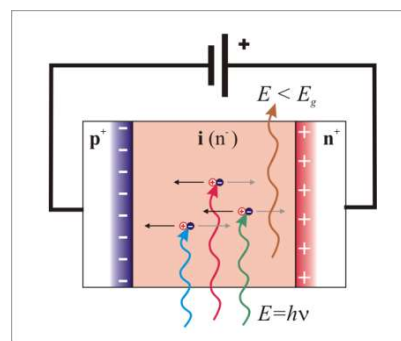
- Class IV: Laser πολύ υψηλής ισχύος περίπου στα 500mW. Πολύ επικίνδυνα και χρειάζονται προσεκτικό χειρισμό καθώς μπορούν να προκαλέσουν ζημιά, ακόμα και να προκαλέσουν φωτιά.

#### 4.9 Φωτοдиодος

Ο τρόπος με τον οποίο λειτουργούν τα LED ή οι διόδους εκπομπής φωτός είναι η διέλευση ενός ρεύματος (ή ηλεκτρονίων) μέσω αυτού που είναι γνωστή ως διασταύρωση N και P. Ο τύπος N είναι το αρνητικό υλικό ημιαγωγού και το P είναι το θετικό υλικό. Το υλικό (πυρίτιο) μπορεί να κολλήσει είτε δίνοντας του ελεύθερα ηλεκτρόνια (ή επιπλέον ηλεκτρόνια) είτε μπορεί να προσθέσει τρύπες ή κενά όπου τα ηλεκτρόνια μπορούν να ρέουν επειδή το πυρίτιο έχει 4 ηλεκτρόνια και συνδέεται τέλεια με τα άλλα άτομα που δεν θα επιτρέψει τη ροή ρεύματος. Το υλικό τύπου N έχει επιπλέον ηλεκτρόνια και επιτρέπει τη ροή ελεύθερων ή επιπλέον ηλεκτρονίων σε θετικά φορτισμένη περιοχή. Το υλικό τύπου P έχει επιπλέον τρύπες και είναι πιο θετικό. Όταν αυτό το υλικό τύπου P διεγείρεται με ρεύμα, τότε τα ηλεκτρόνια θα ρέουν προς τις οπές, καθώς ταξιδεύει προς μια τρύπα που μεταφέρει ενέργεια και για να γεμίσει αυτή η τρύπα πρέπει να απελευθερώσει οποιαδήποτε επιπλέον ενέργεια με τη μορφή φωτός. Αυτή η διαδικασία περιγράφεται στην Εικόνα 4.8.



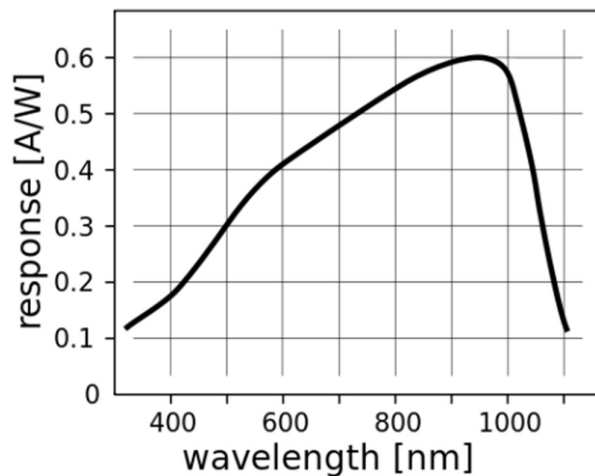
Εικόνα 4.8 Δίοδος LED εκπέμπει φως



Εικόνα 4.9 Δίοδος PIN απορροφάει φως

Στην περίπτωση μιας φωτοδιόδου, γνωστή ως δίοδος PIN, θα απορροφήσει φως και θα δημιουργήσει μια φόρτιση, σύμφωνα με την Εικόνα 4.9. Αυτή η ικανότητα προέρχεται από την περιοχή  $\langle I \rangle$  στη δίοδο PIN.

Αυτή η ενδογενής περιοχή είναι ανεμπόδιστη και οι δύο οπές και τα ηλεκτρόνια μπορούν να κινηθούν σε αυτό το υλικό. Το φως που απορροφάτε στην ενδογενή περιοχή μπορεί να θεωρηθεί παρόμοιο με τα laser του αερίου τους. Τα φωτόνια απορροφούνται στο υλικό και τα ηλεκτρόνια διεγείρονται και παράγουν ρεύμα. Αυτό είναι γνωστό ως το φωτοβολταϊκό φαινόμενο. Οι φωτοδιόδους δεν κάνουν διαχωρισμό με διαφορετικά μήκη κύματος φωτός και μπορούν να προκαλέσουν ανεπιθύμητες μετρήσεις. Στο σχήμα 15 φαίνεται μια τυπική απόκριση από μια φωτοδίοδο ως προς την τάση ή το ρεύμα έναντι του μήκους κύματος του φωτός που ταξιδεύει στον αισθητήρα.



Σχήμα 4.6 Απόκριση φωτοδιόδου στο φάσμα του μήκους κύματος

Για αυτή την εργασία τα φωτοτρανζίστορ έχουν ληφθεί υπόψη, άλλα οι φωτοδιόδοι είναι πιο ελκυστικές με τα γραμμικά χαρακτηριστικά τους. Αυτό σημαίνει ότι όσο περισσότερο φως έρχεται στη δίοδο, τόσο μεγαλύτερο ρεύμα ή τάση θα παράγεται σε έναν πιο γραμμικό τύπο άμεσης συσχέτισης. Τα μη γραμμικά φωτοτρανζίστορ θα παράγουν μια πιο περίπλοκη έξοδος εκθετικής σχέσης ανάλογα με την είσοδο.



Εικόνα 4.10 Κοινό στροφόμετρο του εμπορίου

## 5. Στήσιμο εφαρμογής

### 5.1 Εισαγωγή

Για να μπορέσουν να επιλεχθούν τα κατάλληλα μέλη αυτής της εφαρμογής, πρέπει να ληφθούν υπόψη παράγοντες όπως το κόστος, η απόδοση και η καταλληλότερη μέθοδος για ακριβή μέτρηση του στροφόμετρου. Πριν από την οποιαδήποτε μελέτη που αφορά την επιλογή καθετί υλικού (hardware) της εφαρμογής, απαιτείται η παραγωγή κώδικα, ικανού να μοντελοποιήσει το πρόβλημα, και να μπορέσει να παράξει το επιθυμητό αποτέλεσμα, το οποίο είναι η μέτρηση στροφών. Αφού υλοποιηθεί το κομμάτι του κώδικα, ο μηχανικός έχει έτοιμο, ένα από τα πολλά κομμάτια που απαιτούνται για την υλοποίηση του ψηφιακού στροφόμετρου.

### 5.2 Υλοποίηση στην C

Για την υλοποίηση του προγράμματος, χρησιμοποιήθηκε η γλώσσα C, εν μέσω του περιβάλλοντος MPLAB v8.92. Η μέθοδος που χρησιμοποιήθηκε για την σύλληψη και ακριβής μέτρηση των στροφών, είναι η λεγόμενη Capture Mode, από το ενσωματωμένο περιφερειακό του μικροελεγκτή PIC18F4550. Σύμφωνα με το φύλλο δεδομένων (datasheet), που βρίσκεται στην σελίδα 143, περιγράφει την μέθοδο που πρέπει να χρησιμοποιήσει ο προγραμματιστής, για να μπορέσει να χρησιμοποιήσει την μέθοδο αυτή. Ουσιαστικά η μέθοδος Capture, συλλαμβάνει ένα event που συμβαίνει σε κάποια συγκεκριμένη χρονική στιγμή. Η μέθοδος αυτή, για την εφαρμογή, είναι η πιο αποδοτική, ο μικροελεγκτής καταναλώνει λίγους πόρους, και το μέγεθος των στροφών που επιλέχθηκαν για την εφαρμογή να μετρηθούν, είναι ENTOΣ των ορίων του μικροελεγκτή. Ο πηγαίος κώδικας του προγράμματος που φορτώθηκε στον μικροελεγκτή γράφτηκε στην γλώσσα προγραμματισμού C.[31] Η C είναι μια διαδικαστική γλώσσα προγραμματισμού γενικής χρήσης, η οποία αναπτύχθηκε αρχικά, μεταξύ του 1969 και του 1973, από τον Ντένις Ρίτσι στα εργαστήρια AT&T Bell Labs για να χρησιμοποιηθεί για την ανάπτυξη του λειτουργικού συστήματος UNIX. Όπως οι περισσότερες διαδικαστικές γλώσσες προγραμματισμού που ακολουθούν την παράδοση της ALGOL, η C έχει δυνατότητες δομημένου προγραμματισμού και επιτρέπει τη χρήση αναδρομής (αλλά όχι και εμφωλευμένων συναρτήσεων), ενώ, ο στατικός ορισμός του τύπου των μεταβλητών που επιβάλλει, προλαμβάνει πολλά σφάλματα κατά την χρήση τους. Ο σχεδιασμός της περιλαμβάνει δομές που μεταφράζονται αποδοτικά σε τυπικές εντολές μηχανής (machine instructions) και εξ αιτίας αυτού χρησιμοποιείται συχνά σε εφαρμογές που παλιότερα γράφονταν σε συμβολική γλώσσα (assembly language). Αυτό ακριβώς το χαρακτηριστικό της, που έχει σαν συνέπεια και την αυξημένη ταχύτητα εκτέλεσης των εφαρμογών που γράφονται σε αυτή, καθώς και το γεγονός ότι είναι διαθέσιμη στα περισσότερα σημερινά λειτουργικά συστήματα, συνέβαλε κατά πολύ στην καθιέρωση της και την χρήση της για ανάπτυξη λειτουργικών συστημάτων και λοιπών προγραμμάτων συστήματος (System Software), αλλά και απλών εφαρμογών. Η C συγκαταλέγεται πλέον στις πιο ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού όλων των εποχών και πολλές νεότερες γλώσσες έχουν επηρεαστεί άμεσα ή έμμεσα από αυτήν. Η C είναι μια σχετικά μινιμαλιστική γλώσσα προγραμματισμού. Ανάμεσα στους σχεδιαστικούς στόχους που έπρεπε να καλύψει η γλώσσα περιλαμβανόταν το ότι θα μπορούσε να μεταγλωττιστεί άμεσα με τη χρήση μεταγλωττιστή ενός περάσματος (single-pass compiler), με άλλα λόγια, ότι θα απαιτούνταν μόνο ένας μικρός αριθμός από εντολές σε γλώσσα μηχανής για κάθε βασικό στοιχείο της, χωρίς εκτεταμένη υποστήριξη στον χρόνο εκτέλεσης. Ως αποτέλεσμα, είναι δυνατό να γραφτεί κώδικας σε C σε χαμηλό επίπεδο προγραμματισμού με ακρίβεια ανάλογη της συμβολικής γλώσσας, στην πραγματικότητα η C ορισμένες φορές αποκαλείται «συμβολική γλώσσα υψηλού επιπέδου» («high-level assembly») ή «φορητή συμβολική γλώσσα» («portable assembly»).

### 5.3 Ο προγραμματισμός του μικροελεγκτή

Συγκεκριμένα σε αυτό το κεφάλαιο θα γίνει μια ανάλυση ως προς τα βήματα που ακολούθησε ο συγγραφέας για την υλοποίηση της εφαρμογής. Ένα πολύ βασικό βήμα, είναι ο προγραμματισμός του μικροελεγκτή που περιλαμβάνει το περιβάλλον προγραμματισμού που χρησιμοποιήθηκε για τον προγραμματισμό του μικροελεγκτή, την γλώσσα προγραμματισμού και αναλυτικά τον πηγαίο κώδικα που γράφτηκε και χρησιμοποιήθηκε.

### 5.4 Το περιβάλλον προγραμματισμού MPLAB

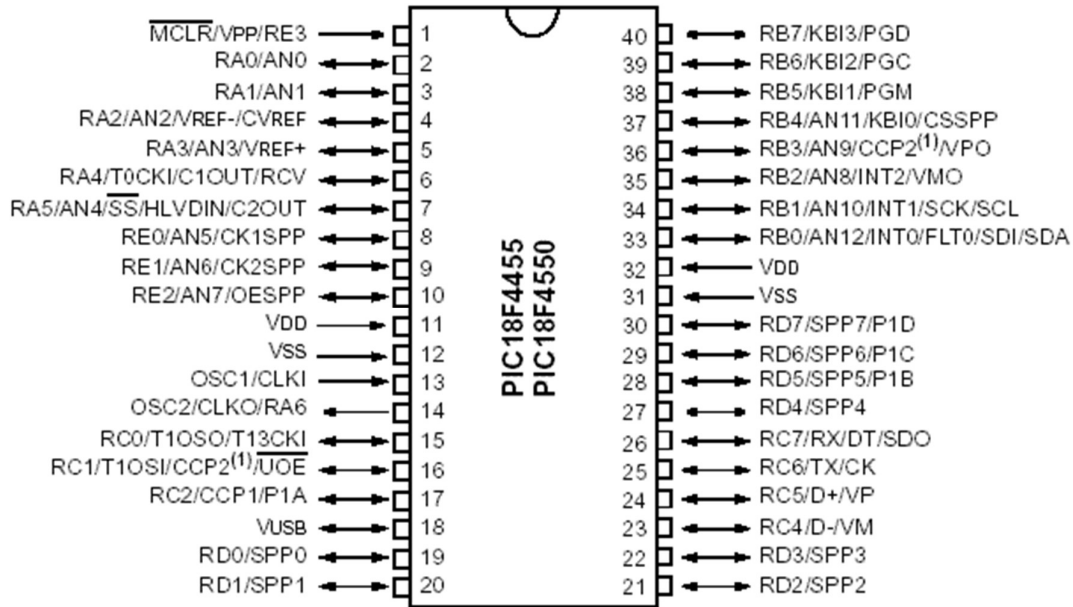
Για την υλοποίηση του πηγαίου κώδικα χρησιμοποιήθηκε το πρόγραμμα MPLAB το οποίο είναι ένα πρόγραμμα εξομοίωσης που διατίθεται δωρεάν από την εταιρία Microchip. Το MPLAB είναι στην πραγματικότητα ένα πολλαπλό εργαλείο ανάπτυξης στο οποίο εμπεριέχονται τα προγράμματα:

- Assembler για τη μετάφραση του αρχείου προγράμματος .asm σε γλώσσα μηχανής
- Μετατροπείς του αρχείου στην κατάλληλη μορφή .hex για τον προγραμματισμό του μικροελεγκτή
- Εξομοιωτής προγραμματισμού της μνήμης ROM του μικροελεγκτή

Η διαδικασία προγραμματισμού μέσω του MPLAB ξεκινάει με την δημιουργία ενός νέου project το οποίο θα αποθηκευτεί σε έναν ξεχωριστό φάκελο που θα δημιουργήσουμε στον υπολογιστή. Μετά από μερικά ακόμα βήματα που απαιτούνται φτάνουμε στο σημείο να δημιουργήσουμε και να εμφανίσουμε ένα άδειο αρχείο στο οποίο εμείς θα γράψουμε τον πηγαίο κώδικα. Αφού γραφτεί το πρόγραμμα θα πρέπει να σωθεί και κατόπιν να μεταφραστεί σε γλώσσα μηχανής από την επιλογή Build All. Ο assembler και ο linker είναι δύο από τα προγράμματα που μετέχουν στη λειτουργία αυτή. Ο assembler μεταφράζει τον κώδικα που είναι σε μνημονική γλώσσα assemble (ή C στην περίπτωσή μας) και έχει κατάληξη .asm σε γλώσσα μηχανής. Τα αρχεία που δημιουργούνται θα έχουν το ίδιο όνομα αλλά κατάληξη hex και Ist. Το αρχείο hex χρειάζεται ο προγραμματιστής για να περάσει τον κώδικα στο chip. Το αρχείο με κατάληξη Ist είναι βοηθητικό αρχείο και χρησιμεύει για τον έλεγχο του προγράμματος που έχουμε γράψει για συντακτικά λάθη και όχι για λάθη λογικής. Αν υπάρχουν λάθη που πρέπει να διορθωθούν θα εμφανιστεί το μήνυμα Build Failed. Εφόσον διορθωθούν ή δεν υπήρχαν από την αρχή εμφανίζεται το μήνυμα Build Succeeded. Για να περάσουμε τώρα αυτό το πρόγραμμα στον PIC18F4550 το MPLAB μας δίνει μέσω του μενού Programmer διάφορες επιλογές μια από τις οποίες είναι ο προγραμματιστής PICkit 3 που χρησιμοποιήσαμε και θα δούμε παρακάτω.

### 5.5 Capture Mode

Σύμφωνα με το φύλλο δεδομένων του PIC18F4550, κάθε περιφερειακό υλικό(hardware peripheral), είναι άμεσα συνδεδεμένο με έναν καταχωρητή ελέγχου (Control Register), και καταχωρητή δεδομένων(Data Register). Για την εφαρμογή, οι δύο αυτοί καταχωρητές είναι οι εξής: CCPxCON, CCPRx. Ο πρώτος καταχωρητής είναι για τον έλεγχο του περιφερειακού, και ο δεύτερος καταγράφει τα δεδομένα που παράγονται από το περιφερειακό του μικροελεγκτή. Για την λειτουργία του λεγόμενου Capture Mode, πρέπει να προστεθεί και ένας χρονοδιακόπτης, σύμφωνα με το φύλλο δεδομένων του PIC18F4550, συγκεκριμένα είτε ο Timer1 είτε ο Timer3. Η ακίδα(pin)όπου χρησιμοποιείται για να τοποθετηθεί ο αισθητήρας ο οποίος θα συλλαμβάνει τα event,ονομάζεται CCP1,και είναι συγκεκριμένα η ακίδα RC2. Στην παρακάτω εικόνα, φαίνεται το διάγραμμα ακίδων του μικροελεγκτή.



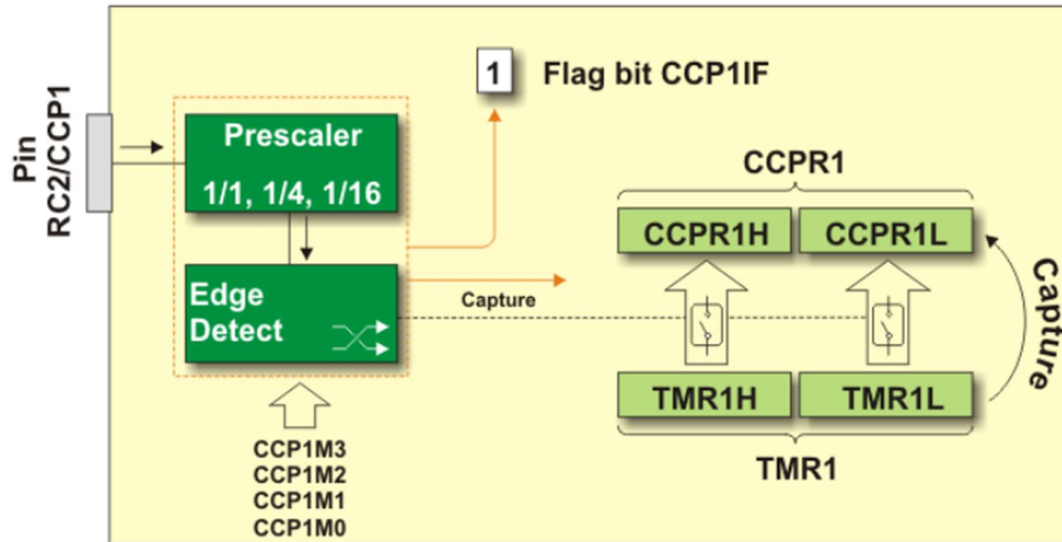
Σχήμα 5.1 Διάγραμμα ακίδων του μικροελεγκτή PIC18F4550

Αφού δει κάποιος το παραπάνω σχήμα, θα καταλάβει ότι ο αισθητήρας θα συνδεθεί στην ακίδα RC2, η οποία έχει την λειτουργία CCP1, λειτουργία που μπορεί να ανιχνεύσει ένα event. Άρα τα βήματα για να ακολουθήσει κάποιος ο οποίος θέλει να ενεργοποιήσει το Capture Module είναι τα εξής:

- Ενεργοποίηση pin RC2 σαν είσοδος. Η εντολή αυτή γράφεται `TRISCbits.TRISC2=1;`

Αυτή η εντολή είναι το πρώτο βήμα ενεργοποίησης του Capture Module

- Το επόμενο βήμα, είναι η αρχικοποίηση είτε του Timer1, είτε του Timer3. Στην περίπτωση της εφαρμογής, ο χρονοδιακόπτης που επιλέχθηκε είναι ο Timer1, και με την βοήθεια του καταχωρητή TICON, αρχικοποιήθηκε. Η τιμή του είναι `TICON=0xB1`, και υπερχειλίζει κάθε `40ms@8MHz`.



Εικόνα 5.1 Ο καταχωρητής CCP1CON

- Το επόμενο βήμα, είναι η αρχικοποίηση του καταχωρητή CCP1CON=0x05; Όπου σημαίνει πως ορίσαμε τον καταχωρητή για Capture Mode και να ανιχνεύει κάθε 1 σαν event.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
INTCON	GIE	PEIE	TOIE	INTE	RBIE <sup>(2)</sup>	TOIF	INTF	RBIF <sup>(2)</sup>	0000 000x	0000 000u
PIR	TMR1IF <sup>(1)</sup>								0	0
PIE	TMR1IE <sup>(1)</sup>								0	0
TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'.

Shaded cells are not used by the Timer1 module.

Note 1: The placement of this bit is device dependent.

Note 2: These bits may also be named GPIE and GPIF.

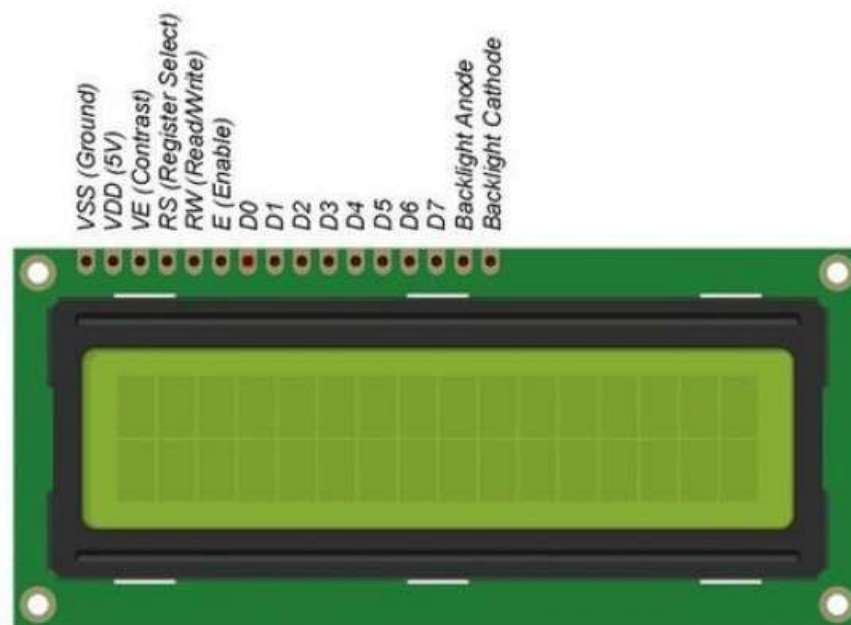
Εικόνα 5.2 Καταχωρητής ελέγχου T1CON

- Αφού αρχικοποιήθηκε το περιφερειακό Capture, και ο χρονοδιακόπτης 1 (Timer1), το περιφερειακό είναι έτοιμο. Στην main ρουτίνα του προγράμματος, εκτελείται ατέρμων μία while, η οποία χρησιμοποιεί σαν συνθήκη το Capture Bit, το οποίο είναι το PIR1bits.CCP1IF. Όταν ο αισθητήρας εντοπίσει κάποιο event, αποθηκεύει την τιμή στην μεταβλητή data1 και data2 αντίστοιχα. Η διαφορά των δύο αυτών μετρήσεων, είναι ο παλμός που συμβαίνει αυτό το event. Η περίοδος αυτού του παλμού, προκύπτει από την

διαίρεση μεταξύ του παλμού και του ρολογιού του μικροελεγκτή. Έτσι παράχθηκε η συχρότητα. Για την παραγωγή των RPM, απλά πολλαπλασιάζεται με το 60.

### 5.6 LCD Module

Για την ακόλουθη άσκηση, η έξοδος που προτιμήθηκε, δεν είναι άλλη από την οθόνη υγρών κρυστάλλων 16x2. Εύκολη στην χρήση, ο χρόνος απόκρισης της είναι μέσα στα όρια του μικροελεγκτή, και στην ουσία απεικονίζει την έξοδο του αισθητήρα του μικροελεγκτή.



Εικόνα 5.3 LCD 16x2 Module

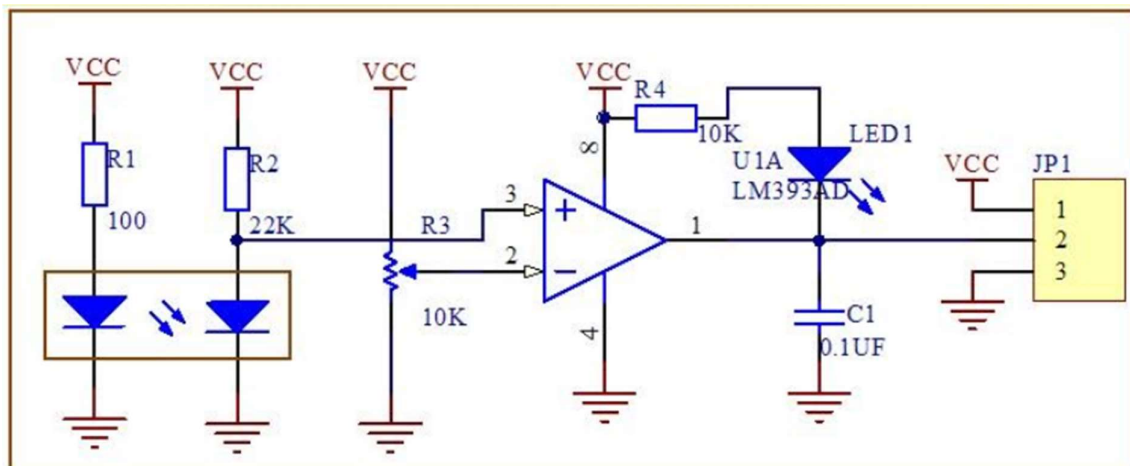
Η οθόνη που ενσωματώνεται στην αναπτυξιακή πλακέτα της εφαρμογής, είναι μία αλφανουμερική <έξυπνη> οθόνη με 2 σειρές και 16 χαρακτήρες σε κάθε γραμμή με ύπαρξη πίσω φωτισμού από Led κίτρινου-πράσινου χρώματος. Όπως αναφέρθηκε και παραπάνω, περιέχει τον μικροελεγκτή της Hitachi HD44780 και για την επικοινωνία με τον μικροελεγκτή της πλακέτας χρησιμοποιεί μια σειρά από εντολές που θα αναλυθούν παρακάτω. Στην οθόνη υπάρχει η δυνατότητα απεικόνισης λατινικών χαρακτήρων. Οι ελληνικοί χαρακτήρες υπάρχουν σε κάποιες οθόνες σαν δεύτερο σετ χαρακτήρων αλλά όχι σε όλες τις οθόνες. Όπου δεν υπάρχουν μπορούν να δημιουργηθούν σαν επιπλέον προγραμματισμένοι χαρακτήρες, δυνατότητα που δίνουν όλες οι οθόνες αυτού του τύπου.

### 5.7 Αισθητήρας Απόστασης

Ο αισθητήρας που προτιμήθηκε για αυτή την εφαρμογή βρίσκεται στην παρακάτω εικόνα. Έχει μέγιστη είσοδο μέχρι τα 5,5V και κάθε φορά που πλησιάζει ένα αντικείμενο τότε ανάβει το led στα αριστερά, χρησιμοποιείται σε στροφόμετρα, ταχογράφους και σε RC αμαξάκια για αποφυγή εμποδίων.



Εικόνα 5.4 Αισθητήρας Απόστασης



Σχήμα 5.2 Κύκλωμα αισθητήρα απόστασης

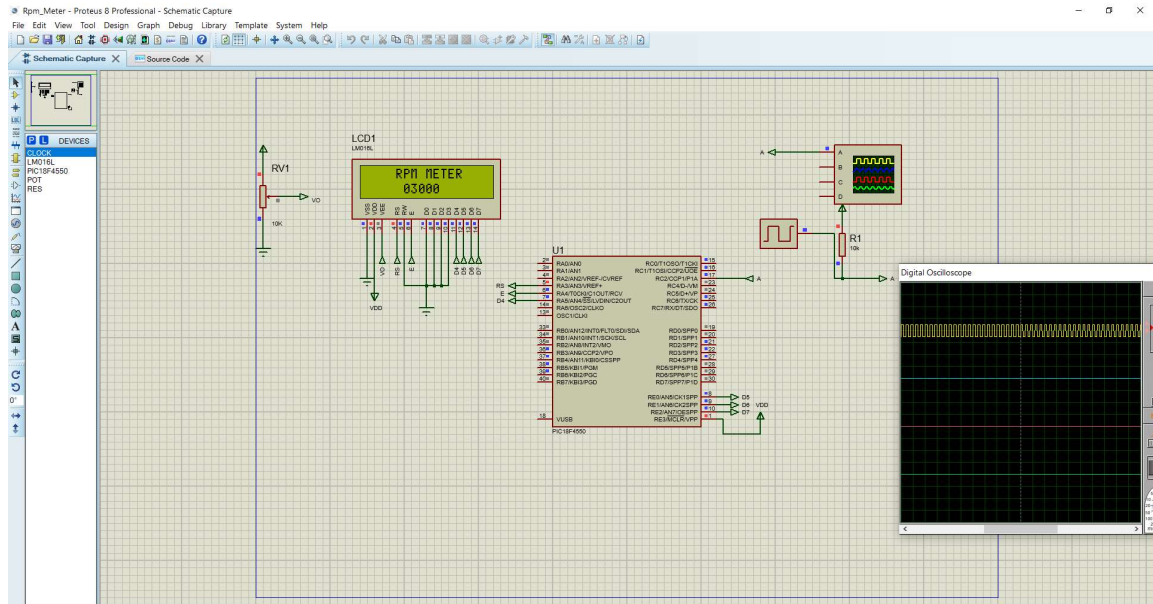
Το κύκλωμα του αισθητήρα απόστασης, είναι στην ουσία ένα κύκλωμα συγκριτή, ο οποίος στην αναστρέφουσα είσοδο έχει ένα κατώφλι που το ορίζει ο χρήστης, ενώ στην μη αναστρέφουσα είσοδο, έχει ένα δικτύωμα από IR Leds που όταν ανιχνεύσει κίνηση ο πομπός, άγει και ταυτόχρονα άγει και ο δέκτης. Αν η τάση της μη αναστρέφουσας εισόδου ξεπεράσει την αναστρέφουσα, τότε ο συγκριτής άγει, το LED στην έξοδο του συγκριτή άγει και η ακίδα 2 στέλνει HIGH. Με αυτή την λογική παράγεται η είσοδος στην ακίδα RPM του μικροελεγκτή που αναφέρθηκε παραπάνω.

### 5.8 Προσομοίωση με το Proteus

Αφού περιγράφηκε το κομμάτι του κώδικα, των εισόδων και εξόδων που αφορούν τον μικροελεγκτή, αφού ο κώδικας εκτελέστηκε επιτυχώς στο περιβάλλον MPLAB, το επόμενο βήμα είναι η προσομοίωση του στο πρόγραμμα Proteus.

Το Proteus είναι ένα επαγγελματικό πρόγραμμα προσομοίωσης ψηφιακών κυκλωμάτων, καθώς διαθέτει την ικανότητα τοποθέτησης του πηγαίου κώδικα στο σχηματικό, με αποτέλεσμα την βήμα-βήμα προσομοίωση του κώδικα, και του σχηματικού που σχεδιάστηκε να συμβαδίζει με τον κώδικα.

Στην παρακάτω εικόνα φαίνεται το σχηματικό προσομοίωσης, το οποίο περιλαμβάνει τον μικροελεγκτή PIC18F4550, την είσοδο του που είναι ένας παλμός για τον σκοπό της προσομοίωσης και την οδήγηση μιας οθόνης LCD για την έξοδο των μετρήσεων του αισθητήρα, αφού επεξεργάστηκε στον «εγκέφαλο» της εφαρμογής, που δεν είναι άλλος από τον μικροελεγκτή.

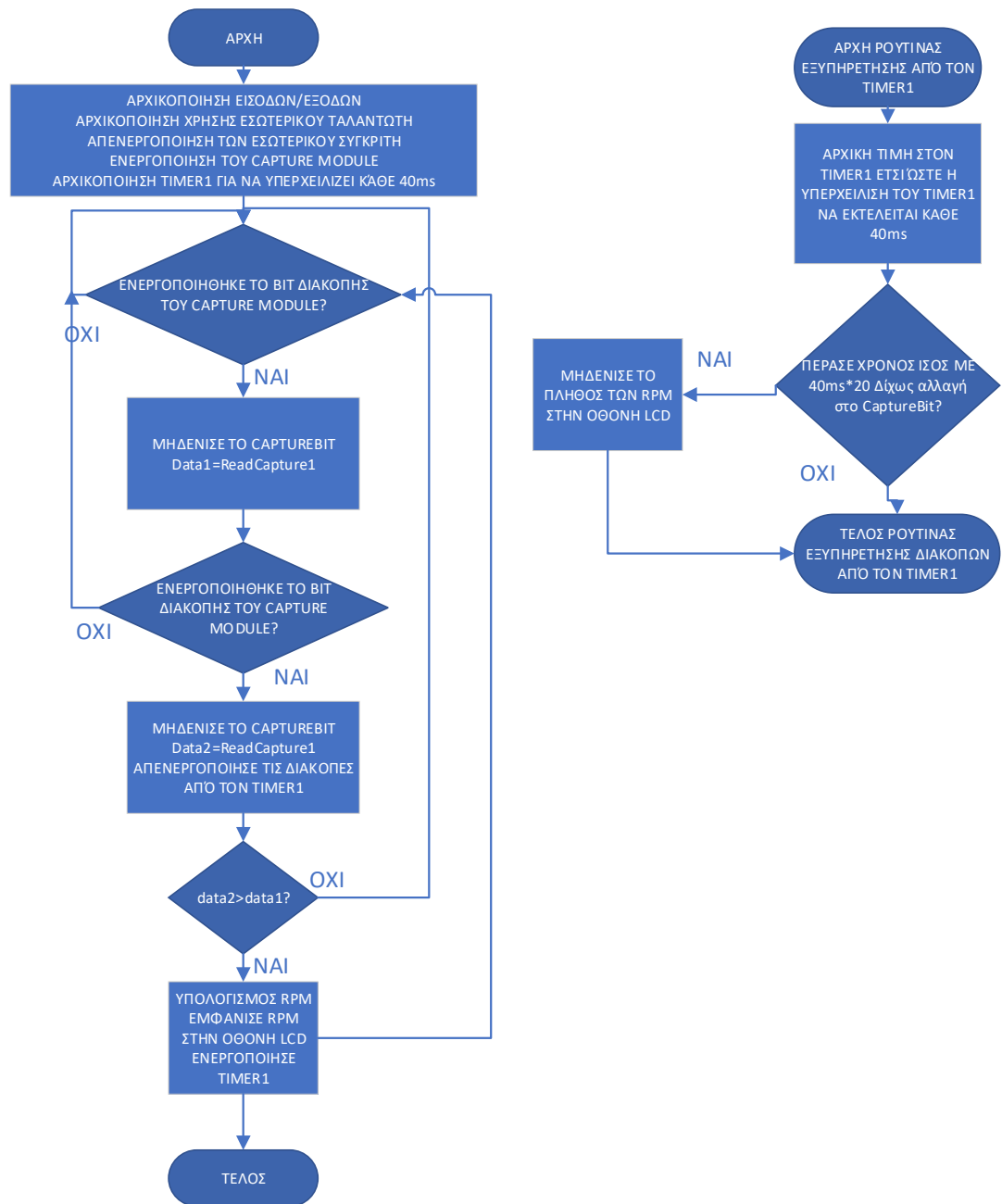


Εικόνα 5.5 Προσομοίωση στο Proteus

Για απεικονιστικούς λόγους, τοποθετήθηκε και ο παλμογράφος, για την επίδειξη του παλμού, ο οποίος είναι στα 50Hz, με την μέτρηση στροφών να προκύπτει και θεωρητικά άλλα και πρακτικά στην προσομοίωση η τιμή 3000RPM, γεγονός που δείχνει πως η μέθοδος υλοποίησης είναι και σωστή αλλά και ακριβής.

### 5.9 Διάγραμμα Ροής

Διάγραμμα ροής είναι ένα κοινού τύπου διάγραμμα που αναπαριστά έναν αλγόριθμο ή μια διαδικασία, δείχνοντας τα βήματα ως κουτιά διαφόρων ειδών που συνδέονται μεταξύ τους με βέλη. Αυτή η διαγραμματική παρουσίαση μπορεί να δώσει λύση βήμα προς βήμα σε ένα γνωστό πρόβλημα. Τα δεδομένα αναπαρίστανται σε κουτιά και τα βέλη δείχνουν τη ροή των δεδομένων. Τα διαγράμματα ροής χρησιμοποιούνται στην ανάλυση, το σχεδιασμό, την τεκμηρίωση ή τον έλεγχο μιας διαδικασίας ή ενός προγράμματος σε διάφορα πεδία. Πριν από κάθε σύνθεση προγράμματος επιβάλλεται πάντοτε η κατασκευή ενός διαγράμματος ροής. Το διάγραμμα ροής μας δίνει την δυνατότητα να βάλουμε σε σειρά τις σκέψεις για την σύνθεση του κώδικα δίνοντας μας παράλληλα μια οπτική σε όλο το φάσμα του προγράμματος. Το διάγραμμα ροής του προγράμματος παρουσιάζεται στο παρακάτω σχήμα.



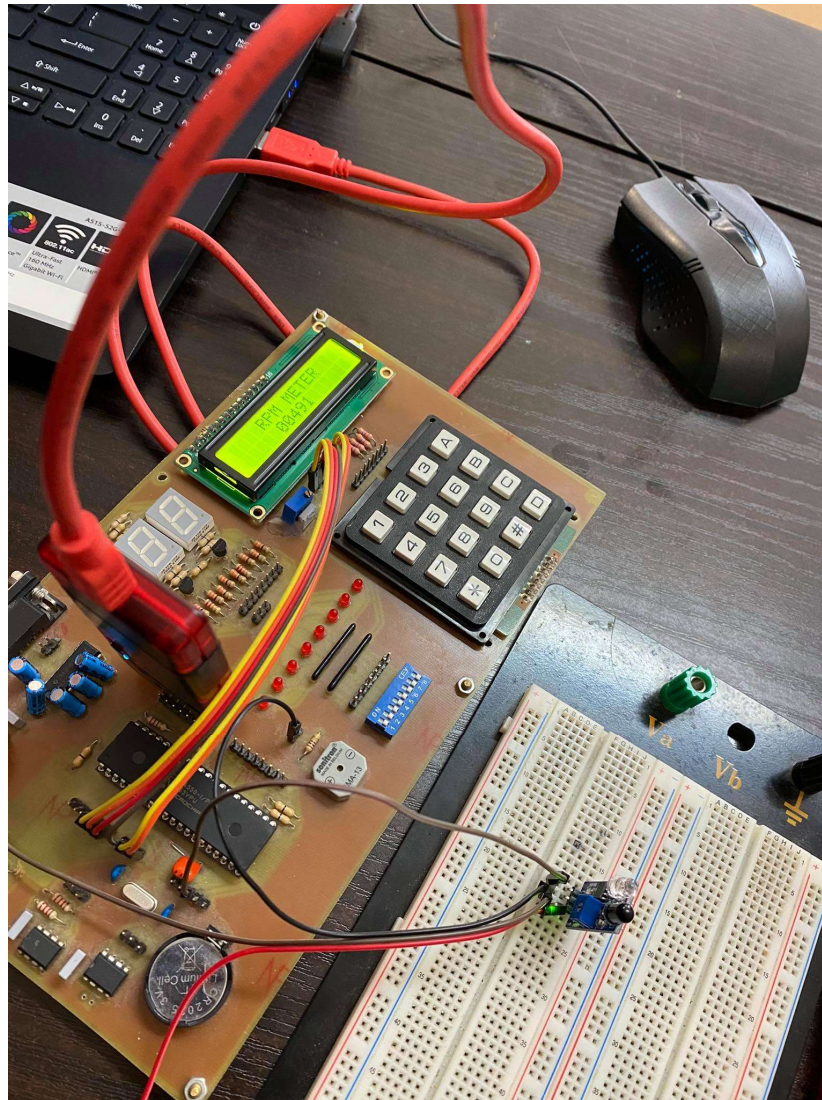
Εικόνα 5.6 Διάγραμμα ροής

## 5.10 Προσομοίωση στο ράστερ

Αφού το κομμάτι της υλοποίησης του κώδικα και της προσομοίωσης στο Proteus ολοκληρώθηκε με επιτυχία, το αμέσως επόμενο βήμα για την υλοποίηση της εφαρμογής είναι η παραγγελία των υλικών και η τοποθέτηση τους στο ράστερ για έλεγχο, διότι στην προσομοίωση όλα τα υλικά είναι σε ιδανικές συνθήκες, ενώ στον πραγματικό κόσμο δεν ισχύει αυτή η συνθήκη.

Αφού υλοποιηθεί και αυτό το κομμάτι, και γίνει κατανοητό πως όλα λειτουργούν σαν την προσομοίωση, το επόμενο βήμα είναι η κατασκευή της πλακέτας και η τοποθέτηση της σε ένα σταθερό μέρος για την λειτουργία της με διάφορους ανεμιστήρες για την μέτρηση των στροφών τους.

Στην παρακάτω εικόνα, φαίνεται η μέτρηση που προκύπτει από τις γρήγορες κινήσεις του χεριού, η απεικόνιση της εξόδου στην οθόνη.



Εικόνα 5.7 Μέτρηση στροφών σε ράστερ

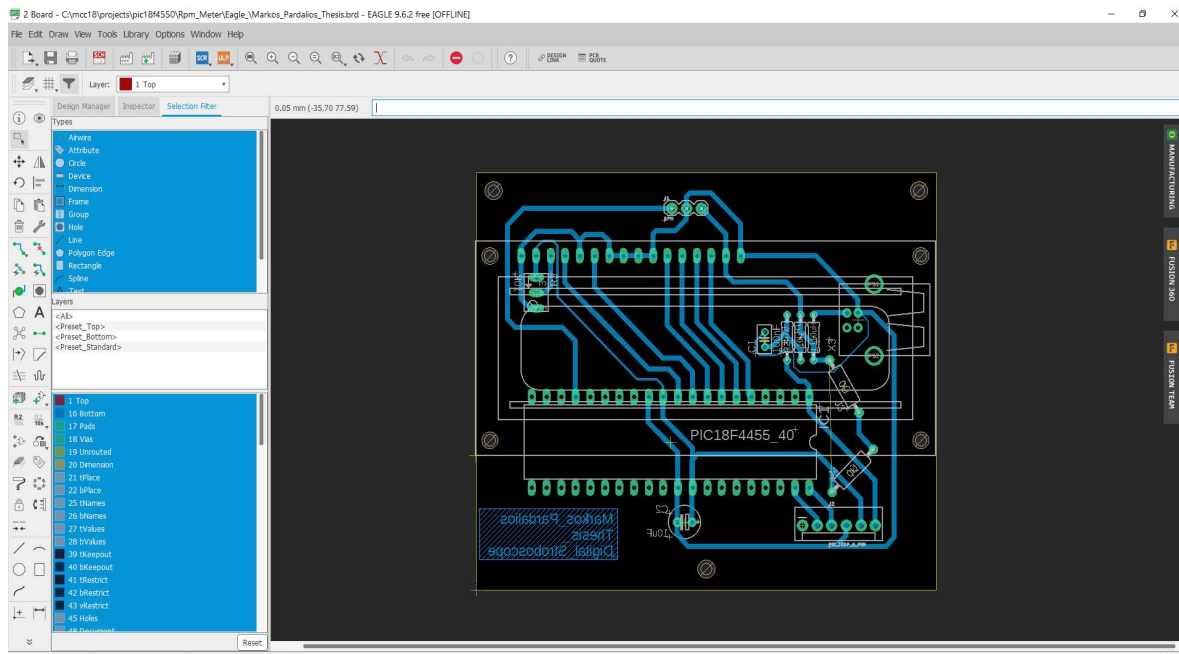
### 5.11 Σχεδίαση πλακέτας

Αφού λειτούργησε η εφαρμογή και στο ράστερ, το μόνο που μένει είναι ο σχεδιασμός πλακέτας, και η τοποθέτηση υλικών πάνω σε αυτήν για να μπορέσει η εφαρμογή να εκτελείται αποκλειστικά και μόνο για τον σκοπό μέτρησης στροφών. Για να επιτευχθεί αυτό το κομμάτι όμως, πρέπει να ληφθούν κάποιες αποφάσεις, που έχουν να κάνουν με την τοποθέτηση των υλικών και με τον τρόπο τροφοδοσίας. Στην εικόνα παραπάνω βλέπουμε την αναπτυξιακή πλακέτα του εργαστηρίου να τροφοδοτείται μέσω του προγραμματιστή PICKIT 3, ενώ για την πλακέτα το κύκλωμα ICSP θα χρησιμοποιείται μόνο για το πέρασμα του κώδικα στον μικροελεγκτή. Από αυτή την πρόκληση προκύπτει το θέμα τροφοδοσίας της πλακέτας

- **Τροφοδοσία:** Για την εφαρμογή, η μέθοδος τροφοδοσίας που επιλέχθηκε, είναι το USB τύπου B, επειδή είναι το ίδιο με αυτό που χρησιμοποιείται στην αναπτυξιακή πλακέτα του εργαστηρίου. Ο λόγος που δεν επιλέχθηκε κάποια μπαταρία είναι ότι ο μικροελεγκτής PIC18F4455 τροφοδοτείται στα 5V και εφόσον επιλεγόταν θα έπρεπε να τοποθετηθεί και σταθεροποιητής τάσης και κατ' επέκταση και δύο πυκνωτές για αποθρονοποίηση. Επίσης όταν η τάση της μπαταρίας φτάσει περίπου στα 7 volt, δεν θα άγει ο σταθεροποιητής τάσης, με αποτέλεσμα την μη λειτουργία του κυκλώματος και της οθόνης LCD.
- **Έλεγκτής:** Ένας ελεγκτής ικανοποιητικής ταχύτητας και ακρίβειας, απαιτείται για να στέλνει και να λαμβάνει εντολές για την ομαλή μέτρηση των στροφών.
- **Μηχανισμός ελέγχου:** Για τον έλεγχο των μετρήσεων, θα τοποθετηθεί ένας ανεμιστήρας στα 50-60Hz, έτσι ώστε να δείχνει σταθερό αριθμό στην έξοδο της εφαρμογής (LCD)
- **Έξοδος:** Η έξοδος, δηλαδή το μέσο απεικόνισης των στροφών, είναι η οθόνη υγρών κρυστάλλων 16x2, με real time ανανέωση της εξόδου της εφαρμογής, δηλαδή τις στροφές (RPM).
- **Προγραμματισμός:** Για την εγκατάσταση του προγράμματος στον μικροελεγκτή PIC18F4455, στην πλακέτα τοποθετείται το κύκλωμα ICSP, για τον προγραμματιστή (PICKIT 3), να έχει την δυνατότητα για αποσφαλμάτωση, και για απευθείας εγκατάσταση του προγράμματος στην πλακέτα της εφαρμογής.

Αφού περιγράφηκαν οι παραπάνω προκλήσεις που προέκυψαν από τον σχεδιασμό της πλακέτας, λαμβάνονται υπόψη και οι αποφάσεις και προκύπτει το τελικό σχέδιο, με σκοπό την όσο το δυνατόν περισσότερη ακρίβεια στο ψηφιακό στροφόμετρο.

Η σχεδίαση της πλακέτας έγινε με το πρόγραμμα Eagle, είναι μονής όψης και παρακάτω φαίνεται το σχηματικό αλλά και η πλακέτα.



Εικόνα 5.8 Πλακέτα ψηφιακού στροφόμετρου



SUNON

## SPECIFICATIONS

MODEL: SF23080A

P/N : 2083HSL.GN

- 1-1. Rated Voltage : 220-240 VAC 50/60 Hz
- 1-2. Operating Voltage Range : 185-245 VAC
- 1-3. Starting Voltage : 185 VAC(25 deg. C POWER ON/OFF)
- 1-4. Rated Speed : 2300/2750 RPM  $\pm$  10%
- 1-5. Air Delivery : 23.0/30.0 CFM
- 1-6. Static Pressure : 0.12/0.18 Inch-H<sub>2</sub>O
- 1-7. Rated Current : 0.09/0.08 AMP
- 1-8. Rated Power : 18/16 WATTS
- 1-9. Noise Level : 31.0/35.0 dB(A)
- 1-10. Direction of Rotation : Counter-clockwise viewed from front of fan blade
- 1-11. Operating Temperature : -10 to +70 Deg.C
- 1-12. Storage Temperature : -40 to +70 Deg.C
- 1-13. Bearing System : Lubricated sleeve bearing system
- 1-14. Weight : 340g
- 1-15. Safety : TUV/UL/CUR/CE Approvals
- 1-16. Vibration : Vibration of acceleration 1.5G and frequency 5~50~5Hz is applied in all 3 directions(X,Y,Z), in cycles of 1 minute each, for a total vibration time of 30 minutes.

建準電機工業股份有限公司 SUNONWEALTH ELECTRIC MACHINE INDUSTRY CO., LTD	SPEC.NO	A08000520G-00	REVISION DATE
	ISSUE DATE	04.30.2005	E D I T I O N
	P A G E	3 OF 9	



Εικόνα 5.10 Προδιαγραφές Ανεμιστήρα

Το μοντέλο ανεμιστήρα που προτιμήθηκε είναι το SF23080A, της εταιρείας SUNON. Μερικές από τις προδιαγραφές που αναφέρονται είναι:

- Προτιμώμενη τάση τροφοδοσίας (220-240Vac)
- Εύρος τάσης τροφοδοσίας (185-245 Vac)
- Τάση έναυσης (185Vac)
- Εκτιμώμενες στροφές ανεμιστήρα (2300-2750 rpm +-10% ανοχή)
- Εκτιμώμενο ρεύμα (0.09-0.08 Amps)
- Εκτιμώμενη ισχύς (18/16 Watts)
- Επίπεδο θορύβου (31/35 dB(A))
- Η κατεύθυνση του ανεμιστήρα πηγαίνει σύμφωνα με τον δείκτη του ρολογιού
- Θερμοκρασία λειτουργίας (-10 έως 70 βαθμούς Κελσίου)
- Θερμοκρασία αποθήκευσης (-40 έως 70 βαθμούς Κελσίου)
- Πιστοποίηση ασφαλείας (TUV/UL/CUR/CE Approvals)

Με βάση τα παραπάνω χαρακτηριστικά και σύμφωνα με τα αποτελέσματα που προκύπτουν από την δοκιμή της εφαρμογής στροφόμετρου στον ανεμιστήρα, τα αποτελέσματα των μετρήσεων συμπίπτουν με αυτά της προσομοίωσης, άρα η εφαρμογή λειτουργεί, το σφάλμα είναι:

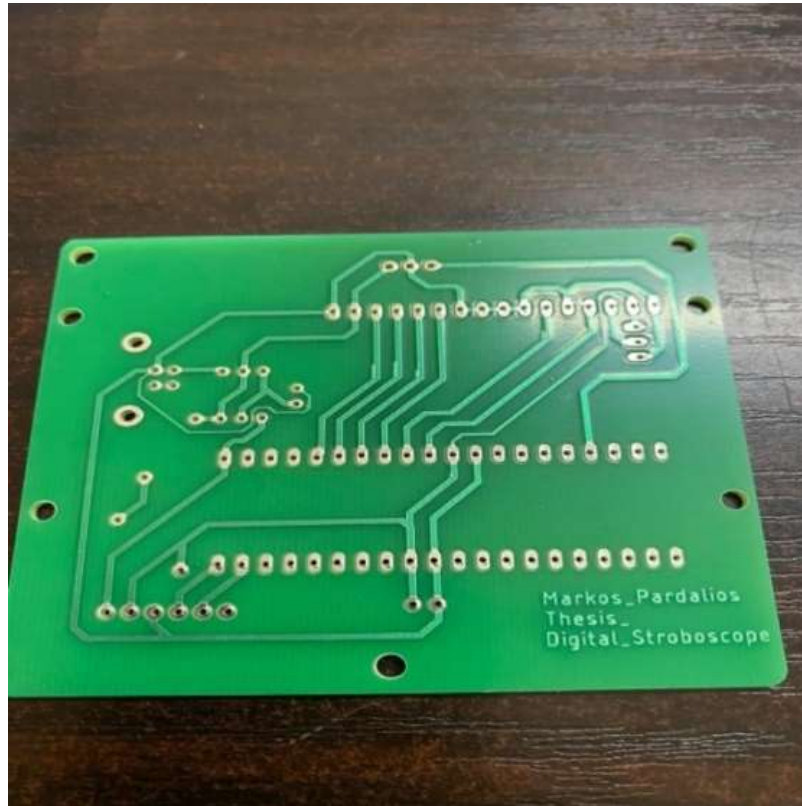
Τύπος υπολογισμού σφάλματος:

$$\left( \frac{\text{Ιδανική Τιμή} - \text{Τιμή Μέτρησης}}{\text{Ιδανική Τιμή}} \right) * 100 = \text{Σφάλμα\%}$$

$$\left( \frac{2300 - 2249}{2300} \right) * 100 = 2\%$$

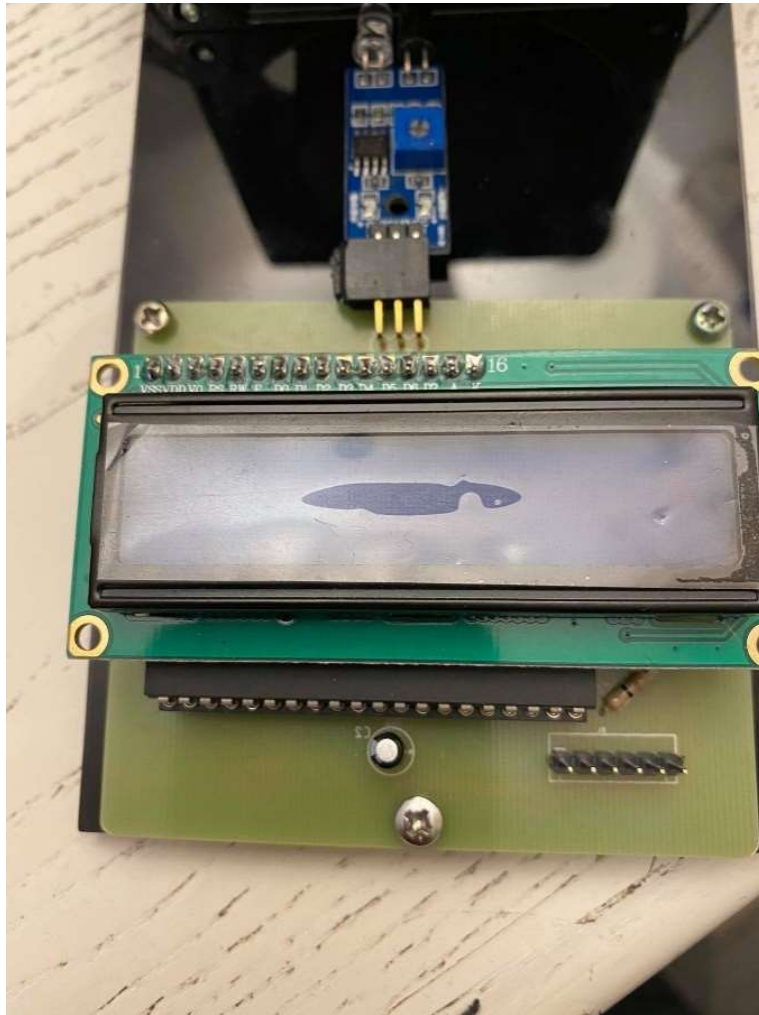
### 5.13 Τύπωση Πλακέτας και τεστ σε πραγματικές συνθήκες

Έπειτα από την αναμονή μερικών εβδομάδων, για την τύπωση του PCB, αφού ήρθε στα χέρια του συγγραφέα, τα υλικά πάρθηκαν, κολλήθηκαν πάνω στην πλακέτα, και με την βοήθεια του Pickit 3, το hexαρχείο περάστηκε στον μικροελεγκτή. Παρακάτω εμφανίζεται η εικόνα του κάτω μέρους (Bottom) της πλακέτας, όπου είναι σε πραγματική μορφή σε σχέση με την προηγούμενη φωτογραφία που φαίνεται σε κλίμακα μεγέθυνσης μέσα από το πρόγραμμα Eagle.



Εικόνα 5.11 Τυπωμένο κύκλωμα pcb

Αφού κολλήθηκαν τα υλικά πάνω στην πλακέτα, το επόμενο βήμα είναι να βρεθεί ένας ανεμιστήρας, ο οποίος θα βρίσκεται πολύ κοντά στον αισθητήρα ταχύτητας, για να μπορεί να συλλάβει τις γρήγορες εναλλαγές των πτερυγίων του ανεμιστήρα. Μετά την κόλληση των υλικών η πλακέτα φαίνεται όπως την παρακάτω εικόνα.



Εικόνα 5.12 Ολοκληρωμένη πλακέτα

Για την σωστή απόσταση μεταξύ του αισθητήρα κίνησης και του ανεμιστήρα, τοποθετήθηκε σαν πατούρα ένα κομμάτι plexi glass 15x9, για την τοποθέτηση των υλικών, σε σωστή απόσταση και για την ευκολία παρουσίασης. Αφού υλοποιήθηκαν τα παραπάνω κομμάτια που περιγράφονται, έγινε η τελική δοκιμή με επιτυχία. Στην παρακάτω επιτυχία φαίνεται το αποτέλεσμα στην εικόνα LCD, που δείχνει τις στροφές του κινητήρα και το μήνυμα RPM METER στην πρώτη γραμμή του LCD, και τον αριθμό των στροφών στην δεύτερη.



Εικόνα 5.13 Ψηφιακό Στροφόμετρο

## 6 Επίλογος και Συμπεράσματα

### 6.1 Επίλογος

Στο πρώτο κεφάλαιο αναφέρονται γενικά χαρακτηριστικά για τους μικροελεγκτές και στην ανάπτυξη εφαρμογών λογισμικού, στο δεύτερο γίνεται μια ευρεία ανάλυση μοντελοποίησης του προβλήματος της μέτρησης στροφών και την ιστορία τους, στο τρίτο κεφάλαιο αναφέρεται η μέθοδος ανάπτυξης σε όλα τα επίπεδα από τον κώδικα μέχρι και την σχεδίαση και λειτουργία της πλακέτας. Μέσω αυτής της πτυχιακής εργασίας, ο συγγραφέας κατάφερε ένα πρόβλημα να το μοντελοποιήσει και να αναπτύξει μια εφαρμογή με παραλληλισμό γνώσεων μηχανικής, μαθηματικών, ηλεκτρονικών και προγραμματισμού. Τα κομμάτια στα οποία υπήρξε δυσκολία ήταν: Συγχρονισμός των ακίδων για το σωστό αποτέλεσμα, χαμηλή ποιότητα του plexi glass με αποτέλεσμα να μην μπορεί να υπολογιστεί η ακριβής απόσταση ανεμιστήρα-αισθητήρα για το βέλτιστο αποτέλεσμα. Λόγω των πολλών μετρήσεων που παίρνει ο αισθητήρας παρατηρήθηκε ένα latch (θόρυβος στην οθόνη lcd) άλλα επιλύθηκε μέσω της διαδικασίας αποσφαλμάτωσης (debugging). Για την ομαλή λειτουργία του σε συνεχόμενη ροή, πρέπει πρώτα να μπει σε κατάσταση λειτουργίας ο ανεμιστήρας και μετά η πλακέτα. Η πλακέτα τροφοδοτείται είτε μέσω του Pickit3 είτε μέσω του USB-B. Αν ο αισθητήρας για κάποιο χρονικό διάστημα, δεν λάβει μέτρηση, τότε πραγματοποιείται Timeout μέσω του Timer1, και στην οθόνη τοποθετείται η τιμή 0, δηλαδή υπάρχει ένα έξτρα χαρακτηριστικό ελέγχου που επιβλέπει την ομαλή λειτουργία του αισθητήρα. Αν ο αισθητήρας έχει βλάβη, ή δεν είναι συνδεδεμένος στην οθόνη είναι γραμμένος ο αριθμός 0 που σημαίνει πως είτε ο αισθητήρας δεν είναι κάνει επαφή είτε ο ανεμιστήρας είναι ανενεργός. Αυτό είναι μια έννοια ελέγχου που χρησιμοποιείται κατά κόρον στα ενσωματωμένα συστήματα, δηλαδή υπάρχει έλεγχος σε πραγματικό χρόνο των περιφερειακών που είναι συνδεδεμένα στον μικροελεγκτή για την κατάσταση στην οποία βρίσκονται. Παρακάτω υπάρχει η λίστα με τα υλικά που χρησιμοποιήθηκαν και το κόστος αυτών, το οποίο μπορεί να συγκριθεί με πολλά στροφόμετρα που υπάρχουν στο εμπόριο και την αποτελεσματικότητά τους.

Λίστα Υλικών (BOM)		
α/α	Υλικό	Τιμή(€)
1	Ανεμιστήρας	16
2	Αισθητήρας ταχύτητας	4
3	Μικροελεγκτής PIC18F4550	8
4	USB B 2.0	2
5	Female Headers	0.5
6	Male Headers	0.5
7	Πλακέτα	20
8	Αντιστάσεις	0.5
9	Πυκνωτές	1
10	Ποτενσιόμετρο 10κ	1.5
11	Οθόνη 16x2	4
12	Επιφάνεια Plexi Glass	15
13	Programmer Pickit 3	20
	Σύνολο	90

Εικόνα 6.1 Πίνακας και κόστος υλικών

## 6.2 Συμπεράσματα ή/και προτάσεις βελτίωσης

Κάποια συμπεράσματα που προκύπτουν από την ανάπτυξη της εφαρμογής είναι πως οι μετρήσεις ήταν ακριβείας, αν και υπάρχει το ποσοστό σφάλματος 2% το οποίο θεωρείται αποδεκτό. Κάποιες προτάσεις βελτίωσης θα ήταν ένας μικροελεγκτής μικρότερος σε κόστος, ακόμη με λιγότερες ακίδες, διότι στην συγκεκριμένη εφαρμογή η επιλογή του PIC18F4550 αποτελεί πολύ υπερβολικά καλή για τις προδιαγραφές της εφαρμογής. Μια ακόμα αποδεκτή λύση θα ήταν η χρήση του PIC18F2550, ο οποίος έχει λιγότερες ακίδες, έχει 16-bit χρονιστές και έχει και το Capture Module που αναφέρθηκε στο κεφάλαιο 1 και στο κεφάλαιο 3. Ακόμη μία πρόταση βελτίωσης είναι τα υλικά που θα τοποθετούνταν στην πλακέτα θα μπορούσαν να ήταν επιφανείας (SMD), αντί τα διάτρητα που χρησιμοποιηθήκαν. Παρόλο αυτά λόγω της έλλειψης στοιχείων σε smd, επιλέχθηκε η λύση με διάτρητα υλικά, έτσι ώστε η πλακέτα να ήταν μικρότερη. Ακόμη μία πρόταση βελτίωσης είναι η προσθήκη της σειριακής θύρας (RS232) έτσι ώστε να μπορεί ο χρήστης να διακρίνει τις τιμές από H/Y και να μπορέσει να κάνει μια μαθηματική ανάλυση ως προς τον μέσο όρο των μετρήσεων, την επικρατούσα τιμή του δείγματος που λαμβάνει ο μικροελεγκτής, καθώς και την τυπική απόκλιση για μια εντελώς επαγγελματική προσέγγιση. Μία ακόμη βελτίωση θα μπορούσε να ήλεγχε τις στροφές του κινητήρα μέσω του Hardware PWM Module CCP2 για την ελεγχόμενη κίνηση του κινητήρα, μέσω του καταχωρητή CCP2. Μία ακόμη βελτίωση του κυκλώματος είναι η χρήση led για την κατάσταση που βρίσκεται ο αισθητήρας (πχ. το κόκκινο LED υποδηλώνει πως δεν παίρνει μέτρηση ο αισθητήρας και το πράσινο υποδηλώνει πως παίρνει μέτρηση ο αισθητήρας). Ωστόσο παρόλες τις προτάσεις βελτίωσης της εφαρμογής, είναι μία αξιόπιστη εφαρμογή, δεν κολλάει, μπορεί να παίρνει μετρήσεις δίχως θόρυβο, και ενδείκνυται για μαζική παραγωγή, αφού το πρωτότυπο λειτουργεί με ακρίβεια σε κινητήρα ισχύος (220VA/C) και μπορεί να κάνει την λειτουργία capture με απόλυτη επιτυχία. Ένα ακόμη συμπέρασμα είναι πως με την

χρήση κώδικα, και τον έλεγχο της οθόνης, μπορούν να αποφευχθούν κάποιες προτάσεις βελτίωσης, όπως η τοποθέτηση LEDs για την κατάσταση του αισθητήρα, αφού το λογισμικό είναι σε θέση να εντοπίσει μέσω του Capture περιφερειακού την κατάσταση στην οποία βρίσκεται ο αισθητήρας, αν δηλαδή μετράει την κίνηση του ανεμιστήρα ή αν δεν την μετράει, και να εμφανιστεί το κατάλληλο μήνυμα στην οθόνη.

## ΠΑΡΑΡΤΗΜΑ

### Κώδικας προγράμματος σε γλώσσα C

```
void main(void)
{
    PORTA = Init_portA; TRISA = Config_portA;           // Config_portA    0b11011111
    PORTB = Init_portB; TRISB = Config_portB;           // Config_portB0b10000111
    PORTC = Init_portC; TRISC = Config_portC;           // Config_portC0b00111100
    PORTD=Init_portD;TRISD=Config_portD;CMCON=0x07;//Config_portD0b11100011
    PORTE = Init_portE; TRISE = Config_portE;// Config_portE           0b11111000
    OSCCON=0x72;                                         //set internal oscillator to 8Mhz
    PIE1bits.CCP1IE=0;
    PIR1bits.CCP1IF=0;
    CCP1CON=0x05;    // Capture mode is selected for detecting Rising edge */
    CCPR1=0x00;    //CCPR1 is capture count Register which is cleared initially*/
    //T1CON=0x81;

    T1CON=0xB1;
    // OpenTimer1 (TIMER_INT_ON & T1_16BIT_RW & T1_SOURCE_INT &
    T1_PS_1_1&T1_OSC1EN_OFF&T1_SYNC_EXT_OFF); //timer1 config 16bit, prescaler 1:1
    WriteTimer1 (RPMclock); //orizoume reg value gianaksekinai to timer counter apo auti tin timi
    IPR1bits.TMR1IP=1;
```

```

// Hight Priority Timer 1

RCONbits.IPEN=1; // Enable priority level
INTCONbits.GIEL = 1;
INTCONbits.GIEH = 1;

Init();

//perilamvaneiarxikopoihsh lcd & flags
LCD_GOTO(1,1);

//phgainestinseira 1 kai grammh 5 tis othonis
//
LCD_PUTS(hi);

//grapse to mnumastinothoni

while (1) {
//endless loop
MainCode(); //ekteleitausunexeia
}
//
void MainCode(void) //ekteleitausunexeia
{
while(!(PIR1bits.CCP1IF)); //capture bit
PIR1bits.CCP1IF==0; //reset capture bit
data1=CCPR1;//data1 holds value of CCPR1
while(!(PIR1bits.CCP1IF)); //capture bit
PIR1bits.CCP1IF==0; //reset capture bit
data2=CCPR1; //data2 holds value of CCPR1
//
if(Block.Comp){Block.Comp==0;

if(data1<data2){

```

```

//upologismos rpm
pulse=data2-
data1;

//upologismos pulse

period=((float)pulse/(float)F_Timer);
//upologismosperiodou

frequency=1/(period*8);
//ypologismossuxnohtas

RPM=frequency*60;

//ypologismos revolutions per minute(RPM)
LCD_GOTO(1,5); //phgainestirseira 1 kai grammh 5 tis othonis
LCD_PUTS(rpm);//grapse to mhnumastinothoni
Display_RPM(RPM); //routinaendeiksis rpm
}

//
T1CON=0xB0;
//deactivate timer1

//
}
//
void Init(void) //perilamvaneiarxikopoihsh lcd & flags
{
LCD_INIT();

//LCD INIT

LCD_CLEAR();

```

```

//LCD CLEAR
}
//
void Display_RPM(unsigned long num)
{
char Frame[8];

//pinakas 9 thesewn
Frame[0]=num/10000+0x30;

//dekadesxiliades
Frame[1]=(num/1000)%10+0x30;

//xiliades
Frame[2]=(num/100)%10+0x30;

//ekatontades
Frame[3]=(num/10)%10+0x30;

//dekades
Frame[4]=num%10+0x30;

//monades
LCD_GOTO(2,6);

//phgainestirseira 1
LCD_PUTS(Frame);

//emfanisepinaka
LCD_GOTO(2,11);

//pane stithesi 11 tis 2is grammis
LCD_PUTS clr);

```

```
                                                                    //katharise ta skoupidia
if(Frame[0]==0){
                                                                    //an den phremetrhsh
LCD_GOTO(2,6);
                                                                    //pane stithesi 6 stin 2i grammi
LCD_PUTS(clear);
                                                                    //katharise ta skoupidia
}
}
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Προγραμματισμός μικροελεγκτών για μηχανικούς 2017
- [2] Malvino Ηλεκτρονική
- [3] Γιακουμής Άγγελος, Σημειώσεις του μαθήματος Μικροελεγκτές 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] Microchip MLA,v4.0
- [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.electronicstutorials.ws/power/three-phase-rectification.html>
- [9] Electronics tutorials, engineersgarage, Capture mode
- [10] PIC MICROCONTROLLER PROGRAMMING IN C, MikroElektronika Milan Verle
- [11] Designing Embedded Systems with Pic Microcontrollers, Tim Wilmshurst