



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σχεδίαση και κατασκευή στερεοφωνικού ενισχυτή
βασισμένου στο νέο τύπο λυχνιών Nutube»



Των φοιτητών
Γεώργιος Αναγνωστάκης (514005)
Παναγιώτης Τσομπανόπουλος (517143)

Επιβλέπων
Άγγελος Γιακουμής
Βαθμίδα: Επίκουρος Καθηγητής

Ημερομηνία 26/01/2025

Τίτλος: Σχεδίαση και κατασκευή στερεοφωνικού ενισχυτή βασιζόμενου στο νέο τύπο λυχνιών Nutube

Κωδικός: 23245

Φοιτητής: Γεώργιος Αναγνωστάκης – Παναγιώτης Τσομπανόπουλος

Εισηγητής: Άγγελος Γιακουμής

Ημερομηνία ανάληψης: 15/09/2023

Ημερομηνία περάτωσης: 29/01/2025

Βεβαιώνουμε ότι είμαστε οι συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχουμε καταγράψει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμάς προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητών Γεώργιος Αναγνωστάκης – Παναγιώτης Τσομπανόπουλος που την εκπόνησαν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, οι συγγραφείς/δημιουργοί εκχωρούν 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 των συγγραφέων/δημιουργών, 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 των συγγραφέων/δημιουργών.

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

Πρόλογος

Η υλοποίηση της εργασίας με θέμα «Σχεδίαση και κατασκευή στερεοφωνικού ενισχυτή βασιζόμενου στο νέο τύπο λυχνιών Nutube» διήρκησε περίπου 10 μήνες, από τον Απρίλιο του 2024 μέχρι τον Ιανουάριο του 2025. Για την καλύτερη οργάνωση των ζητούμενων της, έγινε ο αυθαίρετος διαχωρισμός της σε δύο μεγάλους τομείς, στους οποίους οι εργασίες έγιναν παράλληλα: Το υλικό (hardware) και το λογισμικό (software). Οι φοιτητές συνεργάστηκαν στην έρευνα και την αγορά των κατάλληλων εξαρτημάτων, στη μελέτη και την ανάπτυξη κώδικα με τις απαραίτητες γλώσσες προγραμματισμού και σήμανσης, στη σχεδίαση και κατασκευή του κυκλώματος του συστήματος, στον έλεγχο και τις μετρήσεις επάνω στο σύστημα, στην επίλυση των προβλημάτων που προέκυψαν κατά την υλοποίηση, καθώς και στη συγγραφή του παρόντος εγγράφου. Το τελικό αποτέλεσμα της εργασίας ήταν επιτυχές, καθώς ο ενισχυτής που αναπτύχθηκε είναι πλήρως λειτουργικός και το ψηφιακό χειριστήριο είναι εύχρηστο και φιλικό προς τον χρήστη.

Περίληψη

Η θεματολογία της συγκεκριμένης εργασίας είναι η σχεδίαση και κατασκευή στερεοφωνικού ενισχυτή βασιζόμενου στο νέο τύπο λυχνιών Nutube. Η εργασία διαχωρίζεται σε τρία μεγάλα κεφάλαια: Υλικό (Hardware), Λογισμικό (Software) και Αποτελέσματα. Στο πρώτο κεφάλαιο γίνεται μία εισαγωγή στους ενισχυτές και τις σημαντικότερες τάξεις ενισχυτών, αναφέρονται τα εξαρτήματα που χρησιμοποιήθηκαν για την υλοποίηση του ενισχυτή και το πρόγραμμα σχεδιασμού επάνω στο οποίο έγινε ο σχεδιασμός του ηλεκτρονικού κυκλώματος και του PCB, ενώ γίνεται μια σε βάθος ανάλυση του κυκλώματος που σχεδιάστηκε και του τρόπου λειτουργίας του. Στο δεύτερο κεφάλαιο γίνεται μια εισαγωγή στις γλώσσες προγραμματισμού Arduino και JavaScript, στις γλώσσες σήμανσης HTML και CSS, αλλά και στο πρωτόκολλο I2C και στις βιβλιοθήκες κώδικα που χρησιμοποιήθηκαν, ενώ γίνεται μια σε βάθος ανάλυση του κώδικα που αναπτύχθηκε για την κατασκευή ενός λειτουργικού ψηφιακού χειριστηρίου, το οποίο χρησιμοποιείται για τον έλεγχο του ενισχυτή μέσω WiFi ή Hotspot συνδέσεων. Στο τρίτο κεφάλαιο εξηγείται ο τρόπος λειτουργίας του ενισχυτή, αποτυπώνονται οι μετρήσεις με τις οποίες έγινε ο έλεγχος της ορθής λειτουργίας του κυκλώματος και ο εντοπισμός σφαλμάτων, αναφέρονται οι διάφοροι τρόποι με τους οποίους θα μπορούσε να βελτιωθεί η εργασία σε μία μελλοντική εκδοχή της και αναλύονται τα συμπεράσματα στα οποία κατέληξαν οι φοιτητές κατά τη διάρκεια της υλοποίησης της εργασίας. Στο τέλος της εργασίας αναφέρονται οι 107 βιβλιογραφικές πηγές οι οποίες αξιοποιήθηκαν για την έγκυρη συγγραφή του συγκεκριμένου εγγράφου, ενώ ακολουθούν τα παραρτήματα στα οποία εμπεριέχονται τα σημαντικότερα κομμάτια των datasheet από τα εξαρτήματα που χρησιμοποιήθηκαν στο ηλεκτρονικό κύκλωμα, αλλά και ο ολοκληρωμένος κώδικας της εργασίας.

«Design and manufacture of a stereo amplifier based on the new Nutube tube type»

George Anagnostakis – Panagiotis Tsompanopoulos

Abstract

The subject of this project is the design and construction of a stereo amplifier based on the new type of Nutube tubes. The paper is divided into three major chapters: Hardware, Software and Results. In the first chapter an introduction to amplifiers and the most important classes of amplifiers is made, the components used to implement the amplifier and the design program on which the electronic circuit and the PCB were designed are mentioned, and an in-depth analysis of the designed circuit and its mode of operation is given. The second chapter provides an introduction to the Arduino and JavaScript programming languages, the HTML and CSS markup languages, as well as the I2C protocol and the code libraries used, and an in-depth analysis of the code developed to build a functional digital controller, which is used to control the amplifier via WiFi or Hotspot connections. The third chapter explains how the amplifier works, describes the measurements used to check the correct operation of the circuit and detect errors, lists the different ways in which the project could be improved in a future version and discusses the conclusions that the students came to during the implementation of the project. At the end of the paper the 107 bibliographic sources that were used for the valid writing of this paper are listed, while the appendices follow, which contain the most important parts from the datasheets of the components used in the electronic circuit, as well as the complete code of the project.

Ευχαριστίες

Ευχαριστούμε τους καθηγητές για την καθοδήγηση τους, καθώς και την οικογένεια και τους φίλους μας για την υποστήριξη τους κατά τη διάρκεια εκπόνησης της συγκεκριμένης εργασίας.

Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iii
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Εικόνων	x
Κατάλογος Πινάκων.....	xi
Συνομογραφίες.....	xii
Κεφάλαιο 1ο: Υλικό (Hardware)	1
1.1 Εισαγωγή.....	1
1.2 Ενισχυτές.....	1
1.3 Ενισχυτής Τάξης D PAM8320RDR.....	4
1.4 Ψηφιακός Επεξεργαστής Σήματος (DSP) TDA7718N	4
1.5 Ρυθμιστές Τάσης	6
1.5.1 Ρυθμιστής Τάσης AMS1117-3.3	6
1.5.2 Ρυθμιστής Τάσης L78L09ACZ	7
1.6 JFET LSK389.....	8
1.7 Τελεστικός Ενισχυτής NE5532PS.....	8
1.8 Λυχνία Nutube.....	9
1.9 WEMOS D1 mini Clone με ESP8266.....	10
1.10 EasyEDA.....	13
1.11 Ανάλυση του Κυκλώματος της Εργασίας	13
1.11.1 Κύκλωμα Ψηφιακού Επεξεργαστή Σήματος (DSP).....	14
1.11.2 Κύκλωμα Nutube	16
1.11.3 Ενισχυτές Εξόδου Τάξης D.....	18
1.11.4 Ψηφιακός Έλεγχος	18
1.12 PCB	19
1.13 Επίλογος.....	21
Κεφάλαιο 2ο: Λογισμικό (Software)	22
2.1 Εισαγωγή.....	22
2.2 Arduino & Arduino IDE.....	22
2.3 Πρωτόκολλο I2C	25

2.4	Βιβλιοθήκες Κώδικα	26
2.4.1	Wire	26
2.4.2	ESP8266WiFi	27
2.4.3	ESPAsyncWebServer & ESPAsyncTCP.....	28
2.5	HTML & HTML5	29
2.6	CSS & CSS3.....	30
2.7	JavaScript & AJAX.....	32
2.8	Ανάλυση του Κώδικα της Εργασίας	33
2.8.1	Συμπερίληψη των Βιβλιοθηκών	34
2.8.2	Διαπιστευτήρια Wi-Fi, Στατική IP & Αρχικοποίηση Αντικειμένου Server.....	34
2.8.3	Αρχικοποίηση Βασικών Σταθερών και Μεταβλητών	34
2.8.4	Συνάρτηση: writeToDSP()	36
2.8.5	Συνάρτηση: volumeMapping().....	37
2.8.6	Συνάρτηση: dBMapping()	37
2.8.7	Συνάρτηση: setupDSP().....	38
2.8.8	Συνάρτηση: replaceAll().....	38
2.8.9	Συνάρτηση: getHTML()	39
2.8.10	void setup() και Αρχικοποίηση των Βασικών Pins	43
2.8.11	Ρύθμιση WiFi, Wire και DSP.....	43
2.8.12	Server: Κατασκευή της HTML Σελίδας.....	44
2.8.13	Server: Έλεγχος Τιμών – Ένταση του Ήχου.....	44
2.8.14	Server: Έλεγχος Τιμών – Μπάσα.....	45
2.8.15	Server: Έλεγχος Τιμών – Μεσαία.....	46
2.8.16	Server: Έλεγχος Τιμών – Πρίμα.....	47
2.8.17	Τελικές Ρυθμίσεις και Εκκίνηση του Server.....	47
2.9	Επίλογος.....	48
Κεφάλαιο 3ο:	Αποτελέσματα.....	49
3.1	Εισαγωγή.....	49
3.2	Τρόπος Λειτουργίας του Ενισχυτή.....	49
3.3	Μετρήσεις.....	52
3.4	Τρόποι Βελτίωσης της Εργασίας.....	58
3.5	Συμπεράσματα.....	58
3.6	Επίλογος.....	60
BIBΛΙΟΓΡΑΦΙΑ.....		61
ΠΑΡΑΡΤΗΜΑ Α : PAM8320RDR Datasheet.....		67

ΠΑΡΑΡΤΗΜΑ Β : TDA7718N Datasheet.....	73
ΠΑΡΑΡΤΗΜΑ C : AMS1117-3.3 Datasheet.....	81
ΠΑΡΑΡΤΗΜΑ D : L78L09ACZ Datasheet.....	84
ΠΑΡΑΡΤΗΜΑ Ε : LSK389 Datasheet	89
ΠΑΡΑΡΤΗΜΑ F : NE5532PS Datasheet	97
ΠΑΡΑΡΤΗΜΑ G : Nutube Datasheet.....	101
ΠΑΡΑΡΤΗΜΑ Η : WEMOS D1 mini Clone ESP8266 Datasheet (Παρόμοιο με αυτό που χρησιμοποιήθηκε στην εργασία).....	104
ΠΑΡΑΡΤΗΜΑ Ι : Ο ΚΩΔΙΚΑΣ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	106

Κατάλογος Εικόνων

Εικόνα 1.1: Ρεύμα συλλέκτη σε ενισχυτές τάξεων A, B και C αντίστοιχα.....	2
Εικόνα 1.2: Ενισχυτής τάξης A.....	2
Εικόνα 1.3: Ενισχυτής push-pull τάξης B.....	2
Εικόνα 1.4: Ενισχυτής τάξης AB.....	3
Εικόνα 1.5: Συντονισμένος ενισχυτής τάξης C.....	3
Εικόνα 1.6: Βασικό κύκλωμα ενισχυτή τάξης D.....	4
Εικόνα 1.7: Δομή ακροδεκτών του ενισχυτή PAM8320RDR.....	4
Εικόνα 1.8: Σχήμα του ψηφιακού επεξεργαστή σήματος TDA7718N.....	5
Εικόνα 1.9: Μπλοκ διάγραμμα του DSP TDA7718N.....	5
Εικόνα 1.10: Δομή ακροδεκτών του DSP TDA7718N.....	6
Εικόνα 1.11: Δομή ακροδεκτών του AMS1117-3.3 (1: Γείωση, 2: Έξοδος, 3: Είσοδος).....	7
Εικόνα 1.12: Δομή ακροδεκτών του L78L09ACZ (1: Έξοδος, 2: Γείωση, 3: Είσοδος).....	7
Εικόνα 1.13: Κύκλωμα του ρυθμιστή τάσης L78L09ACZ.....	7
Εικόνα 1.14: Δομή ακροδεκτών του JFET LSK389.....	8
Εικόνα 1.15: Δομή ακροδεκτών του τελεστικού ενισχυτή NE5532PS.....	9
Εικόνα 1.16: Κύκλωμα του τελεστικού ενισχυτή NE5532PS.....	9
Εικόνα 1.17: Σχήμα της λυχνίας Nutube και δομή των ακροδεκτών της.....	10
Εικόνα 1.18: Οι πλακέτες της σειράς D1 από τη WEMOS.....	11
Εικόνα 1.19: Ο κλώνος του WEMOS D1 mini που χρησιμοποιήθηκε στην εργασία.....	11
Εικόνα 1.20: Δομή των ακροδεκτών του ESP8266EX.....	12
Εικόνα 1.21: Μπλοκ διάγραμμα του ESP8266EX.....	13
Εικόνα 1.22: Το επίσημο logo του EasyEDA.....	13
Εικόνα 1.23: Το κύκλωμα της εργασίας.....	14
Εικόνα 1.24: Φίλτρο ισχύος +12 V σε +9 V και βύσματα ήχου.....	15
Εικόνα 1.25: Ψηφιακός επεξεργαστής σήματος (DSP).....	15
Εικόνα 1.26: LED, διακόπτης ενεργοποίησης, αναλογική γείωση και φίλτρο ισχύος για τη μείωση του θορύβου και τον διαχωρισμό +12 V σε αριστερά και δεξιά.....	16
Εικόνα 1.27: Κύκλωμα προενισχυτή Nutube.....	17
Εικόνα 1.28: Πυκνωτές αποσύζευξης και ενισχυτές εξόδου τάξης D.....	18
Εικόνα 1.29: WiFi μικροελεγκτής και φίλτρο ισχύος +12 V σε +3.3 V.....	19
Εικόνα 1.30: Τα δύο εξωτερικά επίπεδα του PCB.....	19
Εικόνα 1.31: Τα δύο εσωτερικά επίπεδα του PCB.....	20
Εικόνα 1.32: Τρισδιάστατη απεικόνιση του επάνω μέρους της πλακέτας.....	20
Εικόνα 1.33: Τρισδιάστατη απεικόνιση του κάτω μέρους της πλακέτας.....	21
Εικόνα 2.1: Το επίσημο logo του Arduino.....	23
Εικόνα 2.2: Οκτώ από τα κλασσικά μοντέλα Arduino.....	23
Εικόνα 2.3: Το Arduino IDE 2.3.2.....	24
Εικόνα 2.4: Η δομή ενός I2C μηνύματος.....	26
Εικόνα 2.5: Το επίσημο logo της HTML5.....	29
Εικόνα 2.6: Το επίσημο logo της CSS3.....	31
Εικόνα 3.1: Διαδικασία ενεργοποίησης του ενισχυτή.....	49
Εικόνα 3.2: Αρχική μορφή της σελίδας του χειριστηρίου με τις προεπιλεγμένες τιμές.....	50
Εικόνα 3.3: Μήνυμα επιτυχίας αποστολής δεδομένων και μήνυμα αποτυχίας.....	51
Εικόνα 3.4: Χρήση του χειριστηρίου και τροποποίηση των τιμών.....	51

Εικόνα 3.5: Η αντίσταση που χρησιμοποιήθηκε για τις μετρήσεις.....	52
Εικόνα 3.6: Μέτρηση του V_{rms}	53
Εικόνα 3.7: Λάθος μέτρηση ολικής παραμόρφωσης & σήματος FFT.....	53
Εικόνα 3.8: Σχέδιο χαμηλοπερατού φίλτρου LC.....	54
Εικόνα 3.9: Κατασκευή του χαμηλοπερατού φίλτρου LC.....	55
Εικόνα 3.10: Μέτρηση ολικής παραμόρφωσης & σήματος FFT με ψαλιδισμό και σωστή μέτρηση ολικής παραμόρφωσης & σήματος FFT.....	55
Εικόνα 3.11: Συχνότητες εισόδου και αντίστοιχες τάσεις εξόδου.....	56
Εικόνα 3.12: Μέτρηση Zin πριν και μετά το ποτενσιόμετρο.....	57

Κατάλογος Πινάκων

Πίνακας 2.1: Τιμές για την ορθή λειτουργία του DSP.....	35
Πίνακας 3.1: Συχνότητες εισόδου και αντίστοιχες τάσεις εξόδου.....	56

Συντομογραφίες

DSP	Digital Signal Processor
JFET	Junction Field Effect Transistor
PCB	Printed Circuit Board
BOM	Bill of Materials
IDE	Integrated Development Environment
I2C	Inter-Integrated Circuit
IP	Internet Protocol
SVG	Scalable Vector Graphics
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
AJAX	Asynchronous JavaScript and XML
LC	Inductor-Capacitor

Κεφάλαιο 1ο: Υλικό (Hardware)

1.1 Εισαγωγή

Στο πρώτο κεφάλαιο της εργασίας γίνεται η παρουσίαση του συστήματος που υλοποιήθηκε για την κατασκευή ενός στερεοφωνικού ενισχυτή βασιζόμενου στο νέο τύπο λυχνιών Nutube. Στις πρώτες υποενότητες του κεφαλαίου γίνεται μια σύντομη περιγραφή των ενισχυτών και των διάφορων τύπων τους, καθώς και των εξαρτημάτων που χρησιμοποιήθηκαν για την κατασκευή του συστήματος (Ενισχυτής Τάξης D PAM8320RDR, Ψηφιακός Επεξεργαστής Σήματος TDA7718N, Ρυθμιστές Τάσης AMS1117-3.3 και L78L09ACZ, JFET LSK389, Τελεστικός Ενισχυτής NE5532PS, Λυχνία Nutube και WEMOS D1 mini Clone με ESP8266).

Την θεωρητική παρουσίαση των ενισχυτών, των τύπων τους και των εξαρτημάτων που χρησιμοποιήθηκαν, ακολουθεί μια σε βάθος ανάλυση του κυκλώματος που υλοποιήθηκε για την κατασκευή του συστήματος. Στην ανάλυση εξετάζεται η συνδεσμολογία μεταξύ των εξαρτημάτων και ο τρόπος λειτουργίας τους, καθώς αποσαφηνίζεται ο λόγος για τον οποίο επιλέχθηκαν.

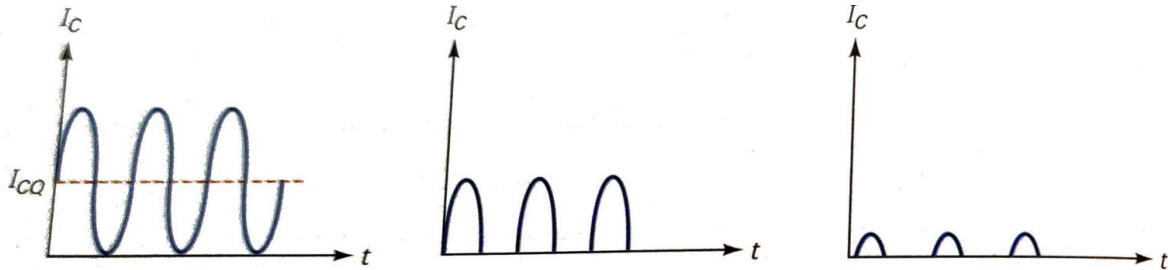
1.2 Ενισχυτές

Ενισχυτής ονομάζεται ένα σύστημα το οποίο είναι υπεύθυνο για την αύξηση της ισχύος ενός σήματος. Οι ενισχυτές λέγεται ότι έχουν κέρδος ισχύος, λόγω του γεγονότος ότι στην έξοδο του ενισχυτή παρέχεται μεγαλύτερη ισχύς από εκείνη της πηγής του σήματος [1]. Υπάρχουν διάφορες τάξεις ενισχυτών. Μερικές από αυτές είναι:

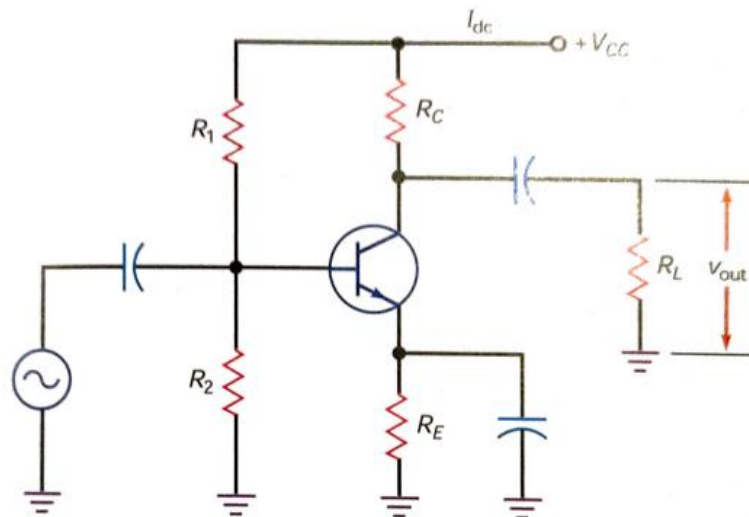
- Ενισχυτές Τάξης A - Χρησιμοποιώντας μόνο ένα τρανζίστορ εξόδου που λειτουργεί σαν διακόπτης, οι ενισχυτές τάξης A προσφέρουν υψηλό κέρδος και μεγάλη γραμμικότητα. Σε αυτούς τους ενισχυτές, το ρεύμα συλλέκτη ρέει καθ' όλη τη διάρκεια της περιόδου (γωνία αγωγής 360 μοιρών) και δεν υπάρχει καμία χρονική στιγμή κατά την οποία υφίσταται ψαλιδισμός του σήματος εξόδου. Παρ' ότι έχουν τον καλύτερο ήχο, αντιμετωπίζουν ζητήματα με την θερμότητα, τις παραμορφώσεις και τον θόρυβο, ενώ δεν θεωρούνται οι πιο αποδοτικοί. Χρησιμοποιούνται καλύτερα σε περιπτώσεις ενίσχυσης μικρών (χαμηλών) σημάτων [2, 3, 4].
- Ενισχυτές Τάξης B – Χρησιμοποιώντας δύο τρανζίστορ σε push-pull δομή τα οποία άγουν εναλλάξ κατά τη διάρκεια της περιόδου, οι ενισχυτές τάξης B κατασκευάστηκαν με στόχο την επίλυση των ζητημάτων απόδοσης που εμφανίζονταν στους ενισχυτές τάξης A. Σε αυτούς τους ενισχυτές, το ρεύμα συλλέκτη ρέει μόνο για μισή περίοδο (180 μοίρες). Το βασικό πρόβλημα με τη συγκεκριμένη τάξη είναι η παραμόρφωση της διασταύρωσης που προκύπτει από την εναλλαγή των τρανζίστορ, καθώς η ισχύς που παρέχεται κατά την ενεργοποίηση του ενός τρανζίστορ δεν είναι ακριβώς η ίδια με αυτήν που παρέχεται κατά την απενεργοποίηση του άλλου. Γι' αυτό τον λόγο, ο συγκεκριμένος δεν είναι ενισχυτής ακριβείας [2, 3, 4].
- Ενισχυτές Τάξης AB – Αποτελούν συνδυασμό των προηγούμενων δύο τάξεων. Η γωνία αγωγής βρίσκεται ανάμεσα στις 180 και τις 360 μοίρες και απόδοση τους ανάμεσα στο 50 και το 60%. Οι συγκεκριμένοι ενισχυτές έχουν το πλεονέκτημα ότι μπορούν, χωρίς να κληρονομήσουν τα προβλήματα της τάξης A, να εξαφανίσουν την παραμόρφωση της διασταύρωσης που εντοπίστηκε στους ενισχυτές τάξης B. Γι' αυτό τον λόγο, χρησιμοποιούνται αρκετά συχνά στη σχεδίαση ενισχυτών ήχου [3, 4].
- Ενισχυτές Τάξης C – Οι ενισχυτές τάξης C λειτουργούν συντονισμένα (μέση τάση εξόδου ίση με τάση τροφοδοσίας) ή μη συντονισμένα (όπου παρατηρείται μεγάλη παραμόρφωση). Το ρεύμα συλλέκτη ρέει λιγότερο από μισή περίοδο (κάτω από 180 μοίρες, κοντά στις 90). Έχουν απόδοση που πλησιάζει το 80-100%, αλλά ταυτόχρονα την χειρότερη γραμμικότητα. Λόγω της μεγάλης παραμόρφωσης στην έξοδο, δεν είναι κατάλληλοι για να λειτουργήσουν ως ενισχυτές ήχου [2, 3, 4].

Κεφάλαιο 1

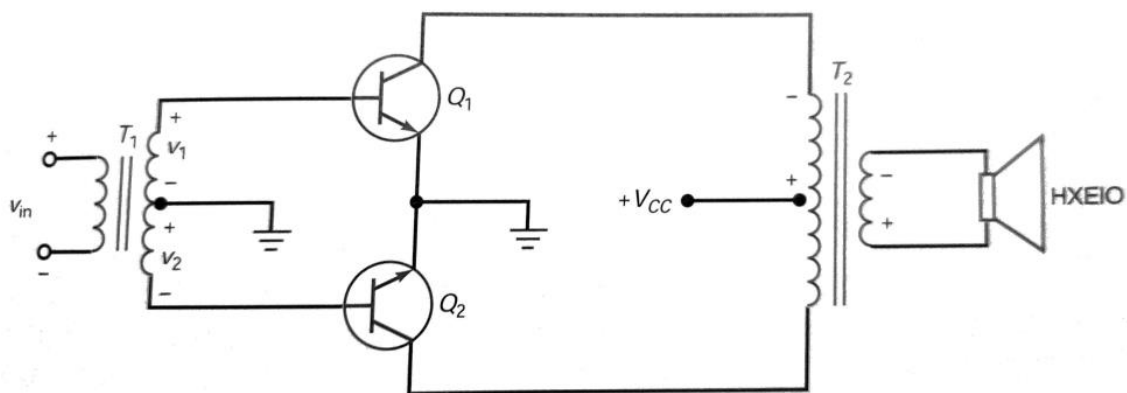
- Ενισχυτές Τάξης D – Οι μη γραμμικοί ενισχυτές τάξης D έχουν πολύ μικρή κατανάλωση ισχύος και απαγωγή θερμότητας, καθώς και μεγάλη απόδοση (πλησιάζουν το 100%), διότι χρησιμοποιούν τρανζίστορ εξόδου ως διακόπτες, τα οποία λειτουργούν στην αποκοπή (μηδενικό ρεύμα) ή στον κόρο (χαμηλή τάση στα άκρα). Η συγκεκριμένη τάξη χρησιμοποιείται σε συσκευές αναπαραγωγής ήχου [2, 3, 4].



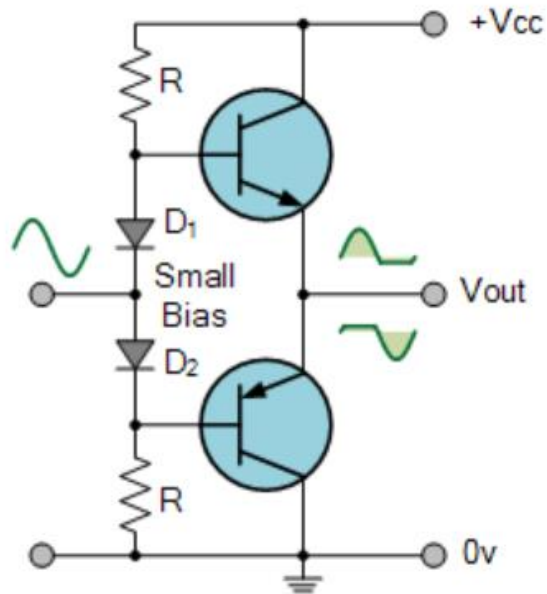
Εικόνα 1.1: Ρεύμα συλλέκτη σε ενισχυτές τάξεων A, B και C αντίστοιχα [2].



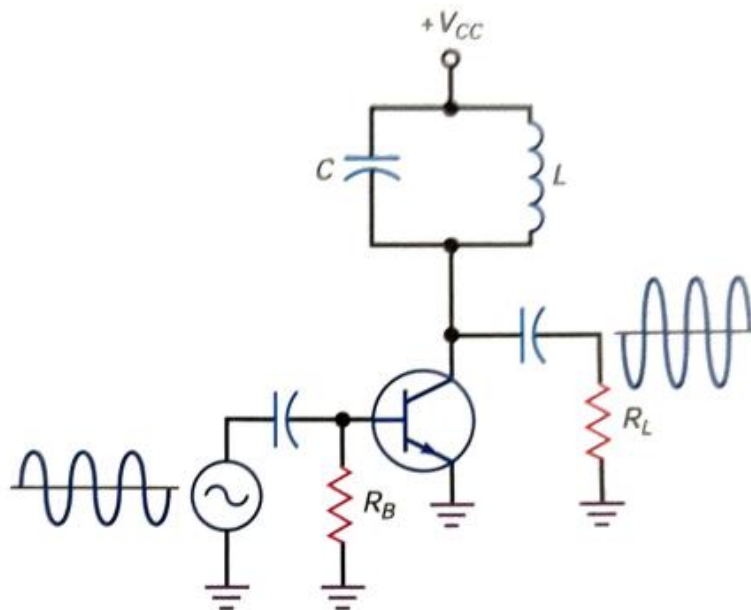
Εικόνα 1.2: Ενισχυτής τάξης A [2].



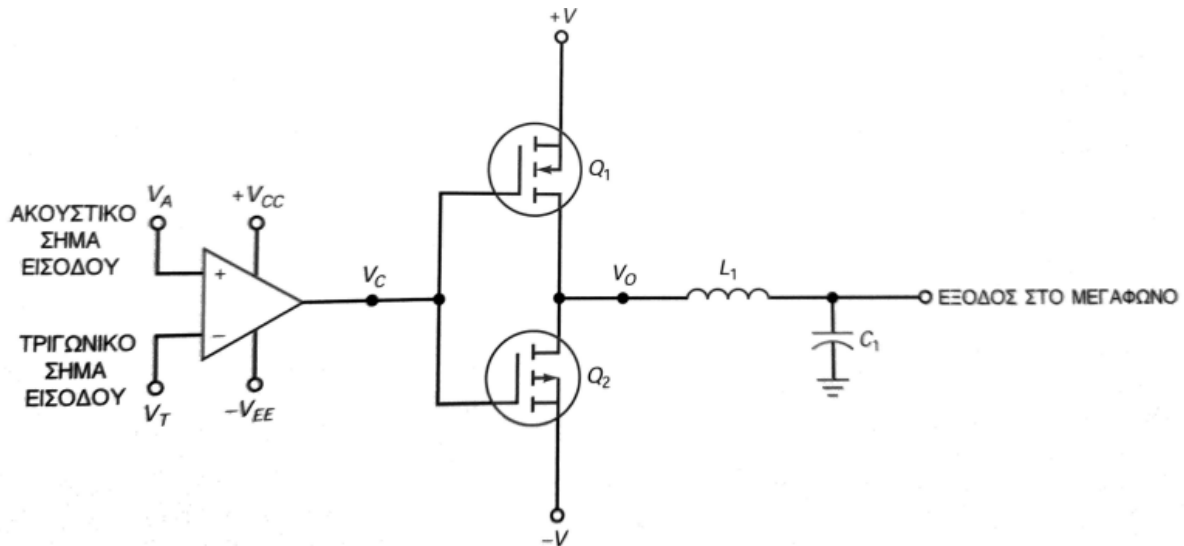
Εικόνα 1.3: Ενισχυτής push-pull τάξης B [2].



Εικόνα 1.4: Ενισχυτής τάξης AB [3].



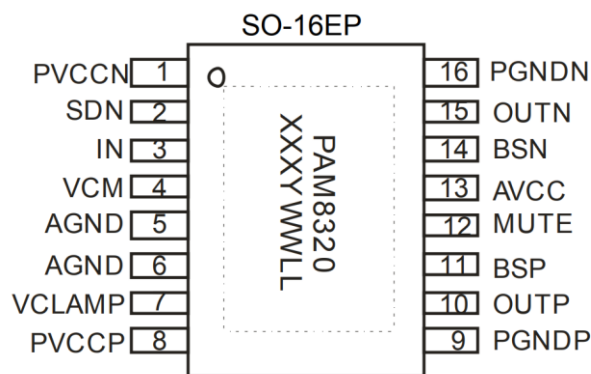
Εικόνα 1.5: Συντονισμένος ενισχυτής τάξης C [2].



Εικόνα 1.6: Βασικό κύκλωμα ενισχυτή τάξης D [2].

1.3 Ενισχυτής Τάξης D PAM8320RDR

Ο Ενισχυτής Τάξης D PAM8320RDR είναι ένας μονοφωνικός ενισχυτής ήχου των 20W, ο οποίος έχει τη δυνατότητα να χρησιμοποιηθεί χωρίς εξωτερικά ψύκτρα σε διάφορες συσκευές, όπως ηχεία υπολογιστών και οικιακά συστήματα ήχου. Ο ενισχυτής προστατεύεται από υπέρταση, λειτουργική διακοπή λόγω θερμότητας, καθώς και βραχυκύκλωμα. Εξοικονομεί ενέργεια σε περιόδους στις οποίες δε χρησιμοποιείται, καθώς εισέρχεται σε κατάσταση χαμηλού ρεύματος. Παράλληλα, είναι ειδικά διαμορφωμένος ώστε να μειώνει τους ήχους που δημιουργούνται συνήθως κατά την ενεργοποίηση και απενεργοποίηση τέτοιων εξαρτημάτων. Ο ενισχυτής λειτουργεί με τάση 4.5 έως 15 V και έχει απόδοση μέχρι και 95% [5].



Εικόνα 1.7: Δομή ακροδεκτών του ενισχυτή PAM8320RDR [5].

1.4 Ψηφιακός Επεξεργαστής Σήματος (DSP) TDA7718N

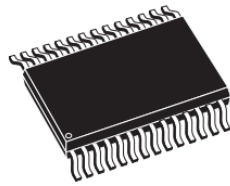
Οι ψηφιακοί επεξεργαστές σήματος, είτε είναι μικροεπεξεργαστές είτε είναι μεγάλοι υπολογιστές, έχουν ως στόχο την εφαρμογή τροποποιήσεων σε ένα αναλογικό σήμα, το οποίο, χάρη σε έναν αναλογικό-σε-ψηφιακό μετατροπέα, μετατρέπεται σε ψηφιακό σήμα το οποίο λαμβάνουν ως είσοδο.

Υπάρχουν πολλά πλεονεκτήματα που έχουν οι ψηφιακοί επεξεργαστές σήματος σε αντίθεση με τους αναλογικούς επεξεργαστές σήματος, όπως [6]:

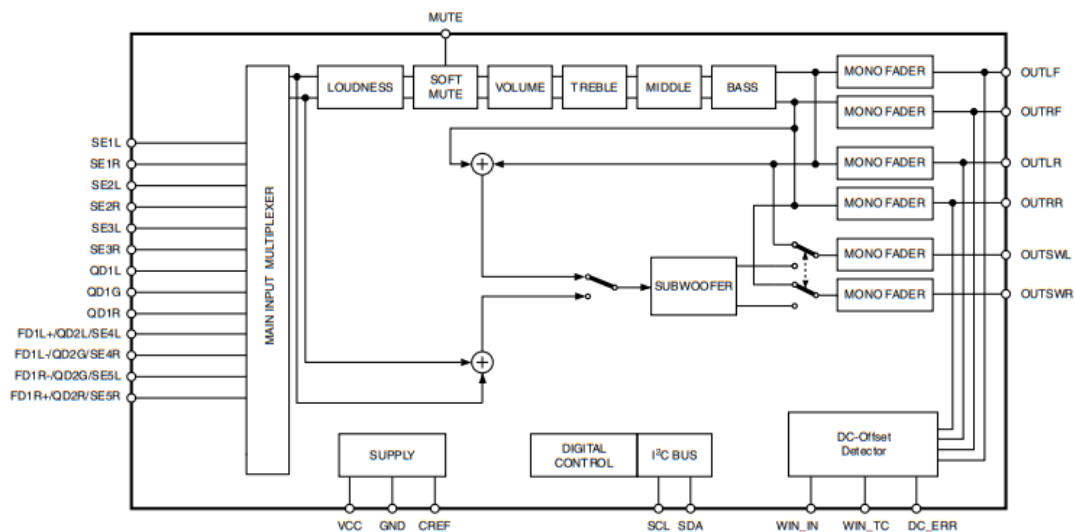
- Χαμηλότερο κόστος,
- Δυνατότητα τροποποίησης των εργασιών επεξεργασίας μέσω της τροποποίησης του λογισμικού και όχι απαραίτητα του υλικού,
- Αποθήκευση των ψηφιακών σημάτων σε μαγνητικούς αποθηκευτικούς χώρους,
- Μεγαλύτερη ακρίβεια υπολογισμών και εφαρμογής ελέγχου [6].

Στη συγκεκριμένη εργασία χρησιμοποιήθηκε ο ψηφιακός επεξεργαστής σήματος TDA7718N, ο οποίος επιτρέπει τον χειρισμό μιας μεγάλης σειράς χαρακτηριστικών φίλτρων. Ειδικότερα, ο συγκεκριμένος ψηφιακός επεξεργαστής σήματος, επιτρέπει, μεταξύ άλλων, τον έλεγχο των παρακάτω στοιχείων [7]:

- Κεντρικός Επιλογέας (Main Selector),
- Απαλή Σίγαση (Soft Mute),
- Απαλό Βήμα (Soft Step),
- Ηχηρότητα (Loudness),
- Ένταση (Volume),
- Πρίμα (Treble),
- Μεσαία (Middle),
- Μπάσα (Bass),
- Subwoofer [7].

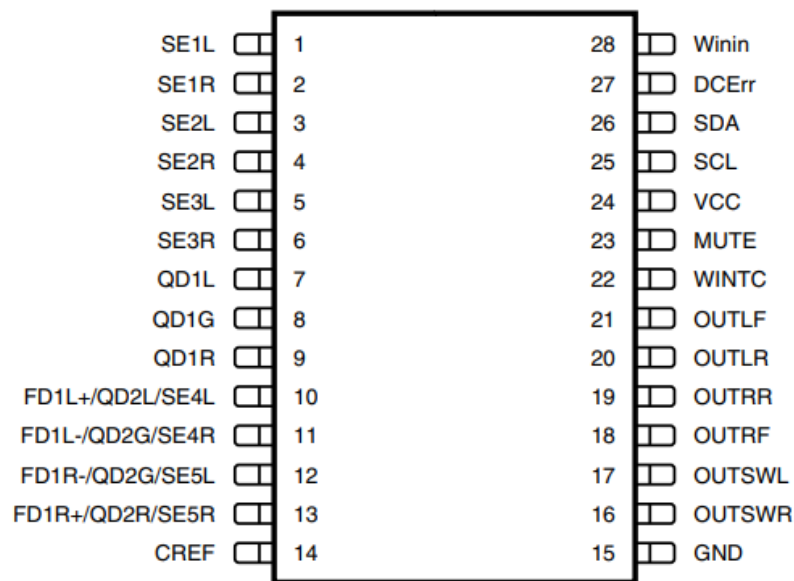


Εικόνα 1.8: Σχήμα του ψηφιακού επεξεργαστή σήματος TDA7718N [7].



Εικόνα 1.9: Μπλοκ διάγραμμα του DSP TDA7718N [7].

Μια από τις δυνατότητες του TDA7718N η οποία τον κάνει να ξεχωρίζει είναι η νέα αρχιτεκτονική Soft Step, χάρη στην οποία ανεπιθύμητοι ήχοι που μοιάζουν με «κλίκ» και προκύπτουν κατά την εναλλαγή μεταξύ διαφορετικών επιπέδων έντασης (λόγω της DC μετατόπισης), σταματάνε να υπάρχουν. Αυτό επιτυγχάνεται με την απαλή μετάβαση μεταξύ των διαφορετικών επιπέδων μέσα σε χρονικά διαστήματα των 5 ή 10 ms. Η λειτουργία Soft Step μπορεί να εφαρμοστεί στην ένταση, την ηχηρότητα, το ηχείο, τα πρίμα, τα μεσαία και τα μπάσα, και ενεργοποιείται είτε κατά την αποστολή I2C δεδομένων, είτε παράλληλα μετά από την αποστολή δεδομένων διαφορετικών στοιχείων, το οποίο επιτρέπει την αποφυγή της αναμονής για την ολοκλήρωση μιας Soft Step διαδικασίας, καθώς εκτελούνται ταυτόχρονα και όχι σε σειρά [7].



Εικόνα 1.10: Δομή ακροδεκτών του DSP TDA7718N [7].

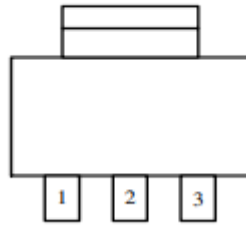
1.5 Ρυθμιστές Τάσης

Λειτουργία ενός ρυθμιστή τάσης αποτελεί η τροποποίηση και σταθεροποίηση μιας τάσης εισόδου σε μια τάση εξόδου, άσχετα από τις αλλαγές που ενδέχεται να προκύψουν σε αυτήν. Οι δύο βασικοί τύποι ρυθμιστών τάσης είναι οι διακοπτικοί (υψηλή απόδοση) και οι γραμμικοί (χαμηλή απόδοση) [8].

1.5.1 Ρυθμιστής Τάσης AMS1117-3.3

Οι γραμμικοί ρυθμιστές τάσης AMS1117, οι οποίοι χρησιμοποιούνται μεταξύ άλλων σε φορτιστές μπαταρίας και όργανα τα οποία λειτουργούν με μπαταρίες, χαρακτηρίζονται από τη δυνατότητα τους να απενεργοποιούνται αυτόματα (εφόσον πιάσουν θερμοκρασίες ανώτερες των 165 βαθμών Κελσίου) και να προσφέρουν χαμηλότερη πτώση τάσης. Το συγκεκριμένο μοντέλο (AMS1117-3.3) έχει είσοδο 4.8 V και έξοδο 3.3 V [9].

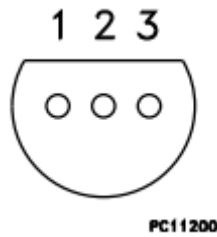
SOT-223 Top View



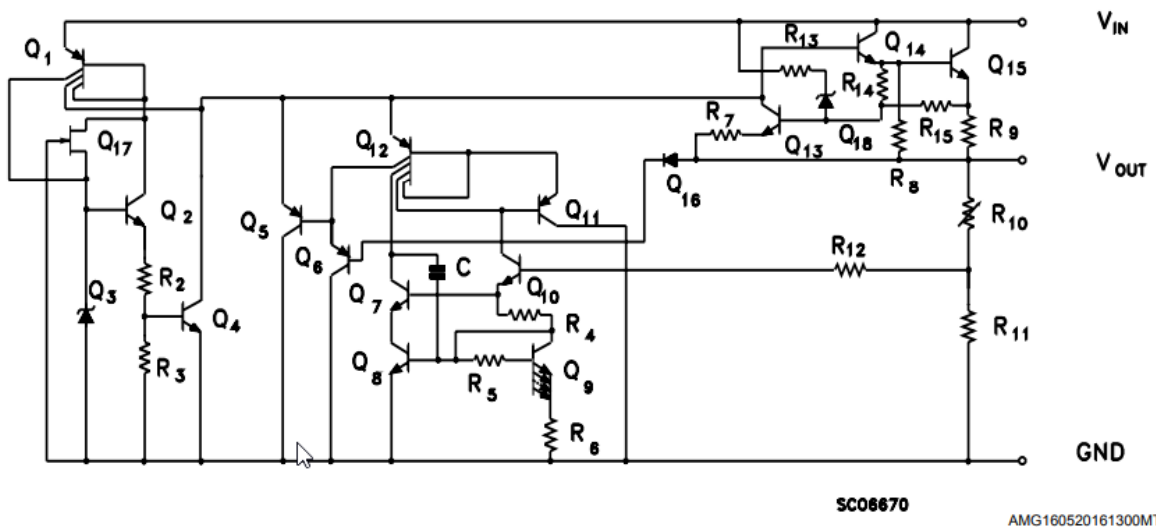
Εικόνα 1.11: Δομή ακροδεκτών του AMS1117-3.3 (1: Γείωση, 2: Έξοδος, 3: Είσοδος) [9].

1.5.2 Ρυθμιστής Τάσης L78L09ACZ

Οι γραμμικοί ρυθμιστές τάσης τριών ακροδεκτών της σειράς L78L αποδίδουν μέχρι και 100 mA ως ρεύμα εξόδου, εφόσον τοποθετηθεί η απαραίτητη ψύκτρα, ενώ παράλληλα περιορίζουν το ρεύμα, απενεργοποιούνται αυτόματα σε υψηλές θερμοκρασίες και προστατεύουν κατά των βραχυκυκλωμάτων. Το συγκεκριμένο μοντέλο (L78L09ACZ) έχει είσοδο 11 έως 30 V και έξοδο 9 V [10].



Εικόνα 1.12: Δομή ακροδεκτών του L78L09ACZ (1: Έξοδος, 2: Γείωση, 3: Είσοδος) [10].

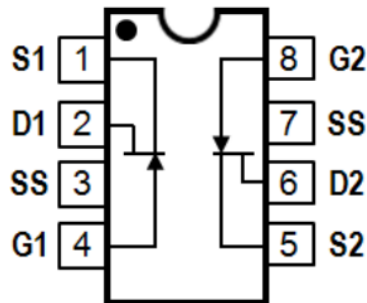


Εικόνα 1.13: Κύκλωμα του ρυθμιστή τάσης L78L09ACZ [10].

1.6 JFET LSK389

Τα Transistor Επαφής Επίδρασης Πεδίου (Junction Field Effect Transistor – JFET) είναι μια κατηγορία μονοπολικών transistor, μικρότερου μεγέθους και σταθερότερης θερμοκρασίας από τα διπολικά transistor. Υπάρχουν δύο τύποι JFET, τα JFET καναλιού-n, των οποίων η κύρια περιοχή αποτελείται από ημιαγωγό τύπου n, ενώ η περιοχή της πύλης αποτελείται από ημιαγωγό τύπου p, και τα JFET καναλιού-p, των οποίων η κύρια περιοχή αποτελείται από ημιαγωγό τύπου p, ενώ η περιοχή της πύλης αποτελείται από ημιαγωγό τύπου n [2].

Το LSK389 είναι ένα εξαιρετικά γνωστό JFET το οποίο χρησιμοποιείται σε ακουστικούς αισθητήρες, μικρόφωνα, κονσόλες μίξης ήχου και, φυσικά, σε ενισχυτές και προενισχυτές ήχου. Χάρη στα ιδιαίτερα χαρακτηριστικά του, το συγκεκριμένο JFET προσφέρει πολύ χαμηλά επίπεδα θορύβου ($1.3\text{nV}/\sqrt{\text{Hz}}$), ενώ υπόσχεται να εξαλείψει τον θόρυβο έκρηξης (burst noise) [11, 12].



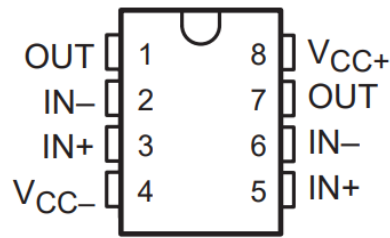
Εικόνα 1.14: Δομή ακροδεκτών του JFET LSK389 [11].

1.7 Τελεστικός Ενισχυτής NE5532PS

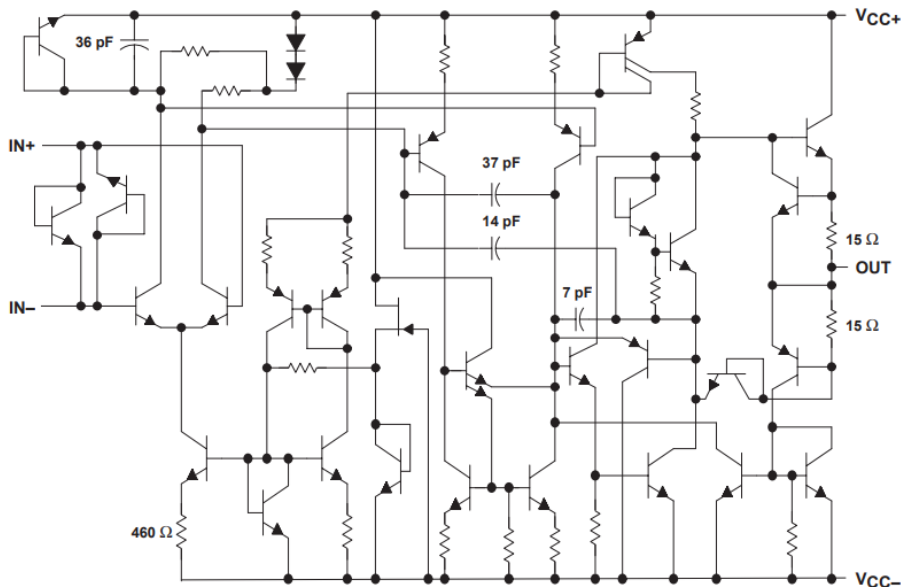
Οι τελεστικοί ενισχυτές είναι μια ειδική ομάδα ενισχυτών η οποία χαρακτηρίζεται από το εξαιρετικά μεγάλο κέρδος που προσφέρει. Ένας τελεστικός ενισχυτής μπορεί, με την κατάλληλη διαμόρφωση να αξιοποιηθεί τόσο για ενίσχυση, όσο και για ολοκλήρωση, διαφόριση, αφαίρεση και πρόσθεση. Αποτελείται από δύο εισόδους (μία αναστρέφουσα και μία μη αναστρέφουσα) και μία έξοδο [13].

Μεταξύ άλλων, ο τελεστικός ενισχυτής NE5532PS, ο οποίος χρησιμοποιήθηκε στην παρούσα εργασία, έχει τα εξής χαρακτηριστικά:

- Πολύ χαμηλό θόρυβο,
- Υψηλό ρυθμό μεταβολής,
- Χαμηλή παραμόρφωση,
- Προστασία κατά πιθανού βραχυκυκλώματος στην έξοδο,
- Λειτουργία μεταξύ 0 και 70 βαθμών Κελσίου,
- Προτεινόμενη τάση τροφοδοσίας μεταξύ 5 και 15 V [14].



Εικόνα 1.15: Δομή ακροδεκτών του τελεστικού ενισχυτή NE5532PS [14].



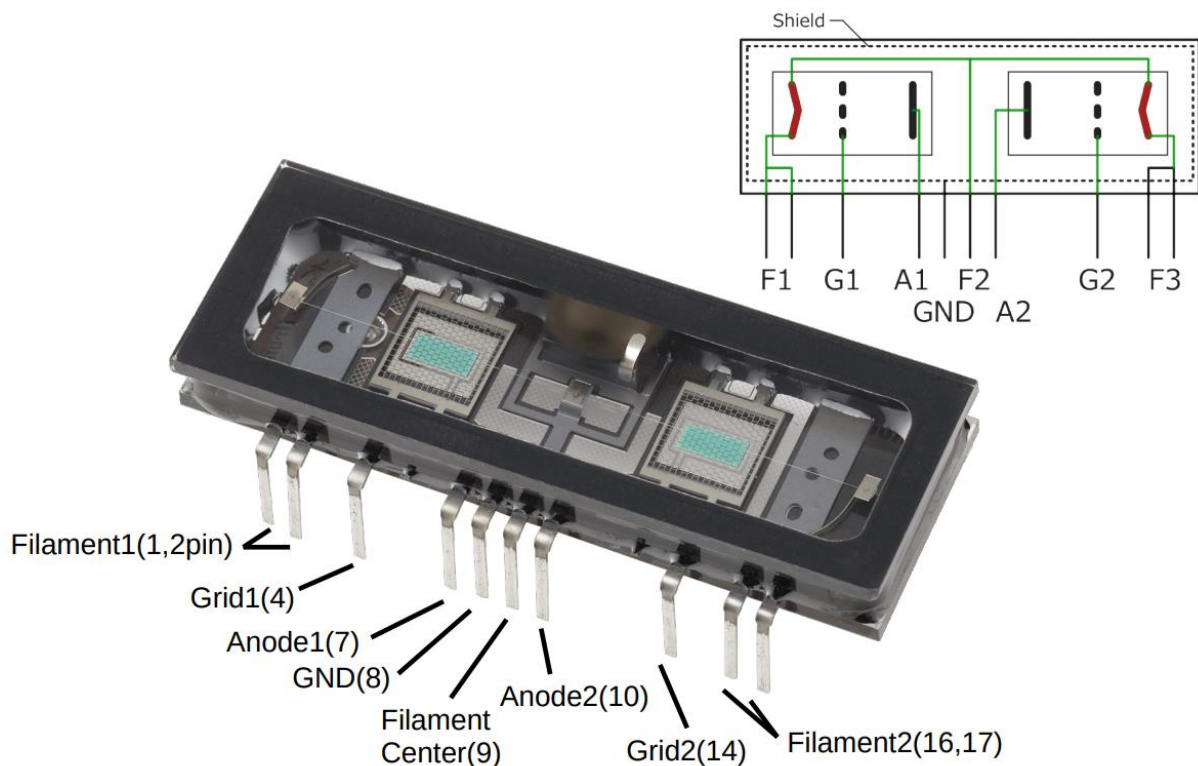
Εικόνα 1.16: Κύκλωμα του τελεστικού ενισχυτή NE5532PS [14].

1.8 Λυχνία Nutube

Μέχρι τη δεκαετία του 1950, οι ηλεκτρονικές λυχνίες, γνωστές και ως λυχνίες κενού, αποτέλεσαν σημαντικό εξάρτημα τηλεοράσεων, ραδιοφώνων, τηλεφώνων και υπολογιστών. Το 1904, η πρώτη λυχνία κενού (Δίοδος, Diode), γνωστή και ως "Fleming Valve", από τον εφευρέτη της, John Ambrose Fleming, χρησιμοποιήθηκε για την ανίχνευση ασύρματων σημάτων. Το 1906 ακολούθησε η Τρίοδος (Triode), γνωστή και ως "Audion", η οποία μπόρεσε να ενισχύσει το σήμα, ενώ το 1926 κατασκευάστηκε η Τέτροδος (Tetrode), η οποία είχε μεγαλύτερη αντίσταση εισόδου και συντελεστή ενίσχυσης, αλλά έριξε την ποιότητα του σήματος. Την ίδια χρονιά (1926), κατασκευάστηκε η Πέντοδος (pentode), η οποία μπόρεσε να ενισχύσει την ισχύ δίχως παραμόρφωση. Εξαιτίας της εισαγωγής των ημιαγωγών στην αγορά, οι λυχνίες κενού χρησιμοποιούνται έκτοτε κυρίως σε συσκευές ήχου, μικροκυμάτων και στρατιωτικά συστήματα. Λόγω της προτίμησης των μουσικών για τον «θερμό» ήχο τους, πολλοί από τους σύγχρονους ενισχυτές εμπεριέχουν λυχνίες κενού [15].

Οι ιαπωνικές εταιρείες Korg και Noritake Itron κυκλοφόρησαν έναν νέο τύπο λυχνίας το 2016 με τίτλο Nutube 6P1 [15]. Η διπλή τρίοδος Nutube χαρακτηρίζεται από το σχέδιο της, καθώς δανείζεται στοιχεία από τις Φθορίζουσες Μονάδες Απεικόνισης Κενού (Vacuum Fluorescent Displays, VFDs), λόγω των δύο μεγάλων μπλε-πράσινων φωτεινών τετραγώνων που έχει στο εσωτερικό της. [16, 17]. Εν συγκρίσει

με τις παλαιότερες λυχνίες, η λυχνία Nutube έχει πολλά πλεονεκτήματα, όπως μεγαλύτερη διάρκεια ζωής και μικρότερο μέγεθος, ενώ διατηρεί τα θετικά τους, όπως τον γνωστό «θερμό» ήχο τους [15].



Εικόνα 1.17: Σχήμα της λυχνίας Nutube και δομή των ακροδεκτών της [17].

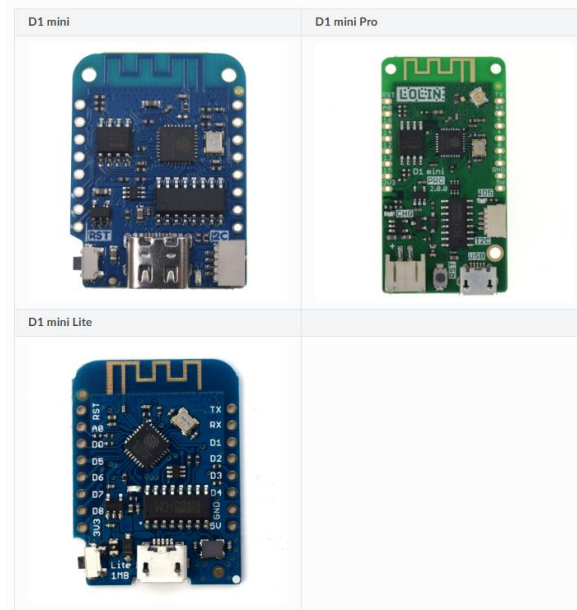
Μερικά από τα χαρακτηριστικά της λυχνίας Nutube έχουν ως εξής:

- Πλάκα από την οποία βγαίνει λάμψη,
- Διάρκεια ζωής 30.000 ωρών,
- Τροφοδοσία χαμηλής τάσης,
- Χαμηλή κατανάλωση ενέργειας,
- Ενσωματωμένος έλεγχος έντασης ήχου,
- Εξάλειψη του θορύβου ενεργοποίησης και απενεργοποίησης [16].

Όσον αφορά τη δομή της λυχνίας, κάθε τρίοδος αποτελείται από μία άνοδο, ένα πλέγμα και ένα νήμα, τα οποία περιβάλλονται από αεροστεγές γυαλί μέσα σε κενό αέρος [17].

1.9 WEMOS D1 mini Clone με ESP8266

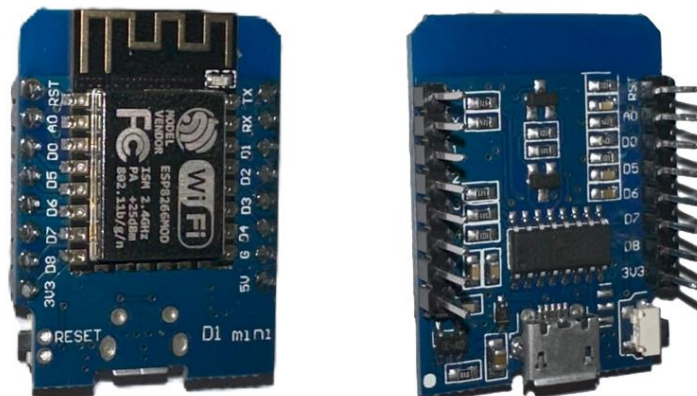
Τα WEMOS είναι μια δημοφιλής σειρά από μικρές πλακέτες οι οποίες έχουν τη δυνατότητα σύνδεσης σε δίκτυο WiFi. Το WEMOS D1 mini (πλέον γνωστό ως LOLIN D1 mini) αποτελεί εξαιρετική επιλογή για συστήματα τα οποία απαιτούν έλεγχο μέσω διαδικτύου, καθώς είναι βασισμένα στον μικροελεγκτή ESP-8266EX της Espressif Systems, ο οποίος παρέχει ποικιλία από λύσεις για τις ανάγκες των χρηστών του διαδικτύου των πραγμάτων [18, 19].



Εικόνα 1.18: Οι πλακέτες της σειράς D1 από τη WEMOS [20].

Στη συγκεκριμένη εργασία χρησιμοποιήθηκε μια πλακέτα η οποία αποτελεί κλώνο του WEMOS D1 mini. Ειδικότερα, ο κλώνος:

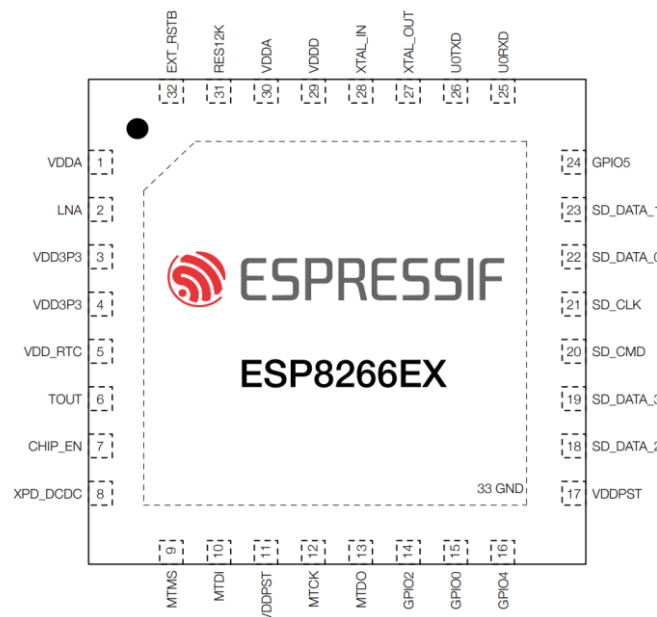
- Έχει ενσωματωμένη μία μονάδα WiFi ESP8266MOD η οποία εμπεριέχει τον μικροελεγκτή ESP8266EX,
- Έχει 4MB flash μνήμη για την αποθήκευση του κώδικα,
- Έχει μία micro-USB θύρα η οποία χρησιμοποιείται για το ανέβασμα του κώδικα στον μικροελεγκτή ESP8266,
- Έχει διακόπτη επανεκκίνησης,
- Έχει LED ανατροφοδότησης,
- Έχει 11 ψηφιακά pin (π.χ. D1 = Ακροδέκτης σειριακού ρολογιού I2C - SCL και D2 = Ακροδέκτης σειριακών δεδομένων I2C - SDA),
- Επιτρέπει τη σύνδεση σε δίκτυα των 2.4 GHz,
- Είναι συμβατός με το πρόγραμμα προγραμματισμού Arduino IDE,
- Δέχεται τροφοδοσία 3.3V, ή 5V την οποία μετατρέπει σε 3.3V [18, 21].



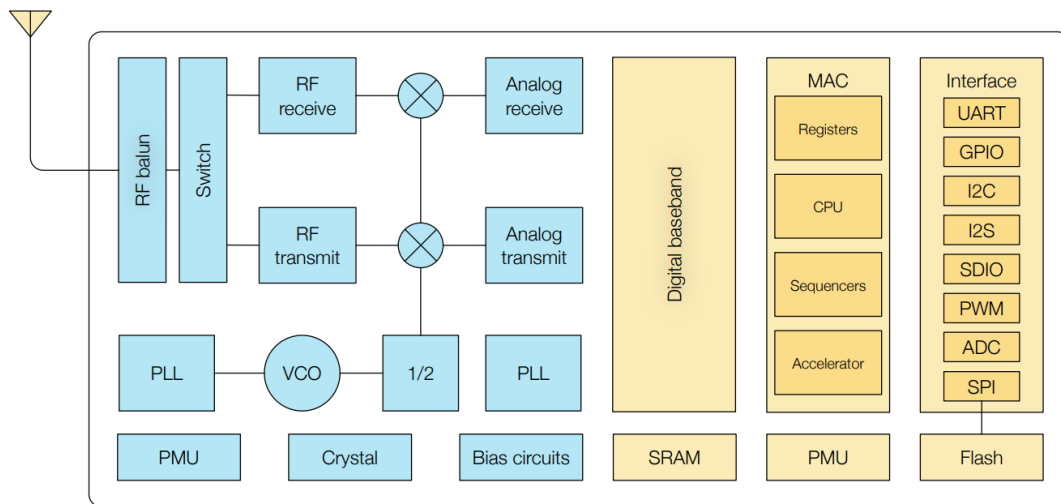
Εικόνα 1.19: Ο κλώνος του WEMOS D1 mini που χρησιμοποιήθηκε στην εργασία.

Ο ESP8266EX της Espressif Systems αποτελεί αξιόπιστη επιλογή στη βιομηχανία του Διαδικτύου των Πραγμάτων, χάρη στις μοναδικές του δυνατότητες και τις πολλαπλές του εφαρμογές. Ο συγκεκριμένος μικροελεγκτής μπορεί να χρησιμοποιηθεί, μεταξύ άλλων, σε διάφορους οικιακούς αυτοματισμούς, κάμερες IP, φορητές ηλεκτρονικές συσκευές και συσκευές Wi-Fi με επίγνωση θέσης. Μερικά από τα βασικά χαρακτηριστικά ενός ESP8266EX έχουν ως εξής [22]:

- Κεντρική Μονάδα Επεξεργασίας (CPU) – Έχει ενσωματωμένο έναν επεξεργαστή RISC 32-bit Tensilica L106, ο οποίος πιάνει μέγιστη ταχύτητα ρολογιού τα 160 MHz και καταναλώνει μικρά ποσά ενέργειας,
- Μνήμη – Έχει ενσωματωμένους ελεγκτές και μονάδες μνήμης, με μέγιστο διαθέσιμο χώρο RAM τα 50 kB,
- Εξωτερικό Flash – Μπορεί να υποστηρίξει θεωρητικά έως 16MB μνήμη για την αποθήκευση του κώδικα του χρήστη,
- Ρολόι Υψηλής Συχνότητας – Χρησιμοποιείται για την καθοδήγηση των μικτών αποστολής και λήψης και έχει συχνότητα κρυστάλλου από 24 έως 52 MHz,
- Ράδιο – Εμπεριέχει έναν συνδυασμό δέκτη / πομπού στα 2.4 GHz,
- WiFi – Ακολουθώντας τα πρωτόκολλα TCP/IP και 802.11 b/g/n WLAN MAC, υποστηρίζει τις λειτουργίες STA (Σταθμός, δηλ. λειτουργεί ως client ο οποίος συνδέεται σε ένα ασύρματο σημείο πρόσβασης) και SoftAP (Σημείο Πρόσβασης Λογισμικού, δηλ. λειτουργεί ο ίδιος ως ασύρματο σημείο πρόσβασης),
- Διεπαφή I2C – Χρησιμοποιείται για τη διασύνδεση του μικροελεγκτή με τα περιφερειακά εξαρτήματα του συστήματος [22].



Εικόνα 1.20: Δομή των ακροδεκτών του ESP8266EX [22].



Εικόνα 1.21: Μπλοκ διάγραμμα του ESP8266EX [22].

1.10 EasyEDA

Το EasyEDA είναι μια ιστοσελίδα η οποία μπορεί να αξιοποιηθεί από ηλεκτρονικούς μηχανικούς για τον σχεδιασμό ηλεκτρονικών κυκλωμάτων & PCB. Επιτρέπει τη συνεργασία μεταξύ των ατόμων μιας ομάδας επάνω σε ένα project σε πραγματικό χρόνο και από οποιαδήποτε συσκευή, ενώ παρέχει διάφορους τύπους εξαγωγής δεδομένων, όπως είναι τα PDF και τα BOM [23, 24]. Το EasyEDA χρησιμοποιήθηκε για την κατασκευή του ηλεκτρονικού κυκλώματος και του PCB της εργασίας.



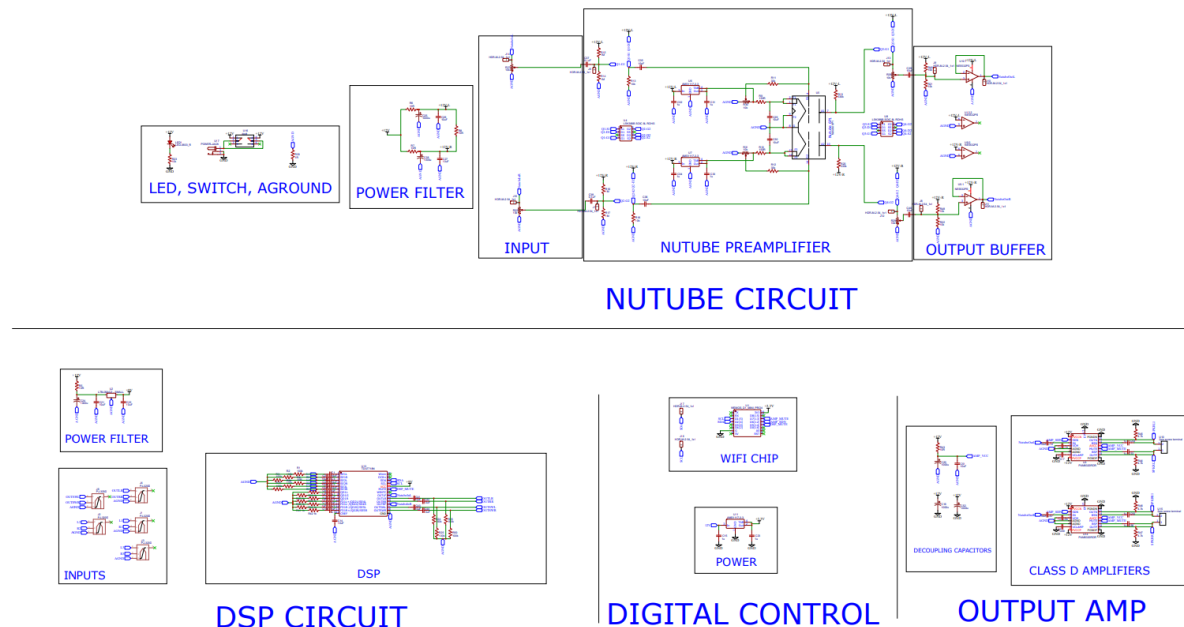
Εικόνα 1.22: Το επίσημο logo του EasyEDA [23].

1.11 Ανάλυση του Κυκλώματος της Εργασίας

Στις επόμενες υποενότητες ακολουθεί η ανάλυση του κυκλώματος της εργασίας. Το κύκλωμα της εργασίας συνδυάζει όλα τα εξαρτήματα που αναφέρθηκαν έως τώρα, με στόχο να κατασκευάσει ένα πλήρως λειτουργικό στερεοφωνικό ενισχυτή με λυχνίες Nutube. Για την καλύτερη ανάλυση του κυκλώματος, έχει γίνει ο αυθαίρετος διαχωρισμός του σε 4 σύνολα:

- Κύκλωμα ψηφιακού ενισχυτή σήματος (DSP),
- Κύκλωμα Nutube,
- Ενισχυτές εξόδου τάξης D,
- Ψηφιακός έλεγχος.

Η εξήγηση του κυκλώματος ακολουθεί, επί το πλείστον, την πορεία του σήματος. Το σύνολο με τίτλο «Κύκλωμα Nutube» βασίζεται στο κύκλωμα που εμπεριέχεται στις σημειώσεις εφαρμογής του Nutube. Για τη σχεδίαση του κυκλώματος χρησιμοποιήθηκε η ιστοσελίδα EasyEDA.

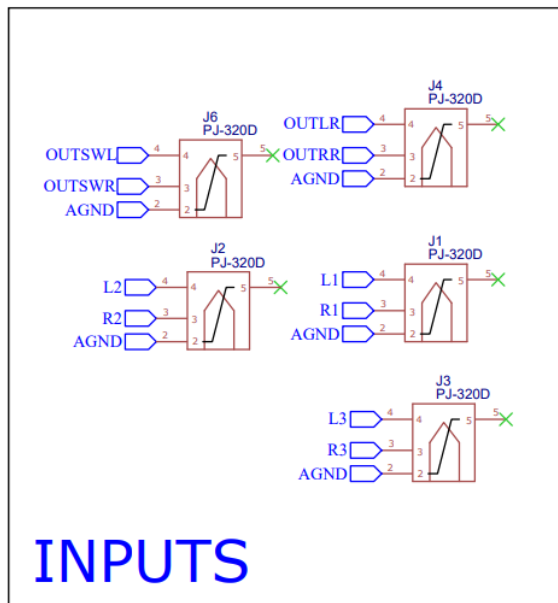
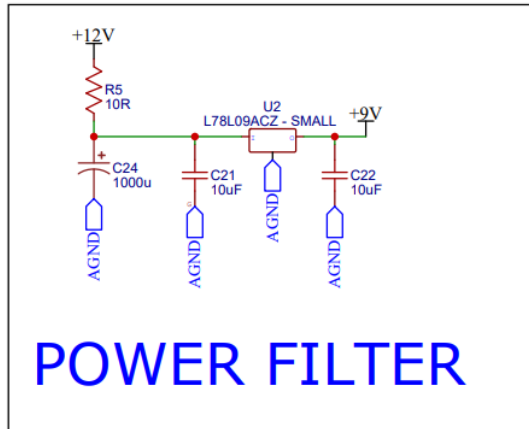


Εικόνα 1.23: Το κύκλωμα της εργασίας.

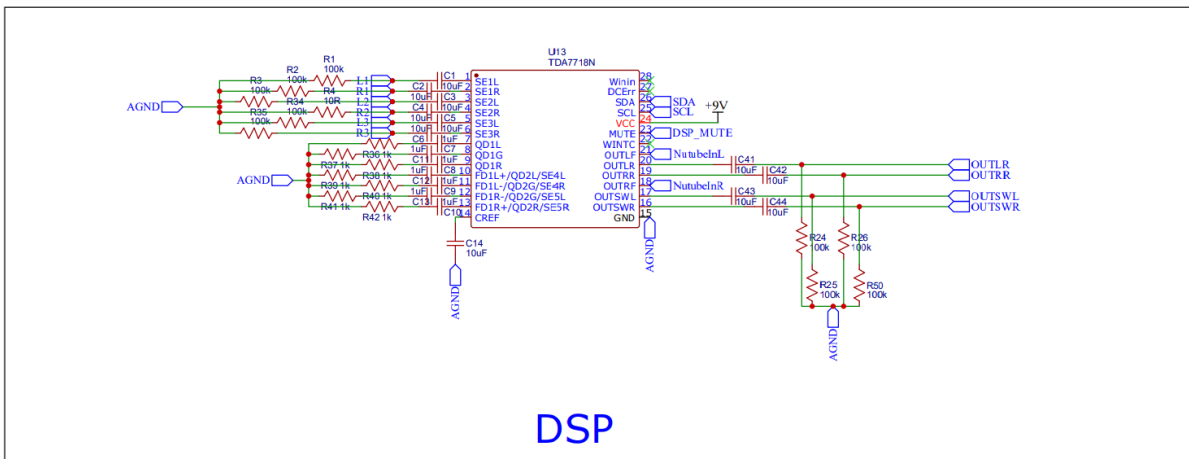
1.11.1 Κύκλωμα Ψηφιακού Επεξεργαστή Σήματος (DSP)

Η πορεία του σήματος στο κύκλωμα ξεκινάει με την εισαγωγή καλωδίου 3.5mm jack (δηλ. αναλογικό σήμα) στα βύσματα ήχου PJ-320D. Από τα βύσματα J1 έως J6, μόνο το J1 χρησιμοποιείται στην τωρινή εκδοχή της εργασίας. Το σήμα συνεχίζει μέσω των ακροδεκτών L1 / R1 και εισέρχεται στον ψηφιακό επεξεργαστή σήματος TDA7718N στους αντίστοιχους ακροδέκτες (SE1L, SE1R). Ο ψηφιακός επεξεργαστής σήματος συνδέεται με τον μικροελεγκτή, ο οποίος αποτελεί κλώνο του WEMOS D1 mini με ενσωματωμένο ESP8266, μέσω των ακροδεκτών SDA (Ακροδέκτης Σειριακών Δεδομένων) και SCL (Ακροδέκτης Σειριακού Ρολογιού), οι οποίοι χρησιμοποιούνται για την I2C αποστολή και λήψη δεδομένων, καθώς και του ακροδέκτη Mute. Οι έξοδοι του ψηφιακού επεξεργαστή σήματος (OUTLF, OUTRF) είναι οι εισοδοί στο κύκλωμα του προενισχυτή Nutube.

Για την ορθή λειτουργία του ψηφιακού επεξεργαστή σήματος, είναι αναγκαία η χρήση ενός φίλτρου ισχύος με ρυθμιστή τάσης L78L09ACZ, ο οποίος κατεβάζει και σταθεροποιεί τα +12 V σε +9 V.



Εικόνα 1.24: Φίλτρο ισχύος +12 V σε +9 V και βύσματα ήχου.



Εικόνα 1.25: Ψηφιακός επεξεργαστής σήματος (DSP).

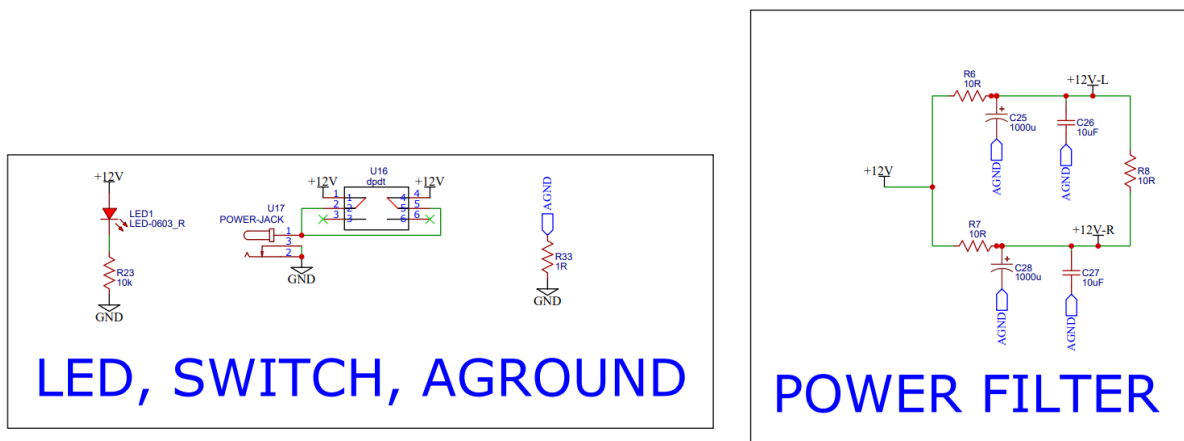
1.11.2 Κύκλωμα Nutube

Το σήμα εισέρχεται στο κύκλωμα του προενισχυτή Nutube από τις εξόδους του ψηφιακού επεξεργαστή σήματος. Θα ακολουθήσουμε την πορεία του από το αριστερό κανάλι, η οποία ταυτίζεται με αυτήν από το δεξί κανάλι – συνεπώς όλα τα εξαρτήματα πλην του Nutube και των JFET εμφανίζονται δύο φορές.

Αφού περάσει από ένα ποτενσιόμετρο, το οποίο χρησιμοποιείται για καλύτερο έλεγχο της τάσης (σε περίπτωση που έρθει μεγαλύτερη τάση από τον ψηφιακό επεξεργαστή σήματος), και από έναν διαιρέτη τάσης, το σήμα καταλήγει στον Vin ακροδέκτη του LSK389B JFET, το οποίο σταθεροποιεί το ρεύμα και μειώνει τον θόρυβο. Από εκεί, το σήμα εισέρχεται στον ακροδέκτη πλέγματος (grid) της λυχνίας Nutube. Για τη θέρμανση του νήματος (filament) στη λυχνία Nutube χρησιμοποιείται ένα φίλτρο ισχύος με ρυθμιστή τάσης AMS1117-3.3, ο οποίος κατεβάζει και σταθεροποιεί τα +12 V σε +3.3 V. Το σήμα συνεχίζει από τους ακροδέκτες ανόδου (anode) και καταλήγει πάλι σε ένα LSK389B JFET. Το κύκλωμα του προενισχυτή Nutube ολοκληρώνεται με το σήμα να εισέρχεται σε ένα buffer εξόδου, το οποίο το διατηρεί σταθερό με στόχο να αποφευχθούν παραμορφώσεις και προσθήκες θορύβου. Για το συγκεκριμένο buffer χρησιμοποιείται ο τελεστικός ενισχυτής NE5532PS. Στη συγκεκριμένη εκδοχή της εργασίας, οι τελεστικοί ενισχυτές NE5532PS έχουν κομμένη την είσοδο στους ακροδέκτες 6 (IN-), διότι το κύκλωμα υπολειπούν.

Το φίλτρο ισχύος, το οποίο διαχωρίζει τα +12 V σε +12 V αριστερά και +12 V δεξιά, έχει ως στόχο την μείωση του θορύβου από την είσοδο της τροφοδοσίας. Αξίζει επίσης να αναφερθούν:

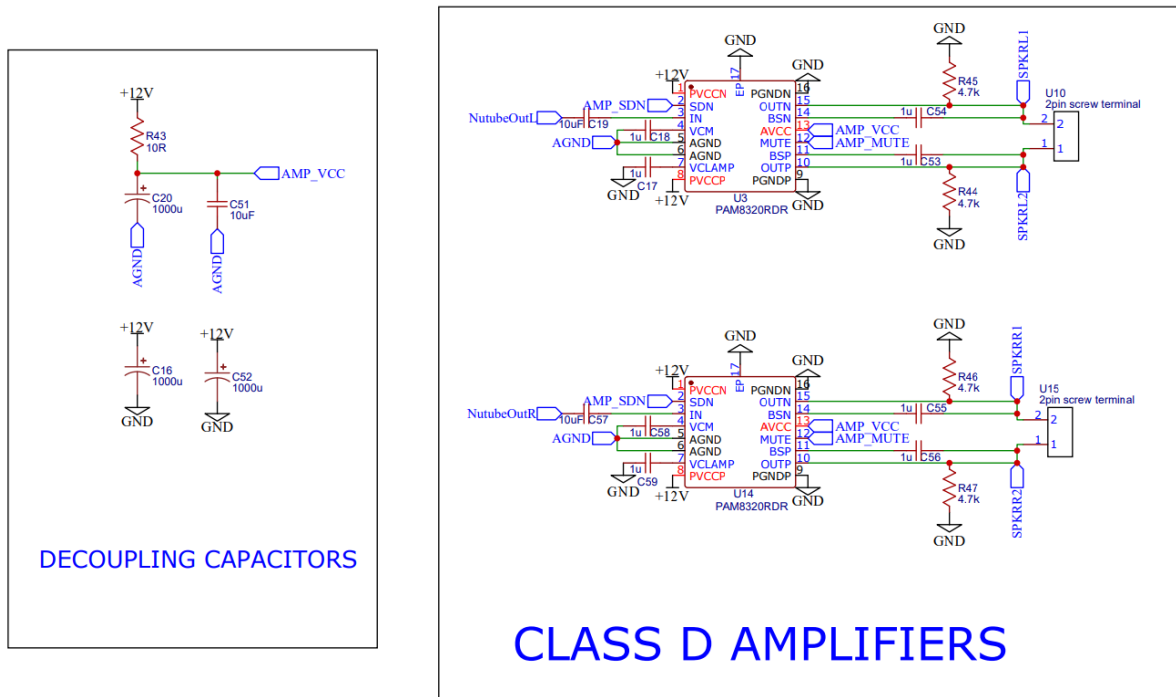
- Το LED χρησιμοποιείται ως ένδειξη ότι ο ενισχυτής βρίσκεται σε λειτουργία.
- Το εξάρτημα με ονομασία U16 στο κύκλωμα είναι ο διακόπτης λειτουργίας.
- Το aground είναι η αναλογική γείωση, η οποία χρησιμοποιείται για να μην περάσει θόρυβος από τα ψηφιακά στα αναλογικά κυκλώματα.



Εικόνα 1.26: LED, διακόπτης ενεργοποίησης, αναλογική γείωση και φίλτρο ισχύος για τη μείωση του θορύβου και τον διαχωρισμό +12 V σε αριστερά και δεξιά.

1.11.3 Ενισχυτές Εξόδου Τάξης D

Κατά την έξοδο του από τους τελεστικούς ενισχυτές NE5532PS, το σήμα εισέρχεται στους ενισχυτές εξόδου τάξης D PAM8320RDR. Οι συγκεκριμένοι ενισχυτές είναι συνδεδεμένοι στους ακροδέκτες SDN, AVCC και MUTE με τρεις αντίστοιχους ακροδέκτες του κλώνου του WEMOS D1 mini. Κατά την έξοδο του από τον ενισχυτή εξόδου, το σήμα καταλήγει στο εξάρτημα με ονομασία U10 / U15, όπου συνδέονται τα ηχεία. Για την ορθή λειτουργία των ενισχυτών εξόδου είναι απαραίτητη η χρήση πυκνωτών αποσύζευξης, προκειμένου να μειωθεί περαιτέρω ο θόρυβος.



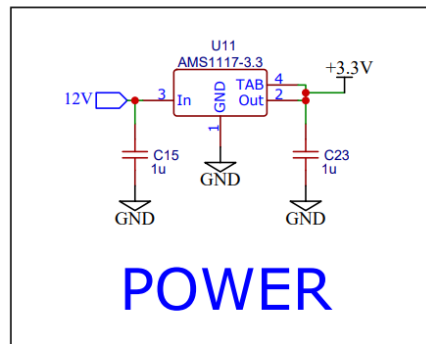
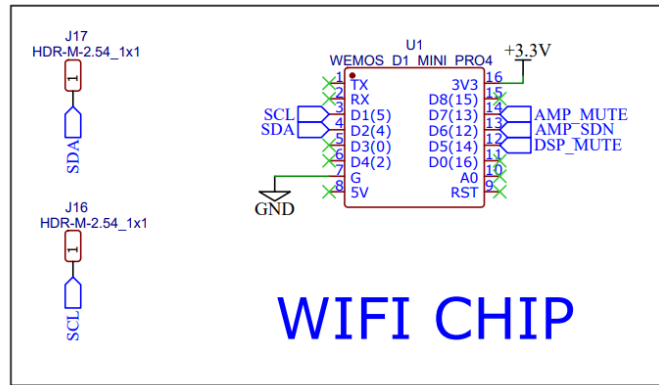
Εικόνα 1.28: Πυκνωτές αποσύζευξης και ενισχυτές εξόδου τάξης D.

1.11.4 Ψηφιακός Έλεγχος

Όπως έχει ήδη αναφερθεί, ο κλώνος του WEMOS D1 mini με ενσωματωμένο ESP8266 αποτελεί ένα εξαιρετικά σημαντικό εξάρτημα καθώς μέσω αυτού γίνεται ο έλεγχος του ενισχυτή. Ο συγκεκριμένος μικροελεγκτής είναι συνδεδεμένος στο κύκλωμα με τους εξής ακροδέκτες:

- SCL (D1) - Ακροδέκτης σειριακού ρολογιού, καταλήγει στον DSP,
- SDA (D2) - Ακροδέκτης σειριακών δεδομένων, καταλήγει στον DSP,
- AMP_MUTE (D7) – Ακροδέκτης σίγασης ενισχυτή, καταλήγει στον ενισχυτή εξόδου,
- AMP_SDN (D6) – Ακροδέκτης απενεργοποίησης ενισχυτή, καταλήγει στον ενισχυτή εξόδου,
- DSP_MUTE (D5) – Ακροδέκτης σίγασης DSP, καταλήγει στον DSP.

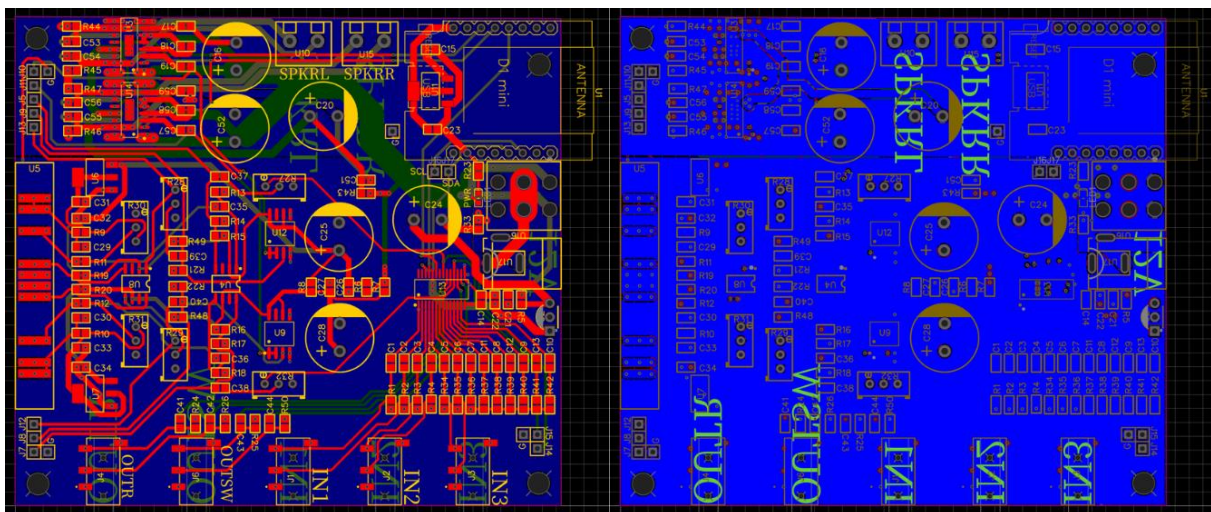
Για την ορθή λειτουργία του μικροελεγκτή είναι αναγκαία η χρήση ενός φίλτρου ισχύος με ρυθμιστή τάσης AMS1117-3.3, ο οποίος κατεβάζει και σταθεροποιεί τα +12 V σε +3.3 V. Λόγω λάθους στο σχηματικό (12 V έναντι +12 V), το JFET δεν λάμβανε τροφοδοσία, οπότε χρειάστηκε να συνδεθεί ένα καλώδιο από την αντίσταση R43 με +12 V.



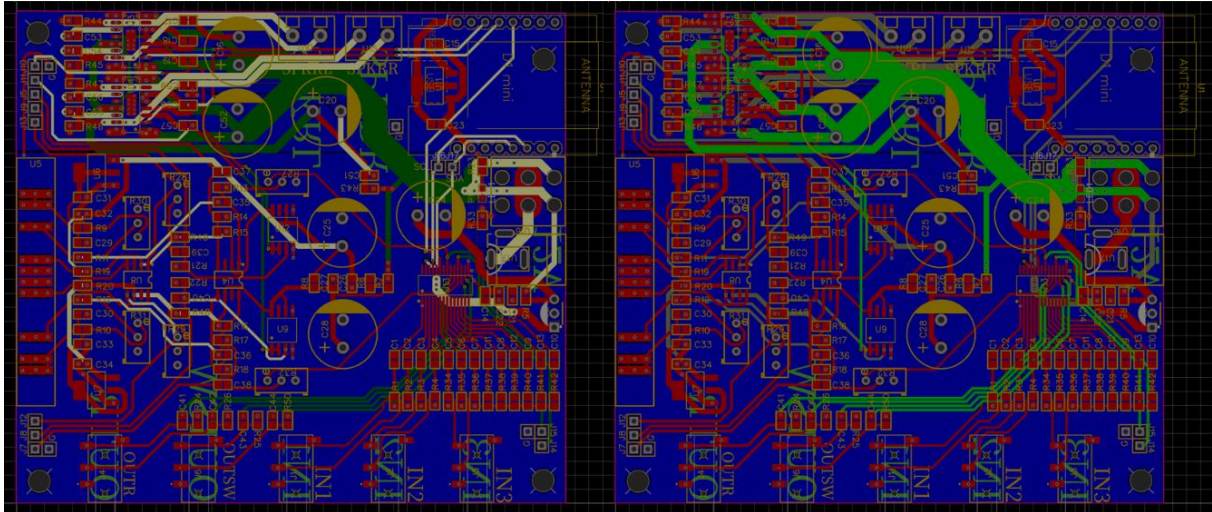
Εικόνα 1.29: WiFi μικροελεγκτής και φίλτρο ισχύος +12 V σε +3.3 V.

1.12 PCB

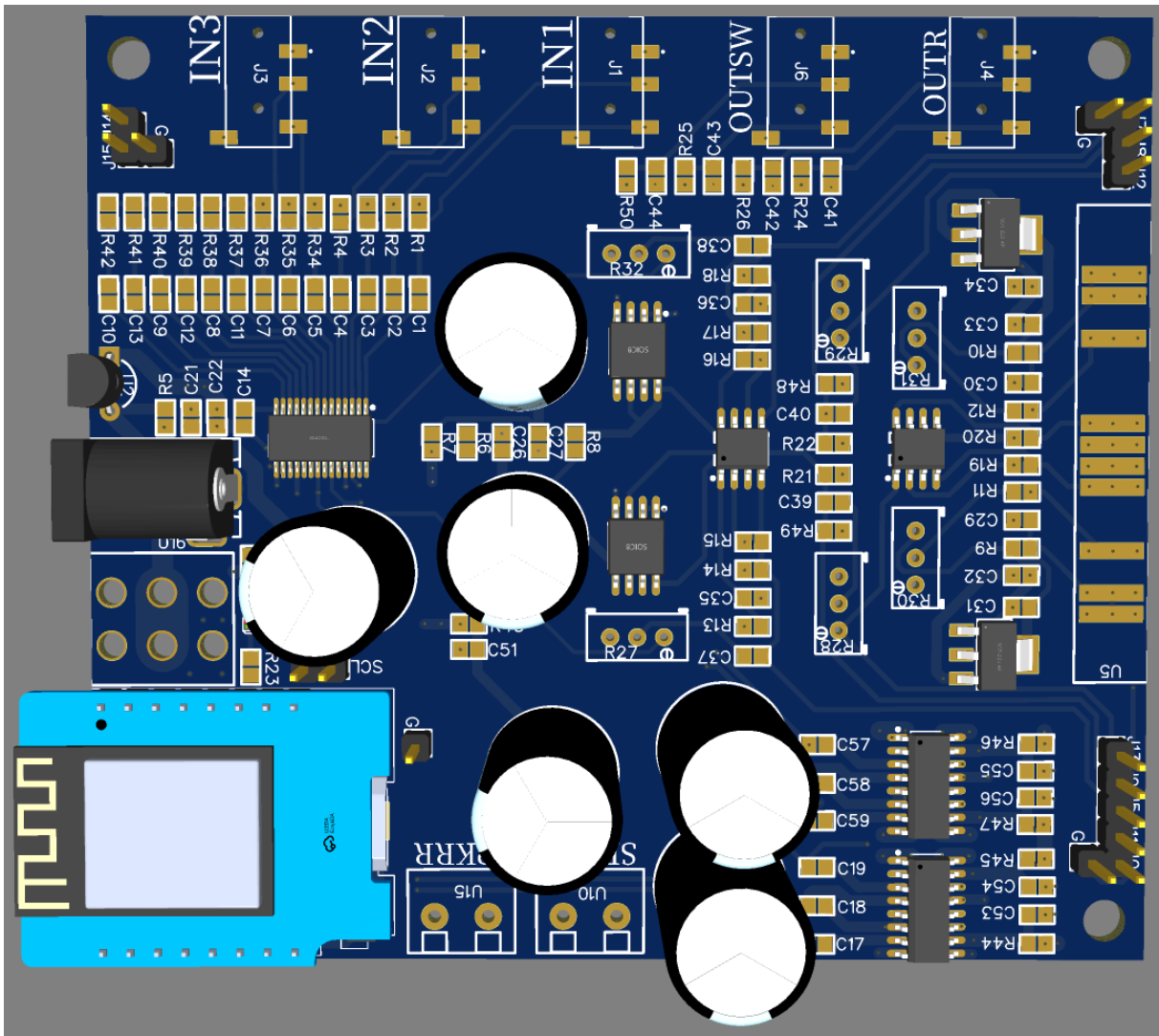
Στις επόμενες εικόνες παρουσιάζεται το PCB που υλοποιήθηκε με τη χρήση του EasyEDA.



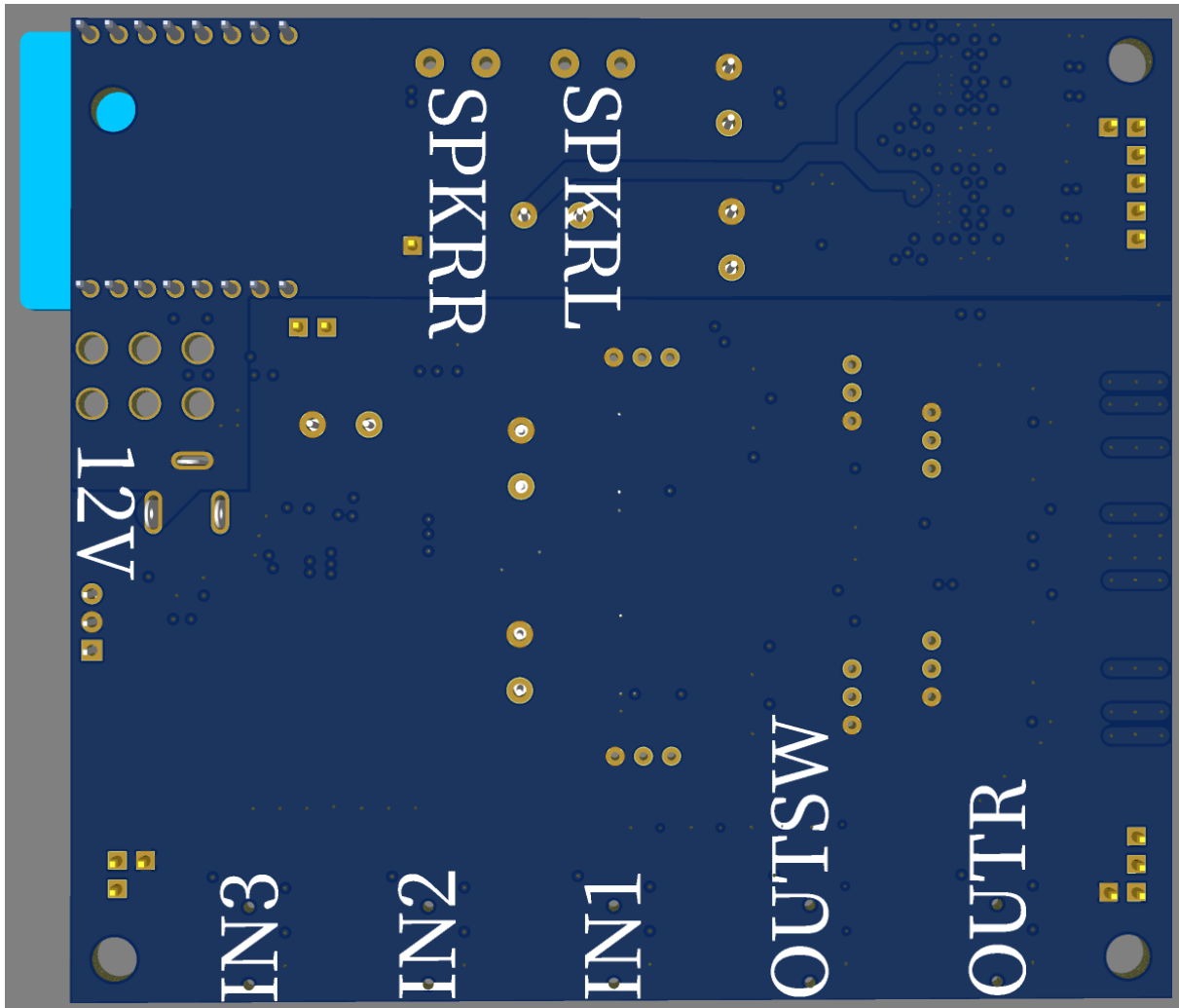
Εικόνα 1.30: Τα δύο εξωτερικά επίπεδα του PCB.



Εικόνα 1.31: Τα δύο εσωτερικά επίπεδα του PCB.



Εικόνα 1.32: Τρισδιάστατη απεικόνιση του επάνω μέρους της πλακέτας.



Εικόνα 1.33: Τρισδιάστατη απεικόνιση του κάτω μέρους της πλακέτας.

1.13 Επίλογος

Στο συγκεκριμένο κεφάλαιο ολοκληρώθηκε η παρουσίαση του υλικού μέρους της εργασίας. Έγινε ανάλυση των διάφορων τύπων ενισχυτών, των εξαρτημάτων που επιλέχθηκαν για την υλοποίηση του συστήματος, καθώς και του κυκλώματος που κατασκευάστηκε για τον ενισχυτή με λυχνίες Nutube.

Στο επόμενο κεφάλαιο ακολουθεί η παρουσίαση του λογισμικού μέρους της εργασίας, όπου θα γίνει η ανάλυση των βιβλιοθηκών, των γλωσσών σήμανσης / προγραμματισμού, καθώς και του κώδικα της εργασίας.

Κεφάλαιο 2ο: Λογισμικό (Software)

2.1 Εισαγωγή

Στο δεύτερο κεφάλαιο της εργασίας γίνεται η παρουσίαση του λογισμικού που αναπτύχθηκε για την ορθή λειτουργία και τον έλεγχο του συστήματος. Στις πρώτες υποενότητες του κεφαλαίου γίνεται μια σύντομη περιγραφή του προγράμματος Arduino IDE, του πρωτοκόλλου I2C, των βιβλιοθηκών Wire, ESP8266WiFi, ESPAsyncWebServer και ESPAsyncTCP, καθώς και των γλωσσών σήμανσης / προγραμματισμού Arduino, HTML5, CSS3 και JavaScript. Παράλληλα, εξηγείται ο τρόπος λειτουργίας των συνθηκών, μεθόδων, ιδιοτήτων, κατασκευαστών, τελεστών και στοιχείων των παραπάνω γλωσσών σήμανσης / προγραμματισμού που χρησιμοποιήθηκαν στον κώδικα της εργασίας, ενώ γίνεται μία σύντομη αναφορά σε όσα δεν χρησιμοποιήθηκαν στην παρούσα εκδοχή της εργασίας, αλλά θα μπορούσαν να αξιοποιηθούν σε μια βελτιωμένη εκδοχή της.

Την θεωρητική παρουσίαση των βιβλιοθηκών, των γλωσσών και των περιεχομένων τους, ακολουθεί μια σε βάθος ανάλυση του κώδικα που αναπτύχθηκε για την ιστοσελίδα. Ο συγκεκριμένος κώδικας, ο οποίος συνδυάζει διαφορετικές γλώσσες και, τόσο έτοιμες, όσο και προσαρμοσμένες συναρτήσεις, χρησιμοποιείται για να αρχικοποιήσει και για να εφαρμόσει ρυθμίσεις στον ενισχυτή και στον ψηφιακό επεξεργαστή σήματος, καθώς και για να κατασκευάσει και να σερβίρει, σε μία στατική IP, μια ολοκληρωμένη ιστοσελίδα, η οποία εμπεριέχει το χειριστήριο με το οποίο ο χρήστης μπορεί να διαχειριστεί την αυξομείωση του ποσοστού της έντασης του ήχου και των decibels από τα πρίμα, τα μεσαία και τα μπάσα. Στο συγκεκριμένο χειριστήριο ο χρήστης αποκτάει πρόσβαση πληκτρολογώντας την στατική IP στο πρόγραμμα περιήγησης μίας συσκευής η οποία είναι συνδεδεμένη στο ίδιο δίκτυο με το σύστημα του ενισχυτή.

Η ανάλυση του κώδικα συμπεριλαμβάνει εξηγήσεις για τις σταθερές και τις μεταβλητές που αρχικοποιούνται κατά τη ρύθμιση του συστήματος, καθώς και για τις προσαρμοσμένες συναρτήσεις που κατασκευάστηκαν από τους συγγραφείς της εργασίας (writeToDSP, volumeMapping, dBMapping, setupDSP, replaceAll, getHTML), αλλά και τον τρόπο με τον οποίο αξιοποιήθηκαν οι έτοιμες συναρτήσεις των γλωσσών.

2.2 Arduino & Arduino IDE






Το Arduino είναι μια γνωστή πλατφόρμα ανοικτού κώδικα, η οποία παρέχει τόσο υλικό όσο και λογισμικό, επιτρέποντας στους χρήστες να εισάγουν δεδομένα - όπως την ανίχνευση του πατήματος ενός κουμπιού ή της λήψης ηλεκτρονικών μηνυμάτων - και να μετατρέψουν αυτά τα δεδομένα σε εξόδους - όπως την ενεργοποίηση LED ή την αποστολή ψηφιακού περιεχομένου. Χάρη στους τέσσερις ιδρυτές του (Massimo Banzi, David Cuartielles, Tom Igoe, David Mellis [25]) και την ομάδα τους, το Arduino έχει καταλήξει να χρησιμοποιείται σήμερα σε χιλιάδες εφαρμογές, τόσο από καθηγητές όσο και από μαθητές, σε διάφορους εκπαιδευτικούς τομείς, συμπεριλαμβανομένων αυτών της ρομποτικής και της πληροφορικής (ή, πιο συγκεκριμένα, του προγραμματισμού). Για τον προγραμματισμό της συγκεκριμένης πλατφόρμας έχει δημιουργηθεί η γλώσσα προγραμματισμού Arduino (βασισμένη στη Wiring - ένα framework προγραμματισμού ανοιχτού κώδικα για μικροελεγκτές [26]) και κατασκευάστηκε το πρόγραμμα Arduino IDE [27].



Εικόνα 2.1: Το επίσημο logo του Arduino [28].

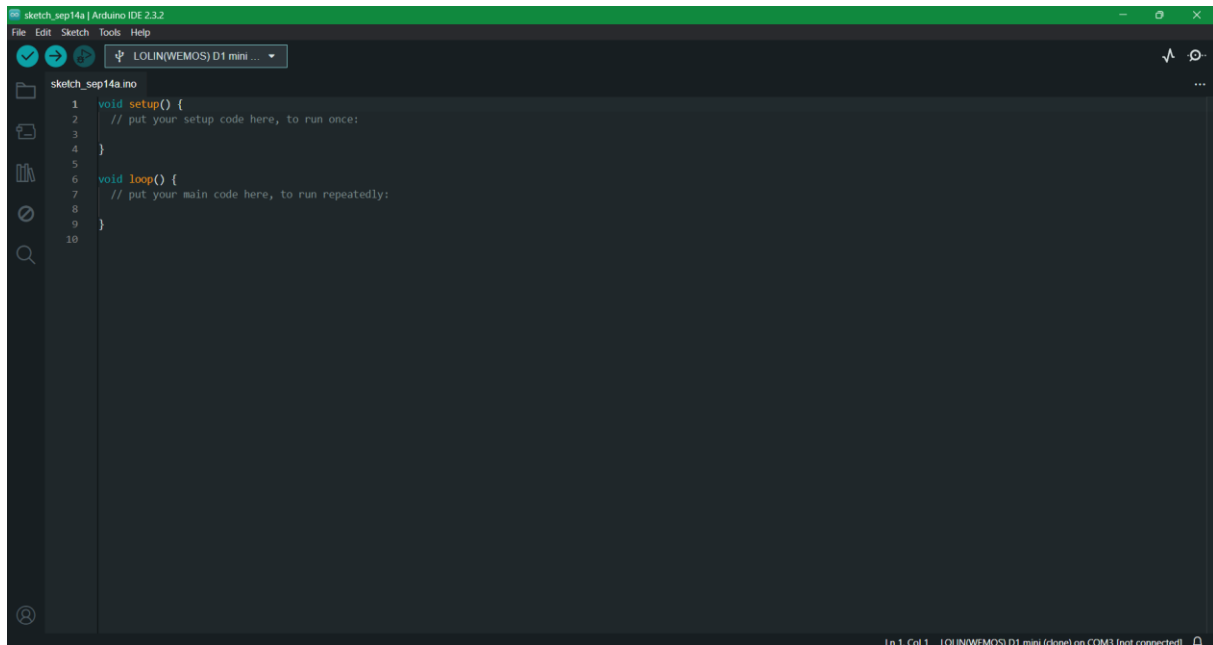
Μερικά από τα πλεονεκτήματα της χρήσης ενός Arduino είναι:

- Χαμηλό κόστος – Οι πλακέτες είναι φθηνότερες συγκριτικά με τις πλακέτες άλλου κατασκευαστή.
- Λογισμικό που τρέχει σε διάφορες πλατφόρμες – Το IDE έχει τη δυνατότητα να τρέξει τόσο σε Windows και Linux, όσο και σε Macintosh OSX.
- Από στη χρήση – Το IDE είναι κατάλληλο τόσο για αρχάριους, όσο και για προχωρημένους χρήστες.
- Επεκτάσιμο λογισμικό ανοιχτού κώδικα – Η γλώσσα μπορεί να επεκταθεί μέσω βιβλιοθηκών C++ [27].

			
Arduino UNO R4 Minima	Arduino UNO R4 WiFi	Arduino UNO R3	Arduino Leonardo
			
Arduino UNO Mini Limited Edition	Arduino Micro	Arduino Zero	Arduino UNO WiFi Rev2

Εικόνα 2.2: Οκτώ από τα κλασικά μοντέλα Arduino [29].

Το Arduino IDE 2 αποτελεί μια βελτιωμένη έκδοση του πρώτου IDE, η οποία προσφέρει σημαντικές αναβαθμίσεις όπως ενσωματωμένο εργαλείο αποσφαλμάτωσης και βελτιωμένο περιβάλλον χρήστη [30]. Στην εργασία χρησιμοποιήθηκε το Arduino IDE 2.3.2 για τη συγγραφή, αποσφαλμάτωση και εισαγωγή του κώδικα στο hardware της εργασίας.



Εικόνα 2.3: Το Arduino IDE 2.3.2.

Συναρτήσεις / μέθοδοι / κατασκευαστές / τελεστές που χρησιμοποιήθηκαν στην εργασία:

- `indexOf()` – Μέθοδος που χρησιμοποιείται για τον εντοπισμό ενός χαρακτήρα ή μιας συμβολοσειράς μέσα σε μία άλλη συμβολοσειρά. Δέχεται ως παραμέτρους τον χαρακτήρα ή τη συμβολοσειρά που θα χρησιμοποιηθεί για την αναζήτηση και (προαιρετικά) έναν δείκτη έναρξης ο οποίος αντιστοιχεί στον χαρακτήρα από τον οποίο θα ξεκινήσει η αναζήτηση. Στην περίπτωση που ο προαιρετικός δείκτης λήξης δεν συμπεριληφθεί, θα ξεκινήσει την αναζήτηση από τον πρώτο χαρακτήρα. Επιστρέφει τον δείκτη που αντιστοιχεί στην τοποθεσία του χαρακτήρα ή της συμβολοσειράς προς αναζήτηση, ή την τιμή -1 εφόσον δεν βρεθεί τίποτα [31].
- `substring()` – Μέθοδος που χρησιμοποιείται για τη λήψη της υποομάδας μιας συμβολοσειράς. Δέχεται ως παραμέτρους τον δείκτη έναρξης ο οποίος αντιστοιχεί στον χαρακτήρα από όπου ξεκινάει η υποομάδα και (προαιρετικά) τον δείκτη λήξης ο οποίος αντιστοιχεί στον χαρακτήρα με τον οποίο τελειώνει η υποομάδα. Στην περίπτωση που ο προαιρετικός δείκτης λήξης δεν συμπεριληφθεί, θα επιστρέψει την υποομάδα η οποία σταματάει στο τέλος της αρχικής συμβολοσειράς [32].
- `length()` – Μέθοδος που χρησιμοποιείται για την εύρεση του μήκους μιας συμβολοσειράς. Δέχεται ως παράμετρο μια συμβολοσειρά. [33]
- `String()` – Κατασκευαστής που χρησιμοποιείται για την μετατροπή μιας μεταβλητής σε συμβολοσειρά. Δέχεται ως παραμέτρους την μεταβλητή και (προαιρετικά) τη βάση στην οποία θα μορφοποιηθεί μια ακέραιη τιμή (π.χ. HEX = hexadecimal, BIN = binary) ή, εφόσον η μεταβλητή είναι δεκαδικός αριθμός, τον αριθμό των δεκαδικών ψηφίων που θα διατηρηθούν [34].
- `setup()` – Συνάρτηση που καλείται όταν ξεκινάει ένα σκίτσο Arduino. Εκτελείται μία φορά και εμπεριέχει αρχικοποιήσεις βιβλιοθηκών, μεταβλητών, κτλ [35].

- `pinMode()` – Συνάρτηση που χρησιμοποιείται για τον καθορισμό της συμπεριφοράς ενός ακροδέκτη. Δέχεται ως παραμέτρους τον αριθμό του ακροδέκτη και τον τρόπο λειτουργίας του (π.χ. `INPUT` = είσοδος, `OUTPUT` = έξοδος) [36].
- `digitalWrite()` – Συνάρτηση που χρησιμοποιείται για τον καθορισμό της κατάστασης ενός ακροδέκτη. Δέχεται ως παραμέτρους τον αριθμό του ακροδέκτη και μία κατάστασή (`HIGH` = ενεργός, `LOW` = ανενεργός) [37].
- `sizeof()` – Τελεστής που χρησιμοποιείται για την εύρεση του αριθμού των bytes από τα οποία αποτελείται ένας πίνακας [38].
- `min()` – Συνάρτηση που χρησιμοποιείται για τον υπολογισμό του ελάχιστου μεταξύ δύο αριθμών. Δέχεται ως παραμέτρους τους δύο αριθμούς [39].
- `max()` – Συνάρτηση που χρησιμοποιείται για τον υπολογισμό του μέγιστου μεταξύ δύο αριθμών. Δέχεται ως παραμέτρους τους δύο αριθμούς [40].
- `loop()` – Συνάρτηση που χρησιμοποιείται για τη δημιουργία διαδοχικών βρόχων, ώστε να γίνεται ενεργός έλεγχος της πλακέτας [41].
- `delay()` – Συνάρτηση που χρησιμοποιείται για την παύση του προγράμματος για ένα συγκεκριμένο χρονικό διάστημα. Δέχεται ως παράμετρο το χρονικό διάστημα σε χιλιοστά του δευτερολέπτου [42].

Συναρτήσεις / μέθοδοι που χρησιμοποιήθηκαν κατά την αποσφαλμάτωση και αφαιρέθηκαν από την τελική εργασία:

- `Serial.begin()` – Συνάρτηση που χρησιμοποιείται για τον ορισμό του ρυθμού των δεδομένων σε bit / sec (baud). Δέχεται ως παραμέτρους την ταχύτητα σε baud και (προαιρετικά) μία παράμετρο που ρυθμίζει τα δεδομένα, την ισοτιμία και τα stop bits [43].
- `Serial.println()` – Συνάρτηση που χρησιμοποιείται για την εκτύπωση δεδομένων ως κείμενο ASCII και ακολουθείται από αλλαγή γραμμής. Δέχεται ως παραμέτρους την τιμή προς εκτύπωση και (προαιρετικά) τη βάση στην οποία θα μορφοποιηθεί η τιμή (π.χ. `HEX` = hexadecimal, `BIN` = binary) [44].
- `Serial.print()` – Συνάρτηση που χρησιμοποιείται για την εκτύπωση δεδομένων ως κείμενο ASCII. Δέχεται ως παραμέτρους την τιμή προς εκτύπωση και (προαιρετικά) τη βάση στην οποία θα μορφοποιηθεί η τιμή (π.χ. `HEX` = hexadecimal, `BIN` = binary) [45].

Συναρτήσεις / μέθοδοι που δεν χρησιμοποιήθηκαν, αλλά είναι επίσης διαθέσιμες στη γλώσσα Arduino και θα μπορούσαν να χρησιμοποιηθούν σε μία βελτιωμένη εκδοχή της εργασίας: `digitalRead()`, `analogRead()`, `analogReadResolution()`, `analogReference()`, `analogWrite()`, `analogWriteResolution()`, `noTone()`, `pulseIn()`, `pulseInLong()`, `shiftIn()`, `shiftOut()`, `tone()`, `delayMicroseconds()`, `micros()`, `millis()`, `abs()`, `constrain()`, `map()`, `pow()`, `sq()`, `sqrt()`, `cos()`, `sin()`, `tan()`, `isAlpha()`, `isAlphaNumeric()`, `isAscii()`, `isControl()`, `isDigit()`, `isGraph()`, `isHexadecimalDigit()`, `isLowerCase()`, `isPrintable()`, `isPunct()`, `isSpace()`, `isUpperCase()`, `isWhitespace()`, `random()`, `randomSeed()`, `bit()`, `bitClear()`, `bitRead()`, `bitSet()`, `bitWrite()`, `highByte()`, `lowByte()`, `attachInterrupt()`, `detachInterrupt()`, `digitalPinToInterrupt()`, `interrupts()`, `noInterrupts()`, `byte()`, `char()`, `float()`, `int()`, `long()`, `word()` [46].

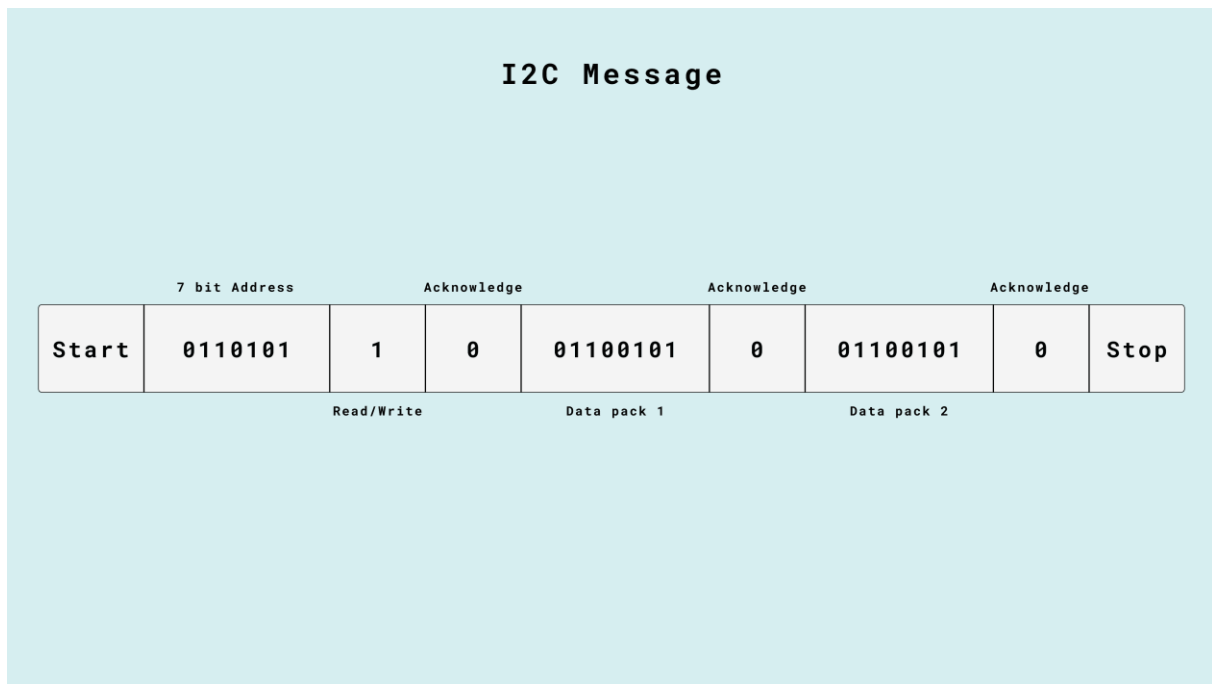
2.3 Πρωτόκολλο I2C

Το πρωτόκολλο I2C περιλαμβάνει δύο γραμμές για αποστολή και λήψη δεδομένων, οι οποίες ονομάζονται SCL (Ακροδέκτης Σειριακού Ρολογιού - παράγει σήμα ρολογιού το οποίο χρησιμοποιείται για τον συγχρονισμό της αποστολής των δεδομένων) και SDA (Ακροδέκτης Σειριακών Δεδομένων - υπεύθυνος για τη μεταφορά των δεδομένων μεταξύ συσκευών). Πιο συγκεκριμένα, με κάθε εναλλαγή

της γραμμής ρολογιού από χαμηλό σε υψηλό, αποστέλλεται ένα bit πληροφορίας από την πλακέτα στη συσκευή I2C μέσω της γραμμής SDA [47].

Ένα μήνυμα I2C αποτελείται από τα εξής δεδομένα:

- Επτά bits για τη διεύθυνση της συσκευής.
- Ένα bit που υποδηλώνει εάν ο ελεγκτής θα διαβάσει ή θα γράψει.
- Ένα bit επιβεβαίωσης ώστε ο ελεγκτής να προχωρήσει στο επόμενο σενάριο από bits.
- Οκτώ bit δεδομένων.
- Ένα ακόμη bit επιβεβαίωσης.
- Οκτώ ακόμη bit δεδομένων.
- Ένα ακόμη bit επιβεβαίωσης [47].



Εικόνα 2.4: Η δομή ενός I2C μηνύματος [47].

2.4 Βιβλιοθήκες Κώδικα

Χρησιμοποιήθηκαν τέσσερις βιβλιοθήκες στο αρχείο του κώδικα προκειμένου να πραγματοποιηθεί επικοινωνία μεταξύ των διάφορων εξαρτημάτων (hardware) της συγκεκριμένης εργασίας.

2.4.1 Wire

Η βιβλιοθήκη Wire χρησιμοποιείται για την επικοινωνία του μικροελεγκτή ESP8266 με τις συσκευές που χρησιμοποιούν το πρωτόκολλο I2C. Η βιβλιοθήκη χρησιμοποιεί διευθύνσεις των επτά bit, παρά το γεγονός ότι πολλές φορές οι I2C διευθύνσεις είναι 8 bit. Σε εκείνη την περίπτωση είναι αναγκαία η μετατόπιση της διεύθυνσης κατά ένα bit προς τα δεξιά [48].

Προκειμένου να χρησιμοποιηθεί η βιβλιοθήκη, είναι αναγκαίο να συμπεριληφθεί στην αρχή του αρχείου του κώδικα ως εξής [48]:

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

Συναρτήσεις / μέθοδοι που χρησιμοποιήθηκαν στην εργασία:

- `begin()` - Καλείται μόνο μια φορά, καθώς αρχικοποιεί τη βιβλιοθήκη και πραγματοποιεί σύνδεση με τον δίαυλο I2C. Εάν δεχθεί ως παράμετρο μία διεύθυνση με επτά bit, θεωρείται περιφερειακή συσκευή. Εάν δεν δεχθεί καμία παράμετρο, θεωρείται ελεγκτής [49].
- `beginTransmission()` - Δέχεται ως παράμετρο τη διεύθυνση της συσκευής στην οποία θα στείλει δεδομένα και ξεκινάει τη μετάδοση προς αυτή τη συσκευή [50].
- `endTransmission()` - Τερματίζει τη μετάδοση προς την περιφερειακή συσκευή, μεταδίδοντας τα δεδομένα που είχαν μπει στην ουρά. Δέχεται ως παράμετρο μια boolean η οποία εάν είναι true, διακόπτεται η σύνδεση, ενώ εάν είναι false, η σύνδεση παραμένει ενεργή [51].
- `write()` - Χρησιμοποιείται είτε από έναν ελεγκτή για να θέσει σε ουρά τα δεδομένα προς μετάδοση στην περιφερειακή συσκευή, είτε από μία περιφερειακή συσκευή για να γράψει δεδομένα κατόπιν αιτήματος ενός ελεγκτή. Δέχεται ως παραμέτρους είτε μια τιμή ενός byte, είτε ένα string (συμβολοσειρά) πολλών bytes, είτε έναν πίνακα δεδομένων πολλών bytes μαζί με τον αριθμό των bytes που θα μεταδοθούν [52].

Συναρτήσεις / μέθοδοι που δεν χρησιμοποιήθηκαν, αλλά είναι επίσης διαθέσιμες στη βιβλιοθήκη Wire και θα μπορούσαν να χρησιμοποιηθούν σε μία βελτιωμένη εκδοχή της εργασίας: `end()`, `requestFrom()`, `available()`, `read()`, `setClock()`, `onReceive()`, `onRequest()`, `setWireTimeout()`, `clearWireTimeoutFlag()`, `getWireTimeoutFlag()` [48].

2.4.2 ESP8266WiFi

Η βιβλιοθήκη ESP8266WiFi χρησιμοποιείται για τη σύνδεση του μικροελεγκτή ESP8266 με ένα δίκτυο WiFi, ώστε να είναι εφικτή η μεταφορά των δεδομένων [49].

Προκειμένου να χρησιμοποιηθεί η βιβλιοθήκη, είναι αναγκαίο να συμπεριληφθεί στην αρχή του αρχείου του κώδικα ως εξής [49]:

```
#include <ESP8266WiFi.h>
```

Συναρτήσεις / μέθοδοι που χρησιμοποιήθηκαν στην εργασία:

- `mode()` – Ρυθμίζει τον τρόπο λειτουργίας του ESP8266. Δέχεται ως όρισμα τον τρόπο λειτουργίας (π.χ. `WIFI_STA` για λειτουργία σταθμού). [49]
- `config()` – Απενεργοποιεί τον πελάτη DHCP και προσφέρει τη δυνατότητα να διαμορφωθεί συγκεκριμένη σύνδεση IP [53].
- `begin()` – Αρχικοποιεί τη σύνδεση στο δίκτυο WiFi. Δέχεται δύο παραμέτρους, το όνομα και τον κωδικό του δικτύου [49].

Συναρτήσεις / μέθοδοι που χρησιμοποιήθηκαν κατά την αποσφαλμάτωση και αφαιρέθηκαν από την τελική εργασία:

- `status()` – Χρησιμοποιείται ως έλεγχος της κατάστασης της σύνδεσης στο δίκτυο WiFi. Έχει έξι διαφορετικές ενδείξεις:
 - 0 (`WL_IDLE_STATUS`) – Στην περίπτωση όπου αλλάζει η κατάσταση.
 - 1 (`WL_NO_SSID_AVAIL`) – Στην περίπτωση όπου υπήρξε αδυναμία επικοινωνίας με το ρυθμισμένο SSID.
 - 3 (`WL_CONNECTED`) – Στην περίπτωση όπου η σύνδεση ολοκληρώθηκε με επιτυχία.
 - 4 (`WL_CONNECT_FAILED`) – Στην περίπτωση όπου η σύνδεση δεν ολοκληρώθηκε με επιτυχία.
 - 6 (`WL_CONNECT_WRONG_PASSWORD`) – Στην περίπτωση όπου ο κωδικός πρόσβασης που έχει εισαχθεί είναι λάθος.
 - 7 (`WL_DISCONNECTED`) – Στην περίπτωση όπου ο μικροελεγκτής δεν έχει ρυθμιστεί σε κατάσταση σταθμού.
- `localIP()` – Χρησιμοποιείται για την εκτύπωση της διεύθυνσης IP που έχει εκχωρηθεί στον μικροελεγκτή [49].

Συναρτήσεις / μέθοδοι που δεν χρησιμοποιήθηκαν, αλλά είναι επίσης διαθέσιμες στη βιβλιοθήκη ESP8266WiFi και θα μπορούσαν να χρησιμοποιηθούν σε μία βελτιωμένη εκδοχή της εργασίας: `config()`, `reconnect()`, `disconnect()`, `isConnected()`, `setAutoConnect()`, `getAutoConnect()`, `setAutoReconnect()`, `waitForConnectResult()`, `macAddress()`, `subnetMask()`, `gatewayIP()`, `dnsIP()`, `hostname()`, `SSID()`, `psk()`, `BSSID()`, `BSSIDstr()`, `RSSI()`, `stationKeepAliveSetIntervalMs()`, `stationKeepAliveStop()`, `stationKeepAliveNow()`, `enableInsecureWEP()`, `getListenInterval()`, `isSleepLevelMax()`, `beginWPSConfig()`, `beginSmartConfig()`, `stopSmartConfig()`, `smartConfigDone()` [54].

2.4.3 ESPAsyncWebServer & ESPAsyncTCP

Η βιβλιοθήκη `ESPAsyncWebServer` χρησιμοποιείται για τη δημιουργία ενός ασύγχρονου διακομιστή (server) στον μικροελεγκτή ESP8266 [55]. Για την ορθή του λειτουργία, εφόσον χρησιμοποιείται ο μικροελεγκτής ESP8266, απαιτείται να κληθεί η συμπληρωματική βιβλιοθήκη `ESPAsyncTCP` στον κώδικα. Εναλλακτικά, εάν χρησιμοποιούνταν ο μικροελεγκτής ESP32, θα ήταν αναγκαίο να κληθεί η βιβλιοθήκη `AsyncTCP` έναντι της βιβλιοθήκης `ESPAsyncTCP` [56].

Προκειμένου να χρησιμοποιηθούν οι βιβλιοθήκες, είναι αναγκαίο να συμπεριληφθούν στην αρχή του αρχείου του κώδικα ως εξής [56]:

```
#include <ESPAsyncWebServer.h>
#include <ESPAsyncTCP.h>
```

Συναρτήσεις / μέθοδοι που χρησιμοποιήθηκαν στην εργασία:

- `on()` – Ορίζει τη συμπεριφορά του διακομιστή σε HTTP αιτήματα. Δέχεται σαν παραμέτρους μια διαδρομή (path), τη μέθοδο των αιτημάτων (`HTTP_GET`, `HTTP_POST`, `HTTP_ANY`) και μια συνάρτηση χειρισμού (handler function).
- `send()` – Χρησιμοποιείται για την αποστολή των HTTP απαντήσεων στον πρόγραμμα περιήγησης.
- `begin()` – Εκκινεί τον διακομιστή [56].

Συναρτήσεις / μέθοδοι που δεν χρησιμοποιήθηκαν, αλλά είναι επίσης διαθέσιμες στη βιβλιοθήκη ESPAsyncWebServer και θα μπορούσαν να χρησιμοποιηθούν σε μία βελτιωμένη έκδοση της εργασίας: `addHandler()`, `removeHandler()`, `addRewrite()`, `removeRewrite()`, `reset()`, `serveStatic()`, `onNotFound()`, `onFileUpload()`, `onRequestBody()`, `end()` [56].

2.5 HTML & HTML5

Η HTML είναι μια γλώσσα σημείωσης (σήμανσης) της οποίας οι προδιαγραφές δημοσιεύτηκαν το 1991 από τον Tim Berners-Lee. Βασισμένη σε αυτές τις προδιαγραφές, η ομάδα WHATWG, η οποία αποτελούνταν από προγραμματιστές των Opera και Mozilla, ολοκλήρωσε το 2014 τη γλώσσα κατασκευής ιστοσελίδων που είναι σήμερα γνωστή ως HTML5. Τα HTML έγγραφα ακολουθούν μια συγκεκριμένη δομή η οποία αποτελείται από στοιχεία (elements), χάρη στα οποία μπορούν να παρουσιαστούν παράγραφοι, πίνακες, σύνδεσμοι, λίστες, εικόνες, κ.α. στην εκάστοτε ιστοσελίδα. Κάθε στοιχείο μπορεί να λάβει συγκεκριμένες ιδιότητες (attributes), οι οποίες αποτελούν πληροφορίες που διαμορφώνουν τη λειτουργία του, ανάλογα με τις ανάγκες χρήσης του στοιχείου. Τα στοιχεία ακολουθούν τη σύνταξη `<ετικέτα>περιεχόμενο</ετικέτα>`, ενώ οι ιδιότητες ακολουθούν τη σύνταξη `όνομα_ιδιότητας="τιμή_ιδιότητας"`. Κενό ονομάζεται το στοιχείο που δεν έχει περιεχόμενο, ακολουθώντας εν τη σύνταξη `<στοιχείο όνομα_ιδιότητας="τιμή_ιδιότητας" />`. Ένα στοιχείο μπορεί στοχευθεί με μια κλάση (class) η οποία μπορεί να εφαρμοστεί σε πολλά στοιχεία, ή μια ταυτότητα (id) η οποία είναι αναγκαστικά μοναδική για κάθε στοιχείο [57].

Παράδειγμα ολοκληρωμένης σύνταξης ενός στοιχείου (ψευδοκώδικας) [57]:

```
<ετικέτα όνομα_ιδιότητας1="τιμή_ιδιότητας1" όνομα_ιδιότητας2="τιμή_ιδιότητας2">
    ...περιεχόμενο...
</ετικέτα>
```



Εικόνα 2.5: Το επίσημο logo της HTML5 [58].

Στοιχεία της HTML5 που χρησιμοποιήθηκαν στην εργασία:

- `<svg>` - Ετικέτα για τη δημιουργία Κλιμακούμενων Διανυσματικών Γραφικών (Scalable Vector Graphics) τα οποία ορίζονται σε μορφή XML και διατηρούν την ποιότητα τους ανεξαρτήτως μεγέθους [59].
- `<html>` - Ετικέτα η οποία εμπεριέχει όλα τα υπόλοιπα HTML στοιχεία, πλην του `<!DOCTYPE>` [60].
- `<head>` - Ετικέτα η οποία εμπεριέχει τα μεταδεδομένα της ιστοσελίδας, όπως `<title>`, `<style>`, `<link>`, `<meta>` κτλ [61].
- `<title>` - Ετικέτα η οποία εμπεριέχει τον τίτλο της σελίδας και εμφανίζεται στην καρτέλα του προγράμματος περιήγησης [62].
- `<meta>` - Ετικέτα η οποία εμπεριέχει συγκεκριμένα μεταδεδομένα όπως περιγραφή και λέξεις κλειδιά, τα οποία χρησιμοποιούνται από τις μηχανές αναζήτησης και δεν εμφανίζονται στην ιστοσελίδα [63].
- `<style>` - Ετικέτα η οποία εμπεριέχει το στυλ εμφάνισης των στοιχείων της ιστοσελίδας [64].
- `<body>` - Ετικέτα η οποία εμπεριέχει όλα τα περιεχόμενα μιας ιστοσελίδας, όπως εικόνες, παραγράφους και επικεφαλίδες [65].
- `<div>` - Ετικέτα η οποία χρησιμοποιείται για την τμηματοποίηση των στοιχείων μιας ιστοσελίδας και ενδέχεται να εμπεριέχει διάφορα στοιχεία HTML5 [66].
- `<h4>` - Μία από τις έξι ετικέτες επικεφαλίδων που παρέχει η HTML5, από τις λιγότερο σημαντικές βάση ιεραρχίας [67].
- `<button>` - Ετικέτα η οποία δημιουργεί ένα κουμπί [68].
- `<script>` - Ετικέτα η οποία εμπεριέχει εντολές JavaScript [69].

Στοιχεία που δεν χρησιμοποιήθηκαν, αλλά είναι επίσης διαθέσιμα στην HTML5 και θα μπορούσαν να χρησιμοποιηθούν σε μία βελτιωμένη έκδοση της εργασίας: `<a>`, ``, `
`, `<footer>`, `<form>`, `<header>`, `<h1>` έως `<h6>`, `<hr>`, `<i>`, `<iframe>`, ``, `<input>`, `<label>`, `<link>`, `<nav>`, ``, `<option>`, `<p>`, `<section>`, `<select>`, ``, `<table>`, `<tbody>`, `<td>`, `<textarea>`, `<th>`, `<thead>`, `<tr>`, `<u>`, ``, `<video>` [70].

2.6 CSS & CSS3

Τα Διαδοχικά Φύλλα Στυλ (Cascading Style Sheets) χρησιμοποιούνται για την περιγραφή της παρουσίασης (εμφάνισης) ενός στοιχείου HTML. Το 1996 η W3C εξέδωσε το σύστημα CSS1, την πρώτη επίσημη έκδοση CSS. Το 1997 ακολούθησε η CSS2, ενώ η βελτιωμένη έκδοσή της, CSS2.1, κυκλοφόρησε το 2011. Παράλληλα με τη σχεδίαση του CSS2.1, είχε ξεκινήσει η σχεδίαση του CSS3, το οποίο χωρίστηκε σε διαφορετικές μονάδες CSS, πολλές από τις οποίες χρησιμοποιούνται σήμερα. Τα έγγραφα CSS ακολουθούν μια συγκεκριμένη δομή η οποία αποτελείται από κανόνες στυλ (style rules), οι οποίοι εμπεριέχουν έναν ή περισσότερους επιλογείς (selectors) και μία ή περισσότερες δηλώσεις (declarations). Οι δηλώσεις ακολουθούν τη σύνταξη όνομα_ιδιότητας:τιμή_ιδιότητας;, ενώ οι πολλές δηλώσεις εντός ενός κανόνα στυλ ονομάζονται μπλοκ δηλώσεων (rule block) [57].

Παράδειγμα ολοκληρωμένης σύνταξης ενός κανόνα στυλ (ψευδοκώδικας) [57]:

```
επιλογέας_στοιχείου .επιλογέας_κλάσης {  
    όνομα_ιδιότητας1: τιμή_ιδιότητας1;  
    όνομα_ιδιότητας2: τιμή_ιδιότητας2;  
}
```



Εικόνα 2.6: Το επίσημο logo της CSS3 [71].

Ιδιότητες της CSS3 που χρησιμοποιήθηκαν στην εργασία:

- font-family – Ιδιότητα που ορίζει τη γραμματοσειρά ενός στοιχείου [72].
- background – Ιδιότητα που λειτουργεί ως συντομογραφία για διάφορες ιδιότητες τροποποίησης του φόντου ενός στοιχείου (background-color, background-image, background-position, background-size, background-repeat, background-origin, background-clip, background-attachment) [73].
- color – Ιδιότητα που ορίζει το χρώμα ενός κειμένου [74].
- text-align – Ιδιότητα που ορίζει την οριζόντια στοίχιση ενός στοιχείου [75].
- margin – Ιδιότητα που λειτουργεί ως συντομογραφία για διάφορες ιδιότητες τροποποίησης των περιθωρίων ενός στοιχείου (margin-top, margin-right, margin-bottom, margin-left) [76].
- font-variant – Ιδιότητα που ορίζει το εάν ένα κείμενο θα εμφανίζεται με γραμματοσειρά μικρών κεφαλαίων γραμμάτων ή όχι [77].
- font-size – Ιδιότητα που ορίζει το μέγεθος μιας γραμματοσειράς [78].
- display – Ιδιότητα που ορίζει τη συμπεριφορά εμφάνισης ενός στοιχείου (π.χ. εμφάνιση στην ίδια γραμμή - inline, ή σε ξεχωριστή γραμμή - block) [79].
- padding – Ιδιότητα που λειτουργεί ως συντομογραφία για διάφορες ιδιότητες τροποποίησης του κενού διαστήματος μεταξύ του περιεχομένου και του ορίου ενός στοιχείου (padding-top, padding-right, padding-bottom, padding-left) [80].
- border – Ιδιότητα που λειτουργεί ως συντομογραφία για διάφορες ιδιότητες τροποποίησης του ορίου ενός στοιχείου (border-width, border-style, border-color) [81].
- border-radius – Ιδιότητα που ορίζει την ακτίνα των γωνιών ενός στοιχείου [82].
- box-shadow – Ιδιότητα που δημιουργεί μία ή περισσότερες σκιές σε ένα στοιχείο [83].

- justify-content – Ιδιότητα που ευθυγραμμίζει τα στοιχεία που εμπεριέχονται σε έναν περιέκτη-στοιχείο στον οριζόντιο άξονα. Απαιτείται ο περιέκτης να έχει την ιδιότητα display με τιμή flex [84].
- margin-bottom – Ιδιότητα που ορίζει το κάτω περιθώριο ενός στοιχείου [85].
- align-items – Ιδιότητα που ευθυγραμμίζει τα στοιχεία που εμπεριέχονται σε έναν περιέκτη-στοιχείο στον κάθετο άξονα. Απαιτείται ο περιέκτης να έχει την ιδιότητα display με τιμή flex [86].
- cursor – Ιδιότητα που ορίζει την εμφάνιση του κέρσορα του ποντικιού όταν αυτός θα περνάει επάνω από ένα στοιχείο [87].
- height – Ιδιότητα που ορίζει το ύψος ενός στοιχείου [88].
- fill – Ιδιότητα που ορίζει το χρώμα ενός στοιχείου SVG [89].
- margin-top – Ιδιότητα που ορίζει το άνω περιθώριο ενός στοιχείου [90].
- font-weight – Ιδιότητα που ορίζει το πάχος των χαρακτήρων ενός κειμένου [91].

Ιδιότητες που δεν χρησιμοποιήθηκαν, αλλά είναι επίσης διαθέσιμες στη CSS3 και θα μπορούσαν να χρησιμοποιηθούν σε μία βελτιωμένη εκδοχή της εργασίας: animation, bottom, clip-path, content, flex, float, font-style, left, letter-spacing, line-height, list-style, max-height, max-width, @media, opacity, overflow-x, overflow-y, position, right, rotate, text-align, text-decoration, top, transform, white-space, width, word-wrap, z-index [92].

2.7 JavaScript & AJAX

Έχοντας ως αρχική ονομασία τη «LiveScript», η JavaScript αποτελεί «μια αντικειμενοστραφή γλώσσα δεσμών ενεργειών με δυναμικό ορισμό τύπων», η οποία χρησιμοποιείται στον προγραμματισμό του web και κατασκευάστηκε από τη Netscape το 1996. Η JavaScript έγινε ιδιαίτερα χρήσιμη στα μέσα της δεκαετίας του 2000, με την εμφάνιση της μεθόδου AJAX (Ασύγχρονη JavaScript και XML – Asynchronous JavaScript and XML) η οποία έκανε την εμπειρία χρήστη πολύ πιο διαδραστική. Τα σενάρια (scripts) JavaScript ακολουθούν μια συγκεκριμένη δομή η οποία αποτελείται από μεταβλητές (variables), τελεστές (operators), συνθήκες (conditions), βρόχους (loops), μεθόδους (methods), ιδιότητες (properties) και συναρτήσεις (functions) [57].

Παράδειγμα ολοκληρωμένης σύνταξης ενός σεναρίου (ψευδοκώδικας) [57]:

```

συνάρτηση όνομαΣυνάρτησης(α, β) {
    επέστρεψε α + β;
}
όρισε μεταβλητή1 = 5;
όρισε μεταβλητή2 = 10;
όρισε μεταβλητή3 = όνομαΣυνάρτησης(μεταβλητή1, μεταβλητή2);

```

Μέθοδοι / ιδιότητες που χρησιμοποιήθηκαν στην εργασία:

- `fetch` – Μέθοδος η οποία εκκινεί τη λήψη ενός πόρου από έναν διακομιστή. Έχει τη δυνατότητα να δεχθεί ως προαιρετική παράμετρο το όνομα του πόρου [93].
- `then` – Μέθοδος η οποία ενσωματώνεται σε μία υπόσχεση (promise). Έχει τη δυνατότητα να δεχθεί δύο συναρτήσεις ανάκλησης (callback functions) ως προαιρετικά ορίσματα, μία για την επιτυχία και μία για την αποτυχία [94].
- `json` – Μέθοδος η οποία χρησιμοποιείται στο αντικείμενο Response του Fetch API και μετατρέπει τα JSON δεδομένα σε αντικείμενο JavaScript [95].
- `replace` – Μέθοδος η οποία χρησιμοποιείται στην αντικατάσταση μιας συμβολοσειράς (string) [96].
- `charAt()` – Μέθοδος η οποία χρησιμοποιείται για την εύρεση ενός χαρακτήρα σε μία συγκεκριμένη θέση. Έχει τη δυνατότητα να δεχθεί ως προαιρετική παράμετρο τη θέση του χαρακτήρα [97].
- `toUpperCase()` – Μέθοδος η οποία χρησιμοποιείται για την τροποποίηση μιας συμβολοσειράς σε κεφαλαία γράμματα [98].
- `slice()` – Μέθοδος η οποία χρησιμοποιείται για την εξαγωγή μιας υποομάδας μιας συμβολοσειράς. Έχει τη δυνατότητα να δεχθεί τη θέση του πρώτου χαρακτήρα της υποομάδας ως υποχρεωτική παράμετρο και τη θέση του τελευταίου χαρακτήρα της υποομάδας ως προαιρετική παράμετρο [99].
- `getElementById` – Μέθοδος η οποία χρησιμοποιείται στη στόχευση και την τροποποίηση ενός HTML στοιχείου [100].
- `textContent` – Ιδιότητα η οποία ορίζει ή επιστρέφει το κείμενο ενός καθορισμένου κόμβου (και των απογόνων του) [101].
- `value` – Ιδιότητα η οποία ορίζει ή επιστρέφει την τιμή ενός πεδίου κειμένου [102].
- `catch` – Δήλωση η οποία ορίζει ένα κομμάτι κώδικα το οποίο εκτελείται σε περίπτωση που προκύψει σφάλμα [103].
- `message` – Ιδιότητα η οποία ορίζει ή επιστρέφει ένα μήνυμα σφάλματος [104].
- `style.color` – Ιδιότητα η οποία ορίζει ή επιστρέφει το χρώμα ενός κειμένου [105].

Μέθοδοι / ιδιότητες που δεν χρησιμοποιήθηκαν, αλλά είναι επίσης διαθέσιμες στην JavaScript και θα μπορούσαν να χρησιμοποιηθούν σε μία βελτιωμένη εκδοχή της εργασίας: `Array()`, `find()`, `indexOf()`, `valueOf()`, `toString()`, `getDate()`, `getTime()`, `parseFloat()`, `parseInt()`, `floor()`, `max()`, `min()`, `concat()`, `substring()`, `trim()`, `innerHeight`, `innerWidth`, `length`, `addEventListener()`, `getElementsByClassName()`, `closest()`, `firstChild`, `innerHTML`, `scrollIntoView()`, `keydown`, `keyup`, `mouseenter`, `mouseleave` [106].

2.8 Ανάλυση του Κώδικα της Εργασίας

Στις επόμενες υποενότητες ακολουθεί η ανάλυση του κώδικα της εργασίας. Ο κώδικας της εργασίας συνδυάζει όλες τις βιβλιοθήκες και τις γλώσσες προγραμματισμού και σήμανσης που αναφέρθηκαν έως τώρα, με στόχο να κατασκευάσει ένα πλήρως λειτουργικό ηλεκτρονικό χειριστήριο, μέσω του οποίου μπορεί ο χρήστης να ελέγξει την ένταση του ήχου, καθώς και τα μπάσα, τα μεσαία και τα πρίμα του ενισχυτή. Ο χρήστης μπορεί να αποκτήσει πρόσβαση στο χειριστήριο, εφόσον είναι συνδεδεμένος στο ίδιο δίκτυο, εισάγοντας την προκαθορισμένη στατική IP του συστήματος. Για τη συγγραφή του κώδικα χρησιμοποιήθηκαν όλες οι πηγές που έχουν προαναφερθεί. Η υλοποίηση του κώδικα έγινε με τη χρήση των προγραμμάτων Arduino IDE και VSCode.

2.8.1 Συμπερίληψη των Βιβλιοθηκών

Το παρακάτω κομμάτι κώδικα συμπεριλαμβάνει τις τέσσερις απαραίτητες βιβλιοθήκες που αναφέρθηκαν σε προηγούμενες υποενότητες του κεφαλαίου, με σκοπό την ορθή λειτουργία του μικροελεγκτή ESP8266.

```
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
```

2.8.2 Διαπιστευτήρια Wi-Fi, Στατική IP & Αρχικοποίηση Αντικειμένου Server

Το παρακάτω κομμάτι κώδικα αρχικοποιεί δύο σταθερές οι οποίες αντιστοιχούν στα διαπιστευτήρια του δικτύου Wi-Fi, στο οποίο και θα συνδεθεί το σύστημα. Η πρώτη σταθερά είναι το όνομα του δικτύου, ενώ η δεύτερη σταθερά είναι ο κωδικός πρόσβασης για το συγκεκριμένο δίκτυο. Στη συνέχεια, ορίζεται μια στατική διεύθυνση IP, μια διεύθυνση πύλης και μία μάσκα υποδικτύου, ώστε ο μικροελεγκτής ESP8266 να έχει σταθερές ρυθμίσεις δικτύου και να χρησιμοποιείται πάντα η ίδια IP για την εμφάνιση του χειριστηρίου. Τέλος, αρχικοποιείται ένα αντικείμενο της κλάσης AsyncWebServer και δέχεται ως παράμετρο την θύρα 80, η οποία χρησιμοποιείται ως προεπιλογή στις HTTP συνδέσεις από πολλά προγράμματα περιήγησης διαδικτύου (browsers). Έτσι, ο μικροελεγκτής ESP8266 αποκτάει τη δυνατότητα να σερβίρει ιστοσελίδες και να διαχειρίζεται τις αλληλεπιδράσεις των χρηστών με αυτές.

```
const char* ssid = "...";
const char* password = "...";

IPAddress staticIP(..., ..., ..., ...);
IPAddress gateway(..., ..., ..., ...);
IPAddress subnet(..., ..., ..., ...);

AsyncWebServer server(80);
```

2.8.3 Αρχικοποίηση Βασικών Σταθερών και Μεταβλητών

Το παρακάτω κομμάτι κώδικα αρχικοποιεί τις βασικές σταθερές και μεταβλητές που θα χρησιμοποιηθούν από τις συναρτήσεις του συστήματος. Αυτές έχουν ως εξής:

- volume, bass, middle, treble – Βασικές μεταβλητές με τις οποίες ελέγχεται η ένταση, τα μπάσα, τα μεσαία και τα πρίμα. Το volume αρχικοποιείται στο 50%, ενώ τα bass, middle και treble αρχικοποιούνται στα 0dB.
- ampMute, ampSDN, dspMute – Σταθερές των βασικών pins που χρησιμοποιήθηκαν για τον έλεγχο της σίγασης του ενισχυτή, της κατάστασης (ενεργοποίηση / απενεργοποίηση) του ενισχυτή και της σίγασης του DSP, αντίστοιχα.
- I2CInit[] – Πίνακας ο οποίος περιέχει τις απαραίτητες τιμές για την ορθή λειτουργία του DSP. Οι ρυθμίσεις περιγράφονται στον Πίνακα 2.1 παρακάτω.
- DSPAddress – Σταθερά στην οποία αποθηκεύεται I2C διεύθυνση του DSP, προκειμένου να επικοινωνήσει μαζί του ο μικροελεγκτής.

- arrowUpIcon, arrowDownIcon – Σταθερές στις οποίες αποθηκεύονται τα SVG εικονίδια που εμφανίζονται ως κουμπιά στην ιστοσελίδα του κέντρου ελέγχου του συστήματος.

Πίνακας 2.1: Τιμές για την ορθή λειτουργία του DSP

Ρύθμιση: Loudness (7) – Τιμή: 64 (01000000)	
Soft step action	Act
High boost	Off
Center frequency	Flat
Attenuation	0 dB
Ρύθμιση: Soft mute / others (4) – Τιμή: 207 (11001111)	
Anti-alias filter	Off
Fast charge	Off
Subwoofer enable	On
Subwoofer input source	Input mux
Soft mute time	15.36 ms
Pin influence for mute	IIC
Soft mute	On
Ρύθμιση: SoftStep II / DC detector (6) – Τιμή: 204 (11110000)	
Spike rejection time constant	44 μ s
Zero-comparator window size	\pm 25 mV
Soft step time	5 ms
Subwoofer right soft step	On
Subwoofer left soft step	On
Speaker RR soft step	On
Ρύθμιση: Speaker attenuation (FL/FR/RL/RR/SWL/SWR) (13-18) – Τιμή: 0 (00000000)	
Soft step action	Act
Gain/attenuation	0 dB
Ρύθμιση: Volume / output gain (8) – Τιμή: 0 (00000000)	
Soft step action	Act
Output gain	1 dB
Gain/attenuation	+0 dB
Ρύθμιση: Main selector (0) – Τιμή: 0 (00000000)	
Subwoofer flat	Off
Main source input gain select	0 dB
FD / QD2 selection	FD
Main source selector	SE1
Ρύθμιση: Main selector (0) – Τιμή: 4 (00000100)	
Subwoofer flat	Off
Main source input gain select	0 dB
FD / QD2 selection	FD
Main source selector	SE2

```

int volume = 50;
int bass = 0;
int middle = 0;
int treble = 0;
int DSPMute = D5;
int ampMute = D7;
int ampSDN = D6;

uint8_t I2CInit[] = {
    7, 64, // 01000000
    4, 207, // 11001111
    6, 240, // 11110000
    13, 0, // 00000000
    14, 0, // 00000000
    15, 0, // 00000000
    16, 0, // 00000000
    17, 0, // 00000000
    18, 0, // 00000000
    8, 0, // 00000000
    0, 0, // 00000000
    0, 4, // 00000100
    0, 0 // 00000000
};

const int DSPAddress = 68; // 01000100

const char* arrowUpIcon = R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 512 512"><path d="M256 512A256 256 0 1 0 256 0a256 256 0 1 0 0
512zM135.1 217.4l107.1-99.9c3.8-3.5 8.7-5.5 13.8-5.5s10.1 2 13.8 5.5l107.1
99.9c4.5 4.2 7.1 10.1 7.1 16.3c0 12.3-10 22.3-22.3 22.3H304v96c0 17.7-14.3 32-
32 32H240c-17.7 0-32-14.3-32-32V256H150.3C138 256 128 246 128 233.7c0-6.2 2.6-
12.1 7.1-16.3z"/></svg>)SVG";
const char* arrowDownIcon = R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 512 512"><path d="M256 0a256 256 0 1 0 0 512A256 256 0 1 0 256
0zM376.9 294.6l269.8 394.5c-3.8 3.5-8.7 5.5-13.8 5.5s-10.1-2-13.8-5.5l135.1
294.6c-4.5-4.2-7.1-10.1-7.1-16.3c0-12.3 10-22.3 22.3-22.3l157.7 0 0-96c0-17.7
14.3-32 32-32l32 0c17.7 0 32 14.3 32 32l10 96 57.7 0c12.3 0 22.3 10 22.3 22.3c0
6.2-2.6 12.1-7.1 16.3z"/></svg>)SVG";

```

2.8.4 Συνάρτηση: writeToDSP()

Το παρακάτω κομμάτι κώδικα κατασκευάζει τη συνάρτηση writeToDSP(), η οποία χρησιμοποιείται για την αποστολή δεδομένων στον DSP μέσω του I2C πρωτοκόλλου επικοινωνίας. Η συνάρτηση δέχεται ως παραμέτρους μια διεύθυνση και μια τιμή. Αρχικά δημιουργεί έναν πίνακα ο οποίος αποτελείται από δύο bytes, στα οποία καταγράφονται η διεύθυνση και η τιμή που δέχθηκε η συνάρτηση ως παραμέτρους και στη συνέχεια ξεκινάει τη μετάδοση των δεδομένων (δηλ. τη μετάδοση του πίνακα) στον DSP. Τέλος, τερματίζει τη μετάδοση προς τον DSP.

```

void writeToDSP(byte reg, byte value) {
    uint8_t array[2];
    array[0] = reg;
    array[1] = value;
    Wire.beginTransaction(DSPAddress);
    Wire.write(array, 2);
    Wire.endTransmission();
}

```

2.8.5 Συνάρτηση: volumeMapping()

Το παρακάτω κομμάτι κώδικα κατασκευάζει τη συνάρτηση volumeMapping(), η οποία χρησιμοποιείται για τη μετάφραση των ποσοστών έντασης σε ακέραιους αριθμούς. Η συνάρτηση δέχεται ως παράμετρο το ποσοστό της έντασης του ήχου. Ανάλογα το ποσοστό, επιστρέφει τον αντίστοιχο ακέραιο. Το default case χρησιμοποιείται για την αποφυγή σφαλμάτων στο σύστημα, επιστρέφοντας τον ακέραιο που αντιστοιχεί στο μηδενικό ποσοστό.

```

byte volumeMapping(int volumePercent) {
    switch (volumePercent) {
        case 0:    return 63; // 111111
        case 10:   return 57; // 111001
        case 20:   return 51; // 110011
        case 30:   return 43; // 101011
        case 40:   return 38; // 100110
        case 50:   return 0;  // 000000
        case 60:   return 6;  // 000110
        case 70:   return 12; // 001100
        case 80:   return 17; // 010001
        case 90:   return 20; // 010100
        case 100:  return 23; // 010111
        default:   return 63; // 111111
    }
}

```

2.8.6 Συνάρτηση: dBMapping()

Το παρακάτω κομμάτι κώδικα κατασκευάζει τη συνάρτηση dBMapping(), η οποία χρησιμοποιείται για τη μετάφραση των decibel σε ακέραιους αριθμούς. Η συνάρτηση δέχεται ως παράμετρο το αριθμό των decibel. Η συνάρτηση καλύπτει δύο περιπτώσεις:

- -15 έως -1 decibel (τα οποία αντιστοιχεί στους ακέραιους 0 έως 14)
- 0 έως +15 decibel (τα οποία αντιστοιχεί στους ακέραιους 31 έως 16)

```

byte dBMapping(int dbValue) {
    if (dbValue < 0) {
        return (dbValue + 15);
    } else {
        return (31 - dbValue);
    }
}

```

2.8.7 Συνάρτηση: setupDSP()

Το παρακάτω κομμάτι κώδικα κατασκευάζει τη συνάρτηση setupDSP(), η οποία χρησιμοποιείται για την αρχικοποίηση της έντασης του ήχου και των τιμών στα μπάσα, τα μεσαία και τα πρίμα. Η συνάρτηση δεν δέχεται καμία παράμετρο. Προκειμένου να πραγματοποιήσει την αρχικοποίηση, αντιστοιχεί τις τιμές μέσω των συναρτήσεων volumeMapping() και dBMapping() και στέλνει τα δεδομένα μέσω της συνάρτησης writeToDSP(), την οποία καλεί πέντε φορές. Πιο συγκεκριμένα:

- Στέλνει στη διεύθυνση 8 (Volume) την αρχική τιμή της έντασης του ήχου, η οποία έχει τεθεί ως 50%.
- Στέλνει στη διεύθυνση 0 (Main selector) τις ρυθμίσεις: Subwoofer flat = off, Main source input gain select = 0 dB, FD / QD2 selection = FD και Main source selector = SE1.
- Στέλνει στη διεύθυνση 11 (Bass filter) την αρχική τιμή για το μπάσο, η οποία έχει τεθεί ως 0 dB.
- Στέλνει στη διεύθυνση 10 (Middle filter) την αρχική τιμή για τα μεσαία, η οποία έχει τεθεί ως 0 dB.
- Στέλνει στη διεύθυνση 9 (Treble filter) την αρχική τιμή για τα πρίμα, η οποία έχει τεθεί ως 0 dB.

```

void setupDSP() {
    writeToDSP(8, volumeMapping(volume));
    writeToDSP(0, 0);
    writeToDSP(11, dBMapping(bass));
    writeToDSP(10, dBMapping(middle));
    writeToDSP(9, dBMapping(treble));
}

```

2.8.8 Συνάρτηση: replaceAll()

Το παρακάτω κομμάτι κώδικα κατασκευάζει τη συνάρτηση replaceAll(), η οποία χρησιμοποιείται για την αντικατάσταση μιας υποομάδας μιας συμβολοσειράς με μια άλλη υποομάδα μέσα σε αυτή τη συμβολοσειρά. Η συνάρτηση δέχεται ως παραμέτρους μια συμβολοσειρά, μία υποομάδα της συμβολοσειράς που θα αντικατασταθεί και την υποομάδα με την οποία θα την αντικαταστήσει. Ο βρόχος εξασφαλίζει ότι θα αντικατασταθεί η υποομάδα όσες φορές και εάν εμφανίζεται μέσα στη συμβολοσειρά.

```
String replaceAll(String str, const String& from, const String& to) {
    int position = str.indexOf(from);
    while (position != -1) {
        str = str.substring(0, position) + to + str.substring(position +
from.length());
        position = str.indexOf(from, position + to.length());
    }
    return str;
}
```

2.8.9 Συνάρτηση: getHTML()

Το παρακάτω κομμάτι κώδικα κατασκευάζει τη συνάρτηση getHTML(), η οποία χρησιμοποιείται για τη δημιουργία μιας δυναμικής HTML σελίδας όπου γίνεται ο έλεγχος της έντασης του ήχου και των τιμών από τα μπάσα, τα μεσαία και τα πρίμα. Η συνάρτηση δεν δέχεται καμία παράμετρο. Η συνάρτηση ξεκινάει κατασκευάζοντας μια συμβολοσειρά πολλαπλών γραμμών στην οποία αποθηκεύεται όλο το HTML, CSS και JavaScript περιεχόμενο της ιστοσελίδας.

```
String getHTML() {
    String html = R"rawliteral(
```

Η συμβολοσειρά ξεκινάει με τη δήλωση της HTML5. Στο εσωτερικό του <head> ορίζεται ο τίτλος της καρτέλας, πραγματοποιείται η ρύθμιση του viewport και ενσωματώνεται το CSS style της ιστοσελίδας. Το style της ιστοσελίδας αποτελείται από τροποποιήσεις στο σώμα της ιστοσελίδας, τους τίτλους, το πλαίσιο του χειριστηρίου, τα κουμπιά του, τα εικονίδια και το μήνυμα που ενημερώνει για την κατάσταση του. Οι τροποποιήσεις επηρεάζουν, μεταξύ άλλων, το χρώμα του φόντου, των γραμμμάτων, των ορίων και των εικονιδίων, τον τύπο, το μέγεθος και το πάχος της γραμματοσειράς, τα κενά μεταξύ των στοιχείων, τη σκιά τους και τον τρόπο με τον οποίο στοιχίζονται μεταξύ τους.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Amplifier CC</title>
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <style>
            body {
                font-family: Arial, sans-serif;
                background: #222;
                color: #fff;
                text-align: center;
                margin: 50px 5px;
            }
            h4 {
                margin: 30px 0px 10px;
```

```

        font-variant: small-caps;
        font-size: 15px;
    }
    h4:first-of-type {
        margin: 20px 0px 10px;
    }
    .container {
        display: inline-block;
        padding: 30px 20px;
        margin: 20px;
        background: #111;
        border: 2px solid #0b41ff;
        border-radius: 10px;
        box-shadow: 0px 0px 20px #0b41ff;
    }
    .button-pair {
        display: flex;
        justify-content: center;
        margin-bottom: 20px;
    }
    button {
        display: flex;
        align-items: center;
        justify-content: center;
        background: whitesmoke;
        color: #222;
        border: 2px solid #208120;
        border-radius: 10px;
        padding: 15px 25px;
        margin: 0px 10px;
        cursor: pointer;
        font-size: 16px;
    }
    button:hover,
    button:active {
        background: #303030;
        border: 2px solid #29ff01;
    }
    svg {
        height: 24px;
        fill: #208120;
    }
    button:hover svg,
    button:active svg {
        fill: #29ff01;
    }
    #feedback {
        margin-top: 20px;
        font-weight: bold;
    }

```

```

        font-size: 20px;
    }
</style>
</head>

```

Η συμβολοσειρά συνεχίζει με το <body>, μέσα στο οποίο βρίσκεται το HTML και JavaScript περιεχόμενο της ιστοσελίδας. Το χειριστήριο ορίζεται ως ένα <div> με κλάση “container”, μέσα στο οποίο εμπεριέχεται η βασική δομή η οποία ακολουθεί το μοτίβο ενός τίτλου και δύο κουμπιών. Το συγκεκριμένο μοτίβο επαναλαμβάνεται τέσσερις φορές προκειμένου να καλύψει τις τέσσερις περιπτώσεις: ένταση ήχου, μπάσα, μεσαία, πρίμα. Οι τίτλοι τροποποιούνται βάσει της τωρινής τιμής της κάθε περίπτωσης. Τα κουμπιά καλούν την συνάρτηση alterStatus(), η οποία αναλύεται παρακάτω, κάθε φορά που δέχονται ένα κλικ, στέλνοντας τις παραμέτρους που αντιστοιχούν στην κάθε περίπτωση. Το <div> με ταυτότητα “feedback” εμφανίζει ένα μήνυμα που εμπεριέχει την τωρινή κατάσταση του συστήματος ή το σφάλμα που ενδέχεται να αντιμετωπίζει.

```

<body>
  <div class="container">
    <h4 id="volumeDisplay">Volume: $volume%</h4>
    <div class="button-pair">
      <button onclick="alterStatus('/volumeup',
'volumeDisplay', '%')">$arrowUpIcon</button>
      <button onclick="alterStatus('/volumedown',
'volumeDisplay', '%')">$arrowDownIcon</button>
    </div>
    <h4 id="bassDisplay">Bass: $bass dB</h4>
    <div class="button-pair">
      <button onclick="alterStatus('/bassup', 'bassDisplay',
' dB')">$arrowUpIcon</button>
      <button onclick="alterStatus('/bassdown',
'bassDisplay', ' dB')">$arrowDownIcon</button>
    </div>
    <h4 id="middleDisplay">Middle: $middle dB</h4>
    <div class="button-pair">
      <button onclick="alterStatus('/middleup',
'middleDisplay', ' dB')">$arrowUpIcon</button>
      <button onclick="alterStatus('/middledown',
'middleDisplay', ' dB')">$arrowDownIcon</button>
    </div>
    <h4 id="trebleDisplay">Treble: $treble dB</h4>
    <div class="button-pair">
      <button onclick="alterStatus('/trebleup',
'trebleDisplay', ' dB')">$arrowUpIcon</button>
      <button onclick="alterStatus('/trebledown',
'trebleDisplay', ' dB')">$arrowDownIcon</button>
    </div>
  </div>
  <div id="feedback"></div>

```

Η συμβολοσειρά ολοκληρώνεται με την κατασκευή δύο JavaScript συναρτήσεων.

Η συνάρτηση `alterStatus()` χρησιμοποιείται για την ανανέωση των τιμών που βρίσκονται στους τίτλους των κουμπιών. Η συνάρτηση δέχεται ως παραμέτρους ένα `url`, την ταυτότητα του στοιχείου που θα τροποποιηθεί και τον τύπο της τιμής (ποσοστό ή decibel). Χρησιμοποιεί το Fetch API προκειμένου να στείλει ένα POST αίτημα στον server και, σε περίπτωση επιτυχίας, τροποποιεί το κείμενο του στοιχείου με την ανάλογη ταυτότητα και καλεί τη συνάρτηση `generateFeedback()` ώστε να αλλάξει το feedback σε “Status updated”. Στην περίπτωση σφάλματος, καλεί τη συνάρτηση `generateFeedback()` και αλλάζει το feedback σε “Error” αποτυπώνοντας το σφάλμα που προέκυψε.

Η συνάρτηση `generateFeedback()` χρησιμοποιείται για την ενημέρωση του χρήστη όσον αφορά την κατάσταση του συστήματος με ένα μήνυμα επιτυχίας ή την εμφάνιση του σφάλματος που προέκυψε. Η συνάρτηση δέχεται ως παραμέτρους το μήνυμα που θα εκτυπωθεί και μία λογική τιμή η οποία καθορίζει το χρώμα του μηνύματος. Στοχεύοντας το `<div>` με ταυτότητα “feedback”, η συνάρτηση ανανεώνει το περιεχόμενο του και τροποποιεί το `style` του.

```
<script>
  function alterStatus(url, elementId, unit) {
    fetch(url, { method: 'POST' })
      .then(response => response.json())
      .then(data => {
        let label = elementId.replace('Display', '');
        label = label.charAt(0).toUpperCase() +
label.slice(1);
        document.getElementById(elementId).textContent
= label + ': ' + data.value + unit;
        generateFeedback('Status updated.', true);
      })
      .catch(error => generateFeedback('Error: ' +
error.message + '.', false));
  }
  function generateFeedback(message, isSuccess) {
    const feedback = document.getElementById('feedback');
    feedback.textContent = message;
    feedback.style.color = isSuccess ? '#228b22' :
'#922724';
  }
</script>
</body>
</html>
```

Η συμβολοσειρά κλείνει και ακολουθεί η αντικατάσταση των απαραίτητων υποομάδων της συμβολοσειράς με τις ανάλογες τιμές, χρησιμοποιώντας τη συνάρτηση `replaceAll()`. Η `replaceAll()` εισάγει τον SVG κώδικα των εικονιδίων, την ανάλογη τιμή της έντασης του ήχου και τις ανάλογες τιμές στα μπάσα, μεσαία και πρίμα. Τέλος, η συνάρτηση `getHTML()` επιστρέφει τη συμβολοσειρά `html` η οποία εμπεριέχει πλέον όλα τα δεδομένα που είναι αναγκαία ώστε να εμφανιστεί και να λειτουργήσει ορθά η ιστοσελίδα του χειριστηρίου.

```

)rawliteral";

html = replaceAll(html, "$arrowUpIcon", arrowUpIcon);
html = replaceAll(html, "$arrowDownIcon", arrowDownIcon);
html = replaceAll(html, "$volume", String(volume));
html = replaceAll(html, "$bass", String(bass));
html = replaceAll(html, "$middle", String(middle));
html = replaceAll(html, "$treble", String(treble));

return html;
}

```

2.8.10 void setup() και Αρχικοποίηση των Βασικών Pins

Το παρακάτω κομμάτι κώδικα χρησιμοποιεί τη συνάρτηση setup(), η οποία τρέχει μία φορά κατά την εκκίνηση και ξεκινάει η αρχικοποίηση των στοιχείων του συστήματος. Με τη χρήση της συνάρτησης pinMode καθορίζονται τα pins της σίγασης του DSP και της σίγασης και της απενεργοποίησης του ενισχυτή ως OUTPUT. Με τη χρήση της συνάρτησης digitalWrite() καθορίζεται το pin της σίγασης του DSP ως HIGH (δηλαδή ενεργό) και τα pins της σίγασης και της ενεργοποίησης του ενισχυτή ως LOW (δηλαδή ανενεργό).

```

void setup() {
  pinMode(DSPMute, OUTPUT);
  pinMode(ampMute, OUTPUT);
  pinMode(ampSDN, OUTPUT);
  digitalWrite(DSPMute, HIGH);
  digitalWrite(ampMute, LOW);
  digitalWrite(ampSDN, LOW);
}

```

2.8.11 Ρύθμιση WiFi, Wire και DSP

Το παρακάτω κομμάτι κώδικα πραγματοποιεί την ανάλογη σύνδεση στο διαδίκτυο και στέλνει, μέσω αυτού, όλες τις απαραίτητες ρυθμίσεις στον DSP. Με τη χρήση του WiFi.mode, ο μικροελεγκτής ESP8266 μπαίνει σε κατάσταση σταθμού, το οποίο του επιτρέπει να συνδεθεί με δίκτυο. Με τη χρήση του WiFi.config, ορίζονται οι ρυθμίσεις του δικτύου. Με τη χρήση του WiFi.begin, ο μικροελεγκτής ESP8266 συνδέεται στο διαδίκτυο, χρησιμοποιώντας τα διαπιστευτήρια που έχουν αποθηκευτεί στις σταθερές ssid και password. Με τη χρήση του Wire.begin, ο μικροελεγκτής ESP8266 ορίζει το pin D2 ως ακροδέκτη σειριακών δεδομένων (SDA) και το pin D1 ως ακροδέκτη σειριακού ρολογιού (SCL). Η λούπα γύρω από τον πίνακα I2CInit στέλνει τις αντίστοιχες ρυθμίσεις στο DSP με τη χρήση της συνάρτησης writeToDSP. Η συνάρτηση setupDSP στέλνει την αρχική τιμή της έντασης του ήχου και τις αρχικές τιμές από τα μπάσα, μεσαία και πρίμα στον DSP.

```

WiFi.mode(WIFI_STA);
WiFi.config(staticIP, gateway, subnet);
WiFi.begin(ssid, password);

```

```
Wire.begin(D2, D1);
for (int i = 0; i < (int)sizeof(I2CInit); i += 2) {
    writeToDSP(I2CInit[i], I2CInit[i+1]);
}
setupDSP();
```

2.8.12 Server: Κατασκευή της HTML Σελίδας

Το παρακάτω κομμάτι κώδικα χρησιμοποιεί τη μέθοδο `on`, προκειμένου να σερβίρει την ιστοσελίδα του χειριστηρίου στον χρήστη, όταν αυτός εισάγει την ανάλογη IP ως `url` στο πρόγραμμα περιήγησης του. Στη μέθοδο εισάγεται ως παράμετρος διαδρομής η κορυφαία διαδρομή (`/`), ως παράμετρος μεθόδου αιτημάτων η `HTTP_GET`, και ως παράμετρος συνάρτησης χειρισμού μία ανώνυμη συνάρτηση η οποία επιστρέφει τη συμβολοσειρά που προκύπτει από τη συνάρτηση `getHTML()`, δηλαδή ολόκληρη την ιστοσελίδα.

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(200, "text/html", getHTML());
});
```

2.8.13 Server: Έλεγχος Τιμών – Ένταση του Ήχου

Το παρακάτω κομμάτι κώδικα χρησιμοποιεί τη μέθοδο `on`, προκειμένου να αυξήσει ή να μειώσει την ένταση του ήχου με το πάτημα του αντίστοιχου κουμπιού. Ανάλογα την περίπτωση, στη μέθοδο εισάγεται ως παράμετρος διαδρομής η διαδρομή `/volumeup` ή η διαδρομή `/volumedown`, η οποία αντιστοιχεί στο πρώτο και στο δεύτερο κουμπί του χειριστηρίου, ως παράμετρος μεθόδου αιτημάτων η `HTTP_POST`, και ως παράμετρος συνάρτησης χειρισμού μια ανώνυμη συνάρτηση η οποία υπολογίζει και επιστρέφει την τιμή της έντασης του ήχου.

Στην περίπτωση της αύξησης της έντασης του ήχου, η αντίστοιχη ανώνυμη συνάρτηση λειτουργεί ως εξής:

- Αποθηκεύει την προηγούμενη τιμή σε μία μεταβλητή “oldVolume”.
- Εκχωρεί ξανά στην μεταβλητή `volume` την τιμή και προσθέτει τον αριθμό 10, με ανώτατο όριο το 100 (καθώς είναι ποσοστό).
- Μεταφράζει το ποσοστό σε ακέραια τιμή χρησιμοποιώντας τη συνάρτηση `volumeMapping()`, ώστε να μπορεί να τη διαβάσει ο DSP.
- Στέλνει τη νέα τιμή στη διεύθυνση `Volume / output gain` του DSP.
- Σε περίπτωση που η παλιά τιμή ήταν 0 και η νέα τιμή είναι μεγαλύτερη του μηδενός, ενεργοποιεί τη σίγαση από τον ενισχυτή.
- Επιστρέφει τη νέα τιμή στην ιστοσελίδα, ώστε το αποτέλεσμα να εμφανιστεί στον χρήστη.

Στην περίπτωση της μείωσης της έντασης του ήχου, η αντίστοιχη ανώνυμη συνάρτηση λειτουργεί ως εξής:

- Αποθηκεύει την προηγούμενη τιμή σε μία μεταβλητή “oldVolume”.
- Εκχωρεί ξανά στην μεταβλητή `volume` την τιμή και αφαιρεί τον αριθμό 10, με κατώτατο όριο το 0 (καθώς είναι ποσοστό).
- Μεταφράζει το ποσοστό σε ακέραια τιμή χρησιμοποιώντας τη συνάρτηση `volumeMapping()`, ώστε να μπορεί να τη διαβάσει ο DSP.

- Στέλνει τη νέα τιμή στη διεύθυνση Volume / output gain του DSP.
- Σε περίπτωση που η παλιά τιμή ήταν μεγαλύτερη του 0 και η νέα τιμή είναι ίση με 0, ενεργοποιεί τη σίγαση από τον ενισχυτή.
- Επιστρέφει τη νέα τιμή στην ιστοσελίδα, ώστε το αποτέλεσμα να εμφανιστεί στον χρήστη.

```
server.on("/volumeup", HTTP_POST, [](AsyncWebServerRequest *request){
    int oldVolume = volume;
    volume = min(100, volume + 10);
    writeToDSP(8, volumeMapping(volume));
    if (oldVolume == 0 && volume > 0) {
        digitalWrite(ampMute, LOW);
    }
    request->send(200, "application/json", "{\"value\":" + String(volume)
+ "}");
});

server.on("/volumedown", HTTP_POST, [](AsyncWebServerRequest *request){
    int oldVolume = volume;
    volume = max(0, volume - 10);
    writeToDSP(8, volumeMapping(volume));
    if (volume == 0 && oldVolume > 0) {
        digitalWrite(ampMute, HIGH);
    }
    request->send(200, "application/json", "{\"value\":" + String(volume)
+ "}");
});
```

2.8.14 Server: Έλεγχος Τιμών – Μπάσα

Το παρακάτω κομμάτι κώδικα χρησιμοποιεί τη μέθοδο `on`, προκειμένου να αυξήσει ή να μειώσει τα μπάσα με το πάτημα του αντίστοιχου κουμπιού. Ανάλογα την περίπτωση, στη μέθοδο εισάγεται ως παράμετρος διαδρομής η διαδρομή `/bassup` ή η διαδρομή `/bassdown`, η οποία αντιστοιχεί στο τρίτο και στο τέταρτο κουμπί του χειριστηρίου, ως παράμετρος μεθόδου αιτημάτων η `HTTP_POST`, και ως παράμετρος συνάρτησης χειρισμού μια ανώνυμη συνάρτηση η οποία υπολογίζει και επιστρέφει τη νέα τιμή από τα μπάσα.

Στην περίπτωση της αύξησης της τιμής από τα μπάσα, η αντίστοιχη ανώνυμη συνάρτηση λειτουργεί ως εξής:

- Εκχωρεί ξανά στη μεταβλητή `bass` την τιμή και προσθέτει τον αριθμό 1, με ανώτατο όριο το 15 (σύμφωνα με το `datasheet` και καθώς είναι dB).
- Μεταφράζει τα dB σε ακέραια τιμή χρησιμοποιώντας τη συνάρτηση `dBMapping()`, ώστε να μπορεί να τη διαβάσει ο DSP.
- Στέλνει τη νέα τιμή στη διεύθυνση Bass filter του DSP.
- Επιστρέφει τη νέα τιμή στην ιστοσελίδα, ώστε το αποτέλεσμα να εμφανιστεί στον χρήστη.

Στην περίπτωση της μείωσης της τιμής από τα μπάσα, η αντίστοιχη ανώνυμη συνάρτηση λειτουργεί ως εξής:

- Εκχωρεί ξανά στη μεταβλητή bass την τιμή και αφαιρεί τον αριθμό 1, με κατώτατο όριο το -15 (σύμφωνα με το datasheet και καθώς είναι dB).
- Μεταφράζει τα dB σε ακέραια τιμή χρησιμοποιώντας τη συνάρτηση dBMapping(), ώστε να μπορεί να τη διαβάσει ο DSP.
- Στέλνει τη νέα τιμή στη διεύθυνση Bass filter του DSP.
- Επιστρέφει τη νέα τιμή στην ιστοσελίδα, ώστε το αποτέλεσμα να εμφανιστεί στον χρήστη.

```
server.on("/bassup", HTTP_POST, [](AsyncWebServerRequest *request){
    bass = min(15, bass + 1);
    writeToDSP(11, dBMapping(bass));
    request->send(200, "application/json", "{\"value\":" + String(bass) +
"}");
});

server.on("/bassdown", HTTP_POST, [](AsyncWebServerRequest *request){
    bass = max(-15, bass - 1);
    writeToDSP(11, dBMapping(bass));
    request->send(200, "application/json", "{\"value\":" + String(bass) +
"}");
});
```

2.8.15 Server: Έλεγχος Τιμών – Μεσαία

Το παρακάτω κομμάτι κώδικα χρησιμοποιεί τη μέθοδο on, προκειμένου να αυξήσει ή να μειώσει τα μεσαία με το πάτημα του αντίστοιχου κουμπιού. Ανάλογα την περίπτωση, στη μέθοδο εισάγεται ως παράμετρος διαδρομής η διαδρομή /middleup ή η διαδρομή /middledown, η οποία αντιστοιχεί στο πέμπτο και στο έκτο κουμπί του χειριστηρίου, ως παράμετρος μεθόδου αιτημάτων η HTTP_POST, και ως παράμετρος συνάρτησης χειρισμού μια ανώνυμη συνάρτηση η οποία υπολογίζει και επιστρέφει τη νέα τιμή από τα μεσαία.

Στην περίπτωση της αύξησης και της μείωσης της τιμής από τα μεσαία, η αντίστοιχη ανώνυμη συνάρτηση λειτουργεί με τον ίδιο τρόπο που λειτουργεί στα μπάσα, με τη μόνη διαφορά ότι χρησιμοποιεί τη μεταβλητή middle και στέλνει τη νέα τιμή στη διεύθυνση Middle filter του DSP.

```
server.on("/middleup", HTTP_POST, [](AsyncWebServerRequest *request){
    middle = min(15, middle + 1);
    writeToDSP(10, dBMapping(middle));
    request->send(200, "application/json", "{\"value\":" + String(middle)
+ "}");
});

server.on("/middledown", HTTP_POST, [](AsyncWebServerRequest *request){
    middle = max(-15, middle - 1);
    writeToDSP(10, dBMapping(middle));
```

```

        request->send(200, "application/json", "{\"value\":" + String(middle)
+ "}");
    });

```

2.8.16 Server: Έλεγχος Τιμών – Πρίμα

Το παρακάτω κομμάτι κώδικα χρησιμοποιεί τη μέθοδο `on`, προκειμένου να αυξήσει ή να μειώσει τα πρίμα με το πάτημα του αντίστοιχου κουμπιού. Ανάλογα την περίπτωση, στη μέθοδο εισάγεται ως παράμετρος διαδρομής η διαδρομή `/trebleup` ή η διαδρομή `/trebledown`, η οποία αντιστοιχεί στο έβδομο και στο όγδοο κουμπί του χειριστηρίου, ως παράμετρος μεθόδου αιτημάτων η `HTTP_POST`, και ως παράμετρος συνάρτησης χειρισμού μια ανώνυμη συνάρτηση η οποία υπολογίζει και επιστρέφει τη νέα τιμή από τα πρίμα.

Στην περίπτωση της αύξησης και της μείωσης της τιμής από τα πρίμα, η αντίστοιχη ανώνυμη συνάρτηση λειτουργεί με τον ίδιο τρόπο που λειτουργεί στα μπάσα, με τη μόνη διαφορά ότι χρησιμοποιεί τη μεταβλητή `treble` και στέλνει τη νέα τιμή στη διεύθυνση `Treble filter` του DSP.

```

server.on("/trebleup", HTTP_POST, [](AsyncWebServerRequest *request){
    treble = min(15, treble + 1);
    writeToDSP(9, dBMapping(treble));
    request->send(200, "application/json", "{\"value\":" + String(treble)
+ "}");
});

server.on("/trebledown", HTTP_POST, [](AsyncWebServerRequest *request){
    treble = max(-15, treble - 1);
    writeToDSP(9, dBMapping(treble));
    request->send(200, "application/json", "{\"value\":" + String(treble)
+ "}");
});

```

2.8.17 Τελικές Ρυθμίσεις και Εκκίνηση του Server

Στο παρακάτω κομμάτι κώδικα γίνονται οι τελικές ρυθμίσεις του συστήματος και ξεκινάει να λειτουργεί ο ασύγχρονος server. Η συνάρτηση `delay()` προσθέτει μια καθυστέρηση μισού δευτερολέπτου, με στόχο την ομαλή λειτουργία του συστήματος. Με τη χρήση της συνάρτησης `digitalWrite()` καθορίζεται το pin της ενεργοποίησης του ενισχυτή ως `HIGH` (δηλαδή ενεργό). Η μέθοδος `begin` ξεκινάει τον ασύγχρονο server. Εφόσον ο server είναι ασύγχρονος, η συνάρτηση `loop()` δεν χρησιμοποιείται.

```

delay(500);
digitalWrite(ampSDN, HIGH);
server.begin();
}

void loop() {
}

```

2.9 Επίλογος

Στο συγκεκριμένο κεφάλαιο ολοκληρώθηκε η παρουσίαση του λογισμικού μέρους της εργασίας. Έγινε ανάλυση των διάφορων βιβλιοθηκών, γλωσσών σήμανσης / προγραμματισμού, καθώς και του κώδικα της εργασίας, δηλαδή της λογικής που ακολουθήθηκε προκειμένου να ρυθμιστεί ορθά το σύστημα και να κατασκευαστεί ένα πλήρως λειτουργικό χειριστήριο, με το οποίο ο χρήστης μπορεί να ελέγξει την ένταση του ήχου και τα decibels από τα μπάσα, τα μεσαία και τα πρίμα.

Στο επόμενο κεφάλαιο ακολουθεί η παρουσίαση του τρόπου λειτουργίας του ενισχυτή, οι μετρήσεις που έγιναν και τα συμπεράσματα που προέκυψαν, καθώς και διάφοροι τρόποι με τους οποίους το συγκεκριμένο σύστημα θα μπορούσε να βελτιωθεί.

Κεφάλαιο 3ο: Αποτελέσματα

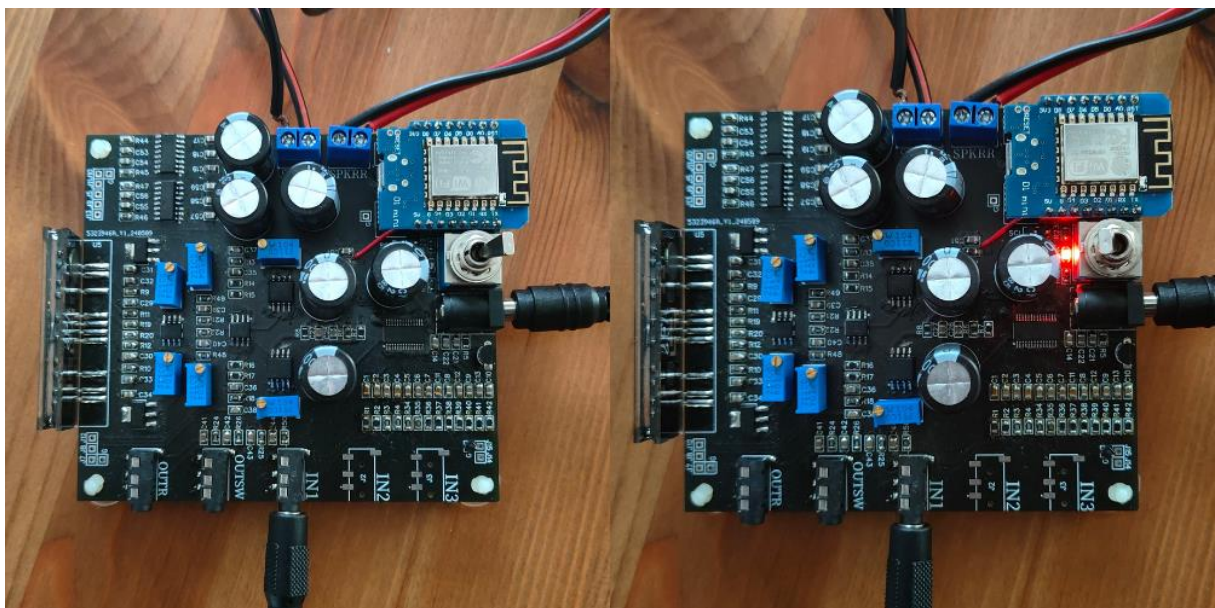
3.1 Εισαγωγή

Στο τρίτο και τελευταίο κεφάλαιο της εργασίας γίνεται η παρουσίαση των αποτελεσμάτων της. Στις πρώτες υποενότητες του κεφαλαίου γίνεται η περιγραφή του τρόπου λειτουργίας του ενισχυτή, καταγράφονται οι μετρήσεις που έγιναν και αναπτύσσονται τα συμπεράσματα που προέκυψαν. Τέλος, αναπτύσσονται οι διάφοροι τρόποι με τους οποίους θα μπορούσε η συγκεκριμένη εργασία να βελτιωθεί, τόσο σε υλικό όσο και σε λογισμικό επίπεδο.

3.2 Τρόπος Λειτουργίας του Ενισχυτή

Προκειμένου να λειτουργήσει ο συγκεκριμένος ενισχυτής, ο χρήστης θα χρειαστεί να ακολουθήσει μερικά πολύ απλά βήματα. Όσον αφορά το υλικό κομμάτι του ενισχυτή:

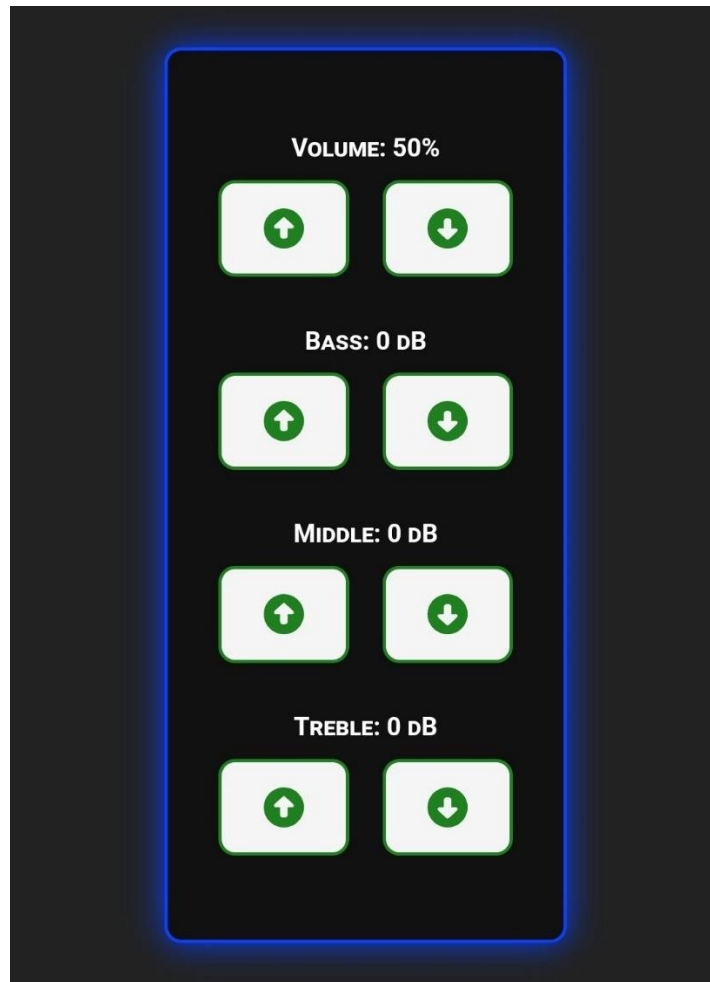
- Σύνδεση καλωδίων: Σύνδεση του τροφοδοτικού στη πρίζα και του αρσενικού jack 3.5mm στη θηλυκή θύρα jack της συσκευής από την οποία θα αναπαραχθεί ο ήχος.
- Άνοιγμα διακόπτη: Ενεργοποίηση του διακόπτη που βρίσκεται επάνω στην πλακέτα ώστε να ενεργοποιηθεί το σύστημα και το LED να ανάψει ως ένδειξη λειτουργίας του ενισχυτή.



Εικόνα 3.1: Διαδικασία ενεργοποίησης του ενισχυτή.

Εφόσον ανάψει το LED, ο χρήστης θα χρειαστεί είτε να συνδεθεί σε μία συσκευή της οποίας τα WiFi διαπιστευτήρια έχουν ενσωματωθεί στον κώδικα του συστήματος, είτε να ενεργοποιήσει τη λειτουργία hotspot από το κινητό του τηλέφωνο και να κάνει τις απαραίτητες ρυθμίσεις, ώστε να ταυτίζονται τα διαπιστευτήρια που είναι ενσωματωμένα στον κώδικα με τα hotspot διαπιστευτήρια που εισάγει ο χρήστης στο κινητό του τηλέφωνο. Εφόσον ολοκληρωθεί η σύνδεση με επιτυχία, ο χρήστης χρειάζεται να πληκτρολογήσει την αντίστοιχη στατική IP που είναι ενσωματωμένη στον κώδικα του συστήματος στο επιθυμητό πρόγραμμα περιήγησης διαδικτύου που βρίσκεται εγκατεστημένο στη συσκευή του.

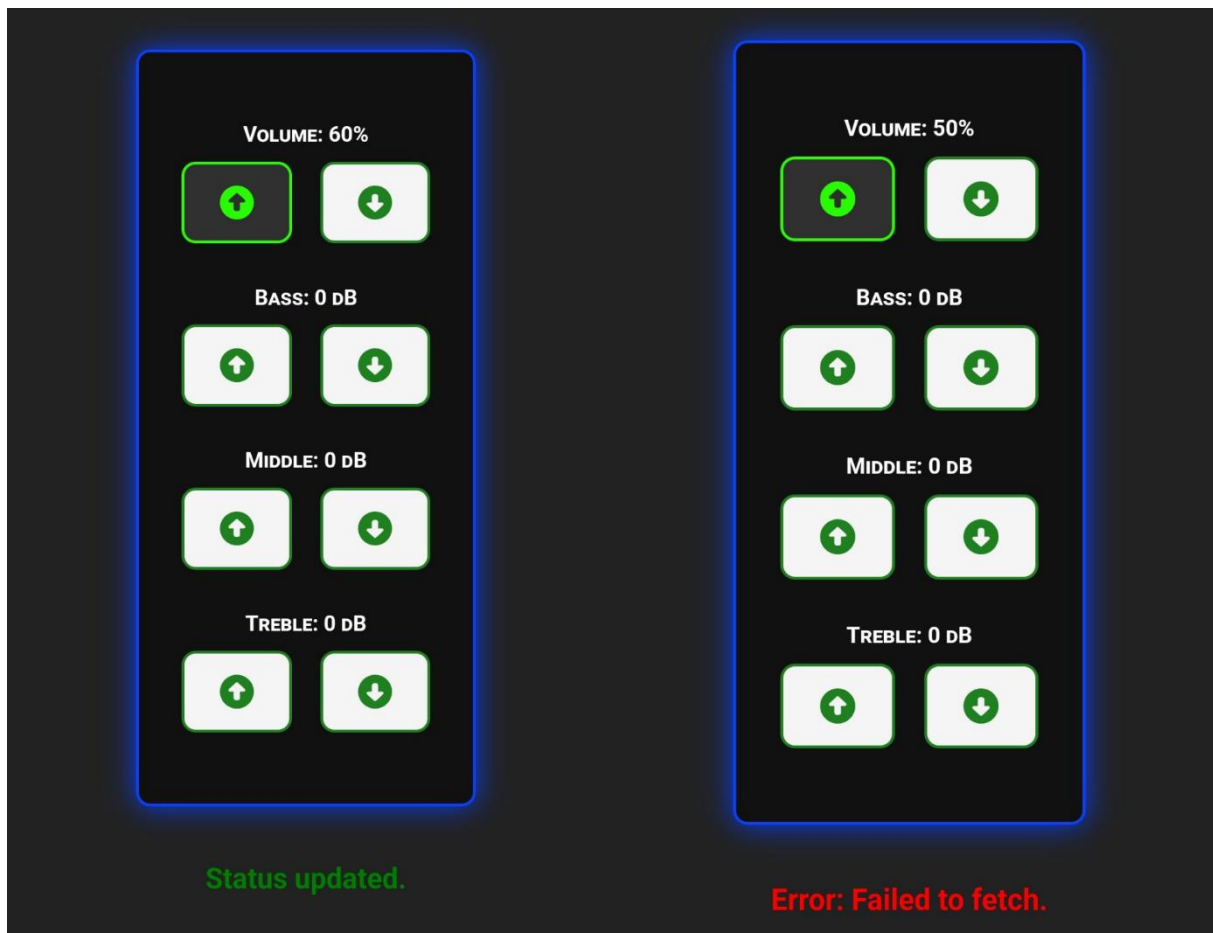
Μόλις ανταποκριθεί το πρόγραμμα περιήγησης, θα εμφανιστεί η αρχική μορφή της σελίδας του χειριστηρίου με τις προεπιλεγμένες (προτεινόμενες) τιμές για την ένταση του ήχου, τα μπάσα, τα μεσαία και τα πρίμα. Ο χρήστης μπορεί να τροποποιήσει τις τιμές χρησιμοποιώντας τα κουμπιά αυξομείωσης της κάθε κατηγορίας.



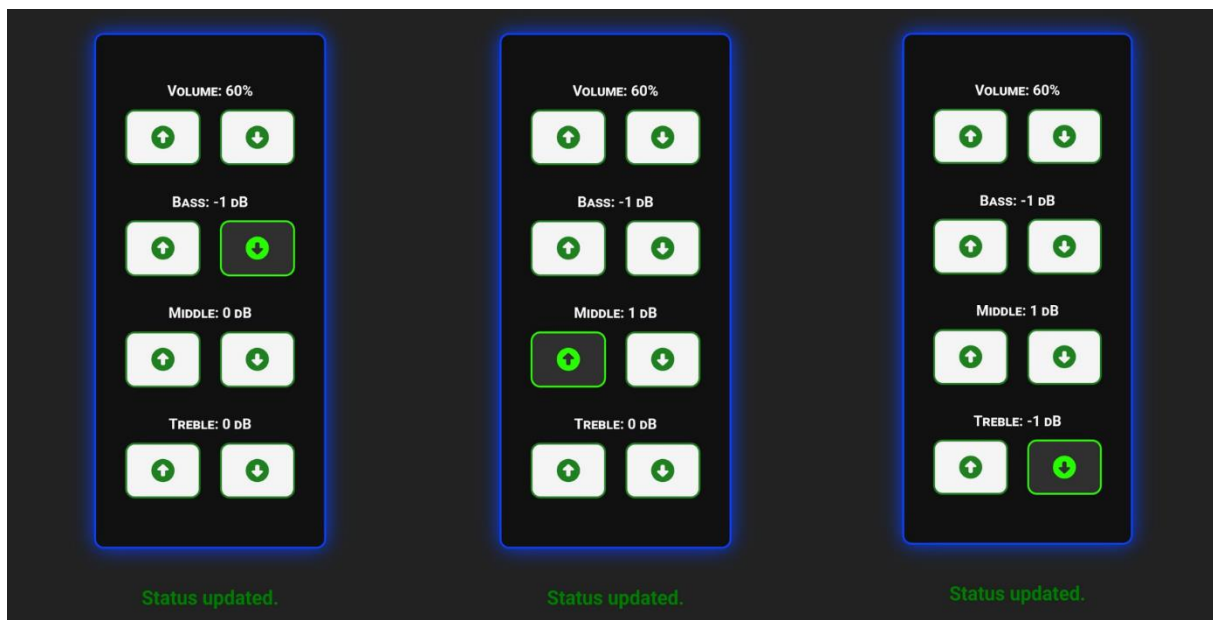
Εικόνα 3.2: Αρχική μορφή της σελίδας του χειριστηρίου με τις προεπιλεγμένες τιμές.

Μόλις ο χρήστης πατήσει ένα από τα κουμπιά, θα λάβει τις εξής ανατροφοδοτήσεις:

- Θα αλλάξει το χρώμα του κουμπιού που επέλεξε,
- Θα αλλάξει η προεπιλεγμένη τιμή που εμφανίζεται στο λεκτικό επάνω από το κουμπί, και θα αντικατασταθεί με τη νέα τιμή,
- Θα εμφανιστεί μήνυμα επιτυχίας αποστολής δεδομένων «Status updated.» κάτω από το χειριστήριο εφόσον η αλλαγή της τιμής είναι επιτυχής, ή θα εμφανιστεί μήνυμα αποτυχίας αποστολής δεδομένων «Error: Failed to fetch.» κάτω από το χειριστήριο εφόσον η αλλαγή της τιμής είναι ανεπιτυχής,
- Θα αλλάξει η τιμή στο σύστημα και θα υπάρξει τροποποίηση στον ήχο.



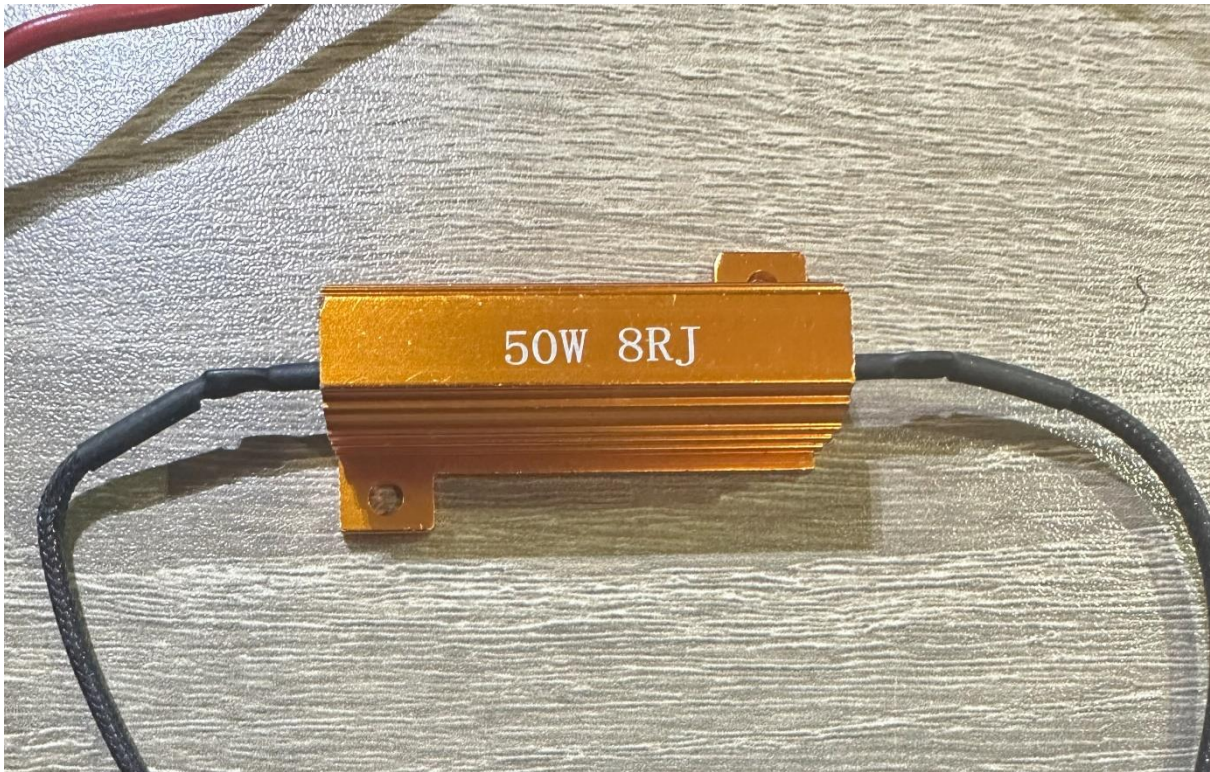
Εικόνα 3.3: Μήνυμα επιτυχίας αποστολής δεδομένων και μήνυμα αποτυχίας.



Εικόνα 3.4: Χρήση του χειριστηρίου και τροποποίηση των τιμών.

3.3 Μετρήσεις

Για τον έλεγχο της ορθής λειτουργίας του ενισχυτή, μετρήθηκε η ισχύς, η ολική παραμόρφωση, το εύρος ζώνης και η αντίσταση εισόδου του. Στις μετρήσεις, για λόγους ακρίβειας, χρησιμοποιήθηκε μια αντίσταση 8 Ω (50 W) ως φορτίο εξόδου, καθώς η αντίσταση των ηχείων ήταν πολύ μικρή (4 Ω). Τα όργανα που χρησιμοποιήθηκαν για τις μετρήσεις ήταν ένα πολύμετρο, ένας παλμογράφος και μία παλμογεννήτρια.



Εικόνα 3.5: Η αντίσταση που χρησιμοποιήθηκε για τις μετρήσεις.

Για τον υπολογισμό της τάσης έγινε μέτρηση της V_{rms} στην έξοδο του ενισχυτή με τη χρήση πολυμέτρου. Η V_{rms} μετρήθηκε στα 4.98 V \approx 5 V. Για τον υπολογισμό της τάσης πραγματοποιήθηκε η εξής πράξη:

$$V_{RMS} = \frac{V_o}{\sqrt{2}} \Rightarrow V_o = V_{RMS} * \sqrt{2} \Rightarrow V_o = 5V * \sqrt{2} \Rightarrow V_o = 7.07 V$$

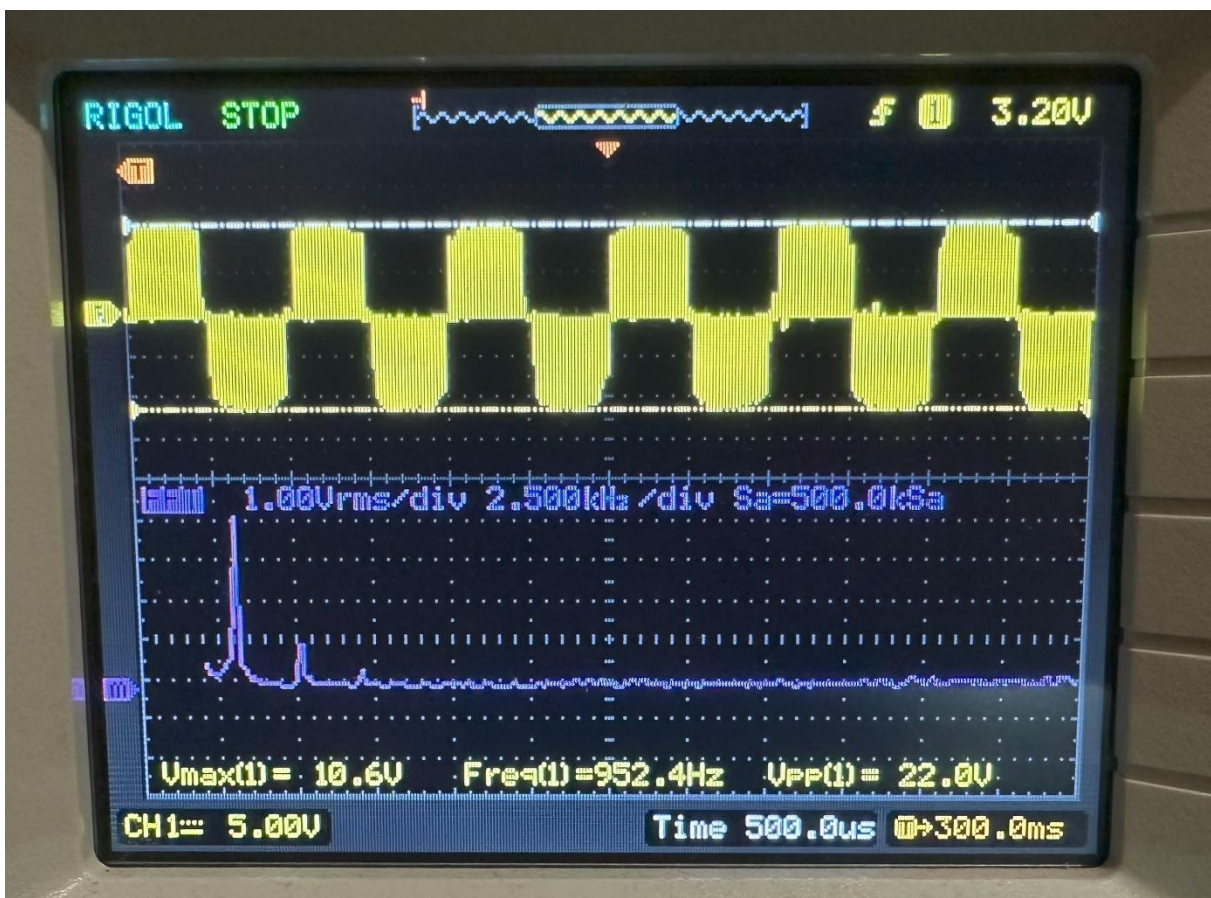
Στη συνέχεια, χρησιμοποιώντας τον νόμο του Ohm, υπολογίζουμε την ισχύ:

$$P = V_o * I \Rightarrow P = \frac{V_o^2}{R} \Rightarrow P = \frac{7.07^2 V}{8 \Omega} \Rightarrow P = 6.24 W$$



Εικόνα 3.6: Μέτρηση του V_{rms} .

Η ολική παραμόρφωση μετρήθηκε με την ίδια αντίσταση και με τη χρήση παλμογράφου και παλμογεννήτριας, μέσω της οποίας εισάγεται σήμα 1 kHz στην είσοδο του ενισχυτή.



Εικόνα 3.7: Λάθος μέτρηση ολικής παραμόρφωσης & σήματος FFT.

Από το αποτέλεσμα παρατηρείται ότι το σήμα είναι λάθος, το οποίο προέκυψε από το γεγονός ότι οι ενισχυτές εξόδου τάξης D είναι ενισχυτές ισχύος και βγάζουν σήμα το οποίο μιμείται το ψηφιακό σήμα (γρήγορη εναλλαγή μεταξύ καταστάσεων λειτουργίας για αποδοτικότερη ενίσχυση), ενώ λείπει από την κατασκευή τους ένα φίλτρο εξόδου LC. Προκειμένου να γίνει η σωστή μέτρηση, κρίθηκε αναγκαία η κατασκευή χαμηλοπερατού φίλτρου LC με ανώτατη επιτρεπόμενη συχνότητα 40 kHz. Για τον υπολογισμό των τιμών του πηνίου και του πυκνωτή πραγματοποιήθηκε η εξής σχέση:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \Rightarrow LC = \frac{1}{(2\pi f_c)^2} \Rightarrow LC = \frac{1}{(2\pi * 40000)^2} \Rightarrow LC = 1.58 * 10^{-11}$$

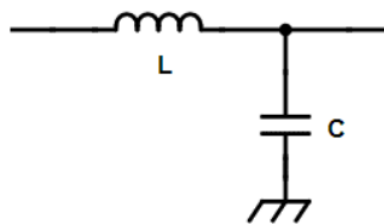
Λαμβάνοντας υπόψιν την παραπάνω μαθηματική σχέση και τα πηνία που βρίσκονται αυτή τη στιγμή στην αγορά, κρίθηκε σωστότερη η χρήση ενός πηνίου 100 uH 5A. Από την παραπάνω σχέση υπολογίζεται ότι ο πυκνωτής που χρειάζεται για την κατασκευή του LC φίλτρου είναι:

$$LC = 1.58 * 10^{-11} \Rightarrow C = \frac{1.58 * 10^{-11}}{L} \Rightarrow C = \frac{1.58 * 10^{-11}}{100 * 10^{-6}} \Rightarrow C = 158 \text{ nF}$$

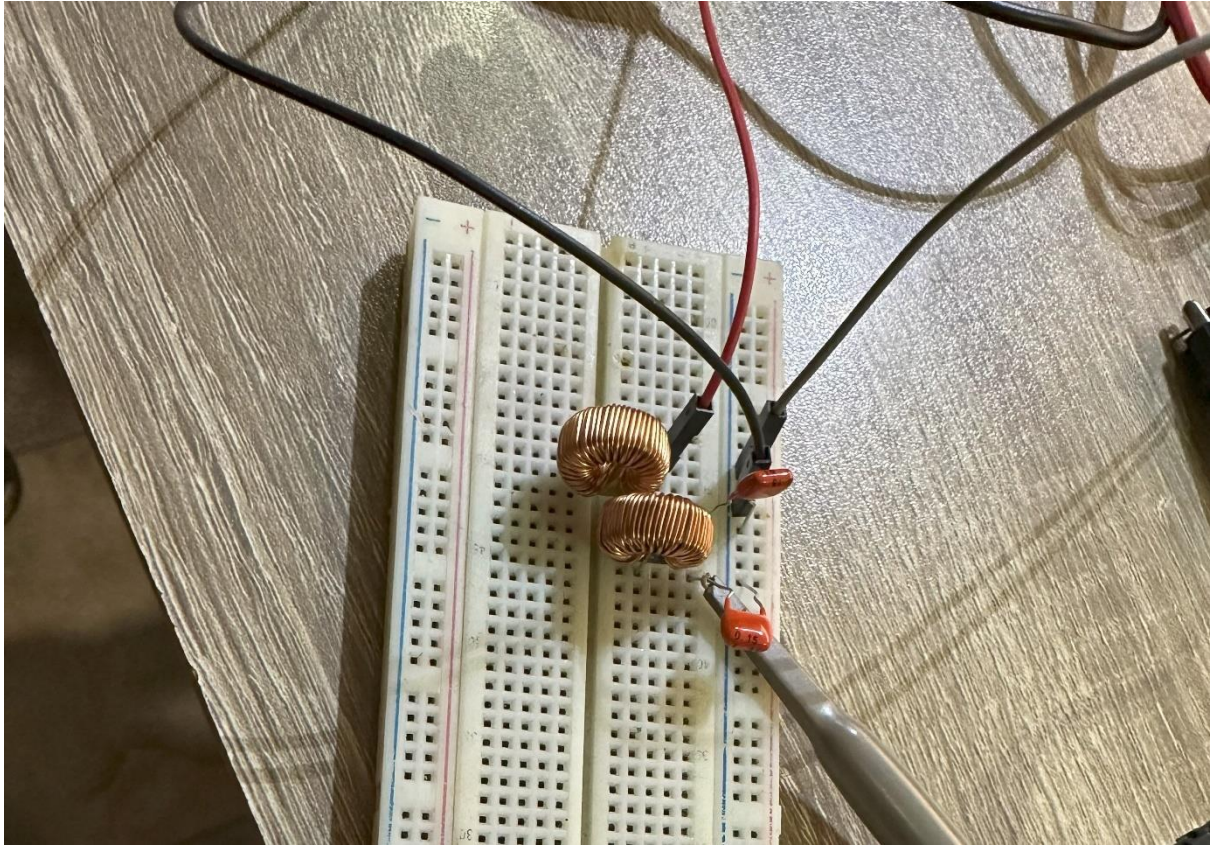
Λόγω του γεγονότος ότι δεν υπάρχει πυκνωτής με 158 nF, έγινε αγορά του πλησιέστερου πυκνωτή 150 nF. Για τον υπολογισμό της πραγματικής συχνότητας αποκοπής του φίλτρου LC, πραγματοποιήθηκε η εξής πράξη:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \Rightarrow f_c = \frac{1}{2\pi\sqrt{100 * 10^{-6} * 150 * 10^{-9}}} \Rightarrow f_c = 41.093 \approx 41 \text{ kHz}$$

Το χαμηλοπερατό φίλτρο LC που κατασκευάστηκε έχει ανώτατη συχνότητα αποκοπής 41 kHz.



Εικόνα 3.8: Σχέδιο χαμηλοπερατού φίλτρου LC [107].



Εικόνα 3.9: Κατασκευή του χαμηλοπερατού φίλτρου LC.

Μετά την κατασκευή του φίλτρου, παρατηρήθηκε ότι η έξοδος ήταν ψαλιδισμένη, συνεπώς τροποποιήθηκαν τα ποτενσιόμετρα R27 και R32. Ο ψαλιδισμός προέκυπτε διότι το κύκλωμα λάμβανε μεγαλύτερη τάση στην είσοδο.

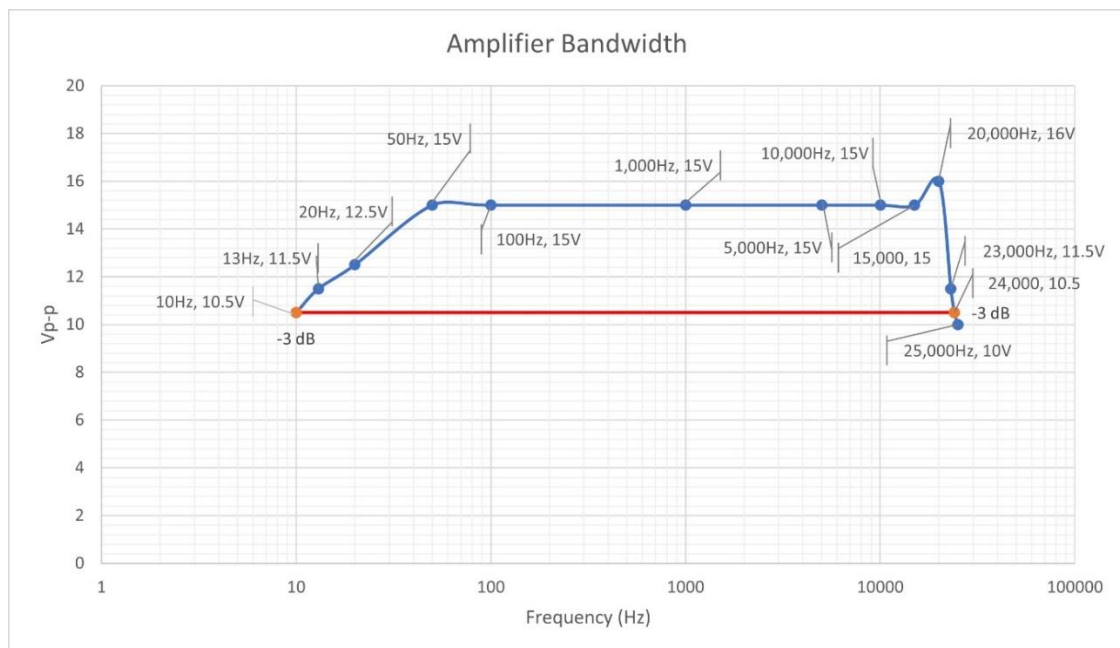


Εικόνα 3.10: Μέτρηση ολικής παραμόρφωσης & σήματος FFT με ψαλιδισμό και σωστή μέτρηση ολικής παραμόρφωσης & σήματος FFT.

Η μέτρηση του εύρους ζώνης του ενισχυτή πραγματοποιήθηκε με τη χρήση παλμογράφου και παλμογεννήτριας. Για τον υπολογισμό έγινε εισαγωγή διαφορετικών συχνοτήτων στην είσοδο και μέτρηση της τάσης εξόδου. Οι διαφορετικές συχνοτήτες και οι αντίστοιχες τάσεις εξόδου φαίνονται στον Πίνακα 3.1.

Πίνακας 3.1. Συχνότητες εισόδου και αντίστοιχες τάσεις εξόδου.

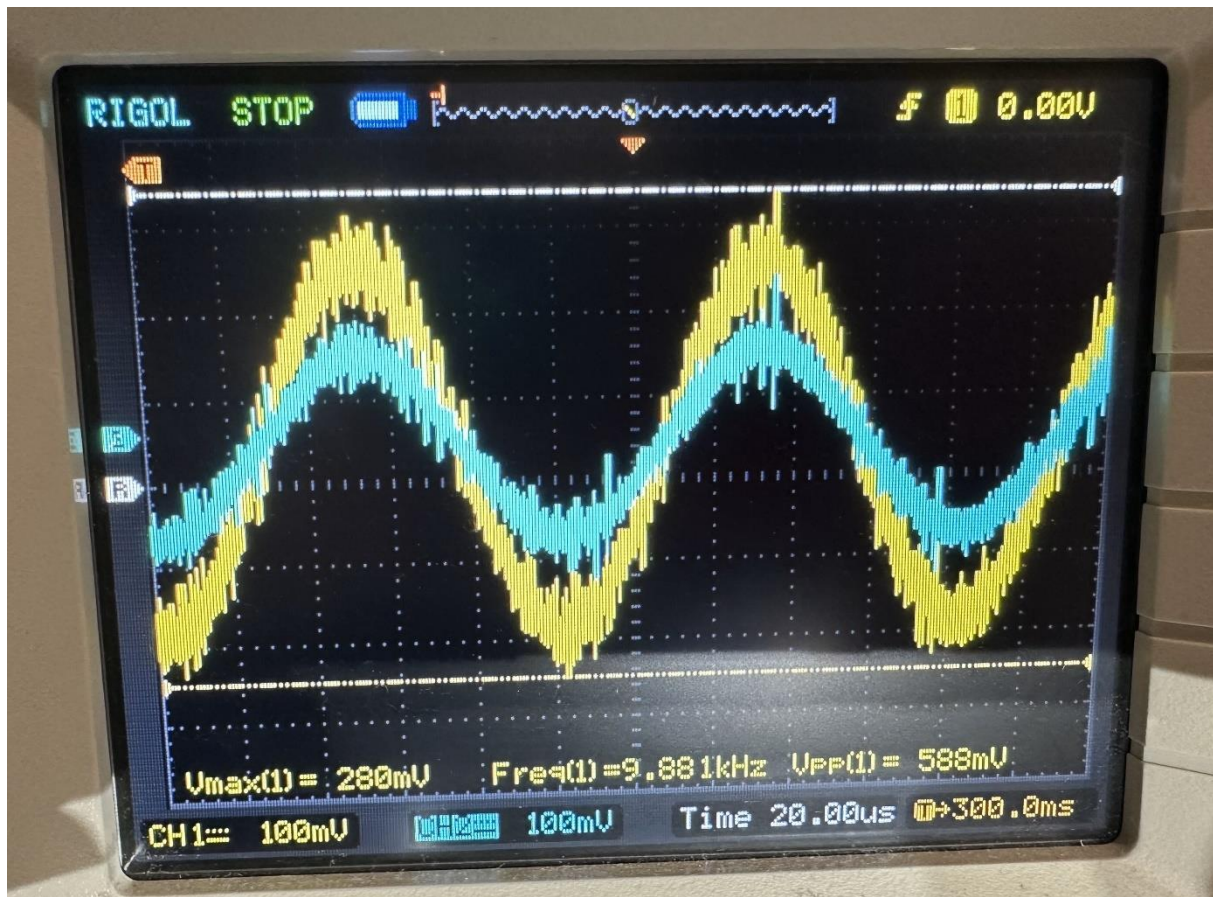
Συχνότητα (Hz)	V _{p-p} (V)
10	10.5
13	11.5
20	12.5
50	15
100	15
1000	15
5000	15
10000	15
15000	15
20000	16
23000	11.5
24000	10.5
25000	10



Εικόνα 3.11: Συχνότητες εισόδου και αντίστοιχες τάσεις εξόδου.

Παρατηρήθηκε από το λογαριθμικό σχεδιάγραμμα ότι το εύρος ζώνης είναι περίπου 10 Hz έως 24 kHz. Λόγω της τιμής της μεσαίας ζώνης (15 V), οποιαδήποτε τιμή κάτω του 70% (10.5 V) της τάσης θεωρείται συχνότητα αποκοπής.

Για τη μέτρηση της σύνθετης αντίστασης εισόδου Z_{in} συνδέθηκε ένα ποτενσιόμετρο σε σειρά με την είσοδο, δημιουργώντας έναν διαιρέτη τάσης, και έγινε εισαγωγή σήματος 1 kHz με την χρήση της παλμογεννήτριας. Το ποτενσιόμετρο τροποποιήθηκε έτσι ώστε η αντίσταση του να γίνει ίση με τη σύνθετη αντίσταση εισόδου, το οποίο έγινε αντιληπτό όταν παρατηρήθηκαν 0.5 kHz στο σημείο μέτρησης ανάμεσα στο ποτενσιόμετρο και τη σύνθετη αντίσταση εισόδου.



Εικόνα 3.12: Μέτρηση Z_{in} πριν και μετά το ποτενσιόμετρο.

Για τη μέτρηση της σύνθετης αντίστασης εξόδου Z_{out} έγινε εισαγωγή σήματος 1 kHz στην είσοδο. Έγινε μέτρηση στην έξοδο χωρίς φορτίο και μετρήθηκαν 12 V. Στη συνέχεια συνδέθηκε ένα φορτίο 8 Ω, το οποίο δημιούργησε έναν διαιρέτη τάσης. Μετρήθηκαν 10 V ανάμεσα στη Z_{out} και στο φορτίο των 8 Ω.

$$\begin{aligned}
 V_{OUT} &= \frac{8 \Omega}{8 \Omega + Z_{OUT}} * V_{IN} \Rightarrow Z_{OUT} = \frac{8 \Omega * V_{IN} - 8 \Omega * V_{OUT}}{V_{OUT}} \Rightarrow Z_{OUT} \\
 &= \frac{8 \Omega * 12 V - 8 \Omega * 10 V}{10 V} \Rightarrow Z_{OUT} = 1.6 \Omega
 \end{aligned}$$

3.4 Τρόποι Βελτίωσης της Εργασίας

Η συγκεκριμένη εργασία μπορεί να λάβει πολλές αναβαθμίσεις, τόσο στο υλικό (hardware), όσο και στο λογισμικό (software) κομμάτι. Μερικές από αυτές τις αναβαθμίσεις για το υλικό μέρος της εργασίας είναι οι εξής:

- Επανασχεδιασμός της πλακέτας και κατασκευή με νέα εξαρτήματα, για παράδειγμα:
 - Καλύτερο ψηφιακό επεξεργαστή σήματος, με λιγότερο θόρυβο στις εξόδους του).
 - Προσθήκη RF φίλτρων στην είσοδο και buffer πολύ μικρών παραμορφώσεων, τα οποία θα τοποθετηθούν στο κύκλωμα πριν το Nutube.
 - Καλύτερο τροφοδοτικό με λιγότερο θόρυβο.
- Προσθήκη περισσότερων ενισχυτών εξόδου τάξης D, ή προσθήκη ενισχυτών με περισσότερες εξόδους και μεγαλύτερη τάση εξόδου, ώστε να συνδεθούν περισσότερα και πιο δυνατά ηχεία.
- Διόρθωση της σύνδεσης των subwoofer και rear ηχείων, καθώς έχουν συνδεθεί απευθείας με τον ψηφιακό επεξεργαστή σήματος (DSP), ενώ θα έπρεπε να βρίσκονται στη συνδεσμολογία πριν από το κύκλωμα του Nutube.
- Προσθήκη φυσικών κουμπιών (knob) με τα οποία θα ήταν εφικτή η ρύθμιση της έντασης του ήχου και ανατροφοδότηση της νέας τιμής στον μικροελεγκτή.
- Ενσωμάτωση της πλακέτας σε ένα δοχείο για ευκολότερη μεταφορά και μεγαλύτερη ασφάλεια.

Όσον αφορά τις βελτιώσεις για το λογισμικό μέρος της εργασίας:

- Προκειμένου να μπορέσει ο χρήστης να αξιοποιήσει όλες τις δυνατότητες που μπορεί να προσφέρει ο ψηφιακός επεξεργαστής σήματος, είναι εφικτό να προστεθούν περισσότερα κουμπιά, τα οποία, πέρα από τον έλεγχο της έντασης του ήχου και των τιμών από τα μπάσα, τα μεσαία και τα πρίμα, θα μπορούσαν να τροποποιούν άλλες ρυθμίσεις και τιμές, όπως:
 - Ενεργοποίηση και απενεργοποίηση της λειτουργίας Απαλής Σίγασης (Soft Mute),
 - Ενεργοποίηση και απενεργοποίηση της λειτουργίας Απαλό Βήμα (Soft Step),
 - Διαχείριση της τιμής της Ηχηρότητας (Loudness),
 - Διαχείριση του Subwoofer.
- Προς διευκόλυνση του χρήστη, θα ήταν εφικτό να τροποποιηθεί ο μικροελεγκτής με τέτοιο τρόπο ώστε ο χρήστης να συνδέεται σε αυτόν, έναντι του τρόπου με τον οποίο λειτουργεί τώρα, δηλαδή σαν σταθμός ο οποίος συνδέεται στο δίκτυο (WiFi / Hotspot) του χρήστη.
- Προς διευκόλυνση του χρήστη, θα ήταν εφικτό να υλοποιηθεί μια εφαρμογή κινητού, μέσω της οποίας ο χρήστης θα μπορούσε να αξιοποιήσει όλες τις ρυθμίσεις και τις λειτουργίες που προσφέρει ο ψηφιακός επεξεργαστής σήματος. Η συγκεκριμένη εφαρμογή θα παρείχε τη δυνατότητα στον χρήστη να αποθηκεύσει τις προσωπικές του ρυθμίσεις και να ξεκινάει τη συσκευή κάθε φορά με τις αντίστοιχες αρχικοποιήσεις τιμών, ώστε να του προσφέρει μια πιο προσωποποιημένη εμπειρία.
- Προς διευκόλυνση του χρήστη, η εικαστική δομή του χειριστηρίου θα μπορούσε να είναι τροποποιήσιμη από αυτόν, είτε στον τομέα των χρωμάτων, είτε στον τομέα της σειράς εμφάνισης των κουμπιών βάση της προσωπικής του χρήσης.
- Για μεγαλύτερη ακρίβεια και προς αποφυγή σφαλμάτων, θα ήταν εφικτό να επιστρέφονται προς εκτύπωση οι νέες τιμές των καταστάσεων πίσω στον μικροελεγκτή, έναντι της αποστολής των νέων τιμών από τον μικροελεγκτή και της εκτύπωσης των δεδομένων χωρίς την επαλήθευση της αλλαγής.

3.5 Συμπεράσματα

Με το πέρας της συγκεκριμένης εργασίας έχει ολοκληρωθεί η κατασκευή ενός πλήρως λειτουργικού στερεοφωνικού ενισχυτή βασιζόμενου στο νέο τύπο λυχνιών Nutube. Η έρευνα για την υλοποίηση της εργασίας, η κατασκευή του υλικού μέρους, η συγγραφή του λογισμικού μέρους και η αποτύπωση της

διαδικασίας στο συγκεκριμένο έγγραφο είχε συνολική διάρκεια περίπου 10 μηνών, από τον Απρίλιο του 2024 μέχρι τον Ιανουάριο του 2025. Μέσα σ' αυτό το διάστημα, οι φοιτητές μελέτησαν τους διάφορους τύπους ενισχυτών, τον τρόπο κατασκευής τους, τον νέο τύπο λυχνιών Nutube, τις δυνατότητες που μπορεί να έχει ένα ολοκληρωμένο λειτουργικό σύστημα ενισχυτή, τον έλεγχο που μπορούν να ασκήσουν είτε μέσω υλικών εξαρτημάτων είτε μέσω λογισμικών εφαρμογών, τη σημασία της κατασκευής ενός χειριστηρίου το οποίο είναι φιλικό προς το χρήστη, καθώς και τις γλώσσες προγραμματισμού και σήμανσης που είναι απαραίτητες για την υλοποίηση τέτοιων εφαρμογών.

Κατά την υλοποίηση του υλικού μέρους οι φοιτητές συνειδητοποίησαν τα διάφορα οφέλη της κατασκευής συστημάτων με καινοτόμα εξαρτήματα και μεθόδους κατασκευής, τον βαθμό με τον οποίο έχει εξελιχθεί η τεχνολογία και η αγορά, καθώς και την ευκολία με την οποία μπορεί ένας ηλεκτρονικός να αναπτύξει ένα τέτοιο σύστημα, εφόσον κάνει τη σωστή έρευνα. Κατά την υλοποίηση του λογισμικού μέρους οι φοιτητές συνειδητοποίησαν το βάθος της ευελιξίας που προσφέρουν οι γλώσσες προγραμματισμού στον έλεγχο τέτοιων συστημάτων, σε αντίθεση με τον περιορισμό που μπορεί να προσφέρει ένα σύστημα του οποίου ο έλεγχος είναι εφικτός μόνο μέσω φυσικών κουμπιών.

Παρ' ότι το τελικό αποτέλεσμα ήταν θετικό, οι φοιτητές συνάντησαν διάφορες προκλήσεις κατά τη διάρκεια της κατασκευής του υλικού μέρους, κάτι που είχε ως αποτέλεσμα την τροποποίηση του αρχικού σχεδιασμού του κυκλώματος και την χρήση ανταλλακτικών εξαρτημάτων, μετά από αποτυχημένες απόπειρες. Η δοκιμαστική χρήση του ενισχυτή με διάφορους ήχους και τιμές, καθώς και οι τελικές μετρήσεις έδειξαν ότι υπάρχουν αρκετές βελτιστοποιήσεις οι οποίες θα μπορούσαν να εφαρμοστούν στο ηλεκτρονικό κύκλωμα, προκειμένου να υπάρξει ακόμη μεγαλύτερη μείωση του θορύβου που παράγεται από πολλά από τα εξαρτήματα που χρησιμοποιούνται.

Στο λογισμικό μέρος της εργασίας, οι φοιτητές ήρθαν αντιμέτωποι με τη μελέτη και την εφαρμογή διάφορων γλωσσών προγραμματισμού και σήμανσης, καθώς και βιβλιοθηκών για την ορθή λειτουργία των μικροελεγκτών και τον φυσιολογικό έλεγχο του ψηφιακού επεξεργαστή σήματος και των εξαρτημάτων στα οποία συνδέονται. Οι φοιτητές εμπέδωσαν τη λογική πίσω από τον προγραμματισμό των συστημάτων που σχετίζονται με ενισχυτές και εφάρμοσαν εξειδικευμένες συναρτήσεις και μεθόδους τροποποίησης και παρουσίασης δεδομένων, ενώ αξιοποίησαν διάφορες μεθόδους αποσφαλμάτωσης κατά τη διάρκεια της συγγραφής του κώδικα, προκειμένου να εντοπίσουν σημεία στα οποία το λογισμικό που κατασκεύαζαν υπολειπούντο.

Καθ' όλη τη διάρκεια, καθώς και μετά το πέρας της ολοκλήρωσης της, οι φοιτητές εντόπισαν ποικίλες αναβαθμίσεις τις οποίες θα μπορούσε να λάβει μια μελλοντική εκδοχή της παρούσας εργασίας. Μεταξύ άλλων, οι συγκεκριμένες αναβαθμίσεις αφορούν επιπρόσθετες λειτουργίες οι οποίες παρέχονται ήδη από τα εξαρτήματα του συστήματος, οι οποίες προσφέρουν καλύτερο έλεγχο από την πλευρά του χρήστη, ένα φιλικότερο περιβάλλον χειρισμού, καθώς και εναλλακτικές μεθόδους σύνδεσης στο σύστημα και αποθήκευσης δεδομένων για μια πιο εξατομικευμένη χρήση.

Πέρα από το κομμάτι της κατασκευής του ενισχυτή, η ενασχόληση με ένα τόσο μεγάλο, ομαδικό έργο, οδήγησε τους φοιτητές να αποκτήσουν εμπειρία επάνω στην οργάνωση και τον σχεδιασμό του, τις απαιτήσεις μιας λεπτομερούς και περιεκτικής συγγραφής ενός τεχνικού κειμένου, αλλά και την αντιμετώπιση της πίεσης που μπορούν να επιφέρουν οι προθεσμίες παράδοσης.

Συνοψίζοντας, με το πέρας της συγκεκριμένης εργασίας, οι φοιτητές έχουν αναπτύξει πλέον μια σφαιρική αντίληψη των σχεδιαστικών απαιτήσεων που μπορεί να έχει ένα τέτοιο σύστημα, των προβλημάτων που ενδέχεται να προκύψουν κατά τη διάρκεια της κατασκευής του υλικού μέρους και της συγγραφής του κώδικα για το λογισμικό μέρος, καθώς και των διάφορων πλεονεκτημάτων που

μπορεί να έχει ένα τόσο σύγχρονο σύστημα, αλλά και η σε βάθος ενασχόληση με τον σχεδιασμό και την κατασκευή του.

3.6 Επίλογος

Στο συγκεκριμένο κεφάλαιο ολοκληρώθηκε η παρουσίαση των αποτελεσμάτων της εργασίας. Έγινε ανάλυση του τρόπου λειτουργίας του ενισχυτή, των μετρήσεων που έγιναν και των συμπερασμάτων που προέκυψαν, καθώς και των διάφορων τρόπων με τους οποίους θα μπορούσε να βελτιωθεί η συγκεκριμένη εργασία. Με αυτό το κεφάλαιο κλείνει η εργασία με τίτλο «Σχεδίαση και κατασκευή στερεοφωνικού ενισχυτή βασιζόμενου στο νέο τύπο λυχνιών Nutube».

Ακολουθεί η βιβλιογραφία και τα παραρτήματα με τα datasheet από τα εξαρτήματα που χρησιμοποιήθηκαν, καθώς και ο πλήρης κώδικας της εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 7th ed. New York: Oxford University Press, 2014.
- [2] A. Malvino and D. J. Bates, *Electronic Principles*, 8th ed. New York: McGraw Hill, 2016.
- [3] Electronics Tutorials, "Amplifier Classes," Electronics Tutorials. [Online]. Available: <https://www.electronics-tutorials.ws/amplifier/amplifier-classes.html>. [Accessed: 2025].
- [4] Sourav Gupta, "Classes of Power Amplifiers," *Circuit Digest*, 2018. [Online]. Available: <https://circuitdigest.com/tutorial/classes-of-power-amplifier-explained>.
- [5] Diodes Incorporated, "PAM8320: 20W Mono Class D Audio Amplifier," PAM8320 datasheet, Oct. 2013.
- [6] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. 3rd Ed.
- [7] STMicroelectronics, "TDA7718N: 3 Band Car Audio Processor," TDA7718N datasheet, Sept. 2013.
- [8] Monolithic Power Systems, "Voltage Regulator Types and Working Principles," *Monolithic Power Systems*. [Online] Available: <https://www.monolithicpower.com/en/learning/resources/voltage-regulator-types>. [Accessed: 2025].
- [9] Advanced Monolithic Systems, "AMS1117 1A Low Dropout Voltage Regulator," AMS1117 datasheet. [Accessed: 2024].
- [10] STMicroelectronics, "L78L: Positive Voltage Regulators," L78L datasheet, Apr. 2021.
- [11] Linear Systems, "LSK389 A/B/C/D: Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier," LSK389 datasheet, Jul. 2021.
- [12] Linear Systems, LSK389 Appl. Note, pp.1-13.
- [13] S. Franko, *Design with Operational Amplifiers and Analog Integrated Circuits*, 4th ed. New York: McGraw Hill, 2015.
- [14] Texas Instruments, "NE5532, NE5532A: Dual Low-Noise Operational Amplifiers," NE5532 datasheet, 2000.
- [15] S. Popovs, *Modern Vacuum Tube Amplifier: VFD as a triode*, Helsinki: Metropolia University of Applied Sciences, 2017.
- [16] M. Pultzer, *Practical Electronics*, vol. 50, no. 1, Brighton: Electron Publishing Limited, Jan. 2021.
- [17] Korg, Nutube Appl. Note, pp. 2-18.
- [18] WEMOS, "LOLIN D1 mini," WEMOS. [Online]. Available: https://www.wemos.cc/en/latest/d1/d1_mini.html. [Accessed: 2024].
- [19] M. Schwartz, *Internet of Things with ESP8266*, Birmingham: Packt Publishing Ltd, 2016.
- [20] WEMOS, "D1 Boards," WEMOS. [Online]. Available: <https://www.wemos.cc/en/latest/d1/index.html>. [Accessed: 2024].

- [21] REALPOY, "ESP8266 Serial WIFI Module ESP-12E," D1 Mini ESP-12F datasheet. [Accessed: 2024]
- [22] Espressif Systems, "ESP8266EX," ESP8266EX datasheet, 2023.
- [23] EasyEDA, "About EasyEDA," EasyEDA. [Online]. Available: <https://easyeda.com/page/about>. [Accessed: 2025].
- [24] EasyEDA, "Introduction to EasyEDA," EasyEDA, 2024. [Online]. Available: <https://docs.easyeda.com/en/Introduction/Introduction-to-EasyEDA>.
- [25] Arduino, "About Arduino," Arduino, 2021. [Online]. Available: <https://www.arduino.cc/en/about>.
- [26] Wiring, "Introduction," Wiring. [Online]. Available: <https://wiring.org.co/about.html>. [Accessed: 2024].
- [27] Arduino, "What is Arduino," Arduino, 2018. [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>.
- [28] Arduino, "Logo," Arduino. [Online]. Available: <https://ui-elements-dev.arduino.cc/logo>. [Accessed: 2024].
- [29] Arduino, "Arduino Hardware," Arduino, 2022. [Online]. Available: <https://www.arduino.cc/en/hardware>.
- [30] K. Söderby & J. Hylén, "Getting Started with Arduino IDE 2," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/>.
- [31] Arduino, "indexOf()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/variables/data-types/stringObject/Functions/indexOf/>.
- [32] Arduino, "substring()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/variables/data-types/stringObject/Functions/substring/>.
- [33] Arduino, "length()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/variables/data-types/stringObject/Functions/length/>.
- [34] Arduino, "String()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/variables/data-types/stringObject/>.
- [35] Arduino, "setup()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/structure/sketch/setup/>.
- [36] Arduino, "pinMode()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/digital-io/pinMode/>.
- [37] Arduino, "digitalWrite()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/digital-io/digitalwrite/>.
- [38] Arduino, "sizeof()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/variables/utilities/sizeof/>.
- [39] Arduino, "min()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/math/min/>.
- [40] Arduino, "max()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/math/max/>.

- [41] Arduino, "loop()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/structure/sketch/loop/>.
- [42] Arduino, "delay()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/time/delay/>.
- [43] Arduino, "Serial.begin()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/communication/serial/begin/>.
- [44] Arduino, "Serial.println()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/communication/serial/println/>.
- [45] Arduino, "Serial.print()," Arduino, 2024. [Online]. Available: <https://www.arduino.cc/reference/en/language/functions/communication/serial/print/>.
- [46] Arduino, "Language Reference," Arduino. [Online]. Available: <https://www.arduino.cc/reference/en/>. [Accessed: 2024].
- [47] N. Zambetti, K. Söderby, J. Hylén, "Inter-Integrated Circuit (I2C) Protocol," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/learn/communication/wire/>.
- [48] Arduino, "Wire," Arduino, 2024. [Online]. Available: <https://www.arduino.cc/reference/en/language/functions/communication/wire/>.
- [49] Arduino, "begin()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/communication/wire/begin/>.
- [50] Arduino, "beginTransaction()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/communication/wire/beginTransmission/>.
- [51] Arduino, "endTransmission()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/communication/wire/endTransmission/>.
- [52] Arduino, "write()," Arduino, 2024. [Online]. Available: <https://docs.arduino.cc/language-reference/en/functions/communication/wire/write/>.
- [53] ESP8266 Arduino Core, "Station Class," ESP8266 Arduino Core. [Online]. Available: <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/station-class.html>. [Accessed: 2024].
- [54] Github, "ESP8266WiFi keywords.txt," ESP8266 Community Forum. [Online]. Available: <https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/keywords.txt>. [Accessed: 2024].
- [55] M. Dobrolyubova, M. Kovalenko, O. Statsenko and S. Shevkun, "Module for Automatic Accounting of Consumed Cold and/or Hot Water Based on Wi-Fi Technology," 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2022, pp. 592-596, doi: 10.1109/PICST57299.2022.10238481. keywords: {Meters;Workability;Databases;Data acquisition;Companies;Meter reading;Servers;data acquisition automation;remote access;water meters;Internet technologies },
- [56] Github, "ESPAsyncWebServer README," lacamera. [Online]. Available: <https://github.com/lacamera/ESPAsyncWebServer>. [Accessed: 2024].
- [57] R. Connolly and R. Hoar, Fundamentals of Web Development, New Jersey: Pearson Education, Inc. 2015.
- [58] W3C, "Take Control - Your Web, Your Logo," W3C. [Online]. Available: <https://www.w3.org/html/logo/index.html>. [Accessed: 2024].

- [59] W3Schools, "SVG Tutorial," W3Schools. [Online]. Available: https://www.w3schools.com/graphics/svg_intro.asp. [Accessed: 2024].
- [60] W3Schools, "HTML <html> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_html.asp. [Accessed: 2024].
- [61] W3Schools, "HTML <head> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_head.asp. [Accessed: 2024].
- [62] W3Schools, "HTML <title> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_title.asp. [Accessed: 2024].
- [63] W3Schools, "HTML <meta> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_meta.asp. [Accessed: 2024].
- [64] W3Schools, "HTML <style> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_style.asp. [Accessed: 2024].
- [65] W3Schools, "HTML <body> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_body.asp. [Accessed: 2024].
- [66] W3Schools, "HTML <div> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_div.asp. [Accessed: 2024].
- [67] W3Schools, "HTML <h1> to <h6> Tags," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_hn.asp. [Accessed: 2024].
- [68] W3Schools, "HTML <button> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_button.asp. [Accessed: 2024].
- [69] W3Schools, "HTML <script> Tag," W3Schools. [Online]. Available: https://www.w3schools.com/TAGS/tag_script.asp. [Accessed: 2024].
- [70] W3Schools, "HTML Element Reference," W3Schools. [Online]. Available: <https://www.w3schools.com/TAGS/default.asp>. [Accessed: 2024].
- [71] W3Schools, "What is CSS?," W3Schools. [Online]. Available: https://www.w3schools.com/whatis/whatis_css.asp. [Accessed: 2024].
- [72] W3Schools, "CSS font-family Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_font_font-family.php. [Accessed: 2024].
- [73] W3Schools, "CSS background Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/css3_pr_background.php. [Accessed: 2024].
- [74] W3Schools, "CSS color Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_text_color.php. [Accessed: 2024].
- [75] W3Schools, "CSS text-align Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_text_text-align.php. [Accessed: 2024].
- [76] W3Schools, "CSS margin Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_margin.php. [Accessed: 2024].
- [77] W3Schools, "CSS font-variant Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_font_font-variant.php. [Accessed: 2024].
- [78] W3Schools, "CSS font-size Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_font_font-size.php. [Accessed: 2024].

- [79] W3Schools, "CSS display Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_class_display.php. [Accessed: 2024].
- [80] W3Schools, "CSS padding Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_padding.php. [Accessed: 2024].
- [81] W3Schools, "CSS border Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_border.php. [Accessed: 2024].
- [82] W3Schools, "CSS border-radius Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/css3_pr_border-radius.php. [Accessed: 2024].
- [83] W3Schools, "CSS box-shadow Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/css3_pr_box-shadow.php. [Accessed: 2024].
- [84] W3Schools, "CSS justify-content Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/css3_pr_justify-content.php. [Accessed: 2024].
- [85] W3Schools, "CSS margin-bottom Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_margin-bottom.php. [Accessed: 2024].
- [86] W3Schools, "CSS align-items Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/css3_pr_align-items.php. [Accessed: 2024].
- [87] W3Schools, "CSS cursor Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_class_cursor.php. [Accessed: 2024].
- [88] W3Schools, "CSS height Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_dim_height.php. [Accessed: 2024].
- [89] W3Schools, "SVG Fill Attributes," W3Schools. [Online]. Available: https://www.w3schools.com/graphics/svg_fill.asp. [Accessed: 2024].
- [90] W3Schools, "CSS margin-top Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_margin-top.php. [Accessed: 2024].
- [91] W3Schools, "CSS font-weight Property," W3Schools. [Online]. Available: https://www.w3schools.com/cssref/pr_font_weight.php. [Accessed: 2024].
- [92] W3Schools, "CSS Reference," W3Schools. [Online]. Available: <https://www.w3schools.com/cssref/index.php>. [Accessed: 2024].
- [93] W3Schools, "JavaScript Fetch API," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/api_fetch.asp. [Accessed: 2024].
- [94] W3Schools, "JavaScript Promise then()," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/jsref_promise_then.asp. [Accessed: 2024].
- [95] MDN Web Docs, "Response: json() method," MDN Web Docs. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/Response/json>. [Accessed: 2024].
- [96] W3Schools, "JavaScript String replace()," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/jsref_replace.asp. [Accessed: 2024].
- [97] W3Schools, "JavaScript String charAt()," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/jsref_charat.asp. [Accessed: 2024].
- [98] W3Schools, "JavaScript String toUpperCase()," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/jsref_toupper.asp. [Accessed: 2024].

- [99] W3Schools, "JavaScript String slice()," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/jsref_slice_string.asp. [Accessed: 2024].
- [100] W3Schools, "HTML DOM Document getElementById()," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/met_document_getelementbyid.asp. [Accessed: 2024].
- [101] W3Schools, "HTML DOM Element textContent," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/prop_node_textcontent.asp. [Accessed: 2024].
- [102] W3Schools, "Input Text value Property," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/prop_text_value.asp. [Accessed: 2024].
- [103] W3Schools, "JavaScript Promise catch()," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/jsref_promise_catch.asp. [Accessed: 2024].
- [104] W3Schools, "JavaScript Error message," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/prop_error_message.asp. [Accessed: 2024].
- [105] W3Schools, "Style color Property," W3Schools. [Online]. Available: https://www.w3schools.com/jsref/prop_style_color.asp. [Accessed: 2024].
- [106] W3Schools, "JavaScript and HTML DOM," W3Schools. [Online]. Available: <https://www.w3schools.com/jsref/>. [Accessed: 2024].
- [107] Panasonic Industry, "Basic Knowledge of LC Filters, " Panasonic Industry, Jul. 2018. [Online]. Available: <https://industrial.panasonic.com/ww/ss/technical/b4>.

ΠΑΡΑΡΤΗΜΑ Α : PAM8320RDR Datasheet



A PRODUCT LINE OF
DIODES INCORPORATED



PAM8320

20W Mono Class D Audio Amplifier

New Product

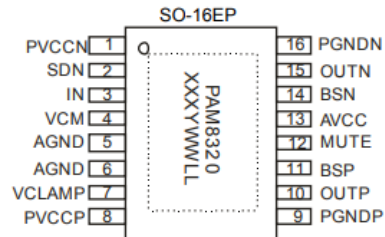
Description

The PAM8320 is an efficient 20W mono Class-D audio power amplifier, designed to drive speakers as low as 4Ω in a bridge-tied-load configuration. Due to the low power dissipation and high efficiency of up to 95%, the device can be used without any external heat sink whilst playing music.

The PAM8320 features short circuit protection, thermal shutdown, over voltage protection and under voltage lock-out.

The PAM8320 is available in a SO-16EP package.

Pin Assignments



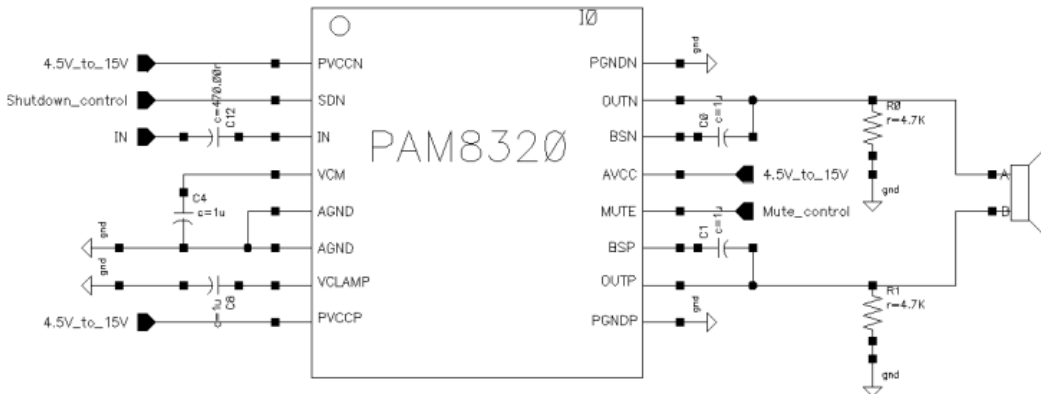
Features

- Operates from 4.5V to 15V
- 20W into 4Ω BTL Load from 12V Supply
- Single-Ended Analog Input
- No Pop Noise for Start-up and Shut-down Sequences
- Internal Oscillator (No External Components Required)
- High Efficient Class-D Operation Eliminates Need for Heat Sinks
- Thermal and Short-Circuit Protection with Auto Recovery
- Over Voltage Protection and Under Voltage Lock-out
- Space-Saving Surface-Mount SO-16EP Package
- Pb-Free Package

Applications

- PC Speaker
- Blue Tooth Speaker
- Home Sound Systems
- Active Speakers
- Docking stations

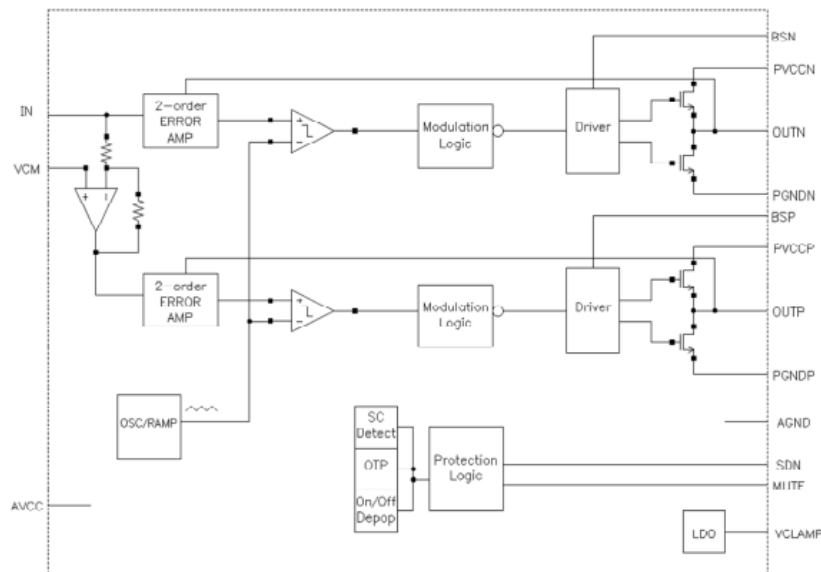
Typical Applications Circuit



Pin Descriptions

Pin	Name	I/O/P	Description
1	PVCCN	P	Power supply for negative H-bridge, not connected to PVCCP or AVCC
2	SDN	I	Shutdown signal for IC (low=shutdown, high=operational). TTL logic levels with compliance to AVCC
3	IN	I	Audio input
4	VCM	O	Reference for analog cells
5,6	AGND	P	Analog ground for digital/analog cells in core
7	VCLAMP	P	Internally generated voltage supply for bootstrap. Not to be used as a supply or connected to any component other than the decoupling capacitor.
8	PVCCP	P	Power supply for positive H-bridge, not connected to PVCCN or AVCC
9	PGNDP	P	Power ground for positive H-bridge
10	OUTP	O	Positive BTL output
11	BSP	I/O	Bootstrap terminal for high-side drive of positive BTL output
12	MUTE	I	A logic high on this pin disables the outputs. A low on this pin enables the outputs. TTL logic levels with compliance to AVCC
13	AVCC	P	High-voltage analog power supply
14	BSN	I/O	Bootstrap terminal for high-side drive of negative BTL output
15	OUTN	O	Negative BTL output
16	PGNDN	P	Power ground for negative H-bridge

Functional Block Diagram



Absolute Maximum Ratings (@T_A = +25°C, unless otherwise specified.)

Parameter	Rating	Unit
Supply Voltage (V _{CC})	18	V
Logic Input Voltage (SDN, MUTE)	-0.3 to V _{CC} +0.3	V
Analog Input Voltage (IN)	-0.3 to 5.5	V
Storage Temperature	-65 to +150	°C
Maximum Junction Temperature	+150	°C
Junction to ambient thermal resistance	40	°C/W

Recommended Operating Conditions (@T_A = +25°C, unless otherwise specified.)

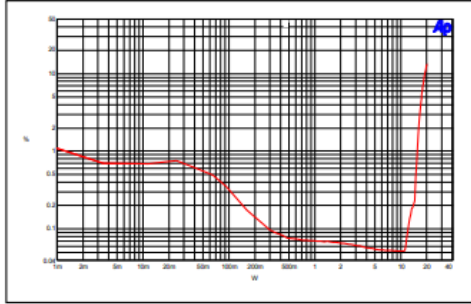
Symbol	Parameter	Min	Max	Unit
V _{CC}	Supply Voltage	4.5	15	V
T _A	Operating Ambient Temperature Range	-40	+85	°C
T _J	Junction Temperature Range	-40	+125	°C

Electrical Characteristics (@T_A = +25°C, V_{CC} = 12V, Gain = 20dB, R_L = L(33μH) + R + L(33μH), unless otherwise noted.)

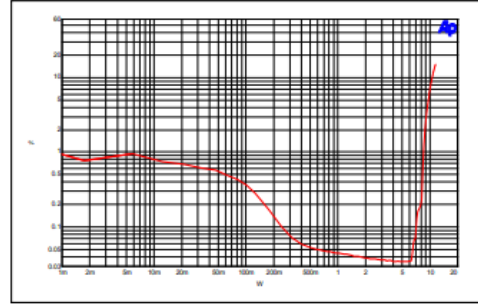
Symbol	PARAMETER	Test Conditions	MIN	TYP	MAX	Units
VOS	Class-D output offset voltage(measured differently)	V _I =0V, A _v =20dB	—	20	100	mV
I _{CC(Q)}	Quiescent supply current	SDN=3.0V, MUTE=0V, No Load	—	15	30	mA
I _{CC(MUTE)}	Quiescent supply current in mute mode	MUTE=2.0V, No load	—	8	20	mA
I _{CC(SDN)}	Quiescent current in shutdown mode	SDN=0.5V, No load	—	20	40	uA
R _{ds(on)}	Drain-source on-state resistance	I _O =0.5A	—	150	—	mΩ
P _{SRR}	Power Supply Rejection Ratio	V _{ripple} =200mVpp, f=1kHz, gain=20dB	—	-60	—	dB
P _O	Output Power at 1% THD+N	f=1kHz	—	15	—	W
	Output Power at 10% THD+N	f=1kHz	—	20	—	
THD+N	Total harmonic distortion + noise	f=1kHz, P _O =7W	—	0.05	—	%
V _n	Output integrated noise floor	20Hz to 22kHz, A-weighted, Gain=20dB	—	300	—	uV
SNR	Signal-to-noise ratio	Max output at THD+N<1%, f=1kHz, Gain=20dB	—	95	—	dB
OTP	Thermal trip point	—	—	+160	—	°C
OTH	Thermal hysteresis	—	—	+40	—	°C
f _{osc}	Oscillator frequency	—	—	300	—	kHz
V _{IH_SDN}	SDN Input High	—	3	—	—	—
V _{IL_SDN}	SDN Input Low	—	—	—	0.5	—
V _{IH_MUTE}	MUTE Input High	—	2	—	—	—
V _{IL_MUTE}	MUTE Input Low	—	—	—	0.5	—

Performance Characteristics (@T_A = +25°C, V_{DD} = 12V, Gain = 20dB, R_L = L(33μH) + R + L(33μH), unless otherwise noted.)

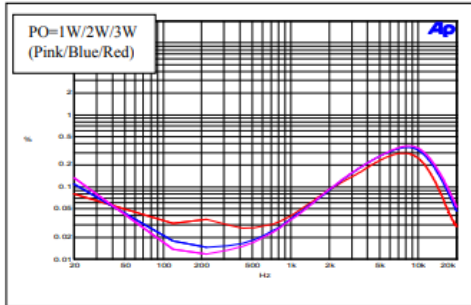
THD+N Vs. Output Power (RL=4Ω)



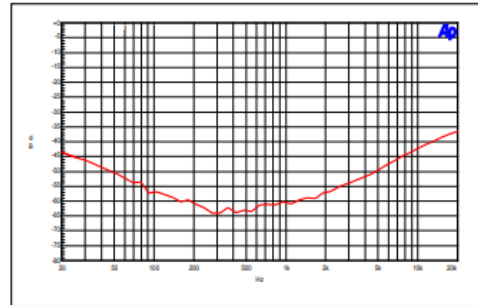
THD+N Vs. Output Power (RL=8Ω)



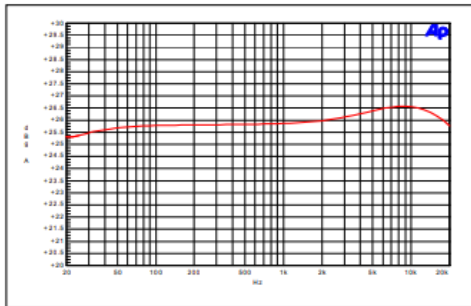
THD+N Vs. Frequency



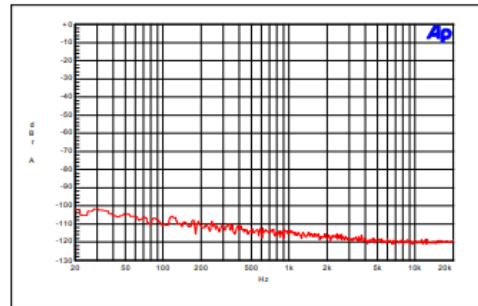
PSRR Vs. Frequency



Frequency Response

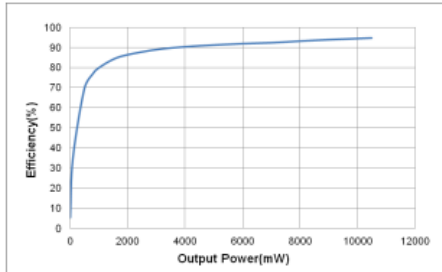


Noise Floor

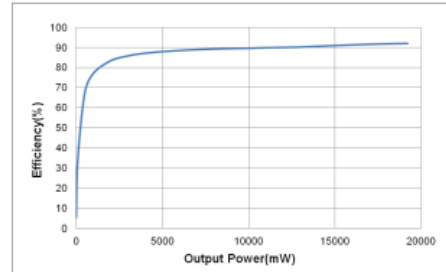


Performance Characteristics (@ $T_A=25^{\circ}\text{C}$, $V_{DD}=12\text{V}$, Gain=20dB, $R_L=L(33\mu\text{H})+R+L(33\mu\text{H})$, unless otherwise noted.)

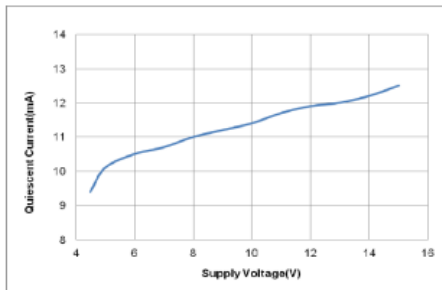
Efficiency Vs. Output Power (RL=8Ω)



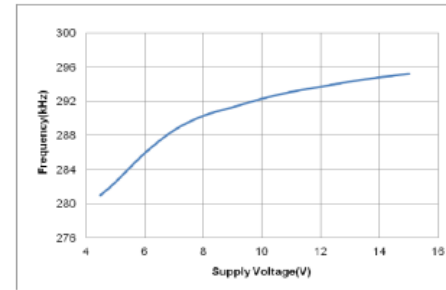
Efficiency Vs. Output Power (RL=4Ω)



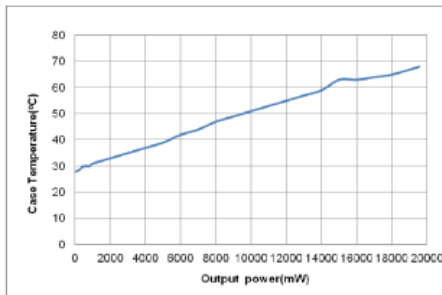
Quiescent Current Vs. Supply Voltage



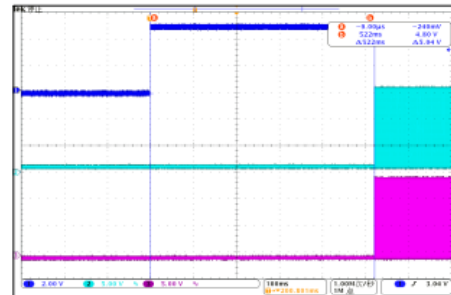
OSC Frequency Vs. Supply Voltage



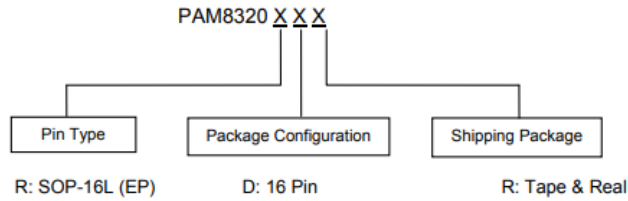
Case Temperature Vs. Output Power (RL=4Ω)



Turn-on Response

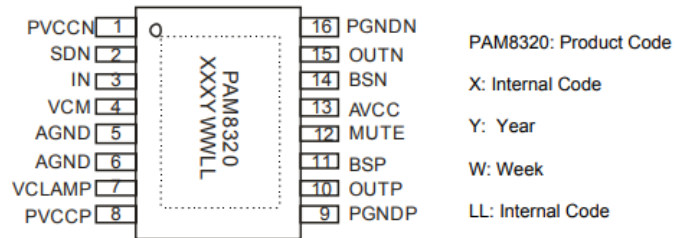


Ordering Information



Part Number	Package	Standard Package
PAM8320RDR	SO-16EP	2,500Units/Tape&Real

Marking Information



ΠΑΡΑΡΤΗΜΑ Β : TDA7718N Datasheet



TDA7718N

3 band car audio processor

Features

- Input multiplexer
 - QD1: quasi-differential stereo inputs
 - SE1: stereo single-ended input
 - SE2: stereo single-ended input
 - SE3: stereo single-ended input
 - FD1 / SE4+SE5: 1 full-differential input or 2 stereo single-ended inputs
- Loudness
 - 2nd order frequency response
 - Programmable center frequency (400 Hz / 800 Hz / 2400 Hz)
 - 15 dB with 1 dB steps
 - Selectable high frequency boost
 - Selectable flat-mode (constant attenuation)
- Volume
 - +23 dB to -31 dB with 1 dB step resolution
 - Soft-step control with programmable blend times
- Bass
 - 2nd order frequency response
 - Center frequency programmable in 4 steps (60 Hz / 80 Hz / 100 Hz / 200 Hz)
 - Q programmable 1.0/1.25/1.5/2.0
 - DC gain programmable
 - -15 dB to 15 dB range with 1 dB resolution
- Middle
 - 2nd order frequency response
 - Center frequency programmable in 4 steps (500 Hz / 1 kHz / 1.5 kHz / 2.5 kHz)
 - Q programmable 0.75/1.0/1.25
 - -15 dB to 15 dB range with 1 dB resolution
- Treble
 - 2nd order frequency response (10 kHz / 12.5 kHz / 15 kHz / 17.5 kHz)
 - Center frequency programmable in 4 steps (10 kHz / 12.5 kHz / 15 kHz / 17.5 kHz)
 - -15 dB to 15 dB with 1 dB resolution



- Speaker
 - 4 independent soft step speaker controls
 - +15 dB to -79 dB with 1 dB steps
 - Direct mute
- Subwoofer
 - 2nd order low pass filter with programmable cut off frequency (55 Hz / 85 Hz / 120 Hz / 160 Hz)
 - 2 independent soft step level control, +15 dB to -79 dB with 1 dB steps
- Mute functions
 - Direct mute
 - Digitally controlled SoftMute with 4 programmable mute-times (0.48 ms/0.96 ms/8 ms/16 ms)
- Offset detection
 - Offset voltage detection circuit for on-board power amplifier failure diagnosis

Description

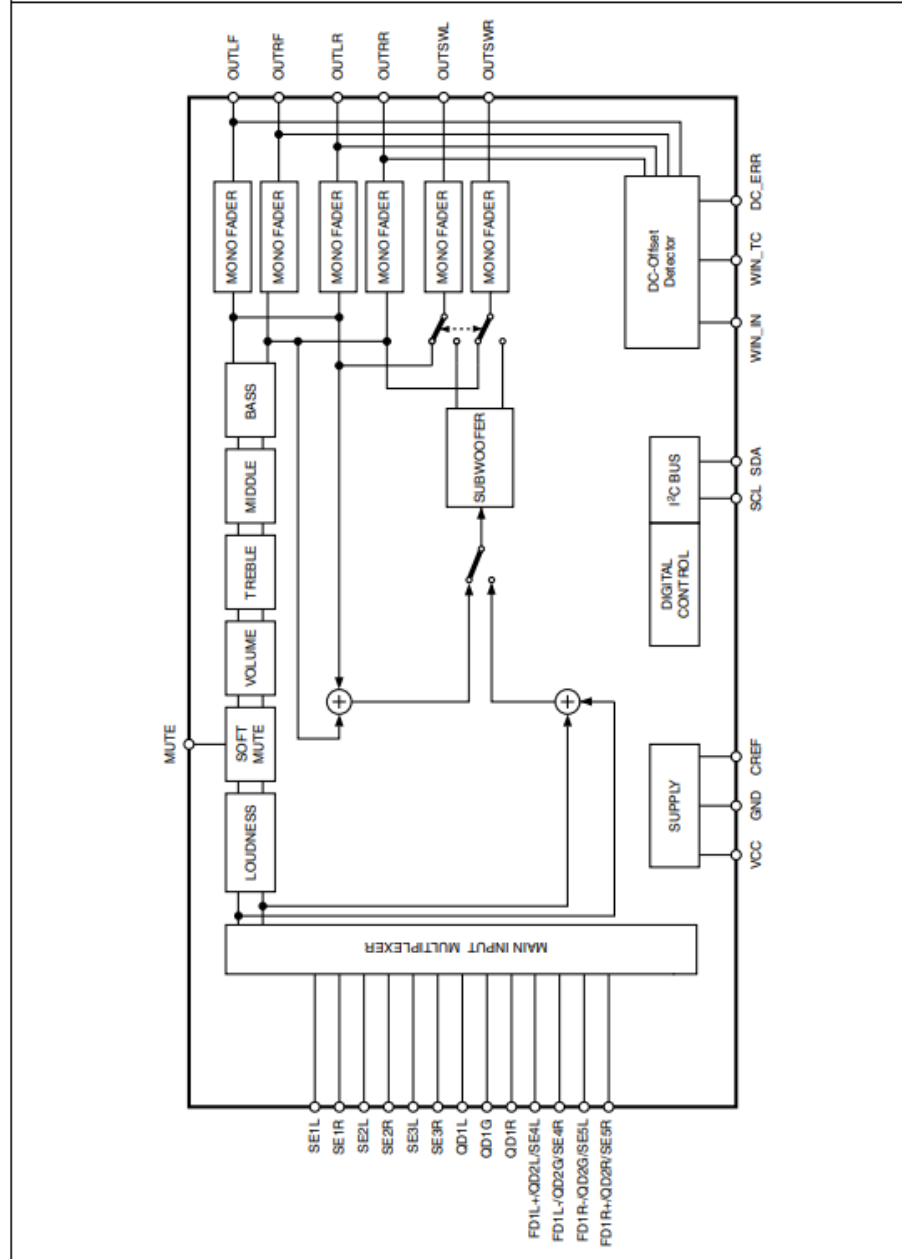
The TDA7718N is a high performance signal processor specifically designed for car radio applications. The device includes a high performance audioprocessor with fully integrated audio filters and new Soft Step architecture. The digital control allows programming in a wide range of filter characteristics.

Table 1. Device summary

Order code	Package	Packing
TDA7718N	TSSOP28	Tube
TDA7718NTR	TSSOP28	Tape and reel

1 Block circuit diagram

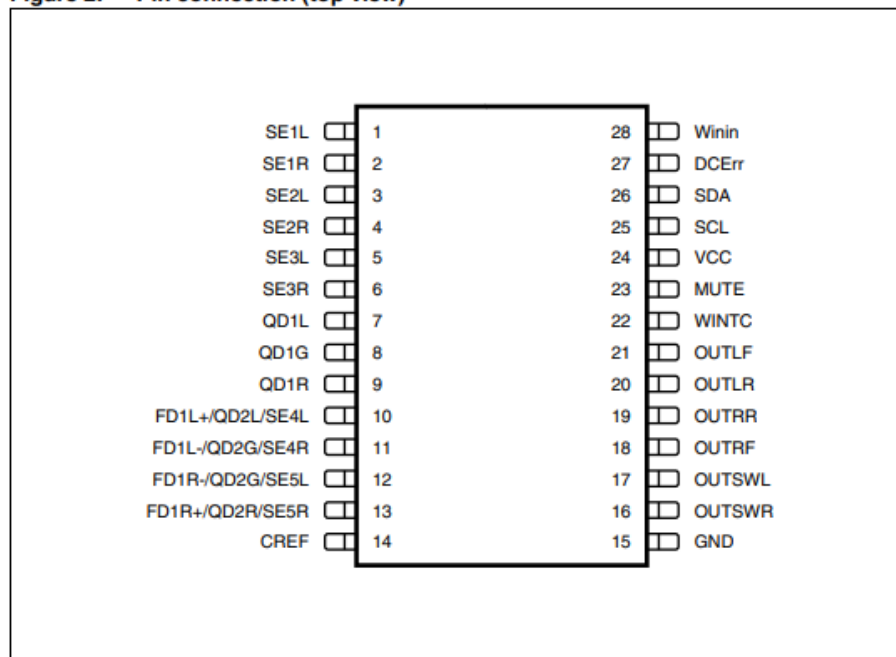
Figure 1. Block circuit diagram



2 Pin connection and pin description

2.1 Pin connection

Figure 2. Pin connection (top view)



2.2 Pin description

Table 2. Pin description

No.	Pin name	Description	I/O
1	SE1L	Single-end input left	I
2	SE1R	Single-end input right	I
3	SE2L	Single-end input left	I
4	SE2R	Single-end input right	I
5	SE3L	Single-end input left	I
6	SE3R	Single-end input right	I
7	QD1L	quasi-differential stereo inputs left	I
8	QD1G	quasi-differential stereo inputs common	I
9	QD1R	quasi-differential stereo inputs right	I
10	FD1L+/QD2L/SE4L	Full differential + input left or quasi-differential left or single-end input left	I

Table 2. Pin description (continued)

No.	Pin name	Description	I/O
11	FD1L-/QD2G/SE4R	Full differential - input left or quasi-differential ground or single-end input right	I
12	FD1R-/QD2G/SE5L	Full differential - input right or quasi-differential ground or single-end input left	I
13	FD1R+/QD2R/SE5R	Full differential + input right or quasi-differential right or single-end input right	I
14	CREF	Reference capacitor	O
15	GND	Ground	S
16	OUTSWR	Subwoofer right output	O
17	OUTSWL	Subwoofer left output	O
18	OUTRF	Front right output	O
19	OUTRR	Rear right output	O
20	OUTLR	Rear left output	O
21	OUTLF	Front left output	O
22	WinTC	DC offset detector filter output	O
23	MUTE	External mute pin	I
24	VCC	Supply	S
25	SCL	I ² C bus clock	I
26	SDA	I ² C bus data	I/O
27	DC_ERR	DC offset detector output	O
28	WIN_IN	DC offset detector input	I

3 Electrical specifications

3.1 Thermal data

Table 3. Thermal data

Symbol	Description	Value	Unit
$R_{th-j\ amb}$	Thermal resistance junction-to-ambient	114	$^{\circ}C/W$

3.2 Absolute maximum ratings

Table 4. Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_S	Operating supply voltage	10.5	V
V_{in_max}	Maximum voltage for signal input pins	7	V
T_{amb}	Operating ambient temperature	-40 to 85	$^{\circ}C$
T_{stg}	Storage temperature range	-55 to 150	$^{\circ}C$

3.3 Electrical characteristics

$V_S = 8.5\text{ V}$; $T_{amb} = 25\text{ }^{\circ}C$; $R_L = 10\text{ k}\Omega$; all gains = 0 dB; $f = 1\text{ kHz}$; unless otherwise specified

Table 5. Electrical characteristics

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
Supply						
V_S	Supply voltage	-	7.5	8.5	10	V
I_S	Supply current	-	23	29	35	mA
Input selector						
R_{in}	Input resistance	All single ended inputs	70	100	130	$k\Omega$
V_{CL}	Clipping level	Input gain = 0 dB	2	-	-	V_{RMS}
S_{IN}	Input separation	-	-	95	-	dB
Differential stereo inputs						
R_{in}	Input resistance	Differential	70	100	-	$k\Omega$
CMRR	Common mode rejection ratio for main source	$V_{CM} = 1\text{ V}_{RMS}$ @ 1 kHz	44	60	-	dB
		$V_{CM} = 1\text{ V}_{RMS}$ @ 10 kHz	44	60	-	dB
e_{No}	Output noise @ speaker outputs	20 Hz - 20 kHz, A-weighted; all stages 0 dB	-	12	22	μV

Table 5. Electrical characteristics (continued)

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
Loudness control						
A_{MAX}	Max attenuation	-	14	15	16	dB
A_{STEP}	Step resolution	-	0.5	1	1.5	dB
f_{Peak}	Peak frequency	f_{P1}	-	400	-	Hz
		f_{P2}	-	800	-	Hz
		f_{P3}	-	2400	-	Hz
Volume control						
G_{MAX}	Max gain	-	22	23	24	dB
A_{MAX}	Max attenuation	-	-	-31	-30	dB
A_{STEP}	Step resolution	-	0.5	1	1.5	dB
E_A	Attenuation set error	-	-0.75	0	+0.75	dB
E_T	Tracking error	-	-	-	2	dB
V_{DC}	DC steps	Adjacent attenuation steps	-3	0.1	3	mV
		From 0 dB to G_{MIN}	-5	0.5	5	mV
Soft mute						
A_{MUTE}	Mute attenuation	-	80	100	-	dB
T_D	Delay time	T1	0.35	0.48	0.65	ms
		T2	0.7	0.96	1.3	ms
		T3	5.6	7.6	9.6	ms
		T4	12.3	15.3	18.3	ms
$V_{TH\ Low}$	Low threshold for SM pin	-	-	-	1	V
$V_{TH\ High}$	High threshold for SM pin	-	2.5	-	-	V
R_{PU}	Internal pull-up resistor	-	32	45	58	k Ω
V_{PU}	Internal pull-up voltage	-	3	3.3	3.6	V
Bass control						
F_c	Center frequency	f_{C1}	-	60	-	Hz
		f_{C2}	-	80	-	Hz
		f_{C3}	-	100	-	Hz
		f_{C4}	-	200	-	Hz
Q_{BASS}	Quality factor	Q_1	-	1	-	-
		Q_2	-	1.25	-	-
		Q_3	-	1.5	-	-
		Q_4	-	2	-	-
C_{RANGE}	Control range	-	± 14	± 15	± 16	dB
A_{STEP}	Step resolution	-	0.5	1	1.5	dB
DC_{GAIN}	Bass-DC-gain	DC = off	-1	0	+1	dB
		DC = on, gain = ± 15 dB	± 4.3	± 4.7	± 5.1	dB

Table 5. Electrical characteristics (continued)

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
Middle control						
C_{RANGE}	Control range	-	±14	±15	±16	dB
A_{STEP}	Step resolution	-	0.5	1	1.5	dB
f_c	Center frequency	f_{C1}	-	500	-	Hz
		f_{C2}	-	1	-	kHz
		f_{C3}	-	1.5	-	kHz
		f_{C4}	-	2.5	-	kHz
Q_{MIDDLE}	Quality factor	Q_1	-	0.75	-	-
		Q_2	-	1	-	-
		Q_3	-	1.25	-	-
Treble control						
C_{RANGE}	Clipping level	-	±14	±15	±16	dB
A_{STEP}	Step resolution	-	0.5	1	1.5	dB
f_c	Center frequency	f_{C1}	-	10	-	kHz
		f_{C2}	-	12.5	-	kHz
		f_{C3}	-	15	-	kHz
		f_{C4}	-	17.5	-	kHz
Speaker attenuators						
G_{MAX}	Max gain	-	14	15	16	dB
A_{MAX}	Max attenuation	-	-	-79	-74	dB
A_{STEP}	Step resolution	-	0.5	1	1.5	dB
A_{MUTE}	Mute attenuation	-	80	90	-	dB
E_E	Attenuation set error	-	-	-	2	dB
V_{DC}	DC steps	Adjacent attenuation steps	-	0.1	5	mV
Audio outputs						
V_{CL}	Clipping level	$d = 0.3\%$; byte8_D6=1	2	-	-	V_{RMS}
		$d = 1\%$; byte8_D6=0	2.2	-	-	V_{RMS}
R_{OUT}	Output impedance	-	-	30	100	Ω
R_L	Output load resistance	-	2	-	-	$k\Omega$
C_L	Output load capacitor	-	-	-	10	nF
V_{DC}	DC voltage level	-	3.8	4.0	4.2	V
Subwoofer lowpass						
f_{LP}	Lowpass corner frequency	f_{LP1}	-	55	-	Hz
		f_{LP2}	-	85	-	Hz
		f_{LP3}	-	120	-	Hz
		f_{LP4}	-	160	-	Hz

Table 5. Electrical characteristics (continued)

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
DC offset detection circuit						
V_{th}	Zero comp. window size	V1	±10	±25	±40	mV
		V2	±30	±50	±70	mV
		V3	±50	±75	±100	mV
		V4	±70	±100	±130	mV
t_{sp}	Max rejected spike length	-	2	11	30	µs
		-	5	22	50	µs
		-	10	33	70	µs
		-	15	44	90	µs
$I_{CHDCerr}$	DCErr charge current	-	2	5	8	µA
$I_{DISDCerr}$	DCErr discharge current	-	4	5	9	mA
V_{OutH}	DCErr high voltage	-	3	3.3	3.6	V
V_{OutL}	DCErr low voltage	-	-	100	300	mV
General						
e_{NO}	Output noise	BW=20 Hz to 20 kHz A-Weighted, all gain = 0 dB	-	12	22	µV
		BW=20 Hz - 20 kHz A-Weighted, Output muted	-	7	12	µV
S/N	Signal to noise ratio	all gain = 0 dB, A-weighted; $V_o = 2 V_{RMS}$	98	104	-	dB
D	Distortion	$V_{IN} = 1 V_{RMS}$; all stages 0 dB	-	0.01	0.1	%
S_C	Channel separation left/right	-	-	90	-	dB

ΠΑΡΑΡΤΗΜΑ C : AMS1117-3.3 Datasheet

**Advanced
Monolithic
Systems**

AMS1117

1A LOW DROPOUT VOLTAGE REGULATOR

RoHs Compliant

FEATURES

- Three Terminal Adjustable or Fixed Voltages*
1.5V, 1.8V, 2.5V, 2.85V, 3.3V and 5.0V
- Output Current of 1A
- Operates Down to 1V Dropout
- Line Regulation: 0.2% Max.
- Load Regulation: 0.4% Max.
- SOT-223, TO-252 and SO-8 package available

APPLICATIONS

- High Efficiency Linear Regulators
- Post Regulators for Switching Supplies
- 5V to 3.3V Linear Regulator
- Battery Chargers
- Active SCSI Terminators
- Power Management for Notebook
- Battery Powered Instrumentation

GENERAL DESCRIPTION

The AMS1117 series of adjustable and fixed voltage regulators are designed to provide up to 1A output current and to operate down to 1V input-to-output differential. The dropout voltage of the device is guaranteed maximum 1.3V, decreasing at lower load currents.

On-chip trimming adjusts the reference voltage to 1.5%. Current limit is set to minimize the stress under overload conditions on both the regulator and power source circuitry.

The AMS1117 devices are pin compatible with other three-terminal SCSI regulators and are offered in the low profile surface mount SOT-223 package, in the 8L SOIC package and in the TO-252 (DPAK) plastic package.

ORDERING INFORMATION:

PACKAGE TYPE			OPERATING JUNCTION TEMPERATURE RANGE
TO-252	SOT-223	8L SOIC	
AMS1117CD	AMS1117	AMS1117CS	-40 to 125° C
AMS1117CD-1.5	AMS1117-1.5	AMS1117CS-1.5	-40 to 125° C
AMS1117CD-1.8	AMS1117-1.8	AMS1117CS-1.8	-40 to 125° C
AMS1117CD-2.5	AMS1117-2.5	AMS1117CS-2.5	-40 to 125° C
AMS1117CD-2.85	AMS1117-2.85	AMS1117CS-2.85	-40 to 125° C
AMS1117CD-3.3	AMS1117-3.3	AMS1117CS-3.3	-40 to 125° C
AMS1117CD-5.0	AMS1117-5.0	AMS1117CS-5.0	-40 to 125° C

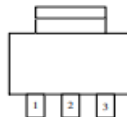
*For additional available fixed voltages contact factory.

PIN CONNECTIONS

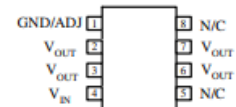
3 PIN FIXED/ADJUSTABLE VERSION

- 1- Ground/Adjust
- 2- V_{OUT}
- 3- V_{IN}

SOT-223 Top View



8L SOIC Top View



TO-252 FRONT VIEW



Advanced Monolithic Systems, Inc. www.advanced-monolithic.com Phone (925) 443-0722 Fax (925) 443-0723

ABSOLUTE MAXIMUM RATINGS (Note 1)

Power Dissipation	Internally limited
Input Voltage	15V
Operating Junction Temperature :	
Control Section	-40°C to 125°C
Power Transistor	-40°C to 125°C
Storage temperature	- 65°C to +150°C

Soldering information	
Lead Temperature (25 sec)	265°C
Thermal Resistance	
SO-8 package	$\phi_{JA} = 160^\circ\text{C/W}$
TO-252 package	$\phi_{JA} = 80^\circ\text{C/W}$
SOT-223 package	$\phi_{JA} = 90^\circ\text{C/W}^*$

* With package soldering to copper area over backside ground plane or internal power plane ϕ_{JA} can vary from 46°C/W to >90°C/W depending on mounting technique and the size of the copper area.

ELECTRICAL CHARACTERISTICS

Electrical Characteristics at $I_{OUT} = 0$ mA, and $T_J = +25^\circ\text{C}$ unless otherwise specified.

Parameter	Device	Conditions	Min	Typ	Max	Units
Reference Voltage (Note 2)	AMS1117	$I_{OUT} = 10$ mA $1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$	1.232	1.250	1.268	V
			1.2125	1.250	1.2875	V
Output Voltage (Note 2)	AMS1117-1.5	$V_{IN} = 3V$	1.478	1.500	1.522	V
			1.455	1.500	1.545	V
	AMS1117-1.8	$V_{IN} = 3.3V$	1.773	1.800	1.827	V
			1.746	1.800	1.854	V
	AMS1117-2.5	$V_{IN} = 4V$	2.463	2.500	2.537	V
			2.425	2.500	2.575	V
	AMS1117-2.85	$V_{IN} = 4.35V$	2.808	2.850	2.892	V
			2.7645	2.850	2.9355	V
	AMS1117-3.3	$V_{IN} = 4.8V$	3.251	3.300	3.349	V
			3.201	3.300	3.399	V
	AMS1117-5.0	$V_{IN} = 6.5V$	4.925	5.000	5.075	V
			4.850	5.000	5.150	V
Line Regulation	AMS1117	$1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$		0.015	0.2	%
				0.035	0.2	%
	AMS1117-1.5	$1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$		0.3	5	mV
				0.6	6	mV
	AMS1117-1.8	$1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$		0.3	5	mV
				0.6	6	mV
	AMS1117-2.5	$1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$		0.3	6	mV
				0.6	6	mV
	AMS1117-2.85	$1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$		0.3	6	mV
				0.6	6	mV
	AMS1117-3.3	$1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$		0.5	10	mV
				1.0	10	mV
	AMS1117-5.0	$1.5V \leq (V_{IN} - V_{OUT}) \leq 12V$		0.5	10	mV
				1.0	10	mV
Load Regulation (Notes 2, 3)	AMS1117	$(V_{IN} - V_{OUT}) = 1.5V, 10mA \leq I_{OUT} \leq 0.8A$		0.1	0.3	%
				0.2	0.4	%
	AMS1117-1.5	$V_{IN} = 3V, 0 \leq I_{OUT} \leq 0.8A$		3	10	mV
				6	20	mV
	AMS1117-1.8	$V_{IN} = 3.3V, 0 \leq I_{OUT} \leq 0.8A$		3	10	mV
				6	20	mV
	AMS1117-2.5	$V_{IN} = 5V, 0 \leq I_{OUT} \leq 0.8A$		3	12	mV
				6	20	mV

ELECTRICAL CHARACTERISTICS

Electrical Characteristics at $I_{OUT} = 0$ mA, and $T_J = +25^\circ\text{C}$ unless otherwise specified.

Parameter	Device	Conditions	Min	Typ	Max	Units
Load Regulation (Notes 2, 3)	AMS1117-2.85	$V_{IN} = 4.35\text{V}$, $0 \leq I_{OUT} \leq 0.8\text{A}$		3 6	12 20	mV mV
	AMS1117-3.3	$V_{IN} = 4.75\text{V}$, $0 \leq I_{OUT} \leq 0.8\text{A}$		3 7	15 25	mV mV
	AMS1117-5.0	$V_{IN} = 6.5\text{V}$, $0 \leq I_{OUT} \leq 0.8\text{A}$		5 10	20 35	mV mV
Dropout Voltage ($V_{IN} - V_{OUT}$)	AMS1117-1.5/-1.8/-2.5/-2.85/-3.3/-5.0	ΔV_{OUT} , $\Delta V_{REF} = 1\%$, $I_{OUT} = 0.8\text{A}$ (Note 4)		1.1	1.3	V
Current Limit	AMS1117-1.5/-1.8/-2.5/-2.85/-3.3/-5.0	$(V_{IN} - V_{OUT}) = 1.5\text{V}$	900	1,100	1,500	mA
Minimum Load Current	AMS1117	$(V_{IN} - V_{OUT}) = 1.5\text{V}$ (Note 5)		5	10	mA
Quiescent Current	AMS1117-1.5/-1.8/-2.5/-2.85/-3.3/-5.0	$(V_{IN} - V_{OUT}) = 1.5\text{V}$		5	11	mA
Ripple Rejection	AMS1117	$f = 120\text{Hz}$, $C_{OUT} = 22\mu\text{F}$ Tantalum, $I_{OUT} = 1\text{A}$, $(V_{IN} - V_{OUT}) = 3\text{V}$, $C_{ADJ} = 10\mu\text{F}$	60	75		dB
	AMS1117-1.5/-1.8/-2.5/-2.85	$f = 120\text{Hz}$, $C_{OUT} = 22\mu\text{F}$ Tantalum, $I_{OUT} = 1\text{A}$, $V_{IN} = 4.35\text{V}$	60	72		dB
	AMS1117-3.3	$f = 120\text{Hz}$, $C_{OUT} = 22\mu\text{F}$ Tantalum, $I_{OUT} = 1\text{A}$ $V_{IN} = 4.75\text{V}$	60	72		dB
	AMS1117-5.0	$f = 120\text{Hz}$, $C_{OUT} = 22\mu\text{F}$ Tantalum, $I_{OUT} = 1\text{A}$ $V_{IN} = 6.5\text{V}$	60	68		dB
Thermal Regulation	AMS1117	$T_A = 25^\circ\text{C}$, 30ms pulse		0.008	0.04	%/W
Adjust Pin Current	AMS1117	$I_{OUT} = 10\text{mA}$, $1.5\text{V} \leq (V_{IN} - V_{OUT}) \leq 12\text{V}$		55	120	μA μA
Adjust Pin Current Change	AMS1117	$I_{OUT} = 10\text{mA}$, $1.5\text{V} \leq (V_{IN} - V_{OUT}) \leq 12\text{V}$		0.2	5	μA
Temperature Stability				0.5		%
Long Term Stability		$T_A = 125^\circ\text{C}$, 1000Hrs		0.3	1	%
RMS Output Noise (% of V_{OUT})		$T_A = 25^\circ\text{C}$, $10\text{Hz} \leq f \leq 10\text{kHz}$		0.003		%
Thermal Resistance Junction-to-Case		All packages			15	$^\circ\text{C}/\text{W}$

Parameters identified with **boldface type** apply over the full operating temperature range.

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. For guaranteed specifications and test conditions, see the Electrical Characteristics. The guaranteed specifications apply only for the test conditions listed.

Note 2: Line and Load regulation are guaranteed up to the maximum power dissipation of 1.2 W for SOT-223, 2.2W for TO-252 and 780mW for 8-Lead SOIC. Power dissipation is determined by the input/output differential and the output current. Guaranteed maximum power dissipation will not be available over the full input/output range.

Note 3: See thermal regulation specifications for changes in output voltage due to heating effects. Line and load regulation are measured at a constant junction temperature by low duty cycle pulse testing. Load regulation is measured at the output lead $\sim 1/8"$ from the package.

Note 4: Dropout voltage is specified up to 0.8A load. For currents over 0.8A dropout will be higher

Note 5: Minimum load current is defined as the minimum output current required to maintain regulation. When $1.5\text{V} \leq (V_{IN} - V_{OUT}) \leq 12\text{V}$ the device is guaranteed to regulate if the output current is greater than 10mA.

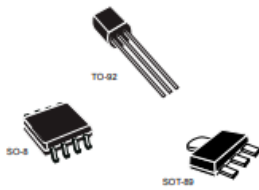
ΠΑΡΑΡΤΗΜΑ Δ : L78L09ACZ Datasheet



L78L

Datasheet

Positive voltage regulators



Features

- Output current up to 100 mA
- Output voltages of 3.3; 5; 6; 8; 9; 10; 12; 15; 18; 24 V thermal overload protection
- Short-circuit protection
- No external components are required
- Available in either $\pm 4\%$ (A) or $\pm 8\%$ (C) selection

Description

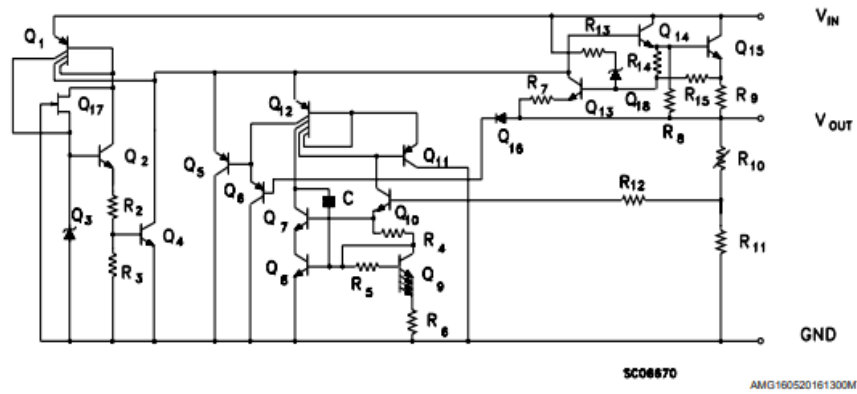
The L78L series of three-terminal positive regulators employ internal current limiting and thermal shutdown, making them essentially indestructible. If adequate heat-sink is provided, they can deliver up to 100 mA output current. They are intended as fixed voltage regulators in a wide range of applications including local or on-card regulation for elimination of noise and distribution problems associated with single-point regulation. In addition, they can be used with power pass elements to make high-current voltage regulators. The L78L series used as Zener diode/resistor combination replacement, offers e improvement along with lower quiescent current and lower noise.

Maturity status link

L78L

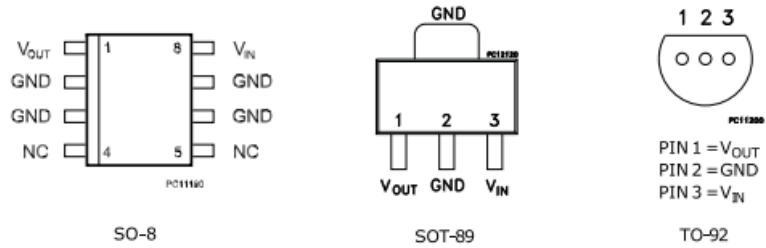
1 Diagram

Figure 1. Schematic diagram



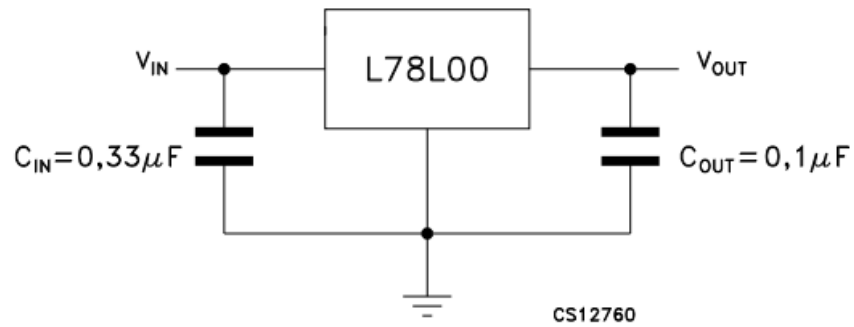
2 Pin configuration

Figure 2. Pin connection (top view, bottom view for TO-92)



AMG160520161301MT

Figure 3. Test circuits



AMG160520161302MT

3 Maximum ratings

Table 1. Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_I	DC Input voltage	$V_O = 3.3$ to 9 V	30
		$V_O = 12$ to 15 V	35
		$V_O = 18$ to 24 V	40
I_O	Output current	100	mA
P_D	Power dissipation	Internally limited ⁽¹⁾	mW
T_{STG}	Storage temperature range	-65 to 150	°C
T_{OP}	Operating junction temperature range	for L78LxxAC / L78LxxC	0 to 125
		for L78LxxAB	-40 to 125

1. Our SO-8 package used for voltage regulators is modified internally to have pins 2, 3, 6 and 7 electrically commuted to the die attach flag. This particular frame decreases the total thermal resistance of the package and increases its ability to dissipate power when an appropriate area of copper on the printed circuit board is available for heat-sinking. The external dimensions are the same as for the standard SO-8.

Table 2. Thermal data

Symbol	Parameter	SO-8	TO-92	SOT-89	Unit
R_{thJC}	Thermal resistance junction-case (max)	20		15	°C/W
R_{thJA}	Thermal resistance junction-ambient (max)	55 ⁽¹⁾	200	55 ⁽¹⁾	°C/W

1. Considering 6 cm² of copper Board heat-sink.

Table 15. Electrical characteristics of L78L08AB and L78L08AC - Refer to the test circuits, $T_J = 0$ to $125\text{ }^\circ\text{C}$ (AC) $T_J = -40$ to $125\text{ }^\circ\text{C}$ (AB), $V_I = 14\text{ V}$, $I_O = 40\text{ mA}$, $C_I = 0.33\text{ }\mu\text{F}$, $C_O = 0.1\text{ }\mu\text{F}$ unless otherwise specified

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage	$T_J = 25\text{ }^\circ\text{C}$	7.68	8	8.32	V
V_O	Output voltage	$I_O = 1$ to 40 mA , $V_I = 10.5$ to 23 V	7.6		8.4	V
		$I_O = 1$ to 70 mA , $V_I = 14\text{ V}$	7.6		8.4	
ΔV_O	Line regulation	$V_I = 10.5$ to 23 V , $T_J = 25\text{ }^\circ\text{C}$			175	mV
		$V_I = 11$ to 23 V , $T_J = 25\text{ }^\circ\text{C}$			125	
ΔV_O	Load regulation	$I_O = 1$ to 100 mA , $T_J = 25\text{ }^\circ\text{C}$			80	mV
		$I_O = 1$ to 40 mA , $T_J = 25\text{ }^\circ\text{C}$			40	
I_d	Quiescent current	$T_J = 25\text{ }^\circ\text{C}$			6	mA
		$T_J = 125\text{ }^\circ\text{C}$			5.5	
ΔI_d	Quiescent current change	$I_O = 1$ to 40 mA			0.1	mA
		$V_I = 11$ to 23 V			1.5	
eN	Output noise voltage	$B = 10\text{ Hz}$ to 100 kHz , $T_J = 25\text{ }^\circ\text{C}$		60		μV
SVR	Supply voltage rejection	$V_I = 12$ to 23 V , $f = 120\text{ Hz}$ $I_O = 40\text{ mA}$, $T_J = 25\text{ }^\circ\text{C}$	37	45		dB
V_d	Dropout voltage			1.7		V

Table 16. Electrical characteristics of L78L09AB and L78L09AC - Refer to the test circuits, $T_J = 0$ to $125\text{ }^\circ\text{C}$ (AC) $T_J = -40$ to $125\text{ }^\circ\text{C}$ (AB), $V_I = 15\text{ V}$, $I_O = 40\text{ mA}$, $C_I = 0.33\text{ }\mu\text{F}$, $C_O = 0.1\text{ }\mu\text{F}$ unless otherwise specified

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage	$T_J = 25\text{ }^\circ\text{C}$	8.64	9	9.36	V
V_O	Output voltage	$I_O = 1$ to 40 mA , $V_I = 11.5$ to 23 V	8.55		9.45	V
		$I_O = 1$ to 70 mA , $V_I = 15\text{ V}$	8.55		9.45	
ΔV_O	Line regulation	$V_I = 11.5$ to 23 V , $T_J = 25\text{ }^\circ\text{C}$			225	mV
		$V_I = 12$ to 23 V , $T_J = 25\text{ }^\circ\text{C}$			150	
ΔV_O	Load regulation	$I_O = 1$ to 100 mA , $T_J = 25\text{ }^\circ\text{C}$			80	mV
		$I_O = 1$ to 40 mA , $T_J = 25\text{ }^\circ\text{C}$			40	
I_d	Quiescent current	$T_J = 25\text{ }^\circ\text{C}$			6	mA
		$T_J = 125\text{ }^\circ\text{C}$			5.5	
ΔI_d	Quiescent current change	$I_O = 1$ to 40 mA			0.1	mA
		$V_I = 12$ to 23 V			1.5	
eN	Output noise voltage	$B = 10\text{ Hz}$ to 100 kHz , $T_J = 25\text{ }^\circ\text{C}$		70		μV
SVR	Supply voltage rejection	$V_I = 12$ to 23 V , $f = 120\text{ Hz}$ $I_O = 40\text{ mA}$, $T_J = 25\text{ }^\circ\text{C}$	37	44		dB
V_d	Dropout voltage			1.7		V

ΠΑΡΑΡΤΗΜΑ Ε : LSK389 Datasheet

LINEAR SYSTEMS

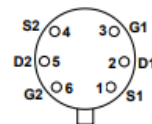
Over 30 Years of Quality Through Innovation

LSK389 A/B/C/D

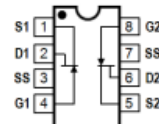
Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

INDUSTRY'S FIRST 100% TESTED LOWEST NOISE JFET

Absolute Maximum Ratings	
@ 25 °C (unless otherwise stated)	
Maximum Temperatures	
Storage Temperature	-65 to +150°C
Junction Operating Temperature	-55 to +150°C
Maximum Power Dissipation	
Continuous Power Dissipation @ +25°C	400mW
Maximum Currents	
Gate Forward Current	$I_{GF} = 10\text{mA}$
Maximum Voltages	
Gate to Source	$V_{GSS} = 40\text{V}$
Gate to Drain	$V_{GDS} = 40\text{V}$



TO-71 6L
Top View



SOIC 8L
Top View

Features

- Ultra-Low Noise: $e_n = 1.3\text{nV}/\sqrt{\text{Hz}}$ (typ), $f = 1.0\text{kHz}$ and $\text{NBW} = 1.0\text{Hz}$
- Ultra-Low Noise: $1.5\text{nV}/\sqrt{\text{Hz}}$ (typ), $f = 10\text{Hz}$ and $\text{NBW} = 1.0\text{Hz}$
- Tight Matching: $|V_{GS1-2}| = 15\text{mV}$ max
- High Breakdown Voltage: $BV_{GSS} = 40\text{V}$ max
- High Gain: $G_{fs} = 20\text{mS}$ (typ)
- Low Capacitance: 25pF (typ)
- Improved Second Source Replacement for 2SK389

Benefits

- Improved System Noise Performance
- Unique Monolithic Dual Design Construction of Interleaving Both JFETs on the Same Piece of Silicon
- Excellent Matching and Thermal Tracking
- Great for Maximizing Battery Operated Applications by Providing a Wide Output Swing
- A High Signal to Noise Ratio as a Result of the LSK389's Low and Tightly Matched Gate Threshold Voltages

Applications

- Audio Amplifiers and Preamps
- Discrete Low-Noise Operational Amplifiers
- Battery-Operated Audio Preamps
- Audio Mixer Consoles
- Acoustic Sensors
- Sonic Imaging
- Instrumentation Amplifiers
- Microphones
- Sonobouys
- Hydrophones
- Chemical and Radiation Detectors

Description

The LSK389 is the industry's lowest noise Dual N-Channel JFET, 100% tested, guaranteed to meet $1/f$ and broadband noise specifications, while eliminating burst (RTN or popcorn) noise entirely. The LSK389 Series, Monolithic Dual N-Channel JFETs were specifically designed to provide users a better performing, less time consuming and cheaper solution for obtaining tighter IDSS matching, and better thermal tracking, than matching individual JFETs. The LSK389 features four grades of IDSS: 2.6-6.5mA, 6.0-12.0mA, 10.0-20.0mA and 17-30mA, with an IDSS match of 10 percent, a gate threshold offset of 15mV, a voltage noise (e_n) of $1.3\text{nV}/\sqrt{\text{Hz}}$ typical at $f = 1.0\text{kHz}$, with a Gain of 20mS typical, and 25pF of capacitance typical. The LSK389 provides a wide output swing, and a high signal

to noise ratio as a result of the LSK389's tightly matched and low gate threshold voltages. The 40V breakdown provides maximum linear headroom in high transient program content amplifiers.

Additionally, the LSK389 provides a low input noise to capacitance product that has nearly zero popcorn noise. The narrow ranges of the IDSS electrical grades combined with the superior matching performance of the LSK389's monolithic dual construction promote ease of device tolerance in low voltage applications, as compared to matching single JFETs. Available in surface mount SOIC 8L and thru-hole TO-71 6L packages.

Contact the factory for tighter noise and other specification selections. For equivalent single N-Channel version, please refer to the LSK170 datasheet.

LSK389 A/B/C/D

Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

Electrical Characteristics @ 25°C (unless otherwise stated)

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS	
BV_{GSS}	Gate to Source Breakdown Voltage	-40	---	---	V	$V_{DS} = 0, I_D = -100\mu A$	
$V_{GS(OFF)}$	Gate to Source Pinch-off Voltage	-0.3	---	-1.6	V	$V_{DS} = 10V, I_D = 0.1\mu A$	
I_{DSS}	Drain to Source Saturation Current	LSK389A	2.6	---	6.5	mA	$V_{DS} = 10V, V_{GS} = 0$
		LSK389B	6	---	12		
		LSK389C	10	---	20		
		LSK389D	17	---	30		
I_{GSS}	Gate to Source Leakage Current	---	-100	-300	pA	$V_{GS} = -25V, V_{DS} = 0$	
I_{G1G2}	Gate to Gate Isolation Current	---	± 1.0	± 50	nA	$V_{G1-G2} = \pm 45V, I_D = I_S = 0A$	
G_s	Full Conduction Transconductance	8	20	---	mS	$V_{DS} = 10V, V_{GS} = 0, f = 1kHz$	
e_n	Noise Voltage	---	1.3	1.9	nV/ \sqrt{Hz}	$V_{DS} = 10V, I_D = 2mA, f = 1kHz, NBW = 1Hz$	
e_n	Noise Voltage	---	1.5	4.0	nV/ \sqrt{Hz}	$V_{DS} = 10V, I_D = 2mA, f = 10Hz, NBW = 1Hz$	
C_{ISS}	Common Source Input Capacitance	---	25	---	pF	$V_{DS} = 10V, V_{GS} = 0, f = 1MHz,$	
C_{RSS}	Common Source Reverse Transfer Cap.	---	5.5	---	pF	$V_{DS} = 10V, I_D = 0, f = 1MHz,$	

Matching Characteristics @ 25°C (unless otherwise stated)

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
$ V_{GS1} - V_{GS2} $	Differential Gate to Source Cutoff Voltage	---	6.0	15	mV	$V_{DS} = 10V, I_D = 1mA$
$\frac{I_{DSS1}}{I_{DSS2}}$	Saturation Drain Current Ratio	0.9	1.0	1.1	n/a	$V_{DS} = 10V, V_{GS} = 0V$

Notes

1. Absolute maximum ratings are limiting values above which serviceability may be impaired.
2. Pulse Test: $PW \leq 300\mu s$, Duty Cycle $\leq 3\%$
3. All characteristics MIN/TYP/MAX numbers are absolute values. Negative values indicate electrical polarity only. Information furnished by Linear Integrated Systems is believed to be accurate and reliable. However, no responsibility is assumed for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Linear Integrated Systems.

LSK389 A/B/C/D

Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

Typical Characteristics

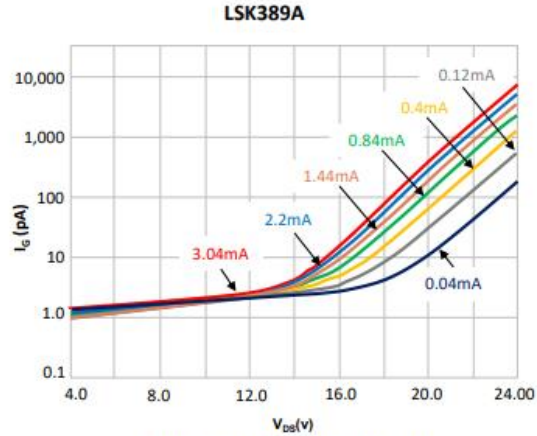


Figure 1. Gate Current (I_G) vs. V_{DS} vs. I_D

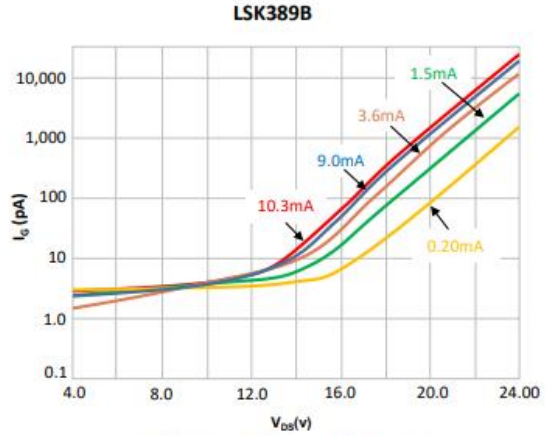


Figure 2. Gate Current (I_G) vs. V_{DS} vs. I_D

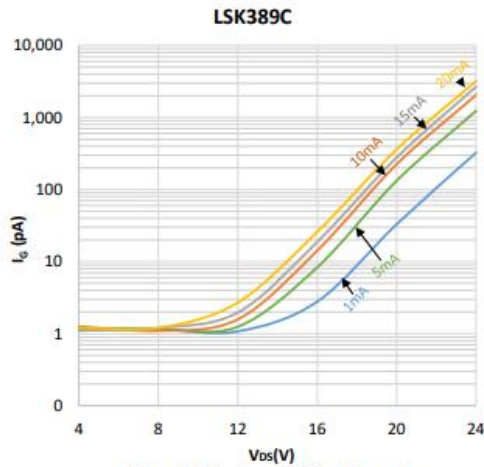


Figure 3. Gate Current (I_G) vs. V_{DS} vs. I_D

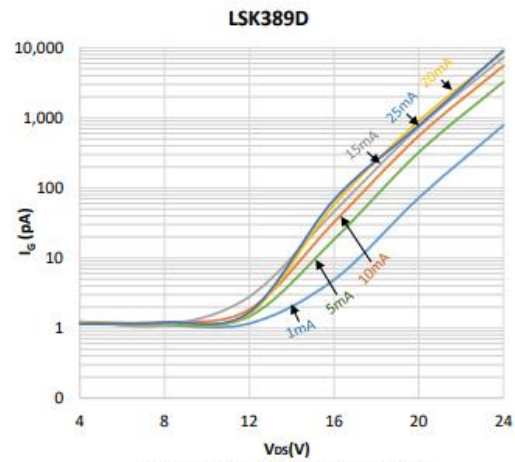


Figure 4. Gate Current (I_G) vs. V_{DS} vs. I_D

LSK389 A/B/C/D

Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

Typical Characteristics

LSK389A

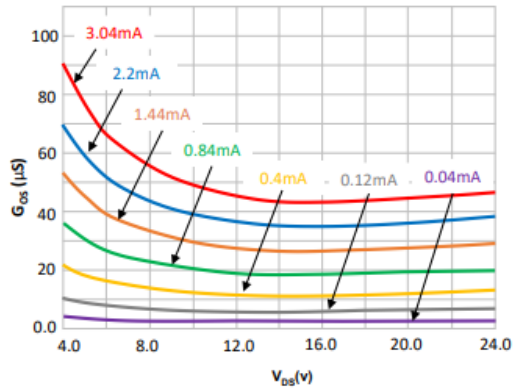


Figure 5. Output Conductance - G_{os} vs. V_{ds} vs. I_D

LSK389B

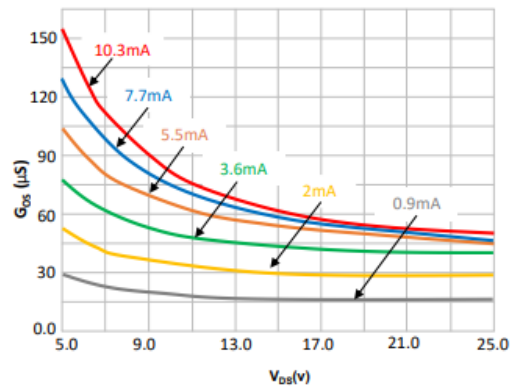


Figure 6. Output Conductance - G_{os} vs. V_{ds} vs. I_D

LSK389C

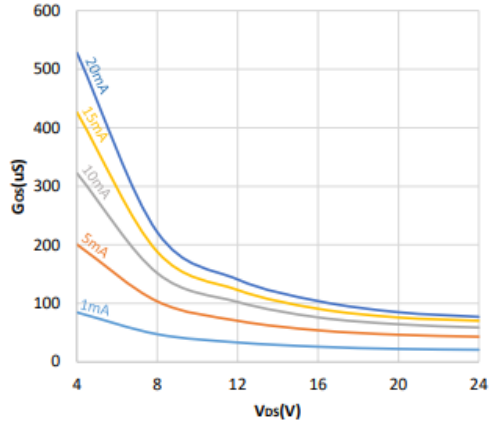


Figure 7. Output Conductance - G_{os} vs. V_{ds} vs. I_D

LSK389D

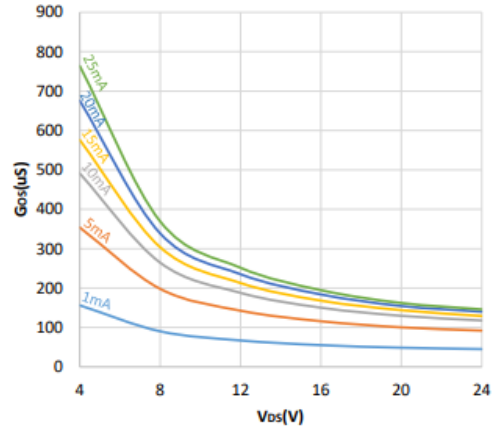


Figure 8. Output Conductance - G_{os} vs. V_{ds} vs. I_D

LSK389 A/B/C/D

Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

Typical Characteristics

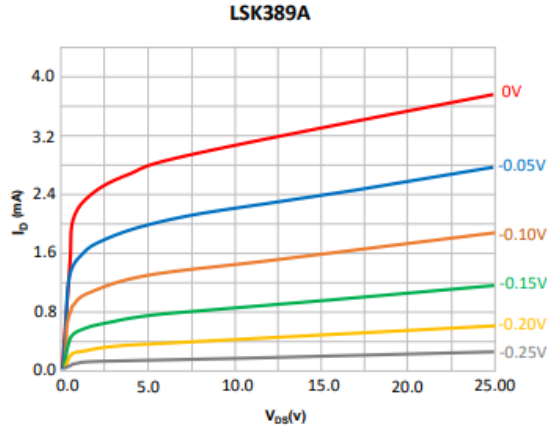


Figure 9. I_D vs. V_{DS} vs. V_{GS}

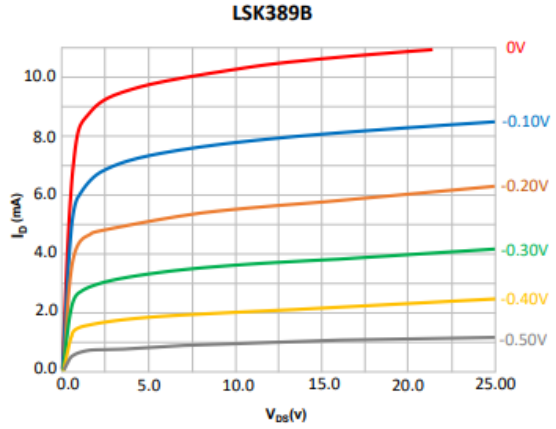


Figure 10. I_D vs. V_{DS} vs. V_{GS}

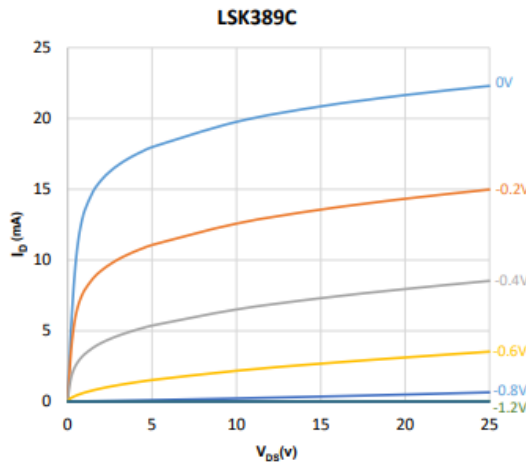


Figure 11. I_D vs. V_{DS} vs. V_{GS}

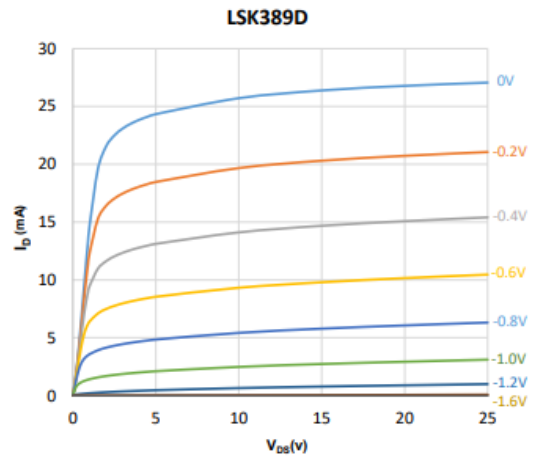


Figure 12. I_D vs. V_{DS} vs. V_{GS}

LSK389 A/B/C/D

Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

Typical Characteristics

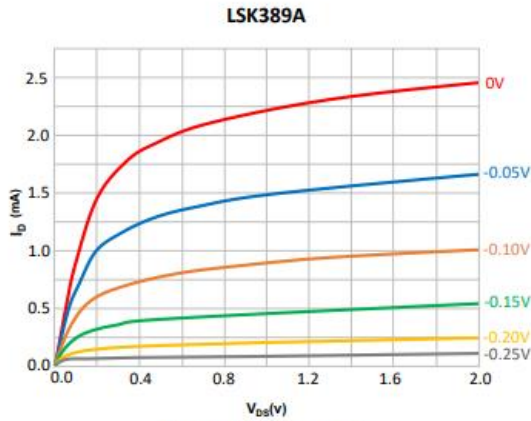


Figure 13. I_D vs. V_{DS} vs. V_{GS}

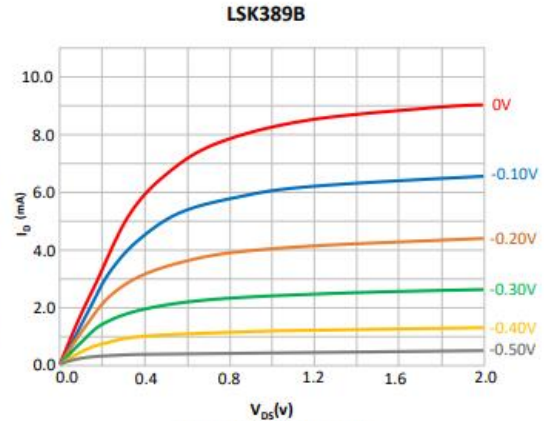


Figure 14. I_D vs. V_{DS} vs. V_{GS}

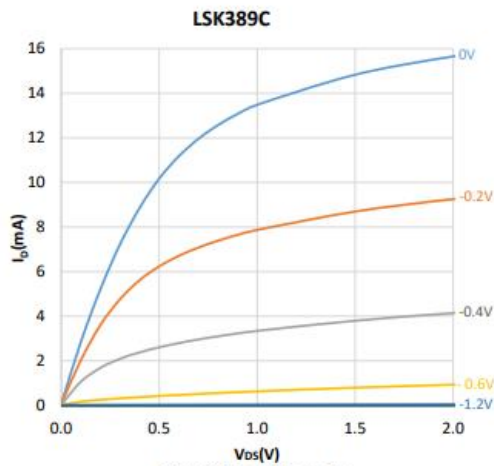


Figure 15. I_D vs. V_{DS} vs. V_{GS}

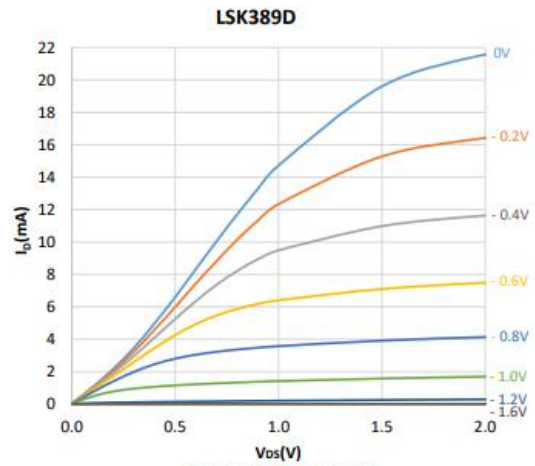


Figure 16. I_D vs. V_{DS} vs. V_{GS}

LSK389 A/B/C/D

Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

Typical Characteristics

Common Source Forward Transconductance vs. Drain Current
LSK389A & B

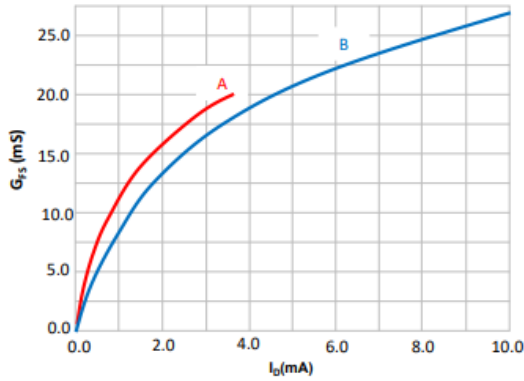


Figure 17. G_{FS} vs. I_D

Common Source Forward Transconductance vs. Drain Current
LSK389C & D

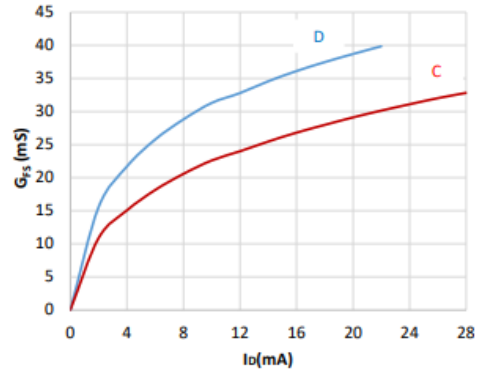


Figure 18. G_{FS} vs. I_D

LSK389A & B

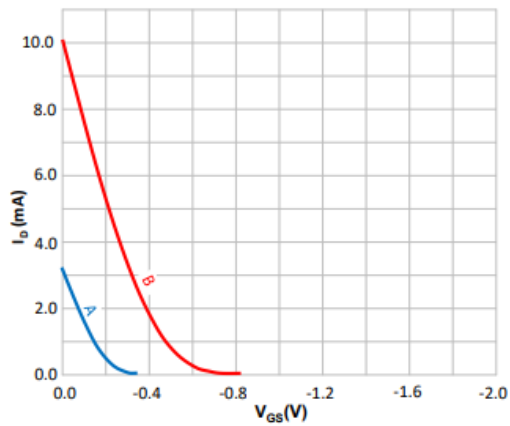


Figure 19. I_D vs. V_{GS}

LSK389C & D

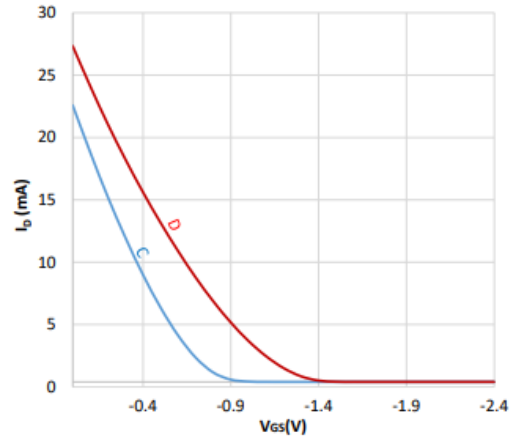


Figure 20. I_D vs. V_{GS}

LSK389 A/B/C/D

Ultra-Low Noise Monolithic Dual N-Channel JFET Amplifier

Typical Characteristics

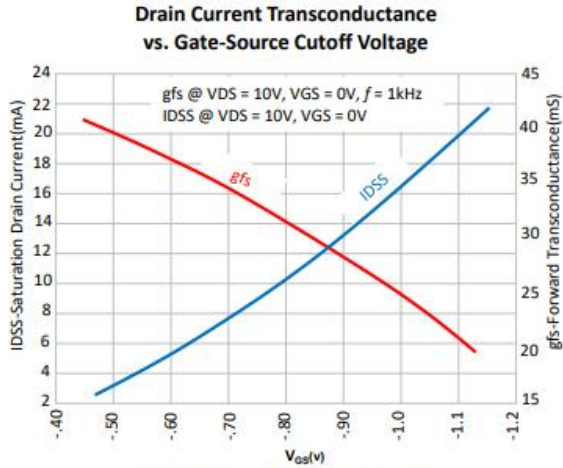


Figure 21. V_{GS} (cutoff) vs. I_{DSS} vs. G_{FS}

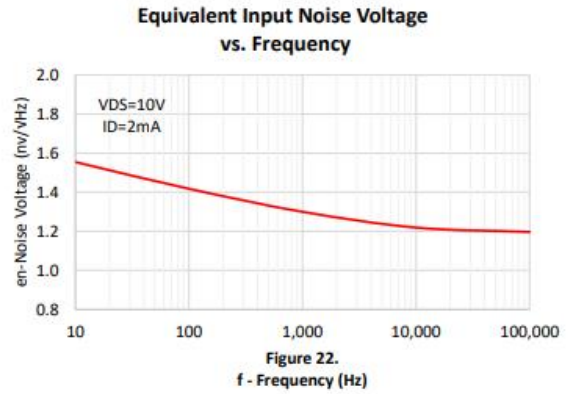
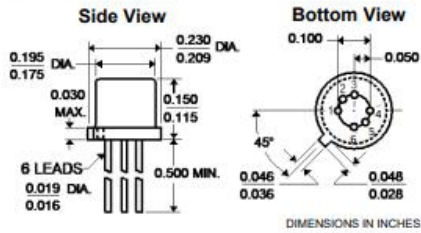


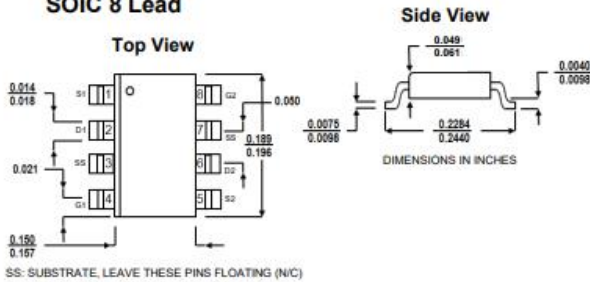
Figure 22. f - Frequency (Hz)

Package Dimensions

TO-71 6 Lead



SOIC 8 Lead



Ordering Information

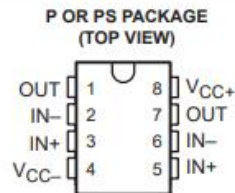
Standard Part Call-Out
LSK389A/B/C or D TO-71 6L RoHS
LSK389A/B/C or D SOIC 8L RoHS
Custom Part Call-Out (Custom Parts Include SEL + 4 Digit Numeric Code)
LSK389A/B/C or D TO-71 6L RoHS SELXXXX
LSK389A/B/C or D SOIC 8L RoHS SELXXXX

ΠΑΡΑΡΤΗΜΑ F : NE5532PS Datasheet

NE5532, NE5532A DUAL LOW-NOISE OPERATIONAL AMPLIFIERS

SLOS075C – NOVEMBER 1979 – REVISED JUNE 2000

- Equivalent Input Noise Voltage
5 nV/√Hz Typ at 1 kHz
- Unity-Gain Bandwidth . . . 10 MHz Typ
- Common-Mode Rejection
Ratio . . . 100 dB Typ
- High dc Voltage Gain . . . 100 V/mV Typ
- Peak-to-Peak Output Voltage Swing
32 V Typ With $V_{CC\pm} = \pm 18$ V and $R_L = 600 \Omega$
- High Slew Rate . . . 9 V/μs Typ
- Wide Supply Voltage Range . . . ±3 V to
±20 V
- Designed to Be Interchangeable With
Signetics NE5532 and NE5532A
- Package Options Include Plastic
Small-Outline (PS) Package and Standard
Plastic (P) DIP



description

The NE5532 and NE5532A are high-performance operational amplifiers combining excellent dc and ac characteristics. They feature very low noise, high output-drive capability, high unity-gain and maximum-output-swing bandwidths, low distortion, high slew rate, input-protection diodes, and output short-circuit protection. These operational amplifiers are compensated internally for unity-gain operation. The NE5532A has specified maximum limits for equivalent input noise voltage.

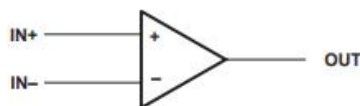
The NE5532 and NE5532A are characterized for operation from 0°C to 70°C.

AVAILABLE OPTIONS

T _A	PACKAGED DEVICES	
	PLASTIC DUAL-IN-LINE (P)	PLASTIC SMALL-OUTLINE (PS)
0°C to 70°C	NE5532P NE5532AP	NE5532PS NE5532APS

The PS package is available taped and reeled. Add the suffix R to the device type (e.g., NE5532PSR).

symbol (each amplifier)



PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2000, Texas Instruments Incorporated

 **TEXAS
INSTRUMENTS**

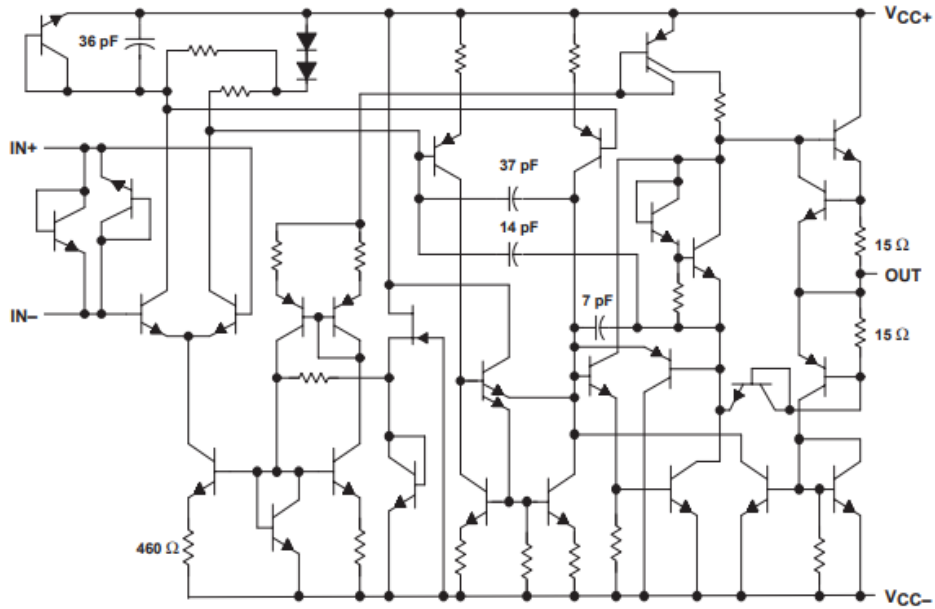
POST OFFICE BOX 865303 • DALLAS, TEXAS 75285

1

NE5532, NE5532A
DUAL LOW-NOISE OPERATIONAL AMPLIFIERS

SLOS075C – NOVEMBER 1979 – REVISED JUNE 2000

schematic (each amplifier)



Component values shown are nominal.

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

Supply voltage, V_{CC+} (see Note 1)	22 V
Supply voltage, V_{CC-} (see Note 1)	-22 V
Input voltage, either input (see Notes 1 and 2)	$V_{CC\pm}$
Input current (see Note 3)	± 10 mA
Duration of output short circuit (see Note 4)	Unlimited
Package thermal impedance, θ_{JA} (see Note 5): P package	85°C/W
PS package	95°C/W
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	-65°C to 150°C

- NOTES:
1. All voltage values, except differential voltages, are with respect to the midpoint between V_{CC+} and V_{CC-} .
 2. The magnitude of the input voltage must never exceed the magnitude of the supply voltage.
 3. Excessive input current will flow if a differential input voltage in excess of approximately 0.6 V is applied between the inputs unless some limiting resistance is used.
 4. The output may be shorted to ground or either power supply. Temperature and/or supply voltages must be limited to ensure the maximum dissipation rating is not exceeded.
 5. The package thermal impedance is calculated in accordance with JESD 51.

NE5532, NE5532A
DUAL LOW-NOISE OPERATIONAL AMPLIFIERS

SLOS075C – NOVEMBER 1979 – REVISED JUNE 2000

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC+}	5		15	V
Supply voltage, V_{CC-}	-5		-15	V
Operating free-air temperature	0		70	°C

electrical characteristics, $V_{CC\pm} = +15\text{ V}$, $T_A = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS†		MIN	TYP	MAX	UNIT
V_{IO} Input offset voltage	$V_O = 0$	$T_A = 25^\circ\text{C}$		0.5	4	mV
		$T_A = 0^\circ\text{C to } 70^\circ\text{C}$			5	
I_{IO} Input offset current	$T_A = 25^\circ\text{C}$			10	150	nA
	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$				200	
I_{IB} Input bias current	$T_A = 25^\circ\text{C}$			200	800	nA
	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$				1000	
V_{ICR} Common-mode input-voltage range			± 12	± 13		V
V_{OPP} Maximum peak-to-peak output-voltage swing	$R_L \geq 600\ \Omega$	$V_{CC\pm} = \pm 15\text{ V}$	24	26		V
		$V_{CC\pm} = \pm 18\text{ V}$	30	32		
A_{VD} Large-signal differential-voltage amplification	$R_L \geq 600\ \Omega$, $V_O = \pm 10\text{ V}$	$T_A = 25^\circ\text{C}$	15	50		V/mV
		$T_A = 0^\circ\text{C to } 70^\circ\text{C}$	10			
	$R_L \geq 2\text{ k}\Omega$, $V_O = \pm 10\text{ V}$	$T_A = 25^\circ\text{C}$	25	100		
		$T_A = 0^\circ\text{C to } 70^\circ\text{C}$	15			
A_{vd} Small-signal differential-voltage amplification	$f = 10\text{ kHz}$			2.2		V/mV
B_{OM} Maximum-output-swing bandwidth	$R_L = 600\ \Omega$	$V_O = \pm 10\text{ V}$		140		kHz
		$V_{CC\pm} = \pm 18\text{ V}$, $V_O = \pm 14\text{ V}$		100		
B_1 Unity-gain bandwidth	$R_L = 600\ \Omega$	$C_L = 100\text{ pF}$		10		MHz
r_i Input resistance			30	300		k Ω
Z_o Output impedance	$A_{VD} = 30\text{ dB}$, $R_L = 600\ \Omega$, $f = 10\text{ kHz}$			0.3		Ω
CMRR Common-mode rejection ratio	$V_{IC} = V_{ICR\text{ min}}$		70	100		dB
k_{SVR} Supply voltage rejection ratio ($\Delta V_{CC\pm}/\Delta V_{IO}$)	$V_{CC\pm} = \pm 9\text{ V to } \pm 15\text{ V}$, $V_O = 0$		80	100		dB
I_{OS} Output short-circuit current			10	38	60	mA
I_{CC} Total supply current	$V_O = 0$, No load			8	16	mA
Crosstalk attenuation (V_{O1}/V_{O2})	$V_{O1} = 10\text{ V}$ peak, $f = 1\text{ kHz}$			110		dB

† All characteristics are measured under open-loop conditions with zero common-mode input voltage unless otherwise specified.

NE5532, NE5532A
 DUAL LOW-NOISE OPERATIONAL AMPLIFIERS

SLOS075C – NOVEMBER 1979 – REVISED JUNE 2000

operating characteristics, $V_{CC\pm} = \pm 15\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	NE5532			NE5532A			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
SR	Slew rate at unity gain	9			9			V/ μ s
	Overshoot factor	10%			10%			
V_n	Equivalent input noise voltage	$V_I = 100\text{ mV}$, $R_L = 600\ \Omega$, $A_{VD} = 1$, $C_L = 100\text{ pF}$						
		$f = 30\text{ Hz}$	8	8	10	nV/ $\sqrt{\text{Hz}}$		
		$f = 1\text{ kHz}$	5	5	6			
I_n	Equivalent input noise current	$f = 30\text{ Hz}$	2.7	2.7			pA/ $\sqrt{\text{Hz}}$	
		$f = 1\text{ kHz}$	0.7	0.7				



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

ΠΑΡΑΡΤΗΜΑ Γ : Nutube Datasheet

DC-Heating
Directly-Heated
Connected in Parallel

KORG

Nutube
6P1
Twin Triode

1. Ratings

Parameter	Symbol	MIN	TYP	MAX	Unit
Operating Temp.	To	-40	-	+85	°C
Storage Temp.	Ts	-50	-	+85	°C
Filament Voltage	Ef #1	0.6	0.7	0.8	V
Grid Voltage	Eg #2	-	-	5.0	V
Anode Voltage	Ea #2	5.0	-	80.0	V
Anode Power Dissipation	Na	-	-	1.7	mW
Insulation Resistance(Anode-Other)	-	3	-	-	MΩ
Insulation Resistance(Grid-Other)	-	3	-	-	MΩ
Color of Illumination	Green				

2. Electrical Characteristics

Parameter	Symbol	Test Condition	MIN	TYP	MAX	Unit	
Filament Current	If	Ef=0.7Vdc #1	Eg=Ea=0V	16.0	17.0	20.0	mAdc
Bias Voltage	Bias		Ia=18.2μA #5#6 Vin=0Vrms #5	1.5	2.5	3.5	Vdc
Voltage Amplification (Vout / Vin)	A		Ia=18.2μA #5#6 Vin=0.245Vrms, 1kHz	3.7	5.0	6.4	-
Resonance Frequency	Fr #7		Ia=18.2μA #5#6 Vin=0Vrms #5	-	5.8	-	kHz
Microphonic Tube Noise	Nm		With designated vibration machine	-	-	4.0	Vpp
Anode Current	Ia #3		Eg=2.0Vdc #2 Ea=12.0Vdc	-	32.0	-	μAdc
Grid Current	Ig #4		Ea=12.0Vdc	-	6.0	-	μAdc
Amplification Factor	μ			-	14.5	-	-
Transconductance	gm			-	54	-	μS
Anode Resistance	Ra			-	330	-	kΩ

#1 Per each filament
#2 See Fig.1
#3 Per each grid
#4 Per each anode
#5 See Fig.2
#6 Ia adjusted by 'Bias'

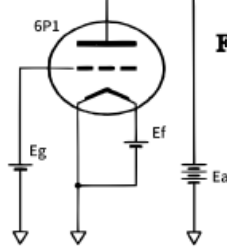


Fig. 1

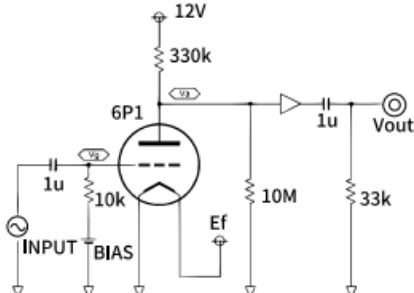


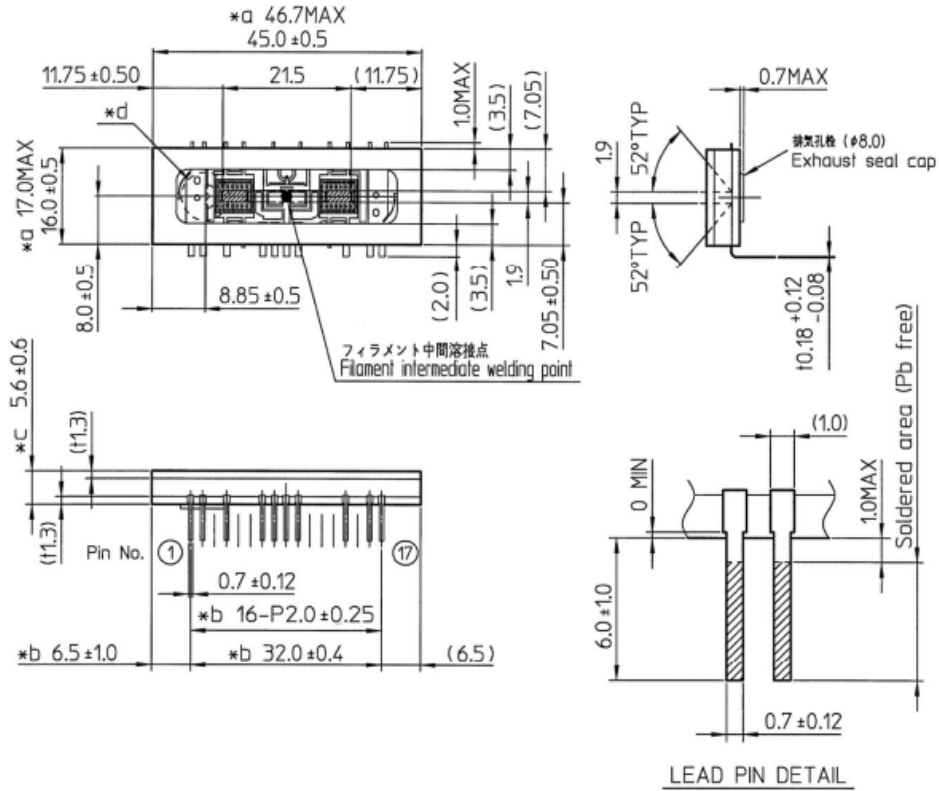
Fig. 2

Capacitances (Typical)

F1-G1	F1-A1	F1-GND	G1-A1	G1-GND	G1-A2	G1-G2	A1-GND	A1-A2
2-4	2-7	2-8	4-7	4-8	4-10	4-14	7-8	7-10
9.1pF	4.3pF	39.2pF	2.5pF	10.9pF	2.3pF	4.3pF	4.5pF	1.4pF

Nutube 6P1 External Dimensions

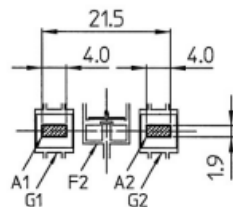
Unit: mm
 (): Reference only
 参考寸法



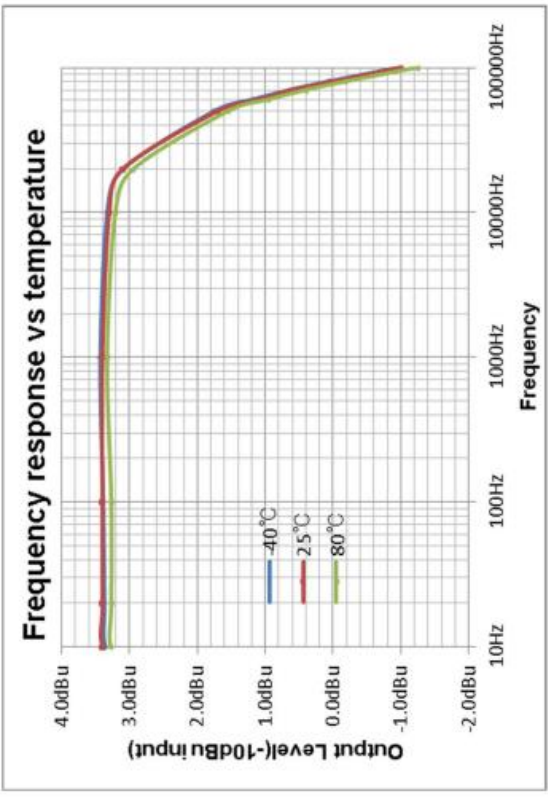
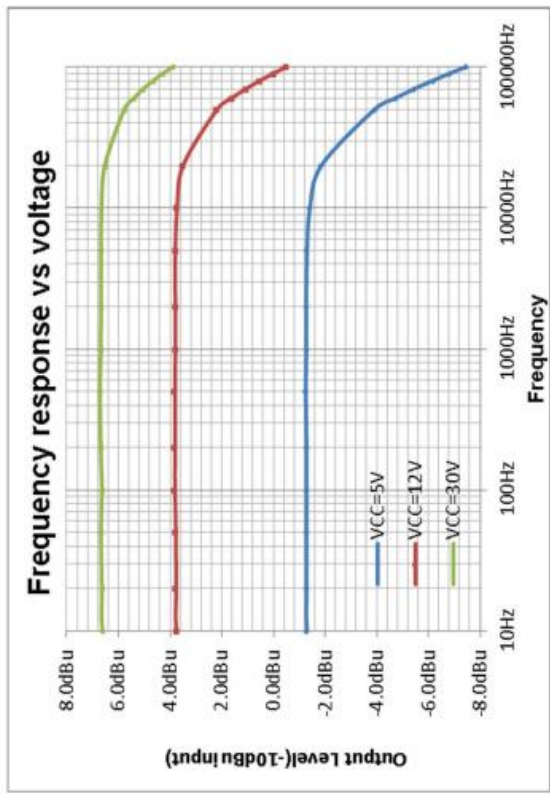
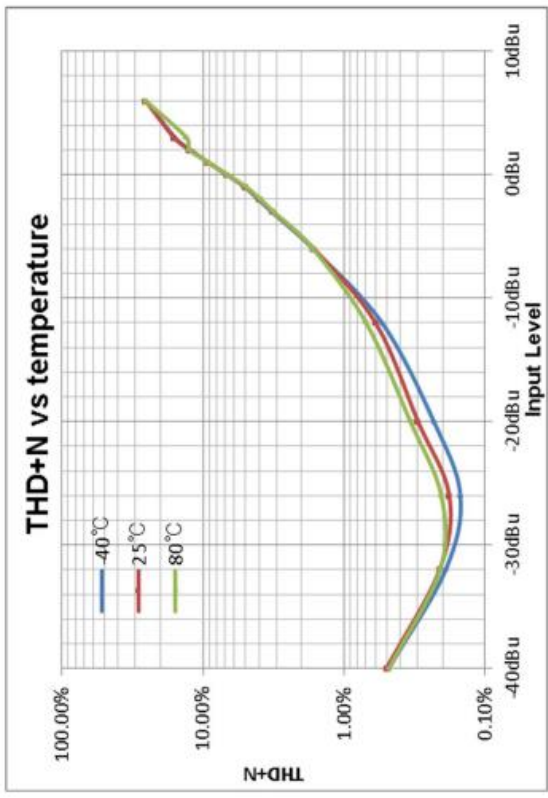
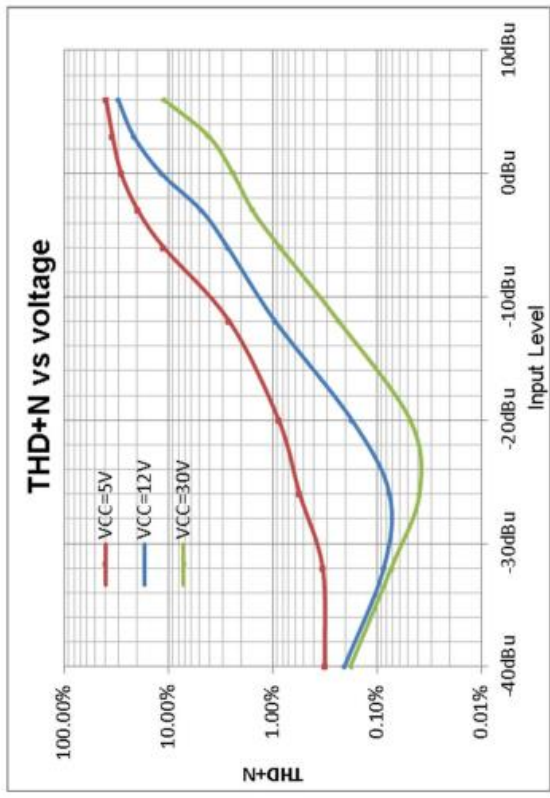
- *a フリットのはみ出しを含む寸法とする。
Including any protruding frit glass.
- *b 基板底面より3mmの位置の寸法とする。
Within 3mm from the bottom of the glass substrate.
- *c 排気孔栓の厚みを含まない。
Does not include the thickness of the exhaust seal cap.
- *d 排気孔栓は排気孔の中心から半径6mmの範囲に収まっていること。
Exhaust seal cap is entirely within a 6mm radius from the center point.

Pin assignment

Pin NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Assignment	F1	F1	NP	G1	NP	NP	A1	GND	F2	A2	NP	NP	NP	G2	NP	F3	F3



- F1: フィラメント 1 (左)/Filament 1 (Left)
- F2: フィラメント 2 (中)/Filament 2 (Center)
- F3: フィラメント 3 (右)/Filament 3 (Right)
- NP: ノーピン/No Pin
- G1: グリッド 1/Grid 1
- G2: グリッド 2/Grid 2
- A1: アノード 1/Anode 1
- A2: アノード 2/Anode 2



ΠΑΡΑΡΤΗΜΑ Η : WEMOS D1 mini Clone ESP8266 Datasheet (Παρόμοιο με αυτό που χρησιμοποιήθηκε στην εργασία)

Product parameters:

Model	ESP-12F	Describe
Packing form	SMD22	Patch
Module Size	24*15.9*3mm	±0.2mm
Certificaton Information	/	FCC/CE/IC/NCC/TELEC/ANATEL/ REACH/ROHS/SRRO
SPI FLASH	32Mbit	Default
Support Interface	/	UART / GPIO / ADC / PWN
Available IO Port	9	
Series Rate	4608000bps	Support 300~4608000bps,default 115200bps
Spectrum Range	2.4GHz	2412~2484MHz
Antenna Form	On board	PCB antenna on board,transmission distance is about 100 meters
Transmitting Power	/	802.11b:16±2dBm;802.11g:16±2d Bm;802.11n:16±2dBm;
Security	/	WEP/WPA-PSK/WPA2-PSK
Scope Of Power Supply	3.3V	3.0~3.6V,current>500MA
Working Temperature	Industrial Grade	-20℃~+85℃
At Support	Built in interlligent processing	It can be read by at command

Features:

- Based on esp8266ex;
- Arduino compatible, programming with Arduino IDE;
- 11x I/O pin;
- 1x ADC pin (input 0-3.3V);
- Support OTA Online;
- On board 5V 1A switch power supply (large input 24V).

Instruction:

Development Board Module (Pin Description Table)		
PIN	Description	IC Internal Pin
D0	I/O, interrupt, PWM, I2C and 1-wire are not supported	GPIO16
D1	I/O, SCL of I2C in default mode	GPIO5
D2	I/O, SDA of I2C in default mode	GPIO4
D3	I/O, Pull up and enter flash mode at low power	GPIO0
D4	I/O, Pull up	GPIO2
D5	I/O, SPI clock	GPIO14
D6	I/O, SPI MISO	GPIO12
D7	I/O, SPI MOSI	GPIO13
D8	I/O, drop down, SPI, default slice selection (SS)	GPIO15
A0	AD input, 0-3.3V	ADC
TX	Serial transmission	GPIO1
RX	Serial reception	GPIO3
RST	Reset	RST
3V3	Power supply 3.3V	3.3V
G	GND	GND
5V	Power supply 5V	External power supply
All IO working levels are 3.3V, and can withstand 5V in an instant All IO supports interrupt, PWM, I2C and 1-wire except D0		

ΠΑΡΑΡΤΗΜΑ Ι : Ο ΚΩΔΙΚΑΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

```
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>

const char* ssid = "...";
const char* password = "...";

IPAddress staticIP(..., ..., ..., ...);
IPAddress gateway(..., ..., ..., ...);
IPAddress subnet(..., ..., ..., ...);

AsyncWebServer server(80);

int volume = 50;
int bass = 0;
int middle = 0;
int treble = 0;
int DSPMute = D5;
int ampMute = D7;
int ampSDN = D6;

uint8_t I2CInit[] = {
    7, 64, // 01000000
    4, 207, // 11001111
    6, 240, // 11110000
    13, 0, // 00000000
    14, 0, // 00000000
    15, 0, // 00000000
    16, 0, // 00000000
    17, 0, // 00000000
    18, 0, // 00000000
    8, 0, // 00000000
    0, 0, // 00000000
    0, 4, // 00000100
    0, 0 // 00000000
};

const int DSPAddress = 68; // 01000100

const char* arrowUpIcon = R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 512 512"><path d="M256 512A256 256 0 1 0 256 0a256 256 0 1 0 0
512zM135.1 217.4l107.1-99.9c3.8-3.5 8.7-5.5 13.8-5.5s10.1 2 13.8 5.5l107.1
99.9c4.5 4.2 7.1 10.1 7.1 16.3c0 12.3-10 22.3-22.3 22.3H304v96c0 17.7-14.3 32-
32 32H240c-17.7 0-32-14.3-32-32V256H150.3C138 256 128 246 128 233.7c0-6.2 2.6-
12.1 7.1-16.3z"/></svg>)SVG";
```

```

const char* arrowDownIcon = R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 512 512"><path d="M256 0a256 256 0 1 0 0 512A256 256 0 1 0 256
0zM376.9 294.6L269.8 394.5c-3.8 3.5-8.7 5.5-13.8 5.5s-10.1-2-13.8-5.5L135.1
294.6c-4.5-4.2-7.1-10.1-7.1-16.3c0-12.3 10-22.3 22.3-22.3l57.7 0 0-96c0-17.7
14.3-32 32-32l32 0c17.7 0 32 14.3 32 32l0 96 57.7 0c12.3 0 22.3 10 22.3 22.3c0
6.2-2.6 12.1-7.1 16.3z"/></svg>)SVG";

void writeToDSP(byte reg, byte value) {
    uint8_t array[2];
    array[0] = reg;
    array[1] = value;
    Wire.beginTransaction(DSPAddress);
    Wire.write(array, 2);
    Wire.endTransmission();
}

byte volumeMapping(int volumePercent) {
    switch (volumePercent) {
        case 0:    return 63; // 111111
        case 10:   return 57; // 111001
        case 20:   return 51; // 110011
        case 30:   return 43; // 101011
        case 40:   return 38; // 100110
        case 50:   return 0;  // 000000
        case 60:   return 6;  // 000110
        case 70:   return 12; // 001100
        case 80:   return 17; // 010001
        case 90:   return 20; // 010100
        case 100:  return 23; // 010111
        default:   return 63; // 111111
    }
}

byte dBMapping(int dbValue) {
    if (dbValue < 0) {
        return (dbValue + 15);
    } else {
        return (31 - dbValue);
    }
}

void setupDSP() {
    writeToDSP(8, volumeMapping(volume));
    writeToDSP(0, 0);
    writeToDSP(11, dBMapping(bass));
    writeToDSP(10, dBMapping(middle));
    writeToDSP(9, dBMapping(treble));
}

```

```

String replaceAll(String str, const String& from, const String& to) {
    int position = str.indexOf(from);
    while (position != -1) {
        str = str.substring(0, position) + to + str.substring(position +
from.length());
        position = str.indexOf(from, position + to.length());
    }
    return str;
}

String getHTML() {
    String html = R"rawliteral(
        <!DOCTYPE html>
        <html>
            <head>
                <title>Amplifier CC</title>
                <meta name="viewport" content="width=device-width, initial-
scale=1.0">
            <style>
                body {
                    font-family: Arial, sans-serif;
                    background: #222;
                    color: #fff;
                    text-align: center;
                    margin: 50px 5px;
                }
                h4 {
                    margin: 30px 0px 10px;
                    font-variant: small-caps;
                    font-size: 15px;
                }
                h4:first-of-type {
                    margin: 20px 0px 10px;
                }
                .container {
                    display: inline-block;
                    padding: 30px 20px;
                    margin: 20px;
                    background: #111;
                    border: 2px solid #0b41ff;
                    border-radius: 10px;
                    box-shadow: 0px 0px 20px #0b41ff;
                }
                .button-pair {
                    display: flex;
                    justify-content: center;
                    margin-bottom: 20px;
                }
                button {

```

```

        display: flex;
        align-items: center;
        justify-content: center;
        background: whitesmoke;
        color: #222;
        border: 2px solid #208120;
        border-radius: 10px;
        padding: 15px 25px;
        margin: 0px 10px;
        cursor: pointer;
        font-size: 16px;
    }
    button:hover,
    button:active {
        background: #303030;
        border: 2px solid #29ff01;
    }
    svg {
        height: 24px;
        fill: #208120;
    }
    button:hover svg,
    button:active svg {
        fill: #29ff01;
    }
    #feedback {
        margin-top: 20px;
        font-weight: bold;
        font-size: 20px;
    }
}
</style>
</head>
<body>
    <div class="container">
        <h4 id="volumeDisplay">Volume: $volume%</h4>
        <div class="button-pair">
            <button onclick="alterStatus('/volumeup',
'volumeDisplay', '%')">$arrowUpIcon</button>
            <button onclick="alterStatus('/volumedown',
'volumeDisplay', '%')">$arrowDownIcon</button>
        </div>
        <h4 id="bassDisplay">Bass: $bass dB</h4>
        <div class="button-pair">
            <button onclick="alterStatus('/bassup', 'bassDisplay',
' dB')">$arrowUpIcon</button>
            <button onclick="alterStatus('/bassdown',
'bassDisplay', ' dB')">$arrowDownIcon</button>
        </div>
        <h4 id="middleDisplay">Middle: $middle dB</h4>

```

```

        <div class="button-pair">
            <button onclick="alterStatus('/middleup',
'middleDisplay', ' dB')">$arrowUpIcon</button>
            <button onclick="alterStatus('/middledown',
'middleDisplay', ' dB')">$arrowDownIcon</button>
        </div>
        <h4 id="trebleDisplay">Treble: $treble dB</h4>
        <div class="button-pair">
            <button onclick="alterStatus('/trebleup',
'trebleDisplay', ' dB')">$arrowUpIcon</button>
            <button onclick="alterStatus('/trebledown',
'trebleDisplay', ' dB')">$arrowDownIcon</button>
        </div>
    </div>
    <div id="feedback"></div>
    <script>
        function alterStatus(url, elementId, unit) {
            fetch(url, { method: 'POST' })
                .then(response => response.json())
                .then(data => {
                    let label = elementId.replace('Display', '');
                    label = label.charAt(0).toUpperCase() +
label.slice(1);
                    document.getElementById(elementId).textContent
= label + ': ' + data.value + unit;
                    generateFeedback('Status updated.', true);
                })
                .catch(error => generateFeedback('Error: ' +
error.message + '.', false));
        }
        function generateFeedback(message, isSuccess) {
            const feedback = document.getElementById('feedback');
            feedback.textContent = message;
            feedback.style.color = isSuccess ? '#228b22' :
'#922724';
        }
    </script>
</body>
</html>
)rawliteral";

html = replaceAll(html, "$arrowUpIcon", arrowUpIcon);
html = replaceAll(html, "$arrowDownIcon", arrowDownIcon);
html = replaceAll(html, "$volume", String(volume));
html = replaceAll(html, "$bass", String(bass));
html = replaceAll(html, "$middle", String(middle));
html = replaceAll(html, "$treble", String(treble));

return html;

```

```

}

void setup() {
  pinMode(DSPMute, OUTPUT);
  pinMode(ampMute, OUTPUT);
  pinMode(ampSDN, OUTPUT);
  digitalWrite(DSPMute, HIGH);
  digitalWrite(ampMute, LOW);
  digitalWrite(ampSDN, LOW);

  WiFi.mode(WIFI_STA);
  WiFi.config(staticIP, gateway, subnet);
  WiFi.begin(ssid, password);
  Wire.begin(D2, D1);
  for (int i = 0; i < (int)sizeof(I2CInit); i += 2) {
    writeToDSP(I2CInit[i], I2CInit[i+1]);
  }
  setupDSP();

  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(200, "text/html", getHTML());
  });

  server.on("/volumeup", HTTP_POST, [](AsyncWebServerRequest *request){
    int oldVolume = volume;
    volume = min(100, volume + 10);
    writeToDSP(8, volumeMapping(volume));
    if (oldVolume == 0 && volume > 0) {
      digitalWrite(ampMute, LOW);
    }
    request->send(200, "application/json", "{\"value\":" + String(volume)
+ "}");
  });

  server.on("/volumedown", HTTP_POST, [](AsyncWebServerRequest *request){
    int oldVolume = volume;
    volume = max(0, volume - 10);
    writeToDSP(8, volumeMapping(volume));
    if (volume == 0 && oldVolume > 0) {
      digitalWrite(ampMute, HIGH);
    }
    request->send(200, "application/json", "{\"value\":" + String(volume)
+ "}");
  });

  server.on("/bassup", HTTP_POST, [](AsyncWebServerRequest *request){
    bass = min(15, bass + 1);
    writeToDSP(11, dBMapping(bass));
  });
}

```

```

        request->send(200, "application/json", "{\"value\":" + String(bass) +
"}");
    });

    server.on("/bassdown", HTTP_POST, [](AsyncWebServerRequest *request){
        bass = max(-15, bass - 1);
        writeToDSP(11, dBMapping(bass));
        request->send(200, "application/json", "{\"value\":" + String(bass) +
"}");
    });

    server.on("/middleup", HTTP_POST, [](AsyncWebServerRequest *request){
        middle = min(15, middle + 1);
        writeToDSP(10, dBMapping(middle));
        request->send(200, "application/json", "{\"value\":" + String(middle)
+ "}");
    });

    server.on("/middledown", HTTP_POST, [](AsyncWebServerRequest *request){
        middle = max(-15, middle - 1);
        writeToDSP(10, dBMapping(middle));
        request->send(200, "application/json", "{\"value\":" + String(middle)
+ "}");
    });

    server.on("/trebleup", HTTP_POST, [](AsyncWebServerRequest *request){
        treble = min(15, treble + 1);
        writeToDSP(9, dBMapping(treble));
        request->send(200, "application/json", "{\"value\":" + String(treble)
+ "}");
    });

    server.on("/trebledown", HTTP_POST, [](AsyncWebServerRequest *request){
        treble = max(-15, treble - 1);
        writeToDSP(9, dBMapping(treble));
        request->send(200, "application/json", "{\"value\":" + String(treble)
+ "}");
    });

    delay(500);
    digitalWrite(ampSDN, HIGH);
    server.begin();
}

void loop() {
}

```