



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ  
& ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σχεδίαση και υλοποίηση φορητής συσκευής για την  
εκτίμηση της αρτηριακής πίεσης μετρώντας το χρόνο  
διάδοσης παλμού**

Διπλωματική Εργασία

**ΖΩΓΡΑΦΟΣ ΑΓΓΕΛΟΣ**

**Επιβλέπων :** Πάυλος-Πέτρος Σωτηριάδης  
Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2019





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ  
& ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σχεδίαση και υλοποίηση φορητής συσκευής για την  
εκτίμηση της αρτηριακής πίεσης μετρώντας το χρόνο  
διάδοσης παλμού**

Διπλωματική Εργασία

**ΖΩΓΡΑΦΟΣ ΑΓΓΕΛΟΣ**

**Επιβλέπων :** Παύλος-Πέτρος Σωτηριάδης  
Αναπλ. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Φεβρουαρίου 2019

.....  
Σωτηριάδης Παύλος-Πέτρος  
Αναπλ. Καθηγητής Ε.Μ.Π.

.....  
Νικήτα Κωνσταντίνα  
Καθηγήτρια Ε.Μ.Π.

.....  
Ματσόπουλος Γεώργιος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2019



.....  
Ζωγράφος Άγγελος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ζωγράφος Άγγελος, 2019

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.



# Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε κατά το ακαδημαϊκό έτος 2018-2019 υπό την επίβλεψη του κ. Π.Π.Σωτηριάδη, καθηγητή του Ε.Μ.Π., της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, στον οποίο οφείλω ιδιαίτερες ευχαριστίες για την ανάθεσή της και την ευκαιρία που μου έδωσε να ασχοληθώ με ένα θέμα που με ενδιέφερε πραγματικά.

Θα ήθελα επίσης να ευχαριστήσω όλους τους συναδέλφους μου από το Circuits & Systems Group, οι οποίοι βοήθησαν στην ολοκλήρωσή της, με τις συμβουλές και την εμπειρία τους.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στην κ. Κ.Νικήτα, καθηγήτρια του Ε.Μ.Π., η οποία μέσα από τα μαθήματα της, μου δημιούργησε το ενδιαφέρον για την βιοϊατρική μηχανική.





# Περίληψη

Στην παρούσα εργασία παρουσιάζεται η σχεδίαση και η υλοποίηση μιας φορητής συσκευής εκτίμησης της αρτηριακής πίεσης μέσω της μέτρησης του χρόνου διάδοσης παλμού. Για την μέτρηση του χρόνου αυτού, η συσκευή καταγράφει και επεξεργάζεται δύο βιοσήματα: Της φωτοπληθυσμογραφίας και του φωνοκαρδιογραφήματος. Παρουσιάζεται η σχεδίαση και κατασκευή των αισθητήρων, των αναλογικών και ψηφιακών κυκλωμάτων και μιας πλακέτας PCB για το σύνολο των κυκλωμάτων, καθώς επίσης και ο προγραμματισμός του μικροελεγκτή που χρησιμοποιείται. Τέλος παρουσιάζεται ένα παράδειγμα χρήσης της συσκευής.

## Λέξεις Κλειδιά

αρτηριακή πίεση, χρόνος διάδοσης παλμού, χρόνος άφιξης παλμού, φωνοκαρδιογράφημα, φωτοπληθυσμογραφία, STM32, PCB, τελεστικός ενισχυτής, επεξεργασία βιοσημάτων



# **Abstract**

This work presents the design and implementation of a portable device capable of estimating blood pressure, by measuring the Pulse Transit Time. In order to measure this quantity, the device records and processes two biosignals: the photoplethysmogram and the phonocardiogram. The design and construction of the sensors, the analog and digital circuits and the PCB for the whole of the circuits, as well as the programming of the Microcontroller that is being used, are presented. Finally, an example of the device's usage is documented.

## **Keywords**

Blood Pressure, Pulse Transit Time (PTT), Pulse Arrival Time (PAT), Phonocardiogram(PCG), Photoplethysmogram(PPG), STM32, PCB, Operational Amplifier, Biosignal Processing



# Περιεχόμενα

Ευχαριστίες	6
Περίληψη	8
Abstract	10
Περιεχόμενα	12
1.Εισαγωγή.....	13
2. Υπόβαθρο.....	14
2.1 Το κυκλοφορικό σύστημα .....	14
2.2 Ο χρόνος διάδοσης παλμού.....	17
3. Η φορητή συσκευή.....	23
3.1 Οι αισθητήρες.....	23
3.2 Αναλογικό κύκλωμα.....	27
3.3 Ψηφιακό κύκλωμα και ο μικροελεγκτής.....	35
3.4 Τελική κατασκευή.....	37
3.5 Προγραμματισμός.....	41
4. Χρήση της συσκευής.....	44
4.1 Ψηφιακή επεξεργασία των σημάτων.....	44
4.2 Βαθμονόμηση και αποτελέσματα.....	48
5. Επεκτάσεις.....	50
Παράρτημα Α - Κώδικας	52
Παράρτημα Β - Χαρακτηριστικά της συσκευής	75
Πηγες	77

# 1. Εισαγωγή

Στην σημερινή εποχή, η τεχνολογία αποτελεί αναπόσπαστο κομμάτι της κοινωνίας αλλά και της ζωής του κάθε ανθρώπου ξεχωριστά. Ο τομέας της υγείας είναι ένας από τους τομείς που έχει επωφεληθεί ιδιαίτερα από την ταχύτατη ανάπτυξή της ( και ίσως ο σημαντικότερος) .

Οι βιοϊατρικοί μηχανικοί είναι αυτοί που εφαρμόζουν ηλεκτρικές,χημικές,οπτικές, μηχανικές και άλλες αρχές, ώστε να δημιουργήσουν εργαλεία που επιτρέπουν στους γιατρούς, βιολόγους, νοσηλευτές και γενικά τους άμεσα σχετιζόμενους με την υγεία επαγγελματίες, να κατανοήσουν, να τροποποιήσουν ή να ελέγξουν βιολογικά συστήματα (ανθρώπων ή ζώων) με αποτέλεσμα να διευκολύνουν και να επιταχύνουν το έργο τους ή ακόμα και να δημιουργούν νέους δρόμους έρευνας ή και μελέτης.

Σύμφωνα με τον παγκόσμιο οργανισμό υγείας, σε μια μελέτη που έγινε το 2012, οι ασθένειες του καρδιαγγειακού συστήματος ευθύνονται συνολικά για το 31% των θανάτων, γεγονός που τις τοποθετεί στην πρώτη θέση των αιτιών θανάτου παγκοσμίως. Άμεσο συμπέρασμα αποτελεί το γεγονός πως η μελέτη και η έρευνα που σχετίζεται με την ανάπτυξη μεθόδων εκτίμησης της υγείας του καρδιαγγειακού συστήματος έχει μεγάλη αξία και σημασία.

Ένας πολύ σημαντικός δείκτης της καρδιαγγειακής υγείας είναι η αρτηριακή πίεση. Η μέτρησή της είναι δυνατό να επιτευχθεί με διάφορους τρόπους, ο καθένας με τα δικά του μειονεκτήματα και πλεονεκτήματα. Στην παρούσα εργασία θα ασχοληθούμε με έναν αρκετά υποσχόμενο τρόπο που τελευταία έχει προσεγγίσει το ενδιαφέρον αρκετών ερευνητών, ένα από τα πλεονεκτήματα του οποίου είναι πως επιτρέπει την συνεχή παρακολούθηση της. Η συγκεκριμένη μέθοδος στηρίζεται στην σχέση της αρτηριακής πίεσης με μία παράμετρο του κυκλοφορικού συστήματος που ονομάζεται χρόνος διάδοσης παλμού, η μέτρηση της οποίας λοιπον μπορεί να χρησιμοποιηθεί για την εκτίμησή της.

## 2. Υπόβαθρο

Πριν μπορέσουμε να αναλύσουμε τη συγκεκριμένη μέθοδο εκτίμησης της αρτηριακής και προχωρήσουμε στην σχεδίαση της συσκευής που υλοποιεί αυτή την εκτίμηση, είναι απαραίτητο να παρουσιάσουμε, σε ένα βαθμό, το τί ακριβώς είναι αυτό που καλούμαστε να μετρήσουμε (ο χρόνος διάδοσης παλμού) , και τα μοντέλα εκείνα που συσχετίζουν την μετρούμενη αυτή ποσότητα, με την αρτηριακή πίεση. Σε αυτό το κεφάλαιο λοιπόν, θα παρουσιάσουμε μερικά απαραίτητα στοιχεία από την ανθρώπινη φυσιολογία και στη συνέχεια θα περιγράψουμε το πώς ο χρόνος διάδοσης παλμού μπορεί να χρησιμοποιηθεί για να εκτιμηθεί η αρτηριακή πίεση.

### 2.1 Το κυκλοφορικό σύστημα και η αρτηριακή πίεση

Το αίμα, για να μεταφέρει θρεπτικές ουσίες και οξυγόνο, μεταξύ των άλλων λειτουργιών του, κυκλοφορεί διαρκώς σε ένα κλειστό σύστημα σωλήνων μέσα στο ανθρώπινο σώμα, με την καρδιά να είναι το μύθος όργανο επιφορτισμένο με την ευθύνη για την κυκλοφορία αυτή.

Η καρδιά, κατά τη διάρκεια της ζωής του ανθρώπου, εκτελεί μια περιοδική ενέργεια με την οποία μεταφέρει κινητική ενέργεια στο αίμα προκειμένου αυτό να κυκλοφορεί. Η διαδικασία με την οποία αυτό συμβαίνει (σε μια υγιή καρδιά και εστιάζοντας στην αριστερή κυκλοφορία), πολύ συνοπτικά είναι η εξής:

-Αρχικά όλα τα μέρη της καρδιάς βρίσκονται σε χάλαση.

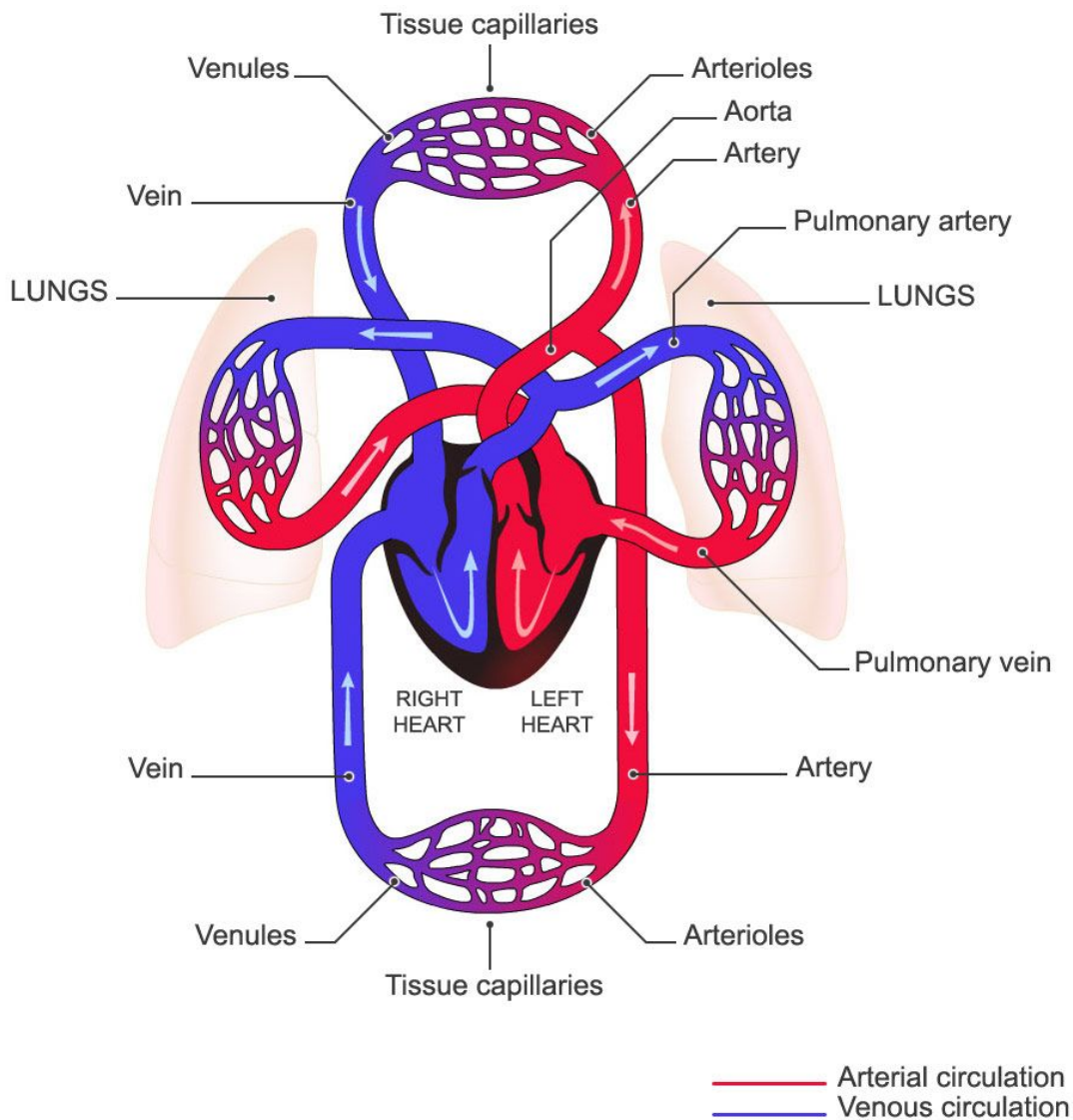
-Ο φλεβόκβος (ή αλλιώς κόμβος των Keith-Flack) όπως ονομάζεται το πρωτεύον κέντρο παραγωγής των διεγέρσεων της καρδιάς, παράγει ένα ηλεκτρικό σήμα. Το ηλεκτρικό αυτό σήμα έχει ως αποτέλεσμα την συστολή των κόλπων κατά την οποία το αίμα εξωθείται διαμέσου των κολποκοιλιακών βαλβίδων προς τις δύο κοιλίες. Αμέσως μετά ενεργοποιείται ο κολποκοιλιακός κόμβος στον οποίο το ηλεκτρικό σήμα καθυστερεί ώστε να ολοκληρωθεί η κολπική συστολή προτού ενεργοποιηθεί η αριστερή κοιλία.

-Το ηλεκτρικό σήμα μεταφέρεται στη συνέχεια ταχέως στις ίνες Purkinje. Οι τελευταίες διαχέουν το δυναμικό ενέργειας στο τοίχωμα των κοιλιών. Η κοιλίες συστέλλονται με επακόλουθη αύξηση της πίεσης στο εσωτερικό τους και το κλείσιμο των κολποκοιλιακών βαλβίδων. Σε αυτό το σημείο η κοιλία μετατρέπεται σε κλειστό θάλαμο, η πίεση στο εσωτερικό του οποίου αυξάνει και η πίεση στην αορτή (στην έξοδο της αριστερής κοιλίας δηλαδή) πέφτει στο ελάχιστο της διαστολικής της τιμής.

-Όταν τελικά η πίεση στην κοιλία υπερβεί την αορτική, ανοίγει η αορτική βαλβίδα και το αίμα εξωθείται προς την αορτή. Η πίεση έτσι εκεί παίρνει τη συστολική (μέγιστη) της τιμή. Μεγάλο μέρος της ενέργειας που μεταφέρεται στο αίμα κατά την καρδιακή συστολή αποθηκεύεται προσωρινά στο ελαστικό τοίχωμα της αορτής και των μεγάλων κλάδων της. Η ενέργεια αυτή εκλύεται κατά την καρδιακή διαστολή συγκρατώντας την αρτηριακή πίεση

υψηλή και την αιματική ροή από την αορτή προς τους αρτηριακούς κλάδους κατά το χρόνο που η καρδιά δεν εξωθεί αίμα.

-Τελικά έχουμε το κλείσιμο της αορτικής βαλβίδας και την εναρξη της ισομετρικής χάλασης. Η πίεση στην κοιλία πέφτει κάτω από εκείνη της αορτής αλλά παραμένει μεγαλύτερη από εκείνη του κόλπου.

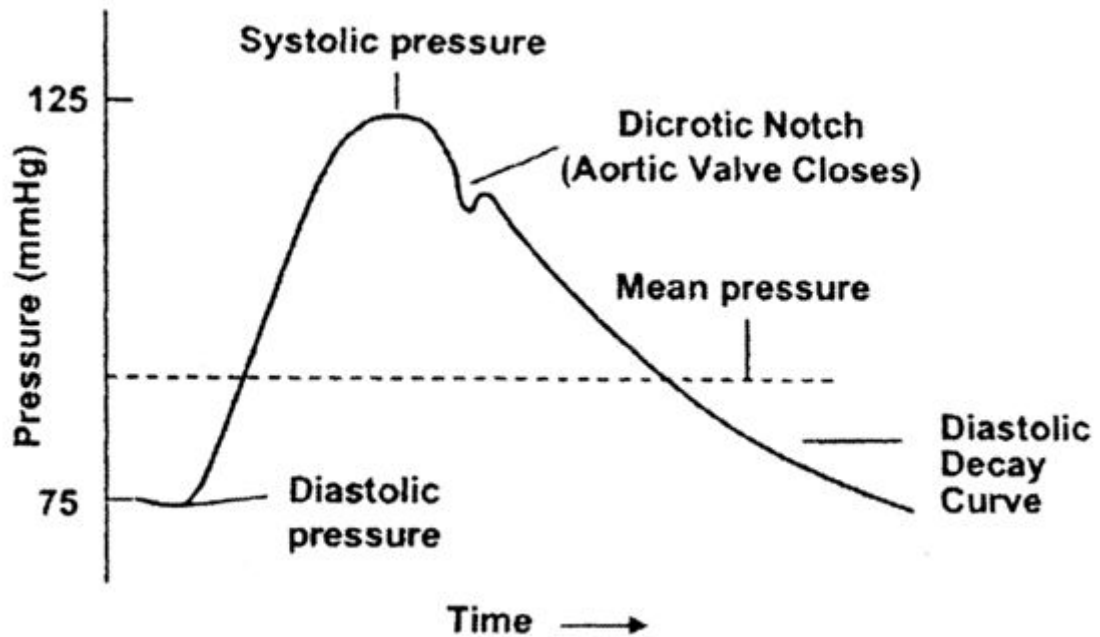


Εικόνα 2.1 : Το κυκλοφορικό σύστημα

Με τον όρο πίεση του αίματος (Blood pressure - BP) συνήθως αναφερόμαστε στην αρτηριακή πίεση στην συστηματική κυκλοφορία και έτσι θα χρησιμοποιηθεί και στην παρούσα εργασία. Αν και η πίεση του αίματος στο φλεβικό σύστημα και στα πνευμονικά αγγεία έχει διαγνωστική αξία, η μέτρηση της γίνεται μόνο επεμβατικά χρησιμοποιώντας καθετήρα. Στην συγκεκριμένη εργασία θα ασχοληθούμε αποκλειστικά με την αρτηριακή πίεση. Κατά σύμβαση, οι μετρήσεις της αιματικής πίεσης στους ανθρώπους δίνονται σε mmHg ( χιλιοστά της στήλης υδραργύρου ) πάνω από την ατμοσφαιρική πίεση. Σε έναν



υγιή άνθρωπο η πίεση αυτή παίρνει τιμές όχι πάνω από 130 mmHg και όχι κάτω από 75 mmHg.



Εικόνα 2.2 : Η μεταβολή της αρτηριακής πίεσης κατά τον καρδιακό κύκλο

Η αιματική πίεση μαζί με την θερμοκρασία σώματος, το καρδιακό ρυθμό (ή παλμό) και τον αναπνευστικό ρυθμό αποτελούν τα 4 επονομαζόμενα και ως ζωτικά σημεία ενός ανθρώπου, στο ιατρικό περιβάλλον. Μαζί, τα τέσσερα αυτά στοιχεία, μπορούν να χρησιμοποιηθούν για να εκτιμηθεί η συνολική σωματική υγεία ενός ατόμου καθώς επίσης και η πρόοδος ενός ασθενή προς την ανάρρωση και τέλος να δώσουν στοιχεία για πιθανές ασθένειες. Εκτός όμως από το νοσοκομειακό περιβάλλον, η μέτρηση της αιματικής πίεσης αποτελεί ένα εργαλείο και για τους ασθενείς με χρόνια προβλήματα του κυκλοφορικού συστήματος (π.χ. ασθενείς με υπέρταση) και όχι μόνο, ώστε να μπορούν εύκολα από την άνεση του σπιτιού τους να παρακολουθούν την υγεία τους και την λειτουργία των φαρμάκων τους.

Υπάρχουν διάφορες μέθοδοι μέτρησης της αρτηριακής πίεσης. Κάθε μία έχει διάφορα πλεονεκτήματα και μειονεκτήματα. Μία μέθοδος που χρησιμοποιείται στην κλινική πρακτική είναι αυτή της επεμβατικής μέτρησης με χρήση καθετήρα. Η συγκεκριμένη μέθοδος επιτρέπει την συνεχή παρακολούθηση της πίεσης σε πραγματικό χρόνο και με εξαιρετική ακρίβεια. Μία επίσης ευρέως χρησιμοποιούμενη τεχνική είναι αυτή του σφυγμομανόμετρου. Δεν επιτρέπει την διαρκή παρακολούθηση της αρτηριακής πίεσης αλλά έχει το πλεονέκτημα να επιτρέπει την αναίμακτη μέτρηση της πίεσης και να μην απαιτεί εξειδικευμένο προσωπικό για την χρήση του απαραίτητου οργάνου.

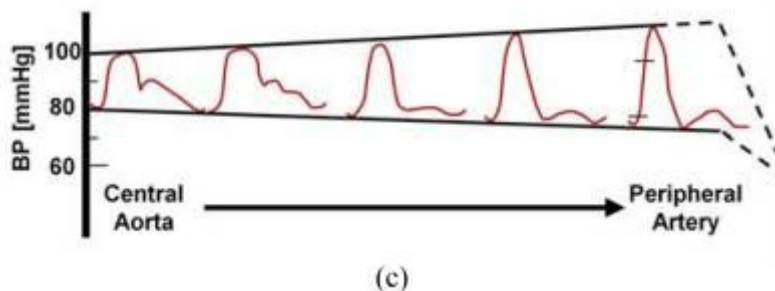
Στην συγκεκριμένη εργασία θα ασχοληθούμε με μια πολλά υποσχόμενη σχετικά νέα τεχνική που συνδυάζει τα δύο πολύ σημαντικά πλεονεκτήματα των προηγούμενων δύο τεχνικών. Πρόκειται για την τεχνική της μέτρησης της ταχύτητας διάδοσης παλμού ή

αλλιώς χρόνου διάδοσης/άφιξης παλμού. Η συγκεκριμένη τεχνική εμφανίστηκε την δεκαετία του 90 και είναι μη επεμβατική, μπορεί να χρησιμοποιηθεί από μη εξειδικευμένο προσωπικό (με χρήση κατάλληλης συσκευής/ εξοπλισμού) και επιτρέπει την διαρκή παρακολούθηση της αρτηριακής πίεσης.

## 2.2 Ο χρόνος διάδοσης παλμού

Για τη συγκεκριμένη τεχνική εκτίμησης της αρτηριακής πίεσης, το άμεσα μετρούμενο μέγεθος είναι ο χρόνος διάδοσης παλμού. Το μέγεθος αυτό επηρεάζεται από διάφορους παράγοντες, όπως θα γίνει αντιληπτό στη συνέχεια, με αποτέλεσμα να έχει χρησιμοποιηθεί ως δείκτης καρδιαγγειακής υγείας (π.χ. μπορεί να χρησιμοποιηθεί ως δείκτης αθηροσκλήρωσης). Στην παρούσα εργασία θα εστιάσουμε στην σχέση του με την αρτηριακή πίεση. Η μέτρησή του θα επιτρέψει τον έμμεσο υπολογισμό- εκτίμηση της αρτηριακής πίεσης.

Πρώτα από όλα, σε ποιο παλμό αναφερόμαστε και πως ορίζεται ο χρόνος διάδοσης του: Όπως ειπώθηκε προηγουμένως, κατά την διάρκεια της συστολής της αριστερής κοιλίας, ενέργεια από την καρδιά μεταφέρεται στο αίμα και κατά ένα μεγάλο μέρος αποθηκεύεται ως ελαστική παραμόρφωση στην αορτή. Πρακτικά αυτό που προκαλεί η καρδιά είναι ένα κύμα πίεσης και παραμόρφωσης (αύξησης της διατομής της αρτηρίας) το οποίο ξεκινά από την αορτική βαλβίδα (ακριβώς στην “έξοδο” της καρδιάς) και προχωράει διαμέσου των αρτηριών προς τα άκρα του σώματος, διαμέσου του συστήματος των αρτηριών με μια σταθερή ταχύτητα  $v$  ( επίσης γνωστή ως ταχύτητα διάδοσης παλμού - Pulse wave velocity - PWV) . Ο “χρόνος διάδοσης παλμού” (Pulse transit time) ορίζεται λοιπόν ως ο χρόνος διάδοσης αυτού του παλμικού σήματος διαμέσου κάποιου μήκους αρτηρίας  $L$ .



Εικόνα 2.3 : Το κύμα πίεσης στο μήκος μια αρτηρίας

Ερευνητές έχουν αναπτύξει διάφορα μοντέλα που ποσοτικοποιούν τα μεγέθη που συναντώνται στο φαινόμενο αυτό: Οι ιδιότητες των αρτηριών (πάχος, ελαστικότητα κτλ.), του αίματος ( πυκνότητα, συμπιεστότητα, πίεση κτλ.) αλλά και του παλμικού σήματος, επηρεάζουν την ταχύτητα διάδοσης του οπότε προφανώς και τον χρόνο του. Ανάλογα με το ποια πληροφορία απαιτείται να εκτιμηθεί με τη χρήση του χρόνου διάδοσης παλμού, το

μοντέλο που αναπτύσσεται πρέπει να εμφανίζει έντονη συσχέτιση με το προς εκτίμηση μέγεθος υπό κάποιες υποθέσεις οι οποίες να είναι ρεαλιστικές και υλοποιήσιμες (δηλαδή να ισχύουν έστω με κάποια προσέγγιση στην πραγματικότητα και να είναι υλοποιήσιμες τουλάχιστον σε κλινικό αλλά βέλτιστα σε οποιοδήποτε περιβάλλον). Για παράδειγμα, όταν το συγκεκριμένο μέγεθος χρησιμοποιείται ως δείκτης αθηροσκλήρωσης, το μοντέλο που θα δημιουργηθεί θα πρέπει να εμφανίζει έντονη συσχέτιση με τις ιδιότητες της αρτηρίας και να περιγράφει τον τρόπο με τον οποίο μπορεί να απαληφεί (ή έστω να μειωθεί αρκετά) η επίδραση της αρτηριακής πίεσης.

Στην συγκεκριμένη περίπτωση όμως απαιτούμε εκτίμηση της αρτηριακής πίεσης χωρίς να επηρεάζουν την εκτίμησή μας τα χαρακτηριστικά των αρτηριών. Αυτό, όπως θα φανεί παρακάτω, μπορεί να επιτευχθεί με την χρήση ενός παραμετρικού μοντέλου το οποίο θα πρέπει να προσαρμοστεί σε κάθε ασθενή ξεχωριστά και για μια συγκεκριμένη διάταξη μέτρησης του χρόνου διάδοσης παλμού.

Τα περισσότερα μοντέλα που χρησιμοποιούνται σήμερα για να συνδέσουν την αρτηριακή πίεση με τον χρόνο διάδοσης παλμού στηρίζονται σε δύο εξισώσεις:

1. εξίσωση Moens–Korteweg:

$$PWV = \sqrt{(Eh_0/2\rho R_0)}$$

PWV : ταχύτητα διάδοσης παλμού

E : δείκτης ελαστικότητας σε δεδομένη πίεση ρευστού στο εσωτερικό του σωλήνα

h0 : πάχος τοιχώματος σωλήνα

R0 : εσωτερική ακτίνα σωλήνα

ρ: πυκνότητα του ρευστού στο εσωτερικό του σωλήνα

Για να καταλήξει κανείς στην συγκεκριμένη εξίσωση θα πρέπει να υποθέσει πως το τοίχωμα του σωλήνα είναι πολύ μικρό σε σχέση με την διάμετρο του σωλήνα, οπότε και το πρώτο μοντελοποιείται πρακτικά σαν μια μεμβράνη και πως το πάχος και η διάμετρος του σωλήνα παραμένουν σταθερά καθώς μεταβάλλεται η πίεση στο ρευστό στο εσωτερικό του σωλήνα.

2. εξίσωση Hughes

$$E = E_0 e^{\zeta P}$$

E0 : δείκτης ελαστικότητας σωλήνα όταν δεν περιέχεται κανένα ρευστό στο εσωτερικό του

ζ : παράγοντας που εξαρτάται από το υλικό κατασκευής της αρτηρίας

P: Πίεση ρευστού στο εσωτερικό του σωλήνα

Η συγκεκριμένη εξίσωση είναι εντελώς εμπειρική. Υπάρχουν εκτεταμένα πειραματικά δεδομένα που την υποστηρίζουν αλλά κανένα θεωρητικό υπόβαθρο.

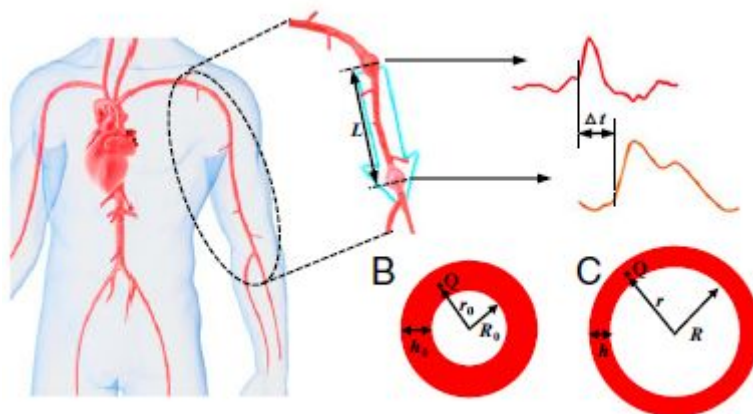
Οι εξισώσεις αυτές, ουσιαστικά περιγράφουν αυτό που θα μπορούσαμε να δεχτούμε ως τον βασικό μηχανισμό ο οποίος συνδέει την πίεση του αίματος με τον χρόνο διάδοσης παλμού: Η μεταβολή της πίεσης του ρευστού επηρεάζει την ελαστικότητα της αρτηρίας (εξίσωση Hughes) η οποία με την σειρά της επηρεάζει την ταχύτητα διάδοσης παλμού (εξίσωση MK) με αποτέλεσμα τον διαφορετικό χρόνο διάδοσης παλμού ( $PWV = L/PTT$ ) Διαφορετικοί χειρισμοί ως προς τις επιπλέον υποθέσεις που απαιτούνται για εφαρμογή των παραπάνω εξισώσεων στο ανθρώπινο κυκλοφορικό σύστημα ή/και χρήση διαφορετικών εμπειρικών ή άλλων εξισώσεων καταλήγει σε διάφορα παραμετρικά μοντέλα συσχέτισης των δύο ποσοτήτων :

- 1) Λογαριθμικά :  $BP = a \ln(PTT) + b$
- 2) Γραμμικά :  $BP = a PTT + b$
- 3) Αντίστροφου τετραγώνου :  $BP = a / PTT^2 + b$
- 4) Αντίστροφου :  $BP = a / PTT + b$

Τέλος, ως προς την μέτρηση του χρόνου διάδοσης παλμού επίσης υπάρχει πληθώρα τεχνικών στην βιβλιογραφία: Το βασικό εργαλείο - κοινός παρονομαστής όλων των τεχνικών είναι η φωτοπληθυσμογραφία (PPG). Πρόκειται για μια αρκετά παλιά τεχνική η οποία βασίζεται στον εντοπισμό της μεταβολής των οπτικών ιδιοτήτων των ιστών όταν ο παλμός περνάει από μέσα τους που ουσιαστικά επιτρέπει την εξαγωγή σε πραγματικό χρόνο και μη επεμβατικά στοιχείων για τον όγκο αίματος στους ιστούς. Η ίδια τεχνική χρησιμοποιείται εκτενώς στους αισθητήρες που υπολογίζουν σχετική οξυγόνωση του αίματος. Περισσότερα για τη συγκεκριμένη τεχνική θα παρουσιαστούν σε επόμενη ενότητα.

Αξίζει να αναφερθούν εδώ δύο τρόποι μέτρησης του χρόνου διάδοσης παλμού:

-Ο πρώτος τρόπος περιλαμβάνει την τοποθέτηση δύο αισθητήρων PPG σε μια δεδομένη απόσταση  $L$  μεταξύ τους και την μέτρηση του χρόνου  $\Delta t$  για την διάδοση του παλμού από το κοντινότερο στην καρδιά σημείο μέτρησης, ως το μακρύτερο.



Εικόνα 2.3 : Ο χρόνος διάδοσης παλμού

- Για τον δεύτερο τρόπο χρησιμοποιούμε έναν αισθητήρα PPG συνήθως τοποθετημένο σε κάποιο άκρο (πόδι/χέρι) και ένα δεύτερο σύστημα/αισθητήρα το οποίο να είναι σε θέση να δίνει σε πραγματικό χρόνο την κατάσταση της καρδιάς και συγκεκριμένα το πότε ολοκληρώθηκε η κοιλιακή συστολή και το παλμικό σήμα παρήχθη από αυτήν. Σε αυτή την περίπτωση μετράμε ουσιαστικά το χρόνο που απαιτήθηκε ώστε το σήμα που παρήχθη από την καρδιά να φτάσει στο άκρο. Εξαιτίας αυτού μερικοί ερευνητές ονομάζουν αυτό το μέγεθος “χρόνο άφιξης παλμού” (Pulse Arrival Time - PAT) για να το διαχωρίσουν από το PTT.

Στις in vivo μελέτες, στην βιβλιογραφία, το δεύτερο μέγεθος (PAT) φαίνεται να εμφανίζει εντονότερη συσχέτιση με την αρτηριακή πίεση, οπότε σε αυτή την εργασία εστιάζουμε στην μέτρηση αυτού. Το τελευταίο που μένει να σχολιαστεί, είναι το σύστημα που αναφέρθηκε προηγουμένως, το οποίο θα παρακολουθεί την καρδιά και θα παράγει τα σημεία αναφοράς για την μέτρηση του χρόνου διάδοσης παλμού. Σε αυτό το σημείο αποκλίνουμε για πρώτη και μοναδική φορά σε αυτή την εργασία από την ευρέως ακολουθούμενη πρακτική: Στην μεγάλη πλειοψηφία των ερευνών, το σύστημα που χρησιμοποιείται για την παρακολούθηση της καρδιακής λειτουργίας είναι το ηλεκτροκαρδιογράφημα (ECG) και συγκεκριμένα, το σημείο R του συμπλέγματος PQRST λαμβάνεται ως το χρονικό σημείο στο οποίο παρήχθη ο παλμός από την καρδιά. Στην εργασία, θα χρησιμοποιήσουμε μια διαφορετική τεχνική, με πολύ μικρότερη παρουσία στην βιβλιογραφία: την τεχνική του φωνοκαρδιογραφήματος (PCG). Θα χρησιμοποιήσουμε δηλαδή τους βασικούς ήχους που παράγει η καρδιά (S1, S2) ως αναφορά για την μέτρηση.

Το φωνοκαρδιογράφημα στην συγκεκριμένη περίπτωση παρουσιάζει αρκετά πλεονεκτήματα έναντι του καρδιογραφήματος:

Ως προς την κατασκευή :

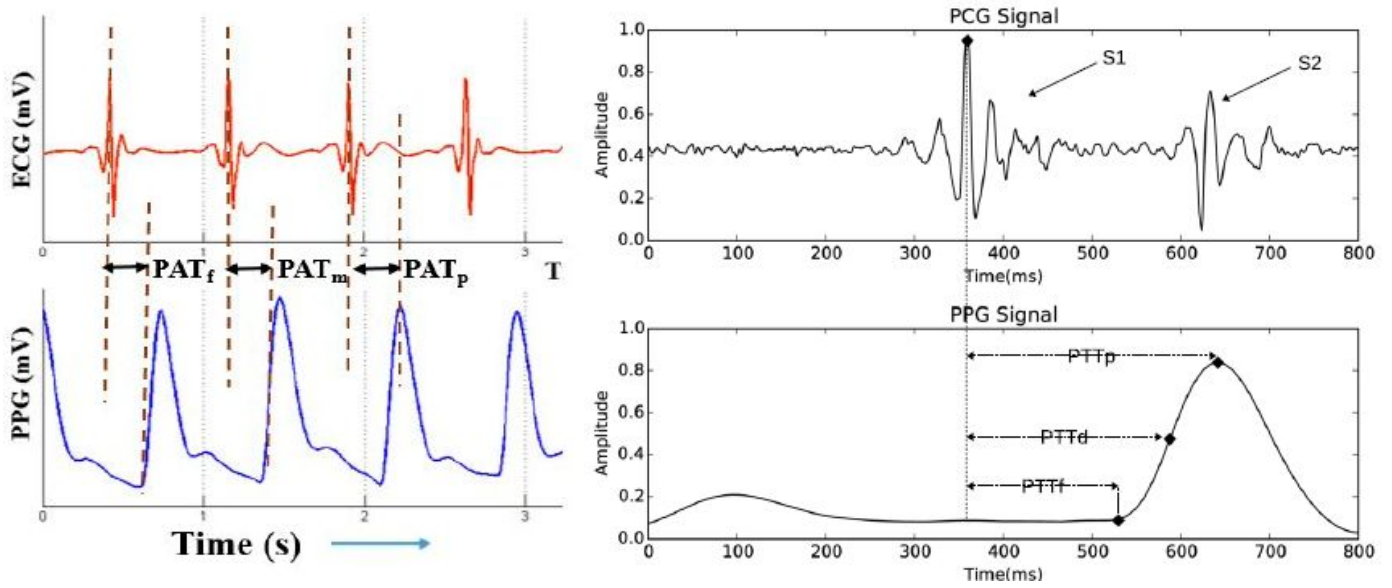
-Απαιτεί απλούστερο και φθηνότερο εξοπλισμό για την υλοποίησή του

-Δεν απαιτεί την ηλεκτρική σύνδεση του ασθενή με κάποια συσκευή με αποτέλεσμα την ασφαλέστερη χρήση του και ευκολότερος ο σχεδιασμός της κατάλληλης συσκευής.

-Η χρήση του είναι πιο άνετη για τον ασθενή αφού δεν απαιτείται η τοποθέτηση ηλεκτροδίων στο δέρμα.

Ως προς τον υπολογισμό του PAT παρουσιάζει επίσης το μεγάλο πλεονέκτημα να αποφεύγει την ηλεκτρομηχανική καθυστέρηση της καρδιάς η οποία “μολύνει” την αντίστοιχη μέτρηση που χρησιμοποιεί το ECG; Από τη στιγμή που το σημείο R καταγραφεί από το ECG υπάρχει μια καθυστέρηση που ονομάζεται Pre-Ejection Period (PEP) προτού η καρδιά καταφέρει να μετατρέψει το ηλεκτρικό σήμα σε μηχανική κίνηση (πίεση/εκτόξευση) του αίματος. Οπότε αυτό που μετράει η συγκεκριμένη διάταξη είναι το

$$PAT^* = PAT + PEP$$



Εικόνα 2.4 : Ο χρόνος διάδοσης παλμού, με χρήση ECG(αριστερά) και με χρήση PCG(δεξιά)

Ο συγκεκριμένος χρόνος (PAT\*) δεν θα δημιουργούσε ιδιαίτερο πρόβλημα στα περισσότερα μοντέλα αν ήταν σταθερός (π.χ θα μπορούσε να απορροφηθεί από την παράμετρο (β) του γραμμικού μοντέλου), αλλά στην πραγματικότητα είναι απρόβλεπτος: εξαρτάται συν των αργά μεταβαλλόμενων παραγόντων όπως η ηλικία, και από παράγοντες όπως το στρες και τη συναισθηματική κατάσταση του ασθενή, κάνοντάς τον σημαντική πηγή λάθους για τα συστήματα που χρησιμοποιούν ECG.

Algorithm	Time Delay	Calibration Technique	Results		
			SBP	DBP	MAP
$\ln(\text{Time Delay}) + b$	$PTT_f$	Subject specific	$-0.22 \pm 0.46^2$	-	-
	$PAT_f$		$-0.85 \pm 0.09^2$	-	-
	$PTT_f$			$0.9^3$	-
$\frac{A}{PAT} + B$	$PAT_p$	A: Dependent on Height B: Subject specific	$0.0790 \pm 11.32^1$	-	-
	$PAT_f$	Subject specific	$0.99-0.90^2$	-	-
$aPAT + b$	$PAT_p$	Subject specific	$-0.92^2$	$-0.38^2$	-
	$PAT_p$	Subject specific	RT: $-0.87^{2,+}$	RT: $-0.30^{2,+}$	-
	$PAT_m$	Subject specific	$0.701^3$	$0.401^3$	-
	$PAT_p$	Subject specific	$0.95-0.87^3$	$0.01-0.73^3$	-
	$PAT_p$	Subject specific	$-0.71^2$	$-0.69^2$	-
	$PAT_m$		$-0.32^2$	$-0.22^2$	-
	$PAT_f$		$-0.09^2$	$-0.02^2$	-
$PAT_p$	Subject specific			$0.32^2$	
$PAT_m$				$0.27^2$	
$PAT_f$				$0.45^2$	
$\frac{a}{PAT} + b$	$PAT_p$	Subject specific	$0.95^3$	$0.26^3$	-
	$PAT_m$	Subject specific	$0.89^3$	$0.78^3$	-
$\frac{a}{PAT} + b + c \cdot VPAT + d \cdot (PATV - PATV_0)$	$PAT_p$	Subject specific	$0.96^3$	$0.70^3$	-
$a + \left(\frac{b}{PAT-c}\right)^2$	$PAT_p$	Subject specific	$0.97 \pm 0.87^3$	$0.54 \pm 0.05^3$	-
	$PAT_f$	Subject specific	$0.93^2$	-	$0.83^2$
$PAT_f$	$0.94^2$		-	$0.86^2$	
$aPAT + bHR + c$	$PAT_m$	Subject specific, Maximum Likelihood	$0.978^3$	$0.974^3$	-
	$PAT_m$	Subject specific, Adaptive Kalman filter	$0.976^3$	$0.989^3$	-
$a + bPAT + cHR + dTDB$	$PAT_m$	Subject specific	$0.85^3$	$0.74^3$	-

$BP_{ij} = b_{ij}e^{-\frac{K_{ij}}{PWV_{ij}}}$ $i = 1, 2, \dots, m(\text{age})$ and $j = \text{Male/Female}$	$PTT_{ij}$ $PTT_{ip} + PTT_{fp}$	$b_{ij}$ and $K_{ij}$ are calculated for a demographically similar group.	- $2.16 \pm 6.23^1$	$-1.49 \pm 6.51^1$ -	-
$DBP = \frac{SBP_0}{3} + \frac{2 \cdot DBP_0}{3} + \ln\left(\frac{PAT_{w0}}{PAT_{w0}}\right) - \frac{(SBP_0 - DBP_0)}{3} \left(\frac{PAT_{w0}}{PAT_{w0}}\right)^2$ $SBP = DBP_0 + (SBP_0 - DBP_0) \left(\frac{PAT_{w0}}{PAT_{w0}}\right)^2$	$PAT_m$	Subject specific	$0.6 \pm 9.8^1$	$0.9 \pm 5.6^1$	-
$BP = a * PWV * e^{b * PMV} + cPWV^d - (BP_{PTT,cal} - BP_{PTT})$	$PAT_m$	Universal	$0.83^2$	-	-
$DBP = DBP_0 \frac{PIR_0}{PIR}$ $SBP = DBP_0 \frac{PIR_0}{PIR} + PP_0 \left(\frac{PAT_0}{PAT}\right)^2$	$PAT_m$	Subject specific	$0.91^2$	$0.88^2$	$0.89^2$

Εικόνα 2.5 : Μοντέλα BP(PTT) και BP(PAT)

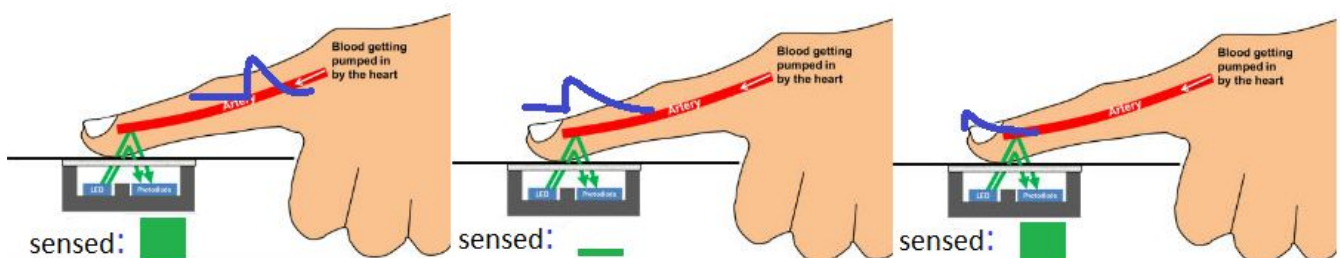
### 3. Η Φορητή Συσκευή

Σύμφωνα με όσα προηγήθηκαν, ο στόχος που τέθηκε ήταν να κατασκευαστεί μια συσκευή η οποία να συνδυάζει ένα σύστημα PPG με ένα σύστημα PCG, και να περιέχει την κατάλληλη υποδομή, ώστε να μπορεί να χρησιμοποιηθεί στην έρευνα (να διευκολύνει δηλαδή μια ερευνητική προσπάθεια για την μελέτη του PTT και την σχέση του με την αρτηριακή πίεση) καθώς επίσης να έχει την δυνατότητα να αναπτυχθεί ως φορητή συσκευή για χρήση από ασθενείς στο σπίτι με την ευκολία που παρέχει ένα συνηθισμένο εμπορικό πιεσόμετρο. Για την επίτευξη των παραπάνω στόχων, με χαμηλό κόστος, στη συνέχεια του κεφαλαίου προτείνεται μια σχεδίαση γύρω από έναν μικροελεγκτή STM32F4 και παρουσιάζεται η ανάπτυξη των απαραίτητων κυκλωμάτων και περιφερειακών.

#### 3.1 Οι αισθητήρες

Όπως έχει γίνει φανερό από προηγούμενο κεφάλαιο, για την μέτρηση του PAT με την τεχνική που επιλέξαμε, απαιτούνται δύο αισθητήρια στοιχεία. Ένα για το PPG το οποίο στηρίζεται στο φως και ένα για το PCG το οποίο στηρίζεται στον ήχο.

Για τη λήψη του PPG σήματος απαιτούνται δύο στοιχεία: Μια πηγή φωτός και ένα φωτοαισθητήριο στοιχείο. Το παραπάνω ζεύγος στοιχείων μπορεί να τοποθετηθεί σε οποιοδήποτε σημείο του σώματος με το δάχτυλο του ενός χεριού να είναι ένα βολικό σημείο. Το φως που εκπέμπεται από το ένα, κατά ένα μέρος ανακλάται και το υπόλοιπο διαχέεται στον ιστό και το φωτοευαίσθητο στοιχείο λαμβάνει ένα μέρος του φωτός. Όταν φτάνει ο παλμός που αναζητούμε στο σημείο στο οποίο είναι τοποθετημένος ο αισθητήρας, έχουμε μια στιγμιαία αύξηση του όγκου του αίματος η οποία έχει ως αποτέλεσμα την αύξηση της απορρόφησης του φωτός από τον ιστό (εφόσον χρησιμοποιούμε συχνότητες φωτός στις οποίες απορροφά το αίμα) . Ο αισθητήρας φωτός σε αυτή την περίπτωση θα μετρήσει λιγότερο φως (τοπικό ελάχιστο) .

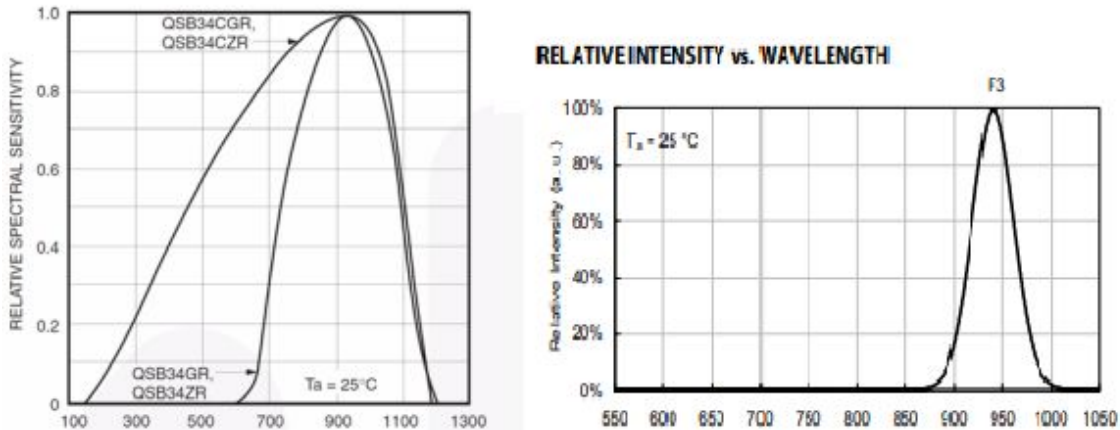


Εικόνα 3.1 : Η καταγραφή του φωτοπληθυσμογραφικού σήματος από το δάχτυλο



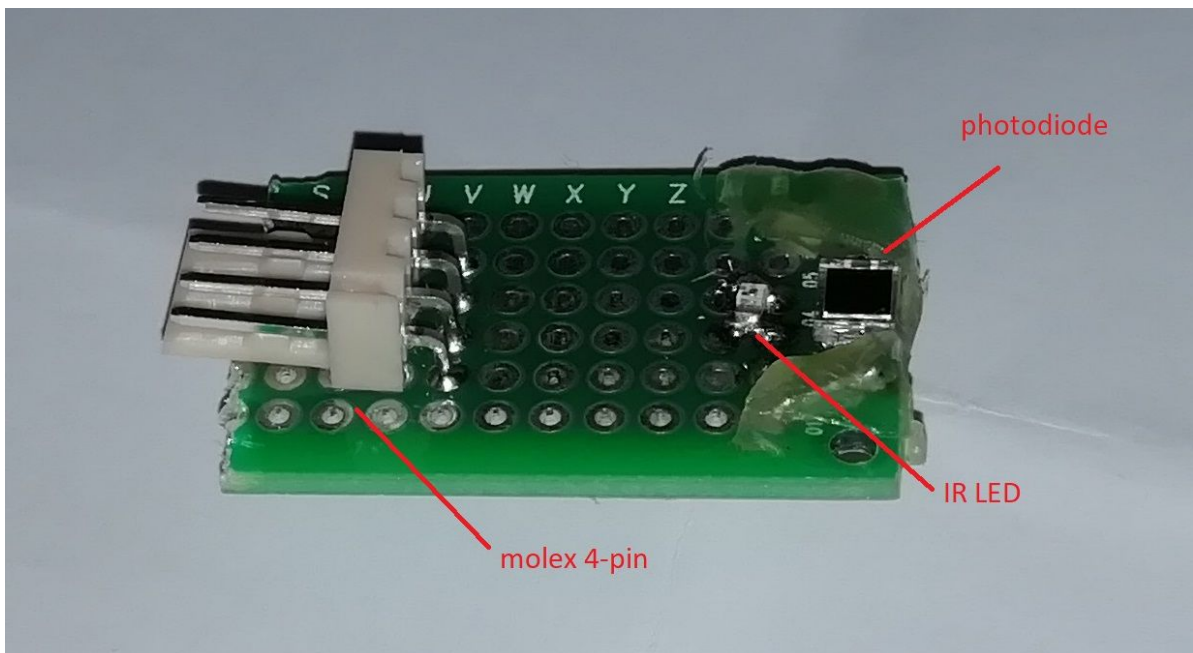
Συνήθως στις συσκευές οξυμετρίας που χρησιμοποιούν τέτοιους αισθητήρες (με στόχο την εκτίμηση της σχετικής οξυγόνωσης του αίματος) χρησιμοποιούν φως μήκους κύματος 940 nm (υπέρυθρο) και 625 nm (κόκκινο). Οποιαδήποτε από τις δύο συχνότητες θα ήταν κατάλληλη για την εφαρμογή μας, μιας και οι δύο συχνότητες απορροφώνται από το αίμα. Επιλέχθηκε η συχνότητα των 940nm λοιπόν για το λεντ που θα λειτουργήσει ως πηγή φωτός.

Ως φωτοαισθητήριο στοιχείο επιλέχθηκε μια φωτοδίοδος με κορυφή στην ίδια συχνότητα.



Εικόνα 3.2 : Η απόκριση ανά φωτεινή συχνότητα της φωτοδίοδου QSB34GR (αριστερα) και το φάσμα που παράγει το υπέρυθρο λεντ(δεξιά) που χρησιμοποιήθηκαν.

Τα δύο στοιχεία τοποθετήθηκαν σε μικρή πλακέτα βακελίτη (HPL) και τοποθετήθηκε Molex αρσενικό βύσμα 4 απολήξεων για την σύνδεση καλωδίου στο αισθητήριο ζεύγος που θα το συνδέει με την συσκευή που κατασκευάστηκε.



Εικόνα 3.3 : Ο αισθητήρας PPG που κατασκευάστηκε

Για τον αισθητήρα για την καταγραφή του PCG υπάρχουν διάφορες επιλογές. Η πιο απλή και λιγότερο αποτελεσματική θα ήταν η χρήση κοινού δυναμικού μικροφώνου τοποθετημένου πάνω στο θώρακα του ασθενή. Ο συγκεκριμένος τρόπος λήψης του ήχου της καρδιάς γενικά δεν χρησιμοποιείται, καθώς ήχοι του περιβάλλοντος πολύ εύκολα μολύνουν το ληφθέν σήμα. Ένας τρόπος λήψης που χρησιμοποιείται στην πράξη (π.χ. σε ψηφιακά στηθοσκόπια) στηρίζεται στη χρήση πιεζοηλεκτρικού στοιχείου συζευγμένου μηχανικά με κατάλληλο τρόπο με ένα διάφραγμα που ακουμπάει στο στήθος. Αυτό έχει ως αποτέλεσμα ο ήχος της καρδιάς να περνάει από τον ιστό, προς το διάφραγμα και τέλος να φτάνει στο πιεζοηλεκτρικό στοιχείο, το οποίο τελικά τον μετατρέπει σε ηλεκτρικό σήμα.

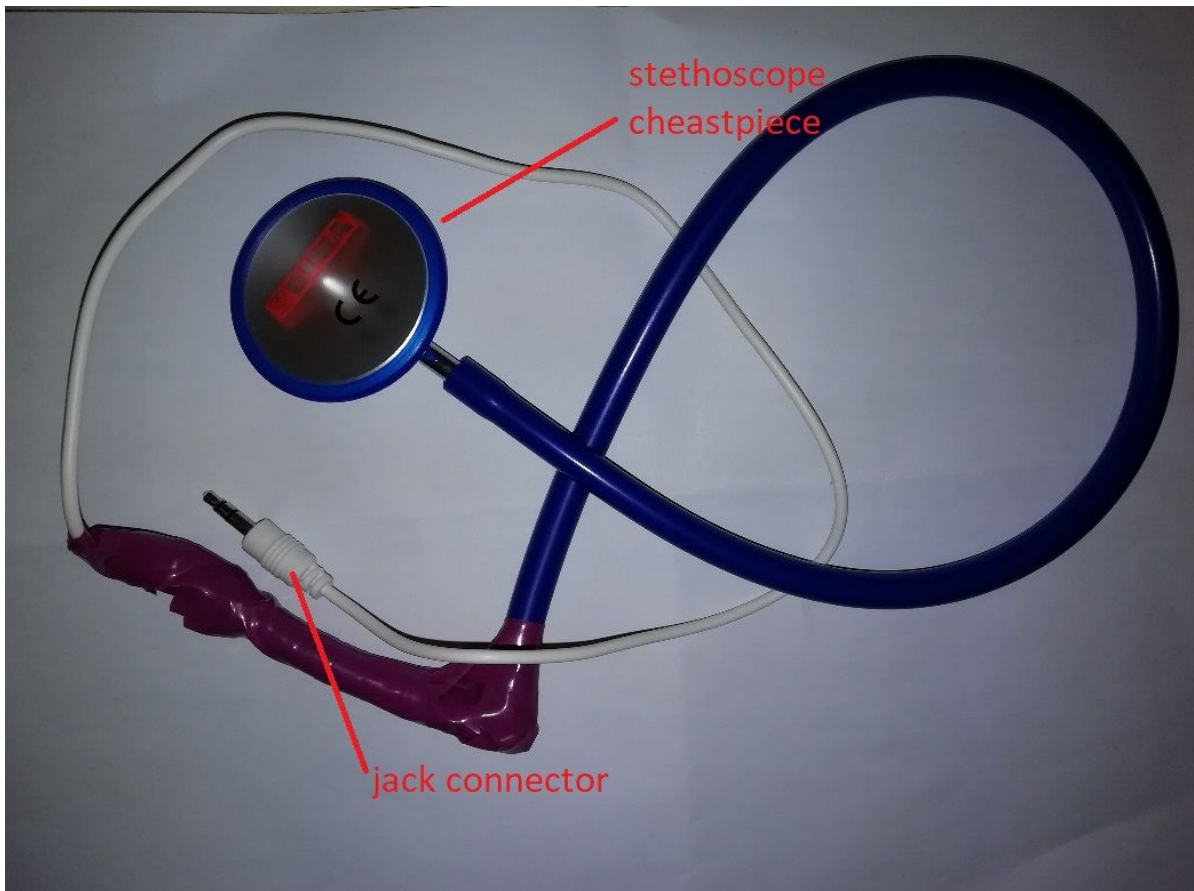
Στη συγκεκριμένη εργασία δεν χρησιμοποιούμε αυτές τις τεχνικές. Η πρώτη είναι εντελώς αναποτελεσματική και η δεύτερη θα απαιτούσε την αγορά του αντιστοιχού αισθητήριου συστήματος ( το κόστος του είναι αρκετά υψηλό καθώς πρόκειται για εξειδικευμένο όργανο ) ή την κατασκευή του η οποία είναι αρκετά πολύπλοκη. Αντίθετα, χρησιμοποιούμε ένα φθινό και αξιόπιστο τρόπο λήψης ο οποίος απαντάται στην βιβλιογραφία και περιγράφεται και σε ένα application report της Texas Instruments σχετικά με την κατασκευή ενός ψηφιακού στηθοσκοπίου. Το σύστημα για το οποίο γίνεται λόγος, αποτελείται από ένα σωλήνα που στην μία άκρη του τοποθετείται λεπτό διάφραγμα και στην άλλη ένα πυκνωτικό μικρόφωνο με φορά προς το εσωτερικό του σωλήνα. Η συγκεκριμένη διάταξη επιτρέπει ουσιαστικά την εύκολη εστίαση του μικροφώνου στο σημείο ενδιαφέροντος (θώρακας) ακουμπώντας τη διάταξη από την πλευρά του διαφράγματος, πάνω σε αυτό.



*Εικόνα 3.4 : Η αρχιτεκτονική του αισθητήρα PCG που κατασκευάστηκε*

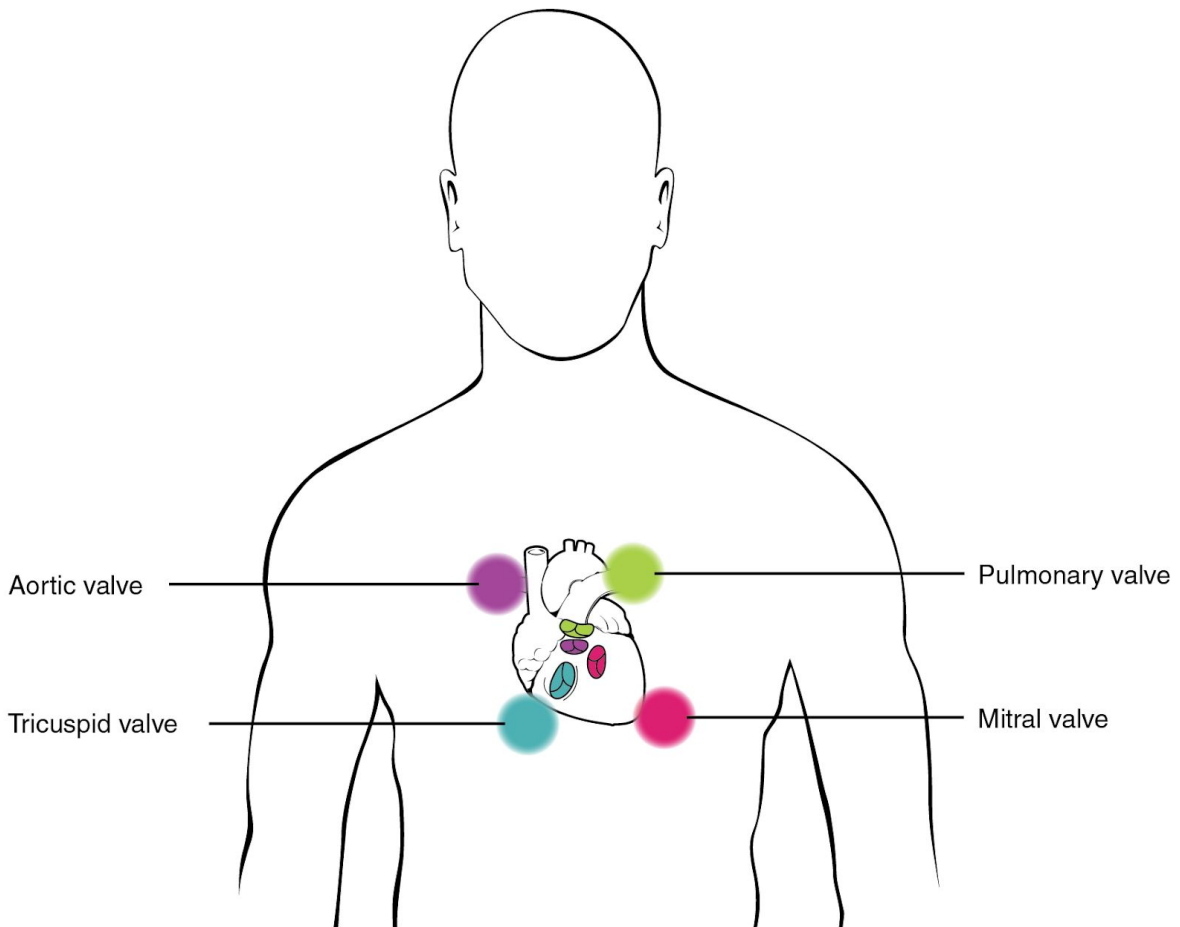
Σύμφωνα με αυτή την αρχιτεκτονική κατασκευάστηκαν μερικά συστήματα μικροφώνου-σωλήνα-διαφράγματος αλλά τελικά τα καλύτερα αποτελέσματα επιτεύχθηκαν με τη χρήση του μπροστινού τμήματος ενός στηθοσκοπίου χαμηλού κόστους στο οποίο προσαρτήθηκε κατάλληλα ένα πυκνωτικό μικρόφωνο ηλεκτρίτη. Συγκεκριμένα χρησιμοποιήθηκε ένα μικρόφωνο με διάμετρο κεφαλής ίση με τη διάμετρο του ελαστικού σωλήνα του στηθοσκοπίου ώστε να μπορεί να εφαρμόσει μέσα σε αυτό.

Τέλος ένα καλώδιο Audio Jack 2.5mm κολήθηκε στο μικρόφωνο για την μετέπειτα σύνδεση του αισθητήρα με την συσκευή που κατασκευάστηκε.



*Εικόνα 3.5 : Ο αισθητήρας PCG που κατασκευάστηκε*

Τέλος, όσον αφορά την τοποθέτηση του αισθητήρα στο στήθος, μπορούμε να βρούμε εύκολα πληροφορίες στη βιβλιογραφία και να ακολουθήσουμε οδηγίες που απευθύνονται σε γιατρούς/νοσηλευτές που αναφέρονται στη χρήση στηθοσκοπίου συγκεκριμένα για τους ήχους της καρδιάς όπως φαίνεται π.χ. στην παρακάτω εικόνα. Αν και ο αισθητήρας δοκιμάστηκε σε διάφορα σημεία στο στήθος και παντού έδινε αξιοπρεπή αποτελέσματα (σημαντική ιδιότητα αν θέλουμε η συσκευή να μπορεί να χρησιμοποιηθεί από μη εξειδικευμένο προσωπικό), στο σημείο πάνω από την αορτική βαλβίδα φαίνεται πραγματικά να έχουμε την μεγαλύτερη ένταση για τον ήχο S1, αν εστιάσουμε σε αυτόν (περισσότερα για τους ήχους της καρδιάς στο κεφάλαιο για την χρήση της συσκευής).



Εικόνα 3.6 : Η τοποθέτηση του αισθητήρα PCG ( Βέλτιστα πάνω από την αορτική βαλβίδα)

### 3.2 Αναλογικό κύκλωμα

Βασικό τμήμα της σχεδίασης της συσκευής, είναι τα αναλογικά κυκλώματα υπεύθυνα για την λήψη και προ-επεξεργασία των σημάτων από τους δύο αισθητήρες.

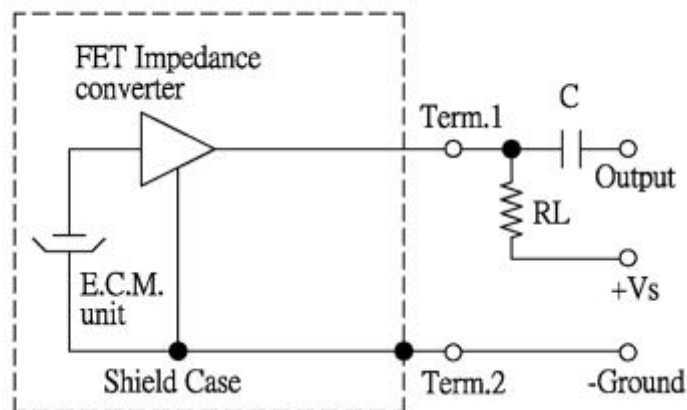
Πριν προχωρήσουμε σε αυτό το κεφάλαιο, κρίνεται απαραίτητο, στο σημείο αυτό να ορίσουμε κάποιες προδιαγραφές της συσκευής οι οποίες τέθηκαν και επηρεάζουν την σχεδίαση των προαναφερθέντων κυκλωμάτων:

-Η τάση των 3.3V (μονή τροφοδοσία), ορίστηκε ως η τάση που θα χρησιμοποιεί η συσκευή συνολικά, για τα αναλογικά και ψηφιακά της ηλεκτρονικά. Η επιλογή αυτή έγινε με γνώμονα την χαμηλή κατανάλωση της συσκευής και την αποφυγή της χρήσης boost converter για τροφοδοσία της είτε από ένα στοιχείο Li-Ion ( 3.7 Volt) είτε από το Vbus μιας σύνδεσης USB (5V +/-10%).

-Με την παραπάνω επιλογή, για να έχουμε όσο το δυνατόν μεγαλύτερο δυναμικό εύρος για τα σήματα που πρόκειται να χειριστούμε στο αναλογικό στάδιο, θα επιλέξουμε τη χρήση Rail-to-Rail input-output τελεστικών ενισχυτών. Οι συγκεκριμένοι ενισχυτές , μπορούν να λειτουργήσουν με είσοδο και έξοδο που είναι δυνατόν να φτάνει στα άκρα της τροφοδοσίας τους (στην περίπτωσή μας 0-3V3)

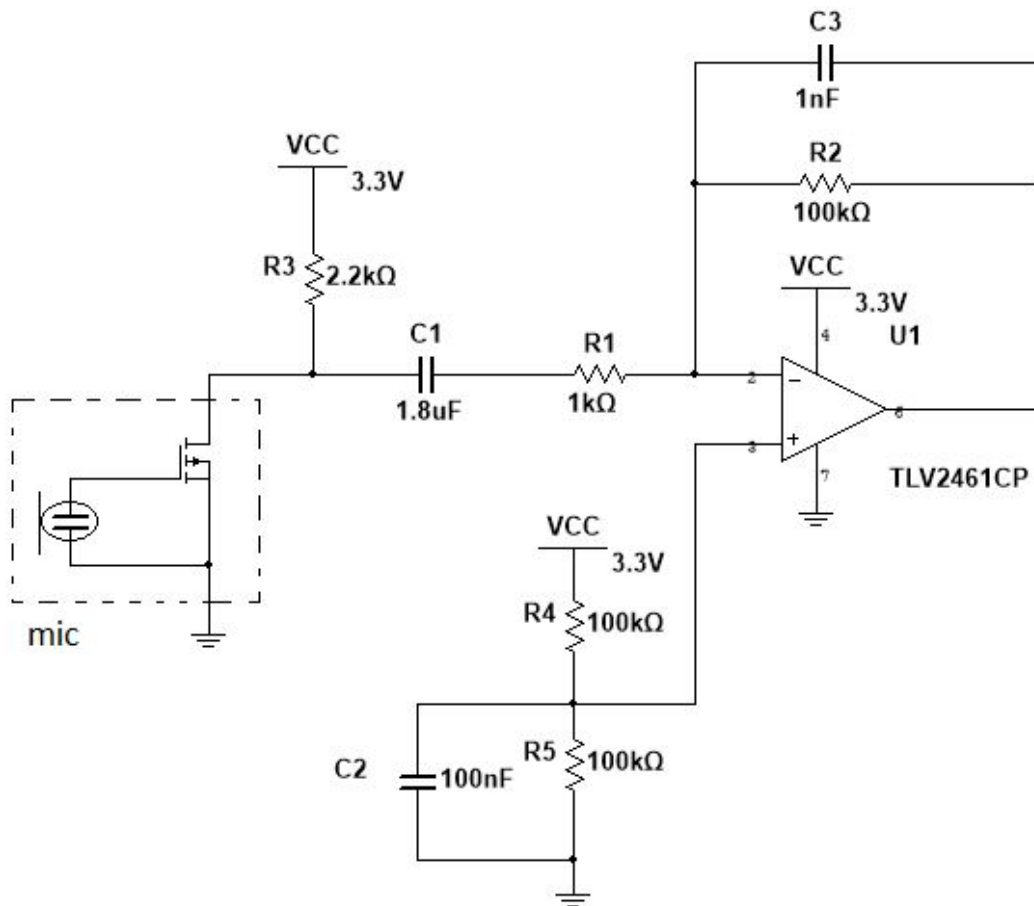
-Τα σήματα στην έξοδο των αναλογικών front-end που θα περιγράψουμε στη συνέχεια, θα τα υποδεχτεί το 12bit Analog-to-Digital-Converter (ADC) περιφερειακό ενός μικροελεγκτή STM32F401 που επίσης θα λειτουργεί στην τάση των 3.3V. Για να αξιοποιηθεί το δυναμικό εύρος του, τα (AC) σήματα αυτά, θα πρέπει να είναι γύρω από την τάση των 1.65V (3.3/2)

Αρχικά παρουσιάζουμε το κύκλωμα για τη λήψη του PCG σήματος: Σύμφωνα με το Datasheet του πυκνωτικού μικροφώνου που επιλέχθηκε ( CME-1538-100LB από την CUI) απαιτείται αρχικά ένα κύκλωμα πόλωσης για την ορθή λειτουργία του. Ένα μικρόφωνο ηλεκτρίτη κανονικά δεν απαιτεί εξωτερική πόλωση για τη λειτουργία του, αλλά το συγκεκριμένο μικρόφωνο (όπως και η μεγάλη πλειοψηφία των μικροφώνων στο εμπόριο αυτού του είδους) ενσωματώνει εκτός από το μικρόφωνο και ένα ενισχυτικό στάδιο στην ίδια συσκευασία, ο σκοπός του οποίου είναι η μείωση της αντίστασης εξόδου του αισθητήρα. Η ύπαρξη αυτού του ενισχυτή υποχρεώνει την εξωτερική πόλωση του.



Εικόνα 3.7 : Το ισodύναμο κύκλωμα του μικροφώνου που χρησιμοποιήθηκε, μαζί με τα απαραίτητα για την λειτουργία του, στοιχεία.

Για το σήμα που αναμένουμε, γνωρίζουμε πως δεν διαθέτει χρήσιμη πληροφορία πέρα των 300Hz και κάτω των 100Hz και σύμφωνα με τις προδιαγραφές που αναφέρθηκαν προηγουμένως η έξοδος του κυκλώματος μας θα πρέπει να είναι γύρω από την τάση των 1.65V. Τέλος, το μέγεθος της ενίσχυσης που απαιτείται ώστε να χρησιμοποιήσουμε όσο το δυνατόν μεγαλύτερο δυναμικό εύρος του ADC, αλλά χωρίς το σήμα να φτάνει σε καμία περίπτωση στα άκρα της τροφοδοσίας (κορεσμός σήματος), λόγω της κατασκευής του αισθητήρα, δεν μπορεί να καθοριστεί αναλυτικά, αλλά πρέπει αφού σχεδιαστεί το κύκλωμα να βρεθεί πειραματικά. Για το αναλογικό front-end που θα χειριστεί το σήμα επιλέχθηκε ένας αναστρέφον ενισχυτής, ώστε να εκπληρώσει σε ένα στάδιο τις παραπάνω απαιτήσεις. Για την δημιουργία της τάσης των 1.65V χρησιμοποιείται ένας απλός διαιρέτης τάσης. Το συνολικό κύκλωμα παρουσιάζεται παρακάτω.



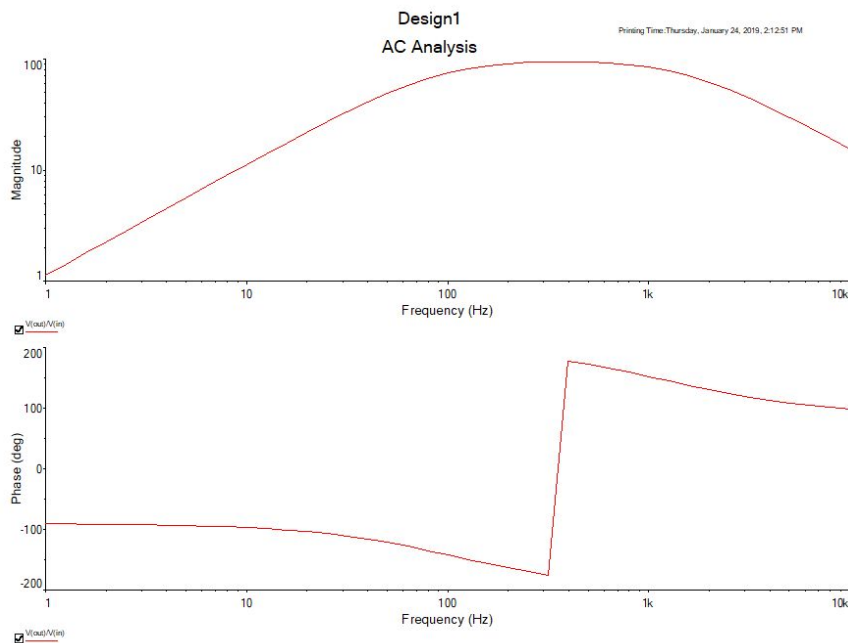
Εικόνα 3.8 : Το analog front-end για τον αισθητήρα PCG

Χαρακτηριστικά κυκλώματος:

Ζωνοπερατό 88Hz με 1.5kHz και ενίσχυση 100 V/V στην ζώνη πέρασης.

Η συχνότητα των 88Hz είναι πάνω από την πρώτη αρμονική (50Hz) του δικτύου ενέργειας (σημαντική πηγή θορύβου). Η απόκριση του φίλτρου παρουσιάζεται στην επόμενη εικόνα.

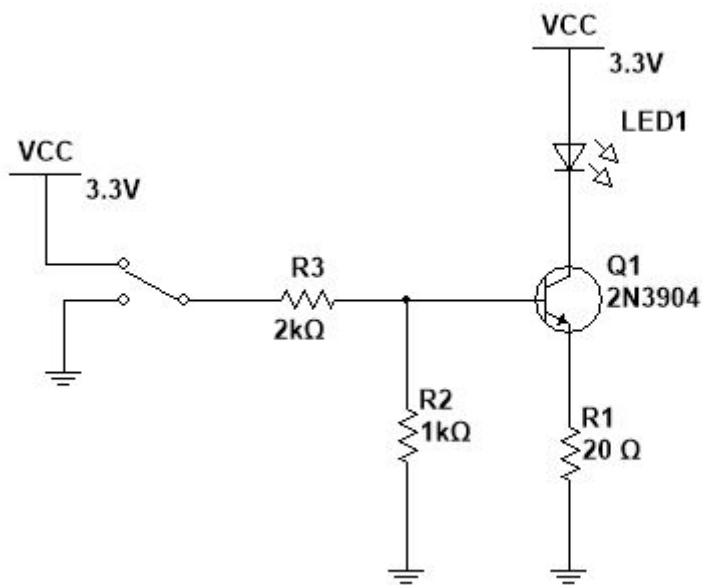
Όλες οι προσομοιώσεις που θα παρουσιαστούν έγιναν στο πρόγραμμα Multisim 12.0



Εικόνα 3.9: Η απόκριση του analog front-end του PCG αισθητήρα

Συνεχίζουμε τώρα με τη σχεδίαση του κυκλώματος για την λήψη του PPG σήματος. Πρακτικά πρόκειται για δύο κυκλώματα. Ένα για την τροφοδοσία του υπέρυθρου λεντ και ένα για την λήψη του σήματος από την φωτοδίοδο.

Για την τροφοδοσία του λεντ επιλέχθηκε ένα κύκλωμα βασισμένο σε ένα συνηθισμένο διπολικό τρανζίστορ (BJT), το 2N3904. Το κύκλωμα σχεδιάστηκε ειδικά ώστε το τρανζίστορ να παραμένει στην ενεργό περιοχή λειτουργίας, ώστε να λειτουργεί συνολικά ως πηγή ρεύματος για το συνδεδεμένο σε αυτό υπέρυθρο λεντ. Η αντίσταση R1 που φαίνεται στο παρακάτω σχήμα ελέγχει το μέγεθος αυτού του ρεύματος. Για είσοδο 3.3V το κύκλωμα λειτουργεί και το λεντ ανάβει, για είσοδο 0V, το λεντ παραμένει σβηστό. Με είσοδο PWM (0-3.3V και duty Cycle 0-100%) μπορούμε εύκολα να ελέγξουμε την ενεργό τιμή του ρεύματος που διαρρέει το λεντ, ελέγχοντας έτσι την φωτεινότητά του ψηφιακά, ενώ αυτό παραμένει σε ιδανική περιοχή λειτουργίας, ως προς το μέγιστο ρεύμα που το διαρρέει. Η χρησιμότητα αυτής της δυνατότητας του κυκλώματος θα φανεί στη συνέχεια.

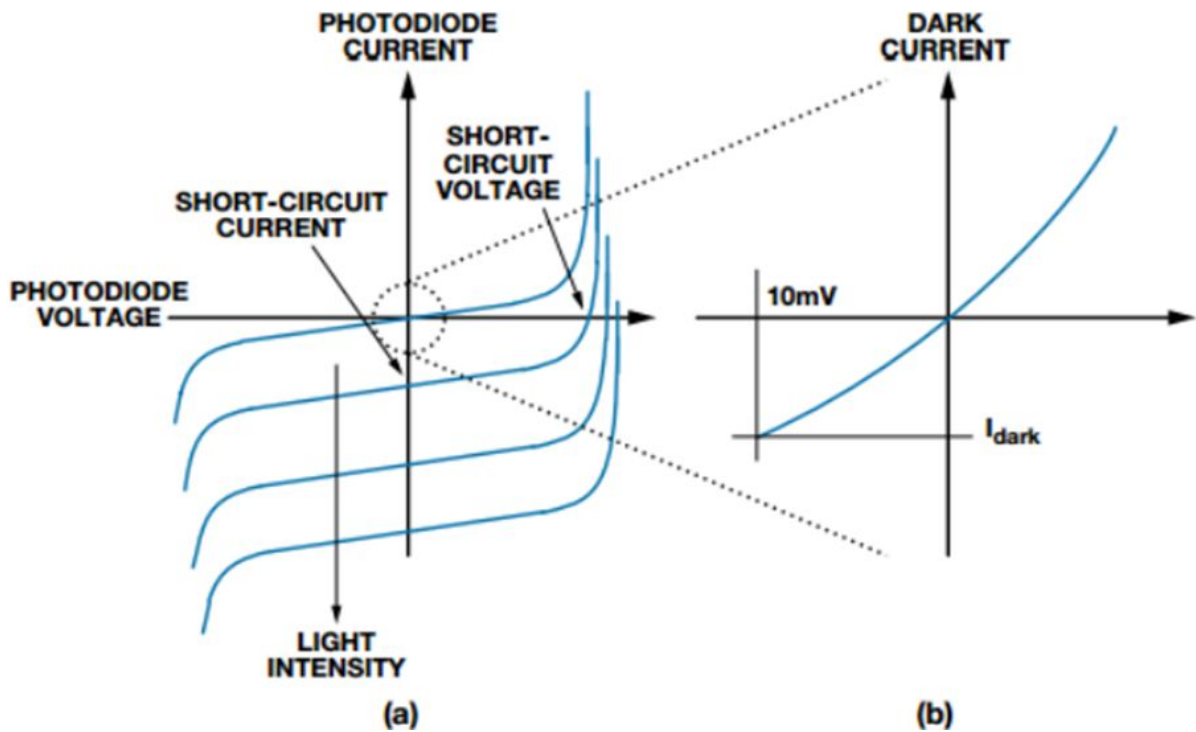


Εικόνα 3.10 : Το κύκλωμα τροφοδοσίας του υπέρυθρου λεντ του PPG αισθητήρα

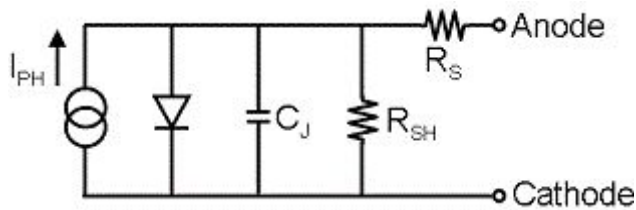
Για το ρεύμα  $I$  που διαρρέει το λεντ του σχήματος θα ισχύει (με καλή προσέγγιση)  
 $I = (V_b - V_{be(active)}) / R_1 = 0.45 / R_1$

Για την λήψη του σήματος από την φωτοδίοδο σχεδιάσαμε το απαραίτητο κύκλωμα ως εξής: Πρώτα από όλα όσον αφορά τον αισθητήρα παρατηρούμε από τις παρακάτω γραφικές παραστάσεις που περιγράφουν την λειτουργία της, αφενός πως η δίοδος για κάποιες τιμές τάσης λειτουργεί ως πηγή ρεύματος ανάλογη με την ποσότητα του φωτός που προσπίπτει σε αυτήν και αφετέρου, πως για οποιαδήποτε μη μηδενική τάση στα άκρα της δίοδου κατά την λειτουργία της έχουμε τη δημιουργία του λεγόμενου “σκοτεινού ρεύματος” (dark current), η οποία “μολύνει” οποιαδήποτε μέτρηση. Συνήθως στις εφαρμογές που απαιτούν μεγάλο εύρος ζώνης (>kHz), η φωτοδίοδος πολώνεται ανάστροφα κατά τη λειτουργία της, ώστε να μικρύνει η παρασιτική της χωρητικότητα και να γίνει με αυτό τον τρόπο, “γρηγορότερη”. Στην δική μας εφαρμογή όμως, οι συχνότητες με τις οποίες ασχολούμαστε είναι χαμηλές (<15Hz), οπότε και θα διατηρήσουμε την τάση στα άκρα της φωτοδίοδου κοντά στο μηδέν και θα μετρήσουμε το παραγόμενο ρεύμα με έναν ενισχυτή διαγωγιμότητας βασισμένο σε έναν τελεστικό ενισχυτή.



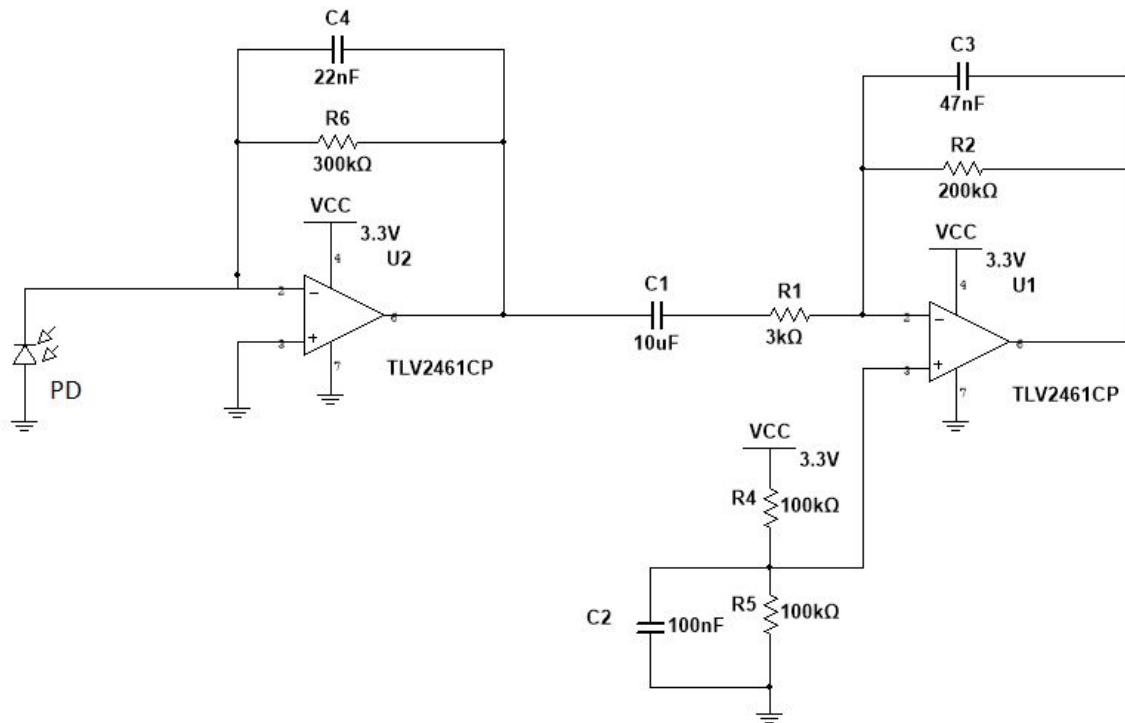


Εικόνα 3.11 : Η χαρακτηριστική καμπύλη της φωτοδιόδου



Εικόνα 3.12 : Το ισοδύναμο κύκλωμα της φωτοδιόδου

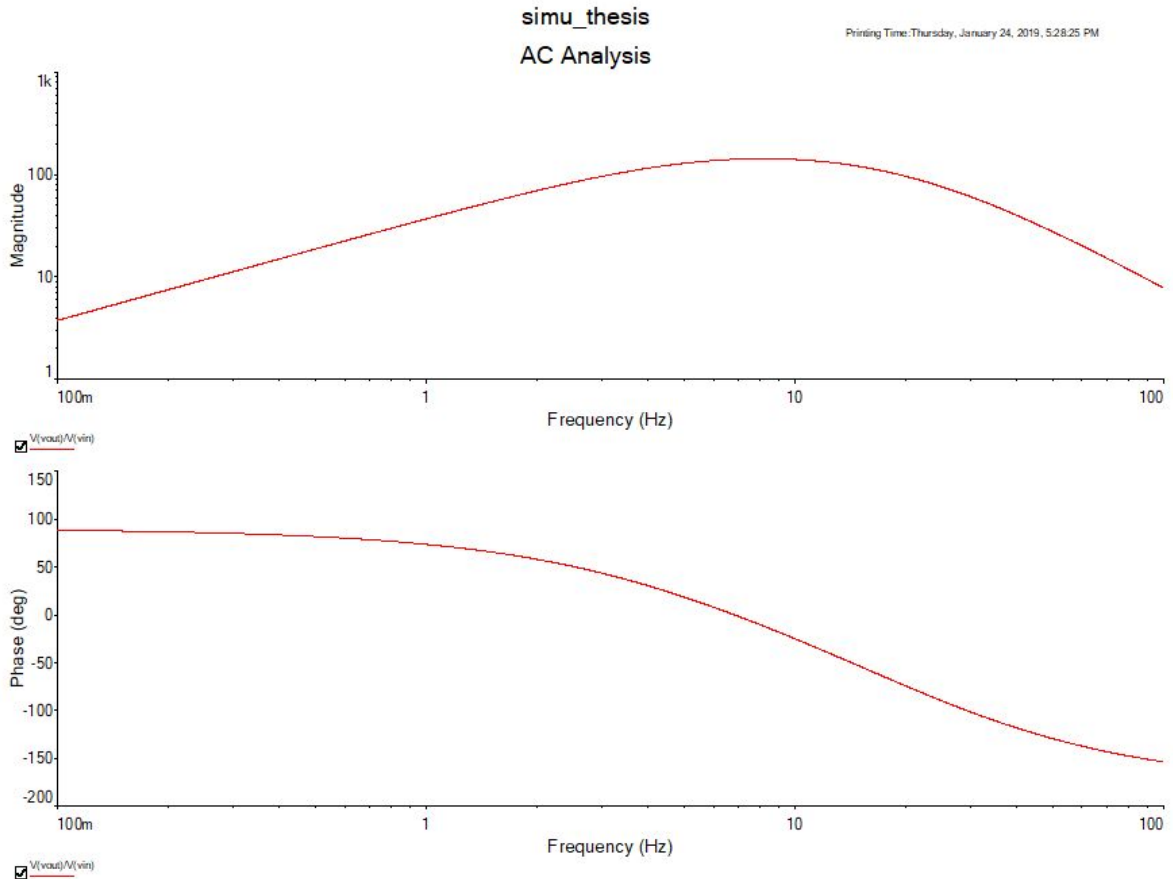
Το σήμα (τάσης πλέον), από την έξοδο του ενισχυτή διαγωγιμότητας, στη συνέχεια περνάει από ένα δεύτερο στάδιο αναστρέφοντος ενισχυτή, σκοπός του οποίου είναι το τελικό φιλτράρισμα, ενίσχυση και φυσικά η προσαρμογή του dc τμήματος του στο μέσο του δυναμικού εύρους του κυκλώματος (1.65V). \*Στην δική μας εφαρμογή δεν μας ενδιαφέρει το DC τμήμα της απόκρισης καθώς μας ενδιαφέρει να καταγράψουμε απλά το “πέρασμα” ενός παλμού (AC). Το τελικό κύκλωμα παρουσιάζεται παρακάτω:



Εικόνα 3.13 : Το συνολικό analog front-end του PPG αισθητήρα

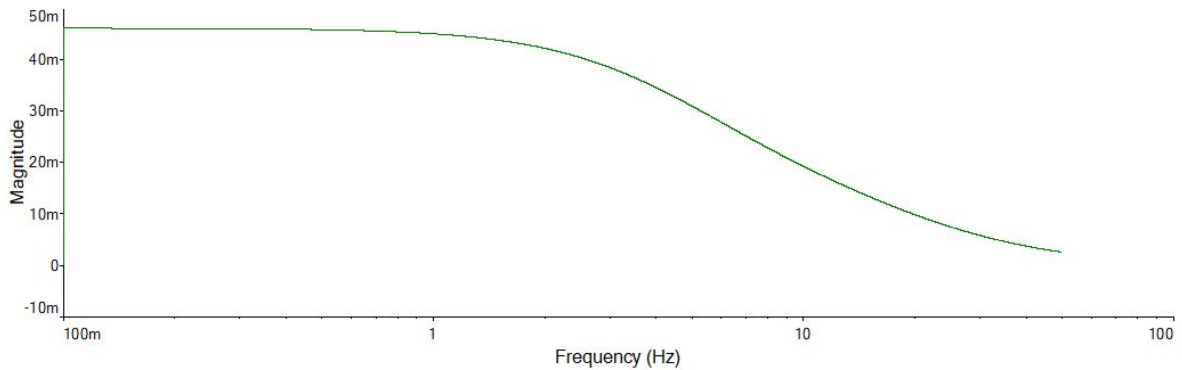
Και τα δύο βαθυπερατά φίλτρα που βρίσκονται ανάμεσα στην είσοδο του συνολικού κυκλώματος και την έξοδο (C4R6,R2C3) έχουν την συχνότητα αποκοπής τους, χαμηλότερα από τα 50Hz του δικτύου ώστε να την αποκόπτουν σε μεγάλο μέρος. Η ενίσχυση του πρώτου σταδίου (τιμή της R6) είναι κατάλληλη ώστε η dc τάση στην έξοδο του να προσεγγίζει τα 3.3V (3V είναι μια καλή τιμή) . Υποθέτοντας πως η DC τάση στην έξοδο αυτού του σταδίου είναι λοιπόν 3V (κάτι που θα επιβάλλουμε σε κάθε περίπτωση με μια σχεδίαση που θα παρουσιαστεί σε επόμενο κεφάλαιο), η ενίσχυση του δεύτερου σταδίου μπορεί να καθοριστεί ως εξής: Στη βιβλιογραφία βλέπουμε πως το AC περιεχόμενο του σήματος από την φωτοδίοδο αναμένουμε να είναι 1-2% του DC μεγέθους του (σε μεγέθη V peak-to-peak). Συμπεραίνουμε πως με την ενίσχυση ~50-60 του δεύτερου σταδίου θα έχουμε ένα σήμα με  $V_{pp} \sim 3.3$  και αξιοποιούμε όλο το εύρος του ενισχυτή.

Πριν κλείσουμε το συγκεκριμένο υποκεφάλαιο όμως, αξίζει να σταθούμε σε ένα λεπτό σημείο του σχεδιασμού του παραπάνω κυκλώματος. Ο στόχος των κυκλωμάτων που σχεδιάζουμε δεν είναι απλά η καταγραφή των σχημάτων των κυματομορφών PPG και PCG, αλλά και κάποιες συγκεκριμένες χρονικές διαφορές μεταξύ τους. Το κύκλωμα του PPG είναι “γρήγορο” οπότε δεν παράγει σημαντικές καθυστερήσεις στο σήμα της εισόδου. Το παραπάνω κύκλωμα όμως είναι σχετικά αργό, παράγει μετρήσιμη καθυστέρηση σε ένα σήμα εισόδου. Αυτό έρχεται ως κόστος στην επιλογή, να λειτουργεί κάτω από τη συχνότητα του δικτύου των 50Hz, ώστε να έχουμε ένα καθαρό σήμα δίχως την ανάγκη ιδιαίτερης ψηφιακής επεξεργασίας σήματος ή την προσθήκη κυκλωμάτων τύπου Notch-Filter. Παρακάτω βλέπουμε την απόκριση του συνολικού κυκλώματος:



Εικόνα 3.14 : Η απόκριση του συνολικού analog front-end του PPG αισθητήρα

Περισσότερες πληροφορίες έχουμε στο γράφημα του Group delay :



Εικόνα 3.15 : Group Delay του analog front-end του PPG αισθητήρα

Στις συχνότητες ενδιαφέροντος βλέπουμε ένα μη σταθερό group delay. Συμπεραίνουμε πως θα έχουμε μια μικρή παραμόρφωση του σήματος εισόδου από το κύκλωμα και μερικές δεκάδες ms καθυστέρηση. Αυτό μπορεί να επηρεάσει τις μετρήσεις μας του PTTp (Αυτές δηλαδή που χρησιμοποιούν την κορυφή του PTT γραφήματος ως σημείο έλευσης παλμού).

### 3.3 Ψηφιακό κύκλωμα & Μικροελεγκτής

Ένας μικροελεγκτής STM32F401RE αποτελεί το βασικό τμήμα του ψηφιακού κυκλώματος που θα περιγράψουμε στη συνέχεια. Πρόκειται για έναν μικροελεγκτή χαμηλού κόστους αλλά με πολλές δυνατότητες. Περιλαμβάνει έναν επεξεργαστή ARM 32bit CORTEX-M4 , 512 KByte flash , 96KByte sRAM και έναν μεγάλο αριθμό από περιφερειακά (μετρητές, ADC,DMA κτλπ) , αρκετά από τα οποία θα χρησιμοποιήσουμε.

Στο σχηματικό που φαίνεται παρακάτω, διακρίνονται τα ψηφιακά τμήματα της συσκευής. Θα περιγράψουμε εδώ κάποια στοιχεία από αυτά:

- Ένα κουμπί για reset. Σύμφωνα με το datasheet του μικροελεγκτή, ένα μηδέν (0V) στο πιν NRST προκαλεί hardware reset.

- Ένα κουμπί για Interrupt από τον χρήστη. Χρήσιμο για αλλαγή κατάστασης ασύγχρονα, από ενέργεια του χρήστη

- Ένα LED ένδειξης λειτουργίας, χαμηλής κατανάλωσης (1mA)

- Υποδοχή 5 πιν για σύνδεση με προγραμματιστή ST-LINK. Πρόκειται για ένα πρωτόκολλο επικοινωνίας απο την STM, το οποίο με 5 πιν επιτρέπει πλήρη έλεγχο του τσιπ, προγραμματισμό και debug.

- Υποδοχή USB για σύνδεση απευθείας με υπολογιστή. Σύμφωνα με το datasheet του συγκεκριμένου μικροελεγκτή εμπεριέχει περιφερειακό ( το οποίο μπορεί να ενεργοποιηθεί μέσω λογισμικού) που περιλαμβάνει το απαραίτητο hardware, ικανό να επικοινωνήσει με ταχύτητες 12Mbit/sec ( USB1.0).

- Υποδοχή 12 πιν για σύνδεση με οθόνη LCD 2x16 χαρακτήρων, που λειτουργεί με τη χρήση ενός τσιπ HD44780. Με το βαθυπερατό φίλτρο το οποίο παρεμβάλλεται ανάμεσα σε πιν του μικροελεγκτή και το πιν στο οποίο ελέγχεται κανονικά η αντίθεση της οθόνης, δίνεται η δυνατότητα του ελέγχου της αντίθεσης μέσω pwm από τον μικροελεγκτή, άρα μέσω software αντί να απαιτείται μεταβλητή αντίσταση. Το παραπάνω τσιπ, διαθέτει λειτουργία 4-πιν και 8-πιν για την επικοινωνία του. Εδώ χρησιμοποιούμε την λειτουργία των 4-πιν.

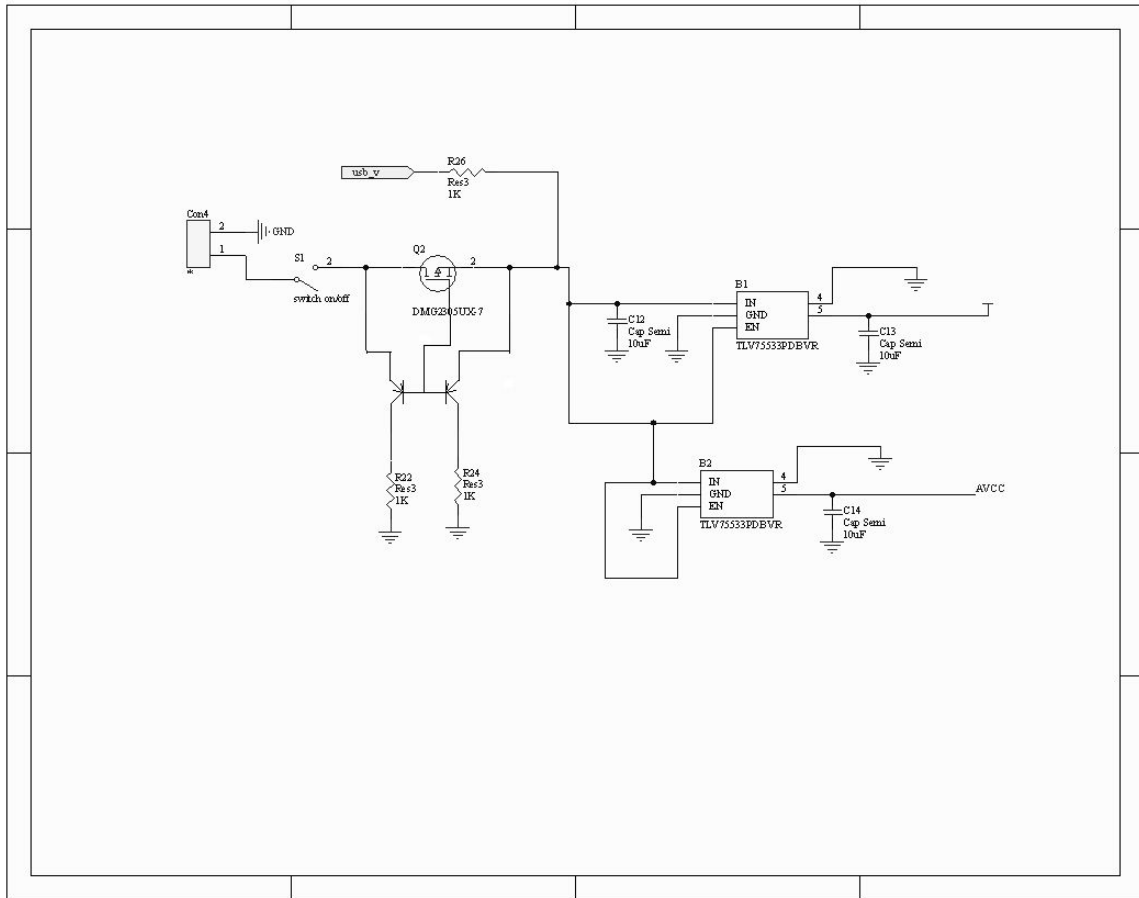
- Ο συγκεκριμένος μικροελεγκτής διαθέτει μία μονάδα ADC, αλλά “μοιράζοντάς” το ένα κανάλι του σε πολλαπλά πιν, είναι δυνατό να διαβάσει πολλά pin διαδοχικά ή αλλιώς.(περισσότερα για αυτό στο κεφάλαιο με τον προγραμματισμό του) Εδώ θα χρησιμοποιήσουμε 4 διαφορετικά πιν για δειγματοληψία και επεξεργασία: 1. PPG σημα, 2.H DC τάση στην έξοδο του πρώτου σταδίου του αναλογικού front end του PPG (ώστε να μπορέσουμε να την ελέγξουμε) , 3.PCG σήμα, 4. Πρόβλεψη για πιθανή 3η είσοδο στη συσκευή (π.χ. μπορεί να συνδεθεί μονοκάναλο ECG για σύγκρισή του με το PCG ως προς την ποιότητα των μετρούμενων χρόνων άφιξης παλμού)

- Ένα πιν χρησιμοποιείται για να ελέγξει το κύκλωμα της τροφοδοσίας του υπέρυθρου λεντ για τον αισθητήρα του PPG. Ο μικροελεγκτής, με χρήση PWM θα μπορεί να ελέγξει την φωτεινότητα του λεντ ανοίγοντας και κλείνοντάς το σε υψηλή συχνότητα (μακριά από τις συχνότητες του PPG σήματος).



### 3.4 Τελική κατασκευή

Για την τελική κατασκευή της συσκευής, σχεδιάστηκε τυπωμένη πλακέτα PCB. Πριν αναφέρουμε μερικά στοιχεία όσον αφορά αυτή, παρακάτω παρουσιάζεται το κύκλωμα της τροφοδοσίας της συσκευής που σχεδιάσαμε:



Εικόνα 3.17 : Το σχηματικό της τροφοδοσίας της συσκευής

Όσον αφορά αυτό επισημαίνουμε τα εξής:

-Χρησιμοποιήθηκε το γραμμικό τροφοδοτικό (linear regulator) TLV75533PDBVR. Το συγκεκριμένο επιλέχθηκε, λόγω του χαμηλού κόστους και μικρού αποτυπώματος (SOT23-5) και της δυνατότητάς του να σταθεροποιεί την έξοδό του στα 3.3V εφόσον η τροφοδοσία στην είσοδό του είναι από 3.52V έως 5.5V (low dropout  $V_{do}=220mV$ ). Αυτό θα επιτρέπει στην συσκευή να λειτουργεί είτε ως φορητή με τη χρήση ενός στοιχείου Li-ion των 3.7V είτε με τη χρήση της τροφοδοσίας από το USB όταν θα συνδέεται με υπολογιστή.

-Για την αποφυγή της μόλυνσης των αναλογικών κυκλωμάτων με θόρυβο από τα ψηφιακά κυκλώματα, χρησιμοποιήθηκαν δύο από τα παραπάνω τροφοδοτικά και διαχωρίστηκε η τροφοδοσία των πρώτων κυκλωμάτων από τα δεύτερα.

-Για την προστασία της μπαταρίας όταν η συσκευή συνδεθεί στον υπολογιστή είναι απαραίτητη η τοποθέτηση διόδου μπροστά από αυτήν, ώστε να αποτραπεί η ροή ρεύματος

προς αυτήν. Όμως, μια απλή δίοδος (π.χ. 1N4001) θα προκαλούσε σημαντική πτώση τάσης που θα έριχνε υπερβολικά την τάση του στοιχείου Li-ion, με αποτέλεσμα αυτή να είναι ανεπαρκής για την τροφοδοσία των γραμμικών τροφοδοτικών. Το υποκύκλωμα που διακρίνεται παραπάνω με τα 2 BJT και το Mosfet λειτουργεί ως ιδανική δίοδος ( $R_{on} \sim 5m\Omega$ ) λύνοντας το πρόβλημα.

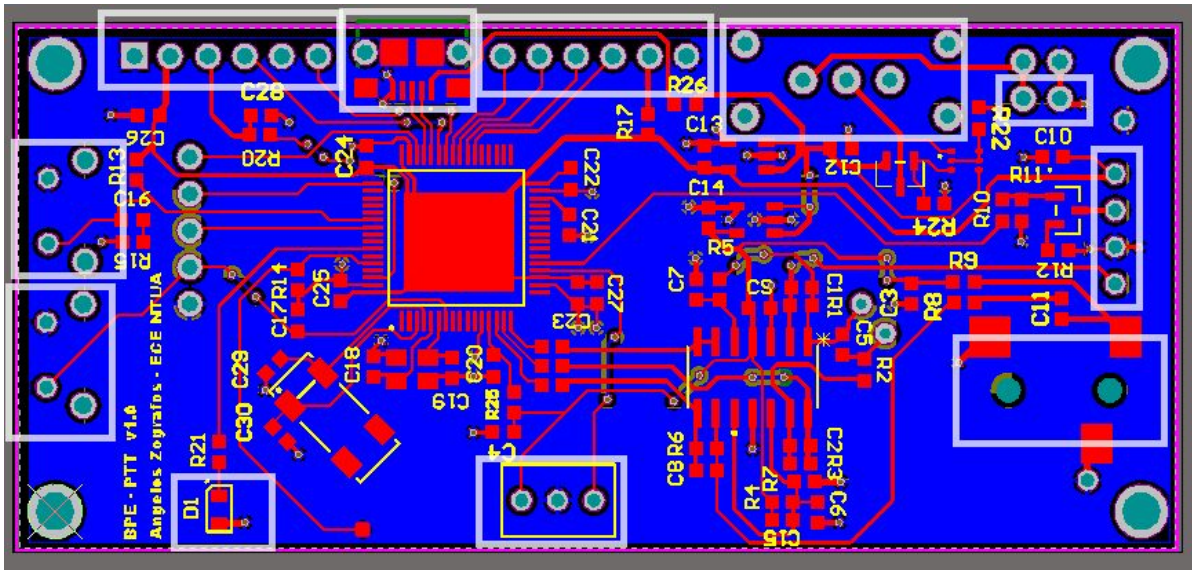
Έχοντας περιγράψει τώρα όλα τα στοιχεία της συσκευής ( αναλογικά και ψηφιακά κυκλώματα και τροφοδοσία) προχωρούμε στην περιγραφή της σχεδίασης της πλακέτας PCB.

Η οθόνη η οποία επιλέχθηκε να χρησιμοποιηθεί είναι μία πολύ συνηθισμένη LCD 2x16 χαρακτήρων με διαστάσεις 80mm x 36mm. Με πρότυπο αυτές τις διαστάσεις, τέθηκε ο στόχος η πλακέτα που θα σχεδιαστεί να έχει ακριβώς αυτές τις διαστάσεις, ώστε η οθόνη να εφαρμόσει πάνω από αυτή. Έτσι για την σχεδίαση της πλακέτας:

-Χρησιμοποιούμε αντιστάσεις και πυκνωτές SMD αποτυπώματος 0603(imperial) ακόμα και για τους μεγάλους σε χωρητικότητα πυκνωτές; Η τεχνολογία MLCC (κεραμικών πολυστρωματικών πυκνωτών) επιτρέπει μεγέθη σχεδόν mF σε συσκευασία 1.6mm.

- Για το σύνολο των λειτουργιών του αναλογικού τμήματος απαιτούνται 4 τελεστικοί ενισχυτές ( 2 για το PPG, 1 για το PCG και 1 σε συνδεσμολογία Buffer για τη λήψη της ενδιάμεσης DC τάσης στο αναλογικό κύκλωμα του PPG που αναφέραμε προηγουμένως). Το ολοκληρωμένο TLV2464IDR διαθέτει 4 τελεστικούς ενισχυτές με όλα τα χαρακτηριστικά που υποθέσαμε (λειτουργία στα 3.3V , RAIL-to-RAIL κτλπ) σε μικρό αποτύπωμα.

Παρακάτω φαίνεται η σχεδιασμένη πλακέτα:



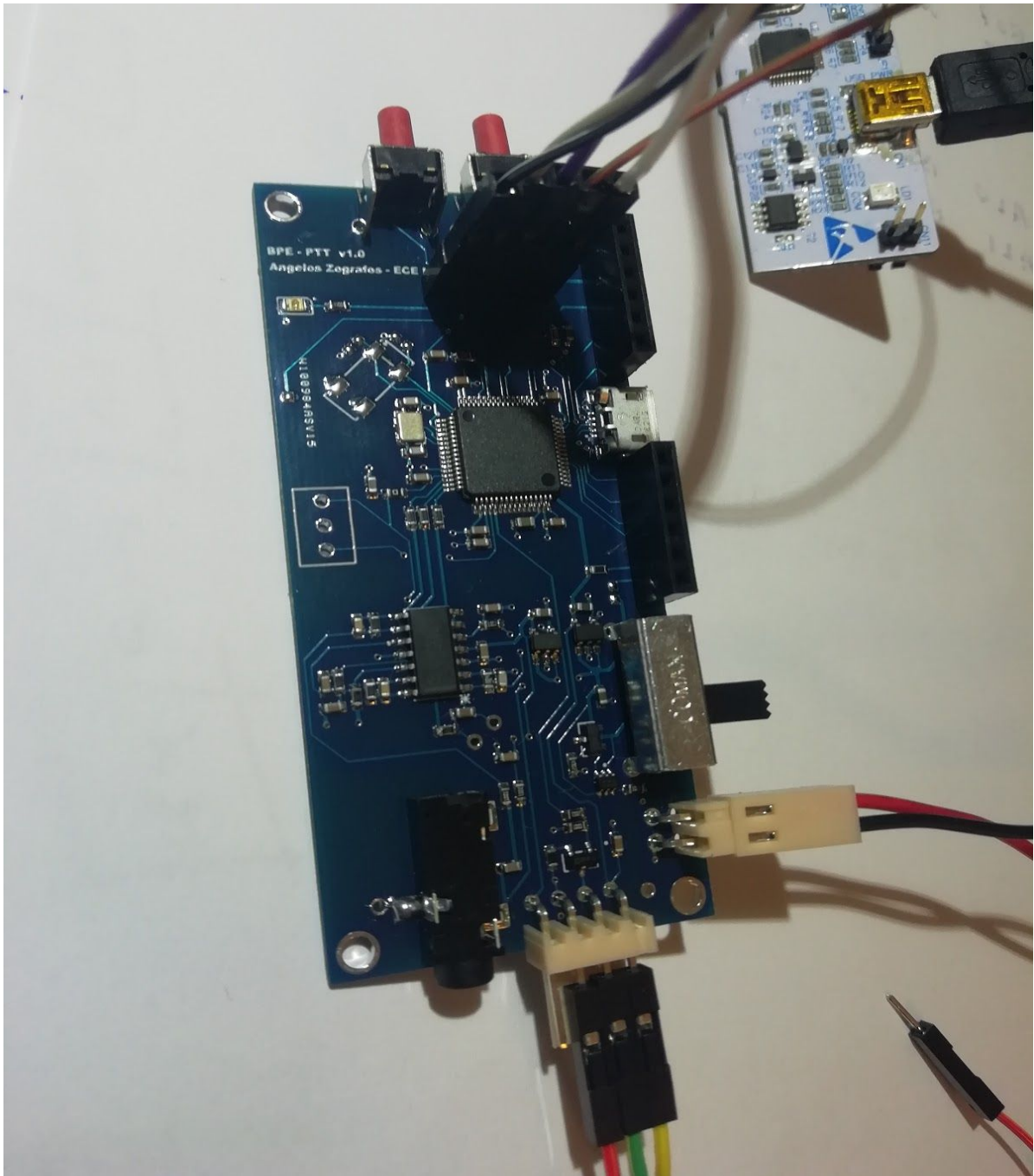
Εικόνα 3.18 : Η πάνω πλευρά της PCB που σχεδιάστηκε

Μέσα σε λευκά τετράγωνα, ξεκινώντας πάνω αριστερά και με τη φορά του ρολογιού διακρίνονται με τη σειρά:

1. LCD connector nr.1
2. Micro USB
3. LCD connector nr.2
4. Οριζόντιος διακόπτης SPDT
5. molex 2-pin male connector για τροφοδοσία (π.χ. από μπαταρία)
6. molex 4-pin male connector για τον αισθητήρα PPG (με πρόβλεψη για σύνδεση γειωμένου-shielded καλωδίου)
7. Audio Jack 2.5mm για τον αισθητήρα PCG (με πρόβλεψη για σύνδεση γειωμένου-shielded καλωδίου)
8. Πρόβλεψη για 3η είσοδο στον ADC του stm32 (π.χ. single channel ECG)
9. Ενδεικτικό λεντ χρώματος μπλε
10. Οριζόντιο πλήκτρο Reset
11. Οριζόντιο πλήκτρο χρήστη
12. Σύνδεση ST-LINK (5-pin) \*διακρίνεται πίσω από το πλήκτρο χρήστη.

Η πλακέτα 2-στρωμάτων κατασκευάστηκε σε εξειδικευμένο εργοστάσιο για PCB στο οποίο στάλθηκαν τα Gerber αρχεία που δημιουργήθηκαν κατά τη σχεδίαση της πλακέτας. Για την κόλληση των στοιχείων χρησιμοποιήθηκε stencil και η τεχνική Reflow σε ειδικό φούρνο υπερύθρων για τα SMD στοιχεία και κόλληση με το χέρι για τα through-hole.

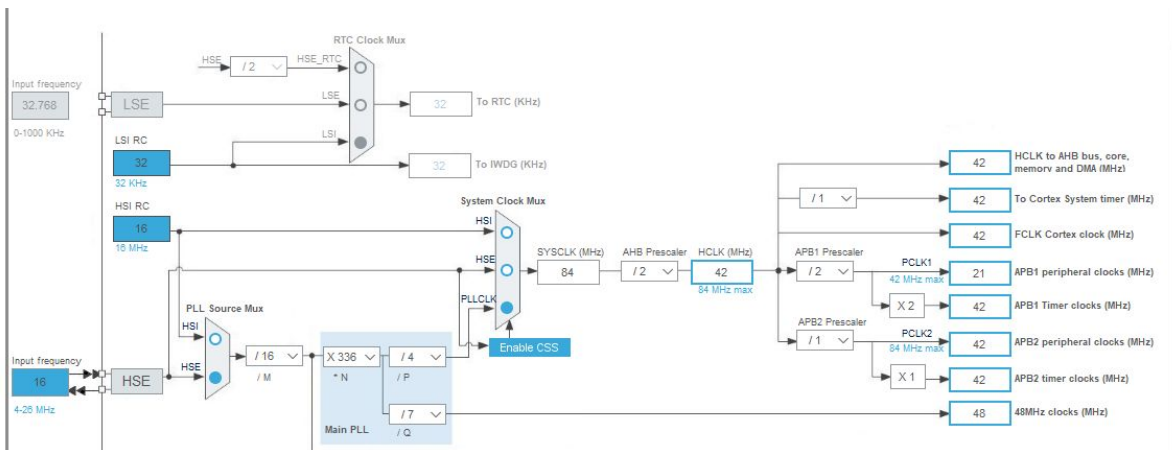
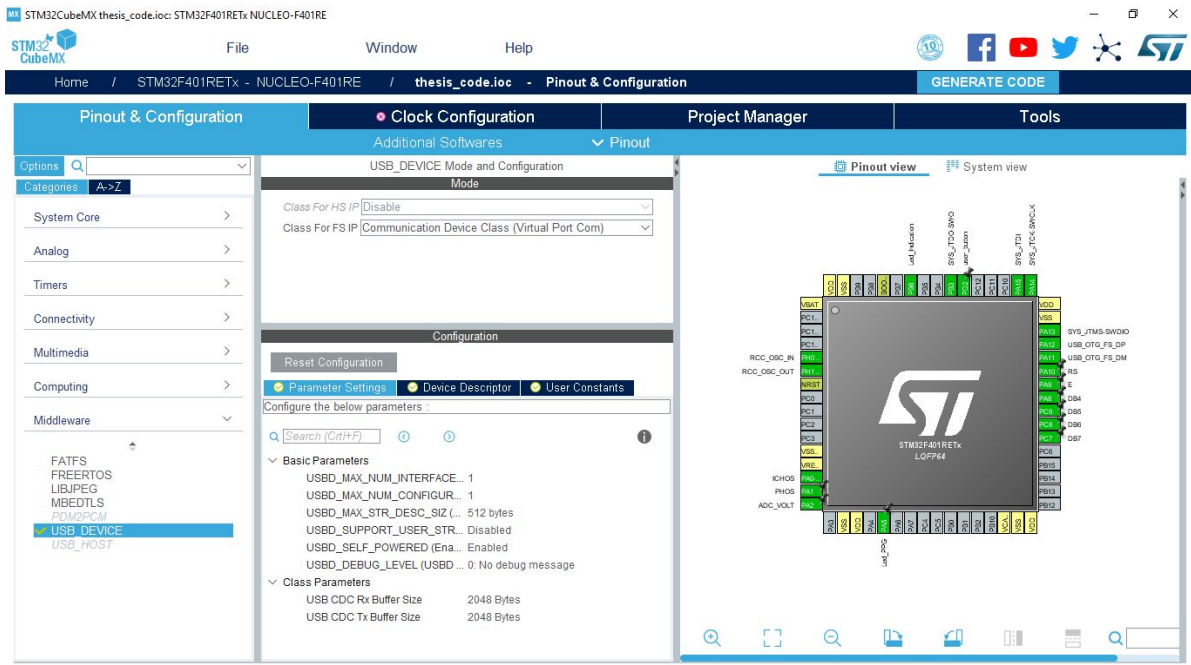




*Εικόνα 3.19 : Η συναρμολογημένη πλακέτα, δίχως την οθόνη, ενόσω προγραμματίζεται μέσω ST-LINK*

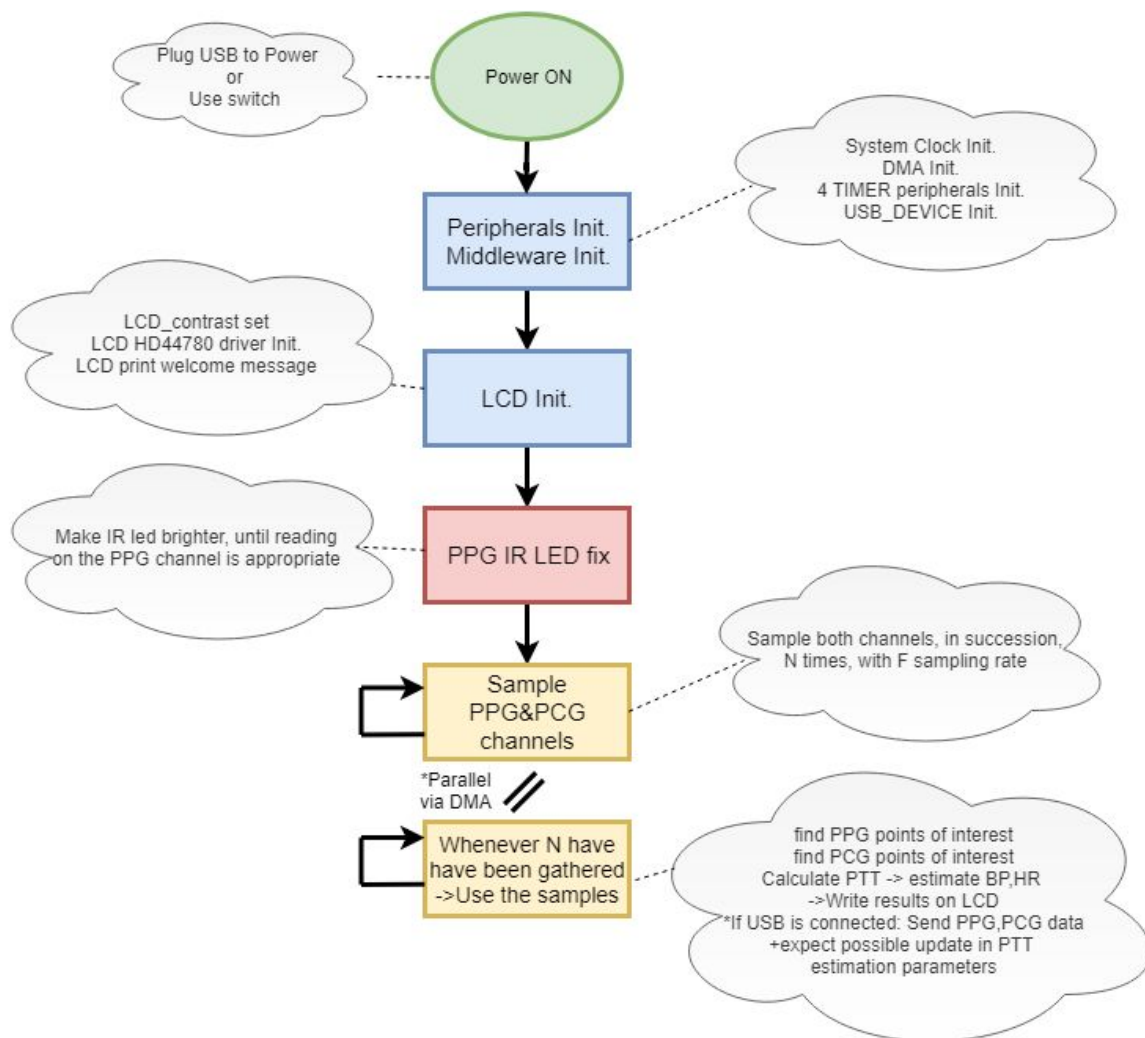
### 3.5 Προγραμματισμός

Για τον προγραμματισμό της συσκευής (συγκεκριμένα του STM32F401RE μικροελεγκτή) χρησιμοποιήθηκε το πρόγραμμα της STM, CubeMX, για την αρχική προσαρμογή κάποιων περιφερειακών, των ρολογιών και την εγκατάσταση κάποιων -απαραίτητων για την εφαρμογή μας- Middleware (π.χ. του USB device). Κάτω, διακρίνονται δύο εικόνες από το περιβάλλον του συγκεκριμένου προγράμματος.



Εικόνα 3.20 : Το περιβάλλον του προγράμματος CubeMX για την αρχικοποίηση των περιφερειακών και του ρολογιού ενός stm32 μικροελεγκτή

Στη συνέχεια, ο προγραμματισμός έγινε στο Atollic TrueSTUDIO στη C γλώσσα προγραμματισμού με χρήση των βιβλιοθηκών HAL για τον έλεγχο των περιφερειακών. Η γενική αρχιτεκτονική του κώδικα που γράφτηκε για την εφαρμογή μας παρουσιάζεται στη συνέχεια.



Εικόνα 3.21 : Διάγραμμα της ροής του προγράμματος που δημιουργήθηκε για το σύστημα

Μερικά στοιχεία που αφορούν την παραπάνω υλοποίηση:

- Για την χρήση της οθόνης LCD γράφτηκε η κατάλληλη βιβλιοθήκη για την απαραίτητη επικοινωνία με το τσιπ HD44780. Το τσιπ αυτό, είναι ενσωματωμένο στην οθόνη που χρησιμοποιήθηκε και είναι υπεύθυνο για τον έλεγχο των στοιχείων της. Με λίγα λόγια αυτό το τσιπ λειτουργεί ως εξής: Δέχεται δεδομένα και εντολές μήκους 8bit σε πακέτα των 4bit (δύο πακέτα για τη συμπλήρωση μιας εντολής/δεδομένου). Ένα 5ο πιν (RS) ορίζει αν τα 8bit που στέλνονται είναι δεδομένα ή εντολή και ένα 6ο πιν (E) χρησιμοποιείται για τον χρονοισμό (latching) της κάθε τετράδας bit. Η λογική της τοποθέτησης των χαρακτήρων είναι αυτή του κέρσορα. Αρχικά τοποθετείται στην θέση εκείνη (από τις συνολικά 16x2) που θέλουμε να γράψουμε και έπειτα στέλνεται ο χαρακτήρας σε ASCII που θα τοποθετηθεί εκεί.

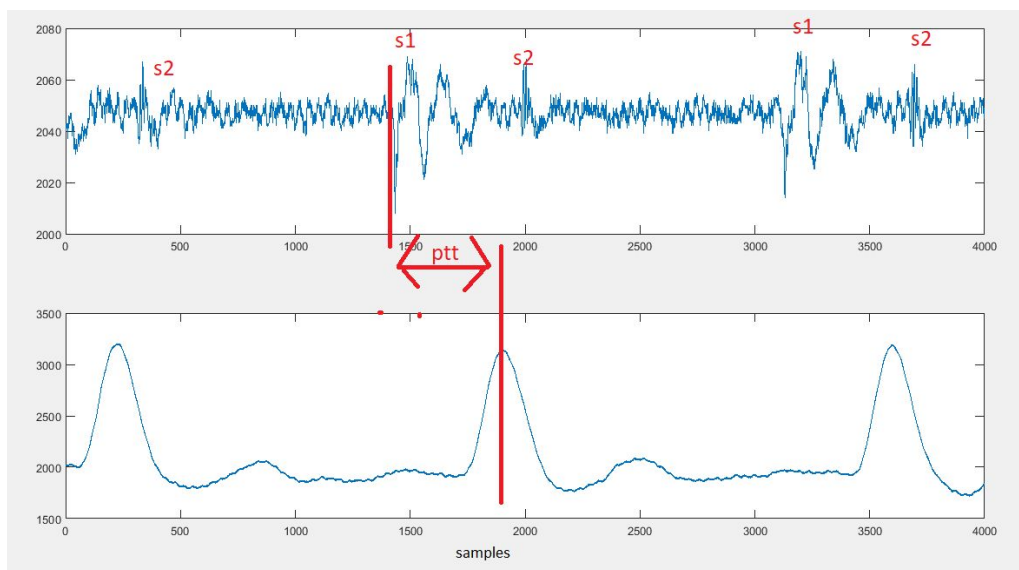
- Στη σχεδίαση του αναλογικού κυκλώματος και συγκεκριμένα στο αναλογικό front-end για το PPG, είχαμε υποθέσει πως το DC μέγεθος του σήματος, στην έξοδο του πρώτου σταδίου, σε αυτό το κύκλωμα, είναι  $\sim 3V$ . Εδώ δείχνουμε πως το επιτυγχάνουμε αυτό. Καθώς αυτό το μέγεθος εξαρτάται από την σχετική με το δέρμα τοποθέτηση του αισθητήρα, τις ιδιότητες του δέρματος του ασθενή (που διαφέρουν από άνθρωπο σε άνθρωπο ανάλογα π.χ. με το ποσό μελανίνης σε αυτό), τη γήρανση της φωτοδιόδου κτλ., ο μόνος τρόπος να πετύχουμε αυτή η υπόθεση να ικανοποιείται, κατά τη λειτουργία της συσκευής, είναι να το επιβάλλουμε κάθε φορά που επαναχρησιμοποιείται (ενεργοποιείται) η συσκευή. Αυτό που επιλέξαμε να κάνουμε εδώ, είναι, πριν ξεκινήσει η διαδικασία της ανάγνωσης των δύο αναλογικών καναλιών, να αυξάνουμε αυτόματα την φωτεινότητα του υπέρυθρου λεντ ( δια της αύξησης του duty cycle του σήματος ελέγχου του λεντ), σταδιακά και διαρκώς να μετράμε το DC που θέλουμε να επιβάλλουμε να είναι 3V. Η διαδικασία της αύξησης της φωτεινότητας σταματάει όταν αυτό επιτευχθεί.
- Όπως ειπώθηκε και σε προηγούμενο κεφάλαιο, ο συγκεκριμένος MCU διαθέτει μία μονάδα ADC. Για να καταγράψουμε στην περίπτωσή μας από δύο κανάλια, ο ADC προγραμματίστηκε να δειγματοληπτεί επαναληπτικά σε σειρά τα δύο αυτά κανάλια. Η χρονική καθυστέρηση που προξενείται ανάμεσα στα δύο κανάλια από τη λειτουργία αυτή, είναι της τάξης των  $\mu s$ , οπότε δεν επηρεάζουν την ακρίβεια της μετρούμενης ποσότητας (PTT)
- Ο συγκεκριμένος μικροελεγκτής, διαθέτει μονάδα DMA (Direct Memory Access). Ο συγκεκριμένος ελεγκτής, εάν χρησιμοποιηθεί, δίνει την δυνατότητα της μεταφοράς δεδομένων από περιφερειακά στην μνήμη, από τη μνήμη στη μνήμη κτλ. δίχως την ανάγκη να απασχολήσει τον επεξεργαστή. Εδώ χρησιμοποιήσαμε φυσικά τη συγκεκριμένη μονάδα με τον εξής τρόπο: Αντί να καθυστερούμε τον επεξεργαστή για κάθε μετατροπή(δειγματοληψία) από τον ADC, η δειγματοληψία και η καταγραφή των αποτελεσμάτων στην RAM γίνεται παράλληλα με την επεξεργασία των δεδομένων που καταγράφηκαν κατά την προηγούμενη δειγματοληψία (σε παράθυρα μεγέθους N δειγμάτων). Όσο δηλαδή καταγράφονται τα νέα δεδομένα PPG και PCG, υπολογίζεται το PTT ,εκτιμάται η BP και στέλνονται τα δεδομένα στην LCD οθόνη (και πιθανώς στον υπολογιστή μέσω του USB) για το προηγούμενο καταγεγραμμένο παράθυρο.

## 4.Χρήση της συσκευής

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις δυνατότητες της συσκευής που κατασκευάστηκε, περιγράφοντας ένα παράδειγμα της χρήσης της. Αρχικά, θα παρουσιάσουμε μια πρόταση για την επεξεργασία των σημάτων PCG και PPG και τον υπολογισμό του PTT. Στη συνέχεια, θα χρησιμοποιήσουμε ένα παραμετρικό μοντέλο  $BP=BP(PTT)$  και θα το προσαρμόσουμε σε έναν ασθενή ώστε να μπορούμε να εκτιμήσουμε την αρτηριακή πίεση.

### 4.1 Ψηφιακή Επεξεργασία σήματος

Για να μπορέσουμε να εκτιμήσουμε το χρόνο διάδοσης παλμού αυτόματα, είναι απαραίτητο να σχεδιάσουμε αλγόριθμους που να είναι σε θέση να εντοπίζουν τα απαραίτητα σημεία ενδιαφέροντος στα σήματα PPG και PCG. Συνδέουμε λοιπόν την συσκευή στον υπολογιστή και τους αισθητήρες πάνω σε έναν ασθενή. Με χρήση του Matlab δημιουργήσαμε ένα πρόγραμμα, που να επικοινωνεί με την συσκευή και να λαμβάνει και να σχεδιάζει γραφήματα, με τα δύο αυτά σήματα, όπως αυτά στέλνονται από αυτήν. Ένα τυχαίο στιγμιότυπο από αυτό φαίνεται παρακάτω.



Εικόνα 4.1 : Καταγεγραμμένα μέσω της συσκευής, ενδεικτικά σήματα. Τα γραφήματα κατασκευασμένα στο πρόγραμμα Matlab.

Η συχνότητα δειγματοληψίας του ADC της συσκευής έχει τεθεί στα 2kHz .Ο οριζόντιος άξονας για να αποκτήσει νόημα τάσης, θα πρέπει να πολλαπλασιαστεί με  $(3.3V/2^{12})$ , αλλά δεν θα μας δώσει κάποια επιπλέον πληροφορία το να κάνουμε αυτή την πράξη. Το ενδιαφέρον εδώ είναι η μορφή των σημάτων. Επίσης στην εικόνα έχει σχεδιαστεί το

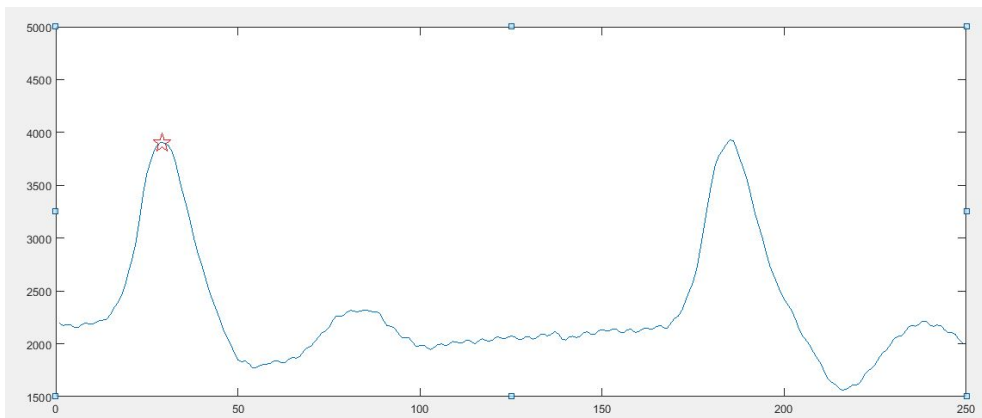
μέγεθος που θέλουμε να μετρήσουμε (αν επιλέξουμε να μετρήσουμε από συστολή καρδιάς, δηλαδή προσεγγιστικά τον ήχο S1 σε κορυφή στο PPG). Το παραπάνω λοιπόν είναι ένα αντιπροσωπευτικό δείγμα των μετρήσεων και των δεδομένων που θα έχει στη διάθεση του ο MCU της συσκευής προκειμένου να υπολογίσει το PTT.

Όπως είναι προφανές, για τον υπολογισμό του PTT, θα πρέπει να κατασκευάσουμε αφενός έναν αλγόριθμο ο οποίος να βρίσκει τα σημεία στο οποίο έχουμε τη συστολή της κοιλίας (κατά προσέγγιση αρχή του ήχου S1) στο PCG σήμα και αφετέρου τις κορυφές του PPG (έχοντας αθόρυβα εδώ επιλέξει να μετρήσουμε το PTTp μέγεθος). Για πρακτικούς λόγους, το Matlab, χρησιμοποιήθηκε ώστε να δημιουργηθούν και να δοκιμαστούν οι αλγόριθμοι που επιτυγχάνουν τους παραπάνω στόχους. Εφόσον οι αλγόριθμοι αυτοί σχεδιάστηκαν και δοκιμάστηκαν επιτυχώς σε έναν αριθμό από δεδομένα-δείγματα, όπως αυτό που απεικονίζεται παραπάνω, οι αλγόριθμοι ξαναγράφηκαν σε γλώσσα C για να μπορέσουν να φορτωθούν στον μικροελεγκτή.

Ο αλγόριθμος για την εύρεση των κορυφών στο PPG είναι σχετικά εύκολος. Χρησιμοποιώντας χαρακτηριστικά που γνωρίζουμε για το συγκεκριμένο σήμα, μπορούμε να εντοπίσουμε τις κορυφές χωρίς ιδιαίτερα προβλήματα ως εξής:

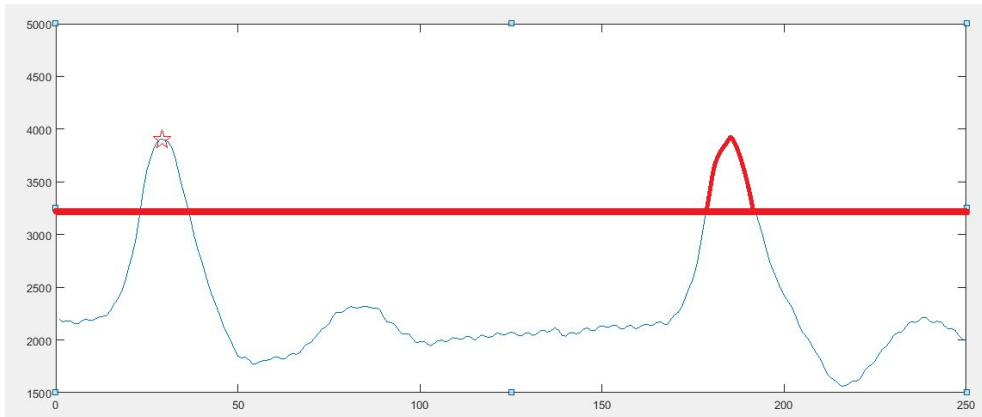
- Αρχικά φιλτράρουμε το σήμα (με φίλτρο μηδενικής φάσης για να αποφύγουμε να το αλλοιώσουμε). Συγκεκριμένα, εφαρμόζουμε υποδειγματοληψία στη συχνότητα 200 Hz χρησιμοποιώντας τη μέση τιμή ανά 10 μετρήσεων του αρχικού σήματος.

- Εντοπίζουμε το ολικό μέγιστο του παραθύρου. Η θέση  $x_p$  και η τιμή  $\max(\text{signal})$  του αποθηκεύεται ως το πρώτο σημείο-κορυφή του ppg.



Εικόνα 4.2 : Εύρεση κορυφής στο συνολικό PPG σήμα

- Στη συνέχεια εντοπίζουμε τα σημεία εκείνα τα οποία έχουν τιμή πάνω από την τιμή  $\text{Min}(\text{Signal}) + \sigma * (\text{Max}(\text{Signal}) - \text{Min}(\text{Signal}))$  όπου  $\sigma = 0.7$ . Δεν αναμένουμε μεγάλη διαφοροποίηση ανάμεσα από διαδοχικά μέγιστα.



Εικόνα 4.3 : Δεύτερο βήμα αλγορίθμου εύρεσης κορυφών στο PPG σήμα

-Στις περιοχές που εντοπίζονται με τον παραπάνω τρόπο βρίσκουμε εκ νέου τα μέγιστα .

Με τον παραπάνω αλγόριθμο (με μερικές επιπλέον λεπτομέρειες όπως π.χ. πως δεν δεχόμαστε μέγιστα που βρέθηκαν στα άκρα του παραθύρου), μπορούμε με ασφάλεια να εντοπίζουμε όλες τις κορυφές του ppg σήματος.

Με τον παραπάνω αλγόριθμο βρίσκουμε όλες τις κορυφές ανεξάρτητα από τον αριθμό τους στο υπό εξέταση παράθυρο (ανεξάρτητα δηλαδή από τον καρδιακό ρυθμό).

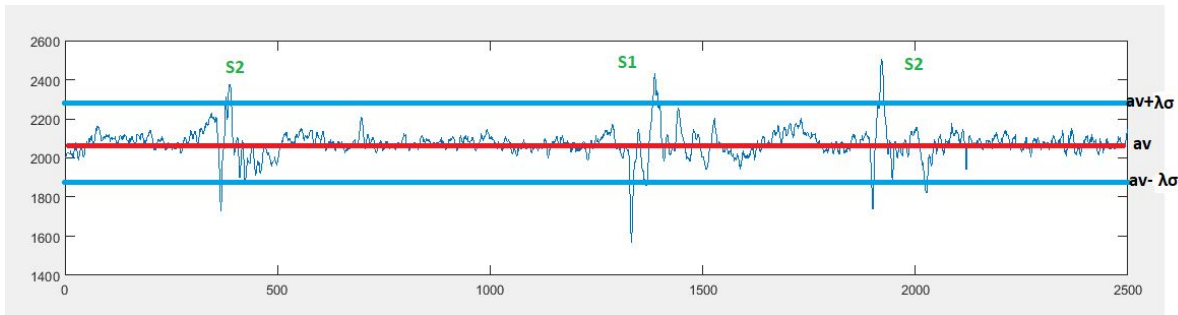
Ο αλγόριθμος για την εύρεση των σημείων ενδιαφέροντος στο PCG σήμα, είναι λιγότερο εύκολος να κατασκευαστεί, ειδικά εφόσον θέλουμε να δίνει την ίδια σχετική με το σήμα θέση, σε κάθε επανάληψη του πειράματος. Η τοποθέτηση της κεφαλής του στηθοσκοπίου ( η θέση στο στήθος και η γωνία τοποθέτησής του) επηρεάζει σημαντικά το σήμα και αυτό όντας ηχητικό, δεν έχει σαφείς κορυφές. Τέλος, ο αισθητήρας θα λάβει διάφορους θορύβους που προέρχονται από την αναπνοή, κυκλοφορία του αίματος κτλ. Για να επιτύχουμε επαναληψιμότητα, αναπτύχθηκε μια μέθοδος η οποία αποδείχθηκε στην πράξη πως δίνει καλά αποτελέσματα. Στηριζόμαστε σε δύο ιδιότητες του σήματος :

1. Η ενέργεια του σήματος (ένταση του ήχου) που προέρχεται από τον καρδιακό χτύπο (και συγκεκριμένα τους ήχους S1 και S2) είναι μεγαλύτερη από όλες τις άλλες πηγές. Προφανώς το παραπάνω δεν ισχύει αν δεν είναι τοποθετημένος σωστά ο αισθητήρας, ή αν π.χ. ο ασθενής μιλάει ή βήχει.
2. Οι ήχοι για τους οποίους ενδιαφερόμαστε (S1, S2) εξαιρώντας σπάνιες παθολογικές καταστάσεις, δεν παρουσιάζονται τυχαία. Γενικά ισχύει πως ο ήχος S1 προηγείται του ήχου S2 και η απόσταση μεταξύ τους είναι αρκετά μικρότερη από την απόσταση μεταξύ του ήχου S2 και του S1 του επόμενου παλμού.

Με αυτά τα δύο στοιχεία ο αλγόριθμος που κατασκευάστηκε λειτουργεί ως εξής:

1. Αρχικά υπολογίζουμε την μέση τιμή ( $\bar{a}_n$ ) και την τυπική απόκλιση  $\sigma$  του παραθύρου.
2. Λόγω της πρώτης ιδιότητας του σήματος που αναπτύχθηκε προηγουμένως, κατά τους ήχους S1 S2 θα έχουμε απόκλιση μεγαλύτερη της τυπικής του παραθύρου. Επομένως εντοπίζουμε τα σημεία  $X_i$  που αντιστοιχούν σε αυτούς, ψάχνοντας για σημεία για τα οποία

ισχύει:  $|PCG(X_i) - av(PCG)| > \lambda * \sigma$  όπου  $\sigma$  μια κατάλληλα επιλεγμένη παράμετρος μεγαλύτερη της μονάδας.



Εικόνα 4.4 : Αλγόριθμος εύρεσης σημείων ενδιαφέροντος στο PCG σήμα

3. Το παραπάνω κριτήριο θα εντοπίσει ένα αριθμό σημείων μόνο πάνω στους ήχους S1, S2. Ο εντοπισμός της “αρχής” του σήματος S1 θα γίνει χρησιμοποιώντας τη δεύτερη ιδιότητα από αυτές που αναλύθηκαν παραπάνω δηλαδή το γεγονός πως το μεγαλύτερο κενό ανάμεσα σε δύο ήχους, είναι αυτό ανάμεσα από τον ήχο S2 και τον αμέσως επόμενο S1. Επομένως εντοπίζοντας το μεγαλύτερο κενό στα εντοπισμένα σημεία, γνωρίζουμε πως τη αρχή του ήχου S1 θα είναι στο πρώτο σημείο που υπακούει την συνθήκη που αναφέραμε στο (2) αμέσως μετά από αυτό. Όμοια, μπορεί να βρεθεί ο ήχος S1, ενός δεύτερου παλμού.

4. Καθότι η εξεύρεση των σημείων ενδιαφέροντος σε αυτό το σήμα με ακρίβεια είναι περισσότερο δύσκολη, πριν χρησιμοποιηθεί το αποτέλεσμα που προκύπτει από αυτόν τον αλγόριθμο, θα περάσει από έναν τελευταίο έλεγχο: Από δύο διαδοχικές κορυφές S1 και τον μεταξύ τους χρόνο  $\Delta t$  του PCG θα εκτιμηθεί ο καρδιακός ρυθμός ( $HR_{PCG} = 60 / \Delta t$  beats/min). Αυτός ο ρυθμός θα συγκριθεί με τον ρυθμό που εκτιμάται από τον PPG αλγόριθμο (χρησιμοποιώντας τον χρόνο ανάμεσα σε δύο κορυφές). Στην περίπτωση που απέχουν σημαντικά, απορρίπτουμε τις μετρήσεις και επεξεργαζόμαστε το επόμενο παράθυρο.

Την ανάπτυξη των παραπάνω αλγορίθμων στο περιβάλλον του Matlab, ακολούθησε η υλοποίησή τους σε γλώσσα C στον μικροελεγκτή. Στο συγκεκριμένο σημείο αξίζει να σημειωθεί ένα χρήσιμο χαρακτηριστικό του STM32F401 που επιλέχθηκε. Στο εσωτερικό του, βρίσκεται ένας επεξεργαστής ARM CORTEX-M4, ο οποίος διαθέτει FPU (Floating Point Unit) και DSP Controller Unit, τα οποία επεκτείνουν τις εντολές Assembly που μπορεί να εκτελέσει ο επεξεργαστής με επιπλέον, ειδικά για την ταχύτερη εκτέλεση κώδικα επεξεργασίας σήματος και πράξεων με floating point αριθμούς. Χρησιμοποιώντας ειδικά σχεδιασμένες για τον επεξεργαστή μας pre-compiled βιβλιοθήκες επιτυγχάνουμε ταχύτερη εκτέλεση των DSP αλγορίθμων μας. Συγκεκριμένα η βιβλιοθήκη που χρησιμοποιήθηκε είναι η CMSIS-DSP η οποία διαθέτει βελτιστοποιημένες συναρτήσεις εύρεσης μεγίστου/ελαχίστου, μέσου όρου, τυπικής απόκλισης κλπ.



## 4.2 Βαθμονόμηση και Αποτελέσματα

Η συσκευή σε αυτό το σημείο είναι έτοιμη να υπολογίσει το PATp. Για να είναι σε θέση να εκτιμήσει την πίεση του αίματος, όπως ειπώθηκε σε προηγούμενο κεφάλαιο, θα πρέπει να επιλεγεί ένα παραμετρικό μοντέλο  $BP(PAT)=BP$  το οποίο να προσαρμοστεί σε έναν συγκεκριμένο ασθενή.

Θα χρησιμοποιήσουμε το μοντέλο του αντιστρόφου τετραγώνου για την εκτίμηση της συστολικής πίεσης, καθώς στη βιβλιογραφία φαίνεται να έχει καλά αποτελέσματα.

$$BP=a/PAT^2+b .$$

Οι μετρήσεις του PAT έγιναν με την χρήση της συσκευής που κατασκευάσαμε, με τον αισθητήρα PPG στον δείκτη του αριστερού χεριού. Οι μετρήσεις συστολικής πίεσης έγιναν με το πιεσόμετρο μπράτσου του εμπορίου, OMRON M6 comfort, σύμφωνα με τις οδηγίες που παρέχει το φύλλο οδηγιών του. Οι μετρήσεις με τις δύο συσκευές γίνονταν διαδοχικά πρώτα με τη χρήση του πιεσόμετρου και στη συνέχεια με την συσκευή που κατασκευάσαμε. Έγινε προσπάθεια οι δύο μετρήσεις να διαφέρουν χρονικά μόνο λίγα δευτερόλεπτα.

Ένας εύκολος τρόπος να αυξηθεί προσωρινά η συστολική τιμή της αρτηριακής πίεσης για να βρεθούν τιμές που να απέχουν μεταξύ τους ώστε να μπορεί να κατασκευαστεί η καμπύλη SBP(PAT) είναι λίγα λεπτά έντονης άσκησης.

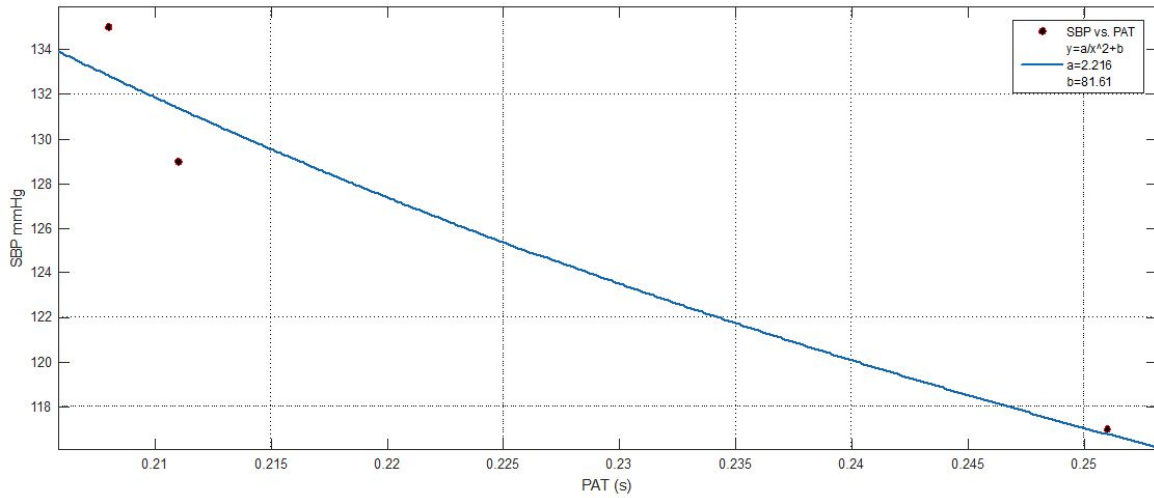
Ηλικία	Ύψος	Βάρος	Φύλο	Καρδιαγγειακές παθήσεις
25 ετών	1.78 m	65 kg	Άρρεν	Όχι

Πίνακας 4.1 : Χαρακτηριστικά ασθενή στον οποίο χρησιμοποιείται η συσκευή

SBP (mmHg)	PATp (ms)
117	251
129	211
135	203

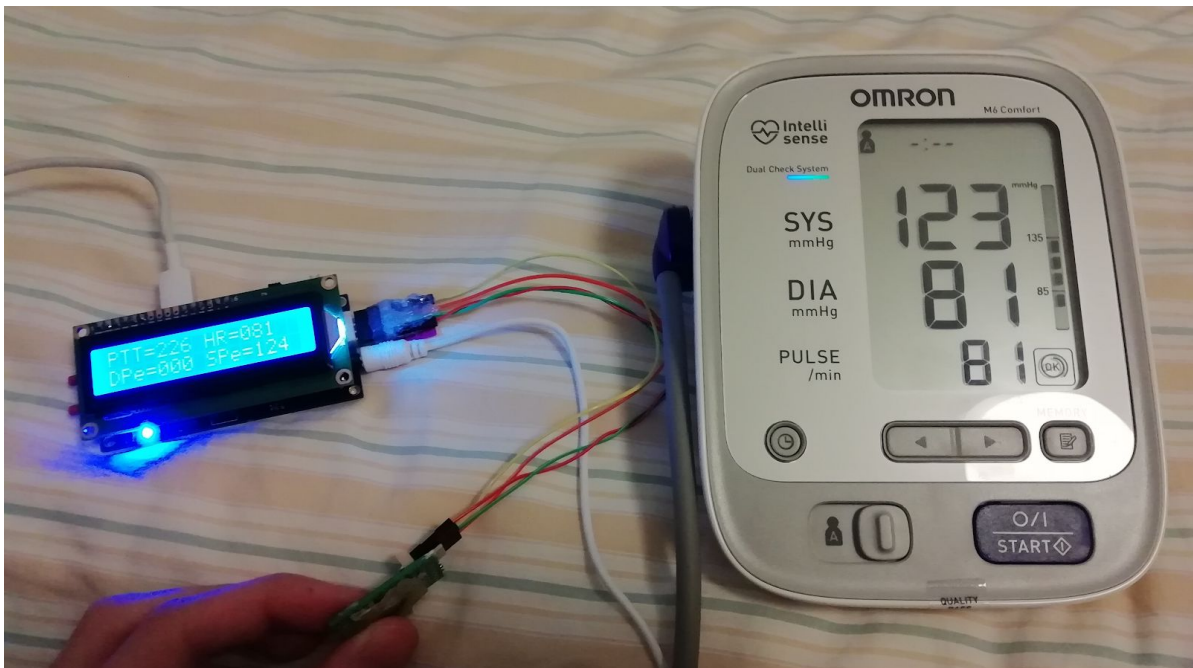
Πίνακας 4.2 : Μετρήσεις SBP και PATp στον ασθενή του πίνακα 4.1

Η καμπύλη  $y=2.2/PAT^2+82$  προσαρμόζεται βέλτιστα στα 3 σημεία. Η καμπύλη αυτή βρέθηκε με χρήση του Curve Fitting tool του Matlab.



Εικόνα 4.5 : Γράφημα προσαρμοσμένης στον ασθενή καμπύλης SBP(PAT) σύμφωνα με τα αποτελέσματα του πίνακα 4.2

Αυτή η συνάρτηση μεταφέρεται λοιπόν στον κώδικα της συσκευής και είναι έτοιμη να παρέχει εκτιμήσεις για την συστολική πίεση του αίματος.



Πίνακας 4.6 : Παράδειγμα χρήσης της (βαθμονομημένης) συσκευής. Στην οθόνη της διακρίνονται : HR (heart rate)=81(beats/min). SPe(Systolic Pressure estim.)=124(mmHg), PTT= 226 (ms)

## 5. Επεκτάσεις

Στο συγκεκριμένο κεφάλαιο, θα αναφερθούμε σε επεκτάσεις που μπορούν να γίνουν στην συγκεκριμένη εργασία. Αρχικά θα παρουσιάσουμε πιθανή ανάπτυξη σε ερευνητικό επίπεδο και έπειτα, το πως θα μπορούσε η συγκεκριμένη συσκευή να εξελιχθεί σε εμπορικό προϊόν.

Σε ερευνητικό επίπεδο, η συγκεκριμένη συσκευή, είναι μια βολική πλατφόρμα για τη μελέτη της συσχέτισης BP και PAT. Μπορεί να συσχετιστεί η συστολική, διαστολική και μέση πίεση με το PAT, καθώς επίσης και να δοκιμαστούν διαφορετικά σημεία ενδιαφέροντος στα PCG και PPG σήματα, από τα οποία αυτό θα υπολογίζεται (π.χ. κορυφή ή το χαμηλότερο σημείο ακριβώς πριν από μια κορυφή για το PPG). Επιπλέον, θα μπορούσε κανείς να εξετάσει διαφορετικούς αλγόριθμους αυτόματης εύρεσης των σημείων ενδιαφέροντος από αυτούς που αναπτύχθηκαν στην προηγούμενη ενότητα, πετυχαίνοντας μεγαλύτερη ακρίβεια και επαναληψιμότητα. Τέλος, από την παρούσα εργασία γίνεται εμφανές, πως θα ήταν σχετικά εύκολο να κατασκευαστεί μια μικρή συσκευή, η οποία να παρέχει πληθώρα δυνατοτήτων, αξιοποιώντας πλήρως τα δύο βιοσήματα που χρησιμοποιήσαμε. Με ελάχιστο επιπλέον hardware και επιπλέον λογισμικό, η συσκευή θα μπορούσε να εξελιχθεί σε μια πλατφόρμα ικανή να παρέχει πληροφορίες για πολλές διαφορετικές παραμέτρους του σώματος και να ανιχνεύει πιθανά προβλήματα. Θα ήταν εφικτό, η συσκευή να παρέχει πληροφορίες, εκτός από τον καρδιακό ρυθμό, το PAT και την συστολική και διαστολική πίεση, για τη σχετική οξυγόνωση του αίματος (μέσω δικάναλου PPG), στοιχεία για την αναπνοή και τη λειτουργία της καρδιάς (μέσω του PCG).

Μια συσκευή που θα προοριζόταν για εμπορική χρήση βασισμένη σε αυτήν που κατασκευάσαμε στη συγκεκριμένη εργασία, θα απαιτούσε την ανάπτυξη μερικών επιπλέον δυνατοτήτων για αυτή:

Πρώτα από όλα, για την βαθμονόμηση του οργάνου, μια εφαρμογή για περιβάλλον WINDOWS/macOS θα ήταν απαραίτητη. Η συγκεκριμένη εφαρμογή, θα μπορούσε να κατευθύνει τον χρήστη για τον σωστό χειρισμό της συσκευής και της δεύτερης συσκευής-αναφοράς(π.χ. πιεσόμετρο), να υπολογίζει τις παραμέτρους του μοντέλου  $BP(PIT)=BP$  και τελικά να τους “φορτώνει στη συσκευή. Μια δεύτερη προσέγγιση, για την βαθμονόμηση του οργάνου, θα μπορούσε να αποφεύγει την χρήση υπολογιστή. Αυτό θα απαιτούσε την ανάπτυξη περισσότερων λειτουργιών και καλύτερης διεπαφής χρήστη για τη συσκευή(π.χ. οθόνη αφής).

Δεύτερον, μια μνήμη EEPROM που θα μπορούσε να ενσωματωθεί στη συσκευή, με μικρό κόστος και ελάχιστο χρόνο ανάπτυξης, θα έδινε την δυνατότητα της αποθήκευσης των μετρήσεων, χρήσιμο για παράδειγμα για την διατήρηση ιστορικού των τιμών της πίεσης και του PAT.

Τέλος, ένα πολύ σημαντικό σημείο στο οποίο συνειδητά αποφύγαμε να εστιάσουμε στη συγκεκριμένη εργασία, αλλά πρέπει να αναφέρουμε έστω εδώ, είναι πως η ανάπτυξη μιας ιατρικής συσκευής όπως αυτή, με σκοπό να προωθηθεί στην αγορά, προϋποθέτει την λήψη έγκρισης της από τον φορέα, ο οποίος είναι υπεύθυνος για την υγεία των καταναλωτών στην εκάστοτε χώρα. Για παράδειγμα στις Ηνωμένες Πολιτείες της Αμερικής, ο οργανισμός φαρμάκων και τροφίμων (FDA) κατατάσσει τις συσκευές υπολογισμού πίεσης του αίματος στις συσκευές κλάσης 2 - “μέσου” κινδύνου (class 2-Moderate Risk) και υποχρεώνει συγκεκριμένα στάνταρ για την συσκευή, (τα οποία στόχο έχουν την προστασία της υγείας των καταναλωτών), καθώς επίσης και την παρουσίαση κλινικών μελετών οι οποίες να επιβεβαιώνουν πως αυτή λειτουργεί, όπως οι κατασκευαστές της διατείνονται. Η σχεδίαση λοιπόν μιας συσκευής μέτρησης πίεσης, όπως αυτή που σχεδιάσαμε εδώ, πιθανώς να διέφερε σε αρκετά σημεία, ώστε να ακολουθεί τους κανόνες που έχουν τεθεί από τους αρμόδιους οργανισμούς, με σκοπό την προστασία των χρηστών της. Παρατάτα, καθώς η συσκευή τροφοδοτείται είτε από μπαταρία, είτε από USB και η χρήση της δεν περιλαμβάνει την σύνδεση του χρήστη με οποιοδήποτε αγωγίμο αισθητήρα, (όπως στην περίπτωση που είχαμε χρησιμοποιήσει ECG), είναι γενικά ασφαλής σίγουρα για ερευνητική τουλάχιστον χρήση ακόμα και στον παρόντα σχεδιασμό της.

# Παράρτημα Α - Code

Εδώ, θα παρουσιάσουμε τα κύρια μέρη του κώδικα για τον μικροελεγκτή (1-6) και τις 3 συναρτήσεις Matlab που κατασκευάστηκαν για την λήψη των σημάτων από τη συσκευή, μέσω USB, την απεικόνιση και την επεξεργασία τους (7-9)

1.main.c 2.ptt.h 3.ptt.c 4.hd44780.h 5.hd44780.c 6.config.h  
7.serial\_com\_debug.m 8.peak\_ppg.m 9.sl\_pvg\_v2.m

1)

```
#include "main.h"
```

```
#include "usb_device.h"
```

```
#include "usbd_cdc_if.h"
```

```
#include "config.h"
```

```
#include "hd44780.h"
```

```
#include "ptt.h"
```

```
ADC_HandleTypeDef hadc1,hadc2;
```

```
DMA_HandleTypeDef hdma_adc1;
```

```
TIM_HandleTypeDef htim2;
```

```
TIM_HandleTypeDef htim3;
```

```
TIM_HandleTypeDef htim4;
```

```
TIM_HandleTypeDef htim11;
```

```
uint32_t adc1_v[ADC_SAMPLES];
```

```
int16_t temp1[ADC_SAMPLES],temp2[ADC_SAMPLES],hr1,hr2,ptt,sbp;
```

```
uint8_t volatile data_ready=0,
```

```
device_state=INIT,
```

```
heart_rate_origin=0;
```

```
/* Private function prototypes -----*/
```

```
void SystemClock_Config(void);
```

```
static void MX_GPIO_Init(void);
```

```
static void MX_DMA_Init(void);
```

```
static void MX_ADC1_Init(void);
```

```
static void MX_ADC1_Init2(void);
```

```
static void MX_TIM4_Init(void);
```

```
static void MX_TIM2_Init(void);
```

```
static void MX_TIM3_Init(void);
```

```
static void MX_TIM11_Init(void);
```

```
void HAL_TIM_MspPostInit(TIM_HandleTypeDef *htim);
```

```
static void fix_led(void);
```

```
static void fix_contrast(void);
```

```
int main(void)
```

```

{

HAL_Init();

SystemClock_Config();

MX_GPIO_Init();
MX_DMA_Init();
MX_USB_DEVICE_Init();
MX_TIM4_Init(); // INDICATION led
MX_TIM2_Init(); // PPG Led
MX_TIM3_Init(); // ADC trigger
MX_TIM11_Init(); // contrast
__HAL_TIM_ENABLE(&htim4);
__HAL_TIM_ENABLE(&htim3);
__HAL_TIM_ENABLE(&htim2);
__HAL_TIM_ENABLE(&htim11);

HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_1);

fix_contrast();
hd44780_Init();
HAL_Delay(1000);
    while (1) {
        if (device_state == INIT) {
            HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1); // control of the led
            fix_led();
            device_state = BPOUT;
            MX_ADC1_Init(); // parameters for bp monitoring #hadc1
            HAL_ADC_Start_DMA(&hadc1, (uint32_t*) adc1_v, 2 *
ADC_SAMPLES);
        }
        if (device_state == BPOUT) {
            if (data_ready == 1) {
                data_ready = 0;

#if debug==1

                uint8_t *a=(uint8_t*)adc1_v;
                CDC_Transmit_FS(a,4*ADC_SAMPLES);

#endif

                //
                uint16_t *b=(uint16_t*)adc1_v;
                uint16_t k=0;
                for (int i=0;i<2*ADC_SAMPLES;i=i+2){
                    temp1[k]=(uint16_t)*(b+i); //pcg
                    temp2[k]=(uint16_t)*(b+i+1); //ppg
                    k++;
                }
                uint16_t ppgpeaks[4]={0,0,0,0},pcgpeaks[4]={0,0,0,0};

```

```

        ppg_peak( &temp2[0], &ppgpeaks[0]);
        hr1=heart_rate(&ppgpeaks[0]);
        sound_peak( &temp1[0], &pcgpeaks[0]);
        hr2=heart_rate(&pcgpeaks[0]);
        if (heart_rate_origin){
            if (hr1>0) write_hr(hr1);
            else write_hr(0);
        }
        else{
            if (hr2>0) write_hr(hr2);
            else write_hr(0);
        }
        ptt=ptt_estimate(&ppgpeaks[0],&pcgpeaks[0]);
        if(ptt>0) write_ptt(ptt);
        int16_t ptt=ptt_estimate(&ppgpeaks[0],&pcgpeaks[0]);
        sbp=sbp_estimate(ptt);
        if(sbp>0) write_spe(sbp);

        MX_ADC1_Init();
        HAL_ADC_Start_DMA(&hadc1, (uint32_t*) adc1_v, 2 *
ADC_SAMPLES);
    }
    }
    HAL_Delay(20);
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /**Configure the main internal regulator output voltage
    */
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
    /**Initializes the CPU, AHB and APB busses clocks
    */
    //RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    //RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    //RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;

```

```

RCC_OscInitStruct.PLL.PLLM = 16;
RCC_OscInitStruct.PLL.PLLN = 336;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV4;
RCC_OscInitStruct.PLL.PLLQ = 7;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}
/**Initializes the CPU, AHB and APB busses clocks
*/
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSClkSource = RCC_SYSClkSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSClk_DIV2;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief ADC1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_ADC1_Init(void)
{
    /* USER CODE BEGIN ADC1_Init 0 */

    /* USER CODE END ADC1_Init 0 */

    ADC_ChannelConfTypeDef sConfig = {0};

    /* USER CODE BEGIN ADC1_Init 1 */

    /* USER CODE END ADC1_Init 1 */
    /**Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of
conversion)
*/
    hadc1.Instance = ADC1;
    hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV2;
    hadc1.Init.Resolution = ADC_RESOLUTION_12B;
    hadc1.Init.ScanConvMode = ENABLE;
    hadc1.Init.ContinuousConvMode = DISABLE;
    hadc1.Init.DiscontinuousConvMode = ENABLE;

```



```

hadc1.Init.NbrOfDiscConversion = 2;
hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_RISINGFALLING;
hadc1.Init.ExternalTrigConv = ADC_EXTERNALTRIGCONV_T3_TRGO;
hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
hadc1.Init.NbrOfConversion = 2;
hadc1.Init.DMAContinuousRequests = DISABLE;
hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
if (HAL_ADC_Init(&hadc1) != HAL_OK)
{
    Error_Handler();
}
/**Configure for the selected ADC regular channel its corresponding rank in the sequencer and its
sample time.
*/
sConfig.Channel = ADC_CHANNEL_0;
sConfig.Rank = 1;
sConfig.SamplingTime = ADC_SAMPLETIME_480CYCLES;
if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
{
    Error_Handler();
}
/**Configure for the selected ADC regular channel its corresponding rank in the sequencer and its
sample time.
*/
sConfig.Channel = ADC_CHANNEL_1;
sConfig.Rank = 2;
if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN ADC1_Init 2 */

/* USER CODE END ADC1_Init 2 */

}

static void MX_ADC1_Init2(void)
{
    ADC_ChannelConfTypeDef sConfig = {0};

    /**Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of
conversion)
    */
    hadc2.Instance = ADC1;
    hadc2.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
    hadc2.Init.Resolution = ADC_RESOLUTION_12B;
    hadc2.Init.ScanConvMode = DISABLE;
    hadc2.Init.ContinuousConvMode = ENABLE;

```

```

hadc2.Init.DiscontinuousConvMode = DISABLE;
hadc2.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
if (HAL_ADC_Init(&hadc2) != HAL_OK)
{
    Error_Handler();
}

/**Configure for the selected ADC regular channel its corresponding rank in the sequencer and its
sample time.
*/
sConfig.Channel = ADC_CHANNEL_2;
//sConfig.Channel = ADC_CHANNEL_4;
sConfig.Rank = 1;
sConfig.SamplingTime = ADC_SAMPLETIME_480CYCLES;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

}

/**
 * @brief TIM2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM2_Init(void)
{

    /* USER CODE BEGIN TIM2_Init 0 */

    /* USER CODE END TIM2_Init 0 */

    TIM_MasterConfigTypeDef sMasterConfig = {0};
    TIM_OC_InitTypeDef sConfigOC = {0};

    /* USER CODE BEGIN TIM2_Init 1 */

    /* USER CODE END TIM2_Init 1 */
    htim2.Instance = TIM2;
    htim2.Init.Prescaler = (uint32_t) 420000/LED_PWM_FREQ-1;
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 99;
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    if (HAL_TIM_PWM_Init(&htim2) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;

```

```

sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
sConfigOC.OCMode = TIM_OCMode_PWM1;
sConfigOC.Pulse = 10;
sConfigOC.OCpolarity = TIM_OCPolarity_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM2_Init 2 */

/* USER CODE END TIM2_Init 2 */
HAL_TIM_MspPostInit(&htim2);

}

/**
 * @brief TIM3 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM3_Init(void)
{
    /* USER CODE BEGIN TIM3_Init 0 */

    /* USER CODE END TIM3_Init 0 */

    TIM_MasterConfigTypeDef sMasterConfig = {0};
    TIM_OC_InitTypeDef sConfigOC = {0};

    /* USER CODE BEGIN TIM3_Init 1 */

    /* USER CODE END TIM3_Init 1 */
    htim3.Instance = TIM3;
    htim3.Init.Prescaler = (uint32_t) (4200000/ADC_SAMPLE_FREQUENCY/2-1);
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 1;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    if (HAL_TIM_PWM_Init(&htim3) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;

```

```

if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
sConfigOC.OCMode = TIM_OCMode_PWM1;
sConfigOC.Pulse = 0;
sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM3_Init 2 */

/* USER CODE END TIM3_Init 2 */

}

/**
 * @brief TIM4 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM4_Init(void)
{

/* USER CODE BEGIN TIM4_Init 0 */

/* USER CODE END TIM4_Init 0 */

TIM_MasterConfigTypeDef sMasterConfig = {0};
TIM_OC_InitTypeDef sConfigOC = {0};

/* USER CODE BEGIN TIM4_Init 1 */

/* USER CODE END TIM4_Init 1 */
htim4.Instance = TIM4;
htim4.Init.Prescaler = (uint32_t) 42000-1;
htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
htim4.Init.Period = 999;
htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
if (HAL_TIM_PWM_Init(&htim4) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim4, &sMasterConfig) != HAL_OK)
{

```

```

    Error_Handler();
}
sConfigOC.OCMode = TIM_OCMode_PWM1;
sConfigOC.Pulse = 500;
sConfigOC.OCpolarity = TIM_OCPolarity_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
if (HAL_TIM_PWM_ConfigChannel(&htim4, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM4_Init 2 */

/* USER CODE END TIM4_Init 2 */
HAL_TIM_MspPostInit(&htim4);

}
static void MX_TIM11_Init(void)
{
    /* USER CODE BEGIN TIM2_Init 0 */

    /* USER CODE END TIM2_Init 0 */

    TIM_OC_InitTypeDef sConfigOC = {0};

    /* USER CODE BEGIN TIM2_Init 1 */

    /* USER CODE END TIM2_Init 1 */
    htim11.Instance = TIM11;
    htim11.Init.Prescaler = 209; // 420.000/x HZ -1;
    htim11.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim11.Init.Period = 99;
    htim11.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    if (HAL_TIM_PWM_Init(&htim11) != HAL_OK)
    {
        Error_Handler();    }
    sConfigOC.OCMode = TIM_OCMode_PWM1;
    sConfigOC.Pulse = 1;
    sConfigOC.OCpolarity = TIM_OCPolarity_HIGH;
    sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
    if (HAL_TIM_PWM_ConfigChannel(&htim11, &sConfigOC, TIM_CHANNEL_1) !=
HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM2_Init 2 */

    /* USER CODE END TIM2_Init 2 */
    HAL_TIM_MspPostInit(&htim11);

}

```

```

/**
 * Enable DMA controller clock
 */
static void MX_DMA_Init(void)
{
    /* DMA controller clock enable */
    __HAL_RCC_DMA2_CLK_ENABLE();

    /* DMA interrupt init */
    /* DMA2_Stream0_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA2_Stream0_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA2_Stream0_IRQn);
}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOC, DB7_Pin|DB6_Pin|DB5_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOA, DB4_Pin|E_Pin|RS_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pins : DB7_Pin DB6_Pin DB5_Pin */
    GPIO_InitStruct.Pin = DB7_Pin|DB6_Pin|DB5_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    /*Configure GPIO pins : DB4_Pin E_Pin RS_Pin */
    GPIO_InitStruct.Pin = DB4_Pin|E_Pin|RS_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;

```

```

HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pin : user_button_Pin */
GPIO_InitStruct.Pin = user_button_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(user_button_GPIO_Port, &GPIO_InitStruct);
HAL_NVIC_SetPriority(EXTI2_IRQn, 3, 0);
HAL_NVIC_EnableIRQ(EXTI2_IRQn);

}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    HAL_ADC_Stop_DMA(hadc);
    data_ready=1;
}

void fix_led(void)
{
    uint16_t volatile command=0,
                goal=3400,
                measurement=0;
    MX_ADC1_Init2(); //parameters for setting up magnitude of led intensity #hadc2
    HAL_ADC_Start(&hadc2);
    while(measurement<goal&&command<95){
        command=command+5;
        __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, command);
        HAL_ADC_PollForConversion(&hadc2, 100);
        measurement = HAL_ADC_GetValue(&hadc2);
        HAL_Delay(400);
    }
    HAL_ADC_Stop(&hadc2);
    HAL_Delay(3000);
}

void fix_contrast(void)
{
    int st=1;
    HAL_TIM_PWM_Start(&htim11, TIM_CHANNEL_1); //control of the led
    while(st<contrast){
        st++;
        __HAL_TIM_SET_COMPARE(&htim11, TIM_CHANNEL_1, st);
        HAL_Delay(60);
    }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{

```

```

/* Prevent unused argument(s) compilation warning */
UNUSED(GPIO_Pin);
heart_rate_origin=~heart_rate_origin;
/* NOTE: This function Should not be modified, when the callback is needed,
the HAL_GPIO_EXTI_Callback could be implemented in the user file
*/
}

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
/* USER CODE BEGIN Error_Handler_Debug */
/* User can add his own implementation to report the HAL error return state */
//int volatile i;
//i=1;
//i=2;
/* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
/* USER CODE BEGIN 6 */
/* User can add his own implementation to report the file name and line number,
tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
/* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

2)
#ifndef PTT_H_
#define PTT_H_

#include "stm32f4xx_hal.h"

void ppg_peak(int16_t*,uint16_t*);

```



```

void sound_peak(int16_t*,uint16_t*);
int heart_rate(uint16_t*); //hb/min out
int16_t ptt_estimate(uint16_t*,uint16_t*); // ms out
int16_t sbp_estimate(int16_t); //ms in , mmHg out
#endif /* PTT_H_ */

```

3)

```

/*
 * ptt.c
 *
 * Created on: Jan 6, 2019
 * Author: Angelos
 */
#include "ptt.h"
#include "arm_math.h"
#include "config.h"
#include "stdlib.h"

```

```

void ppg_peak(int16_t* in,uint16_t* out)
{
    int16_t max_mag,min_mag;
    uint32_t max_where,min_where,k=0;

    arm_min_q15(in,ADC_SAMPLES,&min_mag,&min_where);
    arm_max_q15(in,ADC_SAMPLES,&max_mag,&max_where);
    *(out)=max_where;
    int16_t low_lim=min_mag+(int16_t)(0.8*(float)(max_mag-min_mag));
    int i=1;
    do {
        for(k=0;k<700;k++){
            if ((max_where-350+k)>=0&&(max_where-350+k)<ADC_SAMPLES)
*(in+max_where-350+k)=0;
        }
        arm_max_q15(in,ADC_SAMPLES,&max_mag,&max_where);
        if (max_mag>low_lim) *(out+i)=max_where;
        if (max_where!=0&&max_where!=ADC_SAMPLES-1) i++;
    }while((max_mag>0.7*((*out)-min_mag))&&i<4);
}

```

```

void sound_peak(int16_t* in,uint16_t* out)
{
    #define magic 2
    int16_t sd,aver,help;
    int32_t acc=0;
    arm_mean_q15(in,ADC_SAMPLES,&aver);
    for(int i=0;i<=ADC_SAMPLES;i++){

```

```

        acc=acc+*(in+i)-aver)*(*(in+i)-aver);
    }
    acc=acc/ADC_SAMPLES;
    sd=(int16_t)(magic*sqrt((double)acc));
    int j=0,k=0;
    for(int i=0;i<ADC_SAMPLES;i++){
        help=*(in+i)-aver;
        if (abs(help)<sd||help>0){
            k++;}
        else{
            if (k>600&&j<4){
                *(out+j)=i;
                j++;
            }
            k=0;
        }
    }
}

int16_t ptt_estimate(uint16_t* in1,uint16_t* in2)
{
    int16_t place1=0,place2=0,help;
    for(int i=0;i<4;i++){
        if (*(in1+i)>place1){
            place1=*(in1+i);
        }
    }
    for(int i=0;i<4;i++){
        help=place1-*(in2+i);
        if ((help>400)&&(help<600)){
            place2=*(in2+i);
        }
    }
    if (place2!=0) return (place1-place2)*1000/ADC_SAMPLE_FREQUENCY;
    else return -1;
}

int heart_rate(uint16_t* in)
{
    int16_t x[4];
    for (int i=0;i<4;i++){
        x[i]=(int16_t)*(in+i);
    }
    int16_t max_mag1,max_mag2;
    uint32_t max_where;
    arm_max_q15(&x[0],4,&max_mag1,&max_where);
    x[max_where]=0;
    arm_max_q15(&x[0],4,&max_mag2,&max_where);
    int calc=60*ADC_SAMPLE_FREQUENCY/(max_mag1-max_mag2);
    if (calc>60 && calc<100) {

```

```

        return calc;
    }
    else {
        return -1;
    }
}

int16_t sbp_estimate(int16_t in)
{
    //a=2.2 b=82
    if (in!=0) return 2200000/(in*in)+82;
    else return -1;
}

```

4)

```

#ifndef __HD44780_H
#define __HD44780_H

```

```

#include "config.h"
#include "stm32f4xx_hal.h"

```

```

typedef struct pin_t

```

```

{
    GPIO_TypeDef* port;
    uint16_t pin;
}pin_t;

```

```

int hd44780_Init(void);
void write_dpe(int dpe);
void write_spe(int spe);
void write_hr(int hr);
void write_ptt(int ptt);
#endif

```

5)

```

#include "hd44780.h"

```

```

const pin_t rs = {GPIOA, GPIO_PIN_10};
const pin_t en = {GPIOA, GPIO_PIN_9};
const pin_t db[] = {{GPIOA, GPIO_PIN_8}, {GPIOC, GPIO_PIN_9}, {GPIOC, GPIO_PIN_8},
{GPIOC, GPIO_PIN_7}};

```

```

/* Private interface */

```

```

static void commandWrite(int cmd);
static void senable(void);
static void hd44780_Clear(void);
static void hd44780_Home(void);
static int hd44780_dataWrite(unsigned int data);

```

```

static int hd44780_integerWrite(int data);
static void operation_begin(void);
#define delay(x) HAL_Delay(x)
/**
 * Public interface
 */

void operation_begin(void)
{
    hd44780_Clear();
    hd44780_Home();

    hd44780_dataWrite('P');hd44780_dataWrite('T');hd44780_dataWrite('T');hd44780_dataWrite('=');

    hd44780_dataWrite('x');hd44780_dataWrite('x');hd44780_dataWrite('x');hd44780_dataWrite(' ');
    hd44780_dataWrite('H');hd44780_dataWrite('R');hd44780_dataWrite('=');
    hd44780_dataWrite('x');hd44780_dataWrite('x');hd44780_dataWrite('x');
    commandWrite(192);

    hd44780_dataWrite('D');hd44780_dataWrite('P');hd44780_dataWrite('e');hd44780_dataWrite('=');

    hd44780_dataWrite('x');hd44780_dataWrite('x');hd44780_dataWrite('x');hd44780_dataWrite(' ');

    hd44780_dataWrite('S');hd44780_dataWrite('P');hd44780_dataWrite('e');hd44780_dataWrite('=');

    hd44780_dataWrite('x');hd44780_dataWrite('x');hd44780_dataWrite('x');hd44780_dataWrite(' ');
}
void write_dpe(int hr)
{
    commandWrite(196);
    hd44780_dataWrite(hr/100 +'0');
    hd44780_dataWrite((hr%100)/10+'0');
    hd44780_dataWrite((hr%100)%10 +'0');
}
void write_spe(int hr)
{
    commandWrite(204);
    hd44780_dataWrite(hr/100 +'0');
    hd44780_dataWrite((hr%100)/10+'0');
    hd44780_dataWrite((hr%100)%10 +'0');
}
void write_ptt(int hr)
{
    commandWrite(132);
    hd44780_dataWrite(hr/100 +'0');
    hd44780_dataWrite((hr%100)/10+'0');
    hd44780_dataWrite((hr%100)%10 +'0');
}
void write_hr(int hr)

```

```

{
    commandWrite(139);
    hd44780_dataWrite(hr/100+'0');
    hd44780_dataWrite((hr%100)/10+'0');
    hd44780_dataWrite((hr%100)%10+'0');
}

```

```

int hd44780_Init(void)

```

```

{
    /* Initialize display */

    commandWrite(0x3);
    delay(64);
    commandWrite(0x3);
    delay(50);
    commandWrite(0x3);
    delay(20);
    commandWrite(2);
    delay(20);
    commandWrite(2);
    delay(20);
    commandWrite(1);
    delay(20);
    commandWrite(1);
    delay(20);
    commandWrite(4);
    delay(20);

    /* Set Contrast lowest bits */
    commandWrite(7);
    commandWrite(1);
    delay(20);

    /* Set Contrast highest bits */
    commandWrite(5);
    commandWrite(0x5);
    delay(20);

    /* Power on */
    commandWrite(6);
    delay(20);
    commandWrite(0xa);
    delay(20);
    commandWrite(0);
    delay(100);
    commandWrite(0xc);
    delay(100);
    commandWrite(0);
    delay(100);
}

```

```

commandWrite(1);
delay(100);
commandWrite(0);
delay(100);
commandWrite(6);
delay(20);

hd44780_Clear();
hd44780_Home();

hd44780_dataWrite('T');
hd44780_dataWrite('N');
hd44780_dataWrite('T');
hd44780_dataWrite('T');
hd44780_dataWrite('.');
delay(1000);
hd44780_dataWrite('.');
delay(1000);
hd44780_dataWrite('.');
delay(100);
operation_begin();
return 0;
}

void hd44780_Clear(void)
{
    commandWrite(0x00);
    commandWrite(0x01);
}

void hd44780_Home(void)
{
    commandWrite(0x00);
    commandWrite(0x02);
}

int hd44780_dataWrite(unsigned int data)
{
    int i;
    uint8_t hinibble, lownibble;

    HAL_GPIO_WritePin(rs.port, rs.pin, 1);

    hinibble = data >> 4;
    lownibble = data & 0xf;

```

```

for(i=0; i<4; i++){
    if(hinibble & 1)
        HAL_GPIO_WritePin(db[i].port, db[i].pin,1);
    else
        HAL_GPIO_WritePin(db[i].port, db[i].pin,0);

    hinibble >>= 1;
}

senable();

for(i=0; i<4; i++){
    if(lownibble & 1)
        HAL_GPIO_WritePin(db[i].port, db[i].pin,1);
    else
        HAL_GPIO_WritePin(db[i].port, db[i].pin,0);

    lownibble >>= 1;
}

senable();

return 0;
}

int hd44780_integerWrite(int data)
{
    int divisor = 10000000;

    while(divisor){
        hd44780_dataWrite('0' + ((data/divisor) % 10));
        divisor /= 10;
    }

    return 0;
}

/** Private interface */

static void commandWrite(int cmd)
{
    int i,t;
    t=cmd>>4;
    HAL_GPIO_WritePin(rs.port,rs.pin,0);
    for(i=0; i<4; i++){
        if(t & 1)
            HAL_GPIO_WritePin(db[i].port,db[i].pin,1);
    }
}

```

```

        else
            HAL_GPIO_WritePin(db[i].port,db[i].pin,0);

            t >>= 1;
        }
        senable();
        HAL_GPIO_WritePin(rs.port,rs.pin,0);
        for(i=0; i<4; i++){
            if(cmd & 1)
                HAL_GPIO_WritePin(db[i].port,db[i].pin,1);

            else
                HAL_GPIO_WritePin(db[i].port,db[i].pin,0);

            cmd >>= 1;
        }
        senable();
    }
static void senable(void)
{
    HAL_GPIO_WritePin(en.port, en.pin,1);
    delay(1);
    HAL_GPIO_WritePin(en.port, en.pin,0);
    delay(1);
    HAL_GPIO_WritePin(en.port, en.pin,1);
    delay(1);
}

6)
/*
 * config.h
 *
 * Created on: Dec 26, 2018
 * Author: Angelos
 */

#ifndef CONFIG_H_
#define CONFIG_H_

#define ADC_SAMPLE_FREQUENCY 2000 // in Samples/sec
#define ADC_SAMPLES 4000 // per channel
#define LED_PWM_FREQ 5000
#define contrast 35
#define debug 1 //debug yes-no 1-0

enum State{
    INIT,BPOUT,CALIBRATE

```



```

};
#endif /* CONFIG_H_ */

7)
figure;
instrreset
while true
s = serial('COM23');
s.InputBufferSize = 16000;
fopen(s);
c1=zeros(4000,1);
c2=zeros(4000,1);
a=fread(s,16000);
k=1;
for i=1:4:16000
c1(k)=a(i+1)*2^8+a(i);
c2(k)=a(i+3)*2^8+a(i+2);
k=k+1;
end
pcg_points=s1_pcg_v2(c1);%
ppg_points=peak_ppg(c2);
%[hr_pcg,hr_ppg,ptt]=proc(pcg_points,ppg_points);
clf
subplot(2,1,1);
hold on
plot(c1);
plot(pcg_points,c1(pcg_points),'x','MarkerSize',10);%
hold off
subplot(2,1,2);
hold on
plot(c2);
plot(ppg_points,c2(ppg_points),'x','MarkerSize',10);%
hold off
drawnow
k=1;
fclose(s)
delete(s)
clear s
end

8)
function peak = peak_ppg(in)
k=1;
for i=1:10:3990
    x(k)=sum(in(i:i+10))/10;

```

```

    k=k+1;
end
clear pot peak
[maxi,peak(1)]=max(x);
z=min(x);
katw=z+0.7*(maxi-z);
k=0;
for i=1:399
    if x(i)>=katw && (i<peak(1)-20 || i>peak(1)+20)
        k=k+1;
        pot(k)=i;
    end
end
i=1;maxo=0;f=2;
while i<k-1
    while pot(i+1)==(pot(i)+1) && i<k-1
        if x(pot(i))>maxo
            maxo=x(pot(i));
            maxp=pot(i);
        end
        i=i+1;
    end
    maxo=0;
    if exist('maxp','var') == 1
        peak(f)=maxp;
    end
    f=f+1;
    i=i+1;
end
peak=peak*10+1;

```

9)

```

function s1 = s1_pcg_v2(signal)
s1(1)=1;
t1=mean(signal);
t2=2*std(signal);
j=0;i=1;k=0;
while i<4000
    if abs(signal(i)-t1)<t2
        k=k+1;
    else
        if k>550
            j=j+1;
            s1(j)=i;

```

```
    end  
    k=0;  
  end  
  i=i+1;  
end
```

# Παράρτημα Β - Device Specifications

## Power

Powered by USB Vbus or any DC source via the molex 2-pin Connector (3.6-5.5 Volt)

On-Off switch for the DC source. Device always On when powered by USB.

Consumption ~30mA@5V , ~3mA@5V without lcd screen

## Controls

Button Up - Switch Heart Rate display 1.PCG source 2. PPG source

Button Down - System restart

## Programming

Under LCD screen (Remove LCD to access). 5 pin header ST-LINK

## Computer Communication

When connected via USB, Virtual Com is automatically created. Recorded PPG and PCG data are sent in packets of 8bits: first 16bits(2 packets) come from PPG, next ones from PCG and so on. You can use the provided matlab scripts to plot the incoming data , realtime

## LCD screen Readings

1.PTT=xxx : Pulse transit(arrival) time. (ms)                      2.HR=xxx Heart Rate estim.(beats/min)

3.DPe=xxx Diastolic Pressure estim. (mmHg) 4.DPe=xxx Diastolic Pressure estim (mmHg)

## Usage

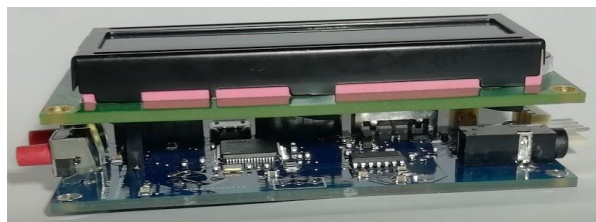
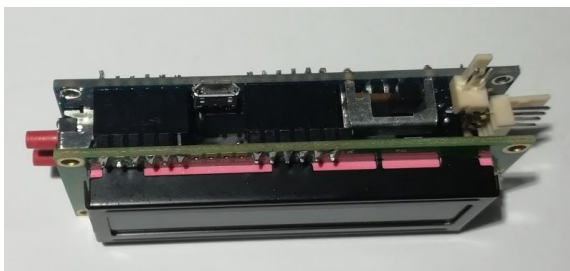
1.Place finger gently on PPG sensor, covering both the photodiode and the IR led

2.Place stethoscope chestpiece on the chest appropriately.

3. Turn on (or press System restart if already in “ON” state) the device.

4. Check Heart rate shown, press Button Down (switch Heart rate estimation source)

Both readings should be the same. That means the sensors are correctly placed and the device will show meaningful PTT reading and if previously calibrated, BP estimation.





# Πηγές

- [1]Yinji Ma, Jungil Choic, et al., “Relation between blood pressure and pulse wave velocity for human arteries”,Proceedings of the National Academy of Sciences Oct 2018, 115 (44) 11144-11149
- [2]Amirhossein Esmaili Dastjerdi et al., “Non-invasive Blood Pressure Estimation Using Phonocardiogram”, IEEE International Symposium on Circuits and Systems (ISCAS) 2017
- [3]Sagar Venkatesh Gubbi,Bharadwaj Amrutur, “Adaptive Pulse Width Control and Sampling for Low Power Pulse Oximetry”, IEEE Transactions on Biomedical Circuits and Systems 9(2) June 2014
- [4]Manuja Sharma 1, Karinne Barbosa et al., “Cuff-Less and Continuous Blood Pressure Monitoring: A Methodological Review”, Technologies 2017, 5(2), 21
- [5]Fukui, Tomohiro & H Parker, Kim & Yamaguchi, Takami. (2012). Pulse wave propagation in large blood vessels based on fluid-solid interactions methods. Single and Two-Phase Flows on Chemical and Biomedical Engineering. 10.2174/978160805295011201010460.
- [6]Aleksandra Zienkiewicz, “Blood Pressure Estimation Using Pulse Transit Time Models”, Master’s Thesis Degree Programme in Computer Science and Engineering November 2017
- [7]MRadha1,GZhang, et al., “Arterial path selection to measure pulse wave velocity as a surrogate marker of blood pressure”, Biomedical Physics & Engineering Express, Volume 3, Number 1 2017
- [8]AN5012 Application Note, ST Microelectronics, “Analog-to-digital audio conversion example using STM32L4 Series microcontroller peripherals”
- [9]STM32F401RE Reference Manual
- [10]Yashaswini B S & Satyanarayana B S, “ The Design of An Electronic Stethoscope-Review”, ternational Conference on Computer Science and Informatics (ICCSI) – Hyderabad - 9 th March, 2012
- [11]AN2834 Application note, ST Microelectronics, “How to get the best ADC accuracy in STM32 microcontrollers”
- [12]Gesche, H., Grosskurth, D., K uchler, G. et al. Eur J Appl Physiol (2012) 112: 309. <https://doi.org/10.1007/s00421-011-1983-3>

