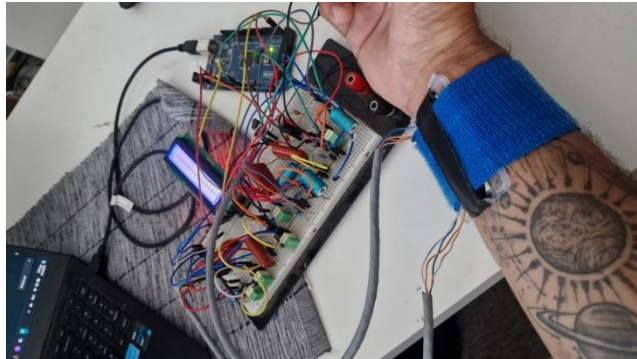


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη συσκευής μέτρησης της αρτηριακής πίεσης
με τη μέθοδο της μαγνητο-πληθυσμογραφίας»



Του φοιτητή
Λύρατση Πασχάλη
Αρ. Μητρώου: 515322

Επιβλέπων
Όνοματεπώνυμο: Κιοσκερίδης
Ιορδάνης
Βαθμίδα: Καθηγητής

Ημερομηνία: 10/5/2024

Τίτλος Π.Ε. «Ανάπτυξη συσκευής μέτρησης της αρτηριακής πίεσης με τη μέθοδο της μαγνητο-
πληθυσμογραφίας»

Κωδικός Π.Ε. 21118

Όνοματεπώνυμο φοιτητή Λύρατζης Πασχάλης
Όνοματεπώνυμο εισηγητή Κιοσκερίδης Ιορδάνης

Ημερομηνία ανάληψης Π.Ε. 26/10/2021

Ημερομηνία περάτωσης Π.Ε. 10/5/2024

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

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

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

Πρόλογος

Η παρούσα πτυχιακή επιλέχθηκε για τον λόγο ότι χρησιμοποιούνται διάφοροι τομείς της ηλεκτρονικής για την σχεδίαση της κατασκευής. Λόγω αυτού και καθώς θα αποκτούνταν γνώσεις στο κομμάτι του hardware όπως των αισθητήρων και του κυκλώματος conditioning τους, των φίλτρων και συνδεσμολογιών τους όπως και στο κομμάτι του software με τον προγραμματισμό πάνω σε arduino και την επεξεργασία δεδομένων που αποκτούνται από το αναλογικό κομμάτι σε real time εφαρμογες.

Περίληψη

Η παρούσα πτυχιακή έχει ως σκοπό την σχεδίαση και κατασκευή ενός πιεσόμετρου με την μέθοδο της μαγνητο-πλυθησμογραφίας. Αποτελείται από το αναλογικό και το ψηφιακό κομμάτι. Στο αναλογικό κομμάτι υπάρχουν δύο αισθητήρες GMR (giant magnetoresistance) οι οποίοι χρησιμοποιούνται με μαγνήτες σε υπολογισμένη απόσταση για να λειτουργούν στην γραμμική περιοχή τους και παίρνουν σήμα από την κερκιδική αρτηρία, το σήμα αυτό περνάει από έναν ενισχυτή οργανολογίας ώστε να ενισχυθεί τόσο ώστε να είναι επεξεργάσιμο. Έπειτα μεταφέρεται μέσα από υπερυπερατά και χαμηλοπερατά φίλτρα σε συνδεσμολογία Sallen-Key για απόρριψη θορύβου και διατήρηση της σημαντικής πληροφορίας. Το σήμα από την έξοδο των φίλτρων μεταφέρεται στον μικροελεγκτή Mega 2560 για επεξεργασία μέσω προγράμματος που γράφτηκε με το λογισμικό Arduino. Στην συνέχεια τα αποτελέσματα απεικονίζονται σε μία οθόνη υγρών κρυστάλλων, η οποία εμφανίζει την πίεση του χρήστη της κατασκευής.

«Development of a device for measurement of arterial pressure with the method of magneto plethysmography»

«Paschalis Lyratzis»

Abstract

The purpose of current thesis is the development and build of a sphygmomanometer with the method of magneto plethysmography. It consists of an analog and a digital part. In the analog part there are two GMR sensors which are used with magnets at a calculated distance for them to operate in their linear area and taking data from the laryngeal arteries, the collected signal pass through an instrumentation amplifier for it to amplify, so that it is processable. Then it passes through high pass and low pass filters implemented with Sallen-Key topology with the purpose to reject noise and to maintain the important information. The processed signal from the filters' output imported in the inputs of microcontroller Mega 2560 for processing through the Arduino software via code written in C. In the end the results are depicted in a LCD monitor, in which appeared the arterial pressure of the user of the device.

Περιεχόμενα

Πρόλογος.....	iii
Περίληψη	iv
Abstract	v
Ευχαριστίες	Error! Bookmark not defined.
Περιεχόμενα	vi
Κατάλογος Σχημάτων	Error! Bookmark not defined.
Κατάλογος Πινάκων	x
Συνομογραφίες.....	xi
Κεφάλαιο 1ο: Πιεσόμετρο	12
1.1 Εισαγωγή	12
1.2 Τι είναι η μέθοδος της μαγνητοπληθισμογραφίας.....	12
1.3 Ιστορική αναδρομή μέτρησης αρτηριακής πίεσης	12
1.4 Συστολική, διαστολική και μέση πίεση	5
1.4.1 Τι είναι η υπέρταση	6
1.4.2 Οδηγίες για σωστή μέτρηση της πίεσης.....	7
1.5 Τι είναι η μαγνητική πληθυσμογραφία	7
1.6 Επίλογος.....	7
Κεφάλαιο 2ο: Αναπτυξιακή πλακέτα Arduino	9
2.1 Εισαγωγή	9
2.2 Τι είναι το Arduino	9
2.3 Ανάλυση των υλικών που αποτελείται η πλακέτα Arduino Mega	9
2.3.1 Ο κρύσταλλος 16MHz	10
2.3.2 Η θύρα USB	10
2.3.3 Το Reset button.....	11
2.3.4 Η θύρα τροφοδοσίας της πλακέτας.....	11
2.3.5 Τα Pins ICSP	11
2.3.6 Τα ψηφιακά πινς	12
2.3.7 Τα αναλογικά πινς	13
2.3.8 Τα υπόλοιπα πινς	13
2.3.9 Χαρακτηριστικά	14
2.4 Προγραμματισμός του Arduino	16
2.4.1 Χρησιμοποιώντας το offline IDE	16

2.4.2 Η γλώσσα προγραμματισμού Arduino	17
2.4.3 Βασικές αρχές σύνταξης κώδικα Arduino	18
2.5 Επίλογος.....	19
Κεφάλαιο 3ο: Ανάλυση του κυκλώματος πιεσόμετρου με την μέθοδο της μαγνητοπλυσμογραφίας	20
3.1 Εισαγωγή	20
3.2 Σχηματικό διάγραμμα.....	20
3.3 Μπλοκ διάγραμμα	20
3.4 Αισθητήρας GMR	22
3.5 Συνδεσμολογία ενισχυτή οργανολογίας	24
3.6 Συνδεσμολογία Sallen-Key φίλτρου.....	26
3.7 Ο αλγόριθμος υπολογισμού της αρτηριακή πίεσης.....	30
3.7.1 Υπολογισμός FFT.....	31
3.7.2 Ανάλυση FFT σε κώδικα Arduino και περαιτέρω βήματα	34
3.8 Επίλογος.....	41
Κεφάλαιο 4ο: Ανάλυση προσομοίωσης και κατασκευής του κυκλώματος πιεσόμετρου	43
4.1 Εισαγωγή	43
4.2 Το πρόγραμμα προσομοίωσης κυκλωμάτων Orcad.	43
4.3 Κατασκευή του κυκλώματος.....	48
4.3.1 Σχεδίαση πλακέτας αισθητήρα AA002-02.....	48
4.3.2 Κατασκευή του κυκλώματος πάνω σε ραστερ	51
4.3.3 Επεξεργασία των δεδομένων της κατασκευής μέσω Arduino Mega	58
Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης.....	68
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	69
ΠΑΡΑΡΤΗΜΑ Α : ΤΙΤΛΟΣ ΠΑΡΑΡΤΗΜΑΤΟΣ.....	71

Κατάλογος Σχημάτων

Εικόνα 1. Σχεδιάγραμμα που δείχνει το ρεύμα δίνης που προκαλείται από κάποιο μαγνητικό πεδίο....	1
Εικόνα 2. Στην φωτογραφία εικονίζεται ο Stephen Hales (1677-1761).....	2
Εικόνα 3. Ο κυματογράφος του Ludwig.....	3
Εικόνα 4. Ο σφυγμογράφος του Vierordt.....	3
Εικόνα 5. Το σφυγμομανόμετρο του Von Basch, εφεύρεση του 1881.....	4
Εικόνα 6. Διάγραμμα ενός παλμού όπου φαίνεται η συστολική και η διαστολική φάση του παλμού....	7
Εικόνα 7. Τα κύρια μέρη του Arduino Mega 2560.....	9
Εικόνα 8. Ο ταλαντωτής που υπάρχει πάνω στην πλακέτα Arduino Uno ο οποίος είναι ίδιο και στο Arduino Mega χρονοσιμμένος στα 16MHz.....	10
Εικόνα 9. Διάγραμμα των pins του Arduino.....	11
Εικόνα 10. Διάγραμμα χαρακτηριστικών πάνω στην πλακέτα Arduino Mega.....	12
Εικόνα 11. Τα pwm pins στο Arduino mega.....	13
Εικόνα 12α. Σχηματικό Arduino Mega 2560.	15
Εικόνα 12β. Σχηματικό Arduino Mega 2560.....	16
Εικόνα 13. Το λογισμικό Arduino IDE.....	17
Εικόνα 14. Παράδειγμα χρήσης των functions μέσα στον κώδικα.....	18
Σχήμα 1: Σχηματικό διάγραμμα όπου φαίνονται τα κυκλώματα και η επεξεργασία που χρειάζεται για τη σωστή λειτουργία της κατασκευής	20
Σχήμα 2. Μπλοκ διάγραμμα ροής του κυκλώματος πιεσόμετρου με την μέθοδο της μαγνητοπυθνησιογραφίας.	21
Εικόνα 15. Χαρακτηριστική καμπύλη της συμπεριφοράς της αντίστασης RG του αισθητήρα GMR σε εξίσωση με το μαγνητικό πεδίο.....	23
Εικόνα 16. Οι μέγιστες τιμές λειτουργίας του αισθητήρα GMR AA002-02.....	23
Εικόνα 17. Τυπικές τιμές λειτουργίας του αισθητήρα GMR AA002-02.....	24
Εικόνα 18. Ένας τυπικός ενισχυτής οργανολογίας.....	25
Εικόνα 19. Οι μέγιστες τιμές λειτουργίας του ενισχυτή οργανολογίας INA129.....	26
Εικόνα 20. Οι προτεινόμενες από τον κατασκευαστή συνθήκες λειτουργίας του ενισχυτή οργανολογίας INA129.....	26
Εικόνα 21. Συνδεσμολογία Sallen-Key χαμηλοπερατού φίλτρου.....	26
Εικόνα 22. Συνδεσμολογία υψιπερατού φίλτρου Sallen-Key.....	27
Εικόνα 23. Πιν διάγραμμα του ολοκληρωμένου κυκλώματος T.E. OP07.....	28

Εικόνα 24. Οι μέγιστες τιμές λειτουργίας του T.E. OP07.....	28
Εικόνα 25. Στον πίνακα φαίνονται οι προτεινόμενες τιμές λειτουργίας του T.E. OP07.....	28
Εικόνα 26. Τα ηλεκτρικά χαρακτηριστικά του T.E. OP07 μέσα από το datasheet της TI (Texas Instruments).....	29
Εικόνα 27. Μπλοκ διάγραμμα του κυκλώματος που αποτελείται ο T.E. OP07.....	30
Εικόνα 28. Μπλοκ διάγραμμα από τα στάδια που περνάει ο αλγόριθμος για την εμφάνιση της αρτηριακής πίεσης.....	30
Εικόνα 29. Απεικόνιση της λειτουργίας του FFT.....	32
Εικόνα 30. Διαφορά ανάλυσης συχνότητας με μικρό αριθμό δειγμάτων και μεγάλο.....	33
Εικόνα 31. Εικονογραφία του υπολογισμού του PTT.....	41
Εικόνα 32. Το κύκλωμα πιεσόμετρου σχεδιασμένο στο Orcad.....	44
Εικόνα 33. Η κυματομορφή προσομοίωσης του αισθητήρα AA002-02 από το Orcad με την προσθήκη θορύβου.....	44
Εικόνα 34. Τιμές θορύβου οι οποίες έχουν δημιουργηθεί στο Excel.....	45
Εικόνα 35. Κυματομορφή εξόδου του ενισχυτή INA129.....	46
Εικόνα 36. Κυματομορφή εξόδου πρώτου χαμηλοπερατού φίλτρου Sallen-Key.....	47
Εικόνα 37. Κυματομορφή εξόδου υψιπερατού φίλτρου Sallen-Key.....	47
Εικόνα 38. Κυματομορφή εξόδου του κυκλώματος που προορίζεται για επεξεργασία στο Arduino.....	48
Εικόνα 39. Σχεδίο της πλακέτας σε φορματ αρχείου .Cam υλοποιημένο σε Altium Designer.....	49
Εικόνα 40. Σχεδίο της πλακέτας σε φορματ αρχείου .pcbdoc υλοποιημένο σε Altium Designer....	49
Εικόνα 41. Σχεδίο της πλακέτας σε φορματ αρχείου .pcblib υλοποιημένο σε Altium Designer.....	50
Εικόνα 42. Οι αισθητήρες AA002-02 πάνω στο κουφάρι με τον μαγνήτη έτοιμο προς τοποθέτηση στο χέρι.....	50
Εικόνα 43. Στην εικόνα φαίνεται η σύνδεση των αισθητήρων στην πλακέτα, έχει χρησιμοποιηθεί πλαστική επιφάνεια στην οποία έχουν κολληθεί πάνω με ταινία διπλής όψης οι αισθητήρες και ανάμεσα ο μαγνήτης κάθετα με τους αισθητήρες. Για την τοποθέτηση στον καρπό έχει χρησιμοποιηθεί ταινία Velcro(χριτσ-χρατσ), και τέλος για την σύνδεση με το κύκλωμα χρησιμοποιήθηκε καλώδιο Ethernet ώστε να μπορεί να χρησιμοποιηθεί από απόσταση.....	51
Εικόνα 44. Έξοδος του αισθητήρα AA002-02 σε σχέση με το μαγνητικό παιδί για ρεύμα 1mA.....	52
Εικόνα 45. Δύο πυκνωτές παράλληλα συνδεδεμένοι ο ένας 1uF και ο άλλος 0.56uF.....	53
Εικόνα 46. Οι δύο ρυθμιστές τάσεις, LM7805 και LM7905.....	53
Εικόνα 47. Μπαταρίες τροφοδοσίας του κυκλώματος.....	54
Εικόνα 48. Το κύκλωμα πιεσόμετρου υλοποιημένο σε ράστερ.....	54
Εικόνα 49. Η πλακέτα Arduino Mega που χρησιμοποιήθηκε στο κύκλωμα.....	55
Εικόνα 50. Η επιλογή Serial plotter στο Arduino IDE.....	55

Εικόνα 51. Ο κώδικας ελέγχου εξόδου του κυκλώματος.....	56
Σχήμα 3. Μπλοκ διάγραμμα υλοποίησης κυκλώματος και επιπέδου επεξεργασίας και απεικόνισης πιεσομέτρου.....	57
Εικόνα 52. Η οθόνη LCD.....	58
Εικόνα 53. Κυματομορφή εξόδου των δύο κυκλωμάτων.....	59

Κατάλογος Πινάκων

Πίνακας 1: Ενδεικτικός πίνακας τιμών αρτηριακής πίεσης ταξινομημένος κατ' ηλικία.....	6
Πίνακας 2 :Πίνακας χαρακτηριστικών Arduino Mega 2560.....	14
Πίνακας 3: Τα βασικά υλικά που χρησιμοποιήθηκαν για την κατασκευή του κυκλώματος.....	49

Συντομογραφίες

GMR	Giant Magneto-Resistance
PTT	Pulse Transition Time
USB	Universal Serial Bus
PWM	Pulse Wave Modulation
ICSP	In-Circuit Serial Programming
RAM	Random Access Memory
ADC	Analog to Digital Converter
LCD	Liquid Crystal Display
CMRR	Common Mode Rejection Ratio
T.E.	Τελεστικός Ενισχυτής
FFT	Fast Fourier Transform
N.C.	Not Connected

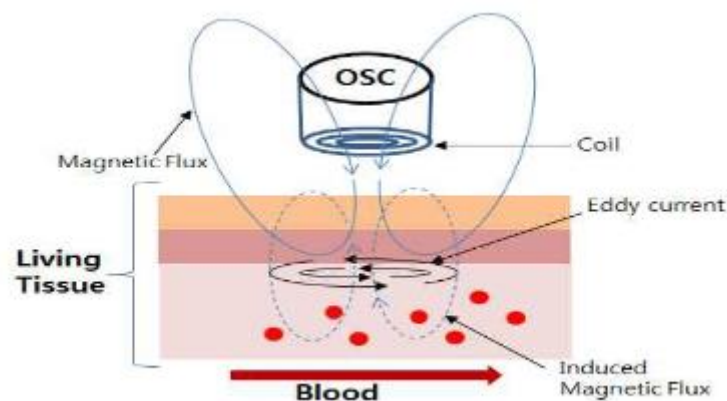
Κεφάλαιο 1ο : Πιεσόμετρο

1.1 Εισαγωγή

Η μέτρηση της αρτηριακής πίεσης μπορεί να είναι καθοριστικής σημασίας στην καθημερινότητα μας καθώς μπορεί να αποτρέψει αρκετές επικίνδυνες καταστάσεις για την ζωή ενός ανθρώπου. Υπάρχουν διάφορες μέθοδοι μέτρησης της αρτηριακής πίεσης, στην εργασία αυτήν θα ασχοληθούμε με την μέτρηση με την μέθοδο της μαγνητο-πλυθισμογραφίας.

1.2 Τι είναι η μέθοδος της μαγνητοπλυθισμογραφίας

Η μέθοδος της μαγνητοπλυθισμογραφίας εφαρμόζεται μετρώντας τις αλλαγές της αγωγιμότητας των ιστών του σώματος με την χρήση του ρεύματος δίνης(eddy currents) που προκαλείται από κάποιο μαγνητικό πεδίο το οποίο αλλάζει με την πάροδο του χρόνου.[1] Για την ενσωμάτωση της μεθόδου αυτής σε ένα σύστημα θα χρειαστεί κάποιος αισθητήρας ανίχνευσης μαγνητικών πεδίων, στην περίπτωση μας ο αισθητήρας GMR, ώστε να καταγράφει τους καρδιακούς παλμούς και κάποιος μαγνήτης σε υπολογισμένη απόσταση ώστε να επιτρέπει στον αισθητήρα να λειτουργεί στην γραμμική περιοχή του.



Εικόνα 1. Σχεδιάγραμμα που δείχνει το ρεύμα δίνης που προκαλείται από κάποιο μαγνητικό πεδίο.[1]

Για την μέτρηση της αρτηριακής πίεσης θα χρησιμοποιηθούν δύο αισθητήρες, με έναν μαγνήτη δίπλα στον καθένα από αυτούς οι οποίοι θα τοποθετηθούν στην κερκιδική αρτηρία σε κάποια απόσταση μεταξύ τους έτσι ώστε να μετρούν την διαφορά της ταχύτητας του καρδιακού παλμού (PWV) και τον χρόνο μετάβασης παλμού (PTT).

1.3 Ιστορική αναδρομή μέτρησης αρτηριακής πίεσης

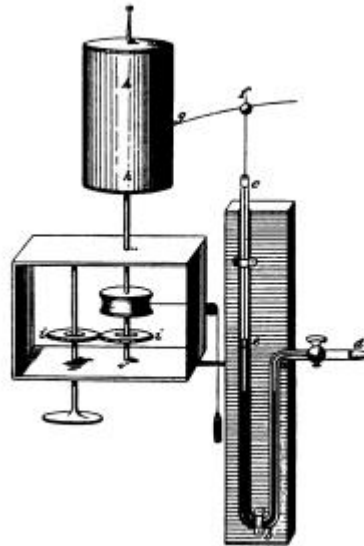
Η αρχική ιδέα της μέτρησης της αρτηριακής πίεσης προήλθε στα μέσα του 18^{ου} αιώνα μέσα από τα πειράματα του Stephen Hales(βλ. Εικόνα 2).[2]



Εικόνα 2. Στην φωτογραφία εικονίζεται ο Stephen Hales (1677-1761).[2]

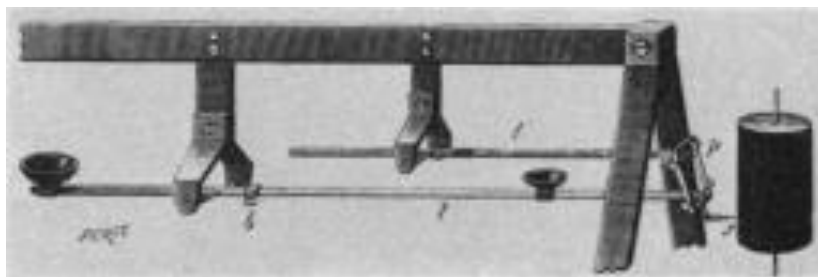
Η ανακάλυψη της αρτηριακής πίεσης μπορεί να χρεωθεί στον Stephen Hales παρόλο που για χρόνια υπήρχαν πριν τον 18^ο αιώνα φυσιολόγοι που ενώ έβλεπαν περιστατικά αίματος να εκτοξεύεται από σχισμένα αγγεία δεν τους κινήθηκε η περιέργεια να καταλάβουν τον λόγο. Οι πρώτες μετρήσεις της αρτηριακής πίεσης έγιναν το 1733 παρόλο που ήταν άτεχνες και πολύ μακριά από τις τεχνικές που χρησιμοποιούνται σήμερα. Ένω ο Hales συνέχισε τα πειράματα πάνω στην μέτρηση της αρτηριακής πίεσης, έπρεπε να φτάσουμε στο 1828 όπου ο Poiseuille ο οποίος θεωρείται ένας από τους πρώτους φυσικούς-γιατρούς, σύστησε στον κόσμο το μανόμετρο υδρογόνου για το οποίο μέσω της διδακτορικής διατριβής του πάνω στην χρήση του για την μέτρηση της αρτηριακής πίεσης βραβεύτηκε με το χρυσό μετάλλιο της Βασιλικής Ακαδημίας Ιατρικής. Ο Poiseuille είναι ο πρώτος μετά τον Stephen Hales που έκανε σημαντική πρόοδο στην μέτρηση της αρτηριακής πίεσης.[2]

Η εφεύρεση του Poiseuille ενέπνευσε μετά από μία εικοσαετία και συγκεκριμένα το 1847 τον Carl Ludwig να αναπτύξει τον κυματογράφο.[2] Ο κυματογράφος ήταν μία βελτιωμένη έκδοση του μανόμετρου του Poiseuille, με το χαρακτηριστικό της μέτρησης σε πραγματικό χρόνο.[3] Στην κατασκευή του μανόμετρου του Poiseuille συνέδεε το μανόμετρο με έναν σωλήνα γεμάτο με ανθρακικό κάλλιο που λειτουργούσε ως αντιπηκτικό, και ο σωλήνας εισχωρούσε στην αρτηρία του πειραματόζωου. Αρτηρίες μικρές με διάμετρο έως και 2 χιλ. διασωληνώθηκαν και έτσι ο Poiseuille κατάφερε να αποδείξει ότι η αρτηριακή πίεση διατηρούνταν στις μικρότερες αρτηρίες.[2] Η κατασκευή του Ludwig κράτησε τα χαρακτηριστικά του μανόμετρου και του σωλήνα, αλλά πρόσθεσε έναν πλωτήρα με ένα στυλό το οποίο έγραφε πάνω σε ένα περιστρεφόμενο τύμπανο.[2]



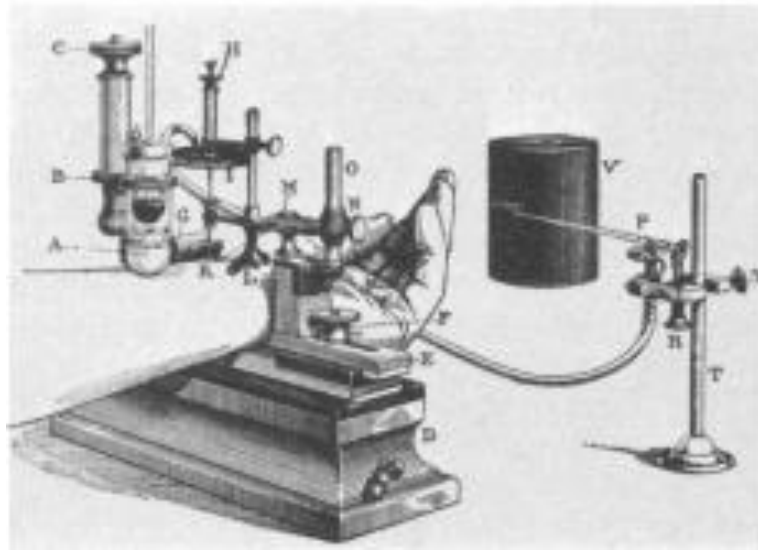
Εικόνα 3. Ο κυματογράφος του Ludwig.[2]

Εώς και το 1855 δεν υπήρχε μη επεμβατική μέθοδος μέτρησης της πίεσης μέχρι τον σφυγμογράφο του Vierordt, ο οποίος θεώρησε ότι θα μπορούσε να υπάρξει κάποια μη επεμβατική μέθοδος μέτρησης μετρώντας την αντίθετη πίεση που θα ήταν απαραίτητη για να προκαλέσει τον παλμό στην αρτηρία να διακοπεί, κάτι που προσπάθησε να το κάνει στην πράξη χωρίς μεγάλη επιτυχία εξαιτίας του δυσκίνητου σχεδιασμού της συσκευής. 5 χρόνια μετά και έχοντας την ιδέα του σφυγμογράφου του Vierordt, ο Etienne Jules Marey προσπάθησε να την εξελίξει βελτιώνοντας την τεχνική απεικονιστικής καταγραφής του παλμού όπως και την ακρίβεια της καταγραφής της αρτηριακής πίεσης. Στην κατασκευή του Marey χρησιμοποιήθηκε η ίδια αρχή με τον Vierordt αλλά στην δική του συσκευή το χέρι ήταν κλεισμένο μέσα σε έναν γυάλινο θάλαμο γεμάτο με νερό όπου ήταν συνδεδεμένος με έναν σφυγμογράφο κι έναν κυματογράφο όπου κατέγραφαν τους αρτηριακούς παλμούς του χεριού. Στον θάλαμο υπήρχε και μία κινούμενη δεξαμενή γεμάτη με νερό που βοηθούσε στις εναλλαγές της πίεσης μέσα στον θάλαμο και υπήρχε κι ένα μανόμετρο για την ένδειξη της πίεσης του θαλάμου. Η καταγραφή της πίεσης γινόταν αρχικά σημειώνοντας την μεγαλύτερη απόσταση από κορυφή σε κορυφή που ιχνηλατούσε ο σφυγμογράφος κι έπειτα μέσω του της δεξαμενής με το νερό ανέβαζε την πίεση του θαλάμου εως ώτου δεν υπήρχε άλλη κίνηση στον σφυγμογράφο, σε εκείνη την τιμή καταγραφόταν η συστολική πίεση. Παρόλο που η συγκεκριμένη κατασκευή ήταν εντυπωσιακή, ήταν πολύ δύσκολη στην χρήση για τους γιατρούς έτσι ώστε να την χρησιμοποιούν σε καθημερινή βάση.[2]



Εικόνα 4. Ο σφυγμογράφος του Vierordt.[2]

Στην εξέλιξη της μη επεμβατικής μεθόδου μέτρησης της πίεσης ανέλαβε τα ηνία ο Samuel Siegfried Karl Ritter von Basch, απόφοιτος του τμήματος ιατρικής στην Βιέννη το 1962. Η μέθοδος του von Basch περιείχε μία φουσκωτή σακούλα από καουτσούκ γεμάτη με νερό, η οποία ήταν σφιχτά δεμένη γύρω από ένα βολβό μανόμετρου γεμάτο με υδράργυρο. Ένα σωληνάκι υπήρχε στον βολβό έτσι ώστε κάθε πίεση που δημιουργούνταν στο σακουλάκι με νερό θα μεταφερόταν στον βολβό, ο υδράργυρος θα γέμιζε τον βολβό και έτσι θα καταγραφόταν η πίεση. [2]



Εικόνα 5. Το σφυγμομανόμετρο του Von Basch, εφεύρεση του 1881.[2]

Το μεγάλο άλμα ωστόσο στην μέτρηση της πίεσης έγινε από τον Riva-Rocci, καθώς η εξέλιξη των μεθόδων κάθε άλλο παρά συνεχώς αναπτυσσόμενη ήταν, καθώς υπήρχε βαθμιαία βελτίωση κάθε προηγούμενης μεθόδου με κάθε νέα μέθοδο να γίνεται αργά αποδεκτή. Στη μέθοδο του Riva-Rocci βασίζεται η σημερινή μέθοδος μέτρησης. Η συσκευή του Riva-Rocci και βλέποντας τις παλαιότερες εφευρέσεις, θεωρήθηκε από αυτόν ότι είναι εύκολη στην εφαρμογή της, γρήγορη στην δράση της, ακριβής και ασφαλής για τον ασθενή. Η μέθοδος του περιείχε συμπίεση σε όλη την περιφέρεια του χεριού. Μία τσάντα από καουτσούκ περιτριγυρισμένη από ένα περιβραχιόνιο φτιαγμένο από μη επεκτάσιμο υλικό, τυλιγόταν γύρω από την περιφέρεια του μπράτσου και γέμιζε με αέρα μέσω ενός βολβού από καουτσούκ. Η πίεση καταγραφόταν μέσω του μανόμετρου υδραργύρου και θα αυξανόταν έως ώτου δεν υπήρχε παλμός. Όταν η πίεση μειωνόταν αργά θα έπεφτε η ένδειξη του μανόμετρου και η στιγμή της ένδειξης όπου θα ξαναεμφανιζόταν ο παλμός θα καταγραφόταν ως η συστολική πίεση. Το σημαντικό πλεονέκτημα αυτής της τεχνικής ήταν ότι η αρτηρία του βραχίονα συμπιεζόταν από όλες τις πλευρές το ίδιο, οπότε εξαφάνιζε την μονομερή πίεση της τεχνικής του Von Basch. Το μεγάλο μειονέκτημα ωστόσο αυτής της τεχνικής ήταν ότι χρησιμοποιούσε στενό περιβραχιόνιο, το οποίο ήταν 5 εκατοστά σε πλάτος, αυτό δημιουργούσε οξείες γωνίες στις δύο μερίες του περιβραχιονίου σε σχέση με το δέρμα κάτι το οποίο είχε ως αποτέλεσμα σε συγκεκριμένα σημεία να δημιουργούταν υψηλή πίεση κάτι το οποίο θα οδηγούσε σε μη ακριβή μέτρηση. Αυτό το σφάλμα διορθώθηκε από τον von Recklinghausen το 1901 χρησιμοποιώντας ένα περιβραχιόνιο με πλάτος 12 εκατοστά.[2]

Οι μέθοδοι που υπήρχαν μέχρι τότε ήταν αρκετά ακριβείς για την μέτρηση της συστολικής πίεσης αλλά υπήρχε πρόβλημα με την μέτρηση της διαστολικής αρτηριακής πίεσης. Για να το διορθώσουν αυτό το πρόβλημα, οι γιατροί άρχισαν να χρησιμοποιούν την μέθοδο των ταλαντώσεων, κάτι το οποίο περιλάμβανε την παρατήρηση των ταλαντώσεων που μεταφέρονταν από την αρτηρία στον υδράργυρο του μανόμετρου όταν η πίεση του περιβραχιόνιου και της αρτηριακής πίεσης ήταν ίση, έτσι η αρτηρία θα παλλόταν δημιουργώντας διακυμάνσεις στην πίεση του περιβραχιόνιου. Έτσι, η εμφάνιση ταλαντώσεων όριζε την συστολική πίεση και η μετάβαση από τις μεγάλες ταλαντώσεις στις μικρές όριζε την διαστολική.[2]

Στις αρχές του 20^{ου} αιώνα ήταν ένας Ρώσος χειρουργός ο N C Korotkoff που θα βελτίωνε ακόμα περισσότερο τις μέχρι τότε υπάρχουσες μεθόδους. Ο Korotkoff παρατήρησε ότι χρησιμοποιώντας ένα στηθοσκόπιο κάτω από το περιβραχιόνιο μέτρησης πίεσης θα διέκρινε τους ήχους των παλμών, έτσι παρατήρησε ότι μία τέλεια σφιγμένη αρτηρία δεν θα εξέπεμπε κάθολου ήχο. Έχοντας αυτό κατα νου, ήρθε με την ιδέα ότι όταν εξαφανιστεί κάθε ήχος παλμού από το περιβραχιόνιο αρχικά και έπειτα θα εμφανιστούν αχνοί τόνοι όπου ο πρώτος τόνος που θα εμφανιστεί υποδεικνύει την μέγιστη πίεση, και ένω ο υδράργυρος του μανομέτρου θα συνεχίσει να πέφτει ακούγονται ξανά αχνοί ήχοι συστολικής πίεσης όπου έπειτα γίνονται κανονικοί ήχοι. Εν τέλει όλοι οι ήχοι εξαφανίζονται κάτι το οποίο υποδεικνύει την ελάχιστη πίεση.[2]

Η συνεισφορά του Korotkoff ήταν τεράστια καθώς οι τεχνικές του χρησιμοποιούνταν για περισσότερο από μισό αιώνα χωρίς καμία αλλαγή στην τεχνική. Από το 1970 και έπειτα η αρτηριακή πίεση μπορεί να μετρηθεί με διάφορες τεχνικές μεταξύ των οποίων, με αισθητήρες τοποθετημένους στο δάχτυλο του ασθενή, την γνωστή τεχνική του περιβραχιόνιου κ.α. έχοντας ληφθεί υπόψιν όργανα καταπόνησης, φωτοκύτταρα και ημιαγωγοί για την κατασκευή λύσεων συστημάτων μέτρησης αρτηριακής πίεσης.[2]

1.4 Συστολική, διαστολική και μέση πίεση

Η αρτηριακή πίεση χωρίζεται σε δύο μέρη, την συστολική και την διαστολική, όπου η φυσιολογική τιμή για την καθεμία είναι διαφορετική. Πιο αναλυτικά, όσο χτυπάει η καρδιά μεταξύ δύο χτύπων διώχνει μία ποσότητα αίματος και ξαναγεμίζει με αίμα, δηλαδή συστέλεται και διαστέλεται, κάτι το οποίο μετράται με δύο αριθμούς, ο ένας αντικατοπτρίζει την συστολική πίεση και ο άλλος την διαστολική. Ο πρώτος αριθμός εκφράζει την πίεση που ασκεί το αίμα στο τοίχωμα των αρτηριών καθώς φεύγει από την καρδιά και ονομάζεται συστολική πίεση ή μεγάλη, ο δεύτερος αριθμός, η διαστολική πίεση ή αλλιώς μικρή είναι η πίεση που ασκεί το αίμα στα τοιχώματα των αρτηριών στο ενδιάμεσο δύο χτύπων, δηλαδή όσο η καρδιά ξεκουράζεται.[4] Οι φυσιολογικές τιμές πίεσης για έναν νέο ενήλικο είναι 120 η μεγάλη και 80 η μικρή, και μετράται σε χιλιοστά υδραργύρου (mmHg).[5]

Ηλικία	Ελάχιστο όριο(mmHg)	Μέση τιμή(mmHg)	Ανώτατο όριο(mmHg)
15-19	105/73	117/77	120/81
20-24	108/75	120/79	132/83
25-29	109/76	121/80	133/84
30-34	110/77	122/81	134/85
35-39	111/78	123/82	135/86
40-44	112/79	125/83	137/87
45-49	115/80	127/84	139/88
50-54	116/81	129/85	142/89
55-59	118/82	131/86	144/90
60-64	121/83	134/87	147/91

Πίνακας 1:Ενδεικτικός πίνακας τιμών αρτηριακής πίεσης ταξινομημένος κατ' ηλικία.[5]

Πέρα από την διαστολική και συστολική αρτηριακή πίεση, κάποια όργανα μέτρησης έχουν και ένδειξη της μέσης αρτηριακής πίεσης, η οποία είναι ένας υπολογισμός που γίνεται από τους γιατρούς για να ελέγξουν αν υπάρχει αρκετή ροή αίματος, αντίσταση και πίεση ώστε το αίμα να παρέχεται σε όλα τα κύρια όργανα του ανθρώπινου σώματος. Η αντίσταση υποδεικνύει τη διάμετρο του αιμοφόρου αγγείου και τον τρόπο που επηρεάζει αυτό την ροή του αίματος, όσο μεγαλύτερη η αντίσταση, τόσο μεγαλύτερη και η πίεση καθώς μειώνεται η ροή του αίματος. Οι φυσιολογικές τιμές της μέσης αρτηριακής πίεσης κινούνται μεταξύ 70 με 100 mmHg, ενώ θα πρέπει να είναι τουλάχιστον 60 mmHg ώστε να υπάρχει επαρκής ροή αίματος σε όλα τα ζωτικά όργανα. [9] Ο γενικός τύπος για τον υπολογισμό της μέσης αρτηριακής πίεσης είναι ο εξής:

$$\text{Μέση πίεση} = \frac{\text{Συστολική πίεση} + 2 \times \text{Διαστολική πίεση}}{3}$$

[10]

Μία υψηλή τιμή μέσης αρτηριακή πίεσης (ΜΑΠ), δηλαδή πάνω από 100 mmHg θα μπορούσε να οδηγήσει σε σχηματισμό θρόμβων ή βλάβη στον καρδιακό μυ. Ενώ μία χαμηλή τιμή ΜΑΠ, δηλαδή μία τιμή χαμηλότερη από 60 mmHg υποδεικνύει ότι δεν φτάνει αρκετό αίμα στα κύρια όργανα, κάτι το οποίο θα μπορούσε να οδηγήσει σε μόνιμη βλάβη των οργάνων καθώς δεν παίρνουν το αίμα και παράλληλα τα θρεπτικά συστατικά που χρειάζονται. [9]

1.4.1 Τι είναι η υπέρταση

Όταν η τιμή της αρτηριακής πίεσης βρίσκεται ανάμεσα στην μέση τιμή και την ανώτερη τιμή αυτό ονομάζεται αρτηριακή υπέρταση.[5] Σύμφωνα με τις οδηγίες της Ευρωπαϊκής Καρδιολογικής Εταιρίας, οι τιμές της υπέρτασης σταδίου 1 αρχίζουν από τις παρακάτω τιμές:

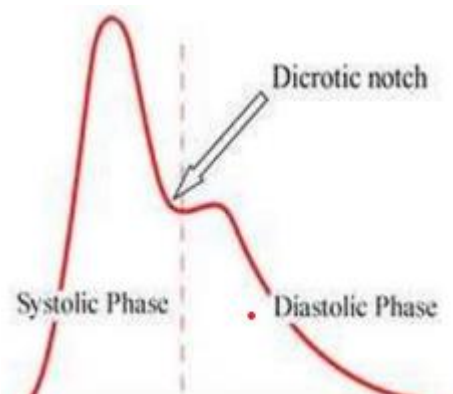
>140mmHg για την συστολική

>90mmHg για την διαστολική[4]

Τα μέτρα πρόληψης για την αποφυγή της υπέρτασης είναι η καταπολέμηση του στρες, η διατήρηση του βάρους σε φυσιολογικά επίπεδα καθώς και η υγιεινή διατροφή και η γυμναστική[5]

1.4.2 Οδηγίες για σωστή μέτρηση της πίεσης

Η πίεση πρέπει πάντα να μετράται αφού έχει προηγηθεί ένα εύλογο χρονικό διάστημα ξεκούρασης ώστε η μέτρηση να ανταποκρίνεται στην πραγματική πίεση του ατόμου, όπως επίσης πρέπει να προηγείται ένα διάστημα τουλάχιστον μίας ώρας νηστείας καθώς και ο καφές και το τσιγάρο επίσης θα μπορούσαν να επηρεάσουν την πίεση οπότε για την σωστή μέτρηση θα πρέπει να αποφεύγονται. Η γυμναστική είναι ένας τομέας επίσης που θα μπορούσε να επηρεάσει την πίεση του ατόμου οπότε εφόσον το άτομο έχει κάνει γυμναστική για να ξεκινήσει την μέτρηση θα πρέπει να μείνει τουλάχιστον 2 ώρες ξεκούραστος μετά την άσκηση. Η πίεση θα πρέπει πάντα να μετράται σε ένα ήσυχο περιβάλλον και με την κατάλληλη θερμοκρασία δωματίου, καθώς και η σύσταση για την σωστή μέτρηση είναι η στάση του σώματος να σε καθιστή θέση με την πλάτη να στηρίζεται στο κάθισμα και το μπράτσο να είναι χαλαρό και να στηρίζεται σε κάποιο σημείο. Επίσης παράγοντας που θα μπορούσε να επηρεάσει την μέτρηση είναι η θέση των ποδιών όπου δεν πρέπει να είναι σταυρωμένα, και θα πρέπει να αποφεύγεται η ομιλία κατά την διάρκεια της μέτρησης. [4]



Εικόνα 6. Διάγραμμα ενός παλμού όπου φαίνεται η συστολική και η διαστολική φάση του παλμού.[6]

1.5 Τι είναι η μαγνητική πληθυσμογραφία

Η μαγνητική μέθοδος που αντιστοιχεί στην πληθυσμογραφία της ηλεκτρικής αντίστασης ονομάζεται μαγνητική πληθυσμογραφία. Αυτή η μέθοδος μπορεί να χρησιμοποιηθεί για το έλεγχο των αλλαγών του όγκου αίματος μη επαμβατικά και εύκολα. Οι περισσότεροι ανθρώπινοι ιστοί είναι διαμαγνητικοί,[7] κάτι που σημαίνει ότι δημιουργούν ένα επαγόμενο μαγνητικό πεδίο με αντίθετη κατεύθυνση από την εξωτερική μαγνητική πηγή[8], κάτι που γίνεται λόγω του ότι οι ιστοί έχουν ως κύριο συστατικό τους το νερό. Έτσι αν εφαρμοστεί ένα μαγνητικό πεδίο πάνω στους ιστούς του σώματος, δημιουργούνται μεταβολές στη μαγνητική ροή.[7].

1.6 Επίλογος

Σε αυτό το κεφάλαιο ασχοληθήκαμε με το πιεσόμετρο, κάναμε μία ιστορική αναδρομή πάνω στην κατασκευή και την εξέλιξη του πιεσόμετρου, αναλύσαμε τις φυσιολογικές μετρήσεις πάνω στην συστολική, διαστολική και την μέση πίεση όπως επίσης και την έννοια της μαγνητικής

Κεφάλαιο 1

πλυθισμογραφίας η οποία είναι μία μη επεμβατική μέθοδος ανάπτυξης μέτρησης πιεσόμετρου με την οποία θα ασχοληθούμε σε αυτήν την πτυχιακή εργασία

Κεφάλαιο 2ο : Αναπτυξιακή πλακέτα Arduino

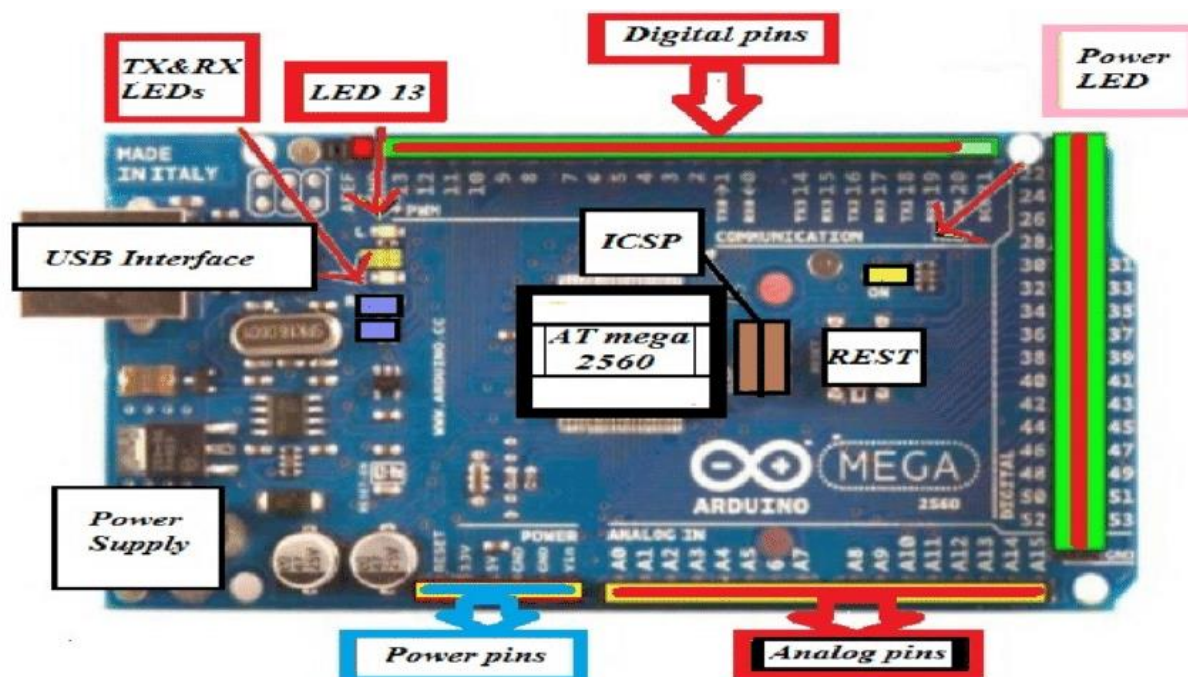
2.1 Εισαγωγή

Οι μικροελεγκτές έχουν παίξει καθοριστική σημασία πάνω σε διάφορους τομείς της τεχνολογίας και τείνουν να χρησιμοποιούνται όλο και περισσότερο σε καινοτόμες ιδέες καθώς υπάρχουν φθηνές λύσεις και εύκολες στην χρήση. Ένας ευρέως διαδεδομένος μικροελεγκτής είναι και η αναπτυξιακή πλακέτα Arduino η οποία βγαίνει σε διάφορες εκδόσεις, άλλες με λιγότερες και άλλες με περισσότερες δυνατότητες.

2.2 Τι είναι το Arduino

Το Arduino είναι μία μητρική πλακέτα με ενσωματωμένο προγραμματιζόμενο μικροελεγκτή και εισόδους/εξόδους η οποία προγραμματίζεται με μία παραλλαγή της γλώσσας C++, την γλώσσα Wiring. Πάνω στην πλακέτα υπάρχουν pins εισόδων/ εξόδων, ο κρύσταλλος, τα pins τροφοδοσίας, η θύρα USB, η θύρα εξωτερικής τροφοδοσίας, τα αναλώσιμα ηλεκτρονικά (αντιστάσεις, πυκνωτές κλπ) κ.α.[11]

Στην παρακάτω εικόνα (Εικόνα 7) βλέπουμε τα κύρια μέρη του Arduino που θα ασχοληθούμε σε αυτήν την πτυχιακή, του Arduino Mega 2560:



Εικόνα 7. Τα κύρια μέρη του Arduino Mega 2560.[12]

2.3 Ανάλυση των υλικών που αποτελείται η πλακέτα Arduino Mega

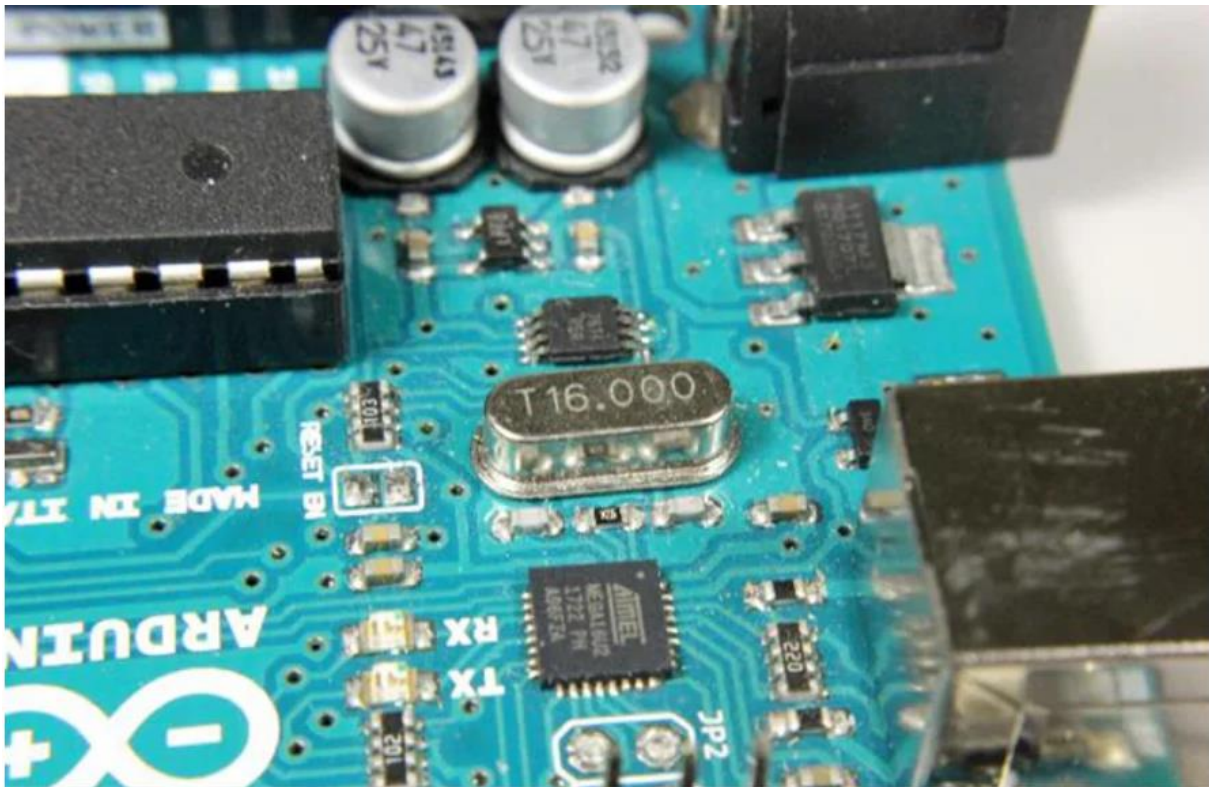
Ο μικροελεγκτής Arduino Mega 2560 αποτελείται από διάφορα μέρη τα οποία λειτουργούν συνδυαστικά για την διεκπεραίωση πολλών και διαφόρων προτζεκτ. Τα κύρια μέρη του είναι τα 54 ψηφιακά pins τα οποία λειτουργούν ως εισοδοί/έξοδοι, αναλόγα με το πως τα δηλώνει ο χρήστης μέσω του προγράμματος που φορτώνει στον μικροελεγκτή, από τα οποία τα 15 μπορούν να λειτουργήσουν ως PWM (Pulse Wave Modulation), όπως επίσης έχει και 16 αναλογικά pins. Πέρα από τα pins εισόδων/εξόδων αποτελείται επίσης από:

- Έναν κρύσταλλο 16MHz
- Ένα USB port
- Ένα Reset button
- Μία θύρα τροφοδοσίας
- Πινακίδα ICSP (In-Circuit Serial Programming)[13]
- Και κάποια επιπλέον περιφερειακά

Οι λειτουργίες των οποίων θα αναλυθούν παρακάτω.

2.3.1 Ο κρύσταλλος 16MHz

Όλοι οι μικροελεγκτές Arduino αποτελούνται από έναν κρύσταλλο ταλαντωτή, στην περίπτωση του Arduino Mega χρησιμοποιείται ένας ταλαντωτής χρονοσιμμένος στα 16MHz. Η αναγνώριση του πάνω στην πλακέτα φαίνεται στην παρακάτω εικόνα (Εικόνα 8) όπου το T16.000 δείχνει τη συχνότητα που είναι χρονοσιμμένος ο ταλαντωτής.[14]



Εικόνα 8. Ο ταλαντωτής που υπάρχει πάνω στην πλακέτα Arduino Uno ο οποίος είναι ίδιο και στο Arduino Mega χρονοσιμμένος στα 16MHz.[14]

Ο ταλαντωτής αυτός είναι υπεύθυνος για την ρύθμιση του ATmega16 οπότε υπάρχει στο Arduino Uno [14]ή στην περίπτωση μας του ATmega16U2 όπου υπάρχει στο Arduino Mega, η χρήση αυτού του επεξεργαστή είναι η μετατροπή USB-to-Serial όταν ο μικροελεγκτής συνδέεται με κάποιον υπολογιστή. [14]

2.3.2 Η θύρα USB

Το Arduino έχει πάνω του μία θύρα USB AB τύπου η οποία έχει δύο χρήσεις, η μία είναι η τροφοδοσία του και η άλλη είναι η επικοινωνία του με τον υπολογιστή. Στην περίπτωση της

τροφοδοσίας, υπάρχουν δύο διαφορετικές λειτουργίες, η μία είναι όταν η σύνδεση αναγνωρίζεται από τον υπολογιστή όπου τότε η πλακέτα λαμβάνει τροφοδοσία της τάξης των 500mA στα 5V. Όταν από την άλλη δεν αναγνωρίζεται η σύνδεση, η πλακέτα λαμβάνει τροφοδοσία της τάξης των 100mA στα 5V.[15]

2.3.3 Το Reset button

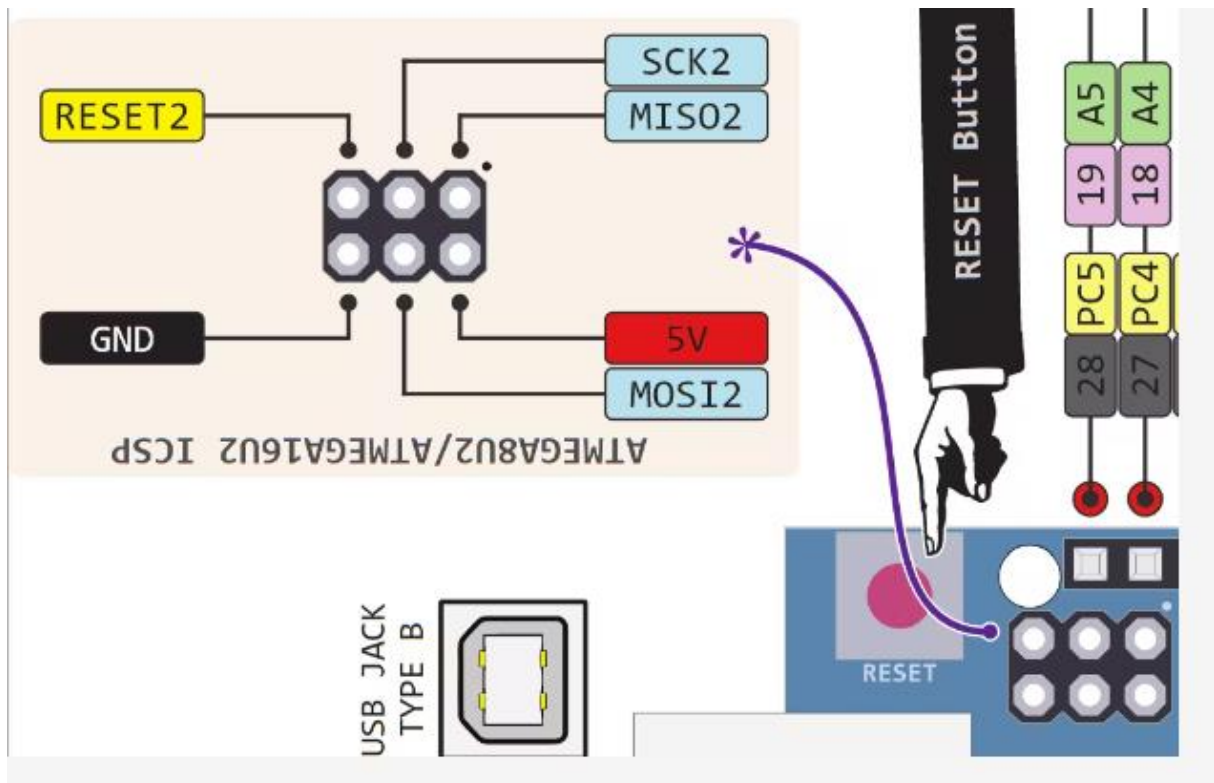
Το reset button στο Arduino mega αναγνωρίζεται ως ένα κόκκινο κουμπί πάνω στην πλακέτα, η χρήση του είναι, πατώντας το αποσυνδέει και επανασυνδέει την πλακέτα στην τροφοδοσία. Η πλακέτα μετά το πάτημα του θα ξεκινήσει να εκτελεί ξανά από την αρχή τον κώδικα ο οποίος έχει φορτωθεί στην πλακέτα. Όταν η πλακέτα χάνει την τροφοδοσία της, σβήνει η μνήμη RAM οπότε οι τιμές οι οποίες έχουν ανατεθεί στις μεταβλητές του κώδικά παύουν να ισχύουν. [16]

2.3.4 Η θύρα τροφοδοσίας της πλακέτας

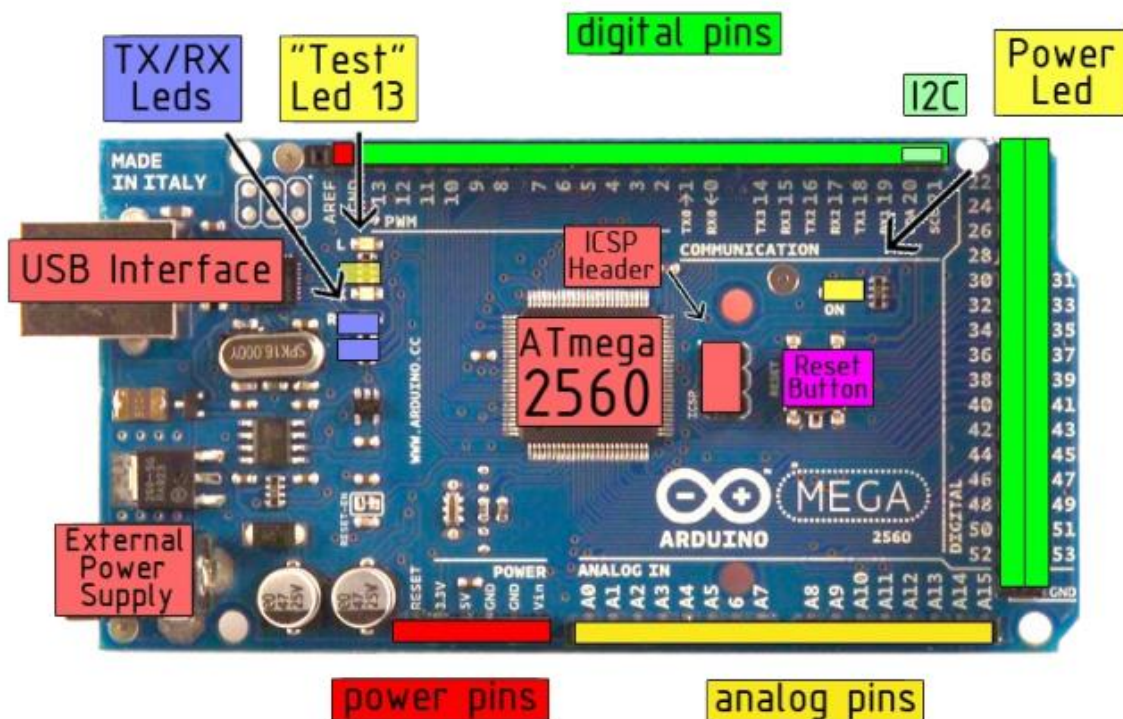
Πέρα από την θύρα USB, το Arduino Mega μπορεί να τροφοδοτηθεί και από εξωτερική πηγή μέσω ενός βύσματος 2.1mm με κεντρικό θετικό αν χρησιμοποιηθεί κάποιος αντάπτορας AC-DC ή αν χρησιμοποιηθεί μπαταρία με την χρήση των pins της πλακέτας Vin και Gnd. Η απαιτούμενη εξωτερική τροφοδοσία για να λειτουργήσει η πλακέτα πρέπει να βρίσκεται μέσα στα όρια των 6 με 20 volt, και η προτεινόμενη τροφοδοσία είναι 7 με 12 volt καθώς κάτω από 7 volt μπορεί να μην λειτουργεί σταθερά και πάνω από 12 μπορεί να υπερθερμανθεί.[15]

2.3.5 Τα Pins ICSP

Το ακρωνύμιο ICSP βγαίνει από τις λέξεις In-Circuit Serial Programming. Αυτά τα pins υπάρχουν για να δώσουν την δυνατότητα στον χρήστη να προγραμματίσει το firmware του μικροελεγκτή. [17]



Εικόνα 9. Διάγραμμα των pins του Arduino.[17]



Εικόνα 10. Διάγραμμα χαρακτηριστικών πάνω στην πλακέτα Arduino Mega.[15]

2.3.6 Τα ψηφιακά πινς

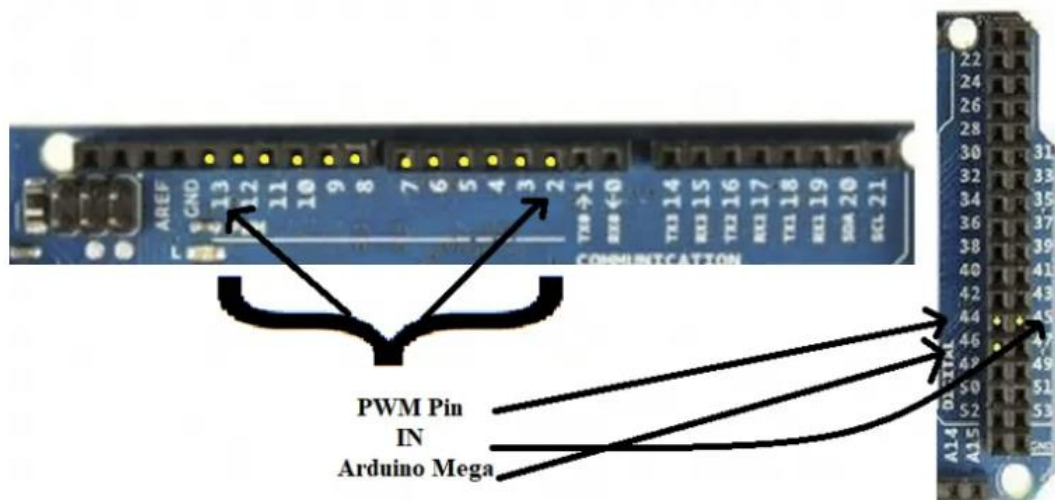
Στην πλακέτα Arduino Mega 2560 υπάρχουν 54 ψηφιακά πινς, εκ των οποίων τα 15 είναι pwm πινς και επίσης τα 4 είναι σειριακές θύρες UARTs. Το UART προέρχεται από τα αρχικά Universal Asynchronous Receiver and Transmitter και η βασική τους λειτουργία είναι να επιτρέπουν στην πλακέτα να επικοινωνεί με σειριακές συσκευές. Οι 4 σειριακές θύρες είναι οι εξής:

- Το pin 0-RXD0, και το pin 1-TXD0
- Το pin 19-RXD1, και το pin 18-TXD1
- Το pin 17-RXD2, και το pin 16-TXD2
- Και το pin 15-RXD3, και το pin 14-TXD3

Αυτά τα πινς χρησιμοποιούνται για σειριακή UART επικοινωνία με τον υπολογιστή ή κάποια άλλη σειριακή συσκευή και η λειτουργία τους είναι ο διαμοιρασμός δεδομένων και το logging. Η χρήση αυτών των θυρών γίνεται με την εντολή `serial.Begin()` για την ρύθμιση του baud rate και για την επικοινωνία χρησιμοποιείται η εντολή `serial.println()` για να τυπώσει μία ακολουθία χαρακτήρων στην έξοδο κάποια συνδεδεμένης συσκευής.

Τα ψηφιακά πινς μπορούν να διαβάσουν μόνο δύο καταστάσεις, το 0 και το 1, δηλαδή το δυαδικό σύστημα.

Επίσης υπάρχουν 15 PWM πινς τα οποία είναι από το 2^ο πιν έως το 13 και επίσης τα 44,45 και 46.[18]



Εικόνα 11. Τα pwm πινς στο Arduino mega. [18]

Κάθε ένα από αυτά τα πινς μπορεί παράγει σήμα pwm ανάλυσης 2^3 bit. Η εντολή που χρησιμοποιείται για τη χρήση pwm είναι η `analogWrite()`.

2.3.7 Τα αναλογικά πινς

Πέρα από τα ψηφιακά πινς υπάρχουν η πλακέτα αποτελείται και από 16 αναλογικά πινς, τα οποία χρησιμοποιούν μετατροπέα ADC. Αυτά τα πινς μπορούν να πάρουν τον ρόλο των αναλογικών εισόδων/εξόδων ενώ ταυτόχρονα μπορούν να χρησιμοποιηθούν και ως ψηφιακά πινς. Αυτά τα πινς μπορούν να πάρουν τιμές μεταξύ του 0 και 1023 λόγω του ότι η πλακέτα έχει μετατροπέα ADC 10 bit ο οποίος μετατροπέας λειτουργεί ως εξής, δειγματοληπτεί, κβαντοποιεί και ψηφιοποιεί. Λόγω του ότι η πλακέτα λειτουργεί μεταξύ 0-5 V το βήμα της πλακέτας είναι $5/1023$ δηλαδή 4.88mV. Έτσι οποιαδήποτε τιμή κάτω από τα 4.88mV θεωρείται 0 και πάνω από 4.99V θεωρείται 5V.[18]

2.3.8 Τα υπόλοιπα πινς

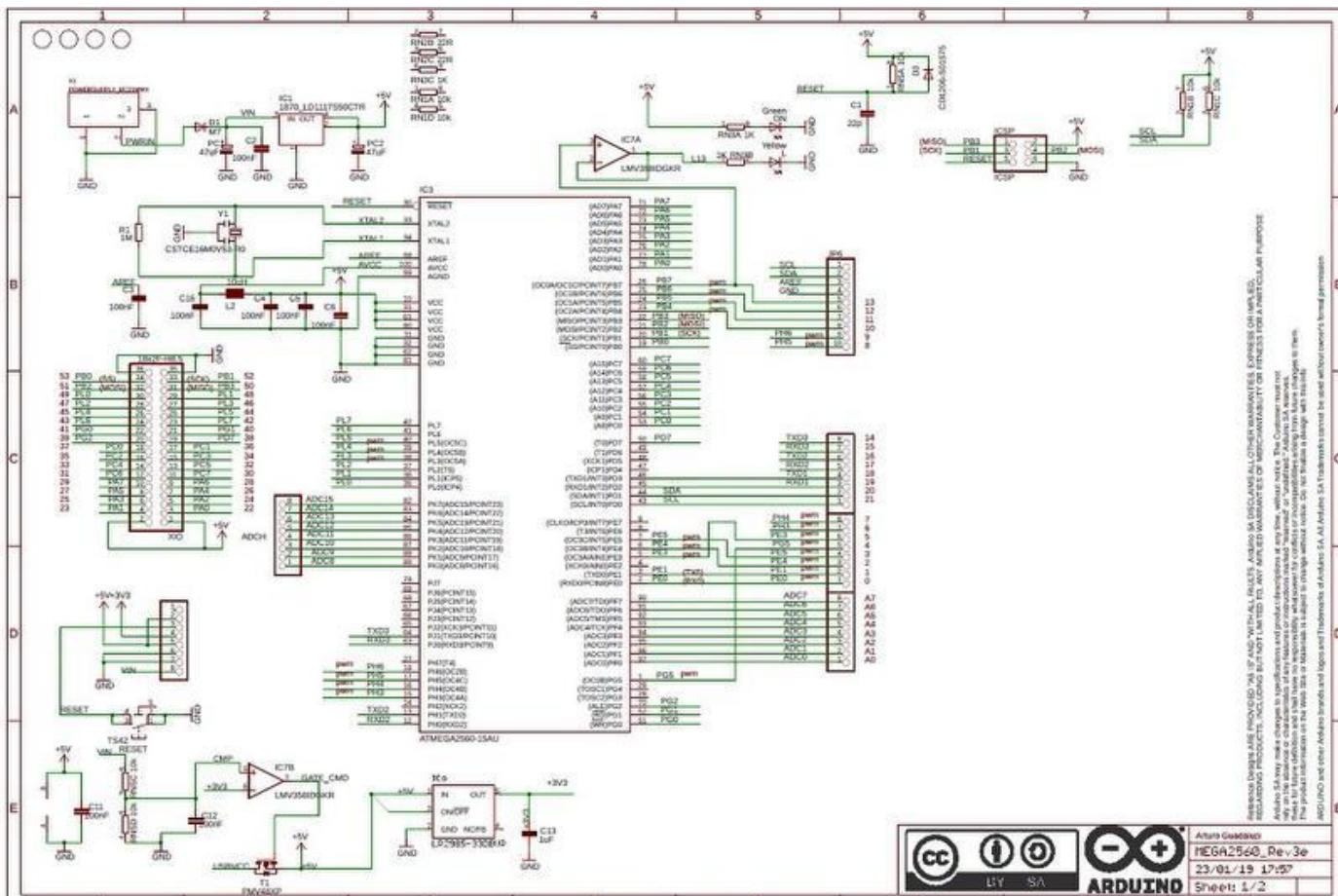
Πέρα από τα πινς που αναφέραμε, η πλακέτα αποτελείται και από κάποια ακόμα πινς όπως το GND, το Reset, το IOREF, το AREF και τα 3,3V και 5V. Αναλύοντας τα, έχουμε αρχικά 7 GND πινς στην πλακέτα τα οποία αναφέρονται στην γείωση, το IOREF δίνει μία τάση αναφοράς στην πλακέτα όπου θα λειτουργεί ο μικροελεγκτής, το AREF δίνει μία τάση αναφοράς όπου θα μετρηθούν οι αναλογικές εισοδοί και τα 3,3V και 5V είναι τάσεις τροφοδοσίας σε εξωτερικές συσκευές-υλικά που συνδέονται σε αυτά τα πινς.[18]

2.3.9 Χαρακτηριστικά

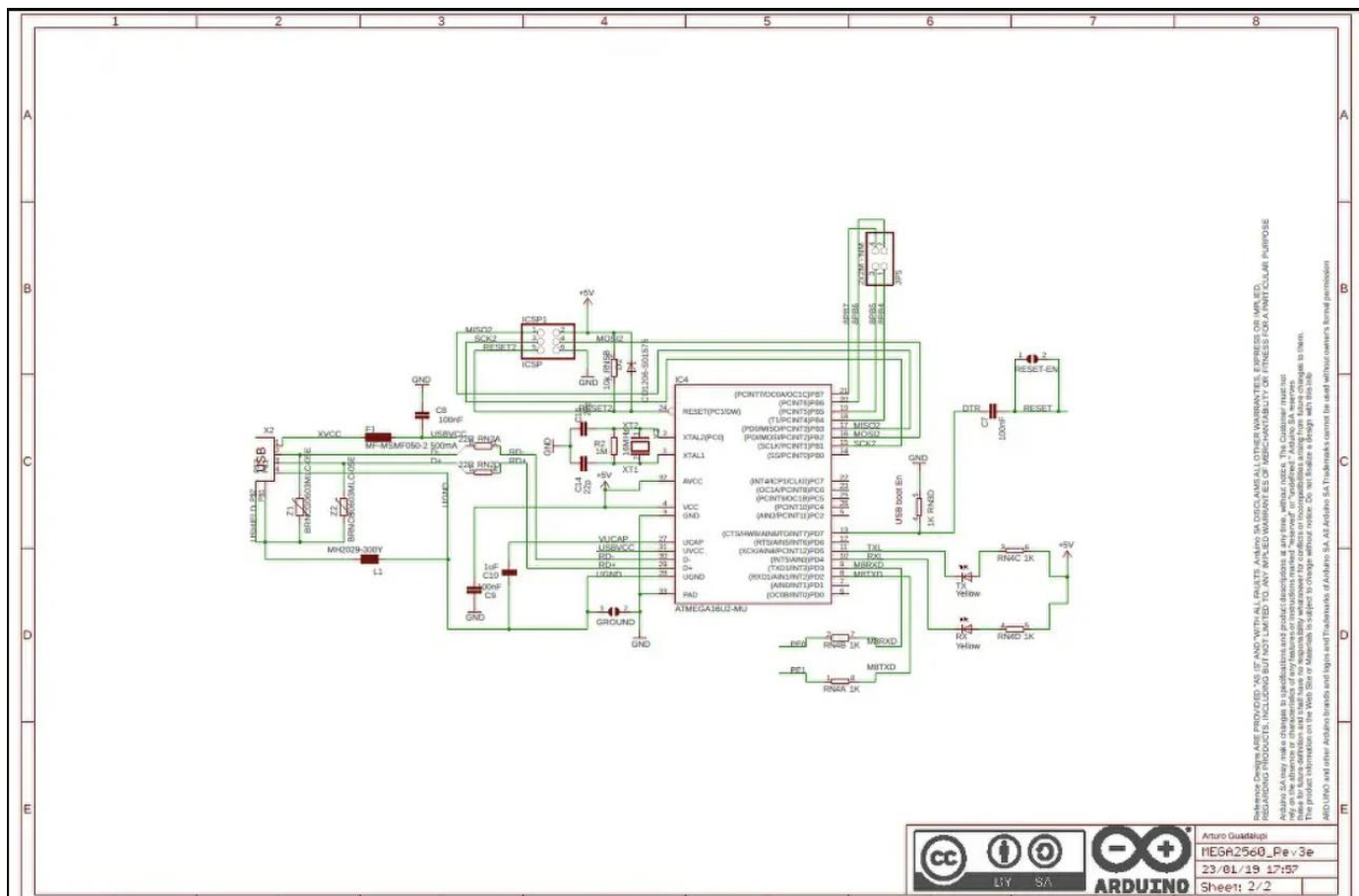
Παρακάτω στον πίνακα φαίνονται όλα τα χαρακτηριστικά του Arduino mega :

Μικροελεγκτής	ATmega2560
Τάση λειτουργίας	5V
Τροφοδοσία	7 V-12 V
Κατανάλωση ρεύματος	50 mA-200 mA
Κατανάλωση ρεύματος σε κατάσταση αδράνειας	500 uA
Ψηφιακά πινς Εισόδων/Εξόδων	54
Ψηφιακά πινς Εισόδων/ Εξόδων με PWM	15
Αναλογικά πινς εισόδων	16
DC ρεύμα για κάθε πιν Εισόδου/Εξόδου	40 mA
DC ρεύμα για το πιν 3,3V	50 mA
Flash Memory	256 kB
SRAM	8 kB
EEPROM	4096 bytes
Clock Speed	16 MHz
Μήκος	102 mm
Πλάτος	53 mm
Power Jack	Ναι
Σύνδεση USB	Ναι

Πίνακας 2 :Πίνακας χαρακτηριστικών Arduino Mega 2560. [18]



Εικόνα 12α: Σχηματικό Arduino Mega 2560. [18]



Εικόνα 12β: Σχηματικό Arduino Mega 2560. [18]

2.4 Προγραμματισμός του Arduino

Για τον προγραμματισμό οποιασδήποτε πλακέτας Arduino απαιτείται ένα αναπτυξιακό περιβάλλον, το Arduino έχει το δικό του ενσωματωμένο αναπτυξιακό περιβάλλον Arduino IDE (Integrated Development Environment). Το Arduino IDE έχει την δυνατότητα ανάπτυξης κώδικα offline κάτι που διευκολύνει την χρήση του Arduino. Το συγκεκριμένο λογισμικό μπορεί να χρησιμοποιηθεί για οποιαδήποτε πλακέτα Arduino, και η χρήση του είναι να ανεβάζει τον κώδικα στην πλακέτα μέσω ενός text editor. Ο κώδικας δημιουργεί κάποια προγράμματα τα οποία έπειτα φορτώνουν στην πλακέτα, αυτά τα προγράμματα ονομάζονται sketches και αποθηκεύονται με τον τύπο αρχείου .ino.[19]

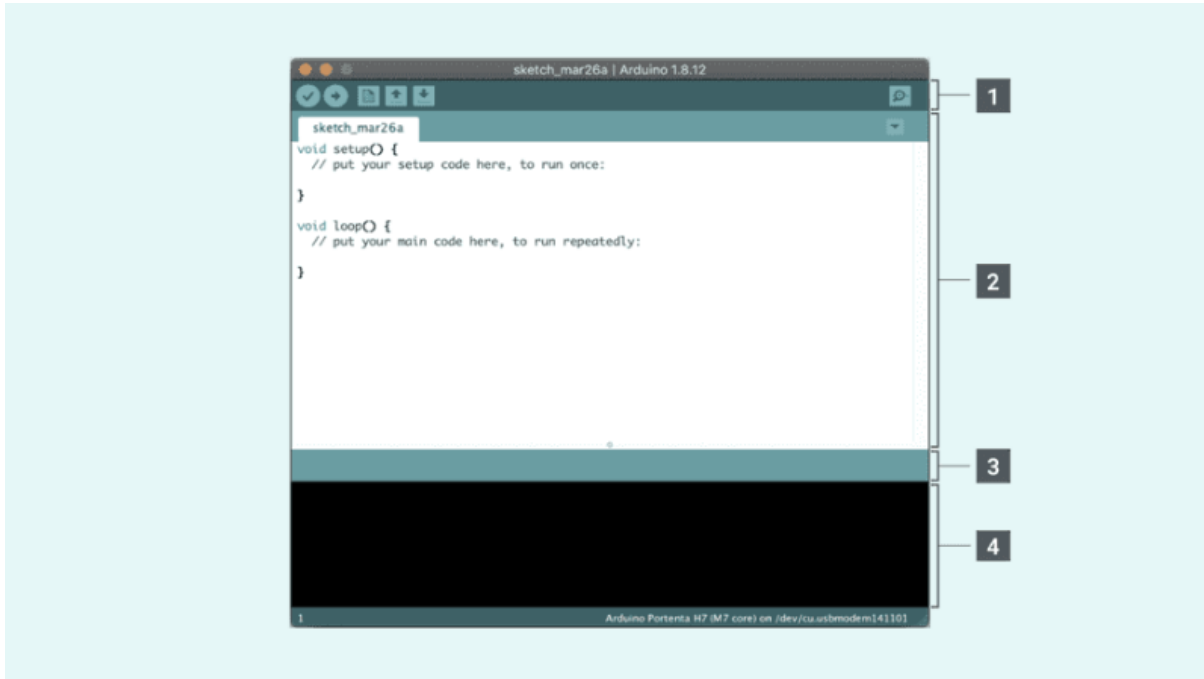
2.4.1 Χρησιμοποιώντας το offline IDE

Ο editor αποτελείται από 4 κύρια πεδία:

- 1) Μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες και μία σειρά από μενού. Τα κουμπιά της γραμμής εργαλείων σου επιτρέπουν να επαληθεύεις και να φορτώνεις προγράμματα, να δημιουργείς, να ανοίγεις και να αποθηκεύεις sketches και να ανοίγεις το σειριακό μόνιτορ.

- 2) Το πεδίο μηνύματος , παρέχει ανάδραση καθώς αποθηκεύει και εξάγει και παράλληλα αναδικνύει τα σφάλματα.
- 3) Ο επεξεργαστής κειμένου για την γραφή κώδικα.
- 4) Η κονσόλα κειμένου απεικονίζει την έξοδο κειμένου από το λογισμικό Arduino , περιλαμβάνοντας ολόκληρα μηνύματα σφαλμάτων και άλλες πληροφορίες.

Όλα αυτά εμφανίζονται στην παρακάτω εικόνα:[19]



Εικόνα 13. Το λογισμικό Arduino IDE.[19]

2.4.2 Η γλώσσα προγραμματισμού Arduino

Το Arduino είναι μία πλατφόρμα open-source η οποία βασίζεται στην γλώσσα προγραμματισμού C++. [19] Η πλατφόρμα αυτή χρησιμοποιείται για τον προγραμματισμό μικροελεγκτών όπως το Arduino Uno και το Arduino Mega έτσι ώστε να αλληλεπιδρούν οι μικροελεγκτές με αισθητήρια, οθόνες και διάφορες άλλες συσκευές. Συνήθως χρησιμοποιείται για πρότζεκτ πάνω στους τομείς της ρομποτικής, των αυτοματισμών σε σπίτια, και το IoT (Internet of Things).

Τα χαρακτηριστικά της γλώσσας προγραμματισμού Arduino είναι τα εξής:

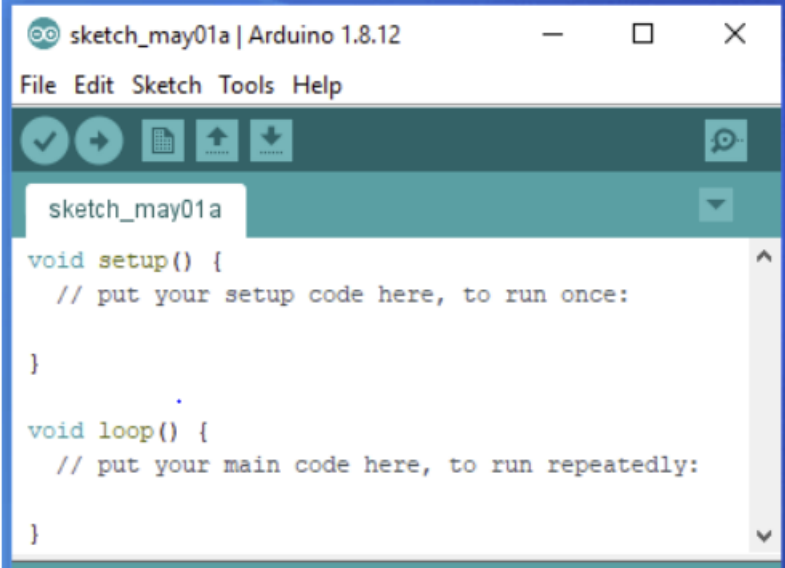
- 1) Είναι μία πλατφόρμα open-source: Τα software και hardware Arduino είναι όλα open-source, κάτι που σημαίνει ότι οι χρήστες έχουν πρόσβαση στον πηγαίο κώδικά και μπορούν να τον τροποποιήσουν σύμφωνα με τα θέλω τους.
- 2) Βασίζεται στην C++: Η γλώσσα προγραμματισμού Arduino βασίζεται στην ευρέως γνωστή γλώσσα C++.
- 3) Έχει μεγάλη βιβλιοθήκη με έτοιμο γραμμένο κώδικα: Το Arduino έχει μεγάλη βιβλιοθήκη με έτοιμο γραμμένο κώδικα κάτι το οποίο απλοποιεί τη χρήση του και κάνει ευκολότερη την αλληλεπίδραση διάφορων συσκευών με την πλακέτα.
- 4) Έχει απλό συντακτικό: Η γλώσσα Arduino έχει από συντακτικό κάτι που το κάνει εύκολα κατανοητό ακόμα και για τους πρωτάρηδες στον κόσμο του προγραμματισμού.
- 5) Έχει συμβατότητα cross-platform: Το λογισμικό Arduino μπορεί να τρέξει είτε σε Windows, είτε σε Linux, είτε σε MacOS. Όποτε απευθύνεται σε όλους τους χρήστες.

- 6) Έχει σειριακή επικοινωνία: Το Arduino έχει ενσωματωμένη σειριακή επικοινωνία κάτι που διευκολύνει την επικοινωνία του με άλλες συσκευές, είτε με Η/Υ μέσω USB.
- 7) Έχει ενσωματωμένη υποστήριξη PWM: Με αυτό το χαρακτηριστικό η πλακέτα έχει εύκολη διαχείριση διανομής ενέργειας σε κάποια συσκευή.[19]

2.4.3 Βασικές αρχές σύνταξης κώδικα Arduino

Το συντακτικό του κώδικα Arduino αποτελείται από κάποιους κανόνες οι οποίοι πρέπει να ακολουθηθούν κατά γράμμα ώστε να τρέξει το πρόγραμμα που έχει γραφτεί, χωρίς σφάλματα. Για να γραφτεί ένας κώδικά χρησιμοποιούνται ορισμένα κομμάτια, αυτά τα κομμάτια εξηγούνται παρακάτω.

Ξεκινώντας έχουμε τα functions. Τα functions συνδυάζουν αρκετά κομμάτια κώδικα σε ένα και δουλειά τους είναι συνήθως να επιστρέφουν κάποια τιμή μετά το πέρας της εκτέλεσης του κώδικα, δύο παραδείγματα functions είναι το **setup()** και το **loop()** και παρακάτω φαίνεται ένα παράδειγμα του πως χρησιμοποιούνται στην **Εικόνα 14**. [20]



```
sketch_may01a | Arduino 1.8.12
File Edit Sketch Tools Help
sketch_may01a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Εικόνα 14. Παράδειγμα χρήσης των functions μέσα στον κώδικα.[20]

Ο κώδικάς που αποτελεί την function πρέπει να γράφεται μέσα σε αγκύλες για τις οποίες για κάθε αγκύλη που ανοίγει πρέπει να υπάρχει στο τέλος του κώδικα και μία που να την κλείνει. Για να ελεγχθεί ένας κώδικάς ότι τρέχει σωστά υπάρχει το κουμπί σύνταξης (επάνω αριστερά στην **Εικόνα 14**), το οποίο τρέχει τον κώδικα χωρίς να τον φορτώσει στην πλακέτα.

Κάθε function μπορεί να αποτελείται από παρενθέσεις μέσα στις οποίες μπαίνουν οι παράμετροι του function. Έπειτα, όπως και στις γλώσσες προγραμματισμού C και C++ για την λήξη μίας γραμμής κώδικα χρησιμοποιείται το σύμβολο ";" .[20]

2.5 Επίλογος

Σε αυτό το κεφάλαιο αναλύσαμε την αναπτυξιακή πλακέτα Arduino, είδαμε επίσης το περιβάλλον ανάπτυξης κώδικα καθώς και βασικές εντολές που χρησιμοποιούνται για την ανάπτυξη κώδικα. Υπήρξε ανάλυση των υλικών της πλακέτας και παράλληλα επεξήγηση της χρήσης τους.

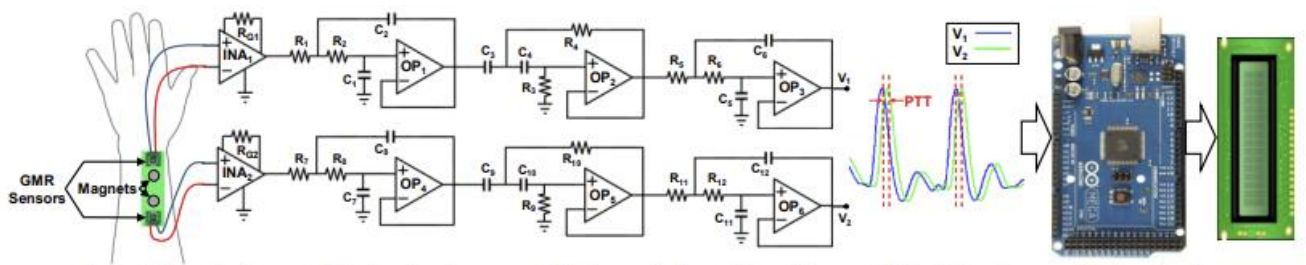
Κεφάλαιο 3ο : Ανάλυση του κυκλώματος πιεσόμετρου με την μέθοδο της μαγνητοπλυθησογραφίας

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλυθεί το κατασκευαστικό και σχεδιαστικό κομμάτι της κατασκευής. Θα αναλυθούν τα υλικά, οι μαθηματικοί τύποι και το λογισμικό που χρησιμοποιήθηκε καθώς και ο κώδικας που πλαισίωσε το κατασκευαστικό κομμάτι.

3.2 Σχηματικό διάγραμμα

Παρακάτω φαίνεται το σχηματικό διάγραμμα της κατασκευής. Από αυτό το διάγραμμα φαίνονται οι συνδεσμολογίες που χρησιμοποιήθηκαν ώστε να ολοκληρωθεί η κατασκευή. Οπότε μπορεί να γίνει πιο εύκολα κατανοητή η λειτουργία των κυκλωμάτων και ποιά θα είναι η σειρά σύνδεσης εισόδων-εξόδων του κάθε κυκλώματος. Η κατασκευή αποτελείται από δύο κομμάτια, το υλικό και το λογισμικό, στο υλικό βρίσκονται όλες οι συνδεσμολογίες που χρησιμοποιήθηκαν, και στο λογισμικό βρίσκεται ο κώδικας που χρησιμοποιήθηκε στον οποίο βρίσκονται μέσα οι μαθηματικοί τύποι που χρειάστηκαν για την ανάλυση της εξόδου του κυκλώματος.

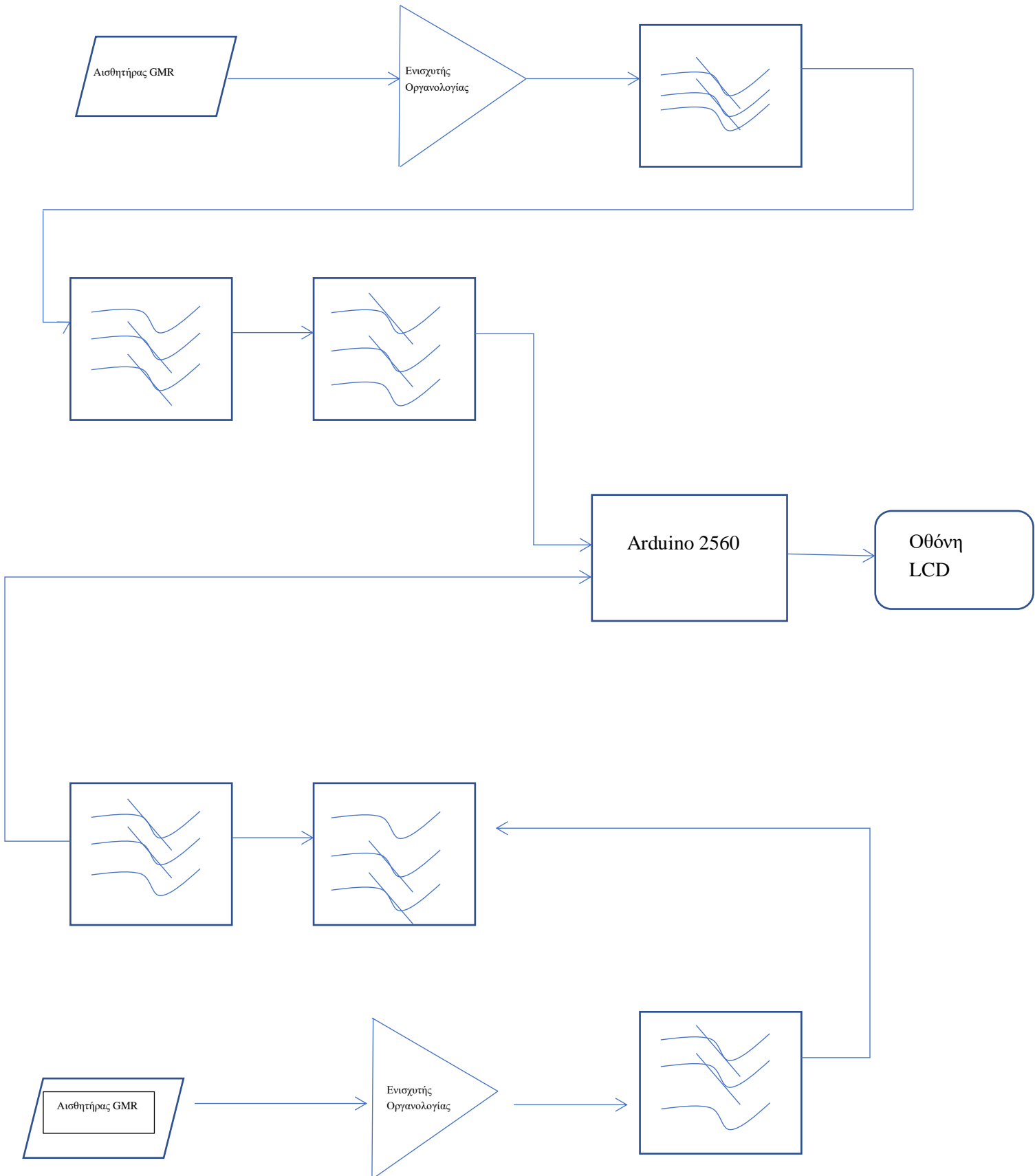


Σχήμα 1. Σχηματικό διάγραμμα όπου φαίνονται τα κυκλώματα και η επεξεργασία που χρειάζεται για τη σωστή λειτουργία της κατασκευής. [21]

3.3 Μπλοκ διάγραμμα

Για την ευκολότερη κατανόηση του κυκλώματος μπορεί να χρησιμοποιηθεί ένα μπλοκ διάγραμμα το οποίο δείχνει την λειτουργία του κυκλώματος ανα βαθμίδα.

Ανάλυση του κυκλώματος πιεσομέτρου με την μέθοδο της μαγνητοπυθσημογραφίας

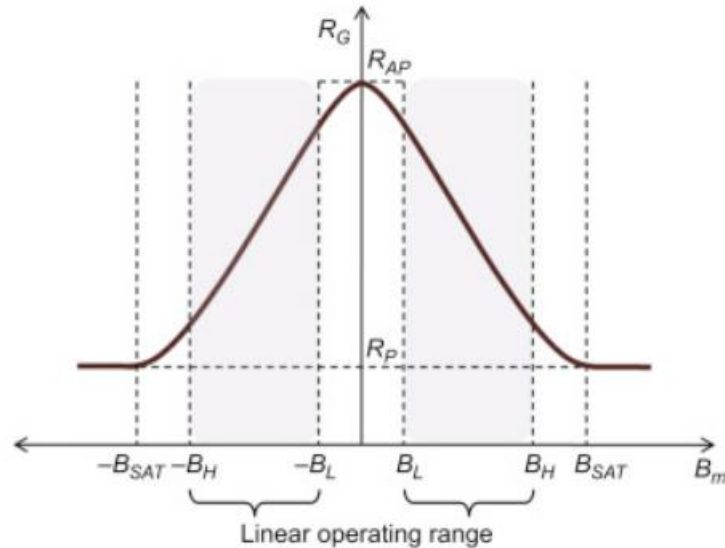


Σχήμα 2. Μπλοκ διάγραμμα ροής του κυκλώματος πιεσομέτρου με την μέθοδο της μαγνητοπυθσημογραφίας.

Όπως γίνεται εύκολα κατανοητό από το παράνω διάγραμμα, για την σωστή λειτουργία του κυκλώματος χρειάζεται ένα αναλογικό σήμα, στην περίπτωση μας αυτό που θα λάβει ο αισθητήρας GMR από την κερκιδική αρτηρία, έπειτα αυτό το σήμα θα περάσει από έναν ενισχυτή οργανολογίας για να ενισχυθεί σε μεγάλο βαθμό ώστε να είναι ευκόλως επεξεργάσιμο, κάτι το οποίο θα προσθέσει θόρυβο στο σήμα. Για την εξάλειψη αυτού του θορύβου θα χρησιμοποιηθεί ένα χαμηλοπερατό φίλτρο 2^{ης} τάξης σε συνδεσμολογία Sallen key ώστε να απορρίψει τον οποιοδήποτε θόρυβο με συχνότητα μεγαλύτερη από τα 10 Hz. Έπειτα, επειδή το σήμα που έχει ληφθεί από τον αισθητήρα περιλαμβάνει και πληροφορία του αναπνευστικού ρυθμού και αυτό θα χρειαστεί να εξαληφθεί, χρησιμοποιείται ένα φίλτρο επίσης 2^{ης} τάξης σε συνδεσμολογία Sallen Key το οποίο αυτή τη φορά θα είναι υπερπερατό με συχνότητα αποκοπής το 0.7 Hz. Και τελικά χρησιμοποιείται και ένα ακόμα χαμηλοπερατό φίλτρο επίσης σε συνδεσμολογία Sallen Key 2^{ης} τάξης για την εξάλειψη οποιουδήποτε άλλου θορύβου. Το σήμα που λαμβάνεται από αυτήν την επεξεργασία καταλήγει στο Arduino για να χρησιμοποιηθεί καταλλήλως για τις ανάγκες του συστήματος με την ένδειξη της πίεσης να οπτικοποιείται για τον χρήστη σε μία οθόνη LCD.

3.4 Αισθητήρας GMR

Ο αισθητήρας που χρησιμοποιείται για την λήψη του σήματος προς επεξεργασία από την κερκιδική αρτηρία είναι ένας αισθητήρας GMR (Giant Magnetoresistance). Αυτός είναι ένας αισθητήρας ο οποίος αποτελείται από δύο σιδηριμαγνητικά στρώματα χωρισμένα από ένα στρώμα μη μαγνητικό αλλά αγωγίμο. Ο σιδηριμαγνητισμός είναι ένας τύπος μόνιμου μαγνητισμού στερεών οφειλόμενος στην αυθόρμητη διεύθυνση των ατόμων και ιόντων. Τα σιδηριμαγνητικά υλικά είναι λιγότερο μαγνητικά από τα σιδηρομαγνητικά. Με απλά λόγια ο αισθητήρας αυτός λειτουργεί με μία εξίσωση αντίστασης και μαγνητικού πεδίου καθώς όσο μεγαλύτερο το μαγνητικό πεδίο του αισθητήρα, τόσο μικρότερη η αντίσταση και ευκολότερη η ροή ηλεκτρονίων μεταξύ των δύο σιδηριμαγνητικών στρωμάτων, η λειτουργία του GMR αισθητήρα είναι γραμμική και φαίνεται στην παρακάτω **Εικόνα 15**. [21]



Εικόνα 15. Χαρακτηριστική καμπύλη της συμπεριφοράς της αντίστασης R_G του αισθητήρα GMR σε εξίσωση με το μαγνητικό πεδίο.[21]

Σε αυτήν την πτυχιακή ο αισθητήρας που επιλέχθηκε είναι ο AA002-02 ο οποίος λειτουργεί με μαγνήτες δίπλα του ώστε να δημιουργείται το οφέλιμο μαγνητικό πεδίο και ο αισθητήρας να λειτουργεί στην γραμμική περιοχή του. Τα χαρακτηριστικά του αισθητήρα αυτού φαίνονται στην **Εικόνα 16.**

Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply voltage	AAxxx/ABxxx/AAL002		24	Volts
	AAHxxx/AKxxx/ABHxxx/AAL004/AAL024		12	
Operating temperature	AAxxx/AKxxx/ABxxx/AALxxx	-50	125	°C
	AAHxxx/ABHxxx		150	°C
Storage temperature	AAxxx/AKxxx/ABxxx/AALxxx	-65	135	°C
	AAHxxx/ABHxxx	-65	150	
ESD (Human Body Model)			400	Volts
Applied magnetic field	H		Unlimited	Tesla
Voltage from sensor connections to center pad (applies to TDFN package only)			63	Volts DC

Εικόνα 16. Οι μέγιστες τιμές λειτουργίας του αισθητήρα GMR AA002-02.[22]

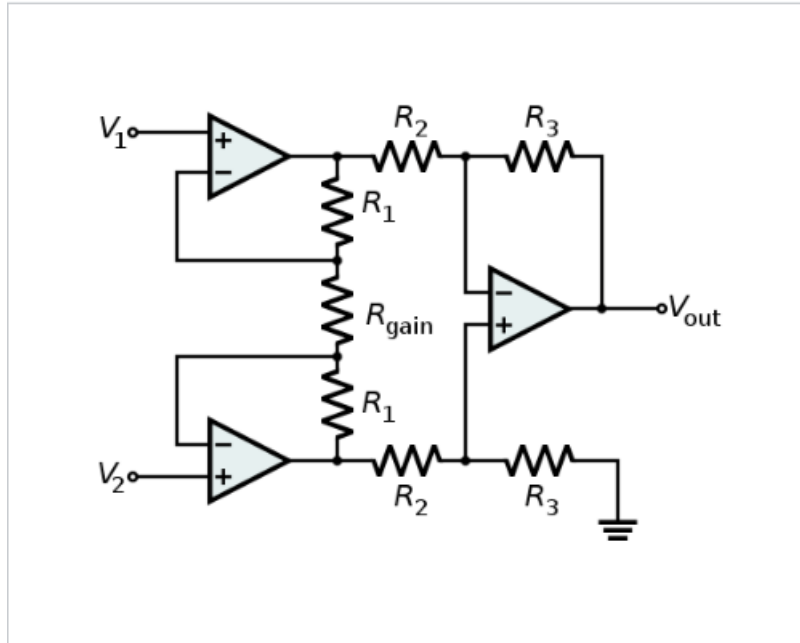
Operating Specifications

Parameter	Symbol	Min.	Typ.	Max.	Units	Test Condition
Supply voltage	AAHxxx/AAKxxx/ ABHxxx/AAL004	V_{CC}	<1	12	Volts	Maximum limited by power dissipation
	AAxxx/ABxxx/AAL002			24		
Operating temperature	AAKxxx	T_{MIN}^* T_{MAX}	-40	85	°C	
	AAxxx/ABxxx/AALxxx			125		
	AAHxxx/ABHxxx			150		
Electrical offset	AAxxx/AAKxxx/AALxxx/ABxxx	V_O	-4	+4	mV/V	
	AAHxxx/ABHxxx			+5		
Output at maximum field	AAxxx/ABxxx	$V_{OUT-MAX}$		60	mV/V	
	AAHxxx/ABHxxx			40		
	AAKxxx			19		
	AALxxx			45		
Nonlinearity	AAxxx/AAKxxx/ABxxx/AAL002			2	%	Unipolar field sweep
	AAHxxx/ABHxxx/AAL0x4			4		
Hysteresis	AAHxxx/ABHxxx			15	%	
	AAxxx/AAKxxx/ABxxx			4		
	AALxxx			2		
Resistance tolerance			-20	+20	%	25°C
Resistance vs. temperature	AAxxx/ABxxx	TC_R		+0.14	%/°C	No applied field
	AAHxxx/AAKxxx/ AALxxx/ABHxxx			+0.11		
Output temperature coefficient	AAxxx/ABxxx	TC_{O-I}		+0.03	%/°C	Constant-current supply
	AAHxxx/ABHxxx			-0.28		
	AAKxxx			+0.13		
	AALxxx			-0.28		
Output temperature coefficient	AAxxx/ABxxx	TC_{O-V}		-0.1	%/°C	Constant-voltage supply
	AAHxxx/ABHxxx			-0.40		
	AAKxxx			-0.3		
	AALxxx			-0.4		
	AAKxxx	TC_{HSAT}		-0.19	%/°C	
Frequency bandwidth	AAKxxx	f_{MAX}	DC	50	kHz	-3 dB bandwidth
	AAxxx/AAHxxx			75		
	AALxxx			500		
	ABxxx/ABHxxx			1		
Junction– Ambient thermal resistance	ULLGA4 (-14 suffix)	θ_{JA}		500	°C/W	Soldered to double-sided board; free air
	TDFN6 (-10 suffix)			320		
	MSOP8 (-00 suffix)			320		
	SOIC8 (-02 suffix)			240		
Power Dissipation	ULLGA4 (-14 suffix)	P_D		100	mW	
	TDFN6 (-10 suffix)			500		
	MSOP8 (-00 suffix)			500		
	SOIC8 (-02 suffix)			675		

Εικόνα 17. Τυπικές τιμές λειτουργίας του αισθητήρα GMR AA002-02.[22]

3.5 Συνδεσμολογία ενισχυτή οργανολογίας

Οι ενισχυτές οργανολογίας είναι κυκλώματα ακριβείας που έχουν μεγάλη αντίσταση εισόδου, μικρή αντίσταση εξόδου, υψηλό λόγο απόρριψης κοινού τρόπου (CMRR), μεγάλο ρυθμό μεταβολής, ακριβής και σταθερό κέρδος και ευκολότερη ρύθμιση του κέρδους. Συνήθως χρησιμοποιείται για να ενισχύσει χαμηλά σήματα λόγω του μεγάλου κέρδους του και να απορρίπτει τον θόρυβο και τα σήματα που παρεμβάλουν.[23]



Εικόνα 18. Ένας τυπικός ενισχυτής οργανολογίας.[24]

Για τον υπολογισμό της απολαβής του ενισχυτή παίρνουμε τον παρακάτω μαθηματικό τύπο:

$$A_v = \frac{V_{out}}{V_2 - V_1} = \left(1 + \frac{2R_1}{R_{gain}}\right) \frac{R_3}{R_2}$$

[24]

Τα χαρακτηριστικά ενός καλού ενισχυτή οργανολογίας αναλύονται παρακάτω:

- Πεπερασμένο, ακριβές και σταθερό κέρδος: Καθώς ο ενισχυτής αυτός χρησιμοποιείται για να ενισχύσει χαμηλά σήματα, απαιτείται να προσφέρει υψηλό και πεπερασμένο κέρδος, καθώς και να είναι ακριβές το κέρδος του και σταθερό στον κλειστό βρόχο.
- Εύκολη ρύθμιση κέρδους: Η ρύθμιση του κέρδους είναι κάτι το οποίο είναι αναγκαίο σε τέτοιου τύπου ενισχυτή και ένας καλός ενισχυτής οργανολογίας πρέπει να το κάνει όσο πιο εύκολα γίνεται.
- Υψηλή αντίσταση εισόδου: Για την αποφυγή της φόρτωσης των πηγών εισόδου, η αντίσταση εισόδου πρέπει να είναι πολύ υψηλή, ιδανικά άπειρη.
- Χαμηλή αντίσταση εξόδου: Για την αποφυγή της φόρτωσης του επόμενου κυκλώματος, η αντίσταση εξόδου θα πρέπει να είναι πολύ χαμηλή, ιδανικά μηδενική.
- Υψηλό CMRR: Η έξοδος ενός μετατροπέα συνήθως περιέχει σήματα κοινού τρόπου, όταν μεταδίδονται μέσω μεγάλων καλωδίων. Ένας καλός ενισχυτής οργανολογίας πρέπει να τα απορρίπτει αυτά τα σήματα και να ενισχύει μόνο την διαφορική είσοδο. Κοινώς ιδανικά το CMRR ενός τέτοιου ενισχυτή ιδανικά είναι άπειρο.
- Μεγάλο ρυθμό μεταβολής: Για την παροχή ενός σήματος εξόδου χωρίς παραμόρφωση ταλάντευσης τάσης ο ενισχυτής πρέπει να έχει υψηλό ρυθμό μεταβολής. [23]

Στις ανάγκες της πτυχιακής αυτής επιλέχθηκε ο ενισχυτής οργανολογίας INA129 ο οποίος έχει τα παρακάτω χαρακτηριστικά:

7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)⁽¹⁾

		MIN	MAX	UNIT
V _S	Supply voltage	Dual supply, V _S = (V+) – (V–)	±18	V
		Single supply, V _S = (V+) – 0 V	36	
	Analog input voltage		±40	V
	Output short-circuit ⁽²⁾	Continuous		
T _A	Operating temperature	–40	125	°C
	Junction temperature		150	°C
	Lead temperature (soldering, 10 s)		300	°C
T _{stg}	Storage temperature	–55	125	°C

Εικόνα 19. Οι μέγιστες τιμές λειτουργίας του ενισχυτή οργανολογίας INA129.[25]

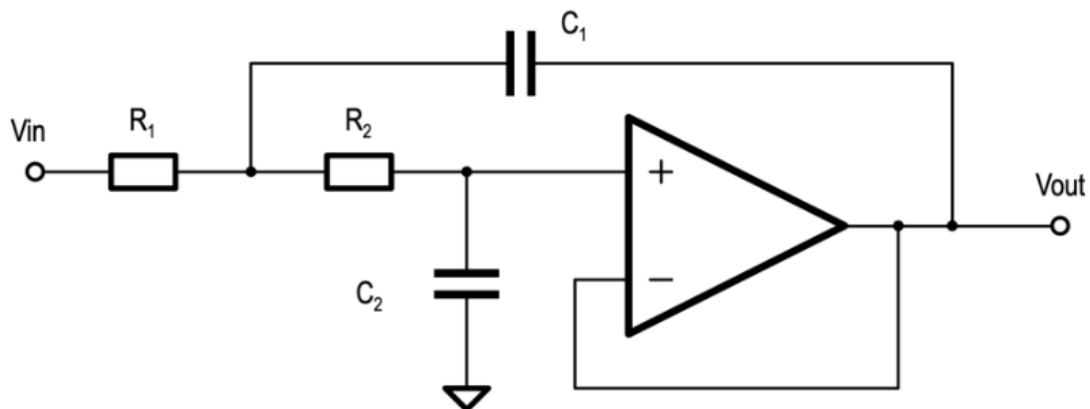
7.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	TYP	MAX	UNIT	
V _S	Supply voltage	Single-supply	4.5	30	36	V
		Dual-supply	±2.25	±15	±18	
	Input common-mode voltage range for V _O = 0 V	(V–) + 2		(V+) – 2	V	
T _A	Specified temperature	–40		85	°C	

Εικόνα 20. Οι προτεινόμενες από τον κατασκευαστή συνθήκες λειτουργίας του ενισχυτή οργανολογίας INA129.[25]

3.6 Συνδεσμολογία Sallen-Key φίλτρου

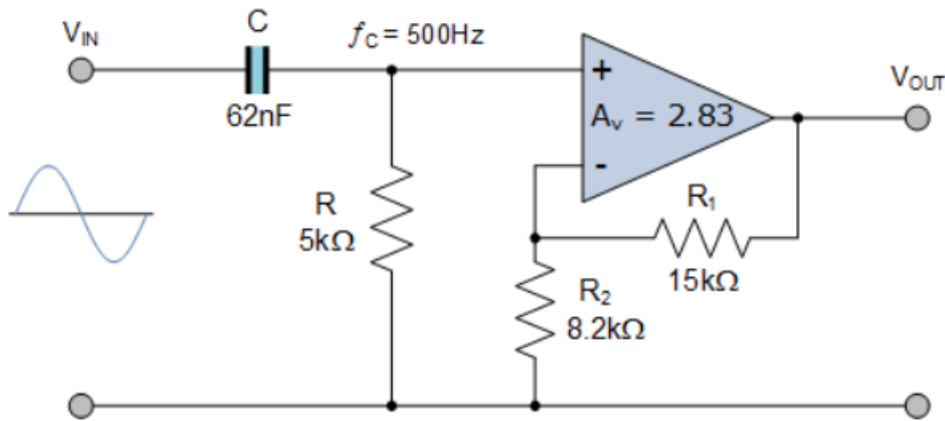


Εικόνα 21. Συνδεσμολογία Sallen-Key χαμηλοπερατού φίλτρου.[26]

Στην περίπτωση αυτής της εργασίας χρησιμοποιούνται 3 φίλτρα Sallen-Key για κάθε κύκλωμα, δηλαδή σε σύνολο 6 φίλτρα. Από αυτά τα 6 φίλτρα είναι 4 χαμηλοπερατά και 2 υψιπερατά. Καθώς τα 2 κυκλώματα είναι το ένα καθρέφτης του άλλου, κάτι που σημαίνει είναι το ίδιο κύκλωμα,

αναλύοντας το ένα κύκλωμα έχουμε πλήρη εικόνα της λειτουργίας του άλλου. Τα φίλτρα αυτά τοποθετούνται στην έξοδο του ενισχυτή οργανολογίας.

Η συνδεσμολογία του χαμηλοπερατού και του υψιπερατού φίλτρου διαφέρουν ελάχιστα, στην **Εικόνα 21** φαίνεται ένα χαμηλοπερατό φίλτρο. Η συνδεσμολογία του υψιπερατού φαίνεται στην **Εικόνα 22**.



Εικόνα 22. Συνδεσμολογία υψιπερατού φίλτρου Sallen-Key.[27]

Η εξίσωση της συχνότητας αποκοπής για τα φίλτρα Sallen-Key είναι:

$$f_c = \frac{1}{2\pi\sqrt{R_A R_B C_A C_B}}$$

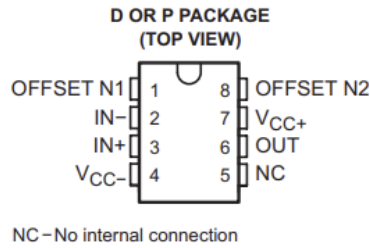
[27]

Και έτσι αν ισχύει $C_A=C_B$ και $R_A=R_B$ τότε η εξίσωση απλοποιείται ως:

$$f_c = \frac{1}{2\pi RC}$$

[27]

Σε αυτή την πτυχιακή έχει επιλεγθεί ο τελεστικός ενισχυτής OP07 για τη δημιουργία των φίλτρων Sallen-Key. Τα χαρακτηριστικά του οποίου φαίνονται παρακάτω.



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
IN+	3	I	Noninverting input
IN-	2	I	Inverting input
NC	5	—	Do not connect
OFFSET N1	1	I	External input offset voltage adjustment
OFFSET N2	8	I	External input offset voltage adjustment
OUT	6	O	Output
V _{CC+}	7	—	Positive supply
V _{CC-}	4	—	Negative supply

Εικόνα 23. Πιν διάγραμμα του ολοκληρωμένου κυκλώματος T.E. OP07.[28]

7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)⁽¹⁾

		MIN	MAX	UNIT
V _{CC+} ⁽²⁾	Supply voltage	0	22	V
V _{CC-} ⁽²⁾		-22	0	
	Differential input voltage ⁽³⁾		±30	V
V _I	Input voltage range (either input) ⁽⁴⁾		±22	V
	Duration of output short circuit ⁽⁵⁾		Unlimited	
T _J	Operating virtual-junction temperature		150	°C
	Lead temperature 1.6 mm (1/16 in) from case for 10 s		260	°C

Εικόνα 24. Οι μέγιστες τιμές λειτουργίας του T.E. OP07. [28]

7.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
V _{CC+}	Supply voltage	3	18	V
V _{CC-}		-3	-18	
V _{IC}	Common-mode input voltage	-13	13	
		V _{CC±} = ±15 V		
T _A	Operating free-air temperature	0	70	°C

Εικόνα 25. Στον πίνακα φαίνονται οι προτεινόμενες τιμές λειτουργίας του T.E. OP07. [28]

δ

7.5 Electrical Characteristics

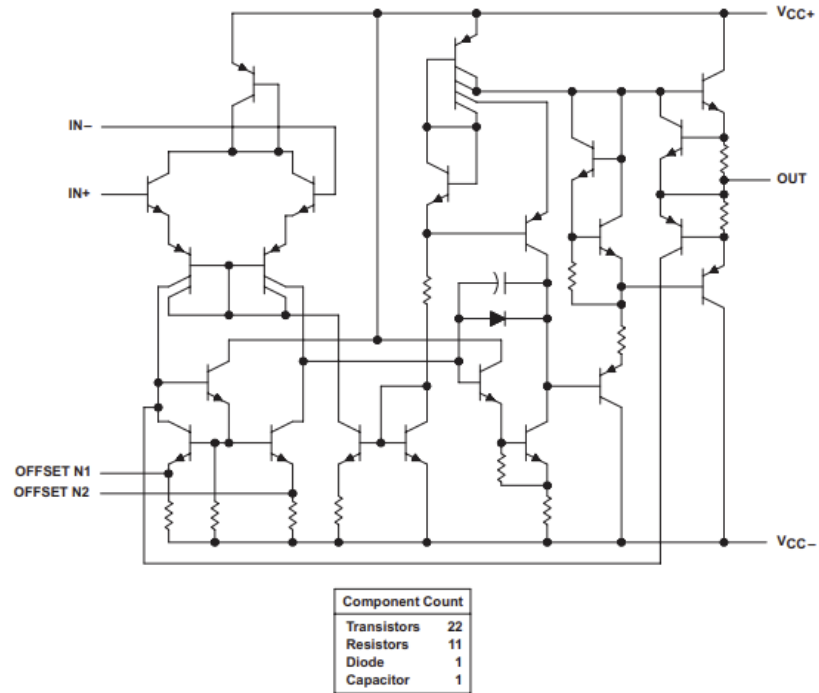
at specified free-air temperature, $V_{CC\pm} = \pm 15 \text{ V}$ (unless otherwise noted)⁽¹⁾

PARAMETER	TEST CONDITIONS	$T_A^{(2)}$	OP07C			OP07D			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
V_{IO}	Input offset voltage	$V_O = 0 \text{ V}$	$R_S = 50 \ \Omega$	25°C	60		150		μV
				0°C to 70°C	85		250		
α_{VIO}	Temperature coefficient of input offset voltage	$V_O = 0 \text{ V}$	$R_S = 50 \ \Omega$	0°C to 70°C		0.5		2.5	$\mu\text{V}/^\circ\text{C}$
	Long-term drift of input offset voltage	See				0.4		$\mu\text{V}/\text{mo}$	
	Offset adjustment range	$R_S = 20 \ \text{k}\Omega$	See Figure 2		25°C		± 4		mV
I_{IO}	Input offset current			25°C	0.8		6		nA
				0°C to 70°C	1.6		8		
α_{IIO}	Temperature coefficient of input offset current			0°C to 70°C		12		50	$\text{pA}/^\circ\text{C}$
I_{IB}	Input bias current			25°C	± 1.8		± 12		nA
				0°C to 70°C	± 2.2		± 14		
α_{IIB}	Temperature coefficient of input bias current			0°C to 70°C		18		50	$\text{pA}/^\circ\text{C}$
V_{ICR}	Common-mode input voltage range			25°C	± 13	± 14	± 13	± 14	V
				0°C to 70°C	± 13	± 13.5	± 13	± 13.5	
V_{OM}	Peak output voltage	$R_L \geq 10 \ \text{k}\Omega$	$R_L \geq 2 \ \text{k}\Omega$	25°C	± 12	± 13	± 12	± 13	V
					± 11.5	± 12.8	± 11.5	± 12.8	
					± 12		± 12		
					0°C to 70°C	± 11	± 12.6	± 11	
A_{VD}	Large-signal differential voltage amplification	$V_{CC} = 15 \text{ V}, V_O = 1.4 \text{ V to } 11.4 \text{ V}, R_L \geq 500 \ \text{k}\Omega$	$V_O = \pm 10, R_L = 2 \ \text{k}\Omega$	25°C	100	400	400		V/mV
				25°C	120	400	120	400	
				0°C to 70°C	100	400	100	400	
B_1	Unity-gain bandwidth			25°C	0.4	0.6	0.4	0.6	MHz
r_i	Input resistance			25°C	8	33	7	31	M Ω
CMRR	Common-mode rejection ratio	$V_{IC} = \pm 13 \text{ V}, R_S = 50 \ \Omega$		25°C	100	120	94	110	dB
				0°C to 70°C	97	120	94	106	
k_{SVS}	Supply-voltage sensitivity ($\Delta V_{IO}/\Delta V_{CC}$)	$V_{CC\pm} = \pm 3 \text{ V to } \pm 18 \text{ V}, R_S = 50 \ \Omega$		25°C	7	32	7	32	$\mu\text{V}/\text{V}$
				0°C to 70°C	10	51	10	51	
P_D	Power dissipation	$V_O = 0, \text{ No load}$		25°C	80	150	80	150	mW
					$V_{CC\pm} = \pm 3 \text{ V}, V_O = 0, \text{ No load}$		4	8	

Εικόνα 26. Τα ηλεκτρικά χαρακτηριστικά του T.E. OP07 μέσα από το datasheet της TI (Texas Instruments).[28]

Στις παραπάνω εικόνες φαίνονται κάποια από τα χαρακτηριστικά του T.E. OP07 τα οποία πάρθηκαν μέσα από τα datasheet της Texas Instruments, για την σωστή τοποθέτηση του και λειτουργία στο κύκλωμα, πέρα από αυτά στην **Εικόνα 27** φαίνεται και το κύκλωμα που υπάρχει μέσα στο ολοκληρωμένο κύκλωμα του OP07:

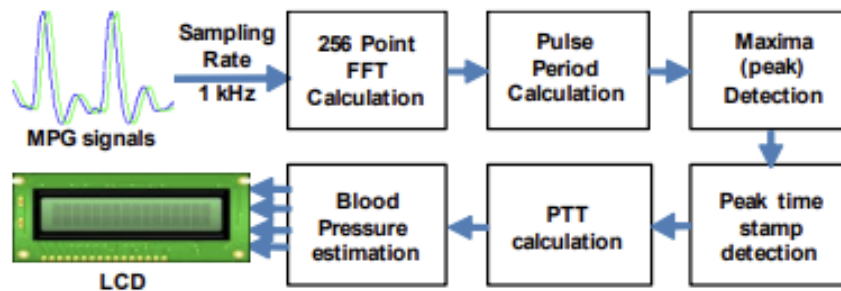
9.2 Functional Block Diagram



Εικόνα 27. Μπλοκ διάγραμμα του κυκλώματος που αποτελείται ο T.E. OP07. [28]

3.7 Ο αλγόριθμος υπολογισμού της αρτηριακή πίεσης

Έχοντας πάρει ο αισθητήρας GMR τους παλμούς και περνώντας από το κύκλωμα conditioning έτσι ώστε να έρθει η έξοδος στις ανάγκες μας ώστε να την επεξεργαστούμε καταλλήλως, η έξοδος του κυκλώματος πρέπει να περάσει μέσα από έναν αλγόριθμο ώστε να υπολογιστεί η αρτηριακή πίεση.



Εικόνα 28. Μπλοκ διάγραμμα από τα στάδια που περνάει ο αλγόριθμος για την εμφάνιση της αρτηριακής πίεσης.[21]

Τα στάδια όπως φαίνεται και στην **Εικόνα 25** είναι τα εξής:

- Υπολογισμός Fast Fourier Transform (FFT) 256 σημείων.
- Υπολογισμός περιόδου παλμού.
- Ανίχνευση κορυφής.
- Ανίχνευση χρονοσήμανσης κορυφής.
- Υπολογισμός χρόνου διέλευσης παλμού.
- Υπολογισμός αρτηριακής πίεσης.

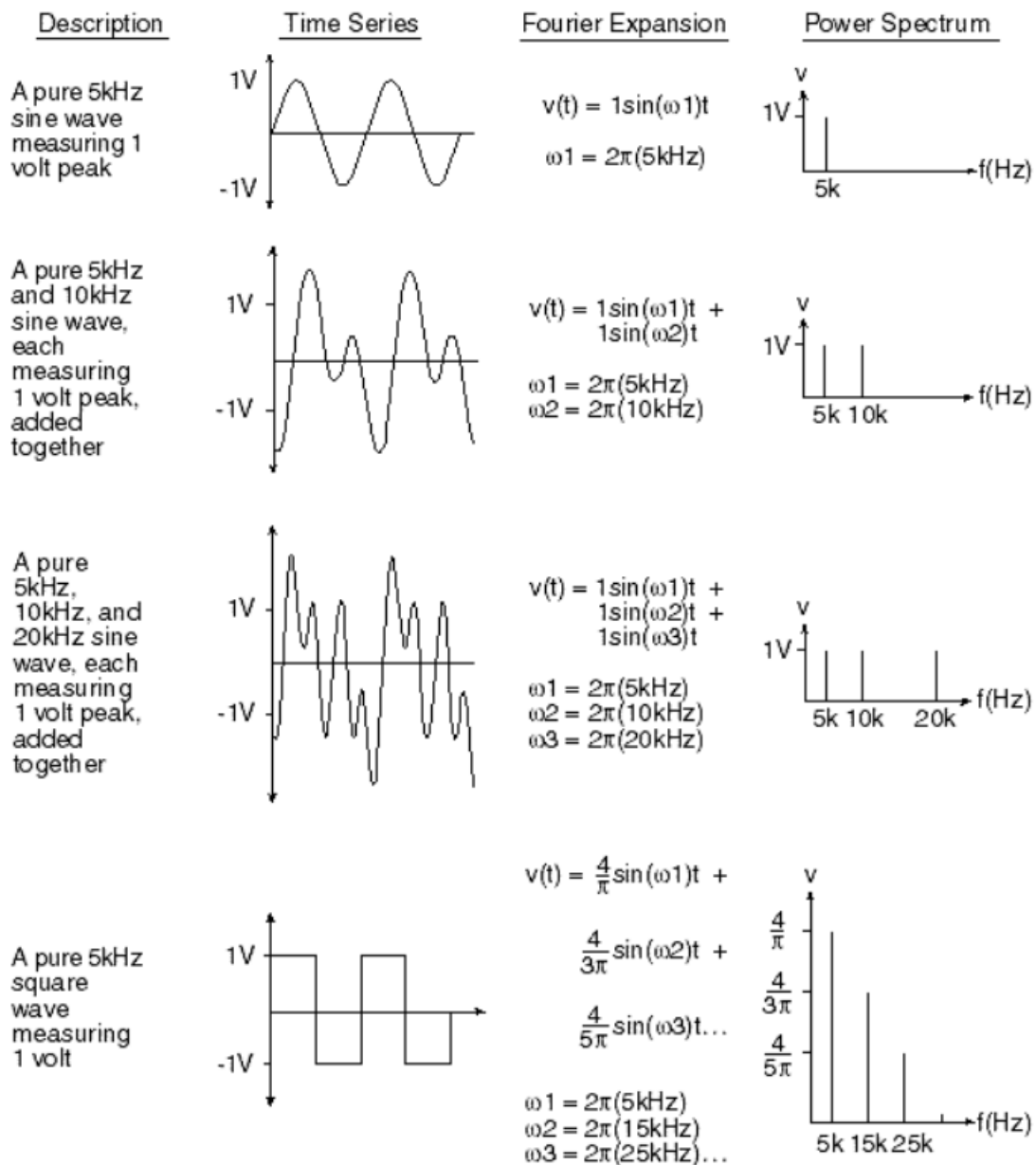
Και έπειτα έχοντας περάσει όλα αυτά τα στάδια τα οποία είναι εργασία της πλακέτας του Arduino να τα εκτελέσει μέσω κάποιο κώδικά, εν τέλει καταλήγει να εμφανίζεται ένδειξη της αρτηριακής πίεσης στην οθόνη LCD η οποία συνδέεται στο Arduino. Το κάθε μέρος του αλγορίθμου αυτού θα αναλυθεί παρακάτω.

3.7.1 Υπολογισμός FFT

Ο αλγόριθμος FFT είναι μία μαθηματική πράξη η οποία αλλάζει το πεδίο ενός σήματος από χρόνο σε συχνότητα. Αυτό είναι χρήσιμο καθώς μπορεί να αποσυνθέσει ένα σήμα το οποίο αποτελείται από πολλές καθαρές συχνότητες.

Η ανάλυση Fourier και ο διακριτός μετασχηματισμός Fourier (DFT) είναι πολύ μεγάλος παράγοντας στην επεξεργασία σήματος. Ο αλγόριθμος FFT μειώνει τον υπολογισμό ενός DFT μήκους N από τάξη N^2 σε τάξη $N \log_2 N$ πράξεων όταν το N είναι δύναμη του 2. Ο FFT πετυχαίνει μία πολύ μεγάλη μείωση στους υπολογισμούς όσο το N γίνεται μεγαλύτερο. Υπάρχουν αλγόριθμοι FFT για αρκετές διαφορετικές παραγοντοποιήσεις του N . [29]

E



Εικόνα 29. Απεικόνιση της λειτουργίας του FFT. [30]

Πιο απλά μιλώντας, ο FFT είναι ένας βελτιστοποιήμενος αλγόριθμος για την εκτέλεση του DFT. Ένα σήμα δειγματοληπτείται μέσα σε μία περίοδο του πεδίου του χρόνου και διαιρείται μέσα στις συνιστώσες της συχνότητας. Αυτές οι συνιστώσες είναι ημιτονοειδής ταλαντώσεις σε διακριτές συχνότητες με την κάθε μία με την δική της φάση και πλάτος.

Βήμα προς βήμα ο υπολογισμός γίνεται αρχικά με δύο παραμέτρους, τον ρυθμό δειγματοληψίας f_s και τον επιλεγμένο αριθμό δειγμάτων BL. Από αυτές τις δύο παραμέτρους, μπορούν να προσδιοριστούν και οι υπόλοιπες παράμετροι της μέτρησης:

- Το **Bandwidth fn**. Η τιμή του δείχνει την θεωρητικά μέγιστη συχνότητα που μπορεί να προσδιοριστεί μέσω του FFT.

$$fn = \frac{fs}{2}$$

Για παράδειγμα σε ρυθμό δειγματοληψίας 48 kHz, μπορούν να προσδιοριστούν θεωρητικά, συνιστώσες συχνότητας έως και 24 kHz.

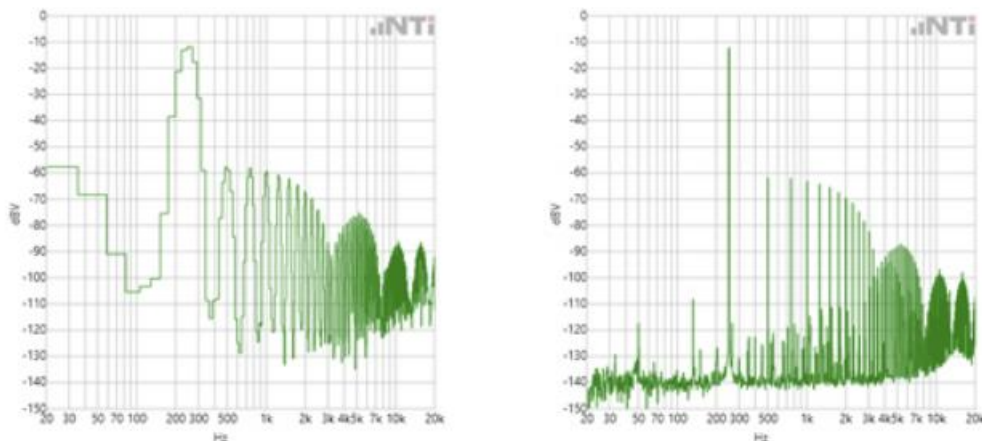
- Η **διάρκεια μέτρησης D**. Η διάρκεια μέτρησης υπολογίζεται με συνάρτηση δύο παραμέτρων, το fs και το BL και βρίσκεται ως:

$$D = \frac{BL}{fs}$$

- Η **ανάλυση συχνότητας df**. Η ανάλυση συχνότητας δείχνει την απόσταση συχνοτήτων μεταξύ δύο μετρήσεων.

$$df = \frac{fs}{BL}$$

Στην πράξη ένας μικρός αριθμός δειγμάτων BL έχει ως αποτέλεσμα γρήγορες επαναλήψεις μετρήσεων με «τραχειά» ανάλυση συχνότητας, ενώ με έναν μεγάλο αριθμό δειγμάτων το αποτέλεσμα είναι αργές επαναλήψεις μετρήσεων με «κομψή» ανάλυση συχνότητας, αυτά φαίνονται παρακάτω στην **Εικόνα 30**. [31]



Εικόνα 30. Διαφορά ανάλυσης συχνότητας με μικρό αριθμό δειγμάτων και μεγάλο. [31]

3.7.2 Ανάλυση FFT σε κώδικα Arduino και περαιτέρω βήματα

Παρακάτω φαίνεται ένας ενδεικτικός κώδικας για εκτέλεση FFT ανάλυσης σε κώδικα Arduino:

```

/*
//Example data:
int data[64]={ 14, 30, 35, 34, 34, 40, 46, 45, 30, 4, -26, -48, -55, -49, -37,
-28, -24, -22, -13, 6, 32, 55, 65, 57, 38, 17, 1, -6, -11, -19, -34,
-51, -61, -56, -35, -7, 18, 32, 35, 34, 35, 41, 46, 43, 26, -2, -31, -50,
-55, -47, -35, -27, -24, -21, -10, 11, 37, 58, 64, 55, 34, 13, -1, -7};
*/

//-----//
byte sine_data [91]=
{
0,
4, 9, 13, 18, 22, 27, 31, 35, 40, 44,
49, 53, 57, 62, 66, 70, 75, 79, 83, 87,
91, 96, 100, 104, 108, 112, 116, 120, 124, 127,
131, 135, 139, 143, 146, 150, 153, 157, 160, 164,
167, 171, 174, 177, 180, 183, 186, 189, 192, 195, //Paste this at top of program
198, 201, 204, 206, 209, 211, 214, 216, 219, 221,
223, 225, 227, 229, 231, 233, 235, 236, 238, 240,
241, 243, 244, 245, 246, 247, 248, 249, 250, 251,
252, 253, 253, 254, 254, 254, 255, 255, 255
};
float f_peaks[5]; // top 5 frequencies peaks in descending order
//-----//

void setup()
{

```

```
Serial.begin(250000);
}

void loop() {

/*
//example
FFT(data,64,100); //to get top five value of frequencies of X having 64 sample at 100Hz
sampling
Serial.println(f_peaks[0]);
Serial.println(f_peaks[1]);
delay(99999);
*/

/*
after ruing above FFT(), frequencies available at
f_peaks[0],f_peaks[1],f_peaks[2],f_peaks[3],f_peaks[4],
*/
}

//-----FFT Function-----//

float FFT(int in[],int N,float Frequency)
{
/*
Code to perform FFT on arduino,
setup:
paste sine_data [91] at top of program [global variable], paste FFT function at end of program
Term:
*/
}
```

Κεφάλαιο 3

1. in[] : Data array,

2. N : Number of sample (recommended sample size 2,4,8,16,32,64,128...)

3. Frequency: sampling frequency required as input (Hz)

If sample size is not in power of 2 it will be clipped to lower side of number.

i.e, for 150 number of samples, code will consider first 128 sample, remaining sample will be omitted.

For Arduino nano, FFT of more than 128 sample not possible due to mamory limitation (64 recomended)

For higher Number of sample may arise Mamory related issue,

Code by ABHILASH

Contact: abhilashpatel121@gmail.com

Documentation:https://www.instructables.com/member/abhilash_patel/instructables/

2/3/2021: change data type of N from float to int for >=256 samples

```
*/
```

```
unsigned int data[13]={1,2,4,8,16,32,64,128,256,512,1024,2048};
```

```
int a,c1,f,o,x;
```

```
a=N;
```

```
for(int i=0;i<12;i++) //calculating the levels
```

```
{ if(data[i]<=a){o=i;} }
```

```
int in_ps[data[o]]={}; //input for sequencing
```

```
float out_r[data[o]]={}; //real part of transform
```

```
float out_im[data[o]]={}; //imaginary part of transform
```

```
x=0;
```

```
for(int b=0;b<o;b++) // bit reversal
```

```
{
```

```
c1=data[b];
```

```
f=data[o]/(c1+c1);
```

```
for(int j=0;j<c1;j++)
```

```
{
```

```
x=x+1;
in_ps[x]=in_ps[j]+f;
}
}

for(int i=0;i<data[o];i++) // update input array as per bit reverse order
{
if(in_ps[i]<a)
{out_r[i]=in[in_ps[i]];}
if(in_ps[i]>a)
{out_r[i]=in[in_ps[i]-a];}
}

int i10,i11,n1;
float e,c,s,tr,ti;

for(int i=0;i<o;i++) //fft
{
i10=data[i]; // overall values of sine/cosine :
i11=data[o]/data[i+1]; // loop with similar sine cosine:
e=360/data[i+1];
e=0-e;
n1=0;

for(int j=0;j<i10;j++)
{
c=cosine(e*j);
s=sine(e*j);
n1=j;

for(int k=0;k<i11;k++)
```

Κεφάλαιο 3

```
{
    tr=c*out_r[i10+n1]-s*out_im[i10+n1];
    ti=s*out_r[i10+n1]+c*out_im[i10+n1];

    out_r[n1+i10]=out_r[n1]-tr;
    out_r[n1]=out_r[n1]+tr;

    out_im[n1+i10]=out_im[n1]-ti;
    out_im[n1]=out_im[n1]+ti;

    n1=n1+i10+i10;
}
}
}

/*
for(int i=0;i<data[o];i++)
{
Serial.print(out_r[i]);
Serial.print("\t"); // un comment to print RAW o/p
Serial.print(out_im[i]); Serial.println("i");
}
*/

//--> here onward out_r contains amplitude and our_in conntains frequency (Hz)
for(int i=0;i<data[o-1];i++) // getting amplitude from complex number
{
    out_r[i]=sqrt(out_r[i]*out_r[i]+out_im[i]*out_im[i]); // to increase the speed delete sqrt
    out_im[i]=i*Frequency/N;
}
/*
Serial.print(out_im[i]); Serial.print("Hz");
Serial.print("\t"); // un comment to print freuency bin
```

```
Serial.println(out_r[i]);
```

```
*/
```

```
}
```

```
x=0; // peak detection
```

```
for(int i=1;i<data[o-1]-1;i++)
```

```
{
```

```
if(out_r[i]>out_r[i-1] && out_r[i]>out_r[i+1])
```

```
{in_ps[x]=i; //in_ps array used for storage of peak number
```

```
x=x+1;}
```

```
}
```

```
s=0;
```

```
c=0;
```

```
for(int i=0;i<x;i++) // re arrange as per magnitude
```

```
{
```

```
for(int j=c;j<x;j++)
```

```
{
```

```
if(out_r[in_ps[i]]<out_r[in_ps[j]])
```

```
{s=in_ps[i];
```

```
in_ps[i]=in_ps[j];
```

```
in_ps[j]=s;}
```

```
}
```

```
c=c+1;
```

```
}
```

```
for(int i=0;i<5;i++) // updating f_peak array (global variable)with descending order
```

Κεφάλαιο 3

```
{  
    f_peaks[i]=out_im[in_ps[i];  
}
```

```
}
```

```
float sine(int i)  
{  
    int j=i;  
    float out;  
    while(j<0){j=j+360;}  
    while(j>360){j=j-360;}  
    if(j>-1 && j<91){out= sine_data[j];}  
    else if(j>90 && j<181){out= sine_data[180-j];}  
    else if(j>180 && j<271){out= -sine_data[j-180];}  
    else if(j>270 && j<361){out= -sine_data[360-j];}  
    return (out/255);  
}
```

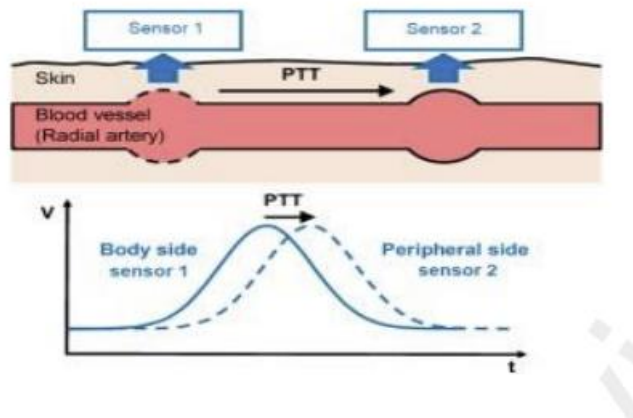
```
float cosine(int i)  
{  
    int j=i;  
    float out;  
    while(j<0){j=j+360;}  
    while(j>360){j=j-360;}  
    if(j>-1 && j<91){out= sine_data[90-j];}  
    else if(j>90 && j<181){out= -sine_data[j-90];}  
    else if(j>180 && j<271){out= -sine_data[270-j];}  
    else if(j>270 && j<361){out= sine_data[j-270];}  
    return (out/255);  
}
```

}

//-----//

[32]

Αφού αναλυθεί το σήμα με τον αλγόριθμο FFT, παίρνοντας την σημαντική πληροφορία, περνάμε στο επόμενο βήμα με τον υπολογισμό της περιόδου του παλμού. Έπειτα χρειάζεται να βρεθεί η μέγιστη κορυφή του παλμού ώστε να προχωρήσουμε στους χρόνους όπου εμφανίστηκαν οι κορυφές για να βρεθεί η διαφορά του χρόνου των κορυφών μεταξύ των εξόδων των δύο αισθητήρων, αυτό καταλήγει στον υπολογισμό του PTT (Pulse Transit Time), δηλαδή ο χρόνος που κάνει ένας παλμός ο οποίος ανιχνεύεται από τον ένα αισθητήρα, να φτάσει στον δεύτερο αισθητήρα.



Εικόνα 31. Εικονογραφία του υπολογισμού του PTT. [33]

3.8 Επίλογος

Σε αυτό το κεφάλαιο αναλύθηκαν τα μέρη της κατασκευής που θα χρειαστούν από το κομμάτι του hardware έως το κομμάτι του software ώστε να υλοποιηθεί το πιεσόμετρο. Είδαμε τα διαγράμματα ώστε να ακολουθηθεί βήμα βήμα η δημιουργία της κατασκευής και επίσης αναλύθηκε εις βάθος ο ρόλος της κάθε συνδεσμολογίας και ο κώδικας που θα χρειαστεί.

Κεφάλαιο 4ο : Ανάλυση προσομοίωσης και κατασκευής του κυκλώματος πιεσόμετρου

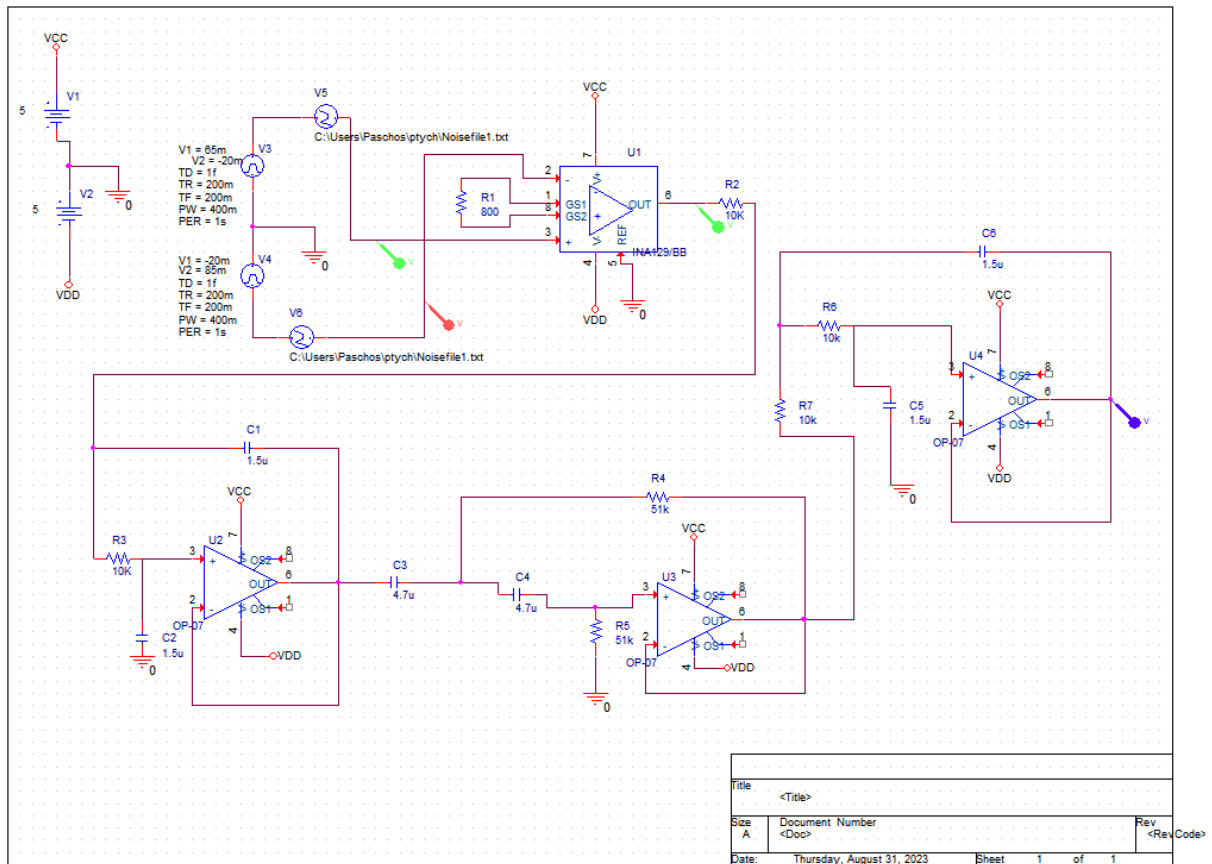
4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλυθούν οι υπολογισμοί και οι πράξεις που χρησιμοποιήθηκαν καθώς και η προσομοίωση του κυκλώματος και το κατασκευαστικό κομμάτι μαζί με τον κώδικά που χρησιμοποιήθηκε στην υλοποίηση αυτού.

4.2 Το πρόγραμμα προσομοίωσης κυκλωμάτων Orcad.

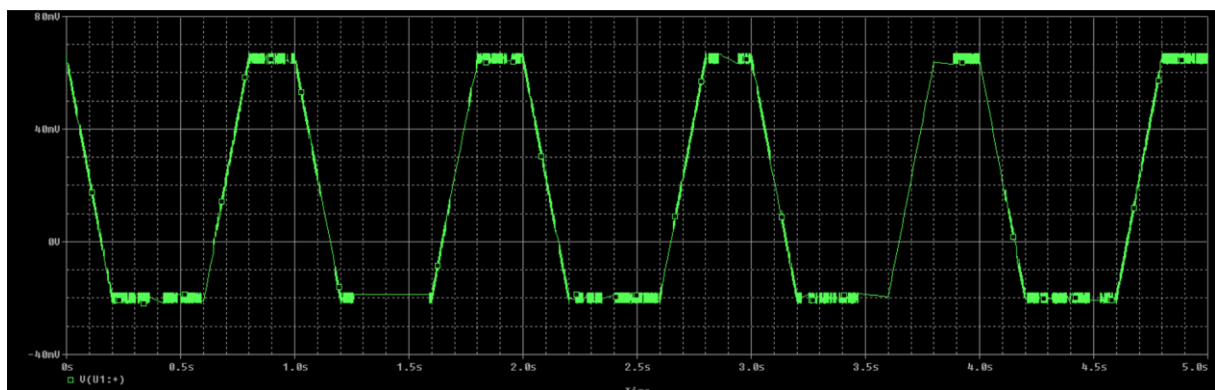
Για την προσομοίωση και το σχέδιο του κυκλώματος χρησιμοποιήθηκε το πρόγραμμα προσομοίωσης Orcad Capture CIS 16.6 ώστε να επαληθευθεί η σωστή λειτουργία του κυκλώματος ώστε έπειτα να γίνει η υλοποίησή του πάνω σε πλακέτα χωρίς προβλήματα.

Παρακάτω φαίνεται το κύκλωμα όπως δημιουργήθηκε στο Orcad:



Εικόνα 32. Το κύκλωμα πιεσόμετρου σχεδιασμένο στο OrCAD.

Αναλύοντας το κύκλωμα κομμάτι κομμάτι στην **Εικόνα 32** έχουν επιλεγθεί τιμές ώστε η προσομοίωση να πλησιάσει όσο γίνεται τις πραγματικές συνθήκες. Αρχικά οι πηγές **V4** και **V5** αναπαριστούν τον αισθητήρα AA002-02 βάζοντας μία πηγή παλμών **VPULSE** η οποία βρίσκεται στην βιβλιοθήκη **SOURCE** του OrCAD επιλέγοντας τιμές ώστε να πλησιάσουν τις τιμές του αισθητήρα, η κυματομορφή της οποίας φαίνεται παρακάτω:



Εικόνα 33. Η κυματομορφή προσομοίωσης του αισθητήρα AA002-02 από το OrCAD με την προσθήκη θορύβου.

Όπως φαίνεται στην **Εικόνα 33** υπάρχει καταγραφή δύο παλμών έχοντας προσθέσει και μία πηγή σε σειρά με τις πηγές παλμών αναπαράστασης του αισθητήρα έτσι ώστε να δημιουργηθεί θόρυβος πάνω

στους παλμούς ώστε να πλησιάσει τις πραγματικές συνθήκες, οι πηγές των παλμών είναι οι πηγές **V5** και **V6**, οι πηγές αυτές βρίσκονται επίσης στην βιβλιοθήκη **SOURCE** με την ονομασία **VPWL_FILE**, για την χρήση τους δημιουργήθηκε ένα αρχείο στο Microsoft Excel ώστε να δημιουργηθεί ένα αρχείο με δεδομένα και τιμές αναπαράστασης του θορύβου.

6	2,3049E-05	0,001538
7	3,79293E-05	0,000616
8	5,79877E-05	-0,00077
9	7,05079E-05	-0,00088
10	0,000115669	0,000352
11	0,000139602	-0,00019
12	0,000154673	-0,00056
13	0,000178415	0,000908
14	0,000213973	0,000258
15	0,000226982	-0,00116
16	0,000239732	0,000889
17	0,000257973	-0,00016
18	0,000375109	0,001685
19	0,00039994	-0,00097
20	0,00044934	0,000563
21	0,000460944	0,000979
22	0,000478712	-0,00099
23	0,000489868	0,001532
24	0,00050796	-0,0009
25	0,000518644	0,000854
26	0,000665501	0,001719
27	0,000691027	-0,00065
28	0,000701466	-0,00122
29	0,001198748	0,001987
30	0,001237904	-0,00077
31	0,001248465	-0,00137

Εικόνα 34. Τιμές θορύβου οι οποίες έχουν δημιουργηθεί στο Excel.

Βλέποντας την **Εικόνα 34** βλέπουμε δύο στήλες, η πρώτη στήλη αναπαριστά το σημείο του χρόνου πάνω στην κυματομορφή και η δεύτερη στήλη την τιμή που προστίθεται στο πλάτος της κυματομορφής, για τον υπολογισμό τους χρησιμοποιήθηκαν οι παρακάτω συναρτήσεις.

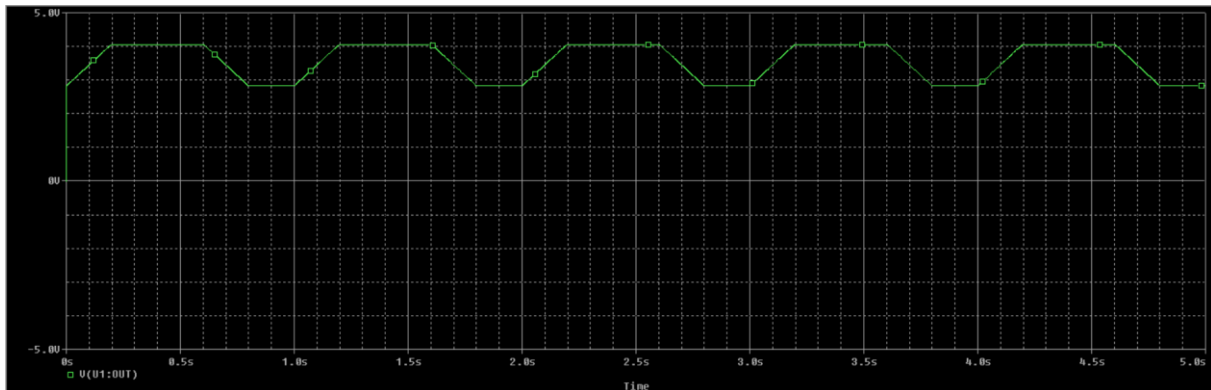
Για τις τιμές στον άξονα του χρόνου επιλέχθηκε η συνάρτηση $=1/(RAND()*100000)$ και για τις τιμές στον κάθετο άξονα επιλέχθηκε η συνάρτηση $=RAND()*0,004-0,002$.

Οι τιμές εξόδου του αισθητήρα που δημιουργήθηκαν παραπάνω καταλήγουν στην είσοδο του ενισχυτή οργανολογίας **INA129** καθώς είναι χαμηλής τάσης τιμές και χρειάζονται μεγάλη ενίσχυση. Η τιμή της ενίσχυσης του ενισχυτή οργανολογίας επιλέγεται από την αντίσταση **R1** και από τον τύπο:

$$G = 1 + \frac{49.4K\Omega}{R1}$$

[25]

Ο τύπος υπάρχει μέσα στο datasheet του ολοκληρωμένου INA129, στην περίπτωση αυτής της πτυχαικής επιλέγεται ενίσχυση 62.25 έτσι ώστε να βρίσκεται η κυματομορφή μέσα στα όρια των +5V και -5V η οποία είναι η τάση τροφοδοσίας του κυκλώματος ώστε να μην κόψει της κορυφές της κυματομορφής. Για να επιτευχθεί αυτή η ενίσχυση η τιμή της **R1** επιλέγεται στα 800Ω, με όλες τις τιμές της αντίστασης του ενισχυτή και των τιμών του αισθητήρα που επιλέχθηκαν να έχουν επιλεγθεί για χάριν της προσομοίωσης και ενδέχεται να διαφέρουν από το πραγματικό κύκλωμα καθώς η τιμή τάσης εξόδου του αισθητήρα καθορίζεται από την ένταση μαγνητικού πεδίου του μαγνήτη που τοποθετείται δίπλα στον αισθητήρα και έτσι χρειάζεται και διαφορετική τιμή ενίσχυσης. Ο ενισχυτής πέρα από την έξοδο του αισθητήρα ενισχύει και τον θόρυβο, έτσι η έξοδος του καταλήγει σε φίλτρο Sallen Key. Η έξοδος του ενισχυτή φαίνεται στην **Εικόνα 35** παρακάτω.



Εικόνα 35. Κυματομορφή εξόδου του ενισχυτή INA129.

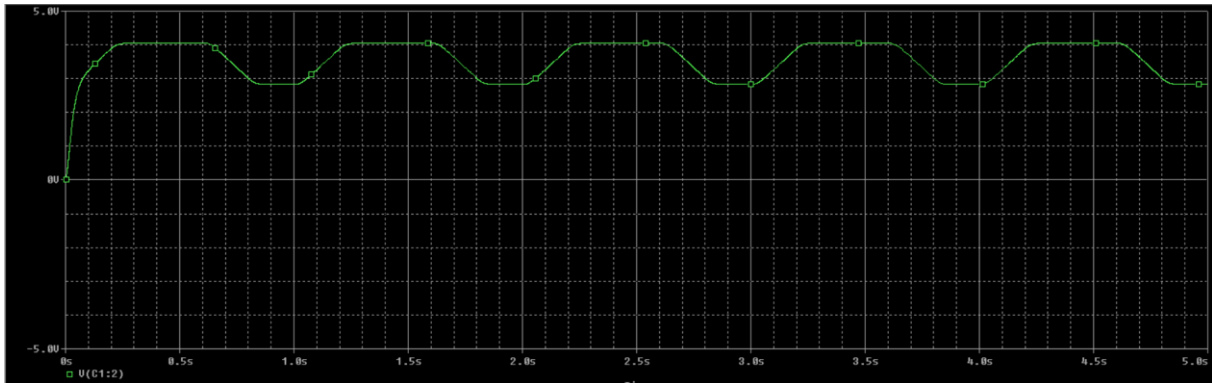
Η έξοδος του ενισχυτή καταλήγει σε ένα χαμηλοπερατό φίλτρο Sallen-Key ώστε να εξαλείψει τον θόρυβο, το φίλτρο αυτό αποτελείται από τις αντιστάσεις **R2** και **R3** και τους πυκνωτές **C1** και **C2**. Για τον υπολογισμό των τιμών των αντιστάσεων και των πυκνωτών για συχνότητα αποκοπής 10Hz χρησιμοποιήθηκε ο παρακάτω τύπος:

$$f_c = \frac{1}{2\pi\sqrt{R_A R_B C_A C_B}}$$

Ο οποίος τύπος μπορεί να απλοποιηθεί περαιτέρω για $R_A=R_B$ και $C_A=C_B$ σε:

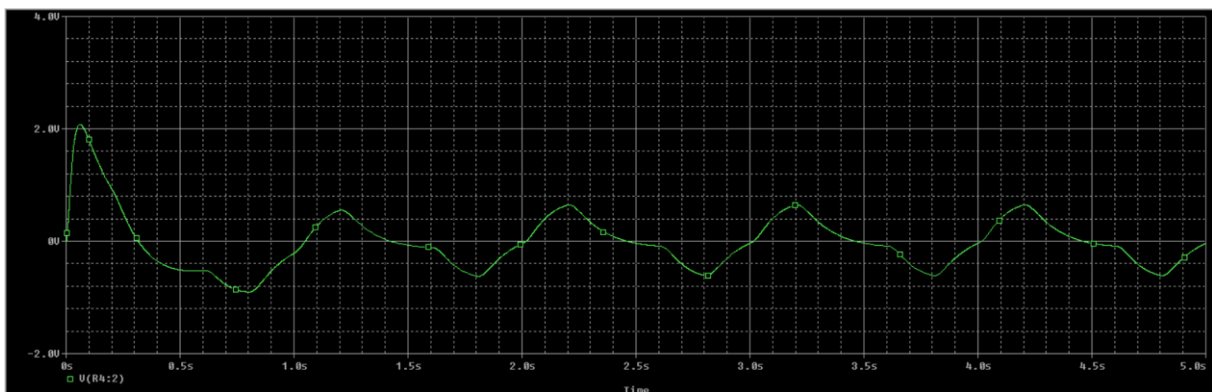
$$f_c = \frac{1}{2\pi RC}$$

Από τον τύπο προκύπτει για συχνότητα αποκοπής 10Hz, τιμή αντίστασης $R_2, R_3=10K$ και τιμή πυκνωτή $C_1, C_2=1.5\mu F$. Έτσι έχοντας δημιουργήσει το φίλτρο η έξοδος που θα πάρουμε από αυτό θα είναι η εξής:



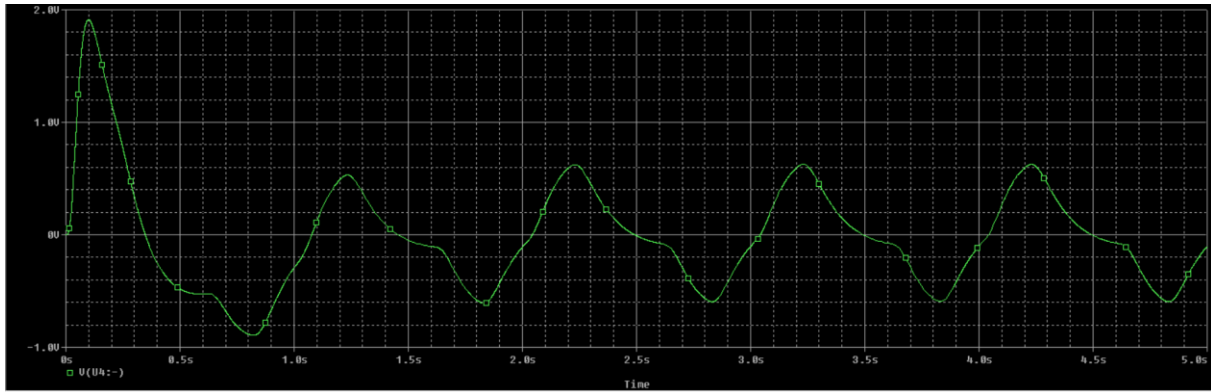
Εικόνα 36. Κυματομορφή εξόδου πρώτου χαμηλοπερατού φίλτρου Sallen-Key.

Αφού έχει περάσει το σήμα από το φίλτρο και έχοντας εξαλείψει τον θόρυβο, η έξοδος αυτού του φίλτρου περνάει από ένα υψιπερατό φίλτρο Sallen-Key ώστε να εξαλείψει το σήμα που έχει ληφθεί από τον αναπνευστικό ρυθμό έτσι ώστε να κρατηθεί μόνο το σήμα που λαμβάνεται από την αρτηρία, με συχνότητα αποκοπής 0.7Hz. Οι τιμές των αντιστάσεων και των πυκνωτών υπολογίζονται με τον ίδιο τύπο με το χαμηλοπερατό και έτσι για 0.7Hz οι τιμές που υπολογίστηκαν είναι, οι τιμές των αντιστάσεων 51kΩ και οι τιμές των πυκνωτών 4.7μF. Η έξοδος του υψιπερατού φίλτρου φαίνεται παρακάτω στην **Εικόνα 37**.



Εικόνα 37. Κυματομορφή εξόδου υψιπερατού φίλτρου Sallen-Key.

Και τέλος αφού περάσει και το υψιπερατό φίλτρο καταλήγει πάλι σε ένα χαμηλοπερατό φίλτρο για εξάλειψη οποιουδήποτε άλλου θορύβου με τιμές αντιστάσεων ίδιες με το πρώτο χαμηλοπερατό φίλτρο για συχνότητα αποκοπής 10Hz και η τελική μορφή του σήματος φαίνεται παρακάτω στην **Εικόνα 38**.



Εικόνα 38. Κυματομορφή εξόδου του κυκλώματος που προορίζεται για επεξεργασία στο Arduino.

Όπως φαίνεται από την κυματομορφή εξόδου, το σήμα εξόδου μοιάζει αρκετά με σήμα καρδιογραφήματος και έτσι καταλήγουμε από την προσομοίωση στο συμπέρασμα ότι το κύκλωμα φαίνεται σωστό και μπορούμε να προχωρήσουμε στην κατασκευή του κυκλώματος.

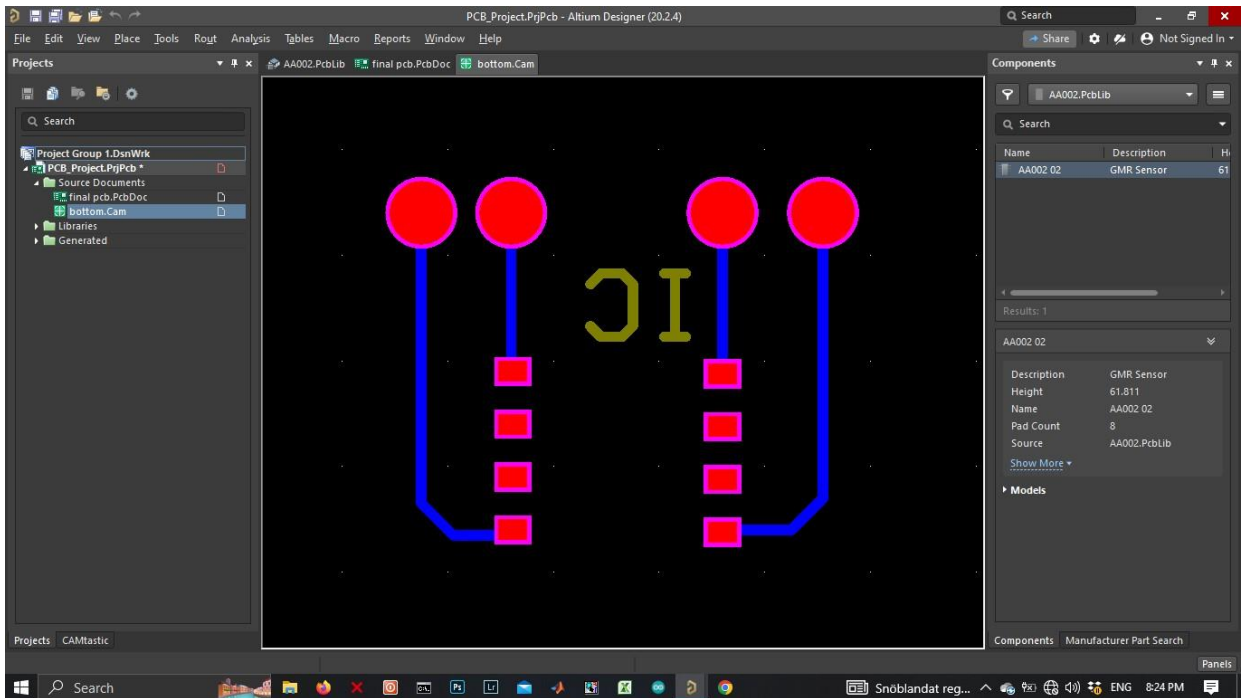
4.3 Κατασκευή του κυκλώματος

Έχοντας κάνει την προσομοίωση του κυκλώματος και βλέποντας ότι η έξοδος είναι η αναμενόμενη, μπορεί να υλοποιηθεί η κατασκευή με τα υλικά που υπολογίστηκαν και επιλέχθηκαν στην προσομοίωση.

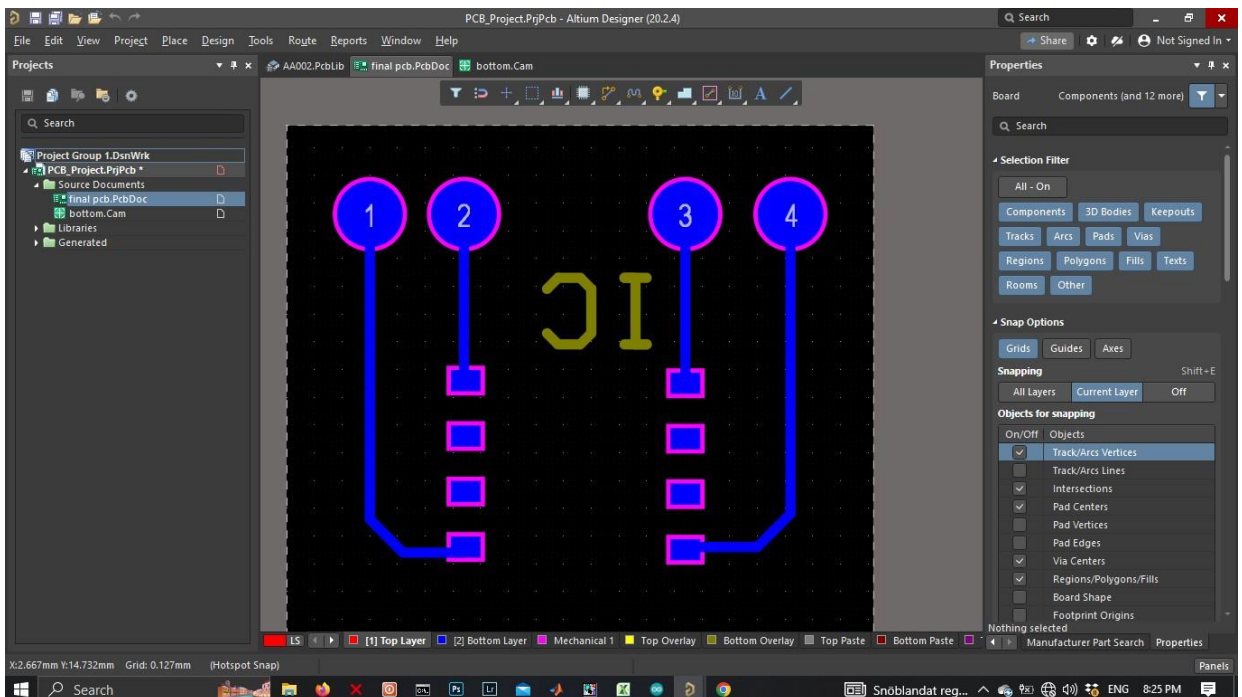
4.3.1 Σχεδίαση πλακέτας αισθητήρα AA002-02

Ο αισθητήρας GMR που επιλέχθηκε για να συνδεθεί με το υπόλοιπο κύκλωμα χρειάστηκε να κατασκευαστεί μία πλακέτα υποστήριξης για τον λόγο ότι ο αισθητήρας είναι smd και το υπόλοιπο κύκλωμα υλοποιήθηκε σε ραστερ. Το πρόγραμμα που χρησιμοποιήθηκε για την κατασκευή της πλακέτας είναι το Altium Designer και το σχέδιο φαίνεται παρακάτω.

Ανάλυση προσομοίωσης και κατασκευής του κυκλώματος πιεσομέτρου

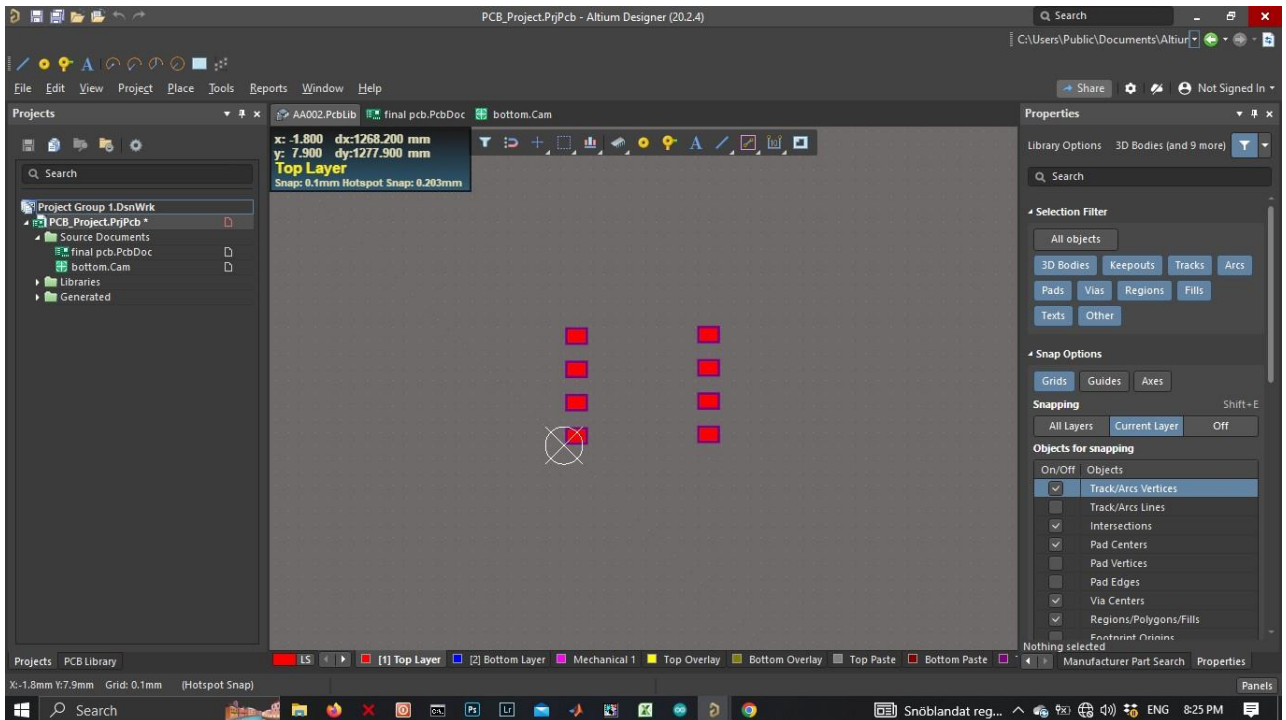


Εικόνα 39. Σχεδίο της πλακέτας σε формат αρχείου .Cam υλοποιημένο σε Altium Designer.



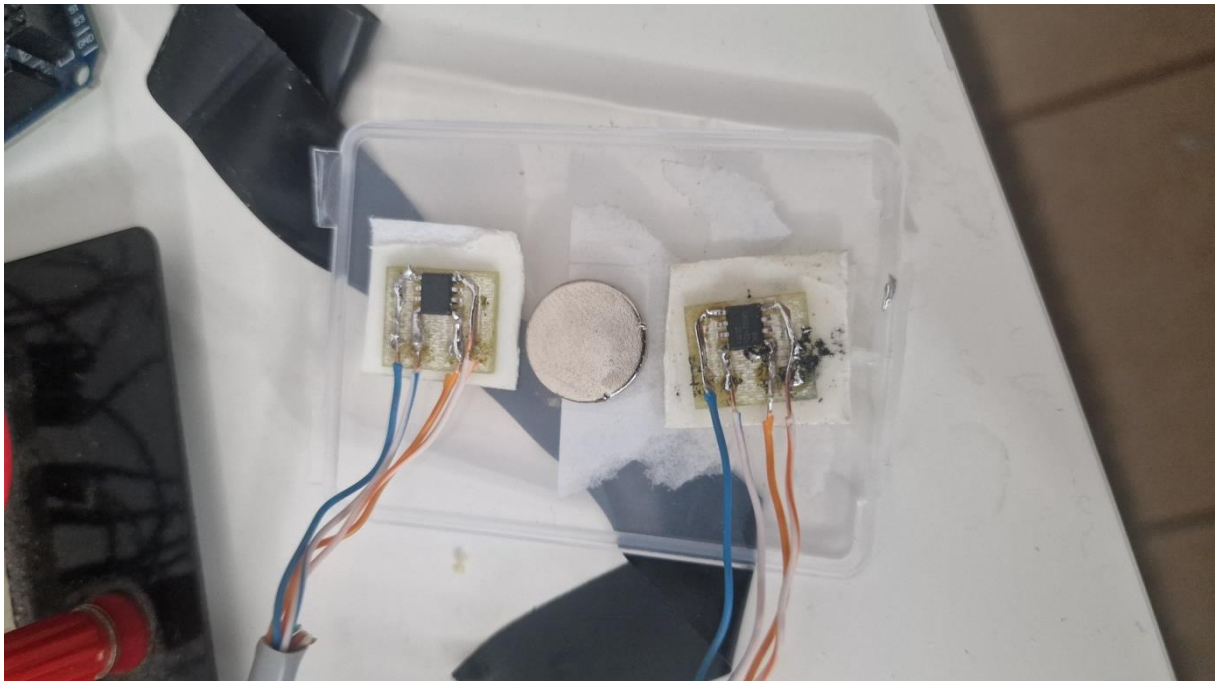
Εικόνα 40. Σχεδίο της πλακέτας σε формат αρχείου .pcbdoc υλοποιημένο σε Altium Designer.

Κεφάλαιο 4

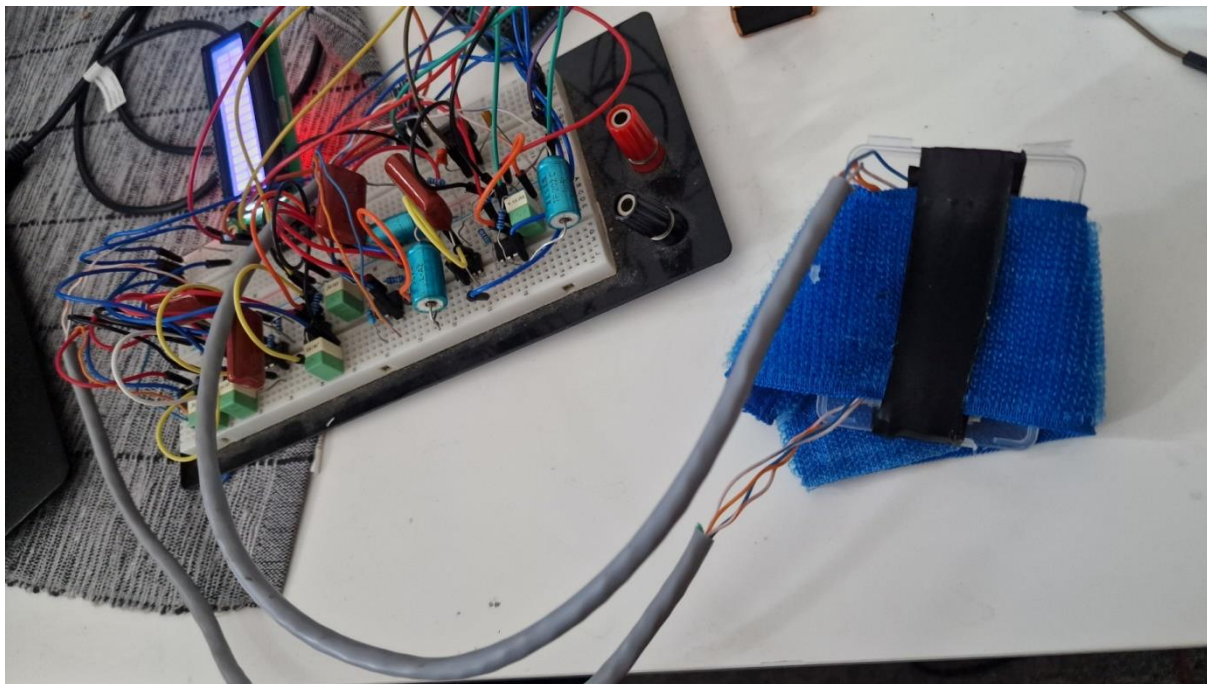


Εικόνα 41. Σχέδιο της πλακέτας σε μορφή αρχείου .pcblib υλοποιημένο σε Altium Designer.

Έχοντας τελειώσει με το σχέδιο της πλακέτας για την υποστήριξη του αισθητήρα AA002-02 το επόμενο βήμα ήταν η υλοποίηση του σχεδίου ώστε να χρησιμοποιηθεί η πλακέτα πάνω στο κύκλωμα.



Εικόνα 42. Οι αισθητήρες AA002-02 πάνω στο κουφάρι με τον μαγνήτη έτοιμο προς τοποθέτηση στο χέρι.



Εικόνα 43. Στην εικόνα φαίνεται η σύνδεση των αισθητήρων στην πλακέτα, έχει χρησιμοποιηθεί πλαστική επιφάνεια στην οποία έχουν κολληθεί πάνω με ταινία διπλής όψης οι αισθητήρες και ανάμεσα ο μαγνήτης κάθετα με τους αισθητήρες. Για την τοποθέτηση στον καρπό έχει χρησιμοποιηθεί ταινία Velcro(χριτσ-χρατσ), και τέλος για την σύνδεση με το κύκλωμα χρησιμοποιήθηκε καλώδιο Ethernet ώστε να μπορεί να χρησιμοποιηθεί από απόσταση.

Έτσι, έχοντας κατασκευάσει την πλακέτα για τον αισθητήρα AA002-02 και τοποθετώντας τον πάνω της, και έχοντας όλα τα υλικά, το επόμενο βήμα είναι η κατασκευή του κυκλώματος.

4.3.2 Κατασκευή του κυκλώματος πάνω σε ραστερ

Το πρώτο βήμα για την κατασκευή του κυκλώματος είναι η ανάλυση του ώστε να βρεθούν τα υλικά που απαιτούνται για την κατασκευή, τα οποία φαίνονται στον παρακάτω πίνακα **Πίνακας 3**:

No.	Είδος	Όνομα	Ποσότητα	Τιμή ενός	Τιμή συνό	Links	Currency	Euro	Σύνολο
1	GMR αισθ	AA002-02	2	21,053	42,106	https://il.f	\$	37,47434	117,8298
2	Μόνιμοι μ	D063D-N3	50	0,399	19,95	https://wv	\$	17,7555	
3	Ενισχυτής	INA129	2	8,56	17,12	https://gr	€	17,12	
4	Τελεστικοί ενισχυτές		6	1,48	8,88	https://gr	€	8,88	
5	Arduino Mega 2560		1	29,61	29,61	https://wv	€	29,61	
6	LCD 16x2		1	3,39	3,39	https://wv	€	3,39	
7	8x12	Διατηρητή π	2	1,8	3,6	https://wv	€	3,6	

Πίνακας 3: Τα βασικά υλικά που χρησιμοποιήθηκαν για την κατασκευή του κυκλώματος.

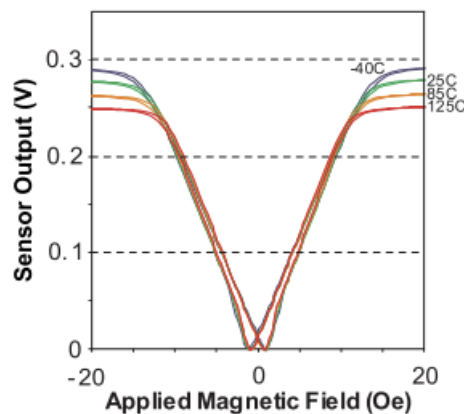
Πέραν των υλικών του **Πίνακα 3** χρησιμοποιήθηκαν και επιπλέον αναλώσιμα(πχ.πυκνωτές, αντιστάσεις, καλώδια-αγωγοί).

Επίσης για την τροφοδοσία του κυκλώματος χρησιμοποιήθηκαν 2 μπαταρίες 9V σε σειρά έτσι ώστε να χρησιμοποιηθούν τάσεις -9V και 9V οι οποίες θα περάσουν από δύο Voltage Regulators, τον LM7805 για θετικές τάσεις 5V και τον LM7905 για αρνητικές τάσεις -5V, οι οποίοι θα τροφοδοτήσουν το κύκλωμα.

Οι αισθητήρες AA002-02 συνδέθηκαν στον ενισχυτή οργανολογίας, με τα pins να έχουν συνδεθεί ως εξής:

- AA002-02 pin 1 Vout- => INA129 pin 2 Vin-
- AA002-02 pin 4 V-/GND => -5V
- AA002-02 pin 5 Vout+ => INA129 pin 3 Vin+
- AA002-02 pin 8 V+ => +5V
- AA002-02 pins 2,3,6,7 => N.C.

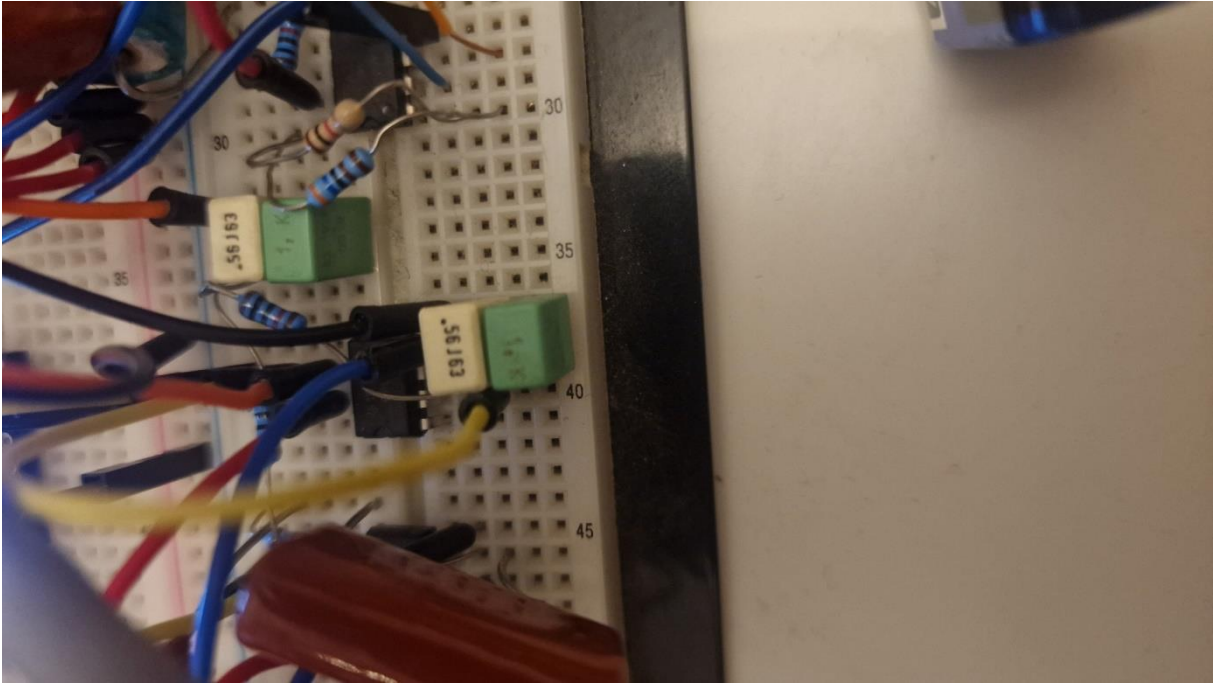
Επίσης ανάμεσα από τους δύο αισθητήρες χρησιμοποιήθηκε ένας μαγνήτης νεοδμίμου N35 15MM x 3MM κάτι που σημαίνει ότι έχει δύναμη έλξης 1kg, σε απόσταση 2 εκ. από τον κάθε αισθητήρα, σε πλαϊνή θέση καθώς οι αισθητήρες AA002-02 δεν λειτουργούν σαν αισθητήρες Hall και χρειάζεται να είναι τοποθετημένοι ανάμεσα στους δύο πόλους του μαγνήτη ώστε να μεταβάλεται η αντίσταση του αισθητήρα με τα μαγνητικά πεδία. Έτσι επιτυγχάνεται η λειτουργία των αισθητήρων στην γραμμική τους περιοχή όπως φαίνεται στην **Εικόνα 44**.



Εικόνα 44. Έξοδος του αισθητήρα AA002-02 σε σχέση με το μαγνητικό πεδίο για ρεύμα 1mA.[22]

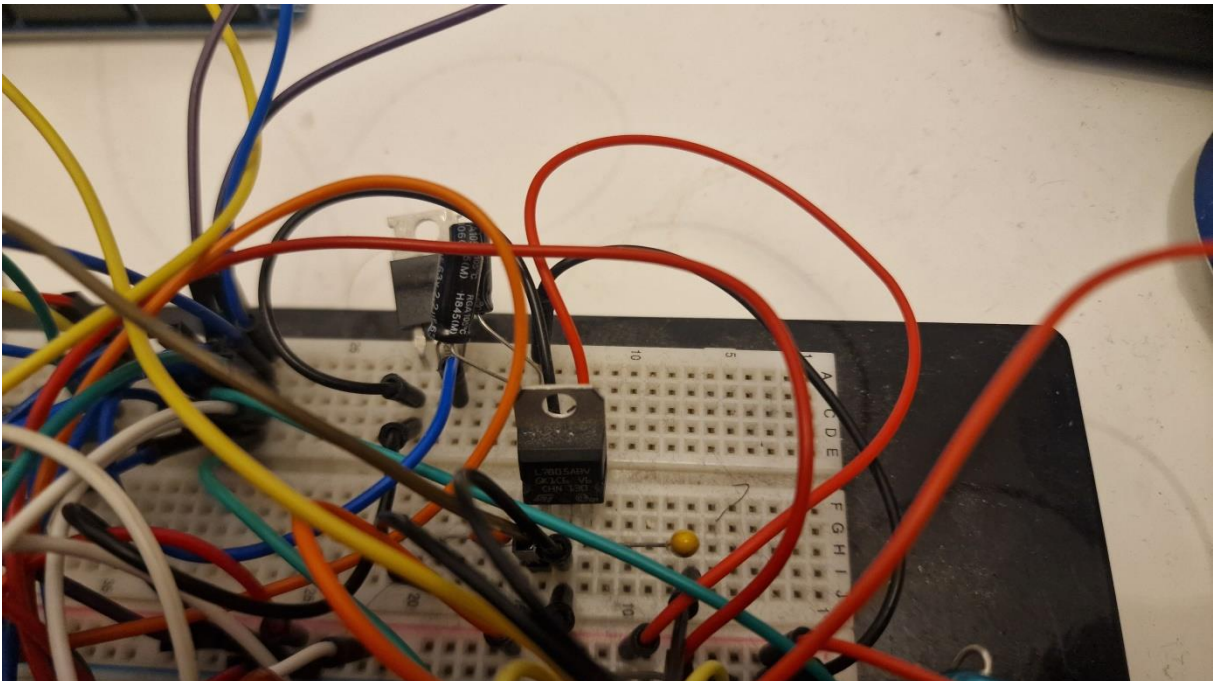
Εφόσον ο αισθητήρας θα τοποθετηθεί πάνω στο χέρι για να μεγάλώσει η αγωγιμότητα του χεριού για να βοηθήσουμε τον αισθητήρα να πάρει τον παλμό, τοποθετήθηκαν κάτω από το μονωτικό υλικό (πλακέτα και πλαστικό κουφάρι), δύο αγωγικά υλικά, στην περίπτωσή μας δύο κέρματα, ένα κάτω από κάθε αισθητήρα.

Τα υλικά που χρησιμοποιήθηκαν για την κατασκευή του κυκλώματος ήταν τα ανάλογα με την προσομοίωση. Στην είσοδο του ενισχυτή οργανολογίας ως RG αντίσταση που ρυθμίζει την ενίσχυση του κυκλώματος χρησιμοποιήθηκαν δύο αντιστάσεις παράλληλα R1 3.6k και R2 1k, οπότε υπολογίζοντας $R1R2/(R1+R2)=782\Omega$. Δοκιμάστηκαν διάφορες αντιστάσεις από 67Ω έως και 3K και η μοναδική κυματομορφή σε λογικά επίπεδα ήταν χρησιμοποιώντας την αντίσταση 782Ω. Έπειτα οι πυκνωτές που χρησιμοποιήθηκαν ήταν διαφορετικών τύπων λόγω έλλειψης στο κατάστημα κατά την αγορά, όλοι οι πυκνωτές ήταν χωρίς πολικότητα και χρησιμοποιήθηκαν είτε δύο πυκνωτές παράλληλα όπως φαίνεται στην **Εικόνα 45**.



Εικόνα 45. Δύο πυκνωτές παράλληλα συνδεδεμένοι ο ένας $1\mu\text{F}$ και ο άλλος $0.56\mu\text{F}$.

Οι πυκνωτές που χρησιμοποιήθηκαν παράλληλα με τιμές $1\mu\text{F}$ και $0.56\mu\text{F}$ χρησιμοποιήθηκαν έτσι ώστε να επιτευχθεί η τιμή $1.56\mu\text{F}$ η οποία είναι κοντά στην τιμή $1.5\mu\text{F}$ της προσομοίωσης. Όλοι οι πυκνωτές που τοποθετήθηκαν, υπολογίστηκαν να έχουν τάση αντοχής πάνω από $2 \times 5\text{V} = 10\text{V}$ καθώς η ονομαστική τάση του κυκλώματος είναι τα 5V .

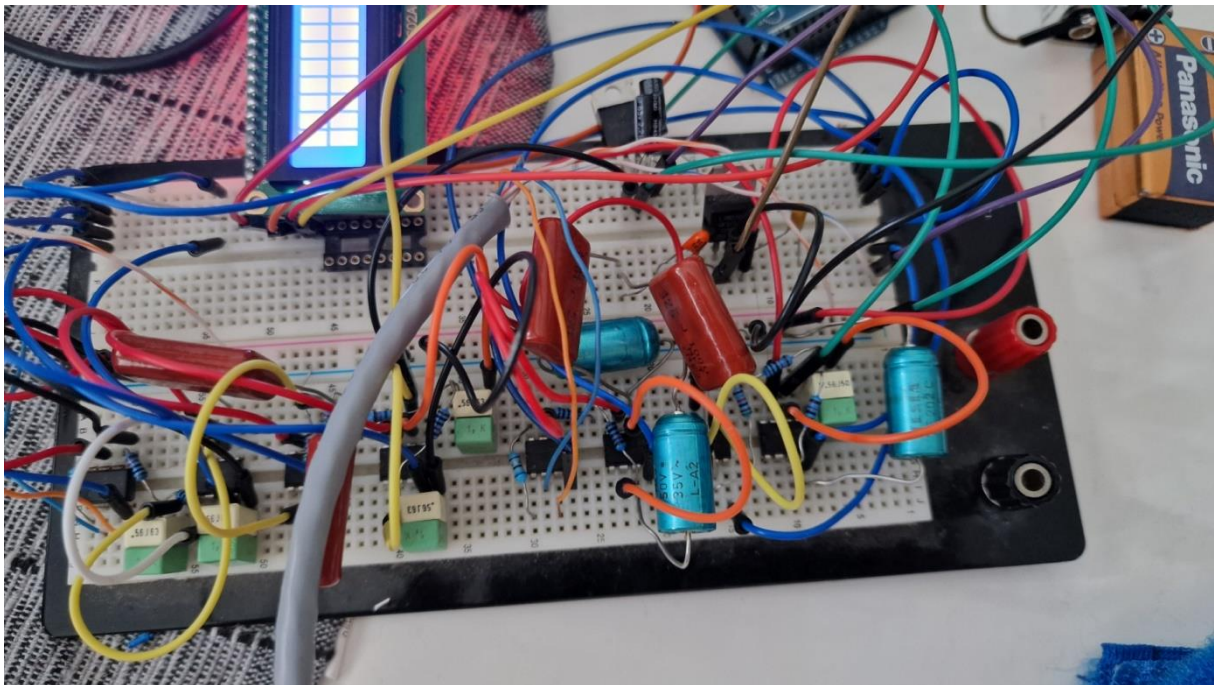


Εικόνα 46. Οι δύο ρυθμιζόμενες τάσεις, LM7805 και LM7905.

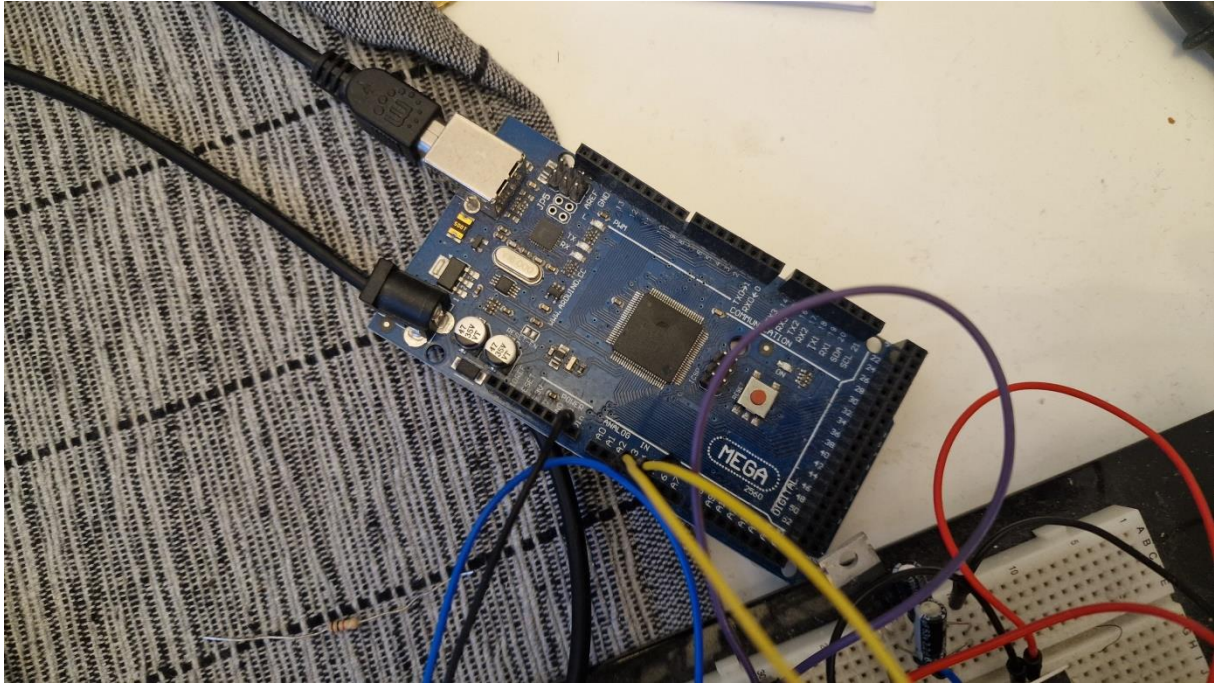


Εικόνα 47. Μπαταρίες τροφοδοσίας του κυκλώματος.

Ο ρόλος του κυκλώματος στο ράστερ είναι να κάνει conditioning το σήμα της εισόδου που λαμβάνεται από τους αισθητήρες ώστε να έρθει στα επίπεδα επεξεργασίας του από την πλακέτα Arduino Mega.

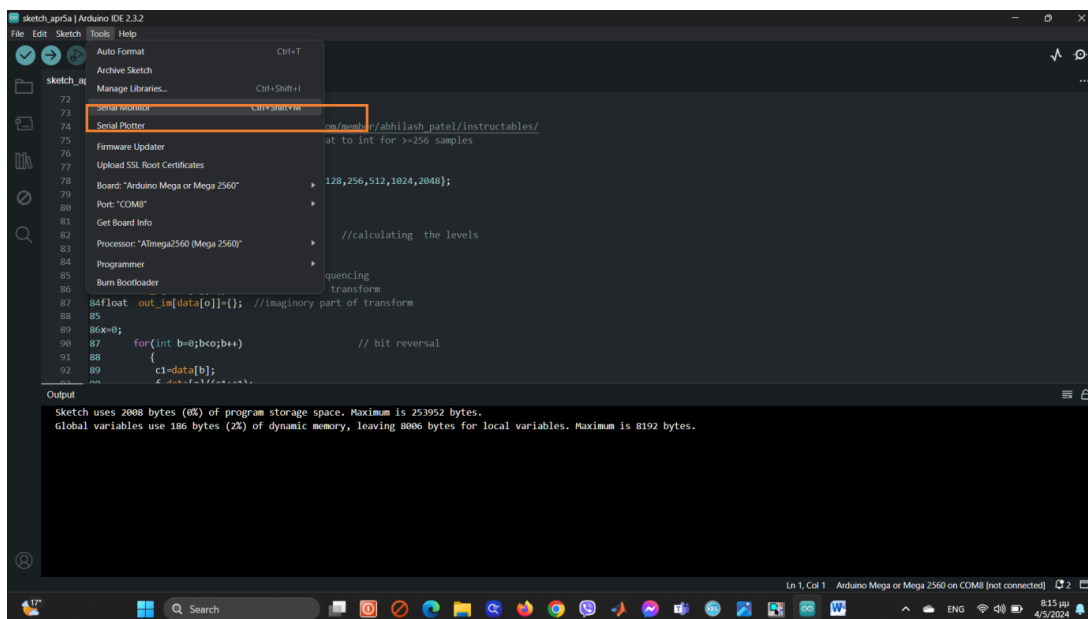


Εικόνα 48. Το κύκλωμα πιεσόμετρου υλοποιημένο σε ράστερ.



Εικόνα 49. Η πλακέτα Arduino Mega που χρησιμοποιήθηκε στο κύκλωμα.

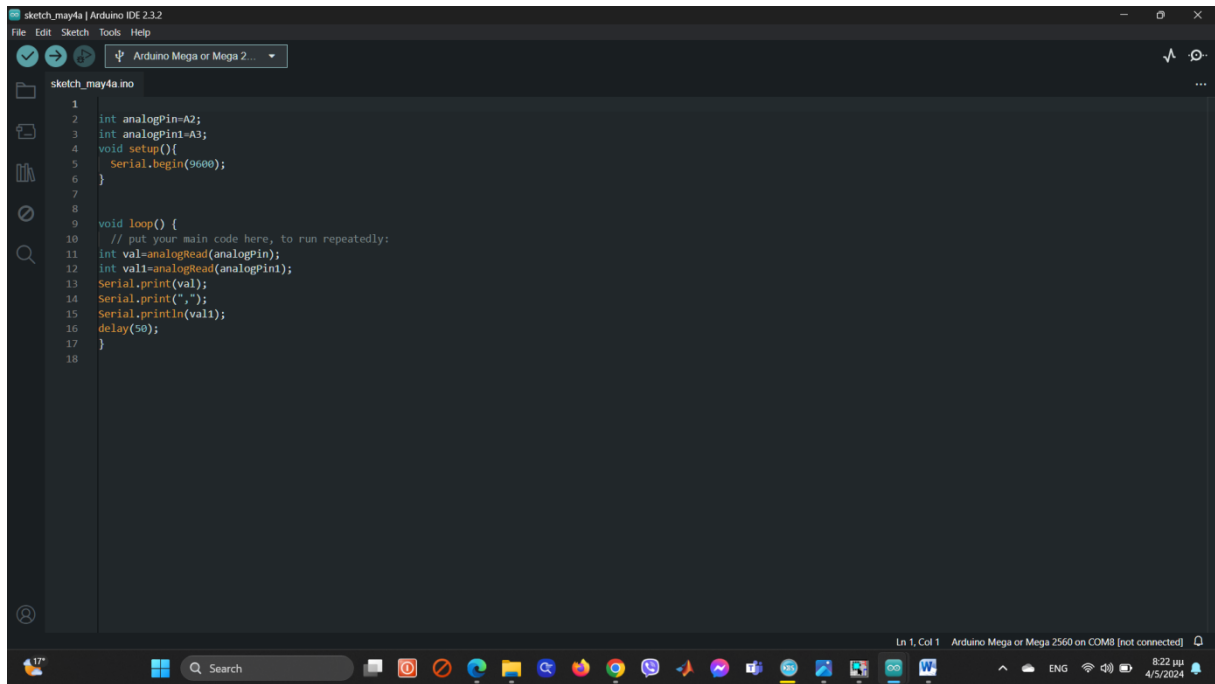
Έχοντας υλοποιήσει το κύκλωμα πάνω στην πλακέτα το επόμενο βήμα είναι να δούμε αν λειτουργεί σωστά είτε χρησιμοποιώντας κάποιον παλμογράφο είτε με την βοήθεια της επιλογής Serial Plotter στο Arduino IDE.



Εικόνα 50. Η επιλογή Serial plotter στο Arduino IDE.

Κεφάλαιο 4

Για τον έλεγχο συμπεριφοράς των εξόδων του κυκλώματος χρησιμοποιήθηκε ο κώδικας που φαίνεται στην **Εικόνα 51** παρακάτω.



```
1
2 int analogPin=A2;
3 int analogPin=A3;
4 void setup(){
5   Serial.begin(9600);
6 }
7
8
9 void loop() {
10  // put your main code here, to run repeatedly:
11  int val=analogRead(analogPin);
12  int val1=analogRead(analogPin);
13  Serial.print(val);
14  Serial.print(",");
15  Serial.println(val1);
16  delay(50);
17 }
18
```

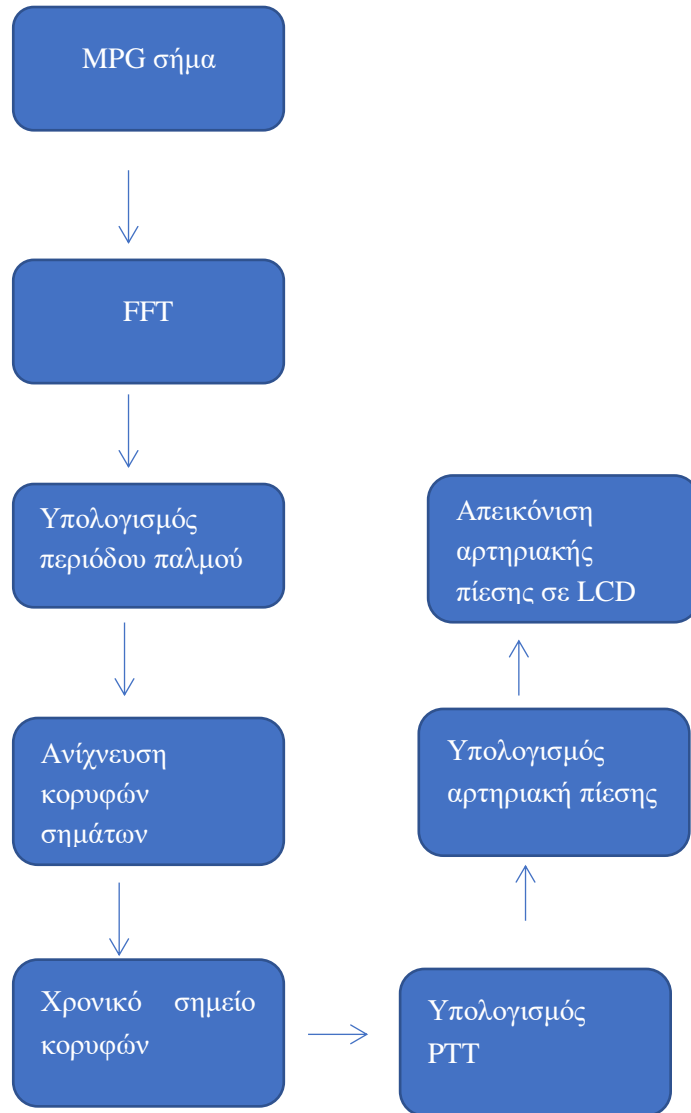
Εικόνα 51. Ο κώδικας ελέγχου εξόδου του κυκλώματος.

Έχοντας τσεκάρει την έξοδο του κυκλώματος τα επόμενα βήματα είναι να περάσουν τα σήματα από τον αλγόριθμο FFT, να υπολογιστεί η περίοδος του παλμού, να βρεθούν οι κορυφές των σημάτων και σε ποιο χρονικό σημείο έγιναν, έπειτα υπολογίζεται η χρονική διαφορά των κορυφών των δύο σημάτων, και τέλος με αυτά τα δεδομένα μπορεί να υπολογιστεί η μέση αρτηριακή πίεση με την χρήση του παρακάτω τύπου:

$$BP = P + \frac{Q}{PTT}$$

[21]

Στον τύπο φαίνεται ότι για να υπολογιστεί η αρτηριακή πίεση χρειάζεται να βρεθεί το PTT και να υπολογιστούν οι P και Q σταθερές μέσω της γραμμικής παλινδρόμησης από δεδομένα που έχουν παρθεί.



Σχήμα 3. Μπλοκ διάγραμμα υλοποίησης κυκλώματος και επιπέδου επεξεργασίας και απεικόνισης πιεσομέτρου.

Τέλος, για τη απεικόνιση του αποτελέσματος της αρτηριακής πίεσης χρησιμοποιήθηκε μία οθόνη LCD 16x2 χαρακτήρων, συνδεδεμένη με ένα module I2C το οποίο συνδέθηκε στην πλακέτα Arduino στα pins SDA, SCL, 5V και GND. Ο λόγος χρήσης του I2C είναι επειδή είναι πιο εύκολη η σύνδεση.



Εικόνα 52. Η οθόνη LCD.

4.3.3 Επεξεργασία των δεδομένων της κατασκευής μέσω Arduino Mega

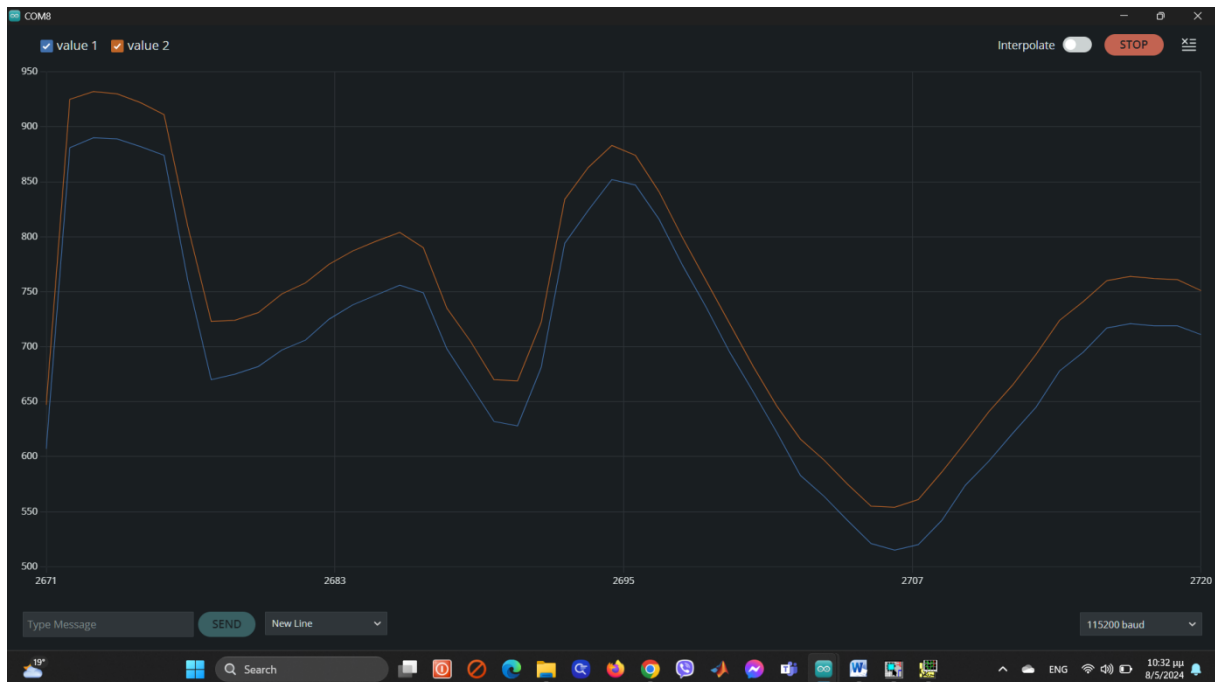
Το επόμενο βήμα μετά την κατασκευή του κυκλώματος είναι η εξαγωγή των δεδομένων από την έξοδο των δύο κυκλωμάτων ώστε να περάσουν από τα στάδια που φαίνονται στο **Σχήμα 3** και να υπολογιστεί η μέση αρτηριακή πίεση. Τα παραπάνω στάδια πρέπει να υλοποιηθούν με κώδικα στην πλακέτα Arduino αφού πρώτα έχουμε καταλήξει σε ποιά pins της πλακέτας θα συνδεθούν οι έξοδοι. Το πρώτο βήμα είναι να συνδεθεί η έξοδος του ενός από τα δύο κυκλώματα σε ένα Analog pin της πλακέτας ώστε να περάσει από έναν κώδικα FFT για να υπολογιστεί ο καρδιακός παλμός του χρήστη αφού πρώτα ελεγχθεί η έξοδος των κυκλωμάτων. Για τον έλεγχο των εξόδων του κυκλώματος καθώς δεν υπήρχε παλμογράφος, χρησιμοποιήθηκε η επιλογή Serial Plotter του Arduino IDE με τον παρακάτω κώδικα:

```
int analogPin=A2;
int analogPin1=A3;
void setup() {
  Serial.begin(115200);
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
  int val=analogRead(analogPin);
```

```
Serial.print(val);
int val1=analogRead(analogPin1);
Serial.print(",");
Serial.println(val1);
delay(100);
}
```

Ο παραπάνω κώδικας διαβάζει τις εξόδους των δύο κυκλωμάτων, άρα των pins A2 και A3 κάθε 100ms δίνοντας την κυματομορφή που φαίνεται στην **Εικόνα 53**.



Εικόνα 53. Κυματομορφή εξόδου των δύο κυκλωμάτων.

Και για την υλοποίηση του FFT για τον υπολογισμό του καρδιακού παλμού χρησιμοποιήθηκε το ένα pin μεταξύ A2 και A3 με τον παρακάτω κώδικα:

```
int data_in[128];
unsigned long t;

void loop() {

t=micros();
for(int i=0;i<128;i++)
{
data_in[i]=analogRead(A2);
delayMicroseconds(10);
}

t=micros()-t;
```

```

t=128000000/t;
FFT(data_in,128,t);
Serial.println(f_peaks[0]);
delay(1000);
}

```

Ο παραπάνω κώδικας διαβάζει τα δεδομένα από το pin A2 της πλακέτα και εφαρμόζει πάνω του FFT 128 σημείων. Τα δεδομένα παρουσιάζονται ως πίνακας 128 δειγμάτων μέσω του πίνακα ακέραιων αριθμών `int data_in` και εξάγεται η μέγιστη συχνότητα μέσω του `f_peaks[0]`. Για να χρησιμοποιηθεί αυτός ο κώδικας, πρέπει πρώτα να γραφεί ο κώδικας του FFT ο οποίος φαίνεται παρακάτω:

```

byte sine_data [91]= {
0,
4, 9, 13, 18, 22, 27, 31, 35, 40, 44,
49, 53, 57, 62, 66, 70, 75, 79, 83, 87,
91, 96, 100, 104, 108, 112, 116, 120, 124, 127,
131, 135, 139, 143, 146, 150, 153, 157, 160, 164,
167, 171, 174, 177, 180, 183, 186, 189, 192, 195,
198, 201, 204, 206, 209, 211, 214, 216, 219, 221,
223, 225, 227, 229, 231, 233, 235, 236, 238, 240,
241, 243, 244, 245, 246, 247, 248, 249, 250, 251,
252, 253, 253, 254, 254, 254, 255, 255, 255, 255
};
float f_peaks[5];
void setup(){
Serial.begin(9600);
}

```

[32]

Αρχικά ο παραπάνω κώδικας μπαίνει στην αρχή του κώδικα ο οποίος δείχνει ένα ημιτονικό σήμα 0 έως 90 μοιρών σε παράγωγα του 255 ώστε να ψηφιοποιηθεί και να μπορεί να αποθηκευτεί ως bytes. Αυτό μειώνει την μνήμη του Arduino στο $\frac{1}{4}$ σε σχέση με το αν αποθηκευόντουσαν τα δεδομένα ως float. Ο πίνακας `float f_peaks[5]` δηλώνεται έτσι ώστε να τραβάει από το FFT τις 5 μέγιστες συχνότητες με το `f_peaks[0]` να είναι η μέγιστη. Έπειτα ο κώδικας που χρησιμοποιείται για τον αλγόριθμο FFT είναι ο παρακάτω:

```

float FFT(int in[],int N,float Frequency)
{
/*
Code to perform FFT on arduino,
setup:
paste sine_data [91] at top of program [global variable], paste FFT function
at end of program
Term:
1. in[] : Data array,
2. N : Number of sample (recommended sample size
2,4,8,16,32,64,128...)

```

```

3. Frequency: sampling frequency required as input (Hz)

If sample size is not in power of 2 it will be clipped to lower side of
number.
i.e, for 150 number of samples, code will consider first 128 sample,
remaining sample will be omitted.
For Arduino nano, FFT of more than 128 sample not possible due to mamory
limitation (64 recomended)
For higher Number of sample may arise Mamory related issue,
Code by ABHILASH
Contact: abhilashpatel121@gmail.com
Documentation:https://www.instructables.com/member/abhilash_patel/instructable
s/
2/3/2021: change data type of N from float to int for >=256 samples
*/

unsigned int data[13]={1,2,4,8,16,32,64,128,256,512,1024,2048};
int a,c1,f,o,x;
a=N;

    for(int i=0;i<12;i++)                //calculating the levels
        { if(data[i]<=a){o=i;} }

int in_ps[data[o]]={}; //input for sequencing
float out_r[data[o]]={}; //real part of transform
float out_im[data[o]]={}; //imaginary part of transform

x=0;
    for(int b=0;b<o;b++)                // bit reversal
    {
        c1=data[b];
        f=data[o]/(c1+c1);
        for(int j=0;j<c1;j++)
        {
            x=x+1;
            in_ps[x]=in_ps[j]+f;
        }
    }

    for(int i=0;i<data[o];i++)        // update input array as per bit
reverse order
    {
        if(in_ps[i]<a)
        {out_r[i]=in[in_ps[i]];}
        if(in_ps[i]>a)
        {out_r[i]=in[in_ps[i]-a]}
    }

```

```

int i10,i11,n1;
float e,c,s,tr,ti;

for(int i=0;i<o;i++) //fft
{
  i10=data[i]; // overall values of sine/cosine :
  i11=data[o]/data[i+1]; // loop with similar sine cosine:
  e=360/data[i+1];
  e=0-e;
  n1=0;

  for(int j=0;j<i10;j++)
  {
    c=cosine(e*j);
    s=sine(e*j);
    n1=j;

    for(int k=0;k<i11;k++)
    {
      tr=c*out_r[i10+n1]-s*out_im[i10+n1];
      ti=s*out_r[i10+n1]+c*out_im[i10+n1];

      out_r[n1+i10]=out_r[n1]-tr;
      out_r[n1]=out_r[n1]+tr;

      out_im[n1+i10]=out_im[n1]-ti;
      out_im[n1]=out_im[n1]+ti;

      n1=n1+i10+i10;
    }
  }
}

/*
for(int i=0;i<data[o];i++)
{
  Serial.print(out_r[i]);
  Serial.print("\ "); // un comment to print
  RAW o/p
  Serial.print(out_im[i]); Serial.println("i");
}
*/

//--> here onward out_r contains amplitude and our_in conntains frequency
(Hz)

```

```

    for(int i=0;i<data[o-1];i++) // getting amplitude from
complex number
    {
        out_r[i]=sqrt(out_r[i]*out_r[i]+out_im[i]*out_im[i]); //
to increase the speed delete sqrt
        out_im[i]=i*Frequency/N;
        /*
        Serial.print(out_im[i]); Serial.print("Hz");
        Serial.print("\ "); // un comment to
print freuency bin
        Serial.println(out_r[i]);
        */
    }

x=0; // peak detection
for(int i=1;i<data[o-1]-1;i++)
{
    if(out_r[i]>out_r[i-1] && out_r[i]>out_r[i+1])
    {in_ps[x]=i; //in_ps array used for storage of peak number
    x=x+1;}
}

s=0;
c=0;
for(int i=0;i<x;i++) // re arrange as per magnitude
{
    for(int j=c;j<x;j++)
    {
        if(out_r[in_ps[i]]<out_r[in_ps[j]])
        {s=in_ps[i];
        in_ps[i]=in_ps[j];
        in_ps[j]=s;}
    }
    c=c+1;
}

for(int i=0;i<5;i++) // updating f_peak array (global variable)with
descending order
{
    f_peaks[i]=out_im[in_ps[i]];
}

```

```

}

float sine(int i)
{
    int j=i;
    float out;
    while(j<0){j=j+360;}
    while(j>360){j=j-360;}
    if(j>-1 && j<91){out= sine_data[j];}
    else if(j>90 && j<181){out= sine_data[180-j];}
    else if(j>180 && j<271){out= -sine_data[j-180];}
    else if(j>270 && j<361){out= -sine_data[360-j];}
    return (out/255);
}

float cosine(int i)
{
    int j=i;
    float out;
    while(j<0){j=j+360;}
    while(j>360){j=j-360;}
    if(j>-1 && j<91){out= sine_data[90-j];}
    else if(j>90 && j<181){out= -sine_data[j-90];}
    else if(j>180 && j<271){out= -sine_data[270-j];}
    else if(j>270 && j<361){out= sine_data[j-270];}
    return (out/255);
}

//-----
-----//

```

[32]

Εφόσον ο αλγόριθμος FFT εξάγει την συχνότητα του σήματος, και ο λόγος που χρησιμοποιείται είναι να βρεθεί η διάρκεια παλμού, χρησιμοποιείται η παρακάτω εντολή για μετατροπή της συχνότητας στο πεδίο του χρόνου δείχνοντας του παλμούς ανα δευτερόλεπτο:

```
float HB=1.0/f_peaks[0];
```

Επόμενο βήμα αφού έχει βρεθεί ο καρδιακός παλμός είναι να βρεθούν οι άνω κορυφές των δύο σημάτων καθώς και ο χρόνος στον οποίο παρατηρήθηκαν οι κορυφές. Για την εύρεση των κορυφών χρησιμοποιήθηκε ο παρακάτω κώδικας:

```
for(int i = 0 ; i<5;i++)
```

```

{
  cur_time = millis();
  while(cur_time - time_elapsed <=time_period)
  {
    if(mx==0)
    { mx = analogRead(A2);
      mx1=analogRead(A3);}
    else
    { data = analogRead(A2);
      data1=analogRead(A3);
      mx = max(data , mx);
      mx1=max(data1,mx1);
    }
    cur_time = millis();
  }
  time_elapsed = cur_time;
  peaks[i] = mx;
  Serial.print("Sensor 1 peak ");
  Serial.println(peaks[i]);
  peaks[i]=mx1;
  Serial.print("Sensor 2 peak ");
  Serial.println(peaks[i]);
  mx = 0;
  mx1=0;
}

```

Έχοντας δηλώσει ως global variables τις παρακάτω:

```

  unsigned long int cur_time = 0 ;
  unsigned long int time_elapsed = 0;
  int peaks[5]; // say you want to find 20 peaks , 1 peak for data arriving
between 5 secs
  int time_period = 10000;
  int data = 0;
  int mx=0;
  int data1 = 0;
  int mx1=0;

```

Έπειτα χρειάστηκε να βρεθούν τα χρονικά σημεία των κορυφών από τα δύο κυκλώματα έτσι ώστε να συγκριθούν και να βρεθεί η χρονική διαφορά τους. Για να γίνει αυτό χρησιμοποιήθηκε ο παρακάτω κώδικας. Αρχικά δηλώθηκαν ως global variables τα:

```

const int sensorPin1=A2;
const int sensorPin2= A3;
int Max1 = 0;
int Max2=0;

```

```

int sensorValue1;
int sensorValue2;

int threshold = 900;

unsigned long time1=0;
unsigned long time2=0;
unsigned long ptt=0;
unsigned long realtime1=0;
unsigned long realtime2=0;

```

Στον παραπάνω κώδικα φαίνεται ότι για να βρεθούν τα peaks των σημάτων χρησιμοποιήκε η μεταβλητή int threshold η οποία δηλώθηκε με αυτήν την τιμή μετά παρατήρηση της συμπεριφοράς του σήματος, αυτή η τιμή στην ουσία λαμβάνει το peak την στιγμή που η κορυφή του σήματος ξεπερνάει την τιμή 900 η οποία σε volt υπολογίζεται $5 \cdot (900/1024) = 4.4V$.

Και ο κώδικας που χρησιμοποιήθηκε μέσα στο void loop() ήταν ο εξής:

```

sensorValue1 = analogRead(sensorPin1);
sensorValue2= analogRead(sensorPin2);
//for detection of peak for sensor1
if (sensorValue1 > Max1)
{
    Max1 = sensorValue1;
    time1=millis();
}
if (sensorValue1 <= threshold && Max1 > threshold)
{
    Serial.print(Max1);
    Serial.print("\t");
    Serial.print(time1);
    Serial.print("\t");
    realtime1=time1;
    Max1=0;
}
//for detection of peak for sensor2
if (sensorValue2 > Max2)
{
    Max2= sensorValue2;
    time2=millis();
}
if (sensorValue2 <= threshold && Max2 > threshold)
{
    Serial.print(Max2);
    Serial.print("\t");
    Serial.print(time2);
    Serial.print("\t");
    realtime2= time2;
    Max2=0;
}

```

```
//time difference between peaks from both sensors  
pdt=abs(realtime1-realtime2);  
Serial.println(pdt);  
pdt=0;
```

Το επόμενο βήμα είναι να υπολογιστούν οι σταθερές P και Q του τύπου [35] ώστε να μπουν στον αλγόριθμο και να υπολογίσουν την μέση αρτηριακή πίεση και να εμφανιστεί στην οθόνη LCD. Για αυτή την διεργασία γράφτηκε ο παρακάτω κώδικας:

```
int BP=P+(Q/pdt);  
  
lcd.print("BP=");  
//lcd.print(BP);  
lcd.print("120");  
delay(100000);  
lcd.clear();
```

Για τον οποίο χρειάστηκε να χρησιμοποιηθεί η βιβλιοθήκη Arduino:

```
#include <LiquidCrystal_I2C.h>
```

Η βιβλιοθήκη χρησιμοποιήθηκε για την αναγνώριση των εντολών μέσω ενός ψευδοκώδικα που υπάρχει μέσα στην βιβλιοθήκη. Αλλά πριν από αυτά έπρεπε να αναγνωρισθεί η διεύθυνση I2C της LCD και να μπει στην παρακάτω εντολή:

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2 rows
```

Κεφάλαιο 5ο : Συμπεράσματα και προτάσεις βελτίωσης

Μετά το πέρας της κατασκευής είδαμε ότι έχει μεγάλη διαφορά η προσωμοίωση από την πραγματική κατασκευή καθώς στην προσωμοίωση όλα υλοποιούνται υπό ιδανικές συνθήκες και με εύκολότερη διάγνωση των λαθών που μπορεί να υπάρξουν. Από την άλλη μεριά όταν πρόκειται για την κατασκευή του κυκλώματος ιδιαίτερα όταν γίνεται σε ράστερ πρέπει να υπάρχει η διαβεβαίωση ότι όλα τα υλικά πατάνε καλά πάνω στο κύκλωμα, να βρεθούν τα υλικά που υπολογίστηκαν με μαθηματικά, τα οποία στον πραγματικό κόσμο μπορεί να διαφέρουν ελάχιστα οι τιμές τους, κάτι που μπορεί να σημαίνει μικρή παρακκλιση σε σχέση με την προσωμοίωση. Στο κύκλωμα επίσης θα πρέπει να ελέγχονται τα υλικά μήπως κάποιο απ' αυτά έχει καεί για τον οποιοδήποτε λόγο (πχ μετατροπή του κυκλώματος) κάτι που μπορεί να σημαίνει λάθος λειτουργία του κυκλώματος καθώς και λάθος μετρήσεις. Στη συνέχεια κάτι που είναι ιδιαίτερα μεγάλη πρόκληση είναι ο κώδικας για την επεξεργασία των εξόδων των κυκλωμάτων καθώς ενώ μπορεί ο κώδικας κατά την αποσφαλμάτωση να είναι σωστός αλλά η λειτουργία του να είναι λάθος, αυτό μπορεί να λυθεί βλέποντας την συμπεριφορά του κυκλώματος και των μετρήσεων που λαμβάνονται κάθε φορά που τρέχει ο κώδικας. Για κάθε κατασκευή αναλογικών κυκλωμάτων είναι απαιτούμενο πέρα από το πολύμετρο(το οποίο είναι απαιτούμενο για κάθε κύκλωμα) και ένας παλμογράφος, για την επιβεβαίωση των εξόδων των βαθμίδων και των κυκλωμάτων.

Προτάσεις και βελτιώσεις που θα μπορούσαν να γίνουν σε αυτό το κύκλωμα είναι οι παρακάτω:

- Βελτιστοποίηση του κώδικα για καλύτερη διαχείριση μνήμης και επίτευξη στόχων.
- Σχεδίαση πλακέτας PCB για το κύκλωμα καθώς και χρήση SMD υλικών για σμίκρυνση του κυκλώματος.
- Βελτιστοποίηση κώδικα ώστε να μετράει μικρή και μεγάλη πίεση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Ο τρόπος γραφής των βιβλιογραφικών αναφορών γίνεται σύμφωνα με τα παρακάτω παραδείγματα (IEEE style):

[1] Sang-Min Kim.Kang-Hwi Lee.Seong-Su Lee.Hyeok-Jae Lee.Byoung-Hun Lee.Kyeoung-Seop Kim.Jeong-Whan Lee, “A Study on the Detection Characteristics of the Magneto-Plethysmography According to Fluid Properties,” Trans. KIEE. Vol. 67, No. 7, JUL, 2018.

[2] President F F Cartwright FFARC S, *Section of the History of Medicine*, Volume 70 November 1977.

[3] Florian Rader, MD, MSC, Ronald G. Victor, MD, “The Slow Evolution of Blood Pressure Monitoring But Wait, Not So Fast!*” JACC: Basic to Translational Science Volume 2, Issue 6, December 2017, Pages 643-645.

[4] <https://www.vita4you.gr/blog-vita4you/el/item/piesometro-pos-leitoyrgei-amp-poio-na-epilexo.html>

[5] <https://www.onmed.gr/ygeia/story/319079/oi-idanikes-times-tis-piesis-ana-ilikia-pinakas>

[6] ΙΩΑΝΝΑ ΔΗΜΟΠΟΥΛΟΥ ΝΟΣΗΛΕΥΤΡΙΑ Τ.Ε. ΜΕΘ ΒΓΝΗ, ΑΙΜΟΔΥΝΑΜΙΚΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΒΑΡΕΩΣ ΠΙΑΣΧΟΝΤΩΝ ΣΤΗ ΜΕΘ.

[7] Jaakko Malmivuo, Robert Plonsey Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetis Fields Chapter: 25, January 1995.

[8]

<https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%BC%CE%B1%CE%B3%CE%BD%CE%B7%CF%84%CE%B9%CF%83%CE%BC%CF%8C%CF%82>

[9] <https://prirucnik.hr/el/srednji-arterijski-tlak-normalan-nizak-visoki-ocitani-plus-tretman/>

[10] Ιωάννης Γκέκας, Αρτηριακή πίεση, Θέματα Αναισθησιολογίας και Εντατικής Ιατρικής, σελ. 113-122.

[11] https://users.sch.gr/jabatzo/files/yliko/live%20ebooks/robotiki_G_2018_final/_arduino.html

[12] Motaz Daadoo, Saed Tarapiah, Saed Tarapiah, Shadi AtallaShadi Atalla, «*Analysis and Performance of a Low Cost Multiple Alarm Security System for Smart Home Based on GSM Technology and Controlling Based on Android Smartphone*», European Journal of Scientific Research, December 2016.

[13] [https://www.c-sharpcorner.com/UploadFile/7d4524/introduction-of-arduinomega-2560/#:~:text=The%20Arduino%20Mega%202560%20is,Pulse%20Width%20Modulation%20\(PWM\).](https://www.c-sharpcorner.com/UploadFile/7d4524/introduction-of-arduinomega-2560/#:~:text=The%20Arduino%20Mega%202560%20is,Pulse%20Width%20Modulation%20(PWM).)

[14] <https://www.arrow.com/en/research-and-events/articles/oscillators-and-arduino-configurations-and-settings>

[15] <https://scholar.ppu.edu/bitstream/handle/123456789/638/1.pdf?sequence=3&isAllowed=y>

- [16] <https://support.arduino.cc/hc/en-us/articles/5779192727068-Reset-your-board>
- [17] <https://www.circuito.io/blog/arduino-uno-pinout/>
- [18] https://www.etechnophiles.com/arduino-mega-pinout-pin-diagram-schematic-and-specifications-in-detail/?utm_content=cmp-true
- [19] <https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide/>
- [20] <https://emeritus.org/blog/coding-arduino-programming-language/>
- [21] Vinit Kumar Chugh, Kubera Kalyan, Anoop C. S., Amit Patra, Shubham Negi, Department of Electrical Engineering, IIT Kharagpur, India Department of Avionics, IIST, Trivandrum, ” *Analysis of a GMR-Based Plethysmograph transducer and its Utility for Real-time Blood Pressure Measurement*”.
- [22] NVE Corporation, “AA/AB-Series Analog Magnetic Sensors,” AA002-02 datasheet, 2019.
- [23] <https://www.electronicshub.org/instrumentation-amplifier-basics-applications/>
- [24] https://en.wikipedia.org/wiki/Instrumentation_amplifier
- [25] Texas Instruments, “INA12x Precision, Low-Power Instrumentation Amplifiers”, INA129, OCTOBER 1995 – REVISED MAY 2022
- [26] <https://circuitcellar.com/resources/quickbits/sallen-n-key-filter/>
- [27] <https://www.electronics-tutorials.ws/filter/sallen-key-filter.html>
- [28] Texas Instruments, “OP07x Precision Operational Amplifiers”, OP07C, OCTOBER 1983–REVISED NOVEMBER 2014
- [29] <https://allsignalprocessing.com/lessons/fast-fourier-transform-fft-algorithm/>
- [30] <https://www.dataq.com/data-acquisition/general-education-tutorials/fft-fast-fourier-transform-waveform-analysis.html>
- [31] [https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft#:~:text=The%20%22Fast%20Fourier%20Transform%22%20\(%20frequency%20information%20about%20the%20signal.](https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft#:~:text=The%20%22Fast%20Fourier%20Transform%22%20(%20frequency%20information%20about%20the%20signal.)
- [32] <https://projecthub.arduino.cc/abhilashpatel121/easyfft-fast-fourier-transform-fft-for-arduino-03724d>

ΠΑΡΑΡΤΗΜΑ Α : Ο ΠΛΗΡΗΣ ΚΩΔΙΚΑΣ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2 rows

byte sine_data [91]= {
0,
4, 9, 13, 18, 22, 27, 31, 35, 40, 44,
49, 53, 57, 62, 66, 70, 75, 79, 83, 87,
91, 96, 100, 104, 108, 112, 116, 120, 124, 127,
131, 135, 139, 143, 146, 150, 153, 157, 160, 164,
167, 171, 174, 177, 180, 183, 186, 189, 192, 195,
198, 201, 204, 206, 209, 211, 214, 216, 219, 221,
223, 225, 227, 229, 231, 233, 235, 236, 238, 240,
241, 243, 244, 245, 246, 247, 248, 249, 250, 251,
252, 253, 253, 254, 254, 254, 255, 255, 255, 255
};

unsigned long int cur_time = 0;
unsigned long int time_elapsed = 0;
int peaks[5]; // say you want to find 20 peaks , 1 peak for data arriving between 5 secs
int time_period = 10000;
int data = 0;
int mx=0;
int data1 = 0;
int mx1=0;
const int sensorPin1=A2;
const int sensorPin2= A3;
int Max1 = 0;
int Max2=0;

int sensorValue1;
int sensorValue2;

int threshold = 635;

unsigned long time1=0;
unsigned long time2=0;
unsigned long ptt=0;
unsigned long realtime1=0;
unsigned long realtime2=0;

float f_peaks[5];
void setup(){
Serial.begin(9600);
lcd.init();
lcd.clear();
lcd.backlight(); // Make sure backlight is on

// Print a message on both lines of the LCD.
lcd.setCursor(2,0); //Set cursor to character 2 on line 0
```

```

}
//data extraction
int data_in[128];
unsigned long t;

void loop() {

t=micros();
for(int n=0;n<128;n++)
{
data_in[n]=analogRead(A2);
delayMicroseconds(10);
}

t=micros()-t;
t=128000000/t;
FFT(data_in,128,t);
Serial.println(f_peaks[0]);
Serial.println(f_peaks[1]);
float HB=1.0/f_peaks[0];
delay(100);
//peak detection
for(int i = 0 ; i<5;i++)
{
cur_time = millis();
while(cur_time - time_elapsed <=time_period)
{
if(mx==0)
{ mx = analogRead(A2);
mx1=analogRead(A3);}
else
{ data = analogRead(A2);
data1=analogRead(A3);
mx = max(data , mx);
mx1=max(data1,mx1);

}

cur_time = millis();
}
time_elapsed = cur_time;
peaks[i] = mx;
Serial.print("Sensor 1 peak ");
Serial.println(peaks[i]);
peaks[i]=mx1;
Serial.print("Sensor 2 peak ");
Serial.println(peaks[i]);
mx = 0;
mx1=0;
//peak time stamp
sensorValue1 = analogRead(sensorPin1);
sensorValue2= analogRead(sensorPin2);
//for detection of peak for sensor1
if (sensorValue1 > Max1)
{

```

```

    Max1 = sensorValue1;
    time1=millis();
  }
  if (sensorValue1 <= threshold && Max1> threshold)
  {
    Serial.print(Max1);
    Serial.print("\t");
    Serial.print(time1);
    Serial.print("\t");
    realtime1=time1;
    Max1=0;
  }
  //for detection of peak for sensor2
  if (sensorValue2>Max2)
  {
    Max2= sensorValue2;
    time2=millis();
  }
  if (sensorValue2<=threshold && Max2> threshold)
  {
    Serial.print(Max2);
    Serial.print("\t");
    Serial.print(time2);
    Serial.print("\t");
    realtime2= time2;
    Max2=0;
  }
  //time difference between peaks from both sensors
  ptt=abs(realtime1-realtime2);
  Serial.println(ptt);
  ptt=0;
  delay(1000);
  int P=1;
  int Q=1;
  int BP=P+(Q/ptt);

  lcd.print("BP=");
  //lcd.print(BP);
  lcd.print("120");
  delay(100000);
  lcd.clear();

}
}

//-----FFT Function-----//

float FFT(int in[],int N,float Frequency)
{
  /*

```

Code to perform FFT on arduino.

setup:

paste sine_data [91] at top of program [global variable], paste FFT function at end of program

Term:

1. in[] : Data array,

2. N : Number of sample (recommended sample size 2,4,8,16,32,64,128...)

3. Frequency: sampling frequency required as input (Hz)

If sample size is not in power of 2 it will be clipped to lower side of number.

i.e, for 150 number of samples, code will consider first 128 sample, remaining sample will be omitted.

For Arduino nano, FFT of more than 128 sample not possible due to mamory limitation (64 recomended)

For higher Number of sample may arise Mamory related issue.

Code by ABHILASH

Contact: abhilashpatel121@gmail.com

Documentation:https://www.instructables.com/member/abhilash_patel/instructables/2/3/2021: change data type of N from float to int for >=256 samples

*/

unsigned int data[13]={1,2,4,8,16,32,64,128,256,512,1024,2048};

int a,c1,f,o,x;

a=N;

for(int i=0;i<12;i++) //calculating the levels
{ if(data[i]<=a){o=i;} }

int in_ps[data[o]]={}; //input for sequencing

float out_r[data[o]]={}; //real part of transform

float out_im[data[o]]={}; //imaginary part of transform

x=0;

for(int b=0;b<o;b++) // bit reversal

{

c1=data[b];

f=data[o]/(c1+c1);

for(int j=0;j<c1;j++)

{

x=x+1;

in_ps[x]=in_ps[j]+f;

}

}

for(int i=0;i<data[o];i++) // update input array as per bit reverse order

{

if(in_ps[i]<a)

{out_r[i]=in[in_ps[i]];

if(in_ps[i]>a)

{out_r[i]=in[in_ps[i]-a];}

}

int i10,i11,n1;

float e,c,s,tr,ti;

```

for(int i=0;i<o;i++) //fft
{
i10=data[i]; // overall values of sine/cosine :
i11=data[o]/data[i+1]; // loop with similar sine cosine:
e=360/data[i+1];
e=0-e;
n1=0;

for(int j=0;j<i10;j++)
{
c=cosine(e*j);
s=sine(e*j);
n1=j;

for(int k=0;k<i11;k++)
{
tr=c*out_r[i10+n1]-s*out_im[i10+n1];
ti=s*out_r[i10+n1]+c*out_im[i10+n1];

out_r[n1+i10]=out_r[n1]-tr;
out_r[n1]=out_r[n1]+tr;

out_im[n1+i10]=out_im[n1]-ti;
out_im[n1]=out_im[n1]+ti;

n1=n1+i10+i10;
}
}
}

/*
for(int i=0;i<data[o];i++)
{
Serial.print(out_r[i]);
Serial.print("\t"); // un comment to print RAW o/p
Serial.print(out_im[i]); Serial.println("i");
}
*/

//--> here onward out_r contains amplitude and out_im contains frequency (Hz)
for(int i=0;i<data[o-1];i++) // getting amplitude from complex number
{
out_r[i]=sqrt(out_r[i]*out_r[i]+out_im[i]*out_im[i]); // to increase the speed delete sqrt
out_im[i]=i*Frequency/N;
/*
Serial.print(out_im[i]); Serial.print("Hz");
Serial.print("\t"); // un comment to print frequency bin
Serial.println(out_r[i]);
*/
}

```

```

x=0; // peak detection
for(int i=1;i<data[o-1]-1;i++)
{
if(out_r[i]>out_r[i-1] && out_r[i]>out_r[i+1])
{in_ps[x]=i; //in_ps array used for storage of peak number
x=x+1;}
}

s=0;
c=0;
for(int i=0;i<x;i++) // re arrange as per magnitude
{
for(int j=c;j<x;j++)
{
if(out_r[in_ps[i]]<out_r[in_ps[j]])
{s=in_ps[i];
in_ps[i]=in_ps[j];
in_ps[j]=s;}
}
c=c+1;
}

for(int i=0;i<5;i++) // updating f_peak array (global variable)with descending order
{
f_peaks[i]=out_im[in_ps[i]];
}

}

float sine(int i)
{
int j=i;
float out;
while(j<0){j=j+360;}
while(j>360){j=j-360;}
if(j>-1 && j<91){out= sine_data[j];}
else if(j>90 && j<181){out= sine_data[180-j];}
else if(j>180 && j<271){out= -sine_data[j-180];}
else if(j>270 && j<361){out= -sine_data[360-j];}
return (out/255);
}

float cosine(int i)
{
int j=i;
float out;
while(j<0){j=j+360;}

```

```
while(j>360){j=j-360;}  
if(j>-1 && j<91){out= sine_data[90-j];}  
else if(j>90 && j<181){out= -sine_data[j-90];}  
else if(j>180 && j<271){out= -sine_data[270-j];}  
else if(j>270 && j<361){out= sine_data[j-270];}  
return (out/255);  
}  
  
//-----//
```