



Α.Τ.Ε.Ι. Θεσσαλονίκης
Τμήμα Ηλεκτρονικής

Πτυχιακή Εργασία

Σχεδιασμός και κατασκευή συστήματος μέτρησης καρδιακών παλμών με τη χρήση μικροελεγκτή.

Κωδικός Πτυχιακής : 12130ΕΣ

Ημερομηνία Ανάληψης : Μάρτιος 2012

Ημερομηνία Περάτωσης : Αύγουστος 2013

Φοιτητές: Γιαννούλης Δημήτρης (507017)

Πελεκάνου Θεοδώρα (507051)

Υπεύθυνη καθηγήτρια : Παπαδοπούλου Μαρία

ΠΕΡΙΛΗΨΗ

Το κεντρικό θέμα είναι η μελέτη, η σχεδίαση και η κατασκευή μιας συσκευής μέτρησης του καρδιακού ρυθμού από το δάχτυλο με την βοήθεια ενός μικροεπεξεργαστή. Για το πρώτο κομμάτι που αφορά την μελέτη, είναι αναγκαίο να αναφερθούμε στην λειτουργία της καρδιάς και του κυκλοφορικού συστήματος καθώς και στην φωτοπληθυσμογραφία η οποία είναι η μέθοδος στην οποία θα βασιστούμε για την απόκτηση των μετρήσεων. Στην συνέχεια για την σχεδίαση του κυκλώματος απόκτησης και επεξεργασίας μετρήσεων εξηγείται η επιλογή του αισθητήρα, ο υπολογισμός των φίλτρων και της ενίσχυσης. Ακολουθεί η επιλογή του μικροεπεξεργαστή, με αναφορά και ανάλυση των χαρακτηριστικών που απλοποιούν την κατασκευή της συσκευής. Οι μετρήσεις εμφανίζονται σε μια οθόνη LCD, αποθηκεύονται στη μνήμη EEPROM του μικροεπεξεργαστή και παρέχεται η δυνατότητα σύνδεσης της συσκευής μέσω USB στον υπολογιστή για την περαιτέρω μελέτη και επεξεργασία των αποθηκευμένων μετρήσεων. Υπάρχει αναφορά στον κώδικα για τον μικροεπεξεργαστή καθώς και στο λογισμικό που χρησιμοποιείται από την πλευρά του υπολογιστή για την απόκτηση των μετρήσεων κατά την σύνδεση μέσω USB.

Λέξεις κλειδιά : φωτοπληθυσμογραφία, καρδιακός ρυθμός, μικροεπεξεργαστής

SUMMARY

The key point is the study, the design and the manufacturing of a simple heart rate measuring device. The subject's heart rate is measured from the finger with the use of a microcontroller. It is necessary to make note of the way the human heart operates as well as the basic features of photoplethysmography, the principle on which our measuring method is based. Moreover, we analyze the filter design and the choice of sensor, followed by the choice of microcontroller with mention of the registers being used. All measurements are displayed on a LCD screen, saved in the microcontroller's EEPROM memory and can be accessed by connecting the device on a computer by USB. The microcontroller's code is also mentioned as well as the software needed for the computer to retrieve the measurements from EEPROM.

Key words : *photoplethysmography, heart rate, microcontroller*

Πίνακας περιεχομένων

ΠΡΟΛΟΓΟΣ	1
1.Η ΚΑΡΔΙΑ ΚΑΙ ΤΟ ΚΥΚΛΟΦΟΡΙΚΟ ΣΥΣΤΗΜΑ.....	2
1.1 Το κυκλοφορικό σύστημα.....	2
1.2 Βιολογικά σήματα.....	3
1.3 Καρδιακός ρυθμός.....	3
1.4 Τρόποι μέτρησης.....	4
2.Η ΦΩΤΟΠΛΥΘΙΣΜΟΓΡΑΦΙΑ	6
2.1 Βασικές αρχές.....	6
2.2 Τύποι αρτηριακής φωτοπληθυσμογραφίας.....	7
2.3 Μορφή του σήματος.....	8
2.4 Αισθητήρες αρτηριακής φωτοπληθυσμογραφίας.....	8
3.ΤΑ ΕΠΙΜΕΡΟΥΣ ΣΤΟΙΧΕΙΑ ΚΑΙ ΟΙ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ	10
3.1 Ο μικροελεγκτής PIC16F887.....	10
3.1.1 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής :Χρονιστής/μετρητής Timer0, OPTION.....	11
3.1.2 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής : OSCCON.....	13
3.1.3 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής : TXSTA, RCSTA και BAUDCTL για την ασύγχρονη σειριακή επικοινωνία.....	14
3.1.4 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής : Οι καταχωρητές εισόδου/εξόδου.....	17
3.2 Ο αισθητήρας.....	18
3.2.1 Χαρακτηριστικά ενός αισθητήρα.....	18
3.2.2 Ο αισθητήρας TCRT5000L.....	19
3.2.3 Ο αισθητήρας στο κύκλωμα.....	20
3.3 Τα φίλτρα.....	22
3.3.1 Κατηγορίες φίλτρων.....	22
3.3.2Μαθηματική Ανάλυση Φίλτρων.....	25
3.4 Οι επιμέρους λειτουργίες.....	29
3.4.1Μέτρηση.....	29
3.4.2 Απεικόνιση.....	31

3.4.3 Αποθήκευση και διαγραφή μνήμης.....	35
3.4.4 Σειριακή μεταφορά δεδομένων από την συσκευή στον υπολογιστή.....	40
3.5 Κόστος κατασκευής.....	43
4. ΛΟΓΙΣΜΙΚΟ Η/Υ ΓΙΑ ΤΗΝ ΜΕΤΑΦΟΡΑ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ ΜΕΤΡΗΣΕΩΝ.....	45
4.1 Η γλώσσα προγραμματισμού Processing.....	45
4.2 Το πρόγραμμα Create Log File.....	46
4.3 Το πρόγραμμα Create Plot.....	48
5. ΑΠΟΤΕΛΕΣΜΑΤΑ.....	50
5.1 Μέτρηση για 15 δευτερόλεπτα.....	50
5.2 Μέτρηση για 30 δευτερόλεπτα.....	51
5.3 Μέτρηση για 60 δευτερόλεπτα.....	52
5.4 Τελική επιλογή διάρκειας μέτρησης.....	53
6. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	55
6.1 Πλεονεκτήματα.....	55
6.2 Μειονεκτήματα.....	55
6.3 Τρόποι βελτίωσης.....	56

Κώδικας για τον PIC16F887 και διάγραμμα ροής

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΠΡΟΛΟΓΟΣ

Η καρδιά είναι ένα από τα βασικά όργανα του ανθρώπινου οργανισμού και ως εκ τούτου είναι ιδιαίτερα σημαντική η ύπαρξη μεθόδων παρακολούθησης της λειτουργίας της. Η μέτρηση των παλμών της καρδιάς είναι ένας από τους τρόπους με τους οποίους μπορούμε να ελέγξουμε την καλή λειτουργία της ίδιας της καρδιάς αλλά και του κυκλοφορικού συστήματος.

Η παρούσα κατασκευή, είναι μια συσκευή με την οποία ο ενδιαφερόμενος έχει την δυνατότητα να μετρήσει γρήγορα και εύκολα τον καρδιακό του ρυθμό. Υπάρχουν πολλές συσκευές που παρέχουν αυτή την δυνατότητα, η συγκεκριμένη συσκευή ωστόσο βασιζόμενη στην αρχή της φωτοπληθυσμογραφίας, καταγράφει τον καρδιακό ρυθμό από το δάχτυλο με την βοήθεια ενός αισθητήρα ανακλώμενου υπεριώδους φωτός .

Το σύστημα μέτρησης παλμών που σχεδιάστηκε βασίζεται σε μια διάταξη που ανιχνεύει την κυκλοφορία του αίματος σε μια αρτηρία του δαχτύλου αφού με κάθε χτύπο της καρδιάς το αίμα διανέμεται σε όλο το σώμα. Έτσι, η ένταση του ανακλώμενου από το δέρμα υπεριώδους φωτός θα αυξομειώνεται συνεχώς και περιοδικά ανάλογα με το αν περνά αίμα ή όχι από την αρτηρία εκείνη την στιγμή. Η κατασκευή καταγράφει αυτές τις διακυμάνσεις, φιλτράρει και ενισχύει το σήμα κατάλληλα και το οδηγεί στην είσοδο του μικροεπεξεργαστή όπου υπολογίζουμε πλέον τον καρδιακό ρυθμό.

Στην συνέχεια ο καρδιακός ρυθμός εμφανίζεται σε μια οθόνη LCD μετρημένος σε παλμούς ανά λεπτό (beats per minute BPM) και αποθηκεύεται στην μνήμη του μικροεπεξεργαστή. Ο ενδιαφερόμενος έχει την δυνατότητα να συνδέσει την συσκευή με τον υπολογιστή μέσω της θύρας USB και να επεξεργαστεί τις μετρήσεις που έχει πάρει για να παρακολουθεί την πορεία της υγείας του.

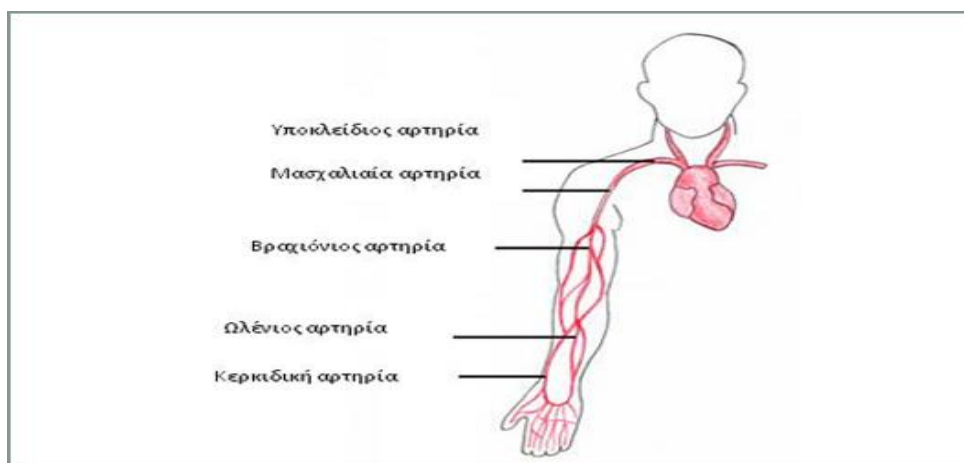
Με βάση τα προηγούμενα, χωρίσαμε την κατασκευή με τρία μέρη :

- 1) Κύκλωμα αισθητήρα για την απόκτηση των μετρήσεων.
- 2) Φίλτρο για την αποκοπή ανεπιθύμητων χαρακτηριστικών του σήματος και για ενίσχυση.
- 3) Μικροεπεξεργαστής που επικοινωνεί με οθόνη LCD και θύρα USB.

1.Η ΚΑΡΔΙΑ ΚΑΙ ΤΟ ΚΥΚΛΟΦΟΡΙΚΟ ΣΥΣΤΗΜΑ

1.1 Το κυκλοφορικό σύστημα

Το κυκλοφορικό σύστημα είναι ο μηχανισμός που αντλεί και μεταφέρει το αίμα σε ολόκληρο το σώμα. Αποτελείται από την καρδιά, τις αρτηρίες, τα τριχοειδή αγγεία και τις φλέβες. Η καρδιά είναι το σημαντικότερο κομμάτι του συστήματος καθώς αντλεί συνεχώς αίμα το οποίο φτάνει σε όλα τα μέρη του σώματος μας. Είναι ένας μυς με μέγεθος όσο περίπου η γροθιά ενός ανθρώπου και αποτελείται από δύο κόλπους και δύο κοιλίες που επικοινωνούν μεταξύ τους με τις βαλβίδες. Οι κόλποι και οι κοιλίες συστέλλονται και διαστέλλονται με συγκεκριμένο τρόπο δημιουργώντας τους καρδιακούς παλμούς. Κάθε καρδιακός παλμός διαρκεί από το τέλος μιας καρδιακής συστολής μέχρι και το τέλος της επόμενης. Το αίμα κυκλοφορεί από την καρδιά σε όλα τα όργανα μέσω του αγγειακού συστήματος μεγάλης κυκλοφορίας με τον ρυθμό που ορίζει ο καρδιακός παλμός. Το ίδιο το κυκλοφορικό σύστημα λειτουργεί σαν ένα αυτορυθμιζόμενο υδραυλικό σύστημα, στο οποίο η παροχή της αντλίας, δηλαδή της καρδιάς, μεταβάλλεται κατά τρόπο ώστε η πίεση του αίματος στην αορτή της καρδιάς να διατηρείται σε ένα σταθερό, προκαθορισμένο επίπεδο σε περίπτωση αυξημένης λειτουργίας των ιστών και των οργάνων αλλά και αντίθετα σε περίπτωση ηρεμίας.



Σχήμα 1.1 – Μέρος του κυκλοφορικού συστήματος.

1.2 Βιολογικά σήματα

Το βιολογικό σήμα (ή βιοσήμα) είναι χωρική, χρονική, ή χωροχρονική καταγραφή ενός βιολογικού γεγονότος όπως μια παλλόμενη καρδιά ή ένας συσπώμενος μυς. Η ηλεκτρική, χημική και μηχανική δραστηριότητα που εμφανίζεται κατά την διάρκεια αυτού του βιολογικού γεγονότος παράγει συχνά σήματα που μπορούν και μετρηθούν και να αναλυθούν. Τα βιολογικά σήματα επομένως, περιέχουν πληροφορίες που μπορούν να χρησιμοποιηθούν για να εξηγήσουν τους φυσιολογικούς μηχανισμούς που υποκρύπτονται σε ένα συγκεκριμένο βιολογικό γεγονός ή σύστημα.

Τα βιοσήματα μπορούν να αποκτηθούν με ποικίλους τρόπους όπως από το στήθοσκόπιο που χρησιμοποιεί ένας γιατρός για ακούσει τον ήχο της καρδιάς ενός ασθενή ή με την χρήση ιδιαίτερα σύνθετων και τεχνολογικά προηγμένων βιοϊατρικών οργάνων. Στις περισσότερες περιπτώσεις δεν αρκεί απλά η απόκτηση του βιολογικού σήματος. Πρέπει να αναλυθούν για να ανακτηθούν πληροφορίες από αυτά. Οι βασικές μέθοδοι ανάλυσης σημάτων δηλαδή η ενίσχυση, το φιλτράρισμα, η ψηφιοποίηση, η επεξεργασία και η αποθήκευση μπορούν να εφαρμοστούν σε πολλά βιολογικά σήματα. Αυτές οι τεχνικές ολοκληρώνονται γενικά με την χρήση υπολογιστών. Επιπρόσθετα σε αυτές τις διαδικασίες, έχουν αναπτυχθεί διάφορες ψηφιακές μέθοδοι για την ανάλυση των βιολογικών σημάτων. Σε αυτές περιλαμβάνονται ο υπολογισμός μέσου όρου σημάτων, η ανάλυση κυματιδίων και οι τεχνικές τεχνητής νοημοσύνης.

1.3 Καρδιακός ρυθμός

Ο καρδιακός ρυθμός είναι το πλήθος των παλμών της καρδιάς στη μονάδα του χρόνου και εκφράζεται σε παλμούς ανά λεπτό (beats per minute BPM). Ο βασικός παράγοντας που επηρεάζει τον ρυθμό λειτουργίας της καρδιάς είναι η κατάσταση του αυτόνομου νευρικού συστήματος, το τμήμα του νευρικού συστήματος που ρυθμίζει την λειτουργία των σπλάχνων. Το αυτόνομο νευρικό σύστημα επηρεάζεται από διάφορες συναισθηματικές αντιδράσεις, την γενικότερη φυσική κατάσταση αλλά και από την κατάσταση στην οποία βρισκόμαστε κατά την μέτρηση του καρδιακού ρυθμού. Επομένως, κάθε άνθρωπος έχει διαφορετικό φυσιολογικό ρυθμό. Η φυσική δραστηριότητα επηρεάζει αυτήν την ισορροπία με αποτέλεσμα ο καρδιακός ρυθμός

ενός ατόμου να μεταβάλλεται κατά την διάρκεια της ημέρας. Εξάλλου, το αυτόνομο νευρικό σύστημα είναι φτιαγμένο έτσι ώστε να βοηθά την καρδιά να λειτουργεί κατάλληλα σε κάθε περίπτωση.

Ο φυσιολογικός καρδιακός ρυθμός ενός ενήλικα σε κατάσταση ηρεμίας κυμαίνεται από 60 έως 100 παλμούς ανά λεπτό. Μικρές διακυμάνσεις παρουσιάζονται ανάλογα με την ηλικία και την φυσική κατάσταση. Ωστόσο, στους αθλητές παρατηρείται χαμηλότερος καρδιακός ρυθμός ενώ αντίθετα τα νεογέννητα εμφανίζουν καρδιακό ρυθμό της τάξης των 150 bpm. Καρδιακοί ρυθμοί με μεγάλες αποκλίσεις από το φυσιολογικό ίσως κρύβουν κάποιο παθολογικό αίτιο. Ρυθμός πολύ χαμηλότερος του φυσιολογικού είναι ένδειξη βραδυκαρδίας ενώ αν είναι υψηλότερος δείχνει ταχυκαρδία.

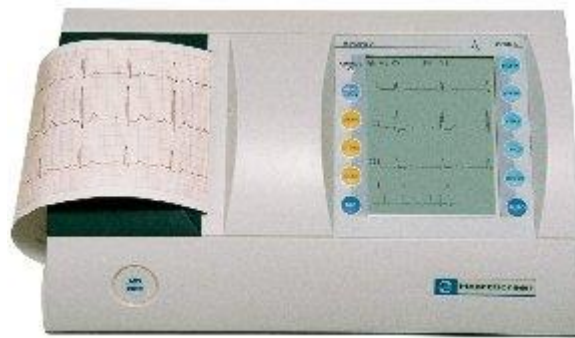
Πίνακας 1.1 – Ο φυσιολογικός καρδιακός ρυθμός με βάση την ηλικία.

Ηλικία	Ρυθμός (παλμοί ανά λεπτό)
Νεογέννητο	150
3 μηνών	140
6 μηνών	130
1 έτους	120
2 ετών	115
3 ετών	100
4 ετών	100
6 ετών	100
8 ετών	90
12 ετών	85
Ενήλικες	60 - 100

1.4 Τρόποι μέτρησης

Παραδοσιακά, η μέτρηση του καρδιακού ρυθμού γίνεται τοποθετώντας τον αντίχειρα πάνω σε μια αρτηρία και μετρώντας τους παλμούς για δεκαπέντε δευτερόλεπτα. Πολλαπλασιάζοντας με τέσσερα έχουμε τον καρδιακό ρυθμό σε bpm. Η ακριβής μέτρηση του ρυθμού μπορεί να γίνει με ηλεκτροκαρδιογράφημα (ECG), το οποίο καταγράφει την ηλεκτρική δραστηριότητα των μυών της παλλόμενης καρδιάς με την βοήθεια του ηλεκτροκαρδιογράφου.

Στο εμπόριο κυκλοφορούν συσκευές μέτρησης που χρησιμοποιούν ζώνες με ηλεκτρόδια που τοποθετούνται στον θώρακα για μεγαλύτερη ακρίβεια μετρήσεων. Υπάρχουν ακόμη συσκευές με την μορφή ρολογιού που είναι σε θέση να πάρουν ακριβείς μετρήσεις, όμως το κόστος τους δεν τις καθιστά την πιο συμφέρουσα επιλογή. Μπορούμε επίσης να πάρουμε μετρήσεις με την χρήση αισθητήρα λοβού αυτιού και στην περίπτωση μας, με χρήση αισθητήρα στο δάχτυλο.



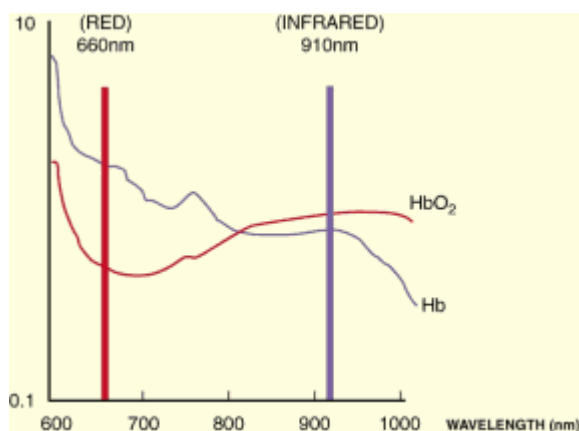
Σχήμα 1.2 – Ο ηλεκτροκαρδιογράφος

2.Η ΦΩΤΟΠΛΥΘΙΣΜΟΓΡΑΦΙΑ

2.1 Βασικές αρχές

Η φωτοπληθυσμογραφία είναι βασισμένη στον προσδιορισμό των οπτικών ιδιοτήτων μιας επιλεγμένης περιοχής του δέρματος. Για αυτό το λόγο ένα μη ορατό υπέρυθρο φως εκπέμπεται στο δέρμα. Το φως απορροφάται περισσότερο ή λιγότερο ανάλογα με τον όγκο του αίματος στο δέρμα. Συνεπώς, το διαχεόμενο φως είναι ανάλογο της αλλαγής του όγκου του αίματος. Οι μεταβολές του όγκου του αίματος μπορούν να καθοριστούν με την μέτρηση του φωτός που ανακλάται και την αξιοποίηση των οπτικών ιδιοτήτων του ιστού και του αίματος.

Χρησιμοποιείται πηγή φωτός με μήκος κύματος κοντά στο υπέρυθρο καθώς έτσι έχουμε την καλύτερη διαμόρφωση σήματος εξαιτίας της απορρόφησης του υπέρυθρου φωτός από την αιμοσφαιρίνη στο αίμα. Η αρτηριακή φωτοπληθυσμογραφία μπορεί να μετρήσει την συγκέντρωση του αίματος στην δερματική μικροκυκλοφορία και εφόσον οι αλλαγές στον όγκο του αίματος είναι σε συγχρονισμό με τον καρδιακό παλμό, χρησιμοποιείται για τον υπολογισμό του καρδιακού παλμού αφού η πληροφορία βρίσκεται στον χρόνο και όχι στο πλάτος ή την μορφή του σήματος. Η εύκολη εφαρμογή στα δάχτυλα, καθιστά αυτή την μέθοδο ιδιαίτερα χρήσιμη για την διάγνωση αρτηριακών παθήσεων στα δάχτυλα, την διάγνωση των λειτουργικών διαταραχών στην ροή του αίματος και τη μέτρηση της περιφερειακής πίεσης του αίματος ακόμα και στα δάχτυλα.

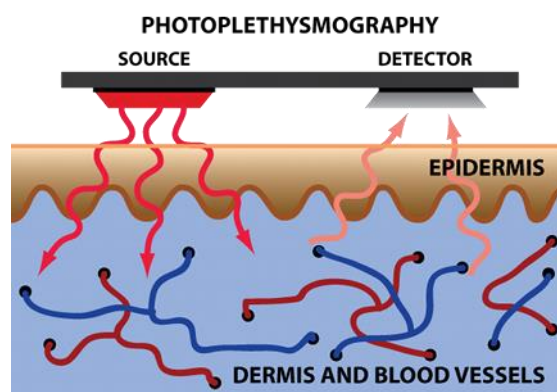


Σχήμα 2.1 – Απορρόφηση του υπέρυθρου φωτός από την αιμοσφαιρίνη.

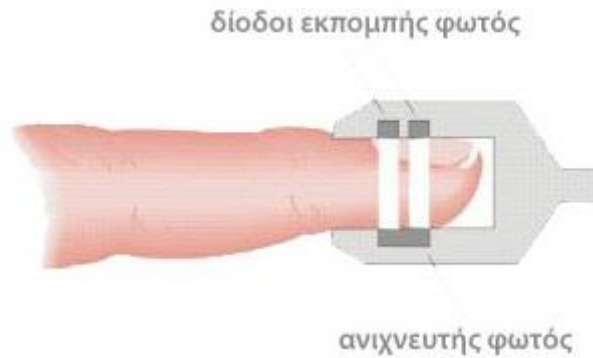
2.2 Τύποι αρτηριακής φωτοπληθυσμογραφίας

Υπάρχουν δύο διαφορετικοί τύποι της αρτηριακής φωτοπληθυσμογραφίας, ο έλεγχος αντανάκλασης και ο έλεγχος μετάδοσης. Στον έλεγχο αντανάκλασης, ο οποίος μπορεί να χρησιμοποιηθεί και για την φλεβική δοκιμασία, η εκπομπή του φωτός και τα αισθητήρια μέρη βρίσκονται το ένα δίπλα στο άλλο σε ένα τμήμα. Οι φωτοαισθητήρες ανιχνεύουν το φως το οποίο διασκορπίζεται από τον ιστό του δέρματος. Λόγω της ανατομίας του σώματος, οι αισθητήρες φωτοπληθυσμογραφίας μπορούν να ανιχνεύουν τα κύματα σφυγμού μόνο στις περιοχές που περιέχουν πολλές αρτηριοφλεβώδεις αναστομώσεις, όπως τα δάχτυλα των ποδιών και των χεριών και οι λοβοί των αυτιών. Αυτές ωστόσο είναι διαφορετικές θέσεις από αυτές που χρησιμοποιούνται στο φλεβικό τεστ με φωτοπληθυσμογραφία.

Ο δεύτερος τύπος της αρτηριακής φωτοπληθυσμογραφίας είναι ο έλεγχος μετάδοσης. Σε αυτόν τον έλεγχο οι φωτοανιχνευτές βρίσκονται στην αντίθετη πλευρά των φωτοεκπεμπόμενων μερών. Ο ιστός βρίσκεται μεταξύ τους. Αυτό περιορίζει τον τομέα της εφαρμογής στις θέσεις όπου το φως μπορεί να διαπεράσει όλο τον ιστό, όπως στα δάχτυλα και τα αυτιά. Σε αντίθεση με τον έλεγχο αντανάκλασης, οι κύριες πηγές παλμού περιέχουν επίσης τα μεγάλα αγγεία.



Σχήμα 2.2 – Θέση αισθητήρων στον έλεγχο αντανάκλασης.



Σχήμα 2.3 – Θέση αισθητήρων στον έλεγχο μετάδοσης.

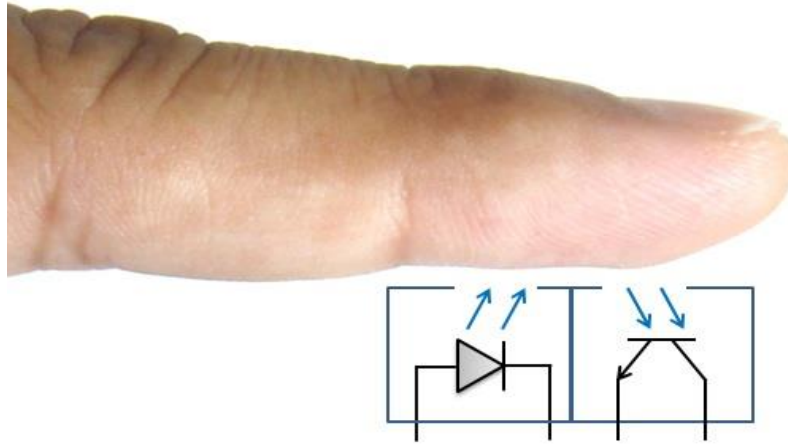
2.3 Μορφή του σήματος

Το σήμα που δημιουργείται κατά την αρτηριακή φωτοπληθυσμογραφία αποτελείται από μια AC και μια DC συνιστώσα. Η AC συνιστώσα δημιουργείται από τις αλλαγές στον όγκο του αίματος, σύγχρονες με τον καρδιακό παλμό, και μπορεί να χρησιμοποιηθεί ως πηγή πληροφοριών για τον καρδιακό παλμό. Η AC συνιστώσα αυτή βρίσκεται υπερθετημένη πάνω στην DC συνιστώσα η ύπαρξη της οποίας έχει να κάνει με τους ιστούς, την αναπνοή και την δραστηριότητα του νευρικού συστήματος.

Η αφαίρεση της DC συνιστώσας είναι απαραίτητη ώστε να μπορέσουμε να μετρήσουμε το εναλλασσόμενο σήμα με υψηλό SNR. Ωστόσο, η AC συνιστώσα αποτελεί μικρό μέρος του συνολικού σήματος με αποτέλεσμα να χρειάζεται μεγάλη ενίσχυση για να εξάγουμε τις πληροφορίες που θέλουμε.

2.4 Αισθητήρες αρτηριακής φωτοπληθυσμογραφίας

Οι σημερινοί αισθητήρες φωτοπληθυσμογραφίας χρησιμοποιούν την χαμηλού κόστους τεχνολογία των ημιαγωγών. Ως εκπομπός φωτός χρησιμοποιείται ένα υπέρυθρο LED, το φως του οποίου εκπέμπεται προς το δέρμα. Ωστόσο, υπάρχουν περιορισμοί στο κατά πόσο το φως από ένα LED διαπερνά το δέρμα, με αποτέλεσμα η μέθοδος να χρησιμοποιείται σε περιορισμένα σημεία του σώματος όπως τα δάχτυλα και οι λοβοί των αυτιών. Το ποσοστό του φωτός που ανακλάται λόγω της ροής του αίματος φτάνει στον ανιχνευτή φωτός, ένα φωτοτρανζίστορ ή σε κάποιες περιπτώσεις μία φωτοδίοδο.



Σχήμα 2.4 – Ο αισθητήρας αποτελείται από ένα LED και ένα φωτοτρανζίστορ.

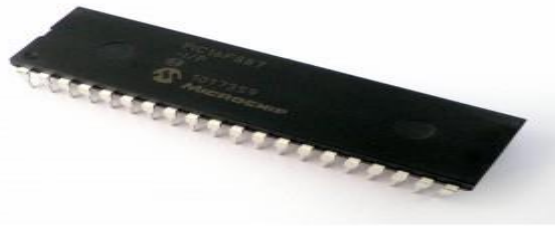
3.ΤΑ ΕΠΙΜΕΡΟΥΣ ΣΤΟΙΧΕΙΑ ΚΑΙ ΟΙ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ

3.1.Ο μικροελεγκτής PIC16F887

Ο μικροελεγκτής που χρησιμοποιήθηκε στην κατασκευή της συσκευής είναι ο PIC 16F887 της Microchip. Επιλέξαμε αυτόν τον μικροελεγκτή λόγω μερικών συγκεκριμένων χαρακτηριστικών του. Ο PIC 16F887 διαθέτει τρεις χρονιστές/απαριθμητές, τον Timer0 ο οποίος είναι ένας οκταψήφιος χρονιστής, τον Timer1 που διαθέτει δεκαέξι ψηφία και τον Timer2 ο οποίος είναι επίσης οκταψήφιος. Επίσης μας παρέχει αρκετή μνήμη EEPROM μιας και διαθέτει 256 θέσεις μνήμης που κυμαίνονται από την 00h έως την FFh. Ο κατασκευαστής αναφέρει πως η μνήμη EEPROM του PIC 16F887 μπορεί να αντέξει μέχρι ένα εκατομμύριο εγγραφές σε αυτήν (1.000.000 write EEPROM endurance). Ένα άλλο χαρακτηριστικό χρήσιμο για την σύνδεση της συσκευής με τον υπολογιστή, είναι η ύπαρξη του USART, μίας μονάδας ασύγχρονης σειριακής επικοινωνίας που επιτρέπει την μεταφορά από και προς περιφερειακές συσκευές. Η ύπαρξη τεσσάρων ports καθιστά εύκολη την αντιστοίχιση κάθε λειτουργίας της συσκευής σε μια από αυτές (μέτρηση, απεικόνιση, επεξεργασία αποθηκευμένων μετρήσεων, μεταφορά δεδομένων). Ακολουθεί συνοπτικός πίνακας με τα πιο χρήσιμα χαρακτηριστικά του μικροελεγκτή.

PIC16F887

Program memory-Flash (words)	Data memory-SRAM (bytes)	Data memory-EEPROM (bytes)	I/O	10-bit ADC (ch)
8192	368	256	35	14
ECCP/CCP	EUSART	MSSP	Comparators	Timers 8-bit/16bit
1/1	1	1	2	2/1



Σχήμα 3.1 – Ο μικροελεγκτής PIC16F887.

3.1.1 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής : Χρονιστής/μετρητής Timer0, OPTION

Για την μέτρηση του καρδιακού ρυθμού από τον μικροελεγκτή, γίνεται χρήση του μετρητή timer0. Ο timer0 είναι ένας μετρητής η τιμή του οποίου αυξάνεται αυτόματα. Είναι δυνατό να ελέγχεται από το clock του μικροελεγκτή, περίπτωση στην οποία λειτουργεί ως χρονιστής καθώς η τιμή του αυξάνεται με σταθερό ρυθμό, ή από ένα εξωτερικό σήμα, περίπτωση στην οποία λειτουργεί ως μετρητής και μετράει τις μεταβολές της κατάστασης ενός pin εισόδου. Η διαμόρφωση του timer0 για την επιλογή της λειτουργίας του γίνεται από τον καταχωρητή OPTION.

Οι timers χρησιμοποιούνται συνήθως για την δημιουργία των interrupts ενώ σε λειτουργία μετρητή η είσοδος T0CKI του μικροελεγκτή μπορεί να οδηγηθεί από μεταβαλλόμενη τάση (για παράδειγμα ένα ημίτονο ή έναν τετραγωνικό παλμό όπως στην περίπτωση μας) ακόμη και αν υπάρχει μια μικρή υπέρθεση θορύβου καθώς το σήμα δεν χρειάζεται να είναι αυστηρά TTL σήμα. Ανεξάρτητα από την λειτουργία του, η τιμή του timer0 βρίσκεται ανά πάσα στιγμή σε έναν οκταψηφίο καταχωρητή που ονομάζεται TMR0. Ο καταχωρητής αυτός μπορεί να διαβαστεί αλλά και να εγγραφεί, περίπτωση στην οποία του δίνουμε αρχική τιμή από την οποία ξεκινά να αυξάνεται.

Ο timer0 έχει χαρακτηριστικά που ελέγχονται από τον καταχωρητή OPTION. Σε λειτουργία χρονιστή προσαυξάνεται σε κάθε κύκλο μηχανής, ενώ ως μετρητής προσαυξάνεται σε κάθε μεταβολή που συναντά στην ακίδα T0CKI ενώ μπορούμε να επιλέξουμε το θετικό η αρνητικό μέτωπο του παλμού ως σκανδαλισμό για την προσαύξηση. Δύναται να γραφεί και να διαβαστεί. Παράγει σήμα διακοπής κατά την υπερχείλιση από το FFh στο 00h. Τέλος, διαθέτει προγραμματιζόμενο μετρητή προκειμένου να μπορεί να προσαυξάνεται όχι σε κάθε κύκλο μηχανής αλλά κάθε δύο έως 256 κύκλους. Ακολουθεί πίνακας με τις δυνατές επιλογές του καταχωρητή OPTION για την λειτουργία του timer0. Με βάση τον πίνακα αυτόν, η τιμή του καταχωρητή που θα χρησιμοποιήσουμε για το πρόγραμμα είναι σε δυαδικό η OPTION_REG=0b01111000.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{RBP\bar{U}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7						bit0	

Όπου: R=αναγνώσιμο ψηφίο, W=εγγράψιμο ψηφίο

Ψηφίο 7 $\overline{RBP\bar{U}}$: Ενεργοποίηση των pull-ups της PORTB.

1 = Απενεργοποίηση των pull-ups, 0 = Ενεργοποίηση των pull-ups

Ψηφίο 6 INTEDG : Επιλογή μετώπου ενεργοποίησης διακοπών.

1 = Διακοπή με θετικό μέτωπο στον ακροδέκτη INT , 0 = Διακοπή με αρνητικό μέτωπο

Ψηφίο 5 T0CS : Επιλογή ρολογιού προσαύξησης του TMR0.

1 = Εξωτερικό σήμα στον ακροδέκτη T0CKI, 0 = Χρήση εσωτερικού ρολογιού

Ψηφίο 4 T0SE : Επιλογή μετώπου για τον TMR0.

1 = Προσαύξηση κατά την μετάβαση από high σε low στο T0CKI, 0 = Προσαύξηση κατά την μετάβαση από low σε high στο T0CKI

Ψηφίο 3 PSA : Εκχώρηση του προμετρητή (Prescaler Assignment).

1 = Εκχώρηση στον WDT, 0 = Εκχώρηση στον TMR0

Ψηφία PS2-PS0 : Επιλογή μεγέθους προμετρητή.

PS2 PS1 PS0	Ρυθμός TMR0	Ρυθμός WDT
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

Επομένως, για την μέτρηση του τετραγωνικού παλμού που φτάνει στον μικροελεγκτή επιλέγουμε την χρήση εξωτερικού σήματος στον ακροδέκτη T0CKI για να έχουμε λειτουργία μετρητή, προσαύξηση κατά την μετάβαση από high σε low, εκχώρηση του προμετρητή στον WDT και ρυθμό WDT 1:1 καθώς θέλουμε να μετράμε ανά έναν παλμό. Αυτές οι ρυθμίσεις δίνουν στον καταχωρητή την τιμή **OPTION_REG=0b01111000**.

3.1.2 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής : OSCCON

Ο μικροελεγκτής PIC 16F887 μας δίνει την δυνατότητα να χρησιμοποιήσουμε είτε έναν εξωτερικό ταλαντωτή είτε τον εσωτερικό. Ο εσωτερικός ταλαντωτής μπορεί να παράγει συχνότητα 31KHz, 125KHz, 250KHz, 500KHz, 1MHz, 2MHz, 4MHz, και 8MHz. Για την επιλογή ταλαντωτή και την ρύθμιση του υπάρχει ο οκταψήφιος καταχωρητής OSCCON. Ακολουθεί πίνακας με τις δυνατές επιλογές για τον καταχωρητή OSCCON.

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
-	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS
bit7							bit0

Ψηφίο 7 : Μη χρησιμοποιούμενο ψηφίο.

Ψηφία 6-4 IRCF<2:0> : Επιλογή συχνότητας εσωτερικού ταλαντωτή.

111 = 8 MHz, 110 = 4 MHz (default), 101 = 2 MHz, 100 = 1 MHz, 011 = 500 kHz, 010 = 250 kHz, 001 = 125 kHz, 000 = 31 kHz

Ψηφίο 3 OSTS : Δηλώνει την χρήση εσωτερικού ή εξωτερικού ταλαντωτή.

1 = Χρήση εξωτερικού ταλαντωτή, 0 = Χρήση εσωτερικού ταλαντωτή

Ψηφίο 2 HTS : Δηλώνει την αστάθεια του εσωτερικού ταλαντωτή σε υψηλές συχνότητες.

1 = Ο ταλαντωτής δεν είναι ασταθής, 0 = Ο ταλαντωτής είναι ασταθής

Ψηφίο 1 LTS : Δηλώνει την αστάθεια του εσωτερικού ταλαντωτή σε χαμηλές συχνότητες.

1 = Ο ταλαντωτής δεν είναι ασταθής, 0 = Ο ταλαντωτής είναι ασταθής.

Ψηφίο 0 SCS : Επιλογή χρήσης εσωτερικού η εξωτερικού ταλαντωτή.

1 = Χρήση εσωτερικού ταλαντωτή, 0 = Χρήση εξωτερικού ταλαντωτή.

Άρα, με βάση τα παραπάνω, για την χρήση εξωτερικού ταλαντωτή με κρύσταλλο στα 20MHz, συχνότητα απαραίτητη για την δημιουργία baud rate 9600 για την επικοινωνία με τον υπολογιστή, ο καταχωρητής OSCCON θα έχει την τιμή **OSCCON=0b00001000**.

3.1.3 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής : TXSTA, RCSTA και BAUDCTL για την ασύγχρονη σειριακή επικοινωνία

Η λειτουργία της σειριακής επικοινωνίας με περιφερειακές συσκευές ελέγχεται με τρεις καταχωρητές. Ο πρώτος αφορά την αποστολή και ονομάζεται TXSTA, ο δεύτερος ελέγχει την λήψη και λέγεται RCSTA, ενώ ο τρίτος με την ονομασία BAUDCTL ο οποίος καθορίζει το baud rate με το οποίο γίνεται η επικοινωνία. Και οι τρεις καταχωρητές παρέχουν πλήθος επιλογών ανάλογα με το αν θέλουμε επικοινωνία με σύγχρονο ή ασύγχρονο τρόπο. Ακολουθούν πίνακες με τις επιλογές που προσφέρει ο κάθε καταχωρητής.

TXSTA : Transmit Status and Control Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
bit7						bit0	

Ψηφίο 7 CSRC : Επιλογή ρολογιού.

Σε ασύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε σύγχρονο τρόπο λειτουργίας : 1= Τρόπος Master, 0= Τρόπος Slave

Ψηφίο 6 TX9 : Επιλογή 9-ψήφιας επιλογής.

1 = Επιλέγεται 9-ψήφια αποστολή, 0 = Επιλέγεται 8-ψήφια αποστολή

Ψηφίο 5 TXEN : Ενεργοποίηση της αποστολής.

1 = Ενεργοποίηση της αποστολής, 0 = Απενεργοποίηση της αποστολής

Ψηφίο 4 SYNC : Επιλογή τρόπου λειτουργίας του USART.

1 = Σύγχρονος τρόπος, 0 = Ασύγχρονος τρόπος

Ψηφίο 3 SENDB : Αποστολή χαρακτήρα διακοπής.

Σε σύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε ασύγχρονο τρόπο : 1= Ενεργοποίηση αποστολής χαρακτήρα διακοπής, 0= Απενεργοποίηση αποστολής χαρακτήρα διακοπής.

Ψηφίο 2 BRGH : Επιλογή υψηλού Baud Rate.

Σε σύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε ασύγχρονο τρόπο : 1= Υψηλή ταχύτητα, 0= Χαμηλή ταχύτητα

Ψηφίο 1 TRMT : Κατάσταση του καταχωρητή ολίσθησης αποστολής.

1 = TSR άδειο, 0 = TSR γεμάτο

Ψηφίο 0 TX9D : Είναι το ένατο ψηφίο δεδομένων αποστολής.

Με βάση τα παραπάνω και λαμβάνοντας υπόψη πως θέλουμε ασύγχρονη επικοινωνία με baud rate 9600 και αποστολή οκταψήφιων δεδομένων, ο καταχωρητής TXSTA έχει την τιμή **TXSTA=0b00101110**.

RCSTA – Receive Status and Control Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit7							bit0

Ψηφίο 7 SPEN : Ενεργοποίηση σειριακής πόρτας.

1 = Ενεργοποίηση σειριακής πόρτας , 0 = Απενεργοποίηση σειριακής πόρτας

Ψηφίο 6 RX9 : Ενεργοποίηση εννιαψήφιας λήψης.

1 = Επιλογή εννιαψήφιας λήψης , 0 = επιλογή οκταψήφιας λήψης

Ψηφίο 5 SREN : Ενεργοποίηση απλής λήψης.

Σε ασύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε σύγχρονο τρόπο :
1 = Ενεργοποίηση, 0 = Απενεργοποίηση

Ψηφίο 4 CREN : Ενεργοποίηση συνεχούς λήψης.

Σε ασύγχρονο τρόπο : 1= Ενεργοποίηση, 0= Απενεργοποίηση. Σε σύγχρονο τρόπο :
1= Ενεργοποίηση μέχρι να μηδενιστεί το ψηφίο CREN, 0= Απενεργοποίηση

Ψηφίο 3 ADDEN : Ενεργοποίηση εντοπισμού διεύθυνσης για εννιαψήφια λήψη.

Για οκταψήφια λήψη είναι αδιάφορο. Για εννιαψήφια : 1= Ενεργοποίηση , 0= Απενεργοποίηση

Ψηφίο 2 FERR : Σφάλμα πλαισίου.

1 = Σφάλμα πλαισίου , 0 = Όχι σφάλμα πλαισίου

Ψηφίο 1 OERR : Σφάλμα υπερχείλισης.

1 = Σφάλμα υπερχείλισης , 0 = Όχι σφάλμα υπερχείλισης

Ψηφίο 0 RX9D : Ένατο ψηφίο λήψης δεδομένων.

Με βάση τα παραπάνω και λαμβάνοντας υπόψη πως θέλουμε ασύγχρονη επικοινωνία με baud rate 9600 και λήψη οκταψήφιων δεδομένων, ο καταχωρητής RCSTA έχει την τιμή **RCSTA=0b10010000**.

BAUDCTL – Baud Rate Control register

R-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	-	SCKP	BRG16	-	WUE	ABDEN
bit7							bit0

Ψηφίο 7 ABDOVF : Έλεγχος υπερχείλισης του auto-baud timer.

Σε σύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε ασύγχρονο τρόπο : 1= Ο timer υπερχείλισε , 0= Ο timer δεν υπερχείλισε

Ψηφίο 6 RCIDL : Έλεγχος αδράνειας του δέκτη.

Σε σύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε ασύγχρονο τρόπο : 1= Ο δέκτης βρίσκεται σε αδράνεια , 0= Ο δέκτης δέχεται δεδομένα

Ψηφίο 5 Δεν χρησιμοποιείται.

Ψηφίο 4 SCKP : Πολικότητα δεδομένων προς αποστολή.

Σε ασύγχρονο τρόπο : 1= Αποστολή μη αναστραμμένων δεδομένων , 0= Αποστολή αναστραμμένων δεδομένων. Σε σύγχρονο τρόπο : 1= Συντονισμός δεδομένων με το θετικό μέτωπο του clock, 0= Συντονισμός δεδομένων με το αρνητικό μέτωπο του clock

Ψηφίο 3 BRG16 : Επιλογή 16-ψήφιας γεννήτριας baud rate.

1 = Χρήση 16-ψήφιας γεννήτριας , 0 = Χρήση οκταψήφιας γεννήτριας

Ψηφίο 2 Δεν χρησιμοποιείται

Ψηφίο 1 WUE : Επιλογή λειτουργίας του δέκτη.

Σε σύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε ασύγχρονο τρόπο : 1= Ο δέκτης περιμένει ένα αρνητικό μέτωπο , 0 = Ο δέκτης λειτουργεί κανονικά.

Ψηφίο 0 ABDEN : Επιλογή εντοπισμού auto-baud.

Σε σύγχρονο τρόπο λειτουργίας είναι αδιάφορο. Σε ασύγχρονο τρόπο : 1= Ενεργοποίηση εντοπισμού auto-baud , 0= Απενεργοποίηση εντοπισμού auto-baud

Για να έχουμε το επιθυμητό baud rate θα πρέπει να χρησιμοποιήσουμε κρύσταλλο 20MHz, να θέσουμε SYNC=0, BRGH=1 και BRG16=0. Σύμφωνα με το datasheet του κατασκευαστή αυτός ο συνδυασμός θα μας δώσει πραγματικό baud rate 9615. Το ποσοστιαίο σφάλμα θα είναι 0.16% σύμφωνα με το datasheet.

Με βάση τα παραπάνω και λαμβάνοντας υπόψη πως θέλουμε ασύγχρονη επικοινωνία με baud rate 9600 και επικοινωνία με οκταψήφια δεδομένα, ο καταχωρητής BAUCTL έχει την τιμή **BAUDCTL=0b00000000**.

3.1.4 Καταχωρητές που χρησιμοποιούνται για την λειτουργία της συσκευής : Οι καταχωρητές εισόδου/εξόδου

Όπως αναφέρθηκε παραπάνω, κάθε μία από τις τέσσερις πόρτες του μικροελεγκτή έχει αναλάβει μία από τις τέσσερις λειτουργίες της συσκευής οι οποίες είναι μέτρηση, απεικόνιση, επεξεργασία αποθηκευμένων μετρήσεων και μεταφορά δεδομένων. Οι καταχωρητές που καθορίζουν την λειτουργία της κάθε πόρτας έχουν συγκεκριμένες τιμές με βάση τις ανάγκες της συσκευής. Οι τιμές των καταχωρητών είναι οι εξής :

TRISA=0b00010010

Το ψηφίο 0 ορίζεται σαν έξοδος για να οδηγεί την βάση του τρανζίστορ ώστε αυτό να λειτουργεί σαν διακόπτης λειτουργίας του αισθητήρα. Το ψηφίο 1 είναι είσοδος για να αντιλαμβάνεται το πάτημα του pushbutton-Start ενώ το ψηφίο 4 είναι είσοδος γιατί εκεί βρίσκεται η ακίδα T0CKI από την κατάσταση της οποίας ελέγχεται η αύξηση του timer0.

TRISB=0b00000000

Ολόκληρη η PORTB είναι έξοδος καθώς με αυτήν επικοινωνεί η LCD οθόνη για την απεικόνιση των μετρήσεων.

TRISC=0b10000000

Το ψηφίο 6 είναι έξοδος γιατί αντιστοιχεί την ακίδα του transmitter για την αποστολή δεδομένων ενώ το ψηφίο 7 είναι είσοδος καθώς αντιστοιχεί την ακίδα του receiver για την λήψη δεδομένων.

TRISD=0b00000001

Η PORTD είναι αφιερωμένη στην επεξεργασία των αποθηκευμένων μετρήσεων. Για τον λόγο αυτό το ψηφίο 0 είναι είσοδος για να αντιλαμβάνεται το πάτημα του pushbutton-memory το οποίο παρέχει πρόσβαση στην μνήμη και δυνατότητα διαγραφής της.

3.2 Ο αισθητήρας

3.2.1 Χαρακτηριστικά ενός αισθητήρα

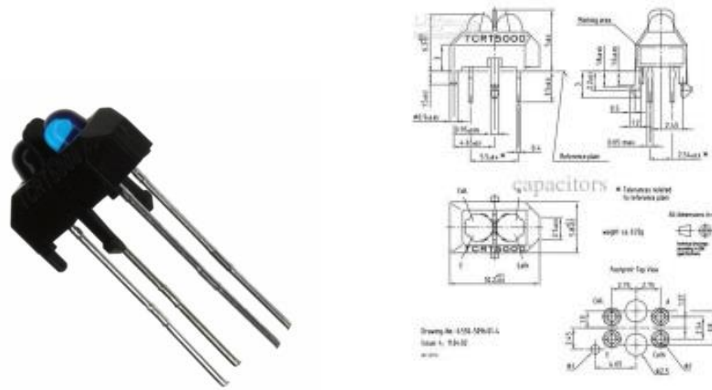
Για την διεκπεραίωση των μετρήσεων είναι απαραίτητη η χρήση κάποιου αισθητήρα. Αισθητήρας ονομάζεται μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος και παράγει από αυτό μία μετρήσιμη έξοδο. Για παράδειγμα, το υδραργυρικό θερμόμετρο μετατρέπει τη μετρούμενη θερμοκρασία σε διαστολή, η οποία μπορεί να αναγνωστεί από ένα βαθμονομημένο σωλήνα. Οι αισθητήρες χρησιμοποιούνται σε καθημερινά αντικείμενα, όπως κουμπιά ανελκυστήρων ευαίσθητα στην αφή και λάμπες φωτισμού που εκπέμπουν λαμπρότερα ή απαλότερα αγγίζοντας τη βάση τους. Υπάρχουν αναρίθμητες ακόμη χρήσεις που οι περισσότεροι άνθρωποι δεν αντιλαμβάνονται. Εφαρμογές τους συναντούμε στα αυτοκίνητα, σε μηχανές, στην αεροναυπηγική, την ιατρική, τη βιομηχανία και τη ρομποτική. Κάποια από τα κύρια μεγέθη ενός αισθητήρα ανεξάρτητα με το τι μετράει είναι:

1. **Το εύρος**, τα όρια στα οποία η συσκευή λειτουργεί αξιόπιστα.
2. **Ακρίβεια**, η εγγύτητα της τιμής εξόδου προς τη τιμή εισόδου.
3. **Σφάλμα**, η διαφορά ανάμεσα στη μετρούμενη τιμή και τη πραγματική τιμή.
4. **Ανοχή**, το μέγιστο σφάλμα που μπορεί να δημιουργήσει ο αισθητήρας.
5. **Διακριτική ικανότητα**, η μικρότερη αλλαγή τιμής εισόδου που μπορεί να ανιχνεύσει.
6. **Απόκριση**, ο χρόνος που απαιτείται για να λάβει τη τελική τιμή η έξοδος.
7. **Ολίσθηση**, η μεταβολή των χαρακτηριστικών του αισθητήρα με το χρόνο και το περιβάλλον.

Με τα παραπάνω χαρακτηριστικά γίνεται η επιλογή του κατάλληλου αισθητήρα ο οποίος θα μας δώσει επαρκή αποτελέσματα. Τα είδη των αισθητήρων είναι πολλά και γενικά χωρίζονται σε πολλές κατηγορίες ανάλογα με το φυσικό μέγεθος που χρησιμοποιούν σαν είσοδο. Έτσι, έχουμε αισθητήρες που χρησιμοποιούν το φως που υπάρχει στον χώρο σαν είσοδο, άλλοι αισθητήρες μετρούν την θερμοκρασία, την πίεση του περιβάλλοντος, την κίνηση και άλλα. Στην δικιά μας περίπτωση χρησιμοποιούμε έναν αισθητήρα που σαν είσοδο χρησιμοποιεί τις αλλαγές ποσότητας του φωτός. Το φως που μετράει είναι υπέρυθρο, καθώς εκπέμπεται από μια πηγή υπέρυθρου φωτός, και λαμβάνεται από έναν δέκτη υπέρυθρου φωτός. Επιλέγουμε το υπέρυθρο φως στην συγκεκριμένη περίπτωση καθώς μπορεί και διαπερνάει το δέρμα με μεγάλη ευκολία. Έτσι, έχουμε ένα αισθητήριο το οποίο μόλις τοποθετήσουμε το δάχτυλο μας πάνω του θα εκπέμψει υπέρυθρο φως και θα ανιχνεύσει το ανακλώμενο επηρεασμένο τώρα από την ροή του αίματος. Με λίγα λόγια θα

χρησιμοποιήσουμε ένα αισθητήριο αποτελούμενο από ένα υπέρυθρο LED και ένα φωτοτρανζίστορ υπέρυθρου.

3.2.2 Ο αισθητήρας TCRT5000L



Σχήμα 3.2 – Αισθητήριο TCRT5000L

Ο TCRT5000L είναι ένας αισθητήρας ή όπως ονομάζεται στο εμπόριο ένας αισθητήρας ανακαλούμενου φωτός με τρανζίστορ στην έξοδο (Reflective Optical Sensor with Transistor Output). Αποτελείται από έναν εκπομπό – LED ο οποίος εκπέμπει υπέρυθρη ακτινοβολία και τον δέκτη – Τρανζίστορ – που ανιχνεύει την υπέρυθρη ακτινοβολία που ανακλάται από ένα αντικείμενο μετατρέποντας την σε ηλεκτρικό ρεύμα, η ένταση του οποίου εξαρτάται από την ισχύ της ανακλώμενης ακτινοβολίας. Ο συγκεκριμένος αισθητήρας έχει ενσωματωμένο προστατευτικό ανάμεσα στον πομπό και τον δέκτη μην επιτρέποντας την μεταξύ τους άμεση επαφή. Με αυτόν τον τρόπο είμαστε σίγουροι ότι η ακτινοβολία που ανιχνεύεται από το τρανζίστορ είναι μόνο ανακλώμενη. Επίσης μιας και στο περιβάλλον υπάρχει υπέρυθρη ακτινοβολία είτε από τον ήλιο – φυσικό φώς – είτε από άλλες πηγές, ο αισθητήρας αυτός διαθέτει φίλτρο απόφραξης αυτής της ακτινοβολίας (DAYLIGHT BLOCKING FILTER INTEGRATED).

Πίνακας 3.1 – Χρήσιμα μεγέθη, Datasheet

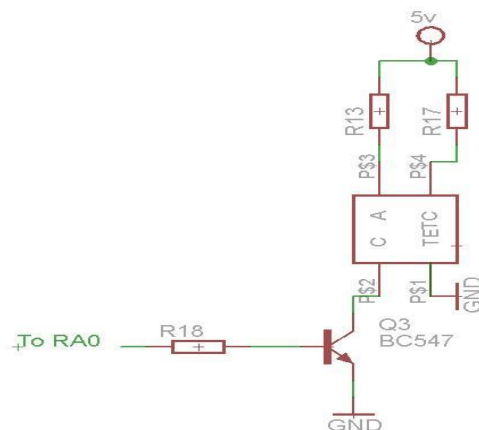
Μέγιστα Μεγέθη	Σύμβολο	Μέγιστη Τιμή
Ανάστροφη τάση	V_r	5 v
Τάση συλλέκτη εκπομπού	V_{CEO}	70 V
Τάση εκπομπού συλλέκτη	V_{ECO}	5 V
Ρεύμα συλλέκτη	I_C	100 mA

Άλλες εφαρμογές του αισθητήρα:

- Αισθητήρας θέσης για κωδικοποιητή άξονα
- Ανίχνευση από ανακλαστικό υλικό όπως χαρτί, κάρτες IBM, μαγνητικές ταινίες κλπ.
- Διακόπτης για περιορισμό μηχανικών κινήσεων σε συσκευές αναπαραγωγής βίντεο.
- Γενικές χρήσεις - όπου ο χώρος είναι περιορισμένος

3.2.3 Ο αισθητήρας στο κύκλωμα

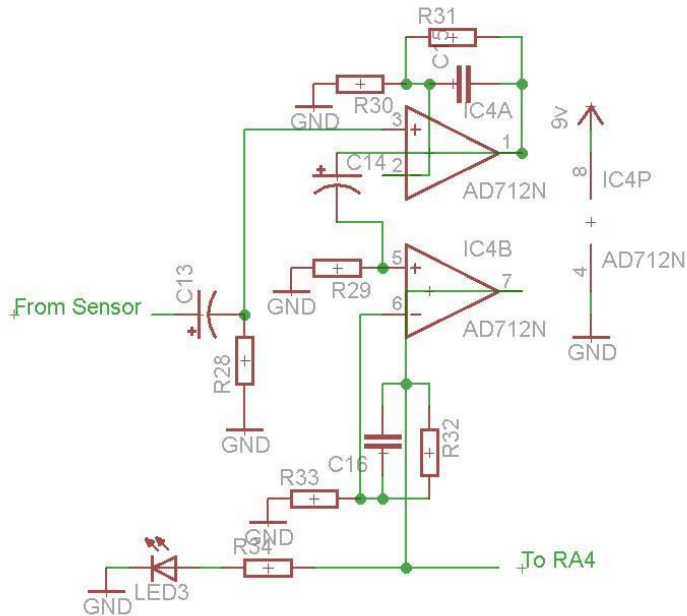
Στο κύκλωμα ο αισθητήρας θα λαμβάνει την ανακλώμενη από το δάχτυλο ακτινοβολία η οποία, αφού θα ενισχύεται και θα φιλτράρεται, θα μας παρέχει την κατάλληλη πληροφορία για τον καρδιακό ρυθμό.



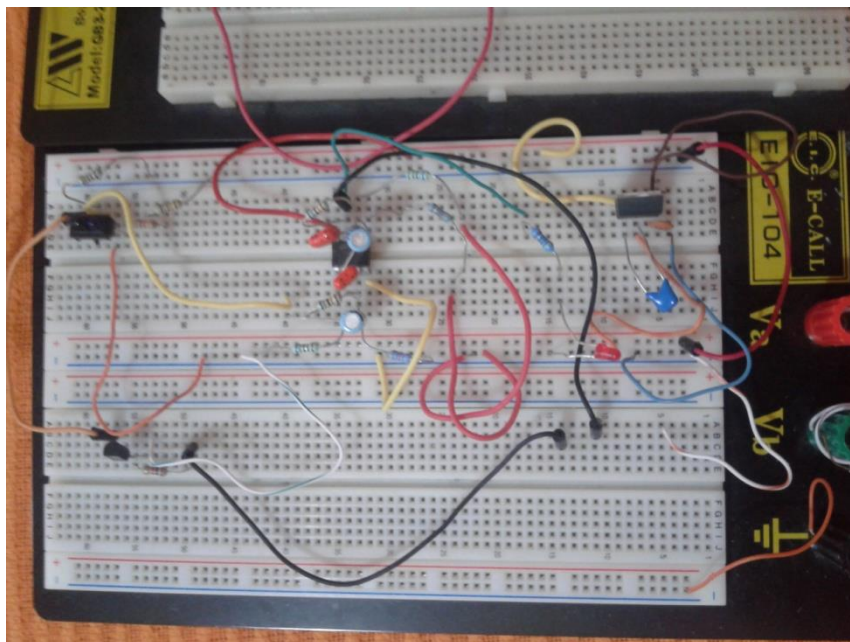
Σχήμα 3.3 – Κύκλωμα αισθητήρα.

Πιο αναλυτικά, όταν τοποθετήσουμε τον δείκτη του αριστερού χεριού πάνω στον αισθητήρα υπέρυθρο φως θα διαπεράσει το δέρμα μας και θα ανακλαστεί ανάλογα με την κυκλοφορία του αίματος. Το τρανζίστορ ως δέκτης είναι σε θέση να αναγνωρίζει αυτές τις μεταβολές και να δημιουργεί διάφορα δυναμικά ανάλογα με την ανακλώμενη ποσότητα υπέρυθρου φωτός. Έτσι κάθε φορά που η καρδιά θα χτυπάει θα αλλάζει και το δυναμικό με αποτέλεσμα να έχουμε την δημιουργία ενός εναλλασσόμενου σήματος – με την προϋπόθεση βέβαια ότι μεταξύ κάθε χτύπου το χαμηλό και το υψηλό δυναμικό θα είναι περίπου ίδια σε κάθε περίοδο. Ωστόσο, αυτό το σήμα περιέχει dc συνιστώσα καθώς και θόρυβο με συνέπεια να μην έχουμε τις κατάλληλες προϋποθέσεις για να οδηγήσουμε ένα ολοκληρωμένο για την εξαγωγή μετρήσεων. Έτσι, χρειαζόμαστε μια ενίσχυση και επεξεργασία του σήματος ώστε να φτάσουμε στην δημιουργία παλμού στα 5V περίπου όπου είναι μια ικανοποιητική στάθμη για

να οδηγηθούν οι περισσότερες οικογένειες ολοκληρωμένων TTL. Ταυτόχρονα με την όλη διαδικασία μέτρησης έχουμε ένα LED στο οποίο μπορούμε να δούμε τον παλμό – χτύπο της καρδιάς – ώστε να σιγουρευτούμε ότι το κύκλωμα παίρνει μετρήσεις.



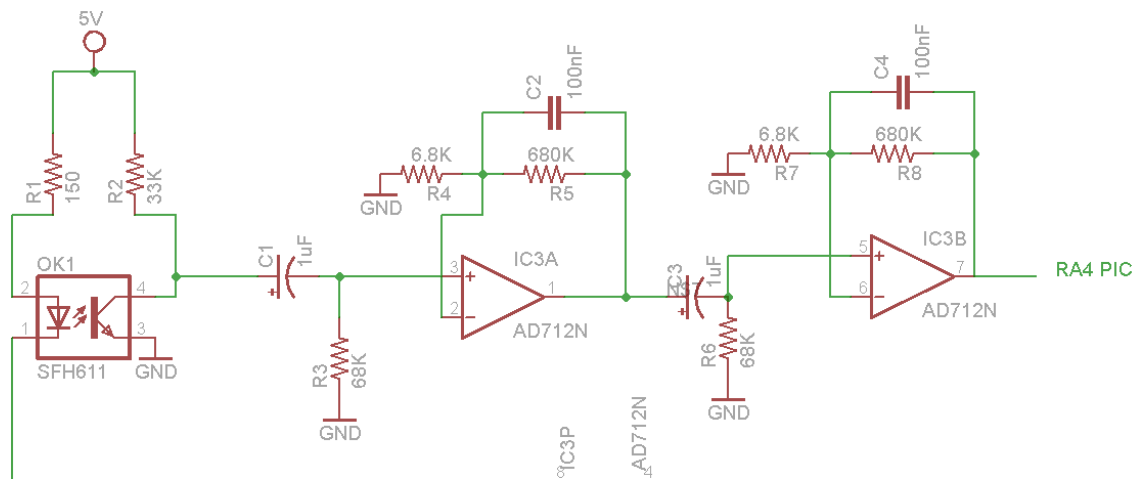
Σχήμα 3.4 – Έξοδος φίλτρου με LED για την απεικόνιση του παλμού.



Σχήμα 3.5 – Ο αισθητήρας και τα φίλτρα στο ράστερ.

Το Led οδηγείται από την έξοδο του φίλτρου, δηλαδή τον παλμό και προσομοιώνει κατά κάποιον τρόπο τον χτύπο της καρδιάς.

Η ανακλώμενη υπέρυθη ακτινοβολία του σήματος που ανιχνεύεται από το τρανζίστορ τροφοδοτείται σε ένα κύκλωμα ρύθμισης σημάτων που φιλτράρει τον θόρυβο και την dc συνιστώσα και ενισχύσει το εναλλασσόμενο σήμα, μετατρέποντας το σε TTL παλμό. Το κύκλωμα στο παρακάτω διάγραμμα δείχνει τον αισθητήρα μαζί με το κύκλωμα ρύθμισης του σήματος, που αποτελείται από δυο βαθμίδες με δύο φίλτρα η κάθε μία, ένα παθητικό υψηλερατό και ένα ενεργό βαθυπερατό με μεγάλη ενίσχυση. Οι cut-off συχνότητες των δύο βαθυπερατών φίλτρων είναι περίπου 2,5 Hz, και γι 'αυτό μπορεί να μετρήσει τον καρδιακό ρυθμό, έως $2,5 * 60 = 150$ bpm. Το κέρδος του κάθε φίλτρο είναι περίπου 100, άρα συνολικά έχουμε κέρδος $100 * 100 = 10000$. Αυτό είναι αρκετό για να μετατρέψει το αδύναμο εναλλασσόμενο σήμα σε έναν παλμό TTL.



Σχήμα 3.6 – Κύκλωμα αισθητήρα με φίλτρα για την δημιουργία παλμού.

3.3 Τα φίλτρα

3.3.1 Κατηγορίες φίλτρων

Όπως είναι γνωστό, φίλτρο είναι μια ηλεκτρονική διάταξη η οποία παρεμβάλλεται μεταξύ δυο βαθμίδων με σκοπό να ελέγξει την ισχύ που μεταφέρεται από την πρώτη βαθμίδα στην δεύτερη με έναν τρόπο που εξαρτάται από την συχνότητα. Με το φίλτρο δηλαδή μπορούμε να οδηγήσουμε μόνο τις επιθυμητές συχνότητες στην επόμενη βαθμίδα με κάποια μικρή εξασθένηση ενώ τις υπόλοιπες με τόσο μεγάλη εξασθένηση που να θεωρείται πως δεν περνούν. Τα φίλτρα τα χωρίζουμε σε πολλές κατηγορίες ανάλογα με τον τρόπο που είναι κατασκευασμένα και την λειτουργία τους. Κάποιες βασικές κατηγορίες είναι:

1. **Χαμηλοπερατά** (Low pass), επιτρέπουν να περάσουν οι χαμηλές συχνότητες και αποκόπτουν τις υψηλές.
2. **Υψηπερατά** (High pass), επιτρέπουν να περάσουν οι υψηλές συχνότητες και αποκόπτουν τις χαμηλές.
3. **Ζωνοδιαβατό** (Band pass), επιτρέπει την διέλευση συχνοτήτων μεταξύ ενός εύρους τιμών.
4. **Ζωνοφρακτικά** (Band stop), αποκόπτει τις συχνότητες μεταξύ ενός εύρους τιμών.

Τα φίλτρα χωρίζονται και σε δυο άλλες μεγάλες κατηγορίες, τα παθητικά και τα ενεργά. Ενεργά είναι τα φίλτρα, τα οποία χρειάζονται εξωτερική τροφοδοσία για να λειτουργήσουν. Εκτός από αντιστάσεις και πυκνωτές, συνήθως περιλαμβάνουν και τελεστικούς ενισχυτές, οι οποίοι απαιτούν εξωτερική τροφοδοσία και δίνουν στα φίλτρα επιπλέον χαρακτηριστικά που είναι πολύ χρήσιμα για πολλές ηλεκτρονικές εφαρμογές. Επιπλέον, παρέχουν μεγαλύτερη ευκολία και ευελιξία στην σχεδίαση και γι' αυτό προτιμώνται συχνά. Τα φίλτρα αυτά χρησιμοποιούνται για εφαρμογές χαμηλών συχνοτήτων.

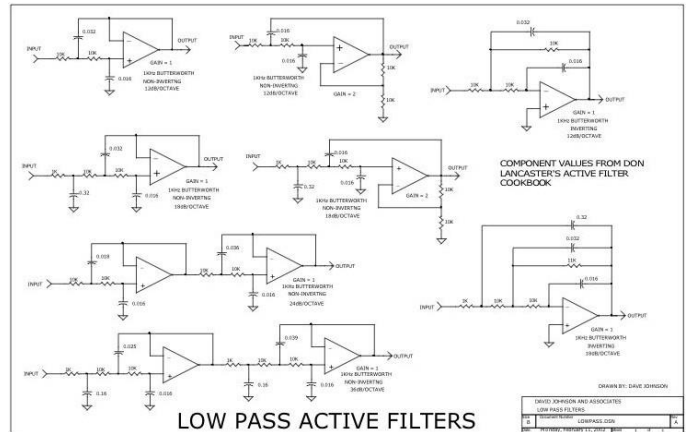
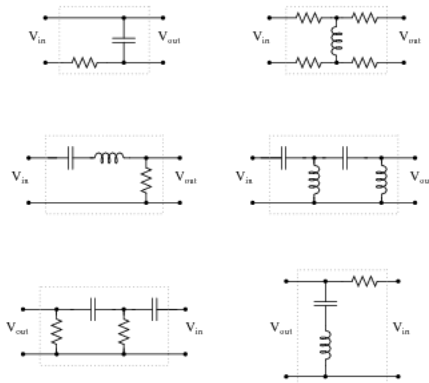
Πλεονεκτήματα ενεργών φίλτρων:

- Δεν συμπεριλαμβάνουν πηνία και ως εκ τούτου μπορούν να καταλαμβάνουν μικρή επιφάνεια και είναι γενικά φτηνότερα.
- Παρέχουν ενίσχυση με τη χρήση τελεστικών ενισχυτών.
- Δίνουν τη δυνατότητα διασύνδεσης πολλών ανεξαρτήτων βαθμίδων, οι οποίες σχεδιάζονται ανεξάρτητα (τελεστικοί ενισχυτές).
- Μπορούν να χρησιμοποιηθούν σε χαμηλές συχνότητες (κάτω από 500Hz) σε αντίθεση με τα παθητικά.

Μειονεκτήματα ενεργών φίλτρων:

- Είναι λιγότερο αξιόπιστα από τα παθητικά και παρουσιάζουν μεγαλύτερη ευαισθησία.
- Η χρήση τους είναι πρακτικά δύσκολη σε πολύ υψηλές συχνότητες (πάνω από 100 MHz) – τα παθητικά χρησιμοποιούνται σε μέχρι αρκετές εκατοντάδες MHz.
- Οι τελεστικοί ενισχυτές προσθέτουν θόρυβο.

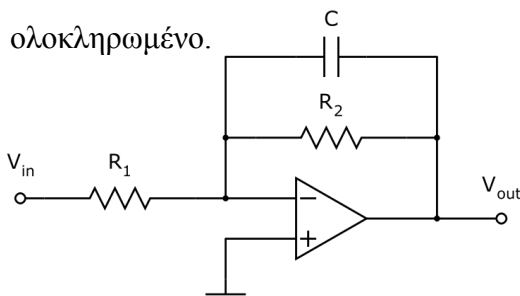
- Παθητικά είναι τα φίλτρα τα οποία στην δομή τους έχουν μόνο παθητικά στοιχεία όπως πυκνωτές και πηνία.



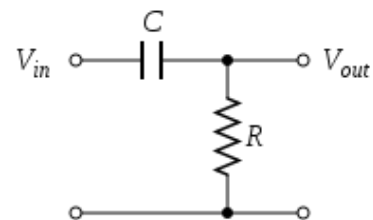
Σχήμα 3.7 – Ενεργά και παθητικά φίλτρα

Τα βασικά χαρακτηριστικά των φίλτρων ασχέτως σε ποια κατηγορία ανήκουν είναι η συχνότητα αποκοπής, η απώλεια ή εξασθένηση, η καθυστέρηση και η ομαδική καθυστέρηση. Η συχνότητα αποκοπής είναι η συχνότητα μισής ισχύος δηλαδή, είναι η τιμή της συχνότητας στην οποία θα ισχύει $P_{out} = \frac{P_{in}}{2}$. Η εξασθένηση έχει να κάνει με το πόσο η έξοδος έχει εξασθενήσει σε σχέση με την είσοδο και η καθυστέρηση έχει να κάνει με την φάση και με την αλλαγή αυτής όταν το σήμα φιλτραριστεί.

Για το δικό μας σήμα θα χρησιμοποιήσουμε ενεργά φίλτρα για να επιτευχθεί η ενίσχυση του ασθενούς σήματος από τον αισθητήρα. Η χρήση παθητικού φίλτρου είναι επίσης απαραίτητη για την αποκοπή της dc συνιστώσας. Συγκεκριμένα θα γίνει χρήση δυο παθητικών υπερπερατών φίλτρων και δυο ενεργών χαμηλοπερατών φίλτρων. Με αυτό τον τρόπο θα μπορέσουμε να πετύχουμε το επιθυμητό αποτέλεσμα που είναι η αποκοπή ανεπιθύμητης πληροφορίας και παράλληλα την ενίσχυση του σήματος για να οδηγεί το ολοκληρωμένο.



Σχήμα 3.8 – Ενεργό χαμηλοπερατό φίλτρο



Σχήμα 3.9 – Παθητικό υπερπερατό φίλτρο

3.3.2 Μαθηματική Ανάλυση Φίλτρων

Όπως είδαμε σε προηγούμενο κεφάλαιο ο μέσος όρος των παλμών ενός ενήλικα είναι περίπου 60 -100 bpm. Στην περίπτωση μας, για να έχουμε μεγάλο εύρος δυνατών μετρήσεων χρησιμοποιούμε συχνότητα αποκοπής στα βαθυπερατά φίλτρα γύρω στα 2,5 Hz. Έτσι για το χαμηλοπερατό φίλτρο θα ισχύει $f_c = \frac{1}{2*\pi*C*R2}$. Επίσης είναι σημαντικό να ρυθμίσουμε το κέρδος που θέλουμε να έχει το φίλτρο για την ενίσχυση του σήματος από τον αισθητήρα. Αυτό το πετυχαίνουμε σύμφωνα με την εξίσωση $G = 1 + \frac{R2}{R1}$. Τώρα έχουμε μια πλήρη εικόνα με τα βασικά κριτήρια που πρέπει να ισχύουν ώστε το φίλτρο μας να ενισχύει στο κατάλληλο βαθμό και να απορρίπτει τον θόρυβο. Από όλα τα παραπάνω καταλήγουμε:

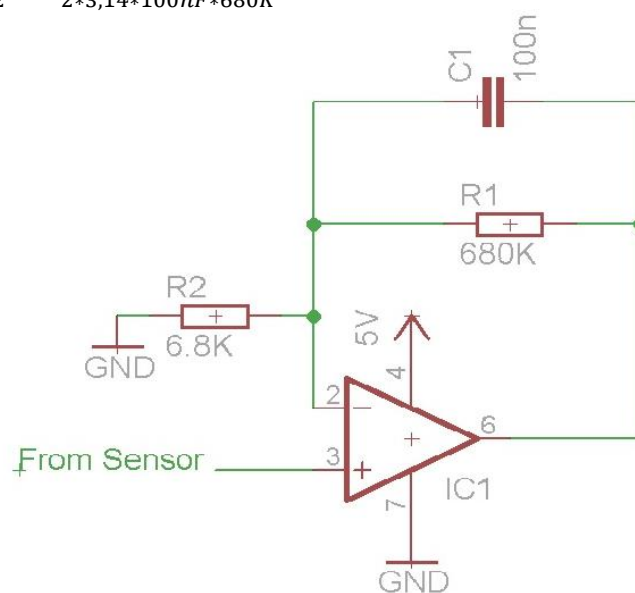
Εξίσωση 1 :

$$G \cong 100 \leftrightarrow 1 + \frac{R2}{R1} \cong 100 \leftrightarrow 1 + \frac{680K}{R1} \cong 100 \leftrightarrow R1 \cong \frac{680K}{99} \cong 6.8K$$

Για C = 100nF

Εξίσωση 2 :

$$f_c = \frac{1}{2*\pi*C*R2} = \frac{1}{2*3,14*100nF*680K} = 2.34 \text{ Hz}$$

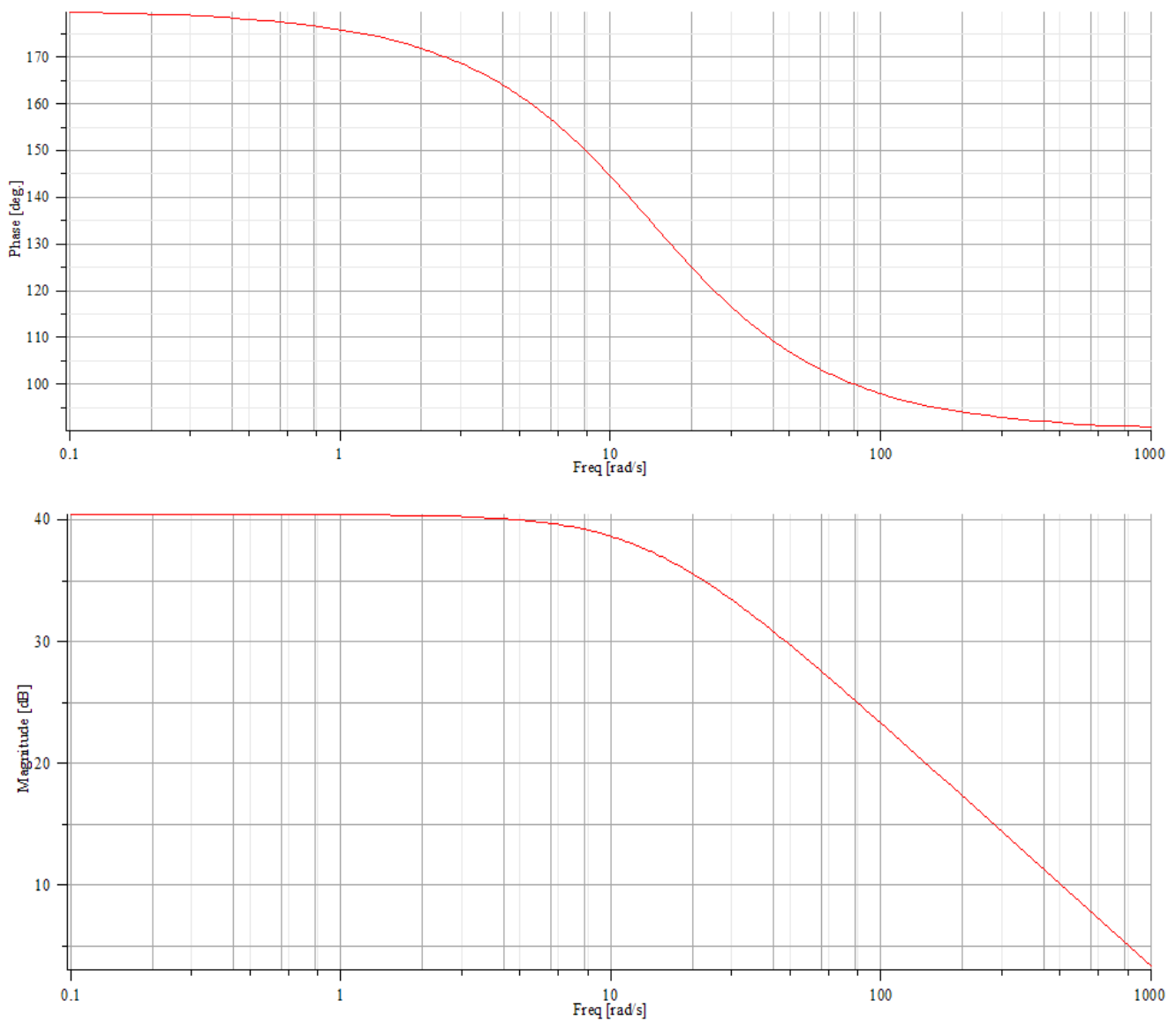


Σχήμα 3.10 – Ενεργό χαμηλοπερατό φίλτρο με τιμές

Η συνάρτηση μεταφοράς του φίλτρου είναι:

Εξίσωση 3 :

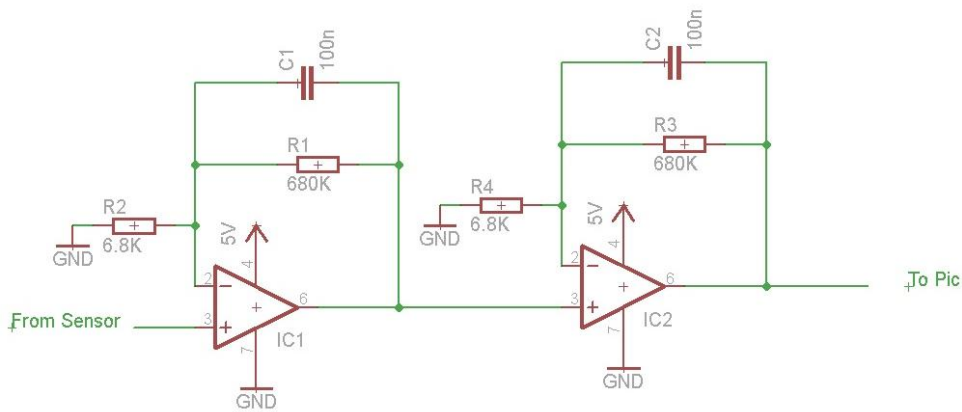
$$H(s) = -\frac{\frac{1}{R_2 \cdot C_1}}{s + \frac{1}{R_1 \cdot C_1}} = -\frac{\frac{1}{6.8K \cdot 100nF}}{s + \frac{1}{680K \cdot 100nF}} = -\frac{1470,58823529}{s + 14,70588235294118}$$



Σχήμα 3.11 – Διάγραμμα Bode

Από το Bode διάγραμμα του φίλτρου διακρίνεται πως μέχρι την συχνότητα αποκοπής έχουμε ενίσχυση 40dB δηλαδή 100 σε καθαρό αριθμό, όπως επίσης το ότι έχουμε αλλαγή φάσης περίπου 180° μέχρι την συχνότητα αποκοπής.

Άρα τα δυο βαθυπερατά φίλτρα που θα χρησιμοποιήσουμε είναι αυτά του σχήματος 4.6 :

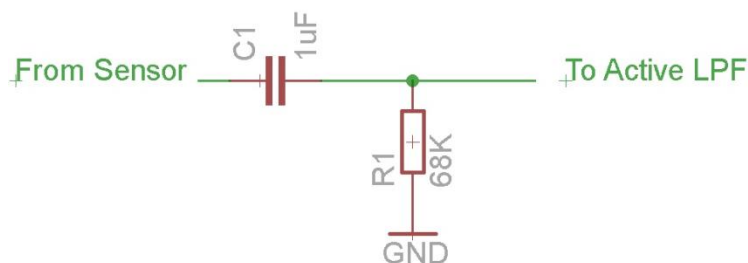


Σχήμα 3.12 – Βαθμίδες βαθυπερατών φίλτρων.

Αναφέραμε παραπάνω πώς είναι απαραίτητη η ύπαρξη και ενός παθητικού υψηπερατού φίλτρου σε κάθε βαθμίδα που θα εξαλείφει την DC συνιστώσα που περιέχει το σήμα μας λόγω των ιστών. Για τα παθητικά υψηπερατά φίλτρα ισχύει:

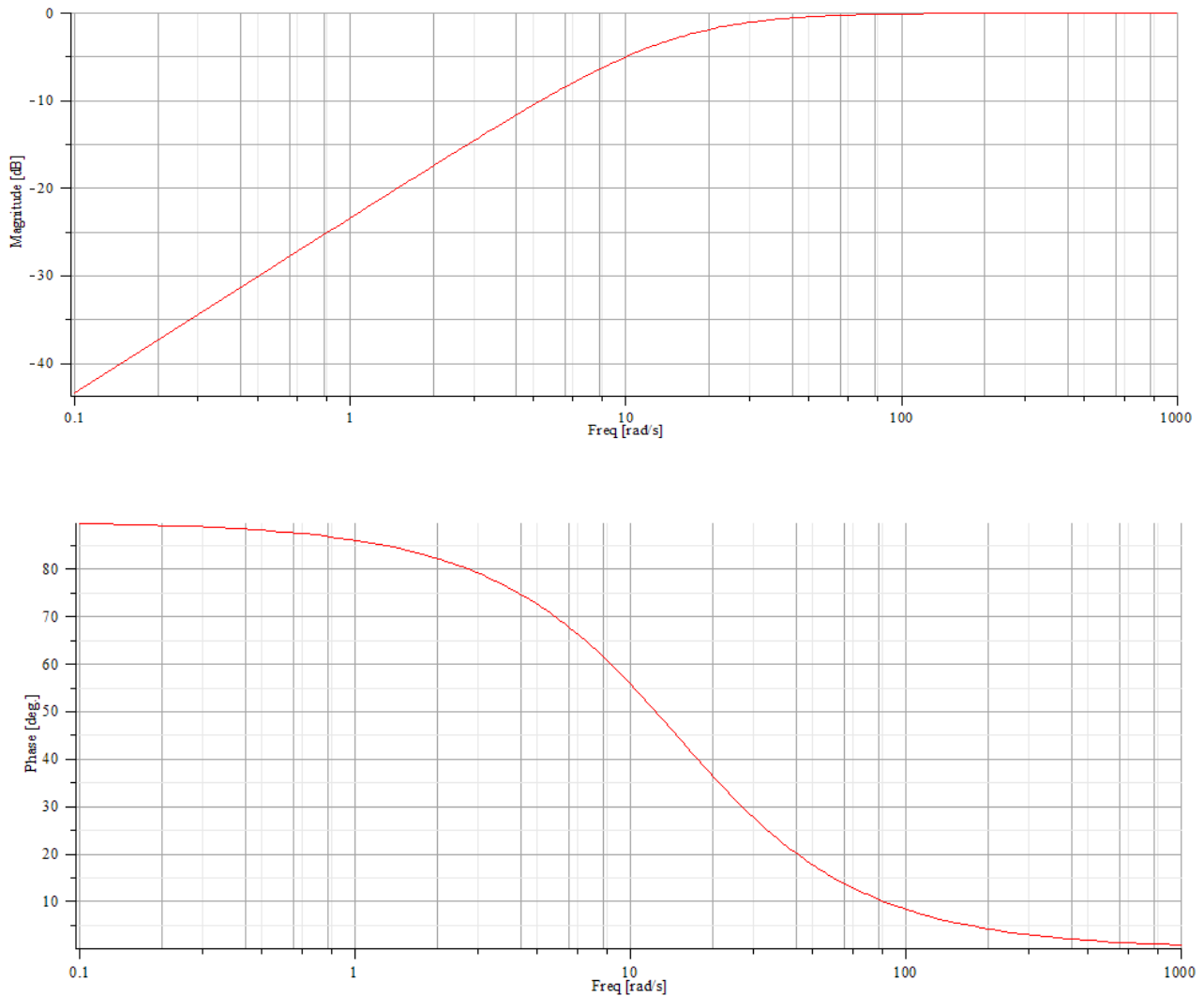
$$\text{Συνάρτηση μεταφοράς: } H(s) = \frac{s \cdot R \cdot C}{1 + s \cdot R \cdot C} = \frac{s \cdot 68K \cdot 1\mu F}{1 + s \cdot 68K \cdot 1\mu F} = \frac{s}{s + 14.7058823529}$$

$$\text{Συχνότητα αποκοπής: } f_c = \frac{1}{2 \cdot \pi \cdot C \cdot R2} = \frac{1}{2 \cdot 3,14 \cdot 1\mu F \cdot 68K} = 2.34 \text{ Hz}$$



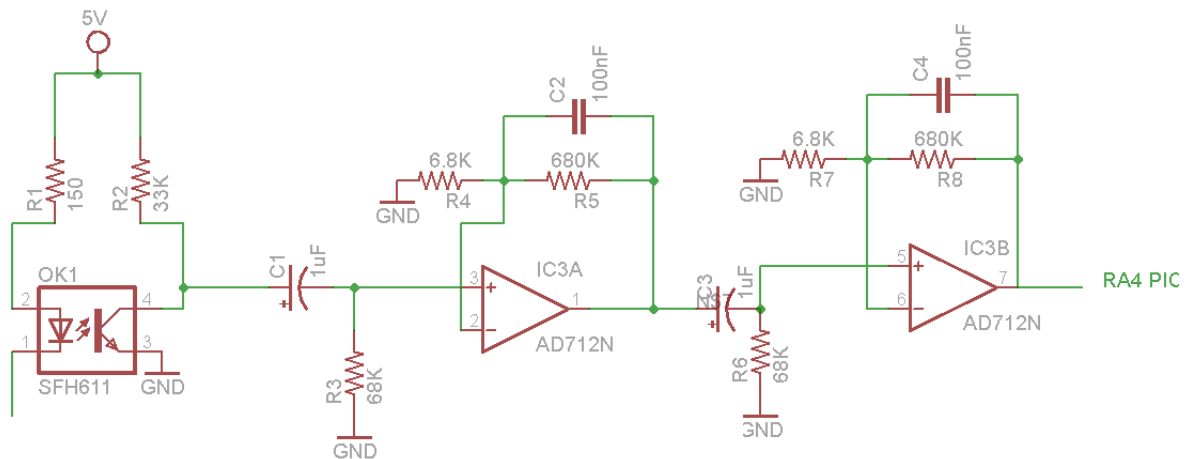
Σχήμα 3.13 – Παθητικό Υψηπερατό φίλτρο

Το διάγραμμα Bode μας δείχνει ξεκάθαρα πως οποιαδήποτε συχνότητα θα περάσει μικρότερη της f_c και συγκεκριμένα οποιαδήποτε DC όπου $f_{DC} = 0$ θα αποκοπεί. Επίσης διακριτή είναι και η διαφορά φάσης του σήματος που φιλτράρετε και είναι 90° μέχρι την f_c .



Σχήμα 3.14 – Διαγράμματα Bode

Τώρα το συνολικό κύκλωμα μαζί με τον αισθητήρα και τα φίλτρα θα είναι το παρακάτω.



Σχήμα 3.15 – Κύκλωμα Αισθητήρα με τις βαθμίδες των φίλτρων.

3.4 Οι επιμέρους λειτουργίες

3.4.1 Μέτρηση

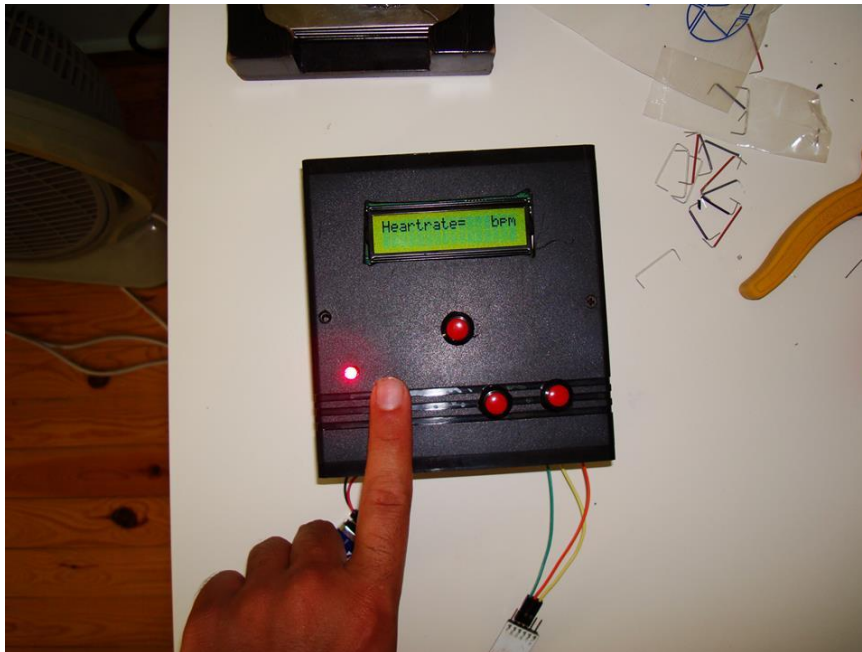
Ο μικροελεγκτής περιμένει από τον χρήστη να πατήσει το pushbutton start για να ξεκινήσει. Όταν πατηθεί, ο μικροελεγκτής γειώνει την βάση του τρανζίστορ, το οποίο λειτουργεί σαν διακόπτης, και θέτει τον αισθητήρα σε λειτουργία. Την ίδια στιγμή ο μετρητής μηδενίζεται και ο μικροελεγκτής ξεκινά να μετράει τον τετραγωνικό παλμό που βλέπει την ακίδα T0CKI, είσοδος για τον timer0. Για κάθε μετάβαση από υψηλό δυναμικό σε χαμηλό δυναμικό, η τιμή του μετρητή timer0 αυξάνεται κατά μία μονάδα. Η διαδικασία αυτή διαρκεί για 30 δευτερόλεπτα έτσι ώστε ο μικροελεγκτής να μπορέσει να συγκεντρώσει αρκετά δείγματα. Ιδανικά, ο μικροελεγκτής θα έπρεπε να μετράει για ένα ολόκληρο λεπτό αλλά κάτι τέτοιο αυξάνει την κατανάλωση ρεύματος της συσκευής και μειώνει δραματικά τον χρόνο ζωής της μπαταρίας με την οποία λειτουργεί. Ωστόσο μετρώντας για 30 δευτερόλεπτα έχουμε απόκλιση στην τελική τιμή εξαιτίας του μικρότερου πλήθους δειγμάτων. Αν πολλαπλασιάσουμε την τελική τιμή του μετρητή με το 2 (καθώς ο καρδιακός ρυθμός μετριέται σε χτύπους ανά λεπτό) δεν θα έχουμε το ίδιο αποτέλεσμα με μια μέτρηση για ένα ολόκληρο λεπτό. Συγκρίνοντας τα αποτελέσματα της συσκευής για μέτρηση δεκαπέντε, τριάντα και εξήντα δευτερολέπτων με τα αποτελέσματα ενός ηλεκτρονικού πιεσόμετρου και

μιας εφαρμογής μέτρησης καρδιακού ρυθμού για κινητά τηλέφωνα με λειτουργικό σύστημα android, καταλήξαμε πως η μικρότερη απόκλιση παρουσιάζεται αν μετρήσουμε το σήμα για τριάντα δευτερόλεπτα και την συνέχεια πολλαπλασιάσουμε με το 2,1 κρατώντας ωστόσο μόνο το ακέραιο μέρος του γινομένου χωρίς στρογγυλοποίηση. Αυτή η τεχνική μειώνει την απόκλιση χωρίς να παραμορφώνει την μέτρηση και ταυτόχρονα εξοικονομεί χρόνο ζωής για την μπαταρία. Οπότε αν ονομάσουμε pulsecount την τιμή του timer0 μετά από 30 δευτερόλεπτα, για τον καρδιακό ρυθμό θα ισχύει :

$$\text{pulse rate} = \text{int pulsecount} * 2,1$$

όπου int το ακέραιο μέρος της πράξης που ακολουθεί.

Στο κεφάλαιο 5 ακολουθούν ενδεικτικοί πίνακες με την απόκλιση του καρδιακού ρυθμού ανάλογα με την διάρκεια της μέτρησης. Οι μετρήσεις σε κάθε σειρά του πίνακα έγιναν με χρονική διαφορά ενός λεπτού.

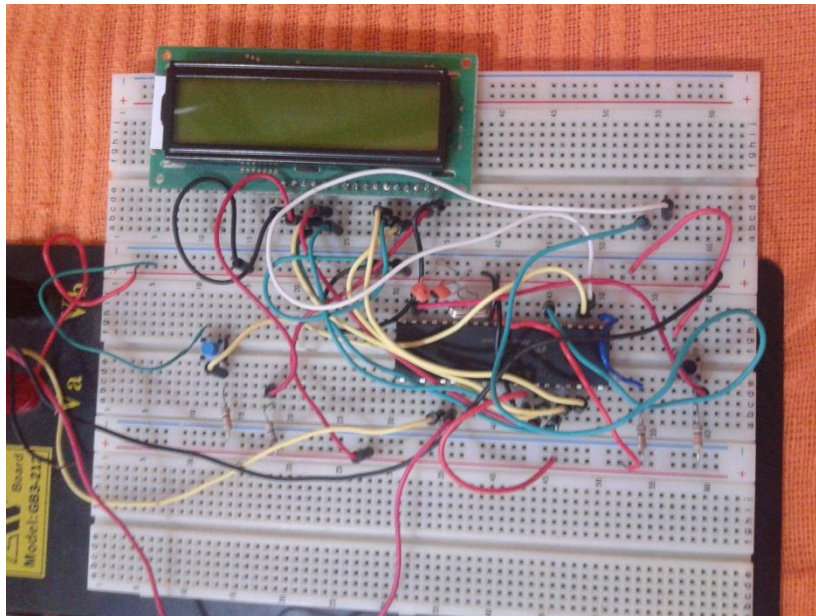


Σχήμα 3.16 – Μέτρηση καρδιακού ρυθμού. Το led αναβοσβήνει σε συγχρονισμό.

3.4.2 Απεικόνιση

Οι οθόνες LCD 16x2

Για την απεικόνιση των μετρήσεων και των μηνυμάτων, ο μικροελεγκτής επικοινωνεί με μια οθόνη LCD 16x2. Αυτές οι οθόνες υπόκεινται σε μια ενιαία τυποποίηση, που αναφέρεται με την ονομασία HD44780. Σύμφωνα με την τυποποίηση αυτήν, υπάρχουν τρεις ακίδες ελέγχου με τις ονομασίες RS,EN,RW, τρεις ακίδες τροφοδοσίας (Vdd,Vss,Vee) και οκτώ ακίδες δεδομένων με τις ονομασίες D0,D1,D2,D3,D4,D5,D6,D7. με την ακίδα RS (Register Select) στην γείωση, ειδοποιείται η οθόνη να χειρισθεί τα δεδομένα σαν διαταγή ή σαν ειδική εντολή ,όπως για παράδειγμα ο καθαρισμός της οθόνης, ενώ αν είναι στην τροφοδοσία τα δεδομένα που αποστέλλονται είναι κείμενο προς εμφάνιση. Όταν η ακίδα RW(Read/Write) είναι στην γείωση γράφονται τα δεδομένα των D0.....D7 στην οθόνη ενώ αν η ακίδα είναι στην τροφοδοσία διαβάζεται η LCD οθόνη. Η ακίδα EN(Enable) ειδοποιεί την οθόνη όποτε της στέλνουν δεδομένα.



Σχήμα 3.17 – Ο μικροελεγκτής και η οθόνη πάνω στο ράστερ.

Πίνακας 3.2 – Οι ακροδέκτες της οθόνης LCD 16x2

PIN CONNECTIONS			
PIN	Symbol	Level	Function
1	VSS	—	GND(0V)
2	VDD	—	Supply Voltage for Logic(+5V)
3	V0	—	Power supply for LCD
4	RS	H/L	H: Data; L: Instruction Code
5	R/W	H/L	H: Read; L: Write
6	E	H/L	Enable Signal
7	DB0	H/L	Data Bus Line
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	BL1	—	Backlight Power(+5V)
16	BL2	—	Backlight Power(0V)

Επικοινωνία με τον μικροελεγκτή

Η μεταφορά των δεδομένων προς την οθόνη γίνεται σε 4-bit mode, δηλαδή τα δεδομένα στέλνονται σε nibbles. Πρώτα στέλνεται το high nibble και στην συνέχεια το low nibble. Το θετικό αυτής της μεθόδου είναι ότι χρειάζονται λιγότερες συνδέσεις ανάμεσα στον μικροελεγκτή και την οθόνη αφού χρησιμοποιούμε μόνο τέσσερις από τις οκτώ ακίδες δεδομένων της LCD οθόνης.

Ο προγραμματισμός της οθόνης μέσω του μικροελεγκτή γίνεται πιο εύκολος καθώς χρησιμοποιούμε τον MikroC for PIC Pro compiler. Ο compiler αυτός διαθέτει βιβλιοθήκες για την επικοινωνία με την οθόνη με αποτέλεσμα να μπορούμε να εμφανίσουμε ένα μήνυμα στην οθόνη με μόνο μία γραμμή κώδικα. Αντίστοιχα υπάρχουν εντολές για την αρχικοποίηση της οθόνης, τον καθαρισμό της κλπ. Είναι απαραίτητο να δηλώσουμε στον κώδικα του μικροελεγκτή ποιες ακίδες θα χρησιμοποιήσουμε στην διασύνδεση με την οθόνη για να μπορούμε να καλούμε τις έτοιμες εντολές του compiler όποτε χρειάζεται. Έτσι στον κώδικα μας έχουμε δηλώσει : sbit LCD_RS at RB4_bit, sbit LCD_EN at RB5_bit, sbit LCD_D4 at RB0_bit, sbit LCD_D5 at RB1_bit, sbit LCD_D6 at RB2_bit, sbit LCD_D7 at RB3_bit, sbit LCD_RS_Direction at TRISB4_bit,sbit LCD_EN_Direction at TRISB5_bit,sbit LCD_D4_Direction at TRISB0_bit,sbit LCD_D5_Direction at TRISB1_bit,sbit

LCD_D6_Direction at TRISB2_bit,sbit LCD_D7_Direction at TRISB3_bit. Αυτές οι αρχικοποιήσεις μας επιτρέπουν να εμφανίζουμε μηνύματα στην οθόνη με την εντολή Lcd_Out().

Η οθόνη εμφανίζει συνολικά δέκα μηνύματα. Με το άνοιγμα της συσκευής εμφανίζεται το μήνυμα “Press button” που προτρέπει στον χρήστη να πατήσει έναν από τους τρεις διαθέσιμους διακόπτες, δύο βρίσκονται πάνω στην πλακέτα και ο τρίτος αποτελεί κομμάτι του software για τον υπολογιστή. Ανάλογα με το ποια λειτουργία επιλέξουμε, εμφανίζονται τα αντίστοιχα μηνύματα. Αν επιλέξουμε να μετρήσουμε τον καρδιακό ρυθμό, κατά την διάρκεια της μέτρησης υπάρχει η ένδειξη “Heartrate= bpm”. Όταν ολοκληρωθεί η μέτρηση το μήνυμα συμπληρώνεται με την τιμή του καρδιακού ρυθμού, για παράδειγμα η οθόνη πλέον θα δείχνει “Heartrate= 082bpm”. Στην συνέχεια εμφανίζονται τα μηνύματα αποθήκευσης. Αν επιλέξουμε να αποκτήσουμε πρόσβαση στην μνήμη, εμφανίζεται το πλήθος των μετρήσεων που απομένουν πριν γεμίσει η μνήμη EEPROM του μικροελεγκτή, καθώς και ερώτηση προς τον χρήστη για το αν επιθυμεί να σβήσει όλες τις αποθηκευμένες μετρήσεις. Αν επιλέξουμε την σύνδεση με υπολογιστή για την μεταφορά των δεδομένων, εμφανίζεται μήνυμα για την κατάσταση αποστολής. Ακολουθεί συγκεντρωτικός πίνακας με τα διαθέσιμα μηνύματα.

Πίνακας 3.3 – Μηνύματα συσκευής.

Λειτουργία	Μήνυμα
Αναμονή για επιλογή λειτουργίας	Press button
Μέτρηση καρδιακού ρυθμού	Heartrate=XXX bpm
Αποθήκευση μέτρησης	Saving Data
Ολοκλήρωση αποθήκευσης	Data Saved
Πλήθος μετρήσεων που απομένουν	XXX remaining
Ερώτηση για διαγραφή μνήμης	Delete memory?
Διαγραφή μνήμης	Memory deleted
Αποστολή δεδομένων στον υπολογιστή	Sending
Ολοκλήρωση αποστολής	Data was sent
Τέλος επικοινωνίας με τον υπολογιστή	Done.



Σχήμα 3.18 – Αρχικό μήνυμα – αναμονή για επιλογή λειτουργίας



Σχήμα 3.19 – Μέτρηση καρδιακού ρυθμού.

Η μνήμη EEPROM

Σχεδόν όλοι οι μικροεπεξεργαστές PIC διαθέτουν ενσωματωμένη μνήμη EEPROM. Η μνήμη του PIC 16F887 έχει μέγεθος 256 bytes, γεγονός που μας επιτρέπει να αποθηκεύσουμε 256 μετρήσεις καρδιακού ρυθμού. Μία θέση μνήμης στην οποία δεν υπάρχει τίποτα αποθηκευμένο, αν διαβαστεί μας δίνει την τιμή FFh και όχι 00h. Η μνήμη EEPROM μπορεί να εγγραφεί και να διαβαστεί χρησιμοποιώντας κάποιους συγκεκριμένους καταχωρητές. Υπάρχουν τέσσερις καταχωρητές που ελέγχουν τις λειτουργίες στην μνήμη του μικροελεγκτή. Είναι οι EECON1, EECON2, EEDATA, EEADR.

Ο EEADR κρατά την διεύθυνση της θέσης μνήμης που μας ενδιαφέρει για ανάγνωση ή εγγραφή ενώ ο EEDATA περιέχει τα οκτανήφια δεδομένα που βρίσκονται στην παραπάνω θέση μνήμης προς ανάγνωση ή τα δεδομένα που πρόκειται να εγγραφθούν σε αυτήν. Οι EECON1 και EECON2 χρησιμοποιούνται για την ανάγνωση και εγγραφή δεδομένων στην μνήμη. Δεν χρειάζεται να εμβαθύνουμε στην λειτουργία των τεσσάρων αυτών καταχωρητών καθώς ο compiler MikroC for PIC PRO διαθέτει ειδικές βιβλιοθήκες για την μνήμη EEPROM, οι οποίες αναλαμβάνουν να διαμορφώσουν κατάλληλα τους παραπάνω καταχωρητές ανάλογα με την επιθυμητή λειτουργία.

Οι δύο εντολές που χρησιμοποιούμε για να επέμβουμε στην μνήμη είναι οι :

EEPROM_Read(address) : επιστρέφει το byte που βρίσκεται αποθηκευμένο στην θέση μνήμης “address”, η οποία μπορεί να παίρνει τιμές από 0 έως και 255.

EEPROM_Write(address, data) : αποθηκεύει την οκτανήφια τιμή “data” στην διεύθυνση “address”.

Αποθήκευση μετρήσεων

Αφού μετρήσουμε τον καρδιακό ρυθμό και εμφανιστεί στην οθόνη ακολουθεί η αποθήκευση της μέτρησης. Η αποθήκευση γίνεται διαδοχικά, την πρώτη θέση μνήμης καταλαμβάνει η παλαιότερη μέτρηση ενώ την τελευταία η πιο πρόσφατη. Για να μην υπάρχουν κενές θέσεις ανάμεσα στις μετρήσεις, γίνεται έλεγχος από τον μικροελεγκτή για το αν υπάρχει αποθηκευμένη μέτρηση σε κάθε θέση μνήμης. Όταν το πρόγραμμα συναντήσει

την πρώτη κενή θέση σταματά τον έλεγχο και προχωρά στην αποθήκευση της μέτρησης. Ο έλεγχος αυτός γίνεται μέσα σε μια απλή for :

```
for (i=0;i<255;i++)  
  
{ temp=EEPROM_Read(i);  
  
  if (temp == 0xFF)  
  
    {  
  
      EEPROM_Write(i,pulserate);  
  
      i=256;  
  
    }  
  
}
```

Διαβάζεται το περιεχόμενο κάθε θέσης μνήμης και συγκρίνεται με το FFh. Αν είναι ίσα τότε η θέση μνήμης είναι κενή και σε αυτήν αποθηκεύεται ο καρδιακός ρυθμός pulserate. Για να σταματήσει ο έλεγχος πρέπει να βγούμε από την for για αυτό το i παίρνει την τιμή 256. Στην οθόνη εμφανίζεται μήνυμα κατά την αποθήκευση και νέο μήνυμα μόλις αυτή ολοκληρωθεί. Με το τέλος της αποθήκευσης γυρνάμε στην αρχική κατάσταση της συσκευής, ζητώντας από τον χρήστη να επιλέξει μια από τις διαθέσιμες λειτουργίες.



Σχήμα 3.20 – Αποθήκευση μέτρησης.

Διακόπτης memory

Μπορούμε να αποκτήσουμε πρόσβαση στην μνήμη EEPROM του μικροελεγκτή με την βοήθεια του διακόπτη memory. Αυτός ο διακόπτης παρέχει δύο λειτουργίες : έλεγχο του πλήθους των μετρήσεων που απομένουν πριν γεμίσει η μνήμη και ολική διαγραφή των αποθηκευμένων μετρήσεων. Αν τον πατήσουμε μία φορά ελέγχει τις κενές θέσεις μνήμης και ρωτά τον χρήστη αν θέλει να γίνει διαγραφή όλων των αποθηκευμένων μετρήσεων. Αν θέλουμε να την διαγράψουμε κρατάμε τον διακόπτη πατημένο μέχρι να εμφανιστεί μήνυμα στην οθόνη ότι η διαγραφή ήταν επιτυχής.

Η διάκριση ανάμεσα στις δύο καταστάσεις του διακόπτη (πατημένος μία φορά και πατημένος συνεχώς) γίνεται ελέγχοντας πόσες φορές ο διακόπτης άλλαξε κατάσταση. Κάθε φορά που πατάμε τον διακόπτη αυξάνεται ένας μετρητής. Ωστόσο, για να μην ερμηνεύει με λάθος τρόπο ο μικροελεγκτής το πάτημα υπάρχει έλεγχος switch debouncing.

Switch debouncing

Κατά το πάτημα ενός διακόπτη προκαλούνται αναπηδήσεις της τάσης, ο διακόπτης ταλαντεύεται για κάποια milliseconds ανάμεσα στις δύο καταστάσεις. Προκειμένου να μην ερμηνευτεί αυτό από τον μικροελεγκτή ως διαδοχικά πατήματα του διακόπτη, χρειαζόμαστε switch debouncing. Για να αντιμετωπίσουμε αυτό το πρόβλημα, ελέγχουμε αν ο διακόπτης άλλαξε κατάσταση χρησιμοποιώντας την βιβλιοθήκη Button του MikroC for PIC. Η ρουτίνα που χρησιμοποιούμε είναι η εξής :

Button(port, pin, time, active_state)

η οποία γνωρίζοντας την ενεργή κατάσταση του διακόπτη όταν αυτός πατηθεί (0 ή 1) ανιχνεύει την αλλαγή στον ακροδέκτη της πόρτας που βρίσκεται ο διακόπτης και εισάγει delay με διάρκεια που ορίζουμε εμείς από την πεδίο time σε milliseconds. Η ρουτίνα Button επιστρέφει λογικό 1 όταν ανιχνεύσει πάτημα του διακόπτη και 0 όταν ο διακόπτης δεν έχει πατηθεί, γεγονός που μας επιτρέπει να ελέγχουμε όχι μόνο πότε πατήθηκε ο διακόπτης αλλά και πόσες φορές έχει πατηθεί.

Button "Bounce"



Σχήμα 3.21 – Η ταλάντωση του διακόπτη ανάμεσα στις δύο καταστάσεις.

Εφόσον ο διακόπτης έχει αλλάξει κατάσταση, ο μετρητής αυξάνεται κατά ένα. Για να επιλέξουμε την επιθυμητή λειτουργία δεν ελέγχουμε την κατάσταση του διακόπτη αλλά την τιμή του μετρητή.

Πρόσβαση στις αποθηκευμένες μετρήσεις

Πατώντας τον διακόπτη memory μία φορά μπορούμε να ενημερωθούμε για το πόσες ακόμα μετρήσεις μπορούν να αποθηκευτούν στην μνήμη EEPROM πριν αυτή γεμίσει. Ο έλεγχος γίνεται με μια απλή for όπως και στην αποθήκευση των μετρήσεων. Ελέγχονται οι θέσεις μνήμης η μία μετά την άλλη μέχρι να βρεθεί η πρώτη κενή θέση μνήμης οπότε και σταματάει ο έλεγχος. Η θέση αυτή αποθηκεύεται σε μια μεταβλητή, η οποία στην συνέχεια αφαιρείται από την τιμή 255 για να προκύψει το πλήθος των κενών θέσεων.

```
for (j=0;j<255;j++)
```

```
{ temp=EEPROM_Read(j);
```

```
  if (temp == 0xFF)
```

```
    { flag=j;
```

```
      j=256;
```

```
    }
```

```
}
```

empty=255-flag;

Ακολουθεί η εμφάνιση μηνύματος που ενημερώνει τον χρήστη για το μέγεθος της διαθέσιμης μνήμης και αμέσως μετά η εμφάνιση ερώτησης προς τον χρήστη για την ολική διαγραφή των αποθηκευμένων μετρήσεων. Αν δεν θέλουμε να διαγράψουμε την μνήμη πατάμε τον διακόπτη reset ενώ αν θέλουμε να την διαγράψουμε κρατάμε πατημένο τον διακόπτη memory.



Σχήμα 3.22 – Πλήθος μετρήσεων που απομένουν πριν γεμίσει η μνήμη.

Διαγραφή αποθηκευμένων μετρήσεων

Αν επιλέξουμε την διαγραφή της μνήμης μέσω του διακόπτη memory, ο μικροελεγκτής ελέγχει τις θέσεις μνήμης την μία μετά την άλλη όπως και παραπάνω. Διαβάζει το περιεχόμενο κάθε θέσης και το συγκρίνει με το FFh, τιμή η οποία αντιστοιχεί σε μια κενή θέση μνήμης. Αν λοιπόν το περιεχόμενο είναι διάφορο του FFh, τότε στην θέση αυτή γράφει την τιμή FFh ώστε αυτή να θεωρείται πλέον άδεια. Για να γίνει σωστά η αντικατάσταση των τιμών θα πρέπει να υπάρχει μια μικρή καθυστέρηση ανάμεσα σε δύο διαδοχικές εγγραφές στην μνήμη. Αντίθετα, αν δεν υπάρχει αυτή η μικρή καθυστέρηση, ενώ έχει προσπελαστεί μια μη κενή θέση μνήμης, η διαγραφή του περιεχομένου της δεν θα είναι επιτυχής. Κάνουμε χρήση της for και σε αυτήν την περίπτωση αφού εξετάζουμε τις θέσεις μνήμης μία προς μία :

```

for(k=0;k<255;k++)

{ temp=EEPROM_Read(k);

    if (temp != 0xFF)

        { EEPROM_Write(k,0xFF);

            delay_ms(100);

        }

}

```

Στην συνέχεια εμφανίζεται μήνυμα στην οθόνη που ενημερώνει τον χρήστη ότι η διαγραφή της μνήμης ήταν επιτυχής και αμέσως μετά μεταφερόμαστε στην αρχική κατάσταση της συσκευής για την επιλογή λειτουργίας εκ νέου.

3.4.4 Σειριακή μεταφορά δεδομένων από την συσκευή στον υπολογιστή

Η μονάδα σειριακής επικοινωνίας USART

Ο μικροελεγκτής PIC16F887 διαθέτει ενσωματωμένη μονάδα σειριακής επικοινωνίας η οποία περιέχει τους απαραίτητους καταχωρητές για να εκτελέσει μια σειριακή μεταφορά δεδομένων. Το USART μπορεί να προγραμματιστεί με δύο τρόπους, σε ασύγχρονο τρόπο λειτουργίας full duplex για την επικοινωνία με περιφερειακές συσκευές όπως τα τερματικά CRT και οι προσωπικοί υπολογιστές και σε σύγχρονο τρόπο λειτουργίας half duplex για την επικοινωνία με μετατροπείς A/D ή D/A, σειριακές EEPROM ή άλλες συσκευές ως master ή slave.

Στον ασύγχρονο τρόπο λειτουργίας χρησιμοποιείται η μέθοδος μη επιστροφής στο μηδέν NRZ (Non Return to Zero) με ένα ψηφίο εκκίνησης (start bit) και ένα ψηφίο τέλους (stop bit) για τον συγχρονισμό του αποστολέα και του δέκτη. Τα δεδομένα προς αποστολή ή λήψη είναι οκταψήφια ή εννιαψήφια και αποστέλλονται σε frames που αποτελούνται από το ψηφίο εκκίνησης, τα ψηφία δεδομένων και το ψηφίο τέλους. Ο αποστολέας και ο δέκτης πρέπει αν έχουν ρυθμιστεί για να χρησιμοποιούν τον ίδιο ρυθμό δεδομένων, πλήθος ψηφίων δεδομένων

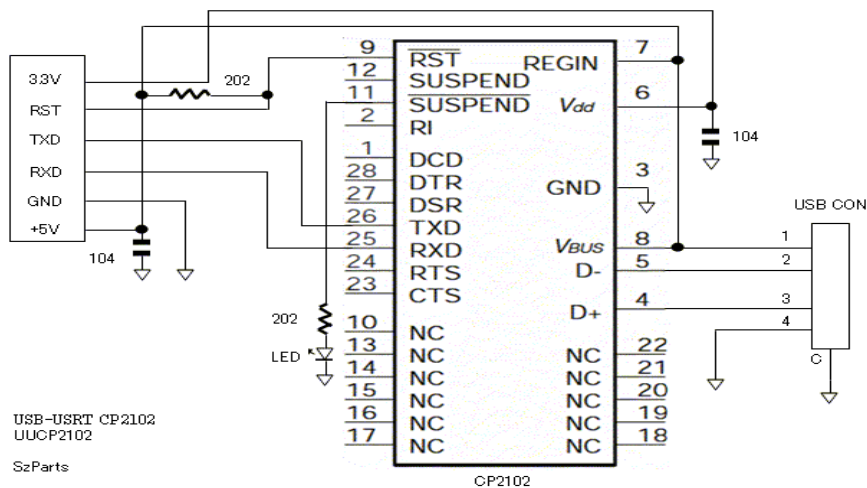
και πλήθος ψηφίων τέλους. Για την παραγωγή baud rate από τον ταλαντωτή του τσιπ χρησιμοποιείται η αφιερωμένη σε αυτόν τον σκοπό οκταψήφια γεννήτρια baud rate.

Κατά την αποστολή και λήψη μεταφέρεται πρώτα το λιγότερο σημαντικό ψηφίο LSB. Σε κατάσταση αδράνειας ο πομπός του μικροελεγκτή βρίσκεται σε λογικό 1. Όταν υπάρχουν δεδομένα προς αποστολή ο πομπός στέλνει το ψηφίο εκκίνησης στον δέκτη το οποίο είναι ένα λογικό 0 με διάρκεια μια περίοδο του clock. Ο δέκτης λαμβάνει τα ψηφία δεδομένων που ακολουθούν μέχρι ο πομπός να βρεθεί σε λογικό μηδέν για μια περίοδο του clock, σήμα που αποτελεί το ψηφίο τέλους. Αυτός ο τρόπος επικοινωνίας ονομάζεται ασύγχρονος γιατί ο δέκτης πρέπει να συγχρονίζει τον εαυτό του με βάση τον πομπό όποτε υπάρχει αποστολή δεδομένων, δηλαδή όποτε ο δέκτης βλέπει το ψηφίο εκκίνησης.

Αντίθετα, στην σύγχρονη επικοινωνία δεν υπάρχουν ψηφία εκκίνησης και τέλους. Ο αποστολέας και ο δέκτης συγχρονίζονται μεταξύ τους με την βοήθεια ενός clock που παρέχει η master συσκευή. Στην σύγχρονη επικοινωνία δεν μπορούμε να έχουμε ταυτόχρονη αποστολή και λήψη δεδομένων.

Σύνδεση με τον υπολογιστή

Οι σύγχρονοι προσωπικοί υπολογιστές δεν διαθέτουν πλέον σειριακές θύρες, οπότε είναι απαραίτητος ένας μετατροπέας USART-USB ο οποίος θα δημιουργήσει μία εικονική σειριακή θύρα στον υπολογιστή. Τέτοιοι μετατροπείς είναι ιδιαίτερα φθηνοί και μπορούν να συνδεθούν απευθείας στην TTL είσοδο ή έξοδο του μικροελεγκτή. Ο μετατροπέας που χρησιμοποιούμε στην κατασκευή χρησιμοποιεί το ολοκληρωμένο CP2102 το οποίο αναλαμβάνει την επικοινωνία ανάμεσα στον υπολογιστή και την συσκευή. Ο μετατροπέας διαθέτει επίσης εξόδους για τον μικροελεγκτή και τον υπολογιστή. Για την δημιουργία της εικονικής σειριακής θύρας είναι απαραίτητη η εγκατάσταση στον υπολογιστή του driver του μετατροπέα.



Σχήμα 3.23 – Το κύκλωμα του μετατροπέα.



Σχήμα 3.24 – Ο μετατροπέας UART-USB.

Αποστολή δεδομένων

Για την αποστολή των αποθηκευμένων μετρήσεων μέσω της σειριακής επικοινωνίας είναι απαραίτητη η αρχικοποίηση του USART μέσα στον κώδικα του μικροελεγκτή. Με την βοήθεια της βιβλιοθήκης UART του MikroC for PIC compiler, ο έλεγχος της επικοινωνίας γίνεται αρκετά εύκολος. Για την αρχικοποίηση αρκεί μόνο η εντολή `UART1_Init()` ακολουθούμενη από το επιθυμητό baud rate μέσα στην παρένθεση. Αφού η επικοινωνία γίνεται με έναν υπολογιστή, το baud rate πρέπει να έχει την τιμή 9600 οπότε η εντολή που προκύπτει είναι `UART1_Init(9600)`.

Μπορούμε να ελέγξουμε αν ο υπολογιστής στέλνει δεδομένα στον μικροελεγκτή χρησιμοποιώντας την συνθήκη `UART1_Data_Ready()` η οποία επιστρέφει 1 αν υπάρχουν δεδομένα στον buffer και 0 αν δεν υπάρχουν. Για να στείλουμε δεδομένα χρησιμοποιούμε την εντολή `UART1_Write()` ακολουθούμενη από μια οκταψήφια μεταβλητή.

Εφόσον ο μικροϋπολογιστής έχει λάβει εντολή από τον υπολογιστή να στείλει τις αποθηκευμένες μετρήσεις, εμφανίζεται στην LCD οθόνη το μήνυμα “Sending” και ξεκινάει η

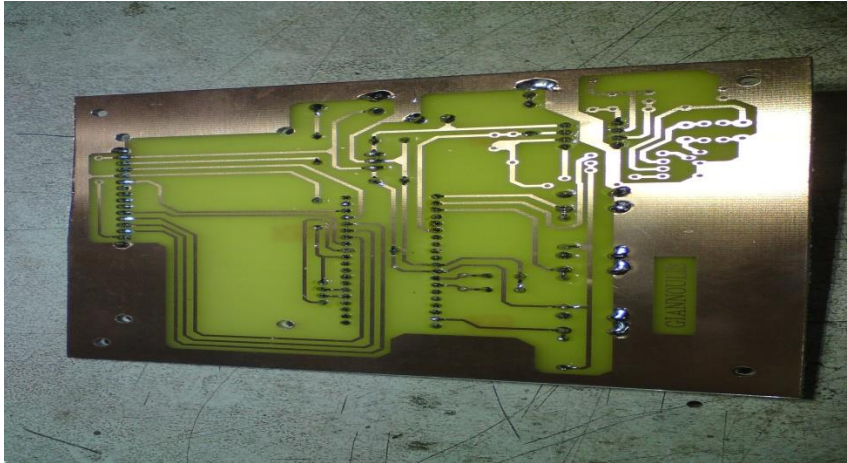
διαδικασία αποστολής. Ο μικροελεγκτής ελέγχει τις θέσεις μνήμης μία προς μία όπως και στις προηγούμενες περιπτώσεις. Αν μια θέση μνήμης δεν είναι κενή στέλνει το περιεχόμενο της στον υπολογιστή ενώ αν είναι κενή δεν στέλνει τίποτα. Όταν σταλούν όλες οι αποθηκευμένες μετρήσεις, στέλνει στον υπολογιστή τον αριθμό 2 σηματοδοτώντας πως δεν θα σταλούν άλλα δεδομένα. Στην συνέχεια αδειάζουμε τον buffer και εμφανίζουμε το μήνυμα “Data was sent”. Και σε αυτήν την περίπτωση κάνουμε χρήση της for :

```
for (i=0;i<255;i++)  
  
{rom=EEPROM_Read(i);  
  
    if (rom!=0xFF)  
  
        {measurement=rom;  
  
          UART1_Write(measurement);  
  
          }  
  
    }  
  
UART1_Write(stopnum);
```

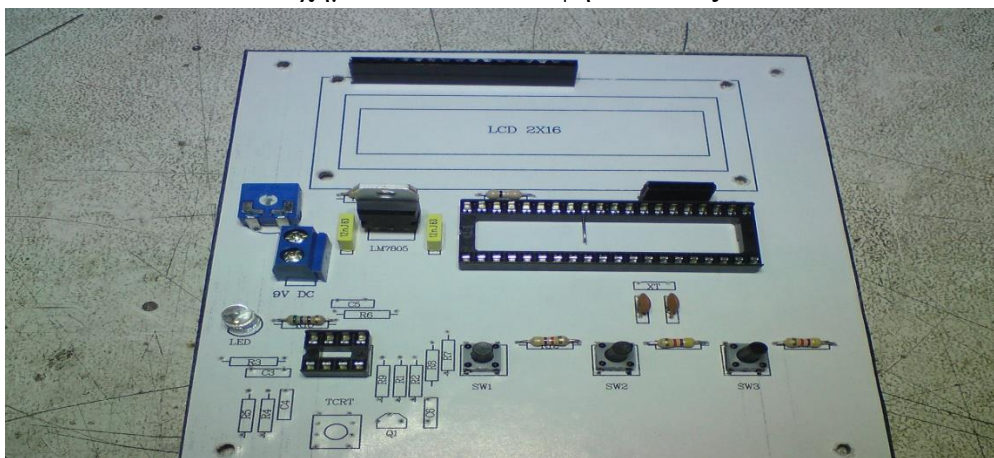
Με το πέρας της αποστολής των μετρήσεων, επιστρέφουμε στην αρχική κατάσταση της συσκευής όπου μπορούμε να διαλέξουμε εκ νέου μια λειτουργία.

3.5 Κόστος κατασκευής

Βασικός άξονας κατά τον σχεδιασμό της κατασκευής ήταν να κρατηθεί χαμηλό το κόστος. Ο μικροελεγκτής στοίχισε 3,80 ευρώ ενώ αισθητήρας και usb-uart μετατροπέας κόστισαν 1,80 ευρώ και 1 ευρώ αντίστοιχα. Το μεγαλύτερο μέρος των χρημάτων πήγε στην κατασκευή της πλακέτας ενώ αντίθετα το κόστος πυκνωτών και αντιστάσεων ήταν αμελητέο. Μικρό ήταν και το κόστος του τελεστικού ενισχυτή αφού ο LM358 στοιχίζει μόλις 30 λεπτά. Η οθόνη LCD 16 x 2 στοιχίζει 6 ευρώ.



Σχήμα 3.25 – Πίσω όψη πλακέτας.



Σχήμα 3.26- Μπροστινή όψη πλακέτας.

Συμπερασματικά το κόστος των υλικών είναι χαμηλό όμως το κόστος κατασκευής μεγαλώνει εξαιτίας των υλικών για την κατασκευή της πλακέτας. Το τελικό κόστος κατασκευής ήταν 33 ευρώ από τα οποία τα 20 αντιστοιχούν στην πλακέτα.

4. ΛΟΓΙΣΜΙΚΟ Η/Υ ΓΙΑ ΤΗΝ ΜΕΤΑΦΟΡΑ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ ΜΕΤΡΗΣΕΩΝ

4.1 Η γλώσσα προγραμματισμού Processing

Για την επικοινωνία του υπολογιστή με την συσκευή είναι απαραίτητη η ύπαρξη λογισμικού που αναλαμβάνει την μεταφορά και αποθήκευση των δεδομένων. Για αυτό τον σκοπό επιλέξαμε να δημιουργήσουμε δύο προγράμματα σε γλώσσα Processing, ένα για την μεταφορά και αποθήκευση των μετρήσεων στον υπολογιστή και ένα για την απεικόνιση των μετρήσεων αυτών σε διάγραμμα.

Η Processing είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα και παράλληλα ένα προγραμματιστικό περιβάλλον. Είναι βασισμένη στη Java όμως το συντακτικό της είναι απλουστευμένο και το προγραμματιστικό της μοντέλο βασίζεται στα γραφικά. Προγράμματα που έχουν γραφτεί σε Processing είναι μεταφρασμένα σε Java και έπειτα τρέχουν σαν Java προγράμματα. Υπάρχουν αρκετές βιβλιοθήκες επέκτασης της γλώσσας για εφαρμογές ήχου, βίντεο κ.α. Ο κώδικας που γράφεται σε αυτήν, μπορεί να γίνει export σε αρχείο jar και να τρέχει μέσα από τον φυλλομετρητή (firefox,chrome κλπ) ή μπορεί να γίνει εκτελέσιμο για όλες τις πλατφόρμες το οποίο τα τρέχει χωρίς να είναι απαραίτητη η Processing, το μόνο που απαιτείται είναι ο JRE (Java Runtime Environment). Η δομή ενός προγράμματος σε Processing είναι η ακόλουθη:

```
//δήλωση μεταβλητών

void setup()

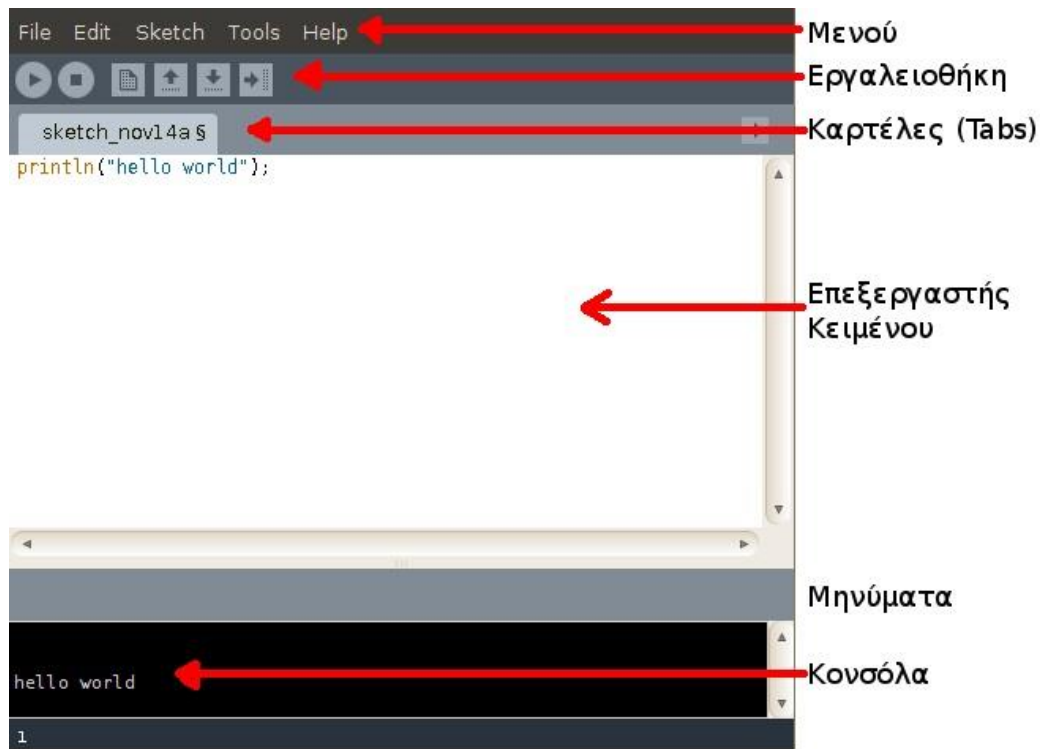
{ //εντολές αρχικοποίησης

}

void draw()

{ //κώδικας

}
```



Σχήμα 4.1 – Το περιβάλλον της Processing.

Η συνάρτηση `setup()` εκτελείται στην αρχή του προγράμματος και για μία μόνο φορά. Χρησιμοποιείται για τις αρχικοποιήσεις ιδιοτήτων και βιβλιοθηκών, όπως το μέγεθος της οθόνης, το χρώμα του φόντου, τη φόρτωση εικόνων κλπ. Οι μεταβλητές που δηλώνονται στην `setup()` δεν είναι προσβάσιμες από άλλες συναρτήσεις, συμπεριλαμβανομένης και της `draw()`. Η συνάρτηση `draw()` εκτελείται μετά την `setup()` και ο κώδικας που γράφεται μέσα στην συνάρτηση αυτή εκτελείται συνεχώς μέχρι να τερματιστεί η εφαρμογή ή μέχρι να κληθεί η συνάρτηση `noLoop()`. Μια παραλλαγή της Processing χρησιμοποιείται από το Arduino για τον προγραμματισμό μικροελεγκτών AVR.

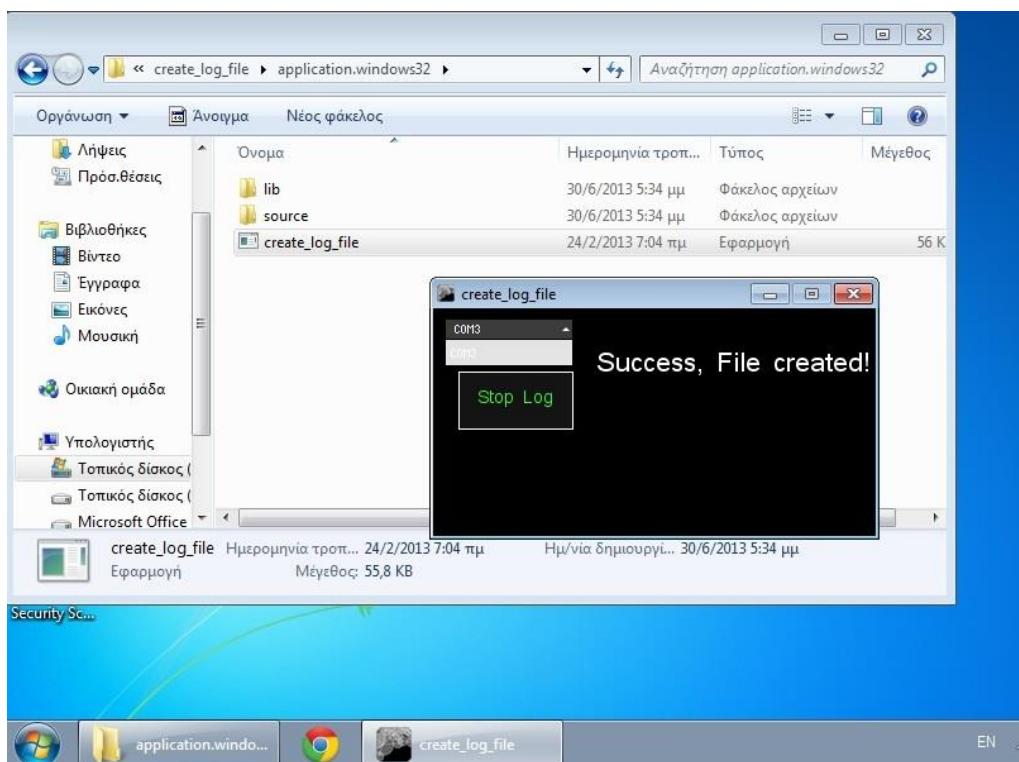
4.2 Το πρόγραμμα Create Log File

Το πρώτο από τα δύο προγράμματα αναλαμβάνει την μεταφορά των μετρήσεων της συσκευής και την αποθήκευσή τους σε ένα αρχείο txt για περαιτέρω επεξεργασία από τον χρήστη. Για να το κάνει αυτό, χρειάζεται να εντοπίσει πόσες εικονικές σειριακές θύρες είναι ενεργές την δεδομένη στιγμή ώστε να επιλέξουμε αυτήν που αντιστοιχεί στην συσκευή για τον δικό μας υπολογιστή. Αυτό είναι απαραίτητο γιατί σε δύο διαφορετικούς υπολογιστές θα

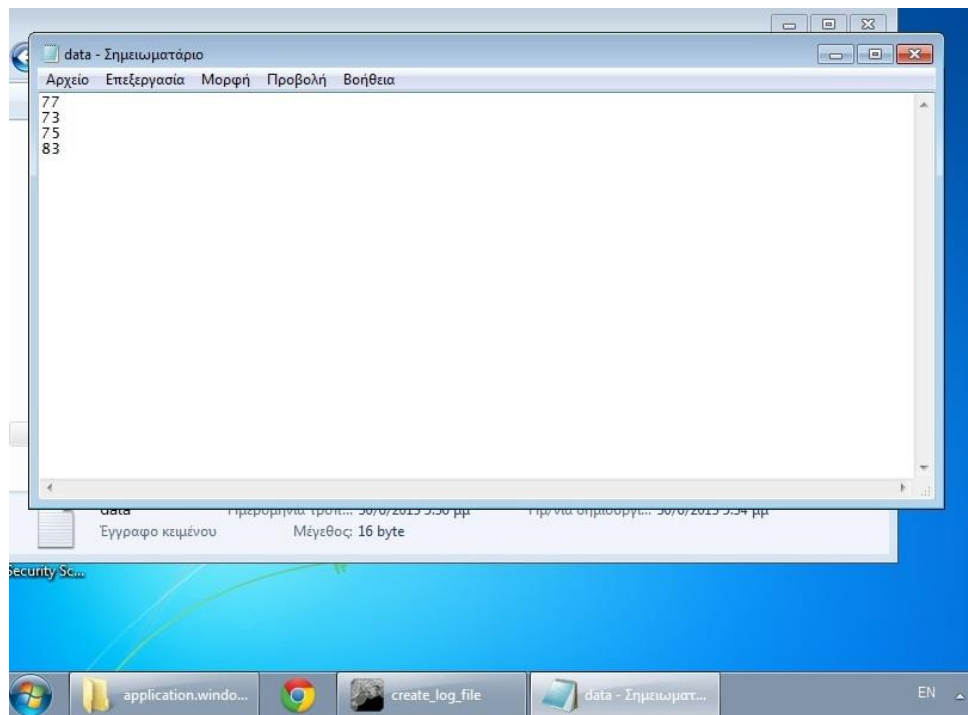
δημιουργηθούν δύο διαφορετικές εικονικές σειριακές θύρες για την ίδια συσκευή, για παράδειγμα στον έναν θα δημιουργηθεί η COM3 και στον άλλο η COM6.

Η βιβλιοθήκη σειριακής επικοινωνίας της processing ωστόσο κατά την αρχικοποίηση χρειάζεται την COM που θα χρησιμοποιήσουμε. Για τον λόγο αυτό υπάρχει μια drop list με τις ενεργές σειριακές θύρες από την οποία επιλέγουμε αυτήν που αντιστοιχεί στην συσκευή ώστε να ενεργοποιηθεί η σειριακή επικοινωνία.

Στην συνέχεια, πατώντας το κουμπί με την ένδειξη “Start Log” ο υπολογιστής στέλνει την συσκευή την εντολή “ok” που σημαίνει πως είναι έτοιμος να δεχθεί τα αποθηκευμένα δεδομένα. Το κουμπί αλλάζει και τώρα έχει την ένδειξη “Stop Log” σε περίπτωση που θελήσουμε να επαναλάβουμε την διαδικασία. Μόλις η συσκευή στείλει τις αποθηκευμένες μετρήσεις, στέλνει στον υπολογιστή τον αριθμό 2 που σηματοδοτεί το τέλος της μεταφοράς δεδομένων. Το πρόγραμμα στην συνέχεια μεταφέρει τα δεδομένα που μόλις έλαβε σε ένα αρχείο txt και εμφανίζει την ένδειξη “Success, File created!”. Το αρχείο αυτό που περιέχει τις μετρήσεις μπορεί να χρησιμοποιηθεί για την δημιουργία διαγράμματος με το δεύτερο πρόγραμμα ή να εισαχθεί στο excel για περαιτέρω επεξεργασία.



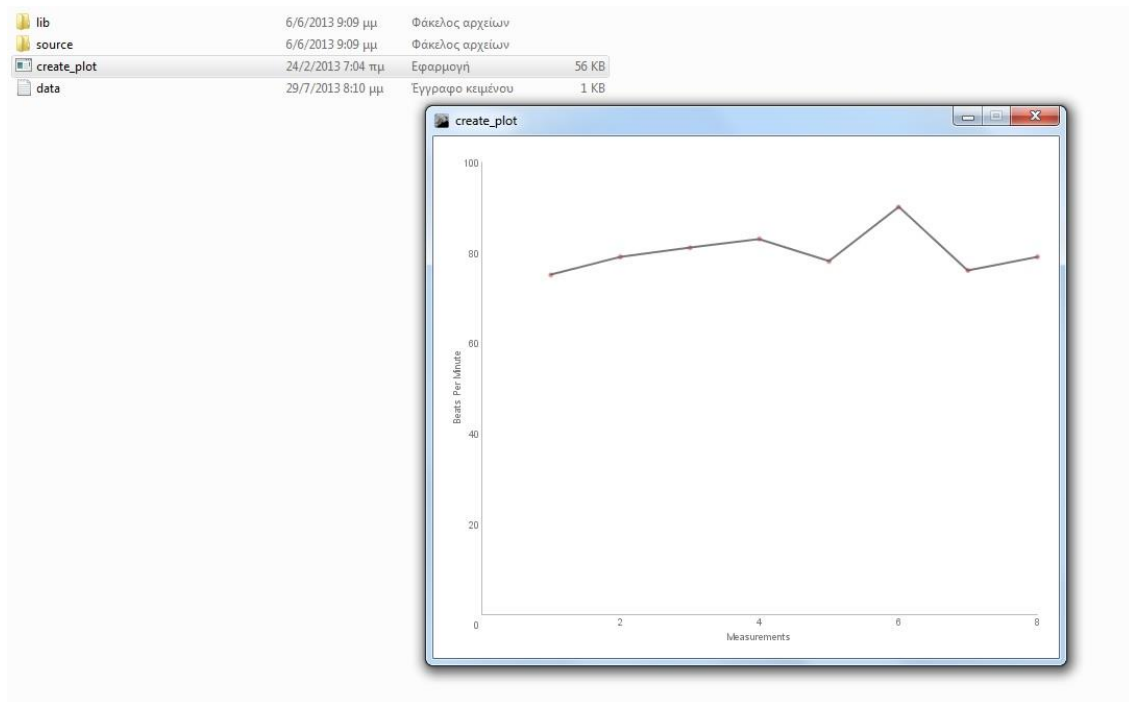
Σχήμα 4.2 – Το πρόγραμμα Create Log File.



Σχήμα 4.3 – Το αρχείο txt με τις αποθηκευμένες μετρήσεις.

4.3 Το πρόγραμμα Create Plot

Το δεύτερο από τα δύο προγράμματα αναλαμβάνει να εξάγει τα δεδομένα από το αρχείο txt και να τα παρουσιάζει σε διάγραμμα από την παλαιότερη μέτρηση προς την πιο πρόσφατη. Δημιουργεί ένα διάγραμμα x-y όπου στον άξονα x είναι ο αύξων αριθμός των μετρήσεων ενώ στον y ο καρδιακός ρυθμός σε beats per minute. Γίνεται και εδώ χρήση έτοιμων βιβλιοθηκών της processing που αναλαμβάνουν την δημιουργία του διαγράμματος αρκεί να τους δοθούν δεδομένα ως είσοδος. Η λειτουργία του είναι απλή για αυτό και ο κώδικας δεν είναι καθόλου πολύπλοκος.



Σχήμα 4.4 – Το πρόγραμμα Create Plot.

Απαραίτητη προϋπόθεση για την λειτουργία και των δύο προγραμμάτων είναι να τοποθετήσουμε τον φάκελο “device software” στην διεύθυνση του σκληρού δίσκου C και όχι κάπου αλλού όπως για παράδειγμα η επιφάνεια εργασίας. Αυτό συμβαίνει γιατί τα προγράμματα χρησιμοποιούν συγκεκριμένο path στον σκληρό δίσκο ώστε να δημιουργηθεί το αρχείο txt αλλά και να διαβαστεί στην συνέχεια με σκοπό την δημιουργία του διαγράμματος. Σχετικές οδηγίες περιέχονται και στον φάκελο “device software” στο αρχείο “Read Me”.

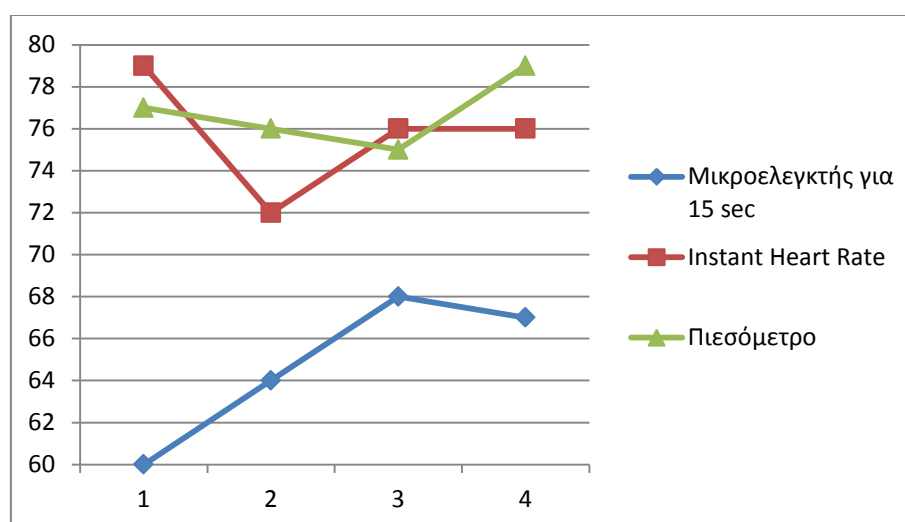
5. ΑΠΟΤΕΛΕΣΜΑΤΑ

Όπως αναφέραμε και στο κεφάλαιο 3, το ιδανικό για την συσκευή μας θα ήταν η κάθε μέτρηση να έχει διάρκεια 1 λεπτό. Αυτό ωστόσο, μειώνει κατά πολύ την διάρκεια ζωής της μπαταρίας κάνοντας την συσκευή ασύμφορη. Για να καταλήξουμε στον κατάλληλο συνδυασμό χρόνου μέτρησης και αριθμητικής πράξης από την οποία θα προκύψει η τελική τιμή του καρδιακού ρυθμού, μετρήσαμε για 15 δευτερόλεπτα πολλαπλασιάζοντας με το 4, για 30 δευτερόλεπτα πολλαπλασιάζοντας με το 2, για 60 δευτερόλεπτα χωρίς να χρειάζεται πολλαπλασιασμός στην συνέχεια και για 30 δευτερόλεπτα πολλαπλασιάζοντας με το 2,1 κρατώντας το ακέραιο μέρος του γινομένου. Κάθε μια μέτρηση για τους παραπάνω χρόνους συγκρίθηκε με την τιμή ενός πιεσόμετρου καθώς και με την τιμή που μας έδινε η εφαρμογή Instant Heart Rate για android κινητά. Ακολουθούν συγκριτικοί πίνακες και διαγράμματα για την κάθε περίπτωση.

5.1 Μέτρηση για 15 δευτερόλεπτα

Πίνακας 5.1 - Μέτρηση για 15 δευτερόλεπτα και πολλαπλασιασμός timer0 με το 4

Μικροελεγκτής για 15sec	Εφαρμογή Instant Heart Rate	Ηλεκτρονικό Πιεσόμετρο
60 bpm	79bpm	77bpm
64bpm	72bpm	76bpm
68bpm	76bpm	75bpm
67bpm	76bpm	79bpm



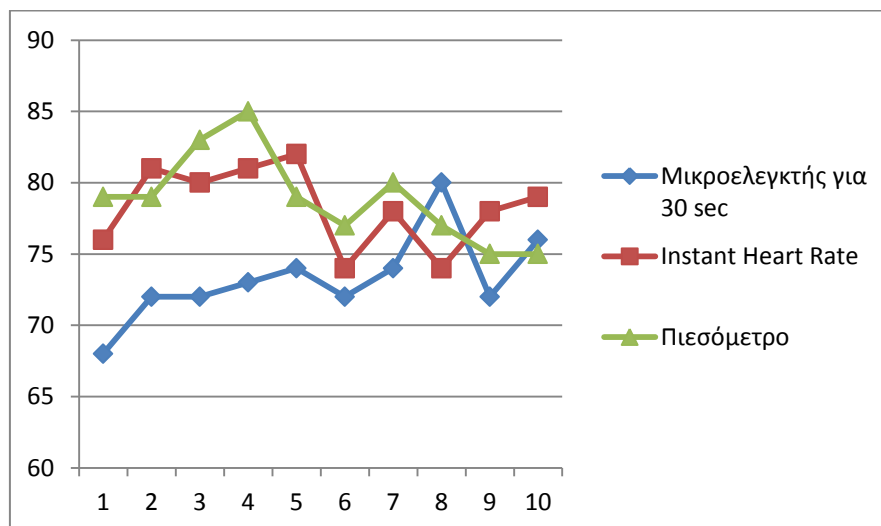
Σχήμα 5.1 – Συγκριτικό διάγραμμα μετρήσεων με τον μΕ να μετράει για 15 sec.

Είναι προφανές ότι η μέτρηση για 15 sec δίνει αποτελέσματα που διαφέρουν κατά πολύ από τα πραγματικά καθώς εμφανίζει απόκλιση από 10% έως και 24% από τις τιμές της εφαρμογής και απόκλιση από 9,3% έως και 22% από τις τιμές του πιεσόμετρου.

5.2 Μέτρηση για 30 δευτερόλεπτα

Πίνακας 5.2 - Μέτρηση για 30 δευτερόλεπτα και πολλαπλασιασμός με το 2

Μικροελεγκτής για 30sec	Εφαρμογή Instant Heart Rate	Ηλεκτρονικό Πιεσόμετρο
68bpm	76bpm	79bpm
72bpm	81bpm	79bpm
72bpm	80bpm	83bpm
73bpm	81bpm	85bpm
74bpm	82bpm	79bpm
72bpm	74bpm	77bpm
74bpm	78bpm	80bpm
80bpm	74bpm	77bpm
72bpm	78bpm	75bpm
76bpm	79bpm	75bpm



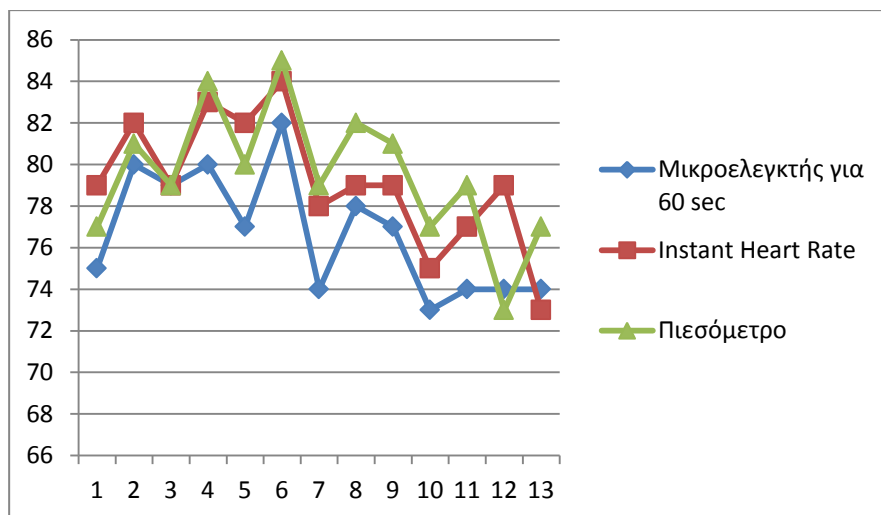
Σχήμα 5.2 – Συγκριτικό διάγραμμα μετρήσεων με τον μΕ να μετράει για 30 sec.

Σε αυτή την περίπτωση οι μετρήσεις μας πλησιάζουν τις θεωρητικές αλλά και πάλι το αποτέλεσμα δεν είναι το καλύτερο δυνατό καθώς ενώ σε ορισμένες μετρήσεις έχουμε μικρή απόκλιση, οι περισσότερες διαφέρουν κατά πολύ. Ενδεικτικά, παρουσιάζεται απόκλιση από τις τιμές της εφαρμογής **από 8% έως και 11%** (μεμονωμένες μετρήσεις έχουν 2-5% απόκλιση αλλά δεν αντιπροσωπεύουν το σύνολο των δειγμάτων). Στην περίπτωση του πιεσόμετρου η απόκλιση έχει τιμές **από 6,4% έως και 14,1%** (μεμονωμένες μετρήσεις έχουν 1,5-4% απόκλιση αλλά και σε αυτήν την περίπτωση δεν αντιπροσωπεύουν το δείγμα).

5.3 Μέτρηση για 60 δευτερόλεπτα

Πίνακας 5.3 - Μέτρηση για 60 δευτερόλεπτα, εδώ δεν χρειάζεται πολλαπλασιασμός γιατί μετράμε για ένα ολόκληρο λεπτό

Μικροελεγκτής για 60sec	Εφαρμογή Instant Heart Rate	Ηλεκτρονικό Πιεσόμετρο
75bpm	79bpm	77bpm
80bpm	82bpm	81bpm
79bpm	79bpm	79bpm
80bpm	83bpm	84bpm
77bpm	82bpm	80bpm
82bpm	84bpm	85bpm
74bpm	78bpm	79bpm
78bpm	79bpm	82bpm
77bpm	79bpm	81bpm
73bpm	75bpm	77bpm
74bpm	77bpm	79bpm
74bpm	79bpm	73bpm
74bpm	73bpm	77bpm



Σχήμα 5.3 Συγκριτικό διάγραμμα μετρήσεων με τον μΕ να μετράει για 60 sec.

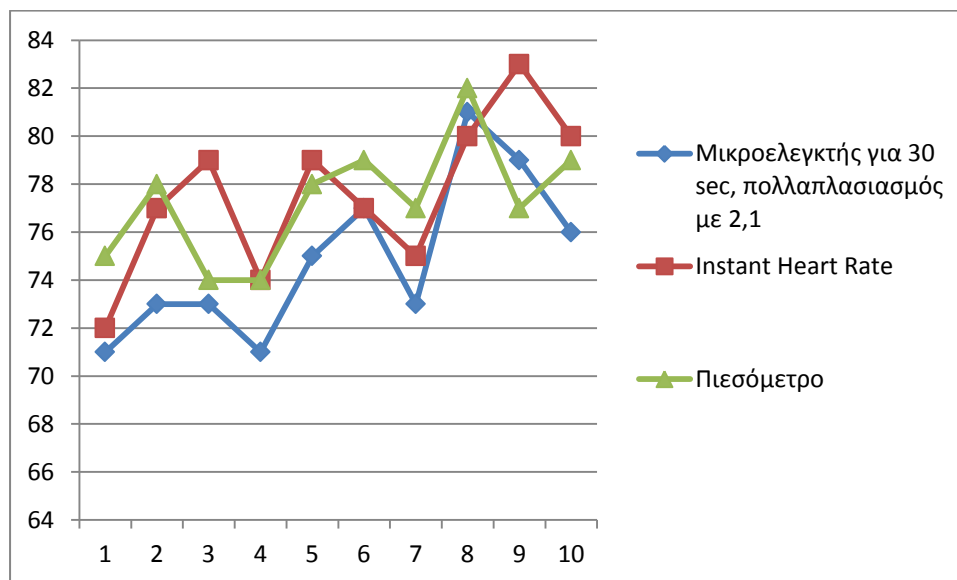
Στην περίπτωση αυτήν έχουμε τις καλύτερες δυνατές μετρήσεις. Η απόκλιση του μικροελεγκτή από την εφαρμογή είναι **από 0% έως και 3,6%** ενώ η απόκλιση από το πιεσόμετρο είναι **από 0% έως και 4,7%**. Για την επιλογή του τελικού χρόνου μέτρησης, ψάξαμε τον συνδυασμό χρόνου μέτρησης και αριθμητικής πράξης που θα μας δώσει αποτελέσματα όσο το δυνατόν πιο κοντά σε αυτά που θα είχαμε αν μετρούσαμε για 60 δευτερόλεπτα. Έτσι, καταλήξαμε στην μέτρηση για 30 δευτερόλεπτα πολλαπλασιάζοντας στην συνέχεια με το 2,1 κρατώντας μόνο το ακέραιο μέρος του γινομένου χωρίς καμία στρογγυλοποίηση για να αποφύγουμε την αλλοίωση του δείγματος μας.

5.4 Τελική επιλογή διάρκειας μέτρησης

Πίνακας 5.4 - Μέτρηση για 30 δευτερόλεπτα και πολλαπλασιασμός με το 2,1 κρατώντας μόνο το ακέραιο μέρος του γινομένου χωρίς στρογγυλοποίηση

Μικροελεγκτής για 30sec	Εφαρμογή Instant Heart Rate	Ηλεκτρονικό Πιεσόμετρο
71bpm	72bpm	75bpm
73bpm	77bpm	78bpm
73bpm	79bpm	74bpm
71bpm	74bpm	74bpm
75bpm	79bpm	78bpm
77bpm	77bpm	79bpm
73bpm	75bpm	77bpm
81bpm	80bpm	82bpm

79bpm	83bpm	77bpm
76bpm	80bpm	79bpm



Σχήμα 5.4 – Συγκριτικό διάγραμμα μετρήσεων τελικής επιλογής.

Με αυτόν τον συνδυασμό έχουμε απόκλιση σε σχέση με την εφαρμογή **από 0% έως και 5%** ενώ σε σχέση με το πιεσόμετρο, η απόκλιση είναι **από 1,2% έως και 5,3%**. Αξίζει να σημειωθεί ότι μεγάλη απόκλιση μπορεί να παρουσιαστεί ακόμη και ανάμεσα σε δύο διαδοχικές μετρήσεις καθώς ο αισθητήρας είναι πού ευαίσθητος στην κίνηση. Ακολουθεί συγκεντρωτικός πίνακας της απόκλισης με βάση τον χρόνο μέτρησης.

Πίνακας 5.5 – Απόκλιση μετρήσεων

Μέθοδος μέτρησης	Απόκλιση
Μέτρηση για 15 sec- πολλαπλασιασμό με το 4	10%-24% / 9,3%-22%
Μέτρηση για 30 sec - πολλαπλασιασμός με το 2	8%-11% / 6,4%-14,1%
Μέτρηση για 60 sec - χωρίς πολλαπλασιασμό	0%-3,6% / 0%- 4,7%
Μέτρηση για 30 sec - πολλαπλασιασμός με το 2,1	0%-5% / 1,2%- 5,3%

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

6.1 Πλεονεκτήματα

Κύριο πλεονέκτημα της κατασκευής είναι το γεγονός ότι έχει χαμηλό κόστος καθώς δεν χρησιμοποιούμε ακριβά υλικά. Επίσης, είναι πολύ εύκολη στην χρήση αφού απλά τοποθετούμε τον δείκτη του αριστερού χεριού επάνω στον αισθητήρα. Παρέχεται η δυνατότητα αποθήκευσης των μετρήσεων και μεταφοράς τους στον υπολογιστή για περαιτέρω επεξεργασία είτε από το πρόγραμμα που συνοδεύει την κατασκευή είτε από το Excel.

Επιπλέον, υπάρχει απεικόνιση των αποτελεσμάτων σε οθόνη και δυνατότητα να παρακολουθούμε τον καρδιακό ρυθμό κατά την διάρκεια της μέτρησης με την βοήθεια του led που αναβοσβήνει. Κυκλωματικά, μπορούμε να έχουμε σήμα εισόδου μέχρι την τάξη των 9-16 V (αλλάζοντας φυσικά τις τροφοδοσίες) ενώ το κύκλωμα μπορεί να αντέξει ρεύμα μέχρι και 1 Ampere. Επίσης, ο αισθητήρας διαθέτει ειδικά φίλτρα για να μην επηρεάζεται από το φως της ημέρας κατά την μέτρηση ενώ η λειτουργία του αισθητήρα και των φίλτρων μόνο για 13 δευτερόλεπτα σε κάθε μέτρηση εξοικονομεί ενέργεια και επιμηκύνει την ζωή της μπαταρίας.

6.2 Μειονεκτήματα

Δυστυχώς, ο αισθητήρας είναι πολύ ευαίσθητος στην κίνηση με αποτέλεσμα να εισάγεται απόκλιση στην μέτρηση με την παραμικρή κίνηση του δείκτη. Για τον λόγο αυτό μπορεί να παρατηρηθεί αρκετή διαφορά ακόμα και μεταξύ δύο διαδοχικών μετρήσεων. Επίσης, εξαιτίας του Voutput swing του τελεστικού ενισχυτή, όταν η μπαταρία πέσει κάτω από 7.5 V, στην έξοδο των φίλτρων δεν παίρνουμε παλμό TTL με αποτέλεσμα οι μετρήσεις να είναι λανθασμένες.

Επίσης, υπάρχει μεγαλύτερο περιθώριο σφάλματος όταν είμαστε κοντά σε συσκευές με ηλεκτρομαγνητική ακτινοβολία ενώ το μέγεθος της κατασκευής δεν είναι βολικό για να το μεταφέρουμε πάνω μας πχ σε μία τσέπη. Τέλος, το ρεύμα που καταναλώνει η συσκευή μας υποχρεώνει να μειώσουμε τον χρόνο μέτρησης σε σχέση με το ιδανικό για να εξοικονομήσουμε ενέργεια.

6.3 Τρόποι βελτίωσης

Η λειτουργία της συσκευής θα μπορούσε να βελτιωθεί με τον επανασχεδιασμό των φίλτρων, χρησιμοποιώντας διαφορετικό τελεστικό ενισχυτή ο οποίος θα έχει καλύτερο Voutput swing και να συμπεριφέρεται καλύτερα όταν πέσει η στάθμη της μπαταρίας. Θα μπορούσαμε επίσης να βλέπουμε τις αποθηκευμένες μετρήσεις χωρίς να απαιτείται η σύνδεση με τον υπολογιστή αλλά κατευθείαν από την μνήμη. Επιπλέον, μπορεί να προστεθεί φωνητική ανάγνωση των αποτελεσμάτων σε κάθε μέτρηση και να μετατραπεί η επικοινωνία με τον υπολογιστή από σειριακή σε ασύρματη.

Εάν επανασχεδιαστεί η πλακέτα μπορούμε να έχουμε μια συσκευή μικρότερου μεγέθους άρα πιο εύκολο στο να μεταφερθεί. Αυτό ωστόσο πιθανό να αυξήσει το κόστος κατασκευής. Επιπλέον, αν αλλάξουμε τις τιμές των φίλτρων μπορούμε να έχουμε μια συσκευή σχεδιασμένη ειδικά για παιδιά ή για άτομα με χαμηλό καρδιακό ρυθμό.

Με την αλλαγή του αισθητήρα, η κατασκευή θα μπορούσε να μετατραπεί για να βλέπουμε τον καρδιακό ρυθμό σε πραγματικό χρόνο και να ενσωματωθεί σε μία συσκευή όπως ένα όργανο γυμναστικής ή μια συσκευή για την παρακολούθηση κατάκοιτων ασθενών.

Κώδικας για τον PIC16F887

```
sbit Tr_base at RA0_bit;
sbit Switch at RA1_bit;
sbit Mem_button at RD0_bit;
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
unsigned int pulsecount,pulserate;
unsigned int i,ADD,temp;
unsigned int num,rom,measurement;
int state=0;
char message1[] = "Press button";
char message2[] = "Heartrate= bpm";
char message3[] = "Saving data ";
char message4[] = "Data Saved";
char message5[] = "Sending";
char message6[] = " remaining";
char message7[] = "Delete memory?";
```

```

char message8[] = "Memory deleted";
char message9[] = "Data was sent";
char message10[] = "Done.";
char *bpm = "000";
char *remain = "000";
int flag,empty,j,k,trash;
char *input;
int stopnum=2;
void Display_Freq(unsigned int bpm2write)
{
    bpm[0] = (bpm2write/100)%10 + 48;
    bpm[1] = (bpm2write/10)%10 + 48;
    bpm[2] = bpm2write%10 + 48;
    Lcd_Out(1, 11, bpm);
}
void Display_Remaining(unsigned int rem2write)
{
    remain[0] = (rem2write/100)%10 + 48;
    remain[1] = (rem2write/10)%10 + 48;
    remain[2] = rem2write%10 + 48;
    Lcd_Out(1, 1, remain);
}
void main()
{
    OSCCON=0b00001000;
    ANSEL=0x00;
    ANSELH=0x00;
    CM1CON0=0x00;
    CM2CON0=0x00;

```

```

TRISD=0b00000001;

TRISC=0b10000000;

TRISB=0b00000000;

TRISA=0b00010010;

OPTION_REG=0b01111000;

TXSTA=0b00101110;

RCSTA=0b10010000;

BAUDCTL=0b00000000;

Lcd_Init();          // Initialize LCD

Lcd_Cmd(_LCD_CLEAR); // CLEAR display

Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off

Lcd_Out(1,1,message1); // Write message1 in 1st row

while (1)

{ Tr_base=0;

  pulserate=0;

  pulsecount=0;

  if(Switch == 0)

  {

    Lcd_Out(1,1,message2);

    delay_ms(300);

    Tr_base=1;

    delay_ms(300);

    delay_ms(300);

    TMR0=0;

    delay_ms(30000);

    Tr_base=0;

    pulsecount=TMR0;

    pulserate=pulsecount*2.1;

```



```

delay_ms(500);

Display_Freq(pulserate);

Delay_ms(1500);          // Wait for 1,5 sec

Lcd_Cmd(_LCD_CLEAR);    // Clear display

Delay_ms(100);

}

if (pulserate>0)

{

Lcd_Out(1,1,message3);

delay_ms (500);

ADD=0;

for (i=0;i<255;i++)

{ temp=EEPROM_Read(ADD+i);

if (temp == 0xFF)

{

EEPROM_Write(ADD+i,pulserate);

i=256;

}

}

Lcd_Cmd(_LCD_CLEAR);

delay_ms(100);

Lcd_Out(1,1,message4);

delay_ms(1000);

Lcd_Cmd(_LCD_CLEAR);

Lcd_Out(1,1,message1);

}

if(Button(&PORTD, 0, 1, 0) )

```

```

    {state++;
    }
switch(state)
{ case 1:
    ADD=0;
    for (j=0;j<255;j++)
    { temp=EEPROM_Read(ADD+j);
      if (temp == 0xFF)
      { flag=j;
        j=256;
      }
    }
    Lcd_Cmd(_LCD_CLEAR);
    empty=255-flag;
    Lcd_Out(1,1,message6);
    Display_Remaining(empty);
    delay_ms (1500);
    Lcd_Cmd(_LCD_CLEAR);
    delay_ms(100);
    Lcd_Out(1,1,message7);
    delay_ms (1000);
break;
case 2:
    ADD=0;
    for(k=0;k<255;k++)
    { temp=EEPROM_Read(ADD+k);
      if (temp != 0xFF)
      { EEPROM_Write(ADD+k,0xFF);

```

```

        delay_ms(100);
    }
}

Lcd_Cmd(_LCD_CLEAR);
delay_ms(100);
Lcd_Out(1,1,message8);
delay_ms(1000);
Lcd_Cmd(_LCD_CLEAR);
state=0;

break;

default :
state=0;

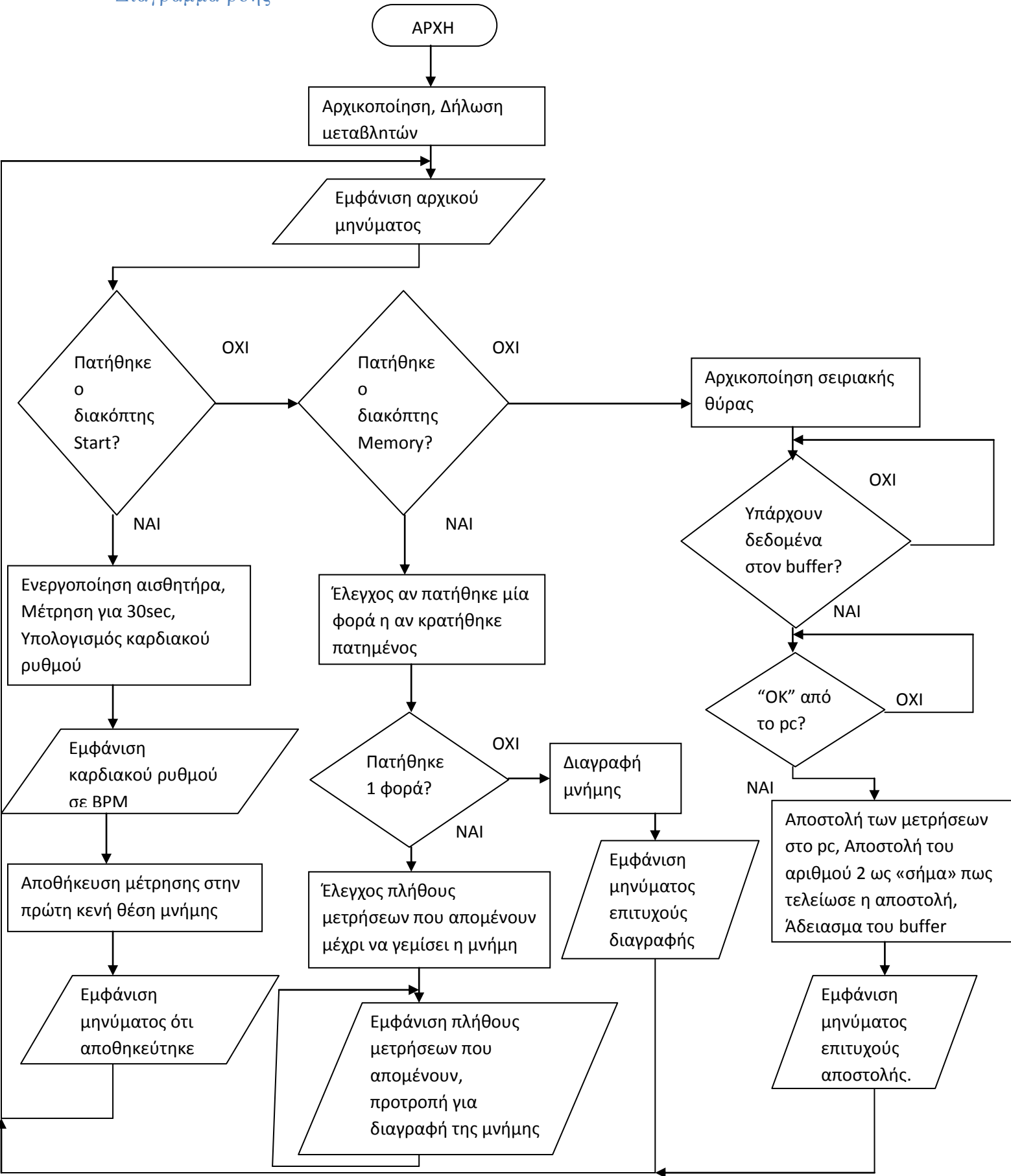
Lcd_Out(1,1,message1);
}

UART1_Init(9600);
delay_ms(100);
if ( UART1_Data_Ready()==1 )
{
    UART1_Read_Text(input, "ok", 10);
    if ( input !=0)
    { Lcd_Cmd(_LCD_CLEAR);
      Lcd_Out(1,1,message5);
      delay_ms(1000);
      ADD=0;
      for (i=0;i<255;i++)
      {
          rom=EEPROM_Read(ADD+i);
          if (rom!=0xFF)

```

```
{measurement=rom;
  UART1_Write(measurement);
}
}
UART1_Write(stopnum);
while (UART1_Data_Ready())
{
  trash = UART1_Read();
}
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(1,1,message9);
delay_ms(1500);
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(1,1,message10);
delay_ms(1500);
  Lcd_Cmd(_LCD_CLEAR);
}
}
}
}
```

Διάγραμμα ροής



ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Design and analysis of artifact-resistive finger photoplethysmographic sensors for vital sign monitoring**, Sokwoo Ree , MIT 2000
2. **Heart rate measurement from the finger using a low-cost microcontroller**, Dogan Ibrahim, Kadri Buruncuk
3. **Multi-wavelength photoplethysmography method for skin microcirculation assessment**, Lasma Gailite, University of Latvia
4. **Optical noninvasive monitoring of skin blood pulsations**, Janis Spigulis, Optical Society of America, 2005
5. **Photoplethysmography and its application in clinical physiological measurement**, John Allen, 2007
6. **Photoplethysmography (PPG) System** , Geert Langereis, February 2010
7. **Εισαγωγή στους μικροελεγκτές PICmicro**, Δρ. Σταμάτης Αλατσαθανός, Αθήνα 2007

1. <http://en.wikipedia.org/wiki/Photoplethysmograph>
2. <http://www.btcenter.gr/userfiles/LRR-PPG.php.content.htm>
3. <http://blog.hamed.gr/%CE%BF%CE%BE%CF%8D%CE%BC%CE%B5%CF%84%CF%81%CE%BF>
4. <http://www.instructables.com/id/Microcontroller-measures-heart-rate-through-finger/>
5. <http://www.lymphedemapeople.com/wiki/doku.php?id=glossary:photoplethysmography>
6. http://www.gooligum.com.au/tutorials/baseline/PIC_Base_A_5.pdf
7. <http://ww1.microchip.com/downloads/en/DeviceDoc/51682A.pdf>
8. <http://ww1.microchip.com/downloads/en/DeviceDoc/51702A.pdf>
9. <http://stackoverflow.com/questions/13778173/how-to-use-processing-to-plot-serial-data>
10. <http://www.gicentre.org/utills/chart/>
11. http://learning.codasign.com/index.php?title=Using_Serial_Events_in_Processing