



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΚΑΤΑΣΚΕΥΉ ΑΝΑΛΥΤΗ BODE ΜΕ ΧΡΗΣΗ ARDUINO

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΚΟΛΟΚΑΘΗ ΑΓΓΕΛΟΥ

Επιβλέπων: Παρίσης Κωνσταντίνος

Καθηγητής

ΚΟΖΑΝΗ/ΟΚΤΩΜΒΡΙΟΣ/2023

ΑΥΤΗ Η ΣΕΛΙΔΑ ΕΙΝΑΙ ΣΚΟΠΙΜΑ ΛΕΥΚΗ



HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

CONSTRUCTION OF A BODE ANALYZER WITH THE USE OF ARDUINO

THESIS

KOLOKATHIS AGGELOS

SUPERVISOR: Constantinos Parisses

Professor

KOZANI/OCTOBER/2023

ΑΥΤΗ Η ΣΕΛΙΔΑ ΕΙΝΑΙ ΣΚΟΠΙΜΑ ΛΕΥΚΗ



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Πτυχιακή Εργασία με τίτλο “_____” καθώς

και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. _____ αποτελεί

αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Ονοματεπώνυμο Φοιτητή & Επιβλέποντα, Έτος, Πόλη

Copyright (C) _____, _____, _____, _____

Υπογραφή Φοιτητή: _____

ΑΥΤΗ Η ΣΕΛΙΔΑ ΕΙΝΑΙ ΣΚΟΠΙΜΑ ΛΕΥΚΗ

Περίληψη

Η παρούσα πτυχιακή δημιουργήθηκε στα πλαίσια του οδηγού σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών (οδηγός σπουδών Ηλεκτρολόγων Μηχανικών Τ.Ε.) του Πανεπιστημίου Δυτικής Μακεδονίας. Στόχος ήταν η δημιουργία μιας κατασκευής που να έχει τη δυνατότητα να μελετά την απόκριση συχνότητας ηλεκτρικών φίλτρων. Στην μελέτη αυτή περιλαμβάνονται μετρήσεις που μπορούν να χρησιμοποιηθούν για την κατασκευή διαγραμμάτων Bode, ώστε να υπάρχει η δυνατότητα να μελετηθούν τα φίλτρα μέσω των διαγραμμάτων Bode. Ωστόσο η κατασκευή, για την υλοποίηση των σκοπών αυτών θα πρέπει να έχει τη δυνατότητα ανίχνευσης της συχνότητας, δυο διαφορετικών τάσεων -η μια εισόδου και η άλλη εξόδου στο φίλτρο που θα μελετηθεί- και της διαφοράς φάσης. Η κατασκευή αυτή ακριβώς για το λόγο ότι είναι εκπαιδευτική θα έπρεπε πάνω από όλα να είναι κατανοητή στην χρήση του και με χαμηλό κόστος υλοποίησης από υλικά που υπάρχουν στην αγορά έτσι ώστε να παρακάμπτει τη χρήση παλμογράφου που είναι ένα όργανο μέτρησης μη διαθέσιμο στους φοιτητές εκτός εργαστηριακών χώρων.

Εμβαθύνοντας περαιτέρω στην πτυχιακή εργασία, στο πρώτο κεφάλαιο γίνεται μια μικρή εισαγωγή με μερικές λεπτομέρειες για το αντικείμενο της εργασίας. Στη συνέχεια του δευτέρου κεφαλαίου αναλύονται οι μικροελεγκτές και η πλακέτα που χρησιμοποιήθηκε για την ανίχνευση των τιμών που προαναφέρθηκαν. Στο τρίτο κεφάλαιο, παρουσιάζεται η θεωρητική ανάλυση στο πεδίο της συχνότητας, που χρησιμοποιούνται τα διαγράμματα bode και γίνεται η μαθηματική ανάλυση του φίλτρου που έχει επιλεγεί προς μελέτη. Στο τέταρτο κεφάλαιο αναλύονται ξεχωριστά τα κυκλώματα και οι πλακέτες που χρησιμοποιήθηκαν για την καταγραφή των τιμών. Συνεχίζοντας, το πέμπτο κεφάλαιο παρουσιάζει τις τιμές που έχουν συλλεχθεί από τους τρεις διαφορετικούς τρόπους μελέτης του φίλτρου, ενώ περαιτέρω, το έκτο κεφάλαιο συνοψίζει τα συμπεράσματα και τις μελλοντικές επεκτάσεις για βελτίωση της κατασκευής.

Λέξεις Κλειδιά: ΔΙΑΓΡΑΜΜΑ BODE; ARDUINO UNO; ΔΙΑΦΟΡΑ ΦΑΣΗΣ;
ΑΠΟΚΡΙΣΗ ΣΥΧΝΟΤΗΤΑΣ; ΒΑΘΥΠΕΡΑΤΟ ΦΙΛΤΡΟ;

Abstract

This bachelor's thesis was created as part of the curriculum of the Department of Electrical and Computer Engineering (Electrical Engineering, Technological Educational Institute program) at the University of Western Macedonia. Its objective was to develop a device capable of studying the frequency response of electrical filters. The study includes measurements that can be used to create Bode plots, allowing for the analysis of filters through Bode diagrams. However, in order to achieve these goals, the device must be capable of detecting the frequency, measuring two different voltages - one at the input and the other at the output of the filter under study - and the phase difference. This construction, being educational in nature, should be designed to be understandable and cost-effective, using materials readily available in the market, to avoid the need for an oscilloscope, which is not readily accessible to students outside of laboratory environments.

Delving further into the thesis, the first chapter provides a brief introduction with some details about the subject of the work. In the following second chapter, microcontrollers and the board used for detecting the values mentioned earlier are analyzed. In the third chapter, the theoretical analysis in the frequency domain is presented, where Bode diagrams are employed, and the mathematical analysis of the selected filter for study is performed. In the fourth chapter, the circuits and boards used for recording the values are separately discussed. Moving on, the fifth chapter presents the values collected through three different methods of studying the filter, while the sixth chapter summarizes the conclusions and future extensions for improving the construction.

Keywords: BODE PLOT; ARDUINO UNO; PHASE DIFFERENCE; FREQUENCY; LOW PASS FILTER;

Ευχαριστίες

Η παρούσα πτυχιακή εργασία που πραγματοποιήθηκε στο πανεπιστήμιο Δυτικής Μακεδονίας, στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών και συγκεκριμένα στο εργαστήριο των συστημάτων αυτομάτου ελέγχου, κατά το έτος 2023 θα ήταν αδύνατη χωρίς την ανάθεση του επιβλέποντος καθηγητή μου, κυρίου Παρίση Κωνσταντίνου. Του χρωστώ ένα ευχαριστώ για την εμπιστοσύνη που έδειξε στο πρόσωπο μου αναθέτοντας μου το συγκεκριμένο θέμα.

Βαθιά ευγνωμοσύνη και θερμές ευχαριστίες θα ήθελα να πω στον κύριο Ιωάννη Βανδίκη, Ε.ΔΙ.Π του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών στο Πανεπιστήμιο Δυτικής Μακεδονίας για την άριστη αγωγή που είχαμε στα πλαίσια εκπόνησης αυτής της εργασίας, τον πολύτιμο χρόνο που μου χάρισε όπως και σημαντικές εξηγήσεις με τις άριστες γνώσεις που διαθέτει εμπράκτως πάνω στον τομέα των ηλεκτρονικών, αλλά και για την προθυμία του και τη βοήθεια που ποτέ δεν δίστασε να μου δώσει.

Ευχαριστώ πολύ επίσης τον υποψήφιο διδάκτορα Μόσχο Ιωάννη του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστήμιο Δυτικής Μακεδονίας για την συνεχή συμπαράσταση και καθοδήγηση που είχαμε όλον αυτό τον καιρό στα πλαίσια της υλοποίησης των επιμέρους ζητημάτων και λεπτομερειών της πτυχιακής.

Εγκάρδιες ευχαριστίες για όλη την ψυχολογική και ηθική υποστήριξη στην φίλη μου Καβουρίδου Βασιλική, πτυχιούχο του τμήματος Ελληνικής Φιλολογίας, Σχολή Κλασσικών & Ανθρωπιστικών Σπουδών του Δημοκρίτειου Πανεπιστημίου Θράκης δωρίζοντας μου αμέτρητη συμπαράσταση και τη διάθεση της προς βοήθεια πάσης φύσεως.

ΑΥΤΗ Η ΣΕΛΙΔΑ ΕΙΝΑΙ ΣΚΟΠΙΜΑ ΛΕΥΚΗ

Περιεχόμενα

Περίληψη	7
Abstract	9
Ευχαριστίες	10
Περιεχόμενα	12
Κατάλογος Σχημάτων	14
Κατάλογος Πινάκων	15
Πρόλογος	16
Κεφάλαιο 1: Εισαγωγή	16
1.1 Αντικείμενο της διπλωματικής	16
1.2 Οργάνωση του τόμου	16
Κεφάλαιο 2: Ο μικροελεγκτής Arduino	17
2.1 Οι Μικροελεγκτές και οι εφαρμογές τους	17
2.2 Πλακέτες Arduino	18
2.2.1 Τροφοδοσία Arduino Uno	20
2.2.2 Μνήμη Arduino Uno	20
2.2.3 Ακροδέκτες Arduino Uno	21
2.3 Γραφικό περιβάλλον Arduino Uno	21
2.4 Προγραμματισμός Arduino Uno	26
Κεφάλαιο 3: Ανάλυση στο πεδίο της συχνότητας	29
3.1 Χρήσεις διαγράμματος Bode	29
3.2 Θεωρητική ανάλυση στο πεδίο της συχνότητας	31
3.3 Ανάλυση βαθυπερατού φίλτρο στο πεδίο της συχνότητας	31
Κεφάλαιο 4: Κύκλωμα διαγράμματος Bode από μετρήσεις	35
4.1 Υλοποίηση κυκλώματος μέτρησης μέτρου και φάσης	35
4.1.1 Κύκλωμα μέτρησης τάσης	35
4.1.2 Κύκλωμα μέτρησης φάση	37
4.1.3 Κύκλωμα μέτρησης συχνότητας	38
4.1.4 Κύκλωμα μέτρησης φάσης και μέτρου	39

Κεφάλαιο 5: Μετρήσεις από πραγματικό κύκλωμα για δημιουργία διαγράμματος Bode	41
5.1 Θεωρητικές τιμές	41
5.2 Μετρήσεις με τη χρήση παλμογράφου και σχήματα Lissajous	42
5.3 Μετρήσεις με τη χρήση της εκπαιδευτικής συσκευής.	44
5.4 Σύγκριση και αποκλίσεις τιμών	47
Κεφάλαιο 6: Επίλογος	48
6.1 Συμπεράσματα	48
6.2 Μελλοντικές επεκτάσεις	48
Παραρτήματα: Κώδικας Arduino	49
Βιβλιογραφία	54

Κατάλογος Σχημάτων

Εικόνα 1. Arduino Uno.....	19
Εικόνα 2. Είσοδοι έξοδοι τροφοδοσίας Arduino Uno	20
Εικόνα 3. Ακροδέκτες Arduino Uno.....	21
Εικόνα 4. Γραφικό περιβάλλον Arduino IDE.....	22
Εικόνα 5. Κύκλωμα βαθυπερατού φίλτρου [29].....	31
Εικόνα 6. Κύκλωμα μετατροπής με τροφοδοσία για το ολοκληρωμένα [26].	36
Εικόνα 7. Κύκλωμα μετατροπής από Vrms σε VDC.	36
Εικόνα 8.Υλοποιημένη διάτρητη πλακέτα για το κύκλωμα μετατροπής τάσης.....	37
Εικόνα 9. Διάταξη και τυπωμένο κύκλωμα σε πλακέτα διπλής όψης [27].	37
Εικόνα 10. Υλοποιημένο κύκλωμα σε επιχαλκωμένη πλακέτα.	38
Εικόνα 11. Γεννήτρια συχνοτήτων.	39
Εικόνα 12. Ολόκληρη η κατασκευή καταγραφής μετρήσεων.	40
Εικόνα 13. Διάγραμμα Bode θεωρητικής ανάλυσης.	42
Εικόνα 14 Υπολογισμός διαφοράς φάσης με χρήση x-y απεικόνισης στον παλμογράφο.....	42
Εικόνα 15. Σχήμα Lissajous (x-y) στα 500Hz	43
Εικόνα 16. Διαγράμματα Bode θεωρητικού μοντέλου και μετρήσεων από παλμογράφο.	44
Εικόνα 17. Ενδείξεις μετρήσεων στα 100Hz.	45
Εικόνα 18. Επαλήθευση μετρήσεων μέσω παλμογράφου.	45
Εικόνα 19. Διαγράμματα Bode θεωρητικού μοντέλου και μετρήσεων από παλμογράφο.	46
Εικόνα 20. Σύγκριση διαγραμμάτων Bode.	47

Κατάλογος Πινάκων

Πίνακας 1. Τεχνικά χαρακτηριστικά Arduino Uno	19
Πίνακας 2. Μετρήσεις από θεωρητική μελέτη εξισώσεων.....	41
Πίνακας 3. Μετρήσεις με τη χρήση παλμογράφου.....	43
Πίνακας 4. Μετρήσεις με τη χρήση της κατασκευής.	46

Πρόλογος

Στην συγκεκριμένη πτυχιακή κατασκευάστηκε ένα οικονομικό σύστημα μέτρησης τιμών για την υλοποίηση διαγραμμάτων bode. Τα διαγράμματα bode χρησιμεύουν στην ηλεκτρολογική μηχανική και της μηχανική συστημάτων ελέγχου για την ανάλυση και την κατανόηση της απόκρισης συχνότητας των γραμμικών συστημάτων. Ο τόπος διεξαγωγής των δοκίμων και υλοποίησης της πτυχιακής ήταν το εργαστήριο των συστημάτων αυτομάτου ελέγχου που εδρεύει στην πανεπιστημιούπολη του νομού Κοζάνης.

Κεφάλαιο 1: Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται το αντικείμενο της πτυχιακής, που είναι η κατασκευή ενός συστήματος που έχει ως αποτέλεσμα, την εμφάνιση τιμών με τις οποίες έχουμε τη δυνατότητα να κατασκευάσουμε διαγράμματα bode. Επιπροσθέτως αναλύουμε την επιμελή εκτέλεση της, χωρισμένη σε πέντε βασικά κεφάλαια.

1.1 Αντικείμενο της διπλωματικής

Κυρίως στόχος της διπλωματικής είναι η μελέτη ενός εκπαιδευτικού συστήματος που μετράει τη διαφορά φάσης και μέτρου ηλεκτρικών φίλτρων που χρησιμεύουν στην διαμόρφωση, το φιλτράρισμα ή ακόμη και τη ρύθμιση σημάτων με βάση το περιεχόμενο της συχνότητάς τους. Για να συμβεί αυτό θα πρέπει να συγκρίνουμε τις τιμές που καταγράψαμε από τα μηχανήματα του εργαστηρίου (παλμογράφος & γεννήτρια συχνότητων) με τις τιμές που μας δίνει το σύστημα που κατασκευάστηκε και τις τιμές του θεωρητικού μαθηματικού μοντέλου. Ωστόσο για να είναι η κατασκευή εύχρηστη θα πρέπει να είναι εύκολη στην σύνδεση και την αποσύνδεση της καθώς και το σύστημα θα πρέπει να είναι χρηστικό στις μετρήσεις του και με καλή ακρίβεια. Όντας όμως ένα κατασκευαστικό εκπαιδευτικό σύστημα θα πρέπει να είναι εύκολο στην κατανόηση και τη λειτουργία του για τον φοιτητή.

1.2 Οργάνωση του τόμου

Στο πρώτο κεφάλαιο παρουσιάζεται περιληπτικά το αντικείμενο και ο στόχος για τον οποίο δημιουργήθηκε η συγκεκριμένη κατασκευή. Στην συνέχεια αναλύεται τι είναι ένας μικροελεγκτής Arduino, που χρησιμοποιούνται αυτοί οι μικροελεγκτές και ποιος ήταν ο μικροελεγκτής που χρησιμοποιήθηκε για την υλοποίηση της κατασκευής. Στο τρίτο κεφάλαιο μελετάται η θεωρητική προσέγγιση του πεδίου της συχνότητας. Στο τέταρτο κεφάλαιο εξετάζεται ο τρόπος λειτουργίας των κυκλωμάτων της φάσης, της τάσης και της συχνότητας και πως όλα αυτά συνδέονται με σκοπό να παρθούν οι κατάλληλες μετρήσεις. Στο πέμπτο κεφάλαιο παρουσιάζεται το διάγραμμα bode που έχει δημιουργηθεί χρησιμοποιώντας τις μετρήσεις του προηγούμενου κεφαλαίου. Στο έκτο και τελευταίο κεφάλαιο καταγράφονται τα συμπεράσματα του εκπαιδευτικού συστήματος μέτρησης που έχει κατασκευαστεί και αναδεικνύονται μερικές ιδέες για μελλοντική επέκταση της κατασκευής.

Κεφάλαιο 2: Ο μικροελεγκτής Arduino

Ο μικροελεγκτής είναι ένας μικροϋπολογιστής, ο οποίος σχεδιάστηκε για να εκτελεί ειδικές λειτουργίες και εργασίες ελέγχου σε διάφορα συστήματα. Αυτό το ενσωματωμένο σύστημα χρησιμοποιείται ευρέως σε πολλούς τομείς, όπως η βιομηχανία, οι οικιακές συσκευές, η αυτοματοποιημένη παραγωγή και η ιατρική.

2.1 Οι Μικροελεγκτές και οι εφαρμογές τους

Οι μικροελεγκτές αποτελούν την πυρήνα του πεδίου της ενσωματωμένης ηλεκτρονικής. Οι περισσότεροι μικροελεγκτές χρησιμοποιούνται για την εκτέλεση συγκεκριμένων λειτουργιών ή για τον έλεγχο ενός συγκεκριμένου συστήματος. Ένας από τους πιο γνωστούς μικροελεγκτές είναι ο Arduino, ο οποίος χρησιμοποιείται ευρέως στην εκπαίδευση και τους αυτοματισμούς.

Οι μικροελεγκτές αποτελούνται από έναν κύριο μικροεπεξεργαστή, ο οποίος είναι υπεύθυνος για την εκτέλεση των εντολών και από περιφερειακούς ελεγκτές, των οποίων η αρμοδιότητα είναι η αλληλεπίδραση με το περιβάλλον. Ο μικροεπεξεργαστής είναι ο εγκέφαλος του μικροελεγκτή και εκτελεί τις εντολές που του δίνονται μέσω του προγράμματος που φορτώνεται στη μνήμη του. Οι περιφερειακοί ελεγκτές παρέχουν διάφορες λειτουργίες, όπως αναλογικές και ψηφιακές εισοδοί, έξοδοι, αισθητήρες, επικοινωνία με άλλες συσκευές και πολλά άλλα [1].

Ο μικροελεγκτής είναι σχεδιασμένος να είναι αξιόπιστος, αποδοτικός και εύκολος στη χρήση. Η προγραμματιζόμενη λογική του, του επιτρέπει να εκτελεί ποικιλία λειτουργιών και να προσαρμόζεται ανάλογα με τις απαιτήσεις του συστήματος. Οι γλώσσες προγραμματισμού που χρησιμοποιούνται για τον προγραμματισμό των μικροελεγκτών περιλαμβάνουν την C, την C++, και την Assembly.

Ένας μικροελεγκτής μπορεί να χρησιμοποιηθεί για να ελέγχει και να παρακολουθεί διάφορες διεργασίες. Μπορεί να διαβάζει αισθητήρες για να συλλέξει δεδομένα, να εκτελεί αλγορίθμους και να αποδίδει αποτελέσματα είτε να ελέγχει ενεργειακές μονάδες ή κινητήρες για να πραγματοποιήσει κάποιες ενέργειες. Ο μικροελεγκτής λαμβάνει αποφάσεις βάσει των προγραμμάτων που του έχουν φορτωθεί, των εισόδων που λαμβάνει και των συνθηκών που καταγραφεί.

Με τη χρήση αισθητήρων και ασύρματων δικτύων, οι μικροελεγκτές μπορούν να συλλέξουν και να αναλύσουν δεδομένα από το περιβάλλον και να λάβουν αποφάσεις βάσει αυτών των δεδομένων. Αυτό επιτρέπει την αυτοματοποίηση και τον έλεγχο έξυπνων συστημάτων, όπως έξυπνα σπίτια, έξυπνες πόλεις και έξυπνες καλλιέργειες.

Ένας από τους σημαντικούς τομείς εφαρμογής των μικροελεγκτών είναι οι βιομηχανικές εφαρμογές. Χρησιμοποιούνται για τον έλεγχο και την αυτοματοποίηση διεργασιών παραγωγής σε εργοστάσια, εξασφαλίζοντας έτσι την ακρίβεια και την διεκπεραίωση των διαδικασιών με αποτελεσματικότητα. Επίσης, χρησιμοποιούνται στις συσκευές ελέγχου κίνησης, όπως ρομπότ και CNC μηχανές, για τον συγχρονισμό και τον έλεγχο των κινητήρων [2].

Ένα παράδειγμα εφαρμογής μικροελεγκτή είναι ένα σύστημα αυτοματισμού σε ένα εργοστάσιο παραγωγής. Ο μικροελεγκτής μπορεί να ελέγχει τη θερμοκρασία, την υγρασία, την πίεση και πληθώρα παραμέτρων στο εργοστάσιο και να προβαίνει σε ρυθμίσεις για να διατηρήσει τις επιθυμητές συνθήκες λειτουργίας. Μπορεί επίσης να ελέγχει τις μηχανές παραγωγής και να διασφαλίζει τη σωστή λειτουργία τους.

Ένας ακόμα τομέας εφαρμογής των μικροελεγκτών είναι η ιατρική τεχνολογία. Χρησιμοποιούνται σε ιατρικές συσκευές και εξοπλισμό, όπως απινιδωτές, μηχανές αναισθησίας και παρακολούθησης βιολογικών λειτουργιών. Οι μικροελεγκτές επιτρέπουν τον έλεγχο και την ακρίβεια σε αυτές τις συσκευές, προσφέροντας τη δυνατότητα για ακριβή διάγνωση και θεραπεία [3].

Οι μικροελεγκτές έχουν εξελιχθεί σημαντικά τα τελευταία χρόνια. Έχουν γίνει μικρότεροι, πιο ισχυροί και ενεργειακά αποδοτικότεροι. Η τεχνολογία IoT (Internet of Things) έχει επίσης επηρεάσει την εξέλιξη των μικροελεγκτών, επιτρέποντάς τους να συνδεθούν με το Διαδίκτυο και να ανταλλάσσουν δεδομένα με άλλες συσκευές. Μερικές πιο εξειδικευμένες χρήσεις μικροελεγκτών είναι οι εξής:

- **Ρομποτική:** Οι μικροελεγκτές αποτελούν τον εγκέφαλο πολλών ρομπότ και αυτοματοποιημένων συστημάτων. Χρησιμοποιούνται για τον έλεγχο της κίνησης, την αντίδραση σε αισθητήρες και την εκτέλεση πολύπλοκων εργασιών σε βιομηχανικά και ερευνητικά περιβάλλοντα.
- **Ενεργειακή απόδοση:** Οι μικροελεγκτές χρησιμοποιούνται για τον έλεγχο και την παρακολούθηση ενεργειακών συστημάτων. Μπορούν να βοηθήσουν στην αυτοματοποίηση και τον έλεγχο των φωτιστικών συστημάτων, των κλιματιστικών, των ηλιακών πάνελ και άλλων συστημάτων ενεργειακής απόδοσης.
- **Αεροναυπηγική και διαστημική τεχνολογία:** Οι μικροελεγκτές χρησιμοποιούνται σε αεροναυπηγικά και διαστημικά συστήματα για τον έλεγχο και την παρακολούθηση των λειτουργιών, καθώς και για την αυτονομία και ασφάλεια των συστημάτων.
- **Κατασκευές:** Οι μικροελεγκτές χρησιμοποιούνται στην κατασκευή για τον έλεγχο και την αυτοματοποίηση διαδικασιών όπως η διανομή υλικών, ο έλεγχος του φωτισμού, ο έλεγχος της θέρμανσης και ο έλεγχος ασφαλείας.

Αυτές είναι μόνο μερικές από τις εξειδικευμένες χρήσεις των μικροελεγκτών. Οι δυνατότητες τους είναι ευρέως εφαρμόσιμες και εξαρτώνται από τη φαντασία και τη δημιουργικότητα των προγραμματιστών και των μηχανικών που τους χρησιμοποιούν.

Συνοψίζοντας, ο μικροελεγκτής είναι ένα ενσωματωμένο σύστημα που χρησιμοποιείται για τον έλεγχο, την διαχείριση και την εκτέλεση ειδικών λειτουργιών σε διάφορα συστήματα. Με την αξιοπιστία, την αποδοτικότητα και την ευελιξία του, ο μικροελεγκτής συμβάλλει στην αυτοματοποίηση και τη βελτίωση της λειτουργίας σε πολλούς τομείς. Από τις βιομηχανικές εφαρμογές μέχρι το IoT και την ιατρική τεχνολογία, οι μικροελεγκτές παρέχουν την αυτοματοποίηση, τον έλεγχο και την ευελιξία που απαιτούνται σε αυτές τις προηγμένες εφαρμογές.

Με την τεχνολογική πρόοδο, οι μικροελεγκτές αναμένεται να συνεχίσουν να εξελίσσονται και να προσφέρουν ακόμα περισσότερες δυνατότητες και λειτουργικότητα στον κόσμο της τεχνολογίας.

2.2 Πλακέτες Arduino

Οι πλακέτες Arduino αποτελούνται από έναν μικροελεγκτή, όπως τον ATmega328P και ένα περιβάλλον προγραμματισμού που βασίζεται στη γλώσσα προγραμματισμού Wiring και το περιβάλλον ανάπτυξης Arduino IDE. Η πλατφόρμα παρέχει μια απλή και φιλική διεπαφή για την προγραμματισμό και τον έλεγχο των μικροελεγκτών [4].

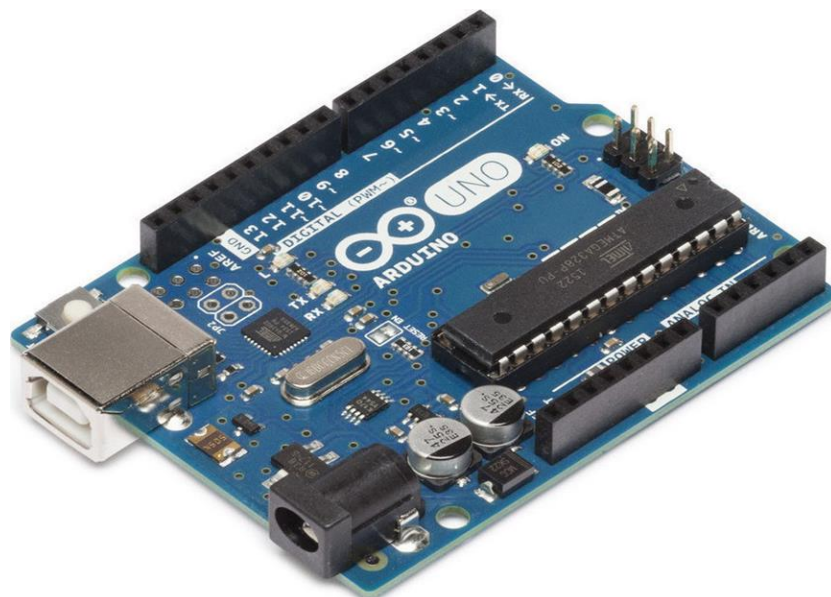
Βασικό χαρακτηριστικό του Arduino είναι η δυνατότητα σύνδεσής του με αισθητήρες, επαγγελματικά ή ερασιτεχνικά περιφερειακά, όπως LED, οθόνες, κουμπιά, αισθητήρες θερμοκρασίας και επιταχυνσιόμετρα. Αυτό επιτρέπει τη δημιουργία διάφορων εφαρμογών σε έξυπνα συστήματα κατοικιών στη ρομποτική και σε κινητικά έργα.

Ένα σημαντικό πλεονέκτημα του Arduino είναι η διαθεσιμότητα μιας μεγάλης κοινότητας που το υποστηρίζει. Υπάρχουν πολλοί χρήστες και προγραμματιστές που μοιράζονται τις γνώσεις, τις ιδέες και τα έργα τους μέσω διαδικτυακών φόρουμ, ιστοσελίδων και κοινοτήτων. Αυτό καθιστά

ευκολότερη την εκμάθηση και την ανάπτυξη προγραμμάτων με τη χρήση του Arduino.

Το Arduino έχει γίνει εξαιρετικά δημοφιλές στον κόσμο του ερασιτεχνικού ανοιχτού κώδικα και του rapid prototyping. Επίσης, χρησιμοποιείται ευρέως στον εκπαιδευτικό τομέα για την εισαγωγή των μαθητών και των φοιτητών στον κόσμο της ηλεκτρονικής και του προγραμματισμού.

Συνολικά, το Arduino αναμένεται να συνεχίσει να προσφέρει μια εύκολη, προσιτή και ευέλικτη πλατφόρμα για την ανάπτυξη έξυπνων ηλεκτρονικών συστημάτων και τη δημιουργία καινοτόμων προτύπων.



Εικόνα 1. Arduino Uno.

Πίνακας 1. Τεχνικά χαρακτηριστικά Arduino Uno

Μικροελεγκτής	ATMEGA328
Τάση λειτουργίας	5V
Τάση εισόδου	7-12V
Όρια τάσης εισόδου	6-20V
Ψηφιακοί ακροδέκτες I/O	14, (6 PWM έξοδοι)
Αναλογικοί ακροδέκτες εισόδου	6
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40mA
Μνήμη flash	32KB(ATMEGA328)
Μνήμη SRAM	2KB(ATMEGA328)
Μνήμη EEPROM	1KB(ATMEGA328)
Ταχύτητα ρολογιού	16MHz

2.2.1 Τροφοδοσία Arduino Uno

Το Arduino Uno λειτουργεί με ρεύμα το οποίο παρέχεται είτε μέσω θύρας USB από τον υπολογιστή ή μέσω φισ τύπου jack 2.1mm που η ανάλογη υποδοχή της πλακέτας βρίσκεται στην κάτω αριστερή γωνία. Για την σωστή λειτουργία της πλακέτας η εξωτερική τροφοδοσία καλό θα ήταν να περιορίζεται από 7 έως 12V. Η παρακάτω εικόνα μας υποδεικνύει τις εισόδους και εξόδους τροφοδοσίας του Arduino Uno [5].



Εικόνα 2. Είσοδοι έξοδοι τροφοδοσίας Arduino Uno

Pin τροφοδοσίας Arduino uno:

- Vin: Χρησιμοποιούμε το συγκεκριμένο Pin για να τροφοδοτήσουμε με ενέργεια από εξωτερική πηγή την πλακέτα, δηλαδή είναι η τάση εισόδου.
- 5V: Η πλακέτα μπορεί να τροφοδοτηθεί με ρεύμα είτε από την υποδοχή τροφοδοσίας DC (7 - 12V), την υποδοχή USB (5V) ή τον ακροδέκτη VIN της πλακέτας (7-12V).
- 3.3V: το ολοκληρωμένο FTDI μας δίνει αυτή τη τάση. Το όριο άντλησης ρεύματος είναι 50mA.
- GND: Είσοδοι γείωσης.
- IOREF: Αυτή η ακίδα στην πλακέτα Arduino παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής. Μια σωστά διαμορφωμένη θωράκιση μπορεί να διαβάσει την τάση του ακροδέκτη IOREF και να επιλέξει την κατάλληλη πηγή ισχύος ή να ενεργοποιήσει τους μεταφραστές τάσης στις εξόδους να λειτουργούν με τα 5V ή 3,3V.

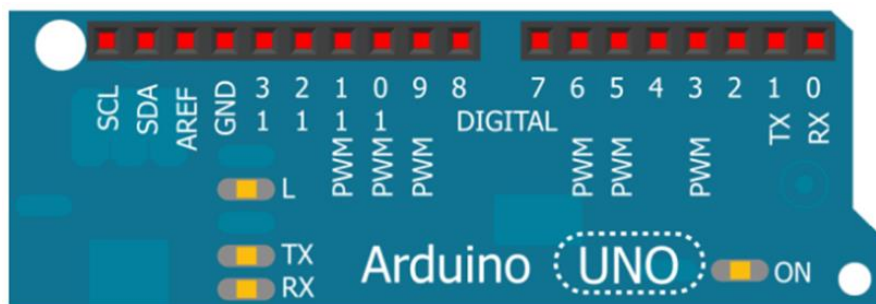
2.2.2 Μνήμη Arduino Uno

Ο ATmega328 είναι ένα δημοφιλές μικροελεγκτής που παράγεται από την Microchip Technology, πρώην Atmel Corporation. Ο ATmega328 είναι γνωστός για την απλότητα, την ευελιξία και τη χαμηλή κατανάλωση ενέργειας, καθιστώντας τον μια κατάλληλη επιλογή για ένα ευρύ φάσμα εφαρμογών [6]. Τεχνικά χαρακτηριστικά ATmega328:

- Flash Memory: Διαθέτει 32KB μνήμης προγράμματος Flash για την αποθήκευση του κωδικού εφαρμογής του χρήστη.
- SRAM: Το ATmega328 διαθέτει 2KB SRAM (Static Random Access Memory), η οποία χρησιμοποιείται για την αποθήκευση δεδομένων κατά την εκτέλεση του προγράμματος.
- EEPROM: Περιλαμβάνει 1 KB EEPROM (Electrically Erasable Programmable Read-Only Memory), η οποία χρησιμοποιείται για μη πτητική αποθήκευση δεδομένων.
- Μετατροπέας αναλογικού σε ψηφιακό (ADC): Το ATmega328 διαθέτει ADC 10-bit με έως και 8 κανάλια, επιτρέποντάς του να μετατρέπει αναλογικά σήματα σε ψηφιακές τιμές.
- Χρονοδιακόπτες και μετρητές: Περιλαμβάνει πολλά χρονομέτρα/μετρητές, συμπεριλαμβανομένων δύο χρονομέτρων 8-bit και ενός χρονοδιακόπτη 16-bit, που μπορούν να χρησιμοποιηθούν για διάφορες εργασίες χρονισμού και ελέγχου.

- Διεπαφές επικοινωνίας: Το τσιπ υποστηρίζει πρωτόκολλα επικοινωνίας USART (Universal Synchronous and Asynchronous Receiver and Transmitter), SPI (Serial Peripheral Interface) και I2C (Inter-Integrated Circuit), καθιστώντας το κατάλληλο για διασύνδεση με άλλες συσκευές και αισθητήρες.
- Λειτουργίες χαμηλής κατανάλωσης: Το ATmega328 προσφέρει διάφορες λειτουργίες χαμηλής κατανάλωσης, οι οποίες μπορούν να βοηθήσουν στην εξοικονόμηση ενέργειας σε εφαρμογές που τροφοδοτούνται από μπαταρία.
- Τάση λειτουργίας: Λειτουργεί συνήθως στα 5V, αλλά μπορεί επίσης να λειτουργήσει σε χαμηλότερα επίπεδα τάσης με μειωμένες συχνότητες.

2.2.3 Ακροδέκτες Arduino Uno



Εικόνα 3. Ακροδέκτες Arduino Uno.

Καθένας από τους 14 ακροδέκτες του Uno μπορούν να χρησιμοποιηθούν σαν είσοδοι και σαν έξοδοι χρησιμοποιώντας τις κατάλληλες εντολές με τάση λειτουργίας 5V. Κάθε ακροδέκτης έχει προτεινόμενο όριο ρεύματος τα 20mA ενώ ανώτερο όριο λειτουργίας είναι τα 40mA καθώς παραπάνω από αυτό το όριο μπορεί να προκαλέσει μόνιμη βλάβη σε κάποιο ακροδέκτη [7]. Πιο αναλυτικά τα χαρακτηριστικά των θυρών είναι:

- Serial: 0 (RX) and 1 (TX). Χρησιμοποιείται για να λαμβάνει (RX) και να μεταδίδει (TX) σειριακών δεδομένων TTL. Αυτοί οι ακροδέκτες συνδέονται αντιστοίχως με τους ακροδέκτες του μικροεπεξεργαστή.
- PWM: 3, 5, 6, 9, 10, and 11. Παρέχει 8-bit PWM έξοδο με την analogWrite() εντολή.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτά τα pin υποστηρίζουν επικοινωνία SPI χρησιμοποιώντας τη βιβλιοθήκη SPI.
- LED: Υπάρχει μια ενσωματωμένη λυχνία LED που οδηγείται από την ψηφιακή ακίδα 13. Όταν η ακίδα είναι HIGH τιμή, η λυχνία LED είναι αναμμένη, όταν η ακίδα είναι LOW, είναι σβηστή.
- TWI: A4 ή SDA pin και A5 ή SCL pin. Υποστήριξη επικοινωνίας TWI χρησιμοποιώντας τη βιβλιοθήκη Wire.

2.3 Γραφικό περιβάλλον Arduino Uno

Το Arduino IDE (Integrated Development Environment) είναι ένα περιβάλλον ανάπτυξης λογισμικού που έχει σχεδιαστεί ειδικά για τον προγραμματισμό και την ανάπτυξη λογισμικού στις πλακέτες Arduino. Είναι μια εύχρηστη εφαρμογή που διευκολύνει την γραφή κώδικα και την μεταφόρτωση του στις πλακέτες Arduino [8]. Το Arduino IDE παρέχει τις ακόλουθες βασικές λειτουργίες:

1. Επεξεργασία κώδικα: Επιτρέπει την εισαγωγή, την επεξεργασία και την αποθήκευση του κώδικα που γράφεται για το Arduino. Το Arduino IDE παρέχει ένα περιβάλλον επεξεργασίας κώδικα όπου γράφεται, μεταποιείται και αποθηκεύεται ο κώδικας για το πρόγραμμα που θέλουμε να φορτώσουμε στο Arduino.
2. Μεταφόρτωση κώδικα: Μπορούμε να μεταφορτώσουμε τον κώδικα που έχουμε γράψει στο Arduino μέσω του USB που συνδέει τον υπολογιστή με την πλακέτα Arduino. Επίσης, μέσω του Arduino IDE υπάρχει η δυνατότητα να μεταφορτωθεί ο κώδικας που έχει γραφεί στο Arduino, ώστε να εκτελεί τις συγκεκριμένες λειτουργίες που έχουν προγραμματιστεί.
3. Παρακολούθηση σειριακής επικοινωνίας: Το IDE μπορεί να παρακολουθεί τις εξόδους και εισόδους που στέλνονται μέσω της σειριακής διεπαφής του Arduino, επιτρέποντας τον έλεγχο και τις διαδικασίες εκτέλεσης του προγράμματός. Το IDE παρέχει μια κονσόλα παρακολούθησης σειριακής επικοινωνίας που επιτρέπει στον χρήστη την παρακολούθηση των δεδομένων που ανταλλάσσονται μεταξύ του Arduino και του υπολογιστή.
4. Παραδείγματα κώδικα: Περιλαμβάνει μια συλλογή από παραδείγματα κώδικα που μπορούν να χρησιμοποιηθούν ως αφετηρία για την ανάπτυξη του κάθε αρχάριου και μη προγραμματιστή.
5. Διαχείριση βιβλιοθηκών: Το IDE διαθέτει μια βιβλιοθήκη που περιέχει πολλές χρήσιμες βιβλιοθήκες που μπορούν να χρησιμοποιηθούν για να διευκολυνθεί ο προγραμματισμός και να επεκταθούν οι δυνατότητες του Arduino.
6. Επαλήθευση κώδικα: Το IDE ελέγχει τον κώδικα για συντακτικά λάθη και παρέχει ειδοποιήσεις για τυχόν προβλήματα που εντοπίζει.

Το Arduino IDE είναι δωρεάν, ανοιχτού κώδικα και είναι διαθέσιμο για διάφορα λειτουργικά συστήματα, όπως Windows, macOS και Linux. Είναι ιδανικό για αρχάριους προγραμματιστές και είναι το κύριο περιβάλλον προγραμματισμού που χρησιμοποιείται από χρήστες που ασχολούνται με τις πλακέτες Arduino (εικόνα 4).



Εικόνα 4. Γραφικό περιβάλλον Arduino IDE.

- **VERIFY/UPLOAD:** Όταν πατηθεί το κουμπί "Verify", το Arduino IDE εκτελεί μια διαδικασία ελέγχου του κώδικα που έχει γραφεί για συντακτικά λάθη και άλλα προβλήματα. Κατά την διάρκεια αυτής της διαδικασίας, το IDE θα ελέγξει αν ο κώδικας συμμορφώνεται με τη σύνταξη της γλώσσας προγραμματισμού που χρησιμοποιείται (κυρίως βασισμένη σε C++), αν είναι σωστά ορισμένες οι μεταβλητές, αν έχουν χρησιμοποιηθεί οι σωστές συναρτήσεις. Εάν υπάρχουν σφάλματα ή προειδοποιήσεις, το IDE ενημερώνει για αυτά, και ο χρήστης δεν θα μπορέσει να προχωρήσει στο επόμενο βήμα δηλαδή το "Upload", μέχρι να διορθωθούν αυτά τα προβλήματα.
- **Upload (Ανέβασμα):** πατώντας το κουμπί "Upload" αν το πρόγραμμά περνάει την διαδικασία επαλήθευσης χωρίς προβλήματα, το Arduino IDE θα μεταφορτώσει τον μεταγλωττισμένο κώδικα στο Arduino board. Αυτό είναι το στάδιο που το πρόγραμμα που γράφτηκε θα εκτελεστεί στη συγκεκριμένη πλακέτα.

Είναι σημαντικό να έχει επιλεγθεί το σωστό μοντέλο του Arduino πριν πατηθεί το "Upload", καθώς αν έχουμε επιλεγθεί λανθασμένο μοντέλο, τότε ο μεταγλωττισμένος κώδικας δεν θα μπορέσει να φορτωθεί σωστά στην πλακέτα και η εφαρμογή δεν θα λειτουργήσει.

Συνοψίζοντας, το πεδίο "Upload/Verify" στο Arduino IDE επιτρέπει στον χρήστη να επαληθεύσει τον κώδικά για σφάλματα και συντακτικά προβλήματα πριν τον ανεβεί στο Arduino board.

- **SELECT BOARD & PORT:** Το πεδίο "Select Board" και "Port" στο περιβάλλον ανάπτυξης Arduino IDE χρησιμεύει για να γίνει η επιλογή του μοντέλο του Arduino board που ο χρήστης επιθυμεί να προγραμματίσει και τη θύρα σύνδεσης μεταξύ του υπολογιστή και του Arduino board.
- **Select Board (Επιλογή Πλακέτας):** Η επιλογή αυτή δίνει τη δυνατότητα να δηλωθεί στο Arduino IDE ποιο μοντέλο του Arduino χρησιμοποιείται για να φορτωθεί το πρόγραμμά. Κάθε μοντέλο έχει διαφορετικά χαρακτηριστικά, δυνατότητες και διαθέτει διαφορετικές αρχιτεκτονικές μικροελεγκτή (microcontroller). Η σωστή επιλογή της πλακέτας είναι σημαντική για την σωστή μεταγλώττιση και φόρτωση του κώδικα στην πλακέτα. Ύστερα από την επιλογή του μοντέλο της πλακέτας, το Arduino IDE θα χρησιμοποιήσει τις κατάλληλες βιβλιοθήκες και ρυθμίσεις για το συγκεκριμένο μοντέλο.
- **Port (Θύρα):** Η θύρα είναι ο τρόπος με τον οποίο ο υπολογιστής επικοινωνεί με το Arduino board. Η επιλογή της σωστής θύρας είναι απαραίτητη για να μπορέσει να μεταφορτωθεί ο κώδικας που έχει γράψει ο χρήστης. Όταν συνδεθεί το Arduino με τον υπολογιστή μέσω καλωδίου USB, ο υπολογιστής δημιουργεί μια εικονική σειριακή θύρα για την επικοινωνία με το Arduino board. Αυτή η θύρα εμφανίζεται στο πεδίο "Port" του Arduino IDE. Πρέπει να γίνει επιλογή της σωστής θύρας για την πλακέτα που συνδέθηκε, διαφορετικά δεν θα μπορέσει να ανεβεί το πρόγραμμά.

Συνοψίζοντας, τα πεδία "Select Board" και "Port" στο Arduino IDE δίνουν τη δυνατότητα να δηλωθεί το μοντέλο του Arduino που θα χρησιμοποιηθεί και να επιλεγεί η σωστή θύρα για την επικοινωνία με την πλακέτα.

- **SKETCHBOOK:** Το πεδίο "Sketchbook" στο περιβάλλον ανάπτυξης Arduino IDE χρησιμεύει για να καθοριστεί ο φάκελος όπου αποθηκεύονται τα σκίτσα (sketches) του Arduino που έχουν δημιουργηθεί ή αποθηκευτεί. Όταν δημιουργηθεί ένα νέο σκίτσο ή

ανοίξει ένα υπάρχον, το Arduino IDE προεπιλέγει τον φάκελο "Sketchbook" ως τον τρέχοντα φάκελο εργασίας. Αυτό σημαίνει ότι τα σκίτσα που αποθηκεύονται από το IDE αποθηκεύονται αυτόματα στον φάκελο "Sketchbook». Έχοντας τον φάκελο "Sketchbook" ορισμένο, μπορούν να οργανωθούν τα σκίτσα σε υπο-φακέλους, προσθέτοντας έτσι μια δομή φακέλων που θα βοηθήσει στην ευκολότερη διαχείριση των προγραμμάτων του χρήστη.

Συνοψίζοντας, το πεδίο "Sketchbook" στο Arduino IDE καθορίζει τον φάκελο όπου αποθηκεύονται τα σκίτσα του Arduino, δηλαδή τα προγράμματα που έχουν δημιουργηθεί και εκτελούνται στην πλακέτα.

- **BOARD MANAGER:** Το πεδίο "Board Manager" στο περιβάλλον ανάπτυξης Arduino IDE επιτρέπει την εγκατάσταση, τη διαχείριση και την ενημέρωση πρόσθετων πακέτων πλακετών (board packages) για διάφορα μοντέλα του Arduino.

Κάθε μοντέλο του Arduino έχει μια αντίστοιχη πλακέτα (board package) που πρέπει να είναι εγκατεστημένη στο περιβάλλον ανάπτυξης προκειμένου να μπορέσει να φορτωθεί ο κώδικας στην πλακέτα Arduino που χρησιμοποιείται. Η πλακέτα περιλαμβάνει τις βιβλιοθήκες που απαιτούνται για τον συγκεκριμένο τύπο πλακέτας.

Όταν εγκατασταθεί το Arduino IDE για πρώτη φορά, η πλειοψηφία των προκαθορισμένων πλακετών προ-εγκαθίστανται, συμπεριλαμβανομένων των πλακετών του Arduino Uno, Arduino Nano, Arduino Mega και άλλων βασικών μοντέλων. Ωστόσο, μπορούν να προστεθούν περισσότερες πλακέτες από το "Board Manager" ανάλογα με τις εκάστοτε ανάγκες.

Συνοψίζοντας, το πεδίο "Board Manager" στο Arduino IDE επιτρέπει την προσθήκη, την διαχείριση και ενημέρωση πλακετών για την υποστήριξη διάφορων μοντέλων του Arduino στο περιβάλλον ανάπτυξης.

- **LIBRARY MANAGER :** Το πεδίο "Library Manager" στο περιβάλλον ανάπτυξης Arduino IDE επιτρέπει την εγκατάσταση, τη διαχείριση και την ενημέρωση βιβλιοθηκών (libraries) για το Arduino. Οι βιβλιοθήκες είναι συλλογές κώδικα που παρέχουν πρόσθετες λειτουργίες και δυνατότητες που μπορούν να χρησιμοποιηθούν στα προγράμματά του Arduino. Αντί για διαμορφωθεί ο κώδικας από το μηδέν για κάθε λειτουργία που χρειάζεται, ο χρήστης επωφελείται από υπάρχουσες βιβλιοθήκες που έχουν δημιουργηθεί από την κοινότητα του Arduino και άλλους προγραμματιστές.

Για να χρησιμοποιηθεί το "Library Manager" ο χρήστης θα πρέπει να:

1. Κάνει κλικ στο μενού "Εργαλεία" (Tools) στο Arduino IDE.
2. Επιλέξει την επιλογή "Διαχείριση Βιβλιοθηκών" (Library Manager).

Αυτό θα ανοίξει το "Library Manager" και θα εμφανίσει μια λίστα με διάφορες διαθέσιμες βιβλιοθήκες που μπορούν να εγκατασταθούν. Η λίστα περιλαμβάνει τις δημοφιλέστερες βιβλιοθήκες που υποστηρίζονται από την κοινότητα του Arduino και παρέχουν λύσεις για διάφορες ανάγκες, όπως αισθητήρες, οθόνες, επικοινωνίες με εξωτερικές συσκευές, και πλήθος άλλων λειτουργιών. Συνοψίζοντας, το πεδίο "Library Manager" στο Arduino IDE επιτρέπει την εγκατάσταση και διαχείριση επιπλέον βιβλιοθηκών που παρέχουν επιπλέον λειτουργικότητα για το Arduino χωρίς να χρειάζεται να δημιουργηθεί ο κώδικας από το μηδέν.

- **DEBUGGER:** Αν και το περιβάλλον ανάπτυξης Arduino IDE προσφέρει πολλές λειτουργίες προγραμματισμού, το πεδίο "Debugger" δεν ανήκει στις κύριες λειτουργίες του IDE. Συγκεκριμένα, το Arduino IDE δεν διαθέτει ενσωματωμένο debugger, τουλάχιστον όχι με

την κύρια έννοια του όρου που συναντώνται σε άλλα περιβάλλοντα ανάπτυξης που χρησιμοποιούν πλήρεις σειριακές διαπαφές με debuggers και breakpoints.

Ο λόγος είναι ότι οι περισσότεροι μικροελεγκτές Arduino υποστηρίζουν περιορισμένες δυνατότητες debugging. Οι μικροελεγκτές Arduino βασίζονται σε μικρές και απλές αρχιτεκτονικές που δεν υποστηρίζουν πλήρη debugging. Ωστόσο, υπάρχουν ορισμένοι τρόποι να αντιμετωπιστεί το debugging στον κώδικα που έχει υλοποιηθεί:

- i. **Serial.print():** Μια από τις πιο συνηθισμένες μεθόδους debugging στο Arduino είναι η χρήση της Serial.print() ή Serial.println() για την εμφάνιση δεδομένων, τιμών μεταβλητών και μηνυμάτων στην σειριακή κονσόλα του Arduino IDE. Τα μηνύματα αυτά εμφανίζονται στο Serial Monitor.
- ii. **Χρήση προσομοιωτή:** Ορισμένα περιβάλλοντα ανάπτυξης παρέχουν προσομοιωτές (simulators) που επιτρέπουν την εκτέλεση του κώδικα σε ένα εικονικό περιβάλλον πριν φορτωθεί στην πραγματική πλακέτα Arduino.
- iii. **Εξωτερικός Debugger:** Σε ορισμένες περιπτώσεις, μερικοί μικροελεγκτές Arduino υποστηρίζουν εξωτερικούς debugger που συνδέονται μέσω προγραμματιζόμενων συγκεκριμένων πορτών.

Συνοψίζοντας, το πεδίο "Debugger" στο Arduino IDE δεν αντιστοιχεί σε πλήρες debugging με δυνατότητες breakpoints. Ωστόσο, μπορούν να χρησιμοποιηθούν οι προαναφερθείσες μέθοδοι για την βοήθεια στην εύρεση και επίλυση προβλημάτων στον κώδικά.

- **SEARCH:** Το πεδίο "Search" (Αναζήτηση) στο περιβάλλον ανάπτυξης Arduino IDE επιτρέπει την αναζήτηση διαθέσιμων βιβλιοθηκών (libraries) και παραδειγμάτων (examples) που είναι διαθέσιμα για το Arduino. Ο χρήστης ανακαλύπτει και εξερευνά μια μεγάλη ποικιλία από βιβλιοθήκες που παρέχονται από την κοινότητα του Arduino. Αυτές οι βιβλιοθήκες περιλαμβάνουν διάφορες λειτουργίες και δυνατότητες που μπορούν να χρησιμοποιηθούν κατά την διαμόρφωση του κώδικα.

Για να αξιοποιηθεί το πεδίο αναζήτησης:

- i. Κάνοντας κλικ στο πεδίο "Search" (Αναζήτηση) που βρίσκεται στο πάνω δεξιά μέρος του παραθύρου του Arduino IDE.
- ii. Πληκτρολογείτε το όνομα ή μια λέξη-κλειδί που σχετίζεται με τον τύπο βιβλιοθήκης που αναζητείται ή τον τύπο του αισθητήρα/κομμάτι που είναι επιθυμητό να χρησιμοποιηθεί. Το Arduino IDE θα εμφανίσει τα αποτελέσματα αναζήτησης από την OnLine βάση δεδομένων του Arduino που περιλαμβάνει τις διαθέσιμες βιβλιοθήκες και παραδείγματα.

Η αναζήτηση είναι ένα πολύ χρήσιμο εργαλείο για τους προγραμματιστές Arduino, καθώς βοηθά στην γρήγορη και εύκολη εύρεση κατάλληλων που χρειάζονται για τις νέες δημιουργίες και την επέκταση των δυνατοτήτων του Arduino.

- **OPEN SERIAL MONITOR:** Το πεδίο "Serial Monitor" (Σειριακός Παρατηρητής) στο περιβάλλον ανάπτυξης Arduino IDE είναι ένα εργαλείο που επιτρέπει την επικοινωνία με την πλακέτα Arduino μέσω της σειριακής θύρας (Serial Port) του υπολογιστή. Οι πλακέτες Arduino συνήθως έχουν μια σειριακή θύρα που επιτρέπει την επικοινωνία με τον υπολογιστή. Χρησιμοποιώντας το Serial Monitor, μπορούν να σταλούν και να ληφθούν δεδομένα μέσω αυτής της σειριακής θύρας. Τα βασικά χαρακτηριστικά και λειτουργίες του Serial Monitor περιλαμβάνουν:

- i. Αποστολή δεδομένων: Δίνεται η δυνατότητα να σταλθούν δεδομένα από το Serial Monitor στο Arduino. Αυτό μπορεί να είναι χρήσιμο για την δοκιμή και τον έλεγχο των λειτουργιών του προγράμματος που εκτελείται στο Arduino.
- ii. Λήψη δεδομένων: Υπάρχει η δυνατότητα να ληφθούν δεδομένα που αποστέλλονται από το Arduino μέσω της σειριακής θύρας. Αυτό είναι χρήσιμο για την παρατήρηση και την ανάλυση των δεδομένων που παράγει το Arduino κατά την εκτέλεση του προγράμματος.
- iii. Ρύθμιση του baud rate: Μπορείτε να ρυθμίσουμε το baud rate (ταχύτητα επικοινωνίας) που χρησιμοποιείται για τη σειριακή επικοινωνία μεταξύ του υπολογιστή και του Arduino. Αυτό είναι σημαντικό για να εξασφαλιστεί ότι το Arduino και το Serial Monitor λειτουργούν στον ίδιο ρυθμό επικοινωνίας.

Το Serial Monitor είναι ένα πολύ χρήσιμο εργαλείο για την ανάπτυξη και τον εντοπισμό προβλημάτων στα προγράμματά του Arduino. Βλέποντας και αλληλοεπιδρώντας με τα δεδομένα που εκπέμπονται από το Arduino, καθίσταται η ανάπτυξη και η επίλυση προβλημάτων πολύ πιο εύκολη.

- OPEN SERIAL PLOTTER: Το πεδίο "Serial Plotter" (Σειριακός Διάγραμμα Επικοινωνίας) στο περιβάλλον ανάπτυξης Arduino IDE είναι ένα εργαλείο που επιτρέπει την οπτικοποίηση και την απεικόνιση δεδομένων που αποστέλλονται από το Arduino στον υπολογιστή μέσω της σειριακής θύρας (Serial Port).

Αντί να εμφανίζονται τα δεδομένα ως κείμενο, όπως γίνεται στο Serial Monitor, το Serial Plotter εμφανίζει τα δεδομένα ως γραφικά. Αυτό είναι χρήσιμο για να παρατηρούν και να αναλυθούν οι μεταβολές των τιμών των αισθητήρων, μεταβλητών ή άλλων δεδομένων που στέλνει το Arduino στον υπολογιστή σε πραγματικό χρόνο.

Τα βασικά χαρακτηριστικά και λειτουργίες του Serial Plotter περιλαμβάνουν:

- Οπτικοποίηση Δεδομένων: Το Serial Plotter αναλύει τα δεδομένα που λαμβάνονται από το Arduino και εμφανίζει τις τιμές ως γραφικά. Δίνεται η δυνατότητα επιλογής για το ποια δεδομένα παρακολουθούνται και πώς θα πρέπει να εμφανίζονται (όπως γραφικά γραμμών ή περιοχών).
- Προσαρμογή Άξονα: δυνατότητα προσαρμογής της κλίμακας των αξόνων X και Y, προκειμένου να διακριθούν οι μεταβολές των δεδομένων με μεγαλύτερη ευκρίνεια.
- Παρακολούθηση Πραγματικού Χρόνου: Τα δεδομένα εμφανίζονται στο Serial Plotter κατά τη λήψη τους από το Arduino, επιτρέποντάς την παρακολούθηση των μεταβολών σε πραγματικό χρόνο.
- Χρησιμοποιώντας το Serial Plotter, ο χρήστης παρακολουθεί τις μετρήσεις αισθητήρων, την κίνηση ή άλλες μεταβλητές του Arduino στη μορφή γραφικών, πράγμα που καθιστά την ανάλυση και την επίβλεψη των δεδομένων πιο ευχάριστη και ευανάγνωστη.

2.4 Προγραμματισμός Arduino Uno

Για να δημιουργηθεί το προγραμματιστικό περιβάλλον το οποίο θα χρησιμοποιηθεί με σκοπό να καταγραφούν οι μετρήσεις που είναι απαραίτητες για την δημιουργία των διαγραμμάτων Bode

θα πρέπει να αξιοποιηθούν μερικές βασικές εντολές στον κώδικα του Arduino IDE. Μερικές από αυτές τις εντολές είναι οι εξής :

- `#include <Wire.h>` : Η συγκεκριμένη εντολή χρησιμοποιείται στο πρόγραμμα Arduino για να συμπεριλάβει τη βιβλιοθήκη Wire.h στον κώδικά. Αυτή η βιβλιοθήκη είναι συνήθως χρήσιμη για την επικοινωνία με διάφορες συσκευές μέσω του πρωτοκόλλου I2C (Inter-Integrated Circuit), όπως αισθητήρες, οθόνες LCD, EEPROM και πολλές άλλες. Ουσιαστικά χρησιμοποιούμε την Wire.h για να μπορέσει το Arduino να επικοινωνήσει με τις συσκευές που χρησιμοποιούν το I2C πρωτόκολλο άρα για να συνδεθεί με την οθόνη LCD [9].
- `#include <LiquidCrystal_I2C.h>`: Το I2C πρωτόκολλο είναι ένας τρόπος επικοινωνίας μεταξύ του Arduino και της οθόνης LCD, και η βιβλιοθήκη "LiquidCrystal_I2C" παρέχει έναν βολικό τρόπο για τον έλεγχο αυτής της οθόνης μέσω του πρωτοκόλλου I2C. Με την συμπερίληψη της βιβλιοθήκης με τη γραμμή `#include <LiquidCrystal_I2C.h>`, μπορούν να χρησιμοποιηθούν οι συναρτήσεις και οι μέθοδοι που παρέχονται από αυτήν τη βιβλιοθήκη για τον έλεγχο της οθόνης LCD μέσω του πρωτοκόλλου I2C, δηλαδή την εμφάνιση κειμένου ή χαρακτήρων στην οθόνη [10].
- `#define` : Στο πρόγραμμα Arduino, η εντολή `#define` χρησιμοποιείται για τον καθορισμό μιας συμβολικής σταθεράς ή μιας μακροεντολής. Αυτή η δήλωση επιτρέπει να δηλωθεί ένα όνομα (σύμβολο) που αντιστοιχεί σε έναν αριθμό ή μια συμβολική τιμή, και στη συνέχεια μπορεί να χρησιμοποιηθεί αυτό το όνομα στον κώδικά αντί για την πραγματική τιμή. Στην προκειμένη περίπτωση έχει χρησιμοποιηθεί αυτή η εντολή για να δηλωθούν οι αναλογικές και ψηφιακές είσοδοι στην πλακέτα [11].
- `int` : Στο Arduino, η εντολή `int` χρησιμοποιείται για τον καθορισμό μιας μεταβλητής τύπου `integer` (ακέραιος). Ο τύπος `int` αντιστοιχεί σε ακέραιες τιμές, δηλαδή αριθμούς χωρίς δεκαδικά ψηφία. Αυτή η μεταβλητή μπορεί να χρησιμοποιηθεί για την αποθήκευση ακεραίων αριθμών στο πρόγραμμα.
- `float` : Στο Arduino, η εντολή `float` χρησιμοποιείται για τον καθορισμό μιας μεταβλητής τύπου `float`. Ο τύπος `float` αντιπροσωπεύει τις κινητές υποδιαστάσεις (`floating-point numbers`) και επιτρέπει την αποθήκευση αριθμών που έχουν δεκαδικά ψηφία. Δηλαδή, μπορεί να χρησιμοποιηθεί ο τύπος `float` για την αποθήκευση αριθμών που έχουν δεκαδικά.
- `void setup()` : Η εντολή `void setup()` στο Arduino χρησιμοποιείται για τον καθορισμό των αρχικών ρυθμίσεων και την εκτέλεση ενεργειών που πρέπει να γίνουν μόλις εκκινήσει το πρόγραμμά. Συνήθως, η `setup()` εκτελείται μόνο μια φορά κατά την εκκίνηση του Arduino ή μετά από επανεκκίνηση του μικροελεγκτή. Στην `setup()`, μπορούν να δηλωθούν αρχικές ρυθμίσεις για τους ακροδέκτες (`pins`), να αρχικοποιηθούν μεταβλητές, να οριστεί ο τρόπος λειτουργίας των αισθητήρων ή των επιπλέον περιφερειακών συσκευών που συνδέονται με το Arduino, να οριστεί ο ρυθμός επανάληψης του κύριου βρόχου (`loop()`) και να εκτελεστούν άλλες αρχικές ενέργειες που απαιτούνται για τη σωστή λειτουργία του προγράμματός [12].
- `Serial.begin()`: Η εντολή `Serial.begin()` στο Arduino χρησιμοποιείται για την αρχικοποίηση και την ρύθμιση της σειριακής (`Serial`) επικοινωνίας μεταξύ του Arduino και ενός υπολογιστή ή άλλης συσκευής μέσω της θύρας UART (`Universal Asynchronous Receiver/Transmitter`). Αυτή η επικοινωνία μπορεί να χρησιμοποιηθεί για την αποστολή και τη λήψη δεδομένων ανάμεσα στο Arduino και την εξωτερική συσκευή [13].

Η συνάρτηση `Serial.begin()` δέχεται έναν όρο, ο οποίος αντιπροσωπεύει τον ρυθμό μετάδοσης (baud rate) της σειριακής επικοινωνίας. Ο ρυθμός μετάδοσης καθορίζει πόσα bits ανταλλάσσονται ανά δευτερόλεπτο ανάμεσα στο Arduino και τη συσκευή που είναι συνδεδεμένη μέσω της σειριακής θύρας UART. Ο σωστός ρυθμός μετάδοσης πρέπει να είναι ο ίδιος και στις δύο συσκευές που επικοινωνούν.

- `pinMode()`: Η εντολή `pinMode()` στο Arduino χρησιμοποιείται για τον καθορισμό της λειτουργίας ενός ακροδέκτη (pin) σε ένα από τα δύο καθεστώτα: είτε ως είσοδο (input) είτε ως έξοδο (output). Αυτό είναι σημαντικό για τον έλεγχο του τρόπου με τον οποίο το Arduino αλληλοεπιδρά με τον εξωτερικό κόσμο, όπως αισθητήρες, ενεργοποιητές, LED [14].
- `void loop()`: Η εντολή `void loop()` στο Arduino δημιουργεί τη βασική λούπα (loop) του προγράμματος Arduino. Αυτή η λούπα εκτελείται συνεχώς μετά την αρχικοποίηση του προγράμματος στην `setup()` και επαναλαμβάνεται συνεχώς μέχρι να διακοπεί η τροφοδοσία του Arduino ή να το διακοπεί χειροκίνητα. Στην `void loop()`, το Arduino εκτελεί τον κύριο κώδικά του προγράμματος. Αυτό σημαίνει ότι μπορούν να τοποθετηθούν εδώ οι εντολές και οι εργασίες που πρέπει το Arduino να εκτελεί επανειλημμένα. Για παράδειγμα, την ανάγνωση αισθητήρων, τον έλεγχο και την αλλαγή της κατάστασης των ακροδεκτών (pins), να αναβοσβήνουν LEDs, να αποστέλλονται δεδομένα σε μια σειριακή θύρα [15].
- `If` : εντολή `if` στο Arduino (και στη γλώσσα προγραμματισμού C/C++) χρησιμοποιείται για τον έλεγχο μιας συνθήκης (condition) και την εκτέλεση ενός κώδικα μόνο εάν η συνθήκη αυτή είναι αληθής (true). Η `if` είναι ένα από τα βασικά δομικά στοιχεία της προγραμματισμού, και χρησιμοποιείται ευρέως για τον έλεγχο συνθηκών και την διακλάδωση της εκτέλεσης του προγράμματος ανάλογα με τις συνθήκες που ισχύουν [16].
- `pulseIn()` : Η εντολή `pulseIn()` στο Arduino χρησιμοποιείται για να μετρήσει τη διάρκεια ενός ψηφιακού παλμού (pulse) σε έναν ακροδέκτη (pin). Συνήθως χρησιμοποιείται για την μέτρηση παλμών όπως όσους παράγονται από αισθητήρες, ενεργοποιητές ή άλλες συσκευές. Η `pulseIn()` επιστρέφει τη διάρκεια του παλμού σε μικροδευτερόλεπτα (μικροδευτερόλεπτο είναι το ένα εκατομμύριο του δευτερολέπτου). Αν δεν ανιχνευθεί παλμός κατά τη διάρκεια του χρόνου που καθορίζετε με το `timeout`, τότε η `pulseIn()` επιστρέφει μηδενικό [17].
- `lcd.setCursor()` : Η εντολή `lcd.setCursor()` χρησιμοποιείται στο Arduino με τη βιβλιοθήκη `LiquidCrystal` για την οθόνη LCD και χρησιμοποιείται για να ορισθεί η θέση του κέρσορα (cursor) στην οθόνη. Ο κέρσορας είναι ένα σημείο στην οθόνη όπου τα δεδομένα θα εμφανιστούν ή θα γραφούν όταν χρησιμοποιείται η `lcd.print()` ή άλλες σχετικές εντολές [18].
- `lcd.print()` : Η εντολή `lcd.print()` στο Arduino χρησιμοποιείται με τη βιβλιοθήκη `LiquidCrystal` για τις οθόνες LCD και χρησιμοποιείται για να εμφανίσει κείμενο, αριθμούς, χαρακτήρες ή οποιαδήποτε άλλα δεδομένα πάνω στην οθόνη LCD στη θέση που έχει καθοριστεί από τον κέρσορα (cursor). Αυτή η εντολή είναι χρήσιμη για την εμφάνιση πληροφοριών, μηνυμάτων ή αποτελεσμάτων αισθητήρων στην οθόνη LCD [19].
- `else`: Η εντολή `else` στο Arduino (και στη γλώσσα προγραμματισμού C/C++) χρησιμοποιείται για να δημιουργήσει έναν "εναλλακτικό" κώδικα που θα εκτελεστεί όταν η συνθήκη που αντιστοιχεί στην `if` προηγούμενη εντολή είναι ψευδής (false). Αυτό επιτρέπει τη διακλάδωση του κώδικα, δηλαδή την εκτέλεση διαφορετικών τμημάτων κώδικα ανάλογα με την συνθήκη [20].

- `analogRead()`: Η εντολή `analogRead()` στο Arduino χρησιμοποιείται για να διαβάσει την αναλογική τάση από έναν ακροδέκτη (pin) που είναι συνδεδεμένος με έναν αισθητήρα ή μια άλλη αναλογική πηγή. Αυτή η τάση μετατρέπεται σε ψηφιακή τιμή (ακέραιος αριθμός) που αντιστοιχεί στην τιμή της αναλογικής τάσης σε ένα εύρος από 0 έως 1023 (στην περίπτωση των Arduino UNO) ή από 0 έως 4095 (στην περίπτωση των Arduino Mega και άλλων μοντέλων με υψηλότερη ανάλυση ADC) [21].
- `Serial.println()` : Η εντολή `Serial.println()` στο Arduino χρησιμοποιείται για την αποστολή δεδομένων (κειμένου, αριθμών κ.λπ.) μέσω της σειριακής θύρας (Serial port) προς έναν υπολογιστή ή άλλη συσκευή που είναι συνδεδεμένη με το Arduino μέσω USB, ή άλλων μεθόδων σειριακής επικοινωνίας. Η `Serial.println()` αποστέλλει τα δεδομένα που παρέχονται ως όρισμα στη σειριακή θύρα και προσθέτει έναν χαρακτήρα νέας γραμμής (\n) στο τέλος του μηνύματος. Αυτό σημαίνει ότι κάθε φορά που καλείτε η `Serial.println()`, τα δεδομένα θα εμφανίζονται σε νέα γραμμή στο παράθυρο του περιβάλλοντος ανάπτυξης Arduino (IDE) ή στο πρόγραμμα που χρησιμοποιείτε για την παρακολούθηση της σειριακής επικοινωνίας (όπως το Serial Monitor στο Arduino IDE). Αυτή η εντολή είναι χρήσιμη για τον εντοπισμό σφαλμάτων, την αποσφαλμάτωση του προγράμματός και την εμφάνιση πληροφοριών ή αποτελεσμάτων από το Arduino στον υπολογιστή [22].
- `Serial.print()` : Η `Serial.print()` αποστέλλει τα δεδομένα που παρέχονται ως όρισμα στη σειριακή θύρα χωρίς να προσθέτει έναν χαρακτήρα νέας γραμμής (\n) στο τέλος του μηνύματος. Αυτό σημαίνει ότι τα δεδομένα θα εμφανίζονται δίπλα δίπλα χωρίς να πηγαίνουν σε νέα γραμμή [23].
- `delay()` : Η εντολή `delay()` στο Arduino χρησιμοποιείται για να μια προσωρινή καθυστέρηση (pause) στην εκτέλεση του προγράμματος. Κατά τη διάρκεια της καθυστέρησης, το Arduino δεν εκτελεί κώδικα και παραμένει αδρανές για τον χρονικό διάστημα που καθορίζεται. Αυτό είναι χρήσιμο για τη δημιουργία χρονικών διακοπών ή καθυστερήσεων μεταξύ των εντολών. Ο αριθμός των χιλιοστών του δευτερολέπτου που πρέπει να γίνει αυτή η καθυστέρηση. Δηλαδή, το πόσο χρόνο απαιτείται το Arduino να παραμείνει αδρανές. Για παράδειγμα, `delay(1000)` θα προκαλέσει μια καθυστέρηση 1000 χιλιοστών του δευτερολέπτου ή 1 δευτερολέπτου [24].

Κεφάλαιο 3: Ανάλυση στο πεδίο της συχνότητας

Η γραφική παράσταση Bode είναι μια γραφική αναπαράσταση που χρησιμοποιείται στην ηλεκτρολογική μηχανική και την ανάλυση συστημάτων ελέγχου για την εμφάνιση της απόκρισης συχνότητας ενός συστήματος. Δείχνει πώς το μέτρο και η φάση της εξόδου (ή της συνάρτησης μεταφοράς) ενός συστήματος ποικίλλει ανάλογα με τη συχνότητα. Τα διαγράμματα Bode είναι ιδιαίτερα χρήσιμα για την κατανόηση της συμπεριφοράς των φίλτρων, των ενισχυτών και άλλων γραμμικών συστημάτων.

3.1 Χρήσεις διαγράμματος Bode

Τα διαγράμματα Bode είναι ένα θεμελιώδες εργαλείο στη μηχανική, ειδικά σε τομείς όπως η ηλεκτρική μηχανική, τα συστήματα ελέγχου και η επεξεργασία σήματος. Παρέχουν πολύτιμες πληροφορίες για την απόκριση συχνότητας των συστημάτων και χρησιμοποιούνται για διάφορους σκοπούς:

- Σχεδιασμός και ανάλυση φίλτρου: Τα διαγράμματα Bode χρησιμοποιούνται συνήθως για το σχεδιασμό και την ανάλυση φίλτρων, συμπεριλαμβανομένων των υψηλοπερατών, χαμηλοπερατών, φίλτρων διέλευσης και φίλτρων απόρριψης ζώνης. Οι μηχανικοί μπορούν να οπτικοποιήσουν πώς συμπεριφέρεται το φίλτρο σε διαφορετικές συχνότητες, βοηθώντας τους να επιλέξουν τα κατάλληλα εξαρτήματα και παραμέτρους για να επιτύχουν τις επιθυμητές συχνότητες αποκοπής.
- Ανάλυση ενισχυτή και ταλαντωτή: Τα διαγράμματα Bode χρησιμοποιούνται για την αξιολόγηση της συμπεριφοράς των ενισχυτών και των ταλαντωτών. Οι ενισχυτές δείχνουν πώς το κέρδος αλλάζει με τη συχνότητα, βοηθώντας στη διασφάλιση σταθερής και επιθυμητής ενίσχυσης. Στους ταλαντωτές, τα διαγράμματα Bode βοηθούν στην ανάλυση των μετατοπίσεων φάσης και των συνθηκών σταθερότητας.
- Σχεδιασμός συστήματος ελέγχου: Οι μηχανικοί χρησιμοποιούν διαγράμματα Bode για να σχεδιάσουν και να αναλύσουν συστήματα ελέγχου, διασφαλίζοντας σταθερότητα, απόδοση και επιθυμητά χαρακτηριστικά απόκρισης. Τα διαγράμματα Bode βοηθούν στην αξιολόγηση των περιθωρίων σταθερότητας του συστήματος, των περιθωρίων φάσης και κέρδους και της συνολικής συμπεριφοράς του συστήματος.
- Επεξεργασία σήματος: Τα διαγράμματα Bode χρησιμοποιούνται για την κατανόηση της συμπεριφοράς των συστημάτων στον τομέα της συχνότητας, βοηθώντας τους μηχανικούς να σχεδιάσουν φίλτρα και επεξεργαστές ήχου, εικόνα και άλλους τύπους εφαρμογών επεξεργασίας σήματος.
- Ανάλυση κυκλώματος και αντιμετώπιση προβλημάτων: Τα διαγράμματα Bode είναι χρήσιμα για την ανάλυση της απόκρισης συχνότητας των κυκλωμάτων. Μπορούν να αναγνωρίσουν συντονισμούς, περιορισμούς εύρους ζώνης και πιθανές αστάθειες σε πολύπλοκα κυκλώματα.
- Συστήματα επικοινωνίας: Τα διαγράμματα Bode χρησιμοποιούνται για το σχεδιασμό και την ανάλυση συστημάτων επικοινωνίας, διασφαλίζοντας ότι τα μεταδιδόμενα σήματα διατηρούν τα επιθυμητά χαρακτηριστικά σε ένα εύρος συχνοτήτων.
- Σχεδίαση εξοπλισμού ήχου: Στη μηχανική ήχου, τα σχέδια Bode βοηθούν τους σχεδιαστές να βελτιστοποιήσουν τα μεγάφωνα, τους ενισχυτές και άλλο εξοπλισμό ήχου για τις επιθυμητές αποκρίσεις συχνότητας και τα χαρακτηριστικά φάσης.
- Ανάλυση θορύβου: Οι μηχανικοί μπορούν να χρησιμοποιήσουν διαγράμματα Bode για να αξιολογήσουν πώς ο θόρυβος επηρεάζει την απόκριση συχνότητας των συστημάτων, επιτρέποντας την επίλυση μείωσης του θορύβου.
- Εκπαίδευση και Έρευνα: Τα διαγράμματα Bode χρησιμοποιούνται ευρέως στην εκπαίδευση για να διδάξουν τους μαθητές σχετικά με την ανάλυση τομέα συχνότητας, τη συμπεριφορά του συστήματος και τη θεωρία ελέγχου. Οι ερευνητές τα χρησιμοποιούν για να διερευνήσουν τις ιδιότητες νέων συστημάτων και να αναπτύξουν καινοτόμες λύσεις.

Σε όλες αυτές τις εφαρμογές, τα διαγράμματα Bode παρέχουν έναν σαφή και διαισθητικό τρόπο κατανόησης του τρόπου με τον οποίο τα συστήματα ανταποκρίνονται σε διαφορετικές συχνότητες, βοηθώντας τους μηχανικούς να λαμβάνουν τεκμηριωμένες αποφάσεις κατά τη διάρκεια των διαδικασιών σχεδιασμού, ανάλυσης και αντιμετώπισης προβλημάτων.

3.2 Θεωρητική ανάλυση στο πεδίο της συχνότητας

Για να σχεδιαστεί το διάγραμμα Bode ενός συστήματος με συνάρτηση μεταφοράς ανοικτού βρόγχου $G(s)H(s)$ απαιτείται η εύρεση της απόκρισης συχνότητας του συστήματος. Η απόκριση συχνότητας μπορεί να παρασταθεί σαν συνδυασμός του μέτρου $|G(j\omega)H(j\omega)|$ και της φάσης $\varphi(\omega)$, όπου $|G(j\omega)H(j\omega)| = \sqrt{Re(\omega)^2 + X(\omega)^2}$ και $\varphi(\omega) = \tan^{-1}\left(\frac{X(\omega)}{R(\omega)}\right)$. Για να υπολογιστεί το διάγραμμα Bode του συστήματος, θέτουμε $s=j\omega$. Στη συνέχεια υπολογίζονται οι τιμές $20\log|G(j\omega)H(j\omega)|$ και τοποθετούνται σε ένα διάγραμμα, και σε ένα άλλο οι υπολογίζονται οι τιμές της φάσης $\varphi(\omega)$ και τοποθετούνται σε ένα δεύτερο διάγραμμα για τις διάφορες τιμές του ω .

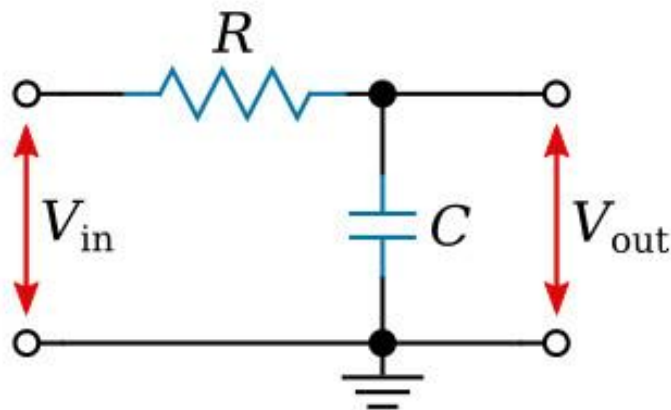
Το διάγραμμα Bode είναι ένα πολύτιμο εργαλείο για την κατανόηση και το σχεδιασμό φίλτρων και άλλων γραμμικών συστημάτων. Μας βοηθούν να κατανοήσουμε γρήγορα τη συμπεριφορά του συστήματος και την απόκρισή του σε διάφορες κλίμακες συχνότητων.

3.3 Ανάλυση βαθυπερατού φίλτρο στο πεδίο της συχνότητας

Το βαθυπερατό φίλτρο, γνωστό και ως φίλτρο χαμηλών περασμάτων (Low-pass filter), είναι ένα είδος φίλτρου που επιτρέπει τη διέλευση χαμηλών συχνοτήτων στο σήμα εξόδου, ενώ αποκόπτει ή αποφράζει τις υψηλές συχνότητες. Ουσιαστικά, αφήνει να περάσουν τα σήματα με συχνότητες κάτω από ένα καθορισμένο όριο και μειώνει ή εξαλείφει τα υψηλότερα σήματα.

Τα βαθυπερατά φίλτρα χρησιμοποιούνται συχνά στην επεξεργασία σήματος και τον ηλεκτρονικό σχεδιασμό για πολλούς σκοπούς, όπως η απομόνωση θορύβου και η μείωση των υψηλών συχνοτήτων.

Πρακτικά, ένα βαθυπερατό φίλτρο μπορεί να υλοποιηθεί με διάφορους τρόπους, όπως πολλαπλασιαστές, πυκνωτές και αντιστάσεις συνδεδεμένες σε σειρά, ψηφιακούς αλγόριθμους και άλλες τεχνικές, ανάλογα με τον τύπο του φίλτρου και τις απαιτήσεις της εφαρμογής. Στην παρούσα πτυχιακή χρησιμοποιήθηκε ένα κύκλωμα RC σε σειρά (εικόνα 5).



Εικόνα 5. Κύκλωμα βαθυπερατού φίλτρου [29].

Για τον υπολογισμό της συνάρτησης μεταφοράς του κυκλώματος ακολουθούμε τα παρακάτω βήματα.

Ισχύει ότι : $V(t) = \frac{1}{C} \int I(t)dt$. Χρησιμοποιώντας το 2ο κανόνα του Kirchhoff στο κύκλωμα μας προκύπτει :

$$V_{in}(t) = V_R(t) + V_{out}(t) \Leftrightarrow V_{in}(t) = I(t) * R + V_{out}(t)$$

$$\left. \begin{aligned} V_{in}(t) &= I(t) * R + V_{out}(t) \\ V_{out}(t) &= \frac{1}{C} \int I(t) dt \end{aligned} \right\} \xrightarrow{\text{Laplace}} \left. \begin{aligned} L\{V_{in}(t)\} &= L\{I(t) * R\} + L\{V_{out}(t)\} \\ L\{V_{out}(t)\} &= L\left\{\frac{1}{C} \int I(t) dt\right\} \end{aligned} \right\} \Rightarrow$$

$$\left. \begin{aligned} V_{in}(s) &= I(s) * R + U_{out}(s) \\ V_0(s) &= \frac{1}{C} \frac{I(s)}{s} \end{aligned} \right\} \Rightarrow \left. \begin{aligned} V_{in}(s) &= I(s) * R + U_{out}(s) \\ I(s) &= sCV_{out}(s) \end{aligned} \right\} \Rightarrow$$

$$\left. \begin{aligned} V_{in}(s) &= sCV_{out}(s)R + V_{out}(s) \\ I(s) &= sCV_{out}(s) \end{aligned} \right\} \Rightarrow \left. \begin{aligned} V_{in}(s) &= (sCR + 1)V_{out}(s) \\ I(s) &= sCV_{out}(s) \end{aligned} \right\} \Rightarrow$$

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{sCR + 1} = G(s) \quad (1)$$

Για τον υπολογισμό του μέτρου και της φάσης της συνάρτησης μεταφοράς ακολουθούμε την παρακάτω μεθοδολογία:

$$\begin{aligned} G(s) &= \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{sCR + 1} = \frac{\frac{1}{sC}}{sRC + 1} = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} \xrightarrow{L^{-1}} \\ L^{-1}\{G(s)\} &= L^{-1}\left\{\frac{\frac{1}{sC}}{R + \frac{1}{sC}}\right\} \\ g(t) &= \frac{V_{out}(t)}{V_i i(t)} = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} = \frac{j}{j^2\omega C} = \frac{-j}{\omega C} = \frac{-j|X_c|}{R - j|X_c|} \\ g(t) &= \frac{V_{out}(t)}{V_i i(t)} = \frac{-j|X_c|}{R - j|X_c|} \end{aligned} \quad (2)$$

Η παραπάνω συνάρτηση είναι ένας μιγαδικός αριθμός. Σε έναν τέτοιο μιγαδικό καθορίζουμε το μέτρο και το όρισμα του. Για παράδειγμα, αν θεωρήσουμε $z = a + jb \Rightarrow |z| = \sqrt{a^2 + b^2}$ και $\varphi = \arctan\left(\frac{b}{a}\right)$. Επομένως έχουμε ότι:

$$\begin{aligned} g(t) &= \frac{u_0(t)}{u_i(t)} = \frac{-j|X_c|}{R - j|X_c|} = \frac{-j|X_c|(R + j|X_c|)}{(R - j|X_c|)(R + j|X_c|)} = \frac{-j|X_c|(R + j|X_c|)}{R^2 - j^2X_c^2} \\ &= \frac{|X_c|^2}{R^2 + X_c^2} - j \frac{|X_c|R}{R^2 + X_c^2} \end{aligned}$$

$$\begin{aligned} |g(t)| &= \left| \frac{V_{out}(t)}{V_{in}(t)} \right| = \sqrt{\left(\frac{|X_c|^2}{R^2 + X_c^2}\right)^2 + \left(-\frac{|X_c|R}{R^2 + X_c^2}\right)^2} = \sqrt{\frac{|X_c|^4}{(R^2 + X_c^2)^2} + \frac{|X_c|^2 R^2}{(R^2 + X_c^2)^2}} \\ &= \sqrt{\frac{(R^2 + |X_c|^2)|X_c|^2}{(R^2 + X_c^2)^2}} = \sqrt{\frac{|X_c|^2}{R^2 + X_c^2}} = \frac{|X_c|}{\sqrt{R^2 + X_c^2}} \end{aligned}$$

και

$$\varphi = \arctan\left(-\frac{\frac{|X_C|R}{R^2}}{\frac{|X_C|^2}{R^2 + X_C^2}}\right) = \arctan\left(-\frac{|X_C|R}{|X_C|^2}\right) = \arctan\left(-\frac{R}{|X_C|}\right)$$

$$\varphi = -\arctan(2\pi fRC)$$

με το $-$ να μας ορίζει ότι η έξοδος έχει υστέρηση σε σχέση με την είσοδο κατά γωνιά φ .

Άρα το μέτρο του κυκλώματος μπορεί να υπολογιστεί από την σχέση (3) και η φάση από την σχέση (4):

$$|g(t)| = \left|\frac{V_{out}(t)}{V_{in}(t)}\right| = \frac{|X_C|}{\sqrt{R^2 + X_C^2}} \quad (3)$$

$$\varphi = -\arctan(2\pi fRC) \quad (4)$$

Η θεωρητική συμπεριφορά του RC φίλτρου στις χαμηλές, υψηλές και στην οριακή συχνότητα είναι η εξής:

- Στις χαμηλές συχνότητες το $X_C = \frac{1}{2\pi fC}$ είναι πολύ μεγαλύτερο του R ($X_C \gg R$):

$$|g(t)| = \left|\frac{V_{out}(t)}{V_{in}(t)}\right| \cong \frac{|X_C|}{\sqrt{X_C^2}} = 1 \Rightarrow u_o(t) \cong u_i(t)$$

και

$$\varphi = \arctan\left(-\frac{R}{|X_C|}\right) \cong -\arctan(0) = 0$$

επομένως συμπεραίνουμε πως το σήμα εισόδου είναι σχεδόν ίσο με το σήμα εξόδου, άρα η είσοδος και η έξοδος είναι συμφασικές.

- Στις υψηλότερες συχνότητες το $X_C = \frac{1}{2\pi fC}$ είναι αρκετά μικρότερο του R ($X_C \ll R$):

$$|g(t)| = \left|\frac{u_o(t)}{u_i(t)}\right| \cong \frac{0}{\sqrt{R^2 + 0}} \Rightarrow u_o(t) \cong 0$$

και

$$\varphi = \arctan\left(-\frac{R}{|X_C|}\right) \cong -\arctan(\infty) = -90^\circ$$

Επομένως το σήμα εξόδου είναι μηδενικό, άρα η έξοδος με την είσοδο έχουν διαφορά φάσης -90° .

- Οριακή συχνότητα $X_C = R$:

$$|g(t)| = \left|\frac{V_{out}(t)}{V_{in}(t)}\right| = \frac{|X_C|}{\sqrt{R^2 + X_C^2}} = \frac{R}{\sqrt{R^2 + R^2}} = \frac{\sqrt{2}}{2} = 0.707 \Rightarrow$$

$$u_0(t) = 0.707U_i(t)$$

και

$$\varphi = \arctan\left(-\frac{R}{|X_C|}\right) = -\arctan\left(\frac{R}{R}\right) = -45^\circ,$$

επομένως το πλάτος του σήματος εξόδου είναι ίσο με το 70.7% του σήματος εισόδου και η είσοδος με την έξοδο έχουν διαφορά φάσης -45° .

Κεφάλαιο 4: Κύκλωμα διαγράμματος Bode από μετρήσεις

Σε αυτό το κεφάλαιο παρουσιάζονται τα κυκλώματα από τα οποία προέκυψαν οι μετρήσεις για την διαμόρφωση των διαγραμμάτων Bode. Επιπλέον αναδεικνύεται το κύκλωμα (βαθυπερατό φίλτρο) που επιλέχθηκε να μελετηθεί.

4.1 Υλοποίηση κυκλώματος μέτρησης μέτρου και φάσης

Για τις ανάγκες λήψης των μετρήσεων χρησιμοποιήθηκε ένα βαθυπερατό φίλτρο αποτελούμενο από έναν πυκνωτή 10nF και μια αντίσταση 10kΩ. Οι τιμές αυτές επιλέχθηκαν από τη στιγμή που αποφασίστηκε πως η μελέτη θα γίνει στο πεδίο της συχνότητας από 100 - 20.000 Hz. Καθόλη τη διάρκεια δοκιμών και συλλογής τιμών η κατασκευή ήταν συνδεδεμένη με τον εργαστηριακό παλμογράφο για την διαπίστωση τυχόν αποκλίσεων.

Περαιτέρω, μέσω του παλμογράφου γινόταν και ο έλεγχος τροφοδοσίας του κυκλώματος πως διατηρούνταν στα 5V pk-pk καθόσον παρατηρήθηκε πως ανεβαίνοντας στο φάσμα της συχνότητας υπήρχε πολύ μικρή πτώση τάσης. Η τιμή αυτή επιλέχθηκε από τη στιγμή που, όπως αναγράφεται αναλυτικότερα παρακάτω, είναι η μέγιστη τάση τροφοδοσίας που μπορεί να λειτουργήσει ο μικροελεγκτής. Όσον αφορά στην πλακέτα διαφοράς φάσης η τιμή 5V pk-pk είναι μέσα στο φάσμα τροφοδοσίας της αφού ο σταθεροποιητής τάσης κάνει αυτή τη ρύθμιση.

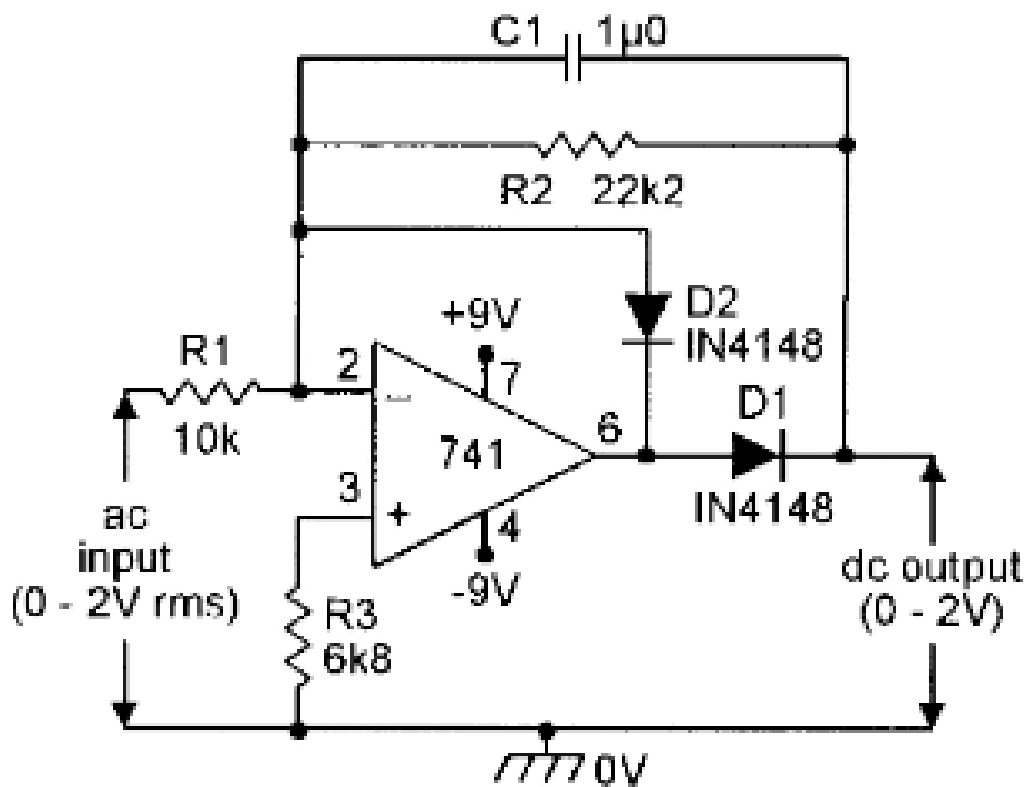
5V pk-pk Είναι σημαντικό να αναφερθεί, πως για να «συνεργαστούν» ο μικροελεγκτής με την πλακέτα που δημιουργεί τη διαφορά φάσης αλλά και την γεννήτρια συχνοτήτων, παρεμβλήθηκαν δυο αντιστάσεις στην έξοδο πηγής σημάτων (της συχνότητας και της διαφοράς φάσης), προς τον μικροελεγκτή που λαμβάνει σαν είσοδο τα σήματα αυτά. Αυτό συνέβη για να εξασφαλισθεί πως θα γίνει αντιστοίχιση της συνθέτης αντίστασης αλλά και να ελαχιστοποιηθούν οι απώλειες σήματος και οι ανακλάσεις του μικροελεγκτή, της γεννήτριας και της πλακέτας της διαφοράς φάσης.

4.1.1 Κύκλωμα μέτρησης τάσης

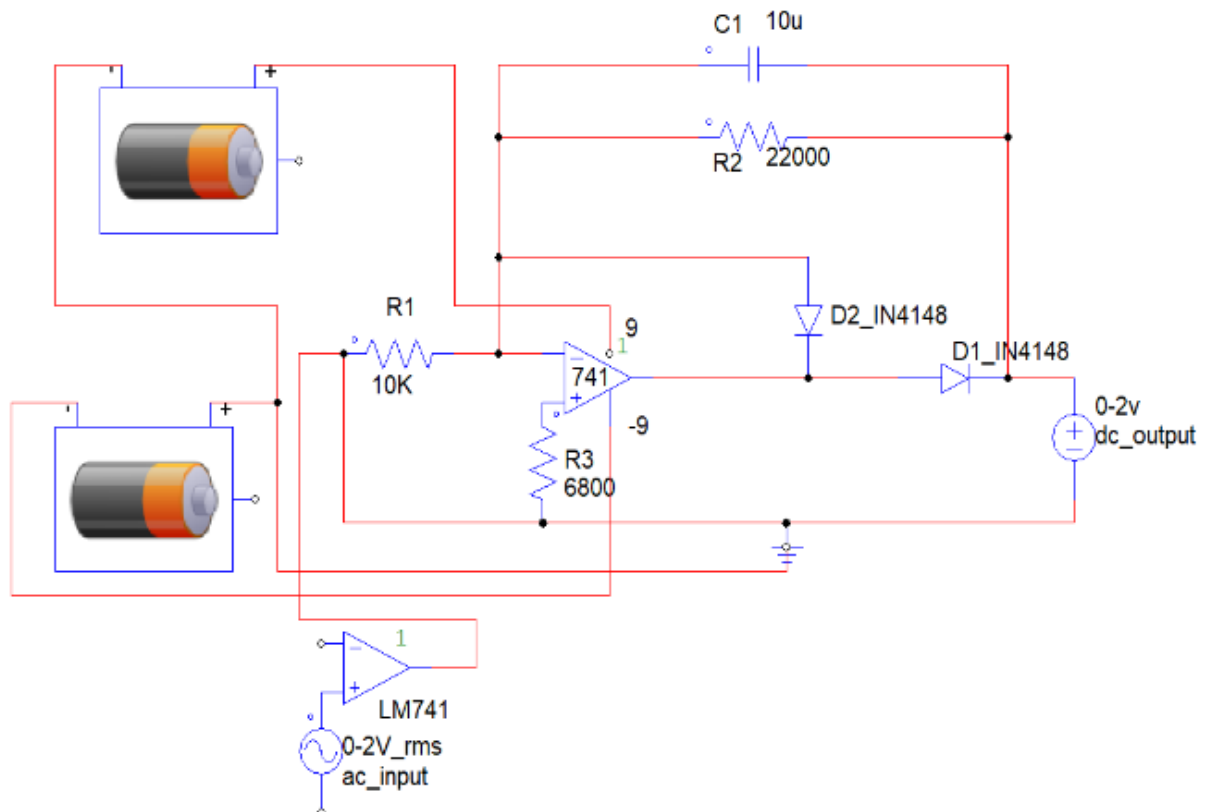
Για να μετρηθεί επιτυχώς, από την πλακέτα Arduino Uno, η τάση στην είσοδο και την έξοδο του φίλτρου που μελετώνται θα πρέπει να υπάρξει κάποιο ενδιάμεσο κύκλωμα το οποίο μετατρέπει την τάση V_{rms} σε τάση DC. Τα κυκλώματα μετατροπέα AC/DC, γνωστά και ως κυκλώματα ανόρθωσης, λειτουργούν μετατρέποντας την τάση εναλλασσόμενου ρεύματος (AC) σε τάση συνεχούς ρεύματος (DC). Η διαδικασία περιλαμβάνει την αλλαγή της κατεύθυνσης της ροής του ρεύματος, συνήθως χρησιμοποιώντας συσκευές ημιαγωγών όπως διόδους, για την εξαγωγή των θετικών τμημάτων της κυματομορφής AC και τη δημιουργία συνεχούς τάσης συνεχούς ρεύματος.

Στην πραγματικότητα η πλακέτα τροφοδοτείται με τάση διαφοράς δυναμικού από -9 έως +9 volts έτσι ώστε να μπορέσουν να λειτουργήσουν τα ολοκληρωμένα LM741. Για την επίτευξη του σκοπού αυτού δημιουργήθηκαν δυο ιδία κυκλώματα. Το πρώτο κύκλωμα είναι συνδεδεμένο στην είσοδο του φίλτρου που μελετάται. Η τάση περνάει από το ολοκληρωμένο ώστε να γίνει η μετατροπή εναλλασσόμενου σε συνεχές και πηγαίνει απευθείας σε μια αναλογική είσοδο του Arduino Uno για ανάγνωση.

Το δεύτερο κύκλωμα είναι συνδεδεμένο στην έξοδο του φίλτρου που μελετάται και αφού γίνει η μετατροπή από AC σε DC καταλήγει σε μια δεύτερη αναλογική είσοδο του Arduino Uno. Ειδοποιός διαφορά όμως στην προκείμενη περίπτωση είναι πως το φίλτρο παρεμβάλλεται στο κύκλωμα της μετατροπής της τάσης και η συνθέτη αντίσταση παραλληλίζεται με το κύκλωμα στην έξοδο του με αποτέλεσμα οι μετρήσεις να μην ανταποκρίνονται στην πραγματικότητα.

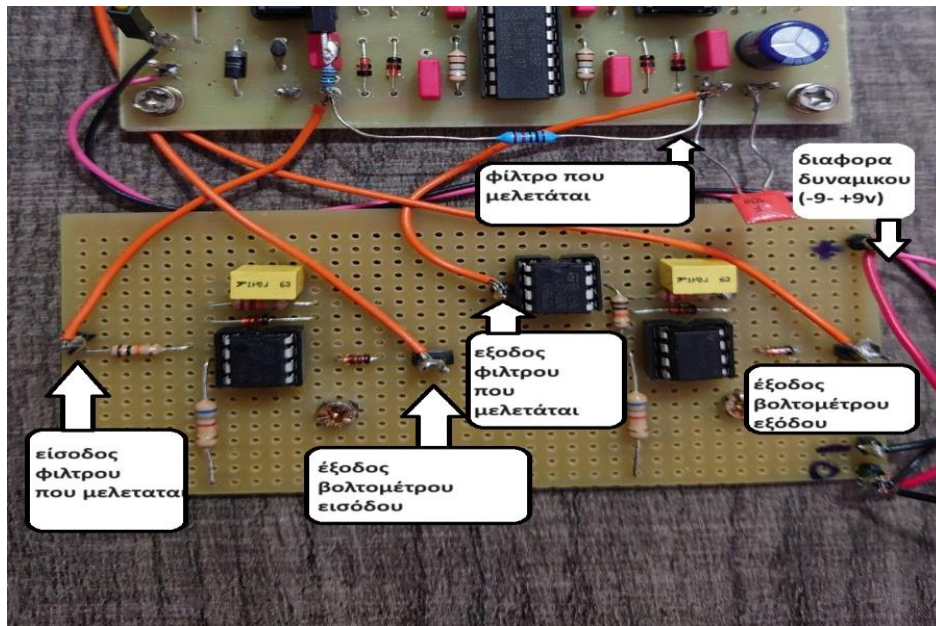


Εικόνα 6. Κύκλωμα μετατροπής με τροφοδοσία για το ολοκληρωμένα [26].



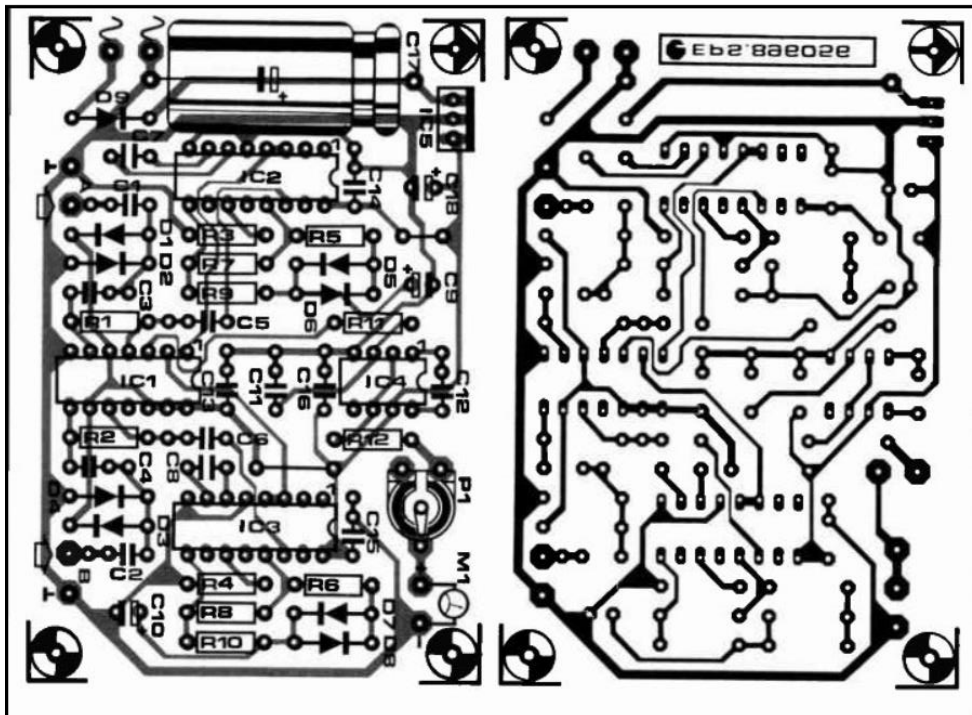
Εικόνα 7. Κύκλωμα μετατροπής από Vrms σε VDC.

Για την επίλυση του προβλήματος αυτού χρησιμοποιήθηκε ένας απομονωτής (buffer) έτσι ώστε να αυξηθεί η συνθήκη αντίστασης και να μειωθεί η αντίσταση εισόδου εξασφαλίζοντας έτσι την απομόνωση της εισόδου από την έξοδο δηλαδή η απολαβή τάσης να είναι ίση με 1.



Εικόνα 8. Υλοποιημένη διάτρητη πλακέτα για το κύκλωμα μετατροπής τάσης

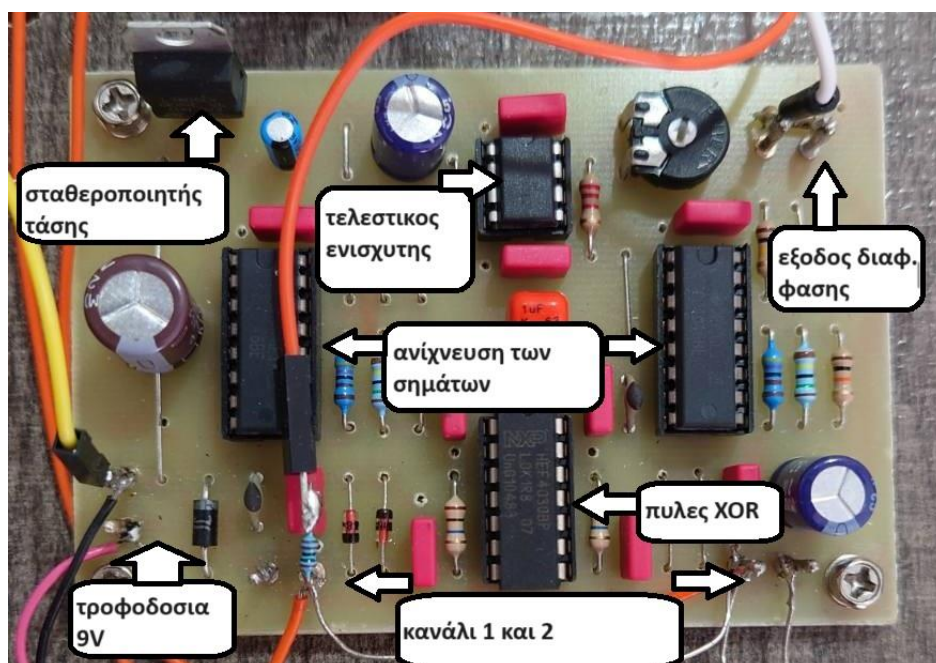
4.1.2 Κύκλωμα μέτρησης φάσης



Εικόνα 9. Διάταξη και τυπωμένο κύκλωμα σε πλακέτα διπλής όψης [27].

Το κύκλωμα που χρησιμοποιήθηκε με σκοπό να μετρηθεί η διαφορά φάσης είναι ένα σύνθετο κύκλωμα με ιδιαίτερη πολυπλοκότητα στην κατασκευή του αποτελούμενο από πολλά εξαρτήματα. Ξεκινώντας την ανάλυση του συνθέτου αυτού κυκλώματος επισημαίνεται πως μπορεί να λειτουργήσει είτε με εναλλασσόμενη είτε με συνεχή τάση από 9-12 volt εξαιτίας του σταθεροποιητή που μετατρέπει την τάση σε 5 volt έτσι ώστε όλο το κύκλωμα να τροφοδοτείται με την συγκεκριμένη τάση. Με σκοπό την δημιουργία διαφοράς φάση ενσωματώνεται αναμεσα στο κανάλι 1 και το κανάλι 2 ένα βαθυπερατό φίλτρο (RC) που εξαρτάται από τη συχνότητα που δίνεται από τη γεννήτρια συχνοτήτων που είναι συνδεδεμένη στην είσοδο του φίλτρου.

Στη συνέχεια τα δυο ολοκληρωμένα PLL συγχρονίζονται με τα δυο σήματα που δίνονται στα δυο κανάλια αντιστοίχως, παρακολουθούν δηλαδή τις γρήγορες εναλλαγές συχνότητας στα σήματα εισόδου. Οι πύλες XOR βρίσκουν τη διαφορά φάσης των δυο σημάτων η οποία με τη σειρά της περνάει από ένα βαθυπερατό φίλτρο με τελεστικό ενισχυτή οπότε όσο πιο μεγάλη είναι η διαφορά φάσης των δυο συχνοτήτων, τόσο πιο μεγάλη να είναι και η τάση που παρέχεται από το buffer. Στην έξοδο της πλακέτας έχουμε προσκολλησει μια αντίσταση 3.2kΩ έτσι ώστε το αποτέλεσμα να εμφανίζεται σαν τάση για να μπορεί να το αναγνωρίσει το Arduino Uno, αλλά θα μπορούσε να αναγνωσθεί και σαν ρεύμα από ένα μιλιαμπερόμετρο. Η πλακέτα τροφοδοτείται από το κάτω αριστερό μέρος της με 9V τάση συνεχούς ρεύματος που παρέχεται από μια μπαταριά.



Εικόνα 10. Υλοποιημένο κύκλωμα σε επιχαλωμένη πλακέτα.

4.1.3 Κύκλωμα μέτρησης συχνότητας

Για την μέτρηση της συχνότητας συνδέουμε στο κύκλωμα μας εξωτερική πηγή από την οποία παράγονται οι συχνότητες που απαιτούνται για την ανάγκη του πειράματος.



Εικόνα 11. Γεννήτρια συχνοτήτων.

Το μοντέλο που χρησιμοποιήθηκε είναι το “TG230” της εταιρίας -Thurlby Thandar Instruments- με τα εξής χαρακτηριστικά[28] :

- Πολύ υψηλή ποιότητα κυματομορφής σε όλες τις συχνότητες και τα επίπεδα
- 20Vpk-pk από 50 Ω ή 600 Ω, συν έξοδο TTL/CMOS
- Αλλαγή συχνότητας 1000:1 με εξωτερική τάση
- Μεταβλητή συμμετρία με σταθερή συχνότητα
- Μεταβλητή μετατόπιση DC με μηδενική αναστολή
- Εσωτερική σάρωση σύνδεσης/καταγραφής ακριβείας
- Ψηφιακή ένδειξη συχνότητας
- Ψηφιακή ένδειξη πλάτους και μετατόπισης

Η γεννήτρια συχνοτήτων συνδέεται στην είσοδο του φίλτρου που ερευνάται, όντας προσκολλημένο στην πλακέτα της διαφοράς φάσης. Από το κανάλι 1 της πλακέτας μέσω μιας αντίστασης 20KΩ με σκοπό να γίνει ρύθμιση του σήματος, το καλώδιο πηγαίνει σε μια ψηφιακή είσοδο του Arduino Uno για την ανάγνωση της τιμής της συχνότητας μέσω του display.

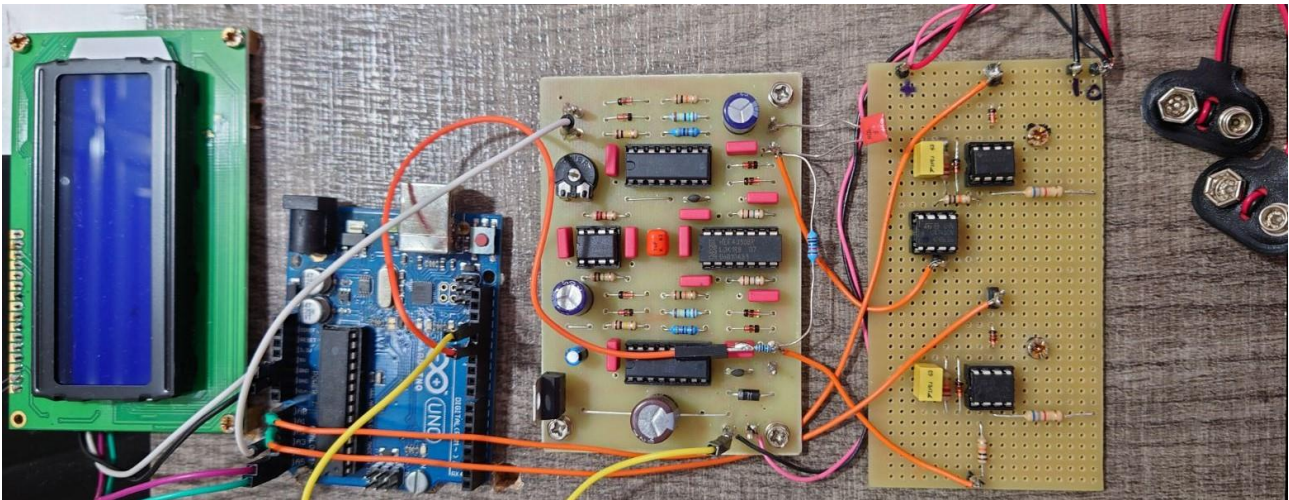
4.1.4 Κύκλωμα μέτρησης φάσης και μέτρου

Στο κεφάλαιο αυτό παρουσιάζεται ολοκληρωμένη η κατασκευή καταμέτρησης τιμών για την δημιουργία των διαγραμμάτων bode. Επισημαίνεται ο τρόπος με τον οποίο είναι συνδεδεμένες οι πλακέτες μεταξύ τους και πως λειτουργούν συλλογικά για να συλλεχτούν όλα τα δεδομένα για την δημιουργία των διαγραμμάτων. Σε ότι αφορά το προγραμματιστικό περιβάλλον που στήθηκε για να συλλεγεί τις τιμές, ανατρέχοντας στο παράρτημα, υπάρχει ολόκληρος ο κώδικας που έχει γραφεί.

Συνοψίζοντας, από τα αριστερά προς τα δεξιά πρωτίστως βρίσκεται η οθόνη LCD στην οποία εμφανίζονται οι μετρήσεις τις οποίες έχει στείλει η πλακέτα Arduino Uno.

Στο Arduino Uno καταφτάνουν στις ψηφιακές και αναλογικές εισόδους του τα σήματα που στέλνουν οι άλλες δυο πλακέτες συνδεδεμένες πάντα με γείωση. Οι γειώσεις, προς αποφυγήν παρεμβολών είναι συνδεδεμένες σε σχηματισμό κεντρικού κόμβου στο κέντρο του κυκλώματος (Starpoint) το οποίο βρίσκεται στην γείωση της τροφοδοσίας της πλακέτας δημιουργίας διαφοράς φάσης. Στην συγκεκριμένη πλακέτα βρίσκεται και το φίλτρο (RC) το οποίο ερευνάται. Για να δημιουργηθεί διαφορά φάσης συνδέουμε την γεννήτρια συχνοτήτων στην αρχή του φίλτρου οπού βρίσκεται συνδεδεμένη και η ψηφιακή είσοδος του Arduino Uno έτσι ώστε να μπορέσει να

ανιχνεύσει την εκάστοτε συχνότητα. Η διαφορά φάσης καταφθάνει και αυτή σαν τάση σε μια αναλογική είσοδο του Arduino Uno.



Εικόνα 12. Ολόκληρη η κατασκευή καταγραφής μετρήσεων.

Επιπλέον στα κανάλια (1, 2) είναι συνδεδεμένη και η τελευταία πλακέτα που κάνει την μετατροπή από V_{rms} σε V_{DC} και στέλνει στο Arduino Uno τις δυο τιμές σε μορφή τάσης, σε δυο ξεχωριστές αναλογικές εισόδους. Η πρώτη τιμή καταμετρά την τάση στην είσοδο του φίλτρου ενώ η δεύτερη τάση αντιστοιχεί στην έξοδο του φίλτρου, αφού έχει δηλαδή δημιουργηθεί η διαφορά φάσης.

Τέλος διακρίνονται δυο κλιπ βύσματα μπαταρίας 9V που δημιουργούν διαφορά δυναμικού για να δουλέψουν τα ολοκληρωμένα της πλακέτας που μετατρέπουν την τάση. Από τη μια εκ των δυο μπαταριών τροφοδοτείται και η πλακέτα διαφοράς φάσης.

Κεφάλαιο 5: Μετρήσεις από πραγματικό κύκλωμα για δημιουργία διαγράμματος Bode

Το κεφάλαιο αυτό είναι χωρισμένο σε τρία μέρη. Στο πρώτο μέρος παρουσιάζονται οι θεωρητικές μετρήσεις που έχουν προκύψει από την θεωρητική μελέτη που έγινε στο κεφάλαιο 3.3.

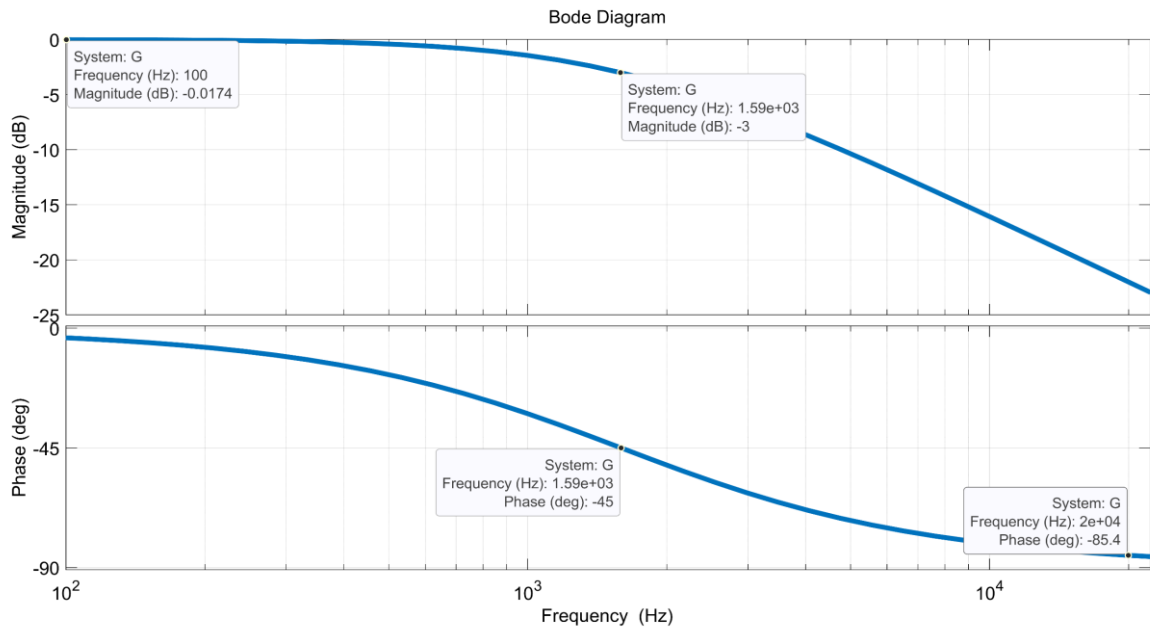
Στην δεύτερη ενότητα παρουσιάζονται οι μετρήσεις που καταγράφηκαν από τον χώρο του εργαστήριου με την χρήση του παλμογράφου. Στο τρίτο μέρος του κεφαλαίου αυτού παρουσιάζονται οι μετρήσεις που προέκυψαν με τη χρήση του εκπαιδευτικού συστήματος που κατασκευάστηκε και αναλύθηκε. Στο τέταρτο και τελευταίο μέρος επισημαίνονται μερικές παρατηρήσεις σχετικά με τις αποκλίσεις και τις διαφορές που προέκυψαν από τους τρεις διαφορετικούς τρόπους μέτρησης του βαθυπερατού φίλτρου στις εκάστοτε συχνότητες.

5.1 Θεωρητικές τιμές

Σε αυτή την ενότητα παρουσιάζονται οι θεωρητικές τιμές της απόκρισης συχνότητας του RC φίλτρου (Πίνακας 2) και το θεωρητικό διάγραμμα Bode (εικόνα 13) του βαθυπερατού φίλτρου για $R = 10k\Omega$ και $C = 10nF$.

Πίνακας 2. Μετρήσεις από θεωρητική μελέτη εξισώσεων.

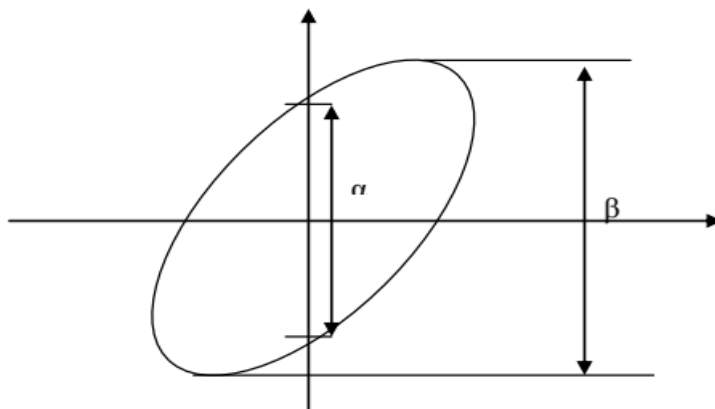
Συχνότητα (Hz)	Τάση εισόδου V_{in} (Vpp)	Τάση εξόδου V_{out} (Vpp)	Διαφορά φάσης $\Delta\phi$ (°)	$\frac{V_{out}}{V_{in}}$
100	5	4.96	3.43	0.992
200	5	4.95	6.84	0.99
400	5	4.85	14	0.97
500	5	4.75	17.22	0.95
800	5	4.45	26.5	0.89
1200	5	3.95	36.86	0.79
1500	5	3.75	43.22	0.75
1591.54 (f_{0p})	5	3.535	45	0.707
2000	5	3.1	51.34	0.62
3000	5	2.3	62	0.46
5000	5	1.5	72.28	0.3
10000	5	0.75	80.95	0.15
15000	5	0.5	83.93	0.1
20000	5	0.39	85.44	0.078



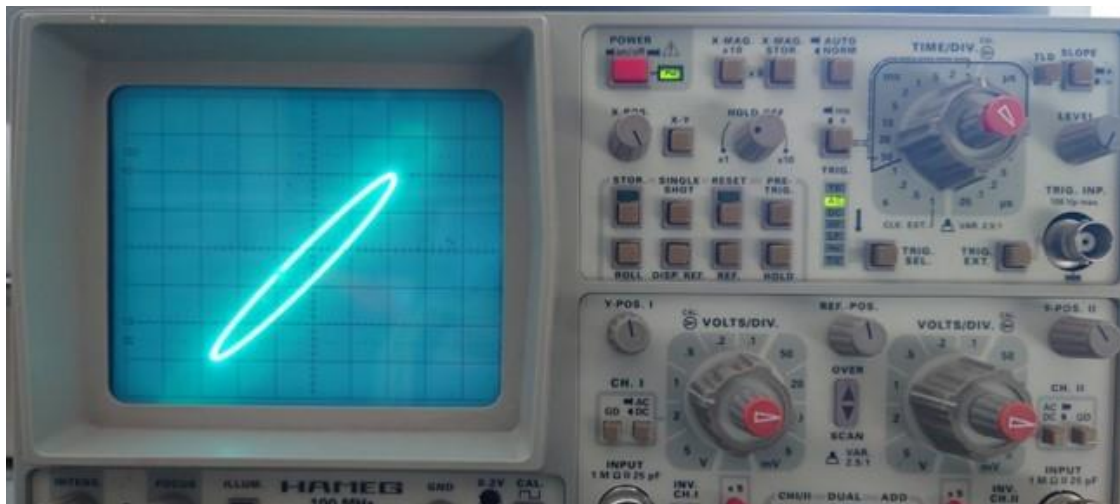
Εικόνα 13. Διάγραμμα Bode θεωρητικής ανάλυσης.

5.2 Μετρήσεις με τη χρήση παλμογράφου και σχήματα Lissajous

Για τον υπολογισμό της διαφοράς φάσης κάνοντας χρήση του παλμογράφου χρησιμοποιήθηκαν τα σχήματα Lissajous (x-y), όπου το κανάλι ένα είναι συνδεδεμένο στην είσοδο του κυκλώματος και το κανάλι δύο στην έξοδο του κυκλώματος. Ο υπολογισμός της διαφοράς φάσης γίνεται υπολογίζοντας την τιμή $\varphi = \sin^{-1} \frac{a}{\beta}$, όπου τα a , β υπολογίζονται από τα σχήματα Lissajous, όπως παρουσιάζεται στις εικόνες 14 και 15.



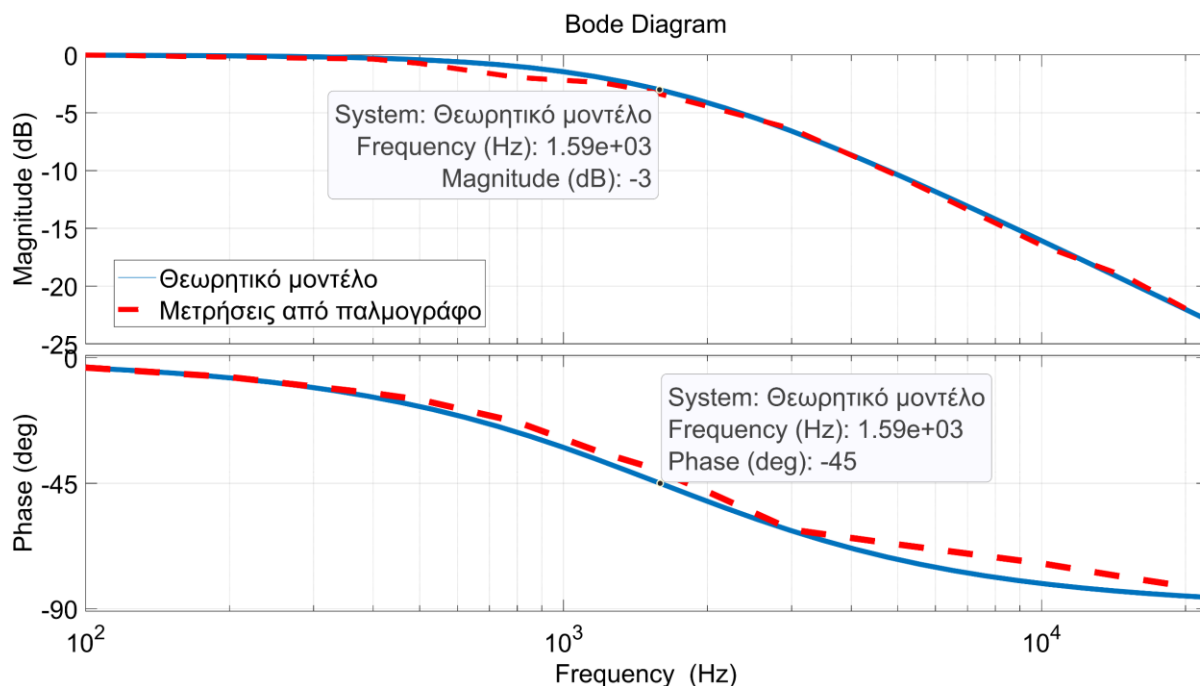
Εικόνα 14 Υπολογισμός διαφοράς φάσης με χρήση x-y απεικόνισης στον παλμογράφο.



Εικόνα 15. Σχήμα Lissajous (x-y) στα 500Hz

Πίνακας 3. Μετρήσεις με τη χρήση παλμογράφου.

Συχνότητα (Hz)	Τάση εισόδου V_{in} (Vpp)	Τάση εξόδου V_{out} (Vpp)	Διαφορά φάσης $\Delta\phi$ ($^{\circ}$)	$\frac{V_{out}}{V_{in}}$
100	5	5	-3.5	1
200	5	4.9	-6.9	0.98
400	5	4.8	-12.7	0.96
500	5	4.6	-15	0.92
800	5	4	-23	0.8
1200	5	3.8	-34	0.76
1500	5	3.5	-39	0.7
2000	5	3	-48	0.6
3000	5	2.4	-61.65	0.48
5000	5	1.5	-66.92	0.3
10000	5	0.75	-73.74	0.15
15000	5	0.55	-78.72	0.11
20000	5	0.4	-82	0.08



Εικόνα 16. Διαγράμματα Bode θεωρητικού μοντέλου και μετρήσεων από παλμογράφο.

Όπως διακρίνεται στην εικόνα 16, με την βοήθεια των διαγραμμάτων Bode που σκιαγραφήθηκαν μέσω του MATLAB, οι καμπύλες του θεωρητικού μοντέλου και των μετρήσεων μέσω παλμογράφου έχουν μικρή απόκλιση, συγκεκριμένα οι καμπύλες του μέτρου σχεδόν ταυτίζονται.

Επίσης, στο διάγραμμα Bode φαίνεται και η οριακή συχνότητα του φίλτρου ($f_{op} = 1591.54 \text{ Hz}$) στα -3dB .

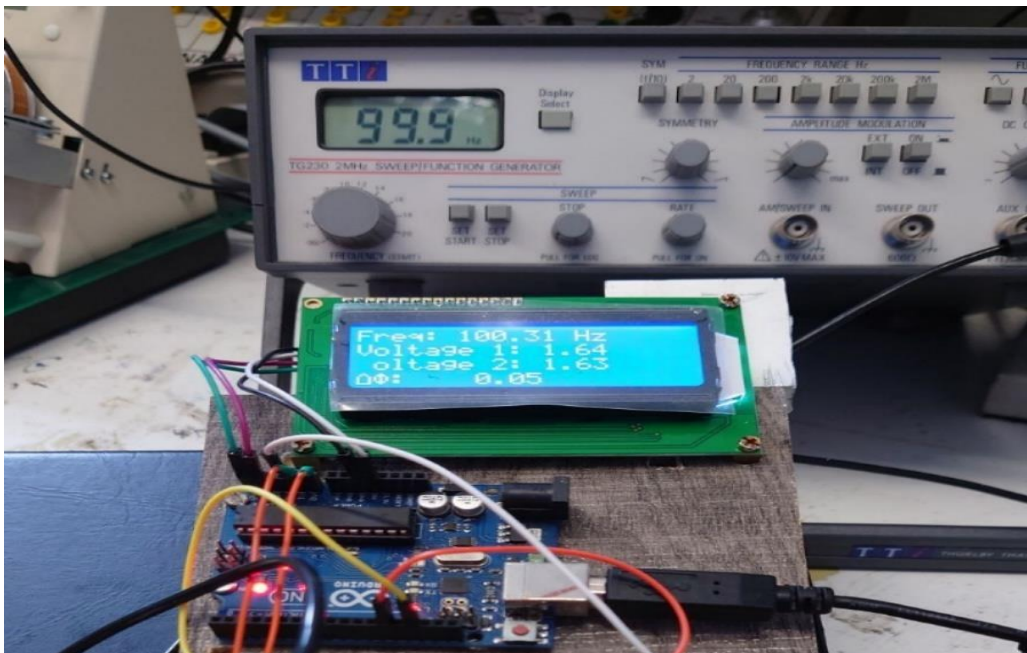
5.3 Μετρήσεις με τη χρήση της εκπαιδευτικής συσκευής.

Το κύκλωμα που κατασκευάστηκε, διαπιστώθηκε πως παρουσιάζει σφάλμα της τάξης του 6.8% στην μέτρηση της τάσης συγκριτικά με την πραγματική τιμή που επαληθεύτηκε από τον παλμογράφο. Αυτό πιθανότατα οφείλεται στο ότι τα ολοκληρωμένα και τα παθητικά στοιχεία χρησιμοποιήθηκαν παρουσιάζουν ανοχή στις ονομαστικές τιμές τους, καθώς και στον περιορισμό που παρουσιάζει το Arduino στο να μετρήσει με μεγάλη ακρίβεια την τάση. Επιπλέον αξίζει να σημειωθεί πως το ποσοστό αυτό είναι ικανοποιητικό από τη στιγμή που το κύκλωμα που ανιχνεύει την τάση είναι πολύ φθινό σε σχέση με τον παλμογράφο ή ενός πολύμετρου μεγάλης ακρίβειας που κοστίζουν περισσότερο. Σε αυτό σημαντικό παράγοντα έχει και ο μικροελεγκτής του Arduino καθώς είναι ένας από τους πιο ευρέως διαδεδομένους, απλοϊκούς μικροελεγκτές, που κυκλοφορούν στο εμπόριο.

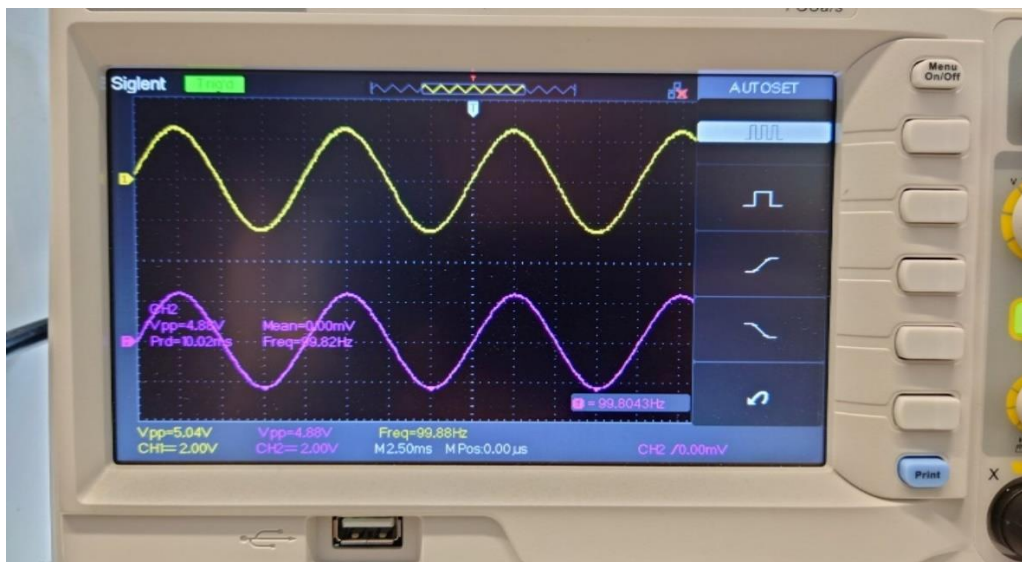
Στη συνέχεια, σχετικά με την συχνότητα παρατηρήθηκε μια πολύ μικρή διαφορά στη μέτρηση της κατασκευής συγκριτικά με εκείνη της πραγματικής. Στην κατασκευή μέτρησης της διαφοράς φάσης παρατηρήθηκε ότι ανεβάζοντας το φάσμα της συχνότητας το σφάλμα μεγαλώνει. Αυτό έχει ως αποτέλεσμα το σύστημα να λαμβάνει μέτρηση μεγαλύτερη των 90° πράγμα που είναι αδύνατο καθώς τα βαθυπερατά φίλτρα πρώτης τάξης δεν παρουσιάζουν διαφορά φάσης μεγαλύτερη των 90 μοιρών. Οι διαφορές αυτές μπορούν να παρατηρηθούν, ανατρέχοντας στους πίνακες 2 και 4 από

οπού αξιοποιήθηκαν και οι τιμές αυτών για την σκιαγράφηση των διαγραμμάτων Bode που βρίσκονται στην εικόνα 19 μέσω MATLAB.

Τέλος αξίζει να σημειωθεί, πως ενώ η θεωρητική μελέτη υποδεικνύει πως η οριακή συχνότητα βρίσκεται στα 1591.54Hz περίπου, πρακτικά αυτό δεν ισχύει. Η οριακή συχνότητα έχει μετατοπιστεί κοντά στα 1200Hz και αυτό οφείλεται στα περιφερειακά εξαρτήματα που παρεμβάλλονται στο φίλτρο, όπως ο μετατροπέας τάσης από AC σε DC και ο απομόνωσής (buffer). Τα στοιχεία αυτά εξακριβώθηκαν με την χρήση του παλμογράφου.



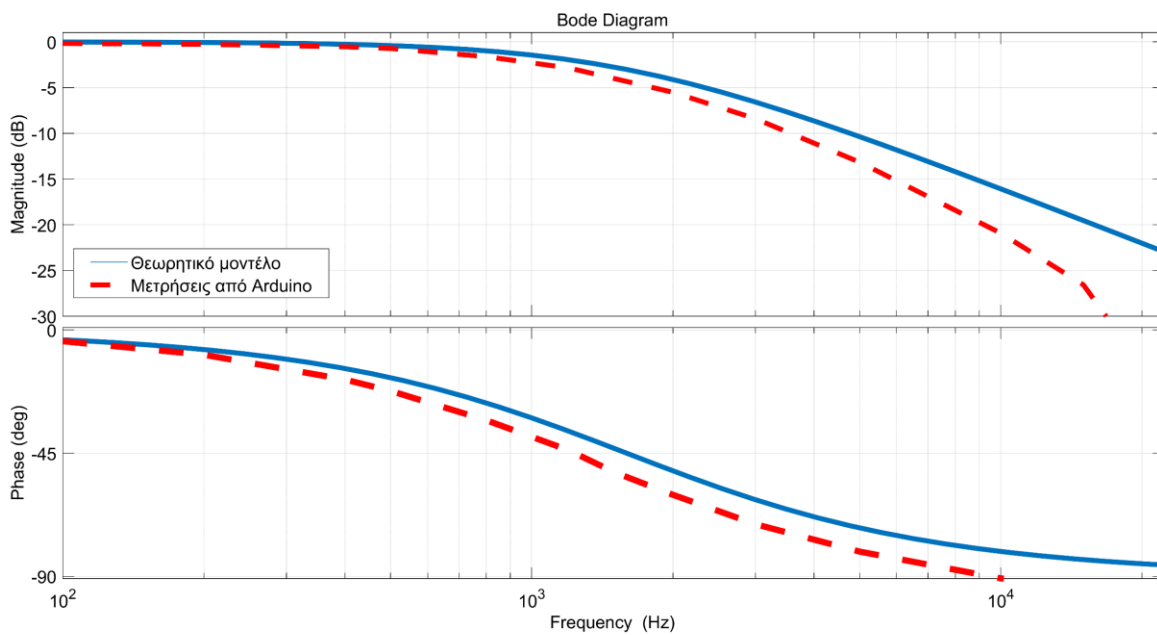
Εικόνα 17. Ενδείξεις μετρήσεων στα 100Hz.



Εικόνα 18. Επαλήθευση μετρήσεων μέσω παλμογράφου.

Πίνακας 4. Μετρήσεις με τη χρήση της κατασκευής.

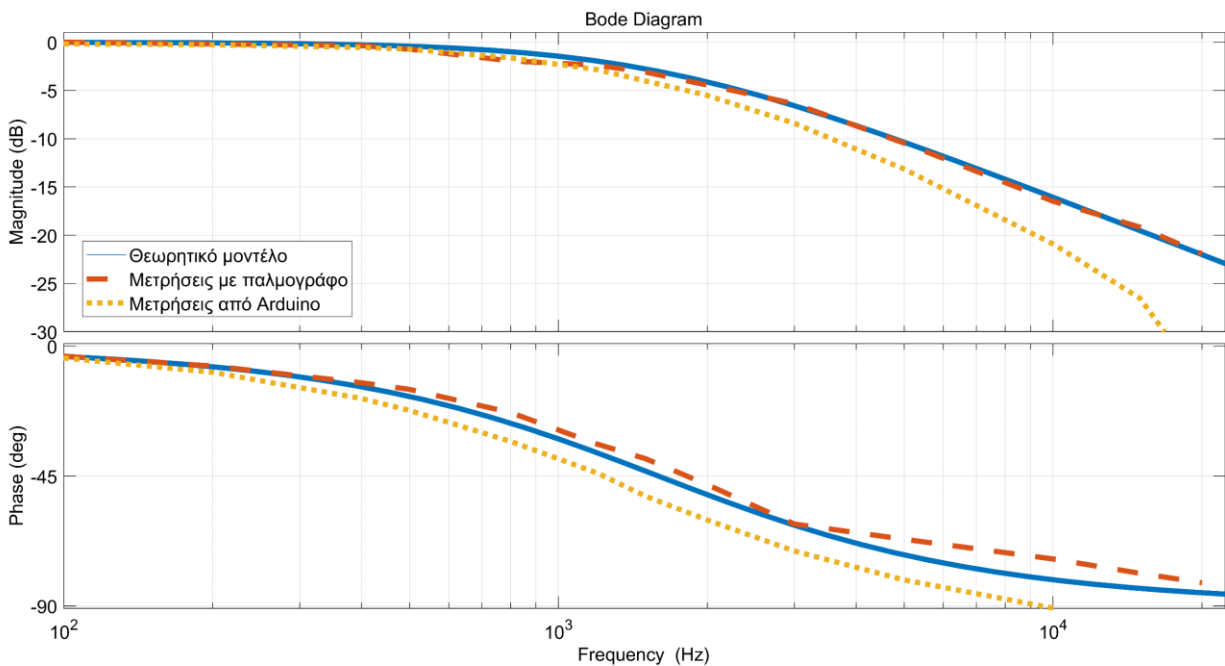
Συχνότητα (Hz)	Τάση εισόδου V_{in} (Vpp)	Τάση εξόδου V_{out} (Vpp)	Διαφορά φάσης $\Delta\phi$ (°)	$\frac{V_{out}}{V_{in}}$
100	4.66	4.58	-4.12	0.98
200	4.66	4.55	-9.07	0.97
400	4.66	4.41	-18.15	0.94
500	4.66	4.29	-22.28	0.92
800	4.66	3.9	-33	0.83
1200	4.66	3.36	-44	0.72
1500	4.66	2.96	-52	0.63
2000	4.66	2.48	-60.25	0.53
3000	4.66	1.78	-71	0.38
5000	4.66	1.07	-81	0.22
10000	4.66	0.42	-90.79	0.09
15000	4.66	0.22	-94	0.047
20000	4.66	0.08	-97	0.017



Εικόνα 19. Διαγράμματα Bode θεωρητικού μοντέλου και μετρήσεων από παλμογράφο.

5.4 Σύγκριση και αποκλίσεις τιμών

Στην εικόνα 20 παρουσιάζονται τα γραφήματα μέτρου και φάσης από τις τρεις μεθόδους μετρήσεων. Συμπερασματικά καταλήγει κανείς στο συμπέρασμα ότι, εξαιτίας του χαμηλού κόστους της εκπαιδευτικής συσκευής, οι μετρήσεις των τάσεων βρέθηκαν να έχουν 6.8% σφάλμα από αυτές του παλμογράφου και η μέτρηση για την διαφορά φάσης από την κατασκευή να έχει μέγιστη διαφορά 15 μοίρες στα 20kHz. Τα όρια αυτά είναι αποδεκτά στα πλαίσια της πειραματικής κατασκευής από τη στιγμή που μέσω αυτής μπορούν να μελετηθούν απλοϊκά κυκλώματα όπως αυτά των βαθυπερατών και υψηλοπερατών φίλτρων. Από την άλλη μεριά οι μετρήσεις του παλμογράφου έχουν λιγότερο από 5% σφάλμα σε σχέση με τις θεωρητικές τιμές.



Εικόνα 20. Σύγκριση διαγραμμάτων Bode.

Κεφάλαιο 6: Επίλογος

6.1 Συμπεράσματα

Εξετάζοντας τις μετρήσεις που προέκυψαν από το θεωρητικό μοντέλο (ενότητα 5.1), από τις μετρήσεις που προέκυψαν με τη χρήση του παλμογράφου (ενότητα 5.2) και από τις μετρήσεις μέσω της εκπαιδευτικής κατασκευής καταλήξαμε στο συμπέρασμα πως υπάρχουν ορισμένες αποκλίσεις.

Το θεωρητικό μοντέλο ταυτίζεται σε μεγάλο ποσοστό με τις μετρήσεις του παλμογράφου, διάφορες όπως είναι διακριτές στην εικόνα 20. Η μέτρηση της φάσης με την χρήση της κατασκευής έχει 15 μοίρες μέγιστη απόκλιση από τον παλμογράφο και οι ενδείξεις των τάσεων εισόδου και εξόδου του φίλτρου έχουν σφάλμα 6.8% συγκρίνοντας παλμογράφο και κατασκευή. Ωστόσο παρατηρώντας τα διαγράμματα πάρα τις αισθητές αποκλίσεις η κατασκευή κάνει σωστή παρατήρηση του φίλτρου οπότε μπορεί να ειπωθεί πως η κατασκευή ικανοποιεί τις συνθήκες για τις οποίες κατασκευάστηκε.

Αξίζει να σημειωθεί πως το βαθυπερατό φίλτρο κατά το οποίο έγιναν οι δοκιμές, είναι απλό κύκλωμα και ενδείκνυται για εκπαιδευτικούς σκοπούς σαν πρώτη γνωριμία με τον φοιτητή. Η κατασκευή φέρει μερικά σημαντικά θετικά χαρακτηριστικά όπως τόσο το χαμηλό κόστος κατασκευής, όσο και η εύκολη χρησιμότητα του στην ανάγνωση των μετρήσεων. Περαιτέρω η κατασκευή είναι μικρή σε μέγεθος πράγμα που την καθιστά εύκολη στην μεταφορά της εκτός του εργαστηρίου σαν μια μικρή φορητή συσκευή σε αντίθεση με τα εργαστηριακά μηχανήματα και τους παλμογράφους τα οποία χρειάζονται για να μελετηθούν τέτοιου είδους κυκλώματα. Ακόμη ένα θετικό εκτός της φορητότητας είναι πως η κατασκευή λειτουργεί με μπαταρίες οπότε μπορεί να λειτουργήσει και σε εξωτερικά σημεία που δεν υπάρχει παροχή ρεύματος.

6.2 Μελλοντικές επεκτάσεις

Μια από τις σημαντικότερες σκέψεις για πιθανή βελτίωση και επέκταση της εκπαιδευτικής κατασκευής είναι να βρεθεί τρόπος ώστε να μειωθεί το σφάλμα των μετρήσεων και η κάθε μέτρηση να βρίσκεται αναμεσά στο φάσμα μετρήσεων του θεωρητικού μοντέλου και του παλμογράφου. Αυτό μπορεί να επιτευχθεί αν χρησιμοποιηθούν ακριβότερα υλικά από αυτά που χρησιμοποιήθηκαν στα επιμέρους κυκλώματα έτσι ώστε να πληρούν τις προδιαγραφές αυστηρότερης ακριβείας.

Ακόμη, μια παρεμβατική βελτίωση η οποία θα μπορούσε να γίνει, είναι να αποθηκεύονται οι μετρούμενες τιμές αυτόματα σε έναν πίνακα. Αυτό πρακτικά σημαίνει πως θα χρειάζονταν πολύ λιγότερος χρόνος να μελετηθεί ένα φίλτρο στο πεδίο της συχνότητας, χωρίς ο χρήστης να σημειώνει χειρόγραφα τις τιμές. Οι μετρήσεις αυτές στη συνέχεια μπορούν να χρησιμοποιηθούν για την δημιουργία του διαγράμματος Bode σε μια μεγαλύτερη έγχρωμη οθόνη στο Arduino. Με αυτό τον τρόπο θα υπάρχει η δυνατότητα να παρακαμφθεί εντελώς ο υπολογιστής. Φυσικά αυτό θα μπορούσε να επιτραπεί με έναν καλύτερο και πιο σύγχρονο μικροελεγκτή, αρά με ένα ακριβότερο Arduino από όπου θα πρόκυπτε και ταχύτερη καταγραφή των μετρήσεων.

Παραρτήματα: Κώδικας Arduino

Στο παρακάτω παράρτημα βρίσκεται ο κώδικας που γράφτηκε με σκοπό την δημιουργία του συχνότερου και των τριών βολτομέτρων για τη διαφορά φάση και τις τάσεις στην είσοδο και έξοδο του φίλτρου.

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

#define LCD_I2C_ADDRESS      0x3f // I/O i2c controller οθόνης
#define LCD_CHARS            20 // Μήκος Οθόνης
#define LCD_LINES            4 // Γραμμές οθόνης
#define INPUT_FREQ_PIN       12 // Ψηφιακή είσοδος πρώτου σήματος
#define INPUT_VOLT_PIN1      A0 // Αναλογική είσοδος βολτόμετρου 1
#define INPUT_VOLT_PIN2      A1 // Αναλογική είσοδος βολτόμετρου 2
#define INPUT_VOLT_PIN3      A2 // Αναλογική είσοδος βολτόμετρου 3

/* Δηλώσεις μεταβλητών */
int high_time, low_time;
float time_period, frequency, volt_in, volt_out, volt_in_sum, volt_out_sum, tmp1,
tmp2, tmp3, volt_in3, volt_in3_sum;
LiquidCrystal_I2C lcd(LCD_I2C_ADDRESS, LCD_CHARS, LCD_LINES);

/* Γραφικός χαρακτήρας Δελτα */
byte gr_delta[] = { B00100,
                    B01010,
                    B10001,
                    B10001,
                    B10001,
                    B10001,
                    B10001,
                    B11111,
                    B00000 }; // Δ

/* Γραφικός χαρακτήρας Φι */
byte gr_fi[] = { B00100,
                 B01110,
                 B10101,
                 B10101,
                 B10101,
                 B01110,
                 B00100,
                 B00000 }; // Φ

void setup() {
```

ΚΑΤΑΣΚΕΥΗ ANALYTH BODE ME
ΧΡΗΣΗ ARDUINO

```

/* Αρχικοποίηση συστήματος */
Serial.begin(9600);          /* Ενεργοποίηση σειριακής θύρας στα 9600 baud
*/
lcd.init();                 /* Ενεργοποίηση οθονης LCD */
lcd.backlight();           /* Ενεργοποίηση του οπίσθιου φωτισμού της
οθόνης */
pinMode(INPUT_FREQ_PIN, INPUT); /* Αρχικοποίηση της ψηφιακής εισόδου 11 */
pinMode(INPUT_VOLT_PIN1, INPUT); /* Αρχικοποίηση της αναλογικής εισόδου A0 */
pinMode(INPUT_VOLT_PIN2, INPUT); /* Αρχικοποίηση της αναλογικής εισόδου A1 */
pinMode(INPUT_VOLT_PIN3, INPUT); /* Αρχικοποίηση της αναλογικής εισόδου A2 */
lcd.createChar(0, gr_delta); /* Δημιουργία Ελληνικού χαρακτήρα Δέλτα (Δ) */
lcd.createChar(1, gr_fi); /* Δημιουργία Ελληνικού χαρακτήρα Φι (Φ) */

lcd.clear();               /* Καθαρισμός οθόνης LCD */
lcd.print("Ptixiaki Ergasia"); /* Εμφάνιση μηνύματος υποδοχής */
lcd.setCursor(0, 1);
lcd.print("KOLOKATHIS");
lcd.setCursor(0, 2);
lcd.print("AGGELOS");
lcd.setCursor(0, 3);
lcd.print("BODE PLOT");
delay(3000);

/* Καθαρισμός και εμφάνιση βασικής οθόνης */
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Freq:");
  lcd.setCursor(0, 1);
  lcd.print("Voltage 1:");
  lcd.setCursor(0, 2);
  lcd.print("Voltage 2:");
  lcd.setCursor(0, 3);
  /* Εμφάνιση ειδικών χαρακτήρων Δελτα και Φι */
  lcd.write(0); // Δελτα
  lcd.write(1); // Φι
  lcd.print(":");
}

/* Κύρια επαναληπτική συνάρτηση */

void loop()
{
  high_time = pulseIn(INPUT_FREQ_PIN, HIGH); /* Αναγνωση ενός παλμου HIGH απο
την ψηφιακή είσοδο και καταχώρηση της χρονικής διάρκειας του παλμου στη μεταβλητή
high_time */

```

```
low_time = pulseIn(INPUT_FREQ_PIN, LOW);      /* Αναγνωση ενός παλμου LOW απο την
ψηφιακή είσοδο και καταχώρηση της χρονικής διάρκειας του παλμου στη μεταβλητή
low_time*/
```

```
/* Έλεγχος και υπολογισμός της περιόδου και της συχνότητας */
if (high_time != 0 || low_time != 0) {
    time_period = (high_time + low_time);
    time_period = (time_period / 1000.0);
    frequency = (1000.0 / time_period);
}
```

```
/* Δειγματοληψία 20 τιμών απο τις αναλογικές θύρες */
```

/*Όλες οι αναλογικές είσοδοι έχουν μέσα τους ένα ADC converter 10bit. Επομένως η τάση που εμφανίζεται στην αναλογική είσοδο γίνεται ψηφιακός αριθμός. Το αναλογικό σήμα που μπορεί να διαβάσει οποιοδήποτε από τα pins του Arduino Uno είναι από 0 έως 5V.

Αν υπερβεί αυτή την τάση θα κάψουμε τον μικροελεγκτή. Άρα χρειάζεται ιδιαίτερη προσοχή έτσι ώστε κανένα ποδαράκι να μην δεχθεί μεγαλύτερη τάση από αυτή των 5V.

Επειδή δεν ξέρουμε αν κατά λάθος δώσουμε μεγαλύτερη τάση από 5V για να διασφαλίσουμε ότι δεν θα κάψουμε τον μικροελεγκτή τοποθετούμε μία δίοδο zener 5,1V με

ανάστροφη πολικότητα από την αναλογική είσοδο προς την γή. Οποιαδήποτε τάση μεγαλύτερη από την τάση της zener θα ψαλιδιστεί στα 5,1V μέσω της διόδου.

Ακρίβεια μέτρησης:

Το 10bit που έχουν οι είσοδοι του arduino σημαίνει ότι δυαδική τιμή που μπορεί να πάρει ο ψηφιακός αριθμός είναι 10 ψηφίων, δηλαδή παίρνει τιμές έως 1024.(από 0 έως 1023)

Αυτό σημαίνει ότι μπορεί να διακρίνει 1023 βήματα του αναλογικού σήματος. Αυτό καθορίζει την ακρίβεια της μέτρησης δηλαδή το πιο μικρό ποσό τάσης που μπορεί να διακρίνει ο μετατροπέας.

Αν έχουμε σήμα 0-5V θα έχουμε $5/1023 = 0,0048875855327468$ δηλαδή το μικρότερο ποσό τάσης που μπορεί να διακρίνει ο μετατροπέας είναι περίπου 5mV.

Αν τώρα θέλουμε να μετρήσουμε μία τάση από 0-30V (μέσω του διατρήτη τάσης) θα έχουμε $30/1023 = 0,0293255131964809$ βλέπουμε δηλαδή ότι η μικρότερη τιμή που μπορεί να διακρίνει

ο μετατροπέας είναι περίπου 30mV. Επομένως όσο μεγαλύτερη είναι η τιμή της τάσης του αναλογικού σήματος τόσο μειώνεται η ακρίβεια.

Αν θέλουμε μεγαλύτερη ακρίβεια από 1024 θα πρέπει να χρησιμοποιήσουμε ένα εξωτερικό μετατροπέα με περισσότερα bit π.χ. 12bit

αυτό σημαίνει 4096 βήματα οπότε έχουμε $30/4096 = 0,00732421875$ δηλαδή περίπου 7mV ακρίβεια.

```

*/

for (unsigned int tmp = 0; tmp < 20; tmp++) {
    volt_in += analogRead(INPUT_VOLT_PIN1);
    volt_out += analogRead(INPUT_VOLT_PIN2);
    volt_in3 += analogRead(INPUT_VOLT_PIN3);
    delay(5);
}

/* Υπολογισμός μέσης τιμής απο τις δειγματοληπτούμενες τιμές */
tmp1 = (volt_in / 20);
tmp2 = (volt_out / 20);
tmp3 = (volt_in3 / 20);

volt_in_sum = (tmp1 * 5.0) / 1024.0;
volt_out_sum = (tmp2 * 5.0) / 1024.0;
volt_in3_sum = (tmp3 * 5.0) / 1024.0;

/* Όταν οι τρεις τάσης που έχουμε δειγματοληπτήσει είναι πάνω απο 0.1v τότε
προχωράμε στην εμφάνιση τους στην οθόνη */
if (volt_in_sum > 0.1 || volt_out_sum > 0.1 || volt_in3_sum > 0.1 ) {

    if (high_time == 0 || low_time == 0) { /* Δεν έχουμε καθόλου συχνότητα,
επομένως εμφανίζουμε μήνυμα N/A (Not available) */
        lcd.setCursor(6, 0);
        lcd.print("N/A          ");
    } else {

        if (frequency > 1000) { /* Αν η συχνότητά μας είναι πάνω απο 1000 hz
εμφανίζουμε μετά τη μέτρηση το khz */

            float tmp = frequency / 1000;

            lcd.setCursor(6, 0);
            lcd.print(tmp);
            lcd.print(" kHz");
        } else { /* Ειδικάλλως εμφανίζουμε hz */
            lcd.setCursor(6, 0);
            lcd.print(frequency);
            lcd.print(" Hz");
        }
    }
}

/* Εμφάνιση των τιμών των τάσεων στην οθόνη */
lcd.setCursor(11, 1);
lcd.print(volt_in_sum);
lcd.setCursor(11, 2);

```

```

    lcd.print(volt_out_sum);
    lcd.setCursor(7, 3);
    lcd.print(volt_in3_sum);

}

// Εμφάνιση μετρήσεων στη σειριακή θύρα
Serial.println(" -- METRHSEIS --");
Serial.print("SIXNOTITA: ");
Serial.println(frequency);
Serial.print("TASH V1: ");
Serial.println(volt_in_sum);
Serial.print("TASH V2: ");
Serial.println(volt_out_sum);
Serial.print("TASH V3: ");
Serial.println(volt_in3_sum);

Serial.println(" -- TELOS METRHSEWN --");

// Μηδενισμός όλων των μεταβλητών
volt_in = volt_out = volt_in_sum = volt_out_sum = time_period = high_time =
low_time = frequency = volt_in3_sum = tmp3 = volt_in3 = 0;

delay(500);          /* Χρονοκαθυστερηση 500μs */

```

Βιβλιογραφία

Αναφορά σε ιστοσελίδα:

[1] <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>

Αναφορά σε ιστοσελίδα :

[2] <https://www.electronicshub.org/microcontrollers-basics-structure-applications/>

Αναφορά σε ιστοσελίδα:

[3] <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/microcontrollers-guide>

Αναφορά σε ιστοσελίδα:

[4] http://users.sch.gr/asal1/material/seminaria/teliko24_1.pdf

Αναφορά σε ιστοσελίδα:

[5] <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

Αναφορά σε ιστοσελίδα:

[6] <https://robocraze.com/blogs/post/what-is-the-microcontroller-used-in-arduino-uno>

Αναφορά σε ιστοσελίδα:

[7] <https://docs.arduino.cc/hardware/uno-rev3-smd?queryID=undefined>

Αναφορά σε ιστοσελίδα:

[8] <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>

Αναφορά σε ιστοσελίδα:

[9] <https://www.arduino.cc/reference/en/language/functions/communication/wire/>

Αναφορά σε ιστοσελίδα:

[10] <https://reference.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>

Αναφορά σε ιστοσελίδα:

[11] <https://www.arduino.cc/reference/en/language/structure/further-syntax/define/>

Αναφορά σε ιστοσελίδα:

[12] <https://www.arduino.cc/reference/tr/language/structure/sketch/setup/>

Αναφορά σε ιστοσελίδα:

- [13] <https://www.arduino.cc/reference/en/language/functions/communication/serial/begin/>
Αναφορά σε ιστοσελίδα:
- [14] <https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>
Αναφορά σε ιστοσελίδα:
- [15] <https://roboticsbackend.com/arduino-setup-loop-functions-explained/>
Αναφορά σε ιστοσελίδα:
- [16] <https://docs.arduino.cc/built-in-examples/control-structures/ifStatementConditional>
Αναφορά σε ιστοσελίδα:
- [17] <https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/>
Αναφορά σε ιστοσελίδα:
- [18] https://arduinogetstarted.com/reference/library/lcd-setcursor?utm_content=cmp-true
Αναφορά σε ιστοσελίδα:
- [19] <https://arduinogetstarted.com/reference/library/lcd-print>
Αναφορά σε ιστοσελίδα:
- [20] https://www.w3schools.com/cpp/cpp_conditions.asp
Αναφορά σε ιστοσελίδα:
- [21] <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>
Αναφορά σε ιστοσελίδα:
- [22] <https://arduinogetstarted.com/reference/serial-println>
Αναφορά σε ιστοσελίδα:
- [23] <https://arduinogetstarted.com/reference/serial-print>
Αναφορά σε ιστοσελίδα:
- [24] <https://www.arduino.cc/reference/en/language/functions/time/delay/>
Αναφορά σε ιστοσελίδα:
- [25] https://www.cse.uoi.gr/~tsiatouhas/MYY203/Section_5-2p.pdf
Αναφορά σε ιστοσελίδα:
- [26] <https://www.nutsvolts.com/magazine/article/op-amp-cookbook-part-4>
-

Αναφορά σε ιστοσελίδα:

[27] https://www.industrial-electronics.com/elektor_usa_1990-10_inter.html

Αναφορά σε ιστοσελίδα:

[28] https://www.brldtest.com/index.php?main_page=product_info&products_id=3067

[29] <https://ecstudiosystems.com/discover/textbooks/basic-electronics/filters/passive-low-pass-filter/>

[30] <https://www.bennett.edu.in/media-center/blog/what-are-microcontrollers-as-b-tech-ece-specializations-and-what-types-of-their-applications/>

