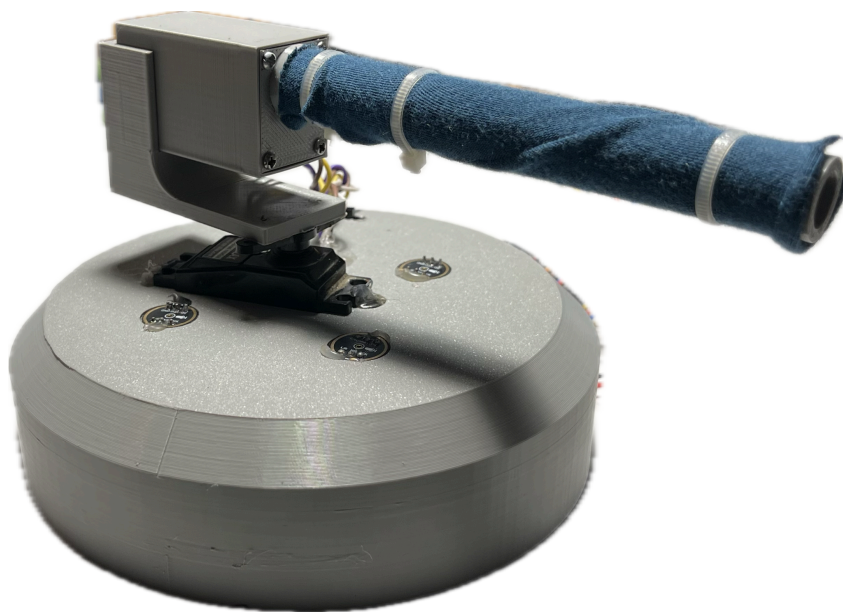


ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σχεδιασμός και Ανάπτυξη Κατευθυντικού Μικροφώνου»



Του φοιτητή :
Μπαχαράκη Κυριάκου
Παλαβάτσιου Κωνσταντίνου Μάριου
Αρ. Μητρώου:
517313
517331

Επιβλέπων
Ιορδάνης Κιοσκερίδης
Καθηγητής

Ημερομηνία ΣΕΠΤΕΜΒΡΙΟΣ 2026

Τίτλος Δ.Ε. Σχεδιασμός και Ανάπτυξη Κατευθυντικού Μικροφώνου

Κωδικός Δ.Ε. 26260

Ονοματεπώνυμο φοιτητή

Μπαχαράκης Κυριάκος

Παλαβάτσιος Κωνσταντίνος Μάριος

Ονοματεπώνυμο εισηγητή Κιοσκεριδης Ιορδάνης

Ημερομηνία ανάληψης Δ.Ε. 16/05/2026

Ημερομηνία περάτωσης Δ.Ε. 05/09/2026

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

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

Η έγκριση της 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.

«Στην οικογένειά μας...»

Πρόλογος

Η ιδέα για την εκπόνηση της παρούσας πτυχιακής εργασίας προήλθε από το διαρκώς αυξανόμενο ενδιαφέρον μου για την επεξεργασία ψηφιακού σήματος (DSP), την τεχνητή νοημοσύνη στο άκρο του δικτύου (Edge AI/TinyML) και τον σχεδιασμό ενσωματωμένων συστημάτων. Στη σύγχρονη εποχή, η αλληλεπίδραση ανθρώπου-μηχανής μέσω φωνής και η ανάγκη για καθαρή λήψη ήχου σε περιβάλλοντα με έντονο θόρυβο αποτελούν κρίσιμες τεχνολογικές προκλήσεις. Παρατηρώντας την αδυναμία των συμβατικών, πανκατευθυντικών μικροφώνων να απομονώσουν την επιθυμητή πηγή ήχου, γεννήθηκε η ανάγκη για τη δημιουργία μιας διάταξης που όχι μόνο θα "ακούει", αλλά και θα "αντιλαμβάνεται" τον χώρο γύρω της.

Στόχος μου ήταν να γεφυρώσω τη θεωρία των αλγορίθμων εντοπισμού ηχητικής πηγής (όπως ο αλγόριθμος GCC-PHAT) με την πρακτική υλοποίηση σε πραγματικό χρόνο, χρησιμοποιώντας μικροελεγκτές χαμηλού κόστους αλλά υψηλών δυνατοτήτων, όπως ο ESP32-S3. Η πρόκληση του να συνδυαστεί ο μηχανολογικός σχεδιασμός (κίνηση με σερβοκινητήρα), ο ηλεκτρονικός σχεδιασμός (μικρόφωνα MEMS) και η ανάπτυξη πολύπλοκου λογισμικού (Firmware, TinyML, Flutter App) αποτέλεσε το μεγαλύτερο κίνητρο. Αυτό το έργο αποτελεί το επιστέγασμα των σπουδών μου, ενσωματώνοντας γνώσεις από πολλαπλά γνωστικά αντικείμενα του Τμήματος Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων, με όραμα τη δημιουργία μιας συσκευής που θα μπορούσε να βρει άμεση εφαρμογή σε τηλεδιασκέψεις, συστήματα ασφαλείας και ρομποτικές διατάξεις.

Περίληψη

Στα πλαίσια της παρούσας εργασίας, μελετήθηκε, σχεδιάστηκε και κατασκευάστηκε εξ ολοκλήρου ένα έξυπνο, κατευθυντικό σύστημα μικροφώνου (Smart Directional Microphone) βασισμένο στον μικροελεγκτή ESP32-S3 [4]. Το σύστημα ενσωματώνει μια διάταξη τεσσάρων ψηφιακών μικροφώνων (MEMS INMP441) [8] και ενός κεντρικού αναλογικού (MAX9814), το οποίο λειτουργεί ως shotgun μικρόφωνο [14], επιτρέποντας τη χωρική αντίληψη του ήχου. Η συσκευή είναι τοποθετημένη σε προσαρμοσμένο 3D εκτυπωμένο περίβλημα και διαθέτει μηχανισμό περιστροφής 360 μοιρών μέσω σερβοκινητήρα (MG996R) [7], επιτρέποντας τη φυσική στόχευση της πηγής του ήχου..

Η λειτουργία του συστήματος διακρίνεται σε τρεις καταστάσεις (Modes): Στη χειροκίνητη λειτουργία (Manual Mode), ο χρήστης ελέγχει τη γωνία του μικροφώνου μέσω μιας εφαρμογής. Στην αυτόματη λειτουργία (Auto Mode - SSL), το σύστημα εκτελεί σε πραγματικό χρόνο αλγορίθμους Time Difference of Arrival (TDOA) και Generalized Cross Correlation (GCC-PHAT) για τον εντοπισμό της γωνίας άφιξης του ήχου, στρέφοντας το μικρόφωνο προς την πηγή. Στην έξυπνη λειτουργία (Smart Mode - TinyML), το σύστημα αξιοποιεί Συνελκτικά Νευρωνικά Δίκτυα (CNN) σχεδιασμένα μέσω του Edge Impulse, ώστε να ταξινομεί ηχητικά συμβάντα. Η καταγραφή των ηχητικών δεδομένων στην κάρτα MicroSD πραγματοποιείται μόνο όταν ανιχνεύεται ανθρώπινη ομιλία ή καθορισμένος στόχος, εξοικονομώντας πόρους και χώρο αποθήκευσης.

Η διαχείριση και η απομακρυσμένη επικοινωνία επιτυγχάνεται χωρίς την ανάγκη εξωτερικού δρομολογητή ή εγκατάστασης μέσω USB, καθώς ο ESP32-S3 λειτουργεί ως αυτόνομο Σημείο Πρόσβασης (SoftAP). Μέσω ενός ενσωματωμένου Captive Portal, ο χρήστης ελέγχει τη συσκευή ασύρματα από τον φυλλομετρητή (browser) του κινητού του ή μέσω εφαρμογής Flutter. Ο έλεγχος της διάταξης και η μεταφορά εντολών πραγματοποιούνται ακαριαία με χρήση μηνυμάτων JSON μέσω μόνιμης σύνδεσης WebSocket (θύρα 81). Παράλληλα, η λήψη ζωντανής ροής ήχου (Live Audio Streaming) επιτυγχάνεται με ελάχιστη καθυστέρηση, μεταδίδοντας τα ακατέργαστα ηχητικά δεδομένα (PCM) ως binary frames απευθείας μέσω του ίδιου καναλιού WebSocket. Το σύστημα τροφοδοτείται αυτόνομα από μπαταρία 18650 με BMS και κύκλωμα ενίσχυσης τάσης (Step-Up Converter).

«Directional Microphone Design and Development»

«name place holder»

Abstract

In the context of this thesis, a smart directional microphone system was studied, designed, and fully developed based on the ESP32-S3 microcontroller [4]. The system integrates a microphone array consisting of four digital MEMS microphones (INMP441) [8] and one central analog microphone (MAX9814) acting as a shotgun microphone [14], enabling spatial audio perception. The device is housed in a custom 3D-printed enclosure and features a 360-degree rotation mechanism driven by an MG996R Servo motor [7], allowing physical targeting of the sound source.

The system's operation is divided into three modes: In Manual Mode, the user controls the microphone's angle via an application. In Auto Mode (Sound Source Localization - SSL), the system executes Time Difference of Arrival (TDOA) and Generalized Cross Correlation (GCC-PHAT) algorithms in real-time to determine the angle of arrival, rotating the microphone towards the source. In Smart Mode (TinyML), the system utilizes Convolutional Neural Networks (CNNs) designed via Edge Impulse to classify audio events, triggering recording to a MicroSD card only when human speech or a specific acoustic target is detected, thus optimizing resource and storage usage.

Management and remote communication is achieved without the need for an external router or installation via USB, as the ESP32-S3 operates as a standalone Access Point (SoftAP). Through a built-in Captive Portal, the user controls the device wirelessly from their mobile browser or via a Flutter application. Control of the device and command transfer are performed instantaneously using JSON messages via a persistent WebSocket connection (port 81). At the same time, live audio streaming is achieved with minimal latency, transmitting raw audio data (PCM) as binary frames directly through the same WebSocket channel. The system is powered autonomously by an 18650 battery with BMS and a voltage boost circuit (Step-Up Converter).

Ευχαριστίες

Η ολοκλήρωση αυτής της πτυχιακής εργασίας δεν θα ήταν δυνατή χωρίς την αμέριστη υποστήριξη, καθοδήγηση και ενθάρρυνση των ανθρώπων που στάθηκαν δίπλα μου. Αρχικά, θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου στον επιβλέποντα καθηγητή μου, Ιορδάνη Κιοσκερίδη, για την πολύτιμη βοήθεια, τις εξαιρετικές συμβουλές του και τον χρόνο που αφιέρωσε κατά τη διάρκεια της εκπόνησης αυτού του έργου. Η εμπειρία και οι κατευθύνσεις του ήταν καθοριστικές για την επίλυση πολύπλοκων προβλημάτων, τόσο σε επίπεδο hardware όσο και σε επίπεδο αλγορίθμων ψηφιακής επεξεργασίας.

Επίσης, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου, η οποία αποτέλεσε το βασικό μου στήριγμα καθ' όλη τη διάρκεια των σπουδών μου, παρέχοντάς μου την ηθική και υλική υποστήριξη που χρειάζομαι για να επικεντρωθώ στους στόχους μου. Τέλος, ευχαριστώ τους φίλους και συμφοιτητές μου για τις ατέλειωτες ώρες συζητήσεων, την ανταλλαγή ιδεών και την αλληλοβοήθεια που έκαναν αυτή την ακαδημαϊκή διαδρομή μια αξέχαστη εμπειρία.

Περιεχόμενα

Πρόλογος.....	4
Περίληψη.....	5
Abstract.....	6
Ευχαριστίες.....	7
Περιεχόμενα.....	8
Κατάλογος Εικόνων.....	11
Κατάλογος Πινάκων.....	12
Συντομογραφίες.....	14
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	17
1.1 Αντικείμενο και Σκοπός της Εργασίας.....	17
1.2 Τι είναι τα Κατευθυντικά Μικρόφωνα.....	18
1.3 Εφαρμογές.....	20
1.4 Προβλήματα Ανίχνευσης Ήχου.....	21
1.5 Στόχοι Εργασίας.....	21
ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΙΑ ΗΧΟΥ ΚΑΙ DSP.....	22
2.1 Βασικές Έννοιες Ήχου.....	22
2.2 Ψηφιοποίηση Ήχου.....	23
2.3 Sampling (Δειγματοληψία) και Θεώρημα Nyquist.....	23
2.4 PCM Audio.....	24
2.5 FFT (Fast Fourier Transform).....	24
2.6 Spectrograms (Φασματογράμματα).....	25
2.6 Beamforming (Μορφοποίηση Δέσμης).....	26
2.7 Sound Source Localization (SSL).....	27
2.8 TDOA (Time Difference Of Arrival).....	27
2.9 GCC-PHAT (Generalized Cross Correlation with Phase Transform).....	28
ΚΕΦΑΛΑΙΟ 3: ESP32-S3 ΚΑΙ HARDWARE.....	29
3.1 ESP32-S3 Architecture.....	29
3.2 Πρωτόκολλο I2S (Inter-IC Sound).....	31
3.2.1 Αρχιτεκτονική και Σήματα του I2S.....	31
3.2.2 Η σημασία της καθυστέρησης του ενός Bit (1-Bit Delay).....	32
3.2.3 Άμεση Πρόσβαση στη Μνήμη (DMA - Direct Memory Access).....	32
3.3 MEMS Microphones (INMP441).....	33
3.4 Αναλογικό Μικρόφωνο MAX9814.....	34
3.5 Σερβοκινητήρας MG996R.....	35
3.6 MicroSD (Αποθήκευση Δεδομένων).....	36

3.7 Τροφοδοσία και Ενεργειακή Διαχείριση.....	37
ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	39
4.1 Αρχιτεκτονική Συστήματος (Block Diagram).....	40
4.2 Διάταξη Μικροφώνων (Microphone Array Configuration).....	41
4.3 Υπολογισμός Γωνιακής Ανάλυσης.....	42
4.4 Σχεδιασμός PCB / Wiring.....	44
4.5 Κατασκευή Τυπωμένης Πλακέτας (PCB Fabrication).....	48
ΚΕΦΑΛΑΙΟ 5: ΜΗΧΑΝΟΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ 3D ΕΚΤΥΠΩΣΗ.....	51
5.1 Εισαγωγή στον Σχεδιασμό CAD και την Εκτύπωση.....	52
5.2 Σχεδιασμός Κελύφους και Διάταξης Ηλεκτρονικών (Το Καλούπι).....	53
5.3 Σύστημα Περιστροφής και Βάση Στήριξης.....	54
5.4 Σχεδιασμός Κατευθυντικού "Κανονιού" (Shotgun Tube).....	55
ΚΕΦΑΛΑΙΟ 6: ΥΛΟΠΟΙΗΣΗ FIRMWARE.....	57
6.1 Περιβάλλον Ανάπτυξης.....	58
6.2 I2S Audio Acquisition.....	59
6.3 Audio Buffers.....	60
6.5 Έλεγχος Σερβοκινητήρα.....	62
6.6 FreeRTOS Tasks.....	63
6.7 Καταγραφή Δεδομένων και Τηλεμετρία (Data Logging).....	64
ΚΕΦΑΛΑΙΟ 7: SOUND SOURCE LOCALIZATION.....	65
7.1 Θεωρία TDOA (Time Difference Of Arrival) - Εμβάθυνση.....	65
7.2 Ο Αλγόριθμος GCC-PHAT (Με εξισώσεις).....	66
7.3 Υπολογισμός Γωνίας (θ).....	67
7.4 Υβριδικός Αλγόριθμος Εντοπισμού (GCC-PHAT & Energy Excess).....	68
7.5 Παραδείγματα (0° , 45° , 90° , 135° , 180°).....	68
ΚΕΦΑΛΑΙΟ 8: TINY ML ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΗΧΟΥ.....	70
8.1 Μηχανική Μάθηση στα Άκρα του Δικτύου (Edge AI / TinyML).....	71
8.2 Μικρο-Μηχανική Μάθηση (TinyML) και Βελτιστοποίηση Μοντέλων.....	71
8.3 Ανάπτυξη Μοντέλου μέσω της Πλατφόρμας Edge Impulse.....	72
8.4 Συγκρότηση και Διαχείριση του Συνόλου Δεδομένων (Dataset Creation).....	74
8.5 Αρχιτεκτονική Συνελκτικού Νευρωνικού Δικτύου (CNN).....	75
8.6 Speech Detection.....	76
8.7 Smart Recording.....	77
ΚΕΦΑΛΑΙΟ 9: FLUTTER APPLICATION.....	78
9.1 Αρχιτεκτονική Εφαρμογής (Application Architecture).....	79
9.2 Λειτουργία Σημείου Πρόσβασης (ESP32 AP Mode) και Δικτύωση.....	80
9.3 Αναμετάδοση Ήχου μέσω Πρωτοκόλλου WebSocket (Audio Streaming).....	81
9.4 Σχεδιασμός Γραφικής Διεπαφής Χρήστη (User Interface).....	82
9.5 Χειροκίνητη Λειτουργία (Manual Mode).....	83

9.6 Αυτόματη Λειτουργία Εντοπισμού (Auto Mode).....	85
9.7 Έξυπνη Λειτουργία Τεχνητής Νοημοσύνης (Smart Mode).....	87
ΚΕΦΑΛΑΙΟ 10: ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	89
10.1 Κατανάλωση Ρεύματος.....	89
10.2 Latency (Χρόνος Απόκρισης).....	89
10.3 Ακρίβεια Εντοπισμού (SSL / 8 Τομείς).....	90
10.4 Ακρίβεια Ταξινομητή (TinyML On-Device).....	91
10.5 Συγκριτικός Πίνακας Δυνατοτήτων.....	91
10.6 Μετρήσεις Παλμογράφου.....	92
ΚΕΦΑΛΑΙΟ 11: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	95
11.1 Συμπεράσματα.....	95
11.2 Μελλοντικές Επεκτάσεις.....	95
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	97
ΠΑΡΑΡΤΗΜΑ Α: BOM (Bill of Materials).....	98
ΠΑΡΑΡΤΗΜΑ Β: Schematics.....	99
ΠΑΡΑΡΤΗΜΑ Γ: ESP32 Source Code.....	102
ΠΑΡΑΡΤΗΜΑ Δ: Flutter Source Code.....	113

Κατάλογος Εικόνων

Εικόνα 1. How microphones work	Πηγή:Αναζήτηση google	18
Εικόνα 2. microphone pickup patterns	Πηγή:Αναζήτηση google	19
Εικόνα 3. Εφαρμογές Μικροφώνων	Πηγή:created in google flow	20
Εικόνα 4. digitization of sound	Πηγή:Αναζήτηση google	23
Εικόνα 5. Spectrograms	Πηγή:Wikipedia	25
Εικόνα 6. Μορφοποίηση Δέσμης	Πηγή:created in google flow	26
Εικόνα 7. Time difference of arrival	Πηγή:created in google flow	27
Εικόνα 8. Sound Source Localization (SSL)	Πηγή:Αναζήτηση google	28
Εικόνα 9. ESP32-S3-DevKitC-1 Module	Πηγή:docs.espressif.com	29
Εικόνα 10.Esp32 Pin Layout	Πηγή:docs.espressif.com	30
Εικόνα 11. protocol i2s explained	Πηγή:Αναζήτηση google	31
Εικόνα 12. dma direct memory access diagram	Πηγή:Αναζήτηση google	32
Εικόνα 13. inmp441 mems microphones	Πηγή:Αναζήτηση google	33
Εικόνα 14. Αναλογικό Μικρόφωνο MAX9814	Πηγή:www.helladigital.gr	34
Εικόνα 15. Servo MG996R	Πηγή:Αναζήτηση google	35
Εικόνα 16. microSD Module	Πηγή:www.grobotronics.com	36
Εικόνα 17. TP4056 wiring	Πηγή:Αναζήτηση google	37
Εικόνα 18. Συνδεσμολογία Παροχής ενέργειας	Πηγή:created in google flow	38
Εικόνα 19. Σχηματικό Ηλεκτρονικού Συστήματος	Πηγή:screenshot kiCad, pc	39
Εικόνα 20. Block Diagram Συστήματος	Πηγή:created in google flow	40
Εικόνα 21. 2D Συστοιχία μικροφώνων INMP441	Πηγή:created in google flow	41
Εικόνα 22. Υπολογισμός Γωνίας ανάλυσης	Πηγή:created in google flow	43
Εικόνα 23.3D Απεικόνιση Πλακέτας PCB	Πηγή:screenshot kiCad, pc	44
Εικόνα 24.Πυκνωτές Αποσύζευξης στην Πλακέτα δοκιμών	Πηγή:κάμερα κινητού	45
Εικόνα 25.Επιλογέας Κέρδους Μικροφώνου	Πηγή:κάμερα κινητού	46
Εικόνα 26.Προετοιμασία πλακέτας PCB	Πηγή:κάμερα κινητού.	48
Εικόνα 27 Θερμική Μεταφορά Γραφίτη σε PCB	Πηγή:κάμερα κινητού.	49
Εικόνα 28.Αποχάλκωση Πλακέτας PCB	Πηγή:κάμερα κινητού	50
Εικόνα 29.Ολοκληρωμένο Σχέδιο Συστήματος	Πηγή:screenshot fushion360, pc	51
Εικόνα 30.3D Printing Βάσης Συστήματος	Πηγή:κάμερα κινητού	52
Εικόνα 31.Σχεδιασμός βάσης Συστήματος	Πηγή:screenshot fushion360, pc	53
Εικόνα 32.Σχεδιασμός βάσης Περιστροφής	Πηγή:screenshot fushion360, pc	54
Εικόνα 33.Σχεδιασμός κατευθυντικού κανονιού	Πηγή:screenshot fushion360, pc	55
Εικόνα 34.Βαμβακερό κάλυμμα κανονιού	Πηγή:κάμερα κινητού	56
Εικόνα 35.Machine learning embedded	Πηγή:Αναζήτηση google	57
Εικόνα 36.platformio ide picture	Πηγή:Αναζήτηση google	58
Εικόνα 37.protocol i2s explained	Πηγή:Αναζήτηση google	59
Εικόνα 38.Esp32s3 Audio processing	Πηγή:created in google flow	60
Εικόνα 39.Esp32 servo control	Πηγή:Αναζήτηση google	62
Εικόνα 40.Time difference of arrival	Πηγή:created in google flow	65

Εικόνα 41. GCC-PHAT algorytm	Πηγή:Αναζήτηση google	66
Εικόνα 42. Direction of Arrival finding	Πηγή:created in google flow	67
Εικόνα 43. TinyML & Embedded systems	Πηγή:created in google flow	70
Εικόνα 44. TinyML Block diagram	Πηγή:Αναζήτηση google	72
Εικόνα 45. Esp32 speech recognition	Πηγή:Αναζήτηση google	73
Εικόνα 46. Esp32 speech recognition	Πηγή:Αναζήτηση google	76
Εικόνα 47. Flutter App Developing	Πηγή:Αναζήτηση google	78
Εικόνα 48. Είσοδος στο websocket	Πηγή:screenshot κινητού	81
Εικόνα 49. Αρχική Μενού εφαρμογής	Πηγή:screenshot κινητού	82
Εικόνα 50. Manual λειτουργία εφαρμογής	Πηγή:screenshot κινητού	84
Εικόνα 51. Auto λειτουργία εφαρμογής	Πηγή:screenshot κινητού	86
Εικόνα 52. smart λειτουργία εφαρμογής	Πηγή:screenshot κινητού	87

Κατάλογος Πινάκων

Πίνακας 1: Ακρίβεια Χωρικής Ταξινόμησης σε 8 Τομείς.	90
Πίνακας 2: Συγκριτικός πίνακας Δυνατοτήτων	91
Πίνακας 3: κόστος υλικων	96

Συντομογραφίες

3D: 3-Dimensional (Τρισδιάστατος)
ACK: Acknowledgement (Επιβεβαίωση λήψης πακέτων)
ADC: Analog-to-Digital Converter (Μετατροπέας Αναλογικού σε Ψηφιακό)
AEC: Acoustic Echo Cancellation (Ακύρωση Ηχούς)
AGC: Automatic Gain Control (Αυτόματος Έλεγχος Κέρδους)
AI: Artificial Intelligence (Τεχνητή Νοημοσύνη)
ANC: Adaptive Noise Cancellation (Προσαρμοστική Ακύρωση Θορύβου)
 AoA: Angle of Arrival (Γωνία Άφιξης)
AP: Access Point (Σημείο Πρόσβασης)
APK: Android Package Kit (Πακέτο Εφαρμογής Android)
ASIC: Application-Specific Integrated Circuit (Ολοκληρωμένο Κύκλωμα Ειδικής Εφαρμογής)
BCLK: Bit Clock (Ρολόι χρονισμού των bits)
BLE / BT: Bluetooth Low Energy / Bluetooth (Bluetooth Χαμηλής Ενέργειας)
BLoC: Business Logic Component (Μοτίβο διαχείρισης κατάστασης στο Flutter)
BMS: Battery Management System (Σύστημα Διαχείρισης Μπαταρίας)
BOM: Bill of Materials (Λίστα Υλικών)
CAD: Computer-Aided Design (Σχεδίαση μέσω υπολογιστή)
CD: Compact Disc (Συμπαγής Δίσκος)
CNN: Convolutional Neural Network (Συνελικτικό Νευρωνικό Δίκτυο)
CPU: Central Processing Unit (Κεντρική Μονάδα Επεξεργασίας)
CS: Chip Select (Επιλογή Τσιπ / Ακίδα διαύλου SPI)
dB: Decibel (Μονάδα μέτρησης ηχητικής πίεσης/έντασης)
DC: Direct Current (Συνεχές Ρεύμα)
DFT: Discrete Fourier Transform (Διακριτός Μετασχηματισμός Fourier)
DIN/DOUT: Data In / Data Out (Γραμμές δεδομένων)
DMA: Direct Memory Access (Άμεση Πρόσβαση στη Μνήμη)
DNS: Domain Name System (Σύστημα Ονομάτων Χώρου)
DoA: Direction of Arrival (Διεύθυνση Άφιξης)
DSP: Digital Signal Processing (Ψηφιακή Επεξεργασία Σήματος)
EMA: Exponential Moving Average (Εκθετικός Κινητός Μέσος Όρος)
EMI: Electromagnetic Interference (Ηλεκτρομαγνητικές Παρεμβολές)
FAT32 / FATFS: File Allocation Table (Σύστημα Αρχείων)
FFT: Fast Fourier Transform (Ταχύς Μετασχηματισμός Fourier)
fps: Frames per second (Καρέ ανά δευτερόλεπτο)
GCC-PHAT: Generalized Cross Correlation with Phase Transform (Γενικευμένη Ετεροσυσχέτιση με Μετασχηματισμό Φάσης)
GND: Ground (Γείωση)
GUI: Graphical User Interface (Γραφική Διεπαφή Χρήστη)
HTTP: Hypertext Transfer Protocol (Πρωτόκολλο Μεταφοράς Υπερκειμένου)
Hz / kHz: Hertz / Kilohertz (Μονάδα μέτρησης συχνότητας)

I2S: Inter-IC Sound (Πρωτόκολλο Μετάδοσης Ψηφιακού Ήχου)
IDE: Integrated Development Environment (Περιβάλλον Ανάπτυξης)
IFFT: Inverse Fast Fourier Transform (Αντίστροφος Ταχύς Μετασχηματισμός Fourier)
iOS: iPhone Operating System (Λειτουργικό Σύστημα της Apple)
IoT: Internet of Things (Διαδίκτυο των Πραγμάτων)
ISR: Interrupt Service Routine (Ρουτίνα Εξυπηρέτησης Διακοπής)
JSON: JavaScript Object Notation (Μορφή ανταλλαγής δεδομένων)
LED: Light Emitting Diode (Δίοδος Εκπομπής Φωτός)
LEDC: LED Control (Ελεγκτής σημάτων PWM)
LRCLK: Left-Right Clock (Ρολόι επιλογής λέξης/καναλιού)
LwIP: Lightweight IP (Στοιβα πρωτοκόλλων δικτύου)
MCU: Microcontroller Unit (Μικροελεγκτής)
MEMS: Micro Electro Mechanical Systems (Μικροηλεκτρομηχανικά Συστήματα)
MFE: Mel-Filterbank Energy (Ενέργεια Συστοιχίας Φίλτρων κλίμακας Mel)
MISO: Master In Slave Out (Γραμμή δεδομένων διαύλου SPI)
MOSI: Master Out Slave In (Γραμμή δεδομένων διαύλου SPI)
MQTT: Message Queuing Telemetry Transport (Πρωτόκολλο επικοινωνίας δικτύου)
MSB: Most Significant Bit (Το πιο σημαντικό bit)
MVDR: Minimum Variance Distortionless Response (Αλγόριθμος προσαρμοστικού Beamforming)
NN: Neural Network (Νευρωνικό Δίκτυο)
PCB: Printed Circuit Board (Τυπωμένη Πλακέτα)
PCM: Pulse-Code Modulation (Ψηφιακή αναπαράσταση αναλογικών σημάτων)
PETG: Polyethylene Terephthalate Glycol (Υλικό τρισδιάστατης εκτύπωσης)
PLA: Polylactic Acid (Υλικό τρισδιάστατης εκτύπωσης)
PSRAM: Pseudo Static Random Access Memory (Ψευδοστατική μνήμη τυχαίας προσπέλασης)
PTZ: Pan-Tilt-Zoom (Τύπος περιστρεφόμενης κάμερας)
PWM: Pulse Width Modulation (Διαμόρφωση Εύρους Παλμού)
RAM: Random Access Memory (Μνήμη Τυχαίας Προσπέλασης)
REST: Representational State Transfer (Αρχιτεκτονική λογισμικού για διαδικτυακές υπηρεσίες)
RGB: Red Green Blue (Χρωματικό μοντέλο)
RTOS: Real-Time Operating System (Λειτουργικό Σύστημα Πραγματικού Χρόνου)
SCADA: Supervisory Control and Data Acquisition (Συστήματα ελέγχου και απόκτησης δεδομένων)
SCK: Serial Clock (Σειριακό ρολόι χρονισμού)
SD (Κάρτα/Μνήμη): Secure Digital (Ψηφιακή κάρτα μνήμης)
SD (γραμμή): Serial Data (Σειριακά δεδομένα)
SDMMC: Secure Digital MultiMediaCard (Δίσκος επικοινωνίας μνήμης)
SNR: Signal-to-Noise Ratio (Λόγος Σήματος προς Θόρυβο)
SoC: System on a Chip (Σύστημα σε ένα Τσιπ)
SoftAP: Software Enabled Access Point (Σημείο Πρόσβασης μέσω Λογισμικού)
SPI: Serial Peripheral Interface (Σειριακή Περιφερειακή Διεπαφή)
SRAM: Static Random Access Memory (Στατική εσωτερική μνήμη)

SSL: Sound Source Localization (Εντοπισμός Ηχητικής Πηγής)
STFT: Short-Time Fourier Transform (Μετασχηματισμός Fourier Βραχέος Χρόνου)
TCP: Transmission Control Protocol (Πρωτόκολλο Ελέγχου Μετάδοσης)
TDM: Time Division Multiplexing (Πολυπλεξία Διαίρεσης Χρόνου)
TDOA: Time Difference of Arrival (Χρονική Διαφορά Άφιξης)
TinyML: Tiny Machine Learning (Μηχανική Μάθηση σε Μικροελεγκτές)
UDP: User Datagram Protocol (Πρωτόκολλο Δεδομένων Χρήστη)
UI: User Interface (Διεπαφή Χρήστη)
USB: Universal Serial Bus (Ενιαίος Σειριακός Δίαυλος)
UX: User Experience (Εμπειρία Χρήστη)
VAD: Voice Activity Detection (Ανίχνευση Φωνητικής Δραστηριότητας)
VDD: Voltage Drain Drain (Θετική τάση τροφοδοσίας)
VSCode: Visual Studio Code (Περιβάλλον Ανάπτυξης Κώδικα)
VU: Volume Unit (Οπτικοποίηση στάθμης ήχου)
WAV: Waveform Audio File Format (Μορφή αρχείου ήχου)
Wi-Fi: Wireless Fidelity (Ασύρματο Τοπικό Δίκτυο)
WS: Word Select (Πολύι επιλογής λέξης)

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο και Σκοπός της Εργασίας

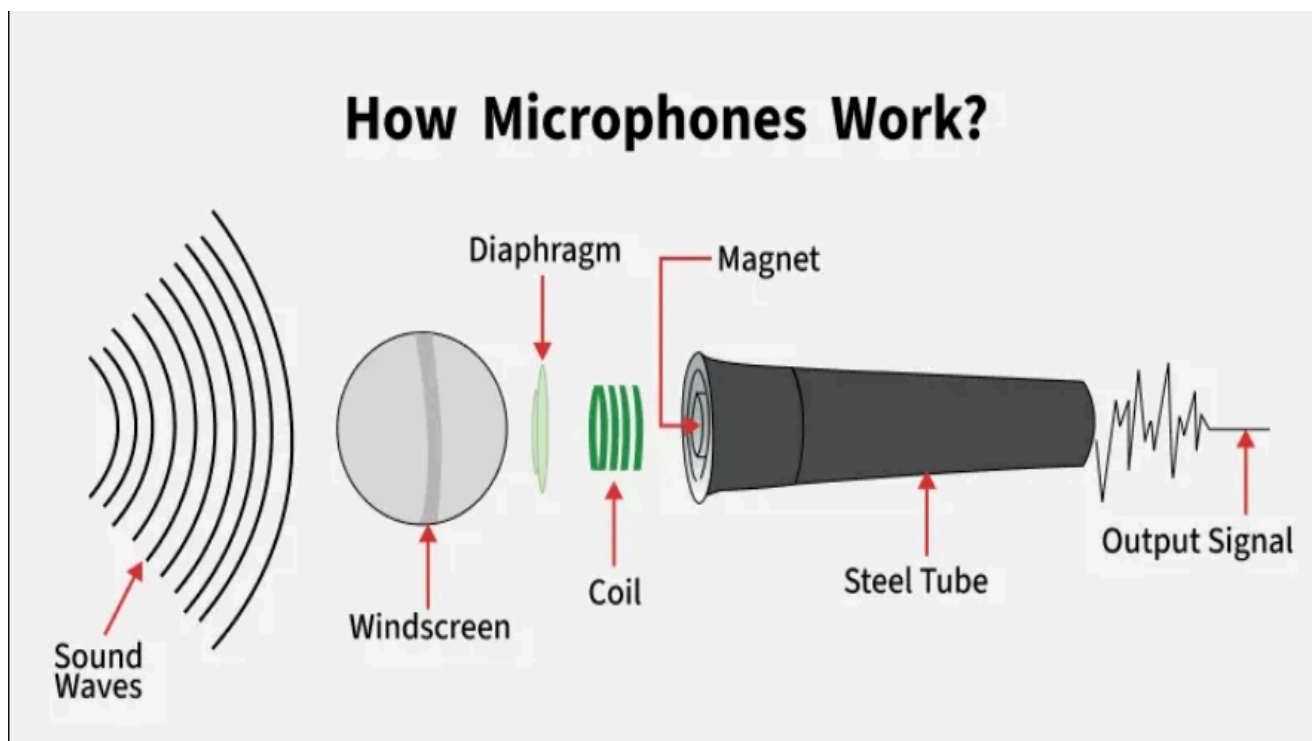
Η ραγδαία εξέλιξη της τεχνολογίας στον τομέα της ψηφιακής επεξεργασίας σήματος (DSP) και της τεχνητής νοημοσύνης έχει μετατοπίσει το υπολογιστικό βάρος από το απομακρυσμένο νέφος (Cloud) απευθείας στο άκρο του δικτύου (Edge AI). Αυτή η μετάβαση έχει επιτρέψει την αποδοτική ενσωμάτωση εξαιρετικά πολύπλοκων μαθηματικών αλγορίθμων σε μικροελεγκτές (Microcontrollers) χαμηλού κόστους και περιορισμένης ενεργειακής κατανάλωσης. Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός, η υλοποίηση, η βελτιστοποίηση και η αξιολόγηση ενός ολοκληρωμένου, αυτόνομου συστήματος, το οποίο περιγράφεται ως «Έξυπνο Κατευθυντικό Μικρόφωνο». Πρόκειται για μια προηγμένη διάταξη υλικού (hardware) και λογισμικού (software), η οποία υπερβαίνει την απλή παθητική καταγραφή του ήχου. Η συσκευή διαθέτει τη δυνατότητα να ανιχνεύει τις ακουστικές μεταβολές στο περιβάλλον, να υπολογίζει με ακρίβεια τη χωρική θέση της ηχητικής πηγής, να στρέφεται μηχανικά προς αυτήν για τη βέλτιστη λήψη του σήματος και, τέλος, να αναλύει το ηχητικό περιεχόμενο σε πραγματικό χρόνο με τη χρήση ενσωματωμένων συνελκτικών νευρωνικών δικτύων.

Παράλληλα, ένας εξίσου θεμελιώδης σκοπός της εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου και εύχρηστου οικοσυστήματος ελέγχου, το οποίο γεφυρώνει τη φυσική κατασκευή με τον τελικό χρήστη. Για τον λόγο αυτό, η προσέγγιση δεν περιορίζεται αποκλειστικά στον σχεδιασμό του υλικού. Αναπτύχθηκε ένα σύγχρονο γραφικό περιβάλλον διεπαφής (GUI), διαθέσιμο τόσο ως εγγενής (Native) εφαρμογή για κινητές συσκευές μέσω του πλαισίου Flutter, όσο και ως ενσωματωμένη ιστοσελίδα (Web UI) που φιλοξενείται απευθείας στη μνήμη του μικροελεγκτή. Το περιβάλλον αυτό επικοινωνεί ασύρματα και αμφίδρομα με την κεντρική μονάδα ESP32-S3. Προκειμένου να διασφαλιστεί η απόλυτη φορητότητα και η ανεξαρτησία του συστήματος, η συσκευή έχει διαμορφωθεί ώστε να λειτουργεί αυτόνομα ως Σημείο Πρόσβασης (SoftAP). Με αυτόν τον τρόπο, επιτρέπει στον χρήστη να αποκτήσει τον πλήρη έλεγχο της μηχανικής στόχευσης και της εναλλαγής των καταστάσεων λειτουργίας, χωρίς να υφίσταται καμία απαίτηση για ενσύρματη σύνδεση (USB) ή για την ύπαρξη κάποιας εξωτερικής υποδομής τοπικού δικτύου (Router).

Τέλος, ιδιαίτερη έμφαση δόθηκε στην αρχιτεκτονική της τηλεπικοινωνιακής ζεύξης. Η επικοινωνία ελέγχου, καθώς και η ταυτόχρονη λήψη ζωντανής ροής ήχου (Live Audio Streaming), υλοποιούνται αποκλειστικά μέσω του πρωτοκόλλου WebSocket. Η συγκεκριμένη τεχνολογία επιλέχθηκε στρατηγικά έναντι του παραδοσιακού HTTP REST και του πρωτοκόλλου UDP, καθώς εξαλείφει τον περιττό διαχειριστικό φόρτο (overhead) των συνεχών αιτημάτων και εξασφαλίζει μια μόνιμη, σταθερή σύνδεση. Η ενοποίηση των καναλιών για τη μεταφορά τόσο των δεδομένων ελέγχου (εντολές JSON) όσο και της ακατέργαστης ακουστικής ροής (binary PCM frames) σε μια κοινή θύρα, επιτυγχάνει την ελαχιστοποίηση της καθυστέρησης (ultra-low latency), προσφέροντας μια απρόσκοπτη και άμεση εμπειρία αλληλεπίδρασης.

1.2 Τι είναι τα Κατευθυντικά Μικρόφωνα

Τα μικρόφωνα είναι ηλεκτροακουστικοί μορφοτροπείς που μετατρέπουν τα ακουστικά κύματα (μεταβολές της πίεσης του αέρα) σε ηλεκτρικά σήματα. Μια από τις πιο κρίσιμες παραμέτρους ενός μικροφώνου είναι το "πολικό του διάγραμμα" (polar pattern), το οποίο καθορίζει την ευαισθησία του ως προς τη γωνία πρόσπτωσης του ήχου. Τα κατευθυντικά μικρόφωνα είναι σχεδιασμένα ώστε να είναι περισσότερο ευαίσθητα στους ήχους που προέρχονται από μια συγκεκριμένη κατεύθυνση, απορρίπτοντας τον θόρυβο από τις υπόλοιπες γωνίες.



Εικόνα 1. How microphones work, Πηγή: Αναζήτηση google

- **Πανκατευθυντικά (Omnidirectional):** Αν και δεν ανήκουν στα κατευθυντικά, αποτελούν τη βάση σύγκρισης [14]. Λαμβάνουν τον ήχο εξίσου καλά από όλες τις κατευθύνσεις (360 μοίρες). Αν και ιδανικά για την καταγραφή της συνολικής ακουστικής ενός χώρου, είναι εντελώς ακατάλληλα για απομόνωση μιας φωνής σε περιβάλλον με θόρυβο.
- **Καρδιοειδή (Cardioid):** Ονομάζονται έτσι λόγω του σχήματος του πολικού τους διαγράμματος που θυμίζει καρδιά [14]. Έχουν τη μέγιστη ευαισθησία στο μπροστινό μέρος (0 μοίρες) και τη μικρότερη στο πίσω μέρος (180 μοίρες). Χρησιμοποιούνται ευρέως σε ζωντανές εμφανίσεις (live vocals) και στούντιο.

- **Shotgun (Υπερ-κατευθυντικά):** Πρόκειται για μικρόφωνα με έντονη κατευθυντικότητα, που μοιάζουν με μακρύ σωλήνα [14]. Λειτουργούν με βάση την αρχή της συμβολής των κυμάτων μέσα στον σωλήνα (interference tube) [14]. Ενισχύουν τους ήχους που έρχονται ακριβώς από μπροστά και ακυρώνουν δραστικά τους ήχους από τα πλάγια [14]. Στην παρούσα εργασία, το MAX9814 αξιοποιείται ως το κεντρικό shotgun μικρόφωνο της διάταξης [14, 15].
- **Συστοιχίες Μικροφώνων (Beamforming Arrays):** Αυτή είναι η πιο σύγχρονη και ευέλικτη προσέγγιση [8, 15]. Αντί να βασιζόμαστε στο φυσικό σχήμα ενός μεμονωμένου μικροφώνου, χρησιμοποιούμε μια διάταξη (array) από πολλαπλά μικρόφωνα [8]. Μέσω ψηφιακής επεξεργασίας σήματος (DSP), ο μικροελεγκτής καθυστερεί (delay) και αθροίζει (sum) τα σήματα, δημιουργώντας μια "ηλεκτρονική κατευθυντικότητα" (beamforming) που μπορεί να στραφεί ψηφιακά προς οποιαδήποτε κατεύθυνση [8, 11].

TYPICAL MICROPHONE POLAR PATTERNS



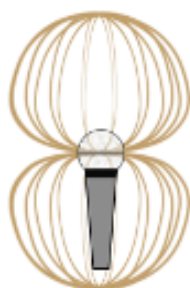
Cardioid



Super-Cardioid



Hyper-Cardioid



Bi-Directional
("Noise Cancelling")



Omni



Shot-Gun

Εικόνα 2. microphone pickup patterns, Πηγή: Αναζήτηση google

1.3 Εφαρμογές

Η ικανότητα μιας συσκευής να εντοπίζει, να στοχεύει και να αναλύει την πηγή ενός ήχου προσφέρει λύσεις σε ένα τεράστιο φάσμα σύγχρονων εφαρμογών:

- **Συστήματα Ασφαλείας και Επιτήρησης (Surveillance):** Ένα έξυπνο κατευθυντικό μικρόφωνο μπορεί να εντοπίζει τον ήχο μιας θραύσης κρυστάλλου ή μιας κραυγής, να στρέψει αυτόματα μια κάμερα PTZ προς το συμβάν και να ταξινομήσει τον ήχο ως κίνδυνο.
- **Τηλεδιασκέψεις (Teleconferencing):** Τα συστήματα εντοπισμού ηχητικής πηγής (SSL) επιτρέπουν στο μικρόφωνο να εστιάζει αποκλειστικά στο άτομο που μιλάει τη δεδομένη στιγμή, ακυρώνοντας τις ομιλίες από διπλανά γραφεία.
- **Drones και Αυτόνομα Οχήματα:** Η χρήση συστοιχιών επιτρέπει τον εντοπισμό εγκλωβισμένων μέσω της φωνής τους, αφαιρώντας τον έντονο θόρυβο από τους έλικες.
- **Ρομποτική (Robotics):** Η ενσωμάτωση του συστήματος επιτρέπει σε ένα ρομπότ να αντιλαμβάνεται τον χώρο σε 360 μοίρες και να στρέφει το "κεφάλι" του προς τον ομιλητή.
- **Έξυπνα Σπίτια (Smart Homes):** Οι ψηφιακοί βοηθοί χρησιμοποιούν κατευθυντικά μικρόφωνα για να καταλαβαίνουν από ποια γωνία προέρχεται η εντολή του χρήστη.
- **Αμυντικές Εφαρμογές (Defense):** Χρησιμοποιούνται για τον εντοπισμό της θέσης ελεύθερων σκοπευτών αναλύοντας τη χρονική διαφορά άφιξης του κρουστικού κύματος.



Εικόνα 3. Εφαρμογές Μικροφώνων, Πηγή: created in google flow

1.4 Προβλήματα Ανίχνευσης Ήχου

Η καταγραφή και ανάλυση του ήχου στο φυσικό περιβάλλον δεν είναι μια αποστειρωμένη διαδικασία. Το πραγματικό περιβάλλον εισάγει σοβαρά προβλήματα:

- **Θόρυβος (Noise):** Χωρίζεται σε στατικό θόρυβο (π.χ. ανεμιστήρας) και παροδικό θόρυβο [13]. Για να λειτουργήσει αποτελεσματικά το TinyML μοντέλο, το Signal-to-Noise Ratio (SNR) πρέπει να είναι επαρκές [3]. Το κατευθυντικό μικρόφωνο λύνει φυσικά αυτό το πρόβλημα.
- **Αντήχηση (Reverberation):** Σε κλειστό χώρο, ο ήχος φτάνει και έπειτα από πολλαπλές ανακλάσεις (ηχώ), δημιουργώντας "φαντάσματα" (ghost sources) στους αλγορίθμους SSL [7, 8]. Η χρήση του μετασχηματισμού GCC-PHAT είναι κρίσιμη ακριβώς επειδή αγνοεί το πλάτος και εστιάζει στη φάση, παρουσιάζοντας ανθεκτικότητα στην αντήχηση [2, 7].
- **Παρεμβολές (Interference):** Ηλεκτρομαγνητικές παρεμβολές στο υλικό μπορούν να παραμορφώσουν το ακουστικό σήμα. Η επιλογή ψηφιακών μικροφώνων INMP441 (μέσω I2S [20]) εκμηδενίζει τον θόρυβο γραμμής

1.5 Στόχοι Εργασίας

Οι στόχοι της παρούσας πτυχιακής εργασίας συνοψίζονται στα εξής:

1. **Σχεδιασμός και Υλοποίηση Υλικού (Hardware):** Σωστή διαστασιολόγηση της συστοιχίας μικροφώνων για αποφυγή Spatial Aliasing. Σχεδιασμός τροφοδοσίας με μπαταρία 18650 και Boost Converter.
2. **Ανάπτυξη Firmware Πραγματικού Χρόνου (RTOS):** Προγραμματισμός ESP32-S3 μέσω PlatformIO για ταυτόχρονη λήψη I2S DMA, επεξεργασία, έλεγχο σερβοκινητήρα και δικτύωση.
3. **Υλοποίηση SSL (Sound Source Localization):** Μαθηματική υλοποίηση GCC-PHAT σε C++, εξαγωγή γωνίας (θ) και μετατροπή της σε κίνηση του σερβοκινητήρα.
4. **Ενσωμάτωση TinyML (Edge AI):** Εκπαίδευση CNN στο Edge Impulse και Deployment στον μικροελεγκτή για Voice Activity Detection. Έξυπνη καταγραφή σε MicroSD.
5. **Ανάπτυξη Εφαρμογής Ελέγχου (Flutter App/Web UI):** Δημιουργία Γραφικής Διεπαφής, δικτύωση μέσω αυτόνομου Access Point, έλεγχος λειτουργιών, διαχείριση εγγραφών (WAV) στην κάρτα SD μέσω HTTP, και ταυτόχρονη λήψη ζωντανού ήχου μέσω μόνιμης σύνδεσης WebSocket (θύρα 81) .

ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΙΑ ΗΧΟΥ ΚΑΙ DSP

Το παρόν κεφάλαιο αποτελεί το θεωρητικό υπόβαθρο της εργασίας, αναλύοντας διεξοδικά τις αρχές που διέπουν τη διάδοση του ήχου, τη μετατροπή του σε ψηφιακή μορφή, και τις πολύπλοκες μαθηματικές τεχνικές της Ψηφιακής Επεξεργασίας Σήματος (Digital Signal Processing - DSP) που καθιστούν εφικτό τον εντοπισμό και την κατευθυντικότητα. Ξεκινώντας από τις θεμελιώδεις φυσικές ιδιότητες των ακουστικών κυμάτων και τη διαδικασία ψηφιοποίησής τους μέσω της δειγματοληψίας και του κβαντισμού, τίθενται τα απαραίτητα θεμέλια για την υπολογιστική διαχείριση της ακουστικής πληροφορίας. Στη συνέχεια, εξετάζονται τα βασικά εργαλεία μετάβασης από το πεδίο του χρόνου στο πεδίο των συχνοτήτων, όπως ο Ταχύς Μετασχηματισμός Fourier (FFT), τα οποία αποτελούν τον κινητήριο μοχλό για τις προηγμένες μεθόδους χωρικού φιλτραρίσματος (Beamforming). Ιδιαίτερη έμφαση δίνεται στις σύγχρονες τεχνικές Εντοπισμού Ηχητικής Πηγής (Sound Source Localization), μελετώντας το θεωρητικό πλαίσιο της Χρονικής Διαφοράς Άφιξης (TDOA) και τη μαθηματική ανθεκτικότητα του αλγορίθμου GCC-PHAT απέναντι στον θόρυβο και την αντήχηση των πραγματικών περιβαλλόντων. Μέσα από αυτή την αναλυτική πορεία, γεφυρώνεται πλήρως το χάσμα μεταξύ της φυσικής ακουστικής και της εφαρμοσμένης πληροφορικής, προετοιμάζοντας το έδαφος για την πρακτική υλοποίηση του συστήματος.

2.1 Βασικές Έννοιες Ήχου

Ο ήχος, από φυσική σκοπιά, είναι ένα διαμήκες μηχανικό κύμα που διαδίδεται μέσα σε ένα ελαστικό μέσο (όπως ο αέρας, το νερό ή τα στερεά) μέσω διαδοχικών συμπυκνώσεων και αραιώσεων των μορίων του μέσου [1]. Η ταχύτητα διάδοσης του ήχου στον αέρα (c) στους 20°C είναι περίπου 343 m/s [10]. Ένα απλό ημιτονοειδές ηχητικό κύμα περιγράφεται μαθηματικά από την εξίσωση:

$$x(t) = A \cdot \sin(2\pi ft + \varphi) \quad [2.1],$$

όπου:

- **Συχνότητα (f - Frequency):** Εκφράζει τον αριθμό των πλήρων κύκλων (ταλαντώσεων) που εκτελεί το κύμα στη μονάδα του χρόνου. Μετριέται σε Hertz (Hz). Το ανθρώπινο αυτί αντιλαμβάνεται συχνότητες από 20 Hz έως 20.000 Hz (20 kHz) [10]. Οι χαμηλές συχνότητες αντιστοιχούν σε μπάσους ήχους, ενώ οι υψηλές σε πρίμους.
- **Πλάτος (A - Amplitude):** Αντιπροσωπεύει τη μέγιστη απομάκρυνση των μορίων από τη θέση ισορροπίας τους, και σχετίζεται με την αντιληπτή ένταση (ήχο) [1]. Μετριέται σε Pascals (Pa) για την ηχητική πίεση, αλλά πρακτικά στην ακουστική χρησιμοποιείται η λογαριθμική κλίμακα των Decibels (dB) [14].
- **Φάση (φ - Phase):** Εκφράζει τη χρονική μετατόπιση του κύματος ως προς ένα σημείο αναφοράς (0) [1]. Η φάση μετριέται σε μοίρες ή ακτίνια (radians). Όπως θα δούμε παρακάτω, η διαφορά φάσης μεταξύ των μικροφώνων είναι το «κλειδί» για τον αλγόριθμο GCC-PHAT [2].

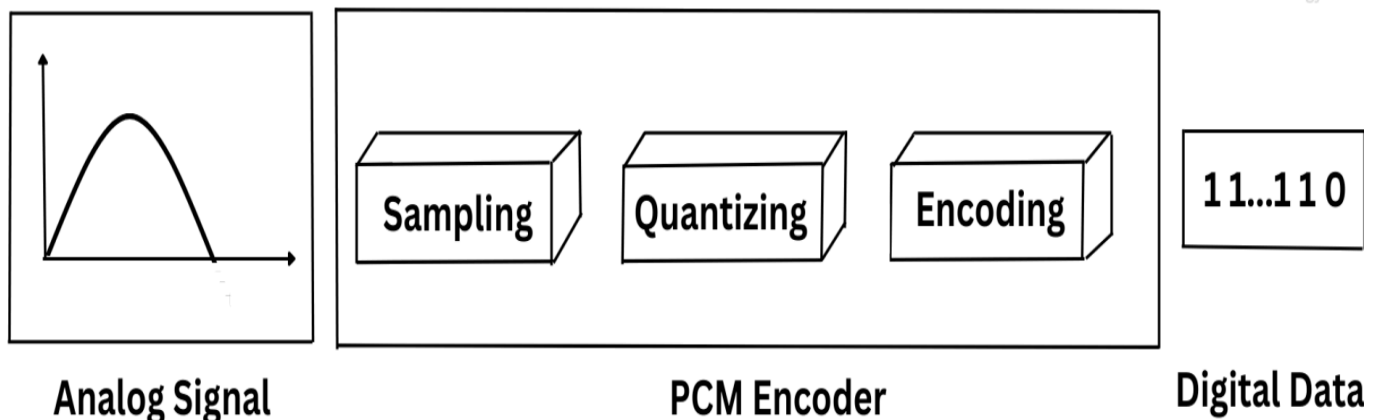
2.2 Ψηφιοποίηση Ήχου

Για να επεξεργαστεί ένας μικροελεγκτής όπως ο ESP32-S3 τον ήχο, το συνεχές αναλογικό σήμα πρέπει να μετατραπεί σε διακριτές ψηφιακές τιμές. Αυτή η διαδικασία γίνεται μέσω των Μετατροπέων Αναλογικού σε Ψηφιακό (Analog-to-Digital Converters - ADCs) και βασίζεται σε δύο θεμελιώδεις διαδικασίες: τη Δειγματοληψία (Sampling) και τον Κβαντισμό (Quantization).

2.3 Sampling (Δειγματοληψία) και Θεώρημα Nyquist

Η δειγματοληψία είναι η διαδικασία λήψης «στιγμιότυπων» της τιμής του αναλογικού σήματος σε τακτά, διακριτά χρονικά διαστήματα (T_s) [1, 12]. Ο αριθμός των δειγμάτων ανά δευτερόλεπτο ονομάζεται Συχνότητα Δειγματοληψίας ($f_s = 1/T_s$). Για να μπορεί το ψηφιακό σήμα να ανακατασκευαστεί πλήρως στην αρχική του αναλογική μορφή χωρίς παραμορφώσεις, πρέπει να ικανοποιείται το **Θεώρημα Δειγματοληψίας των Nyquist-Shannon** [1, 12]. Το θεώρημα ορίζει ότι η συχνότητα δειγματοληψίας πρέπει να είναι τουλάχιστον διπλάσια από τη μέγιστη συχνότητα (f_{max}) που περιέχεται στο σήμα: $f_s \geq 2 \cdot f_{max}$. Αν δεν τηρηθεί αυτός ο κανόνας, εμφανίζεται το φαινόμενο της **Αναδίπλωσης (Aliasing)** [12], όπου οι υψηλές συχνότητες καταγράφονται λανθασμένα ως χαμηλές, καταστρέφοντας το σήμα. Δεδομένου ότι η ανθρώπινη ακοή φτάνει τα 20 kHz, η βιομηχανία έχει καθιερώσει τις συχνότητες των **44.1 kHz** (Audio CD) και **48 kHz** (Επαγγελματικός ήχος/Video) ως πρότυπα, ώστε να καλύπτεται το Nyquist limit με ένα περιθώριο ασφαλείας για τα φίλτρα anti-aliasing.

codegyan.in



Εικόνα 4. digitization of sound, Πηγή: Αναζήτηση google

2.4 PCM Audio

Το Pulse-Code Modulation (PCM) αποτελεί την πιο κοινή μέθοδο ψηφιακής αναπαράστασης αναλογικών σημάτων [1, 12]. Μετά τη δειγματοληψία, η τιμή του πλάτους κάθε δείγματος κβαντίζεται, δηλαδή στρογγυλοποιείται στην κοντινότερη διαθέσιμη διακριτή στάθμη [12]. Ο αριθμός των διαθέσιμων σταθμών εξαρτάται από το **Βάθος Bit (Bit Depth)** [12]. Στον ήχο 16-bit, υπάρχουν $2^{16} = 65.536$ διακριτές στάθμες, ενώ στον ήχο 24-bit (που υποστηρίζεται από τα INMP441 MEMS μικρόφωνα μας), υπάρχουν $2^{24} = 16.777.216$ στάθμες. Μεγαλύτερο Bit Depth σημαίνει μικρότερο Σφάλμα Κβαντισμού (Quantization Error) και άρα ευρύτερη Δυναμική Περιοχή (Dynamic Range) [1, 12]., εξασφαλίζοντας υψηλότερη ποιότητα (High-Resolution Audio).

2.5 FFT (Fast Fourier Transform)

Ο Μετασχηματισμός Fourier είναι το πιο ισχυρό εργαλείο της Ψηφιακής Επεξεργασίας Σήματος [1]. Μας επιτρέπει να μεταβούμε από το πεδίο του χρόνου (Time Domain - πώς αλλάζει το πλάτος σε σχέση με το χρόνο) στο πεδίο των συχνοτήτων (Frequency Domain - ποιες συχνότητες απαρτίζουν το σήμα) [1, 12].

Επειδή τα ψηφιακά συστήματα διαχειρίζονται διακριτά δεδομένα, χρησιμοποιείται ο Διακριτός Μετασχηματισμός Fourier (DFT) [1]. Ωστόσο, ο απευθείας υπολογισμός του DFT απαιτεί $O(N^2)$ πράξεις, κάτι απαγορευτικό για επεξεργασία σε πραγματικό χρόνο. Ο Ταχύς Μετασχηματισμός Fourier (**FFT**) είναι ένας βελτιστοποιημένος αλγόριθμος που μειώνει την πολυπλοκότητα σε $O(N \log N)$ [1, 12].

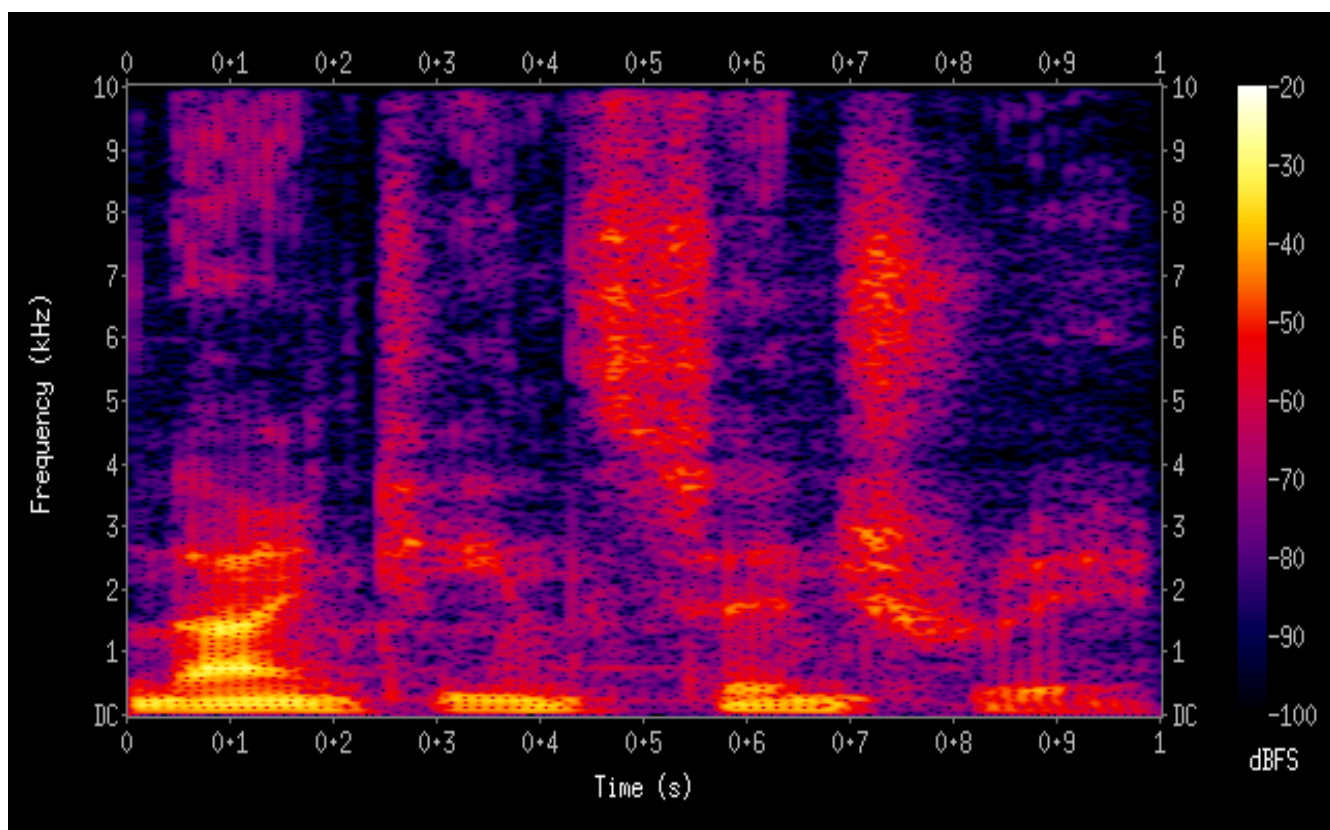
Η μαθηματική έκφραση του DFT δίνεται από τη σχέση:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j(2\pi kn/N)} \quad [2.5]$$

Όπου $x(n)$ είναι το διακριτό σήμα στο χρόνο, $X(k)$ το σήμα στις συχνότητες, και N το πλήθος των δειγμάτων (συνήθως δύναμη του 2, π.χ., 512, 1024, 2048 δείγματα).

2.6 Spectrograms (Φασματογράμματα)

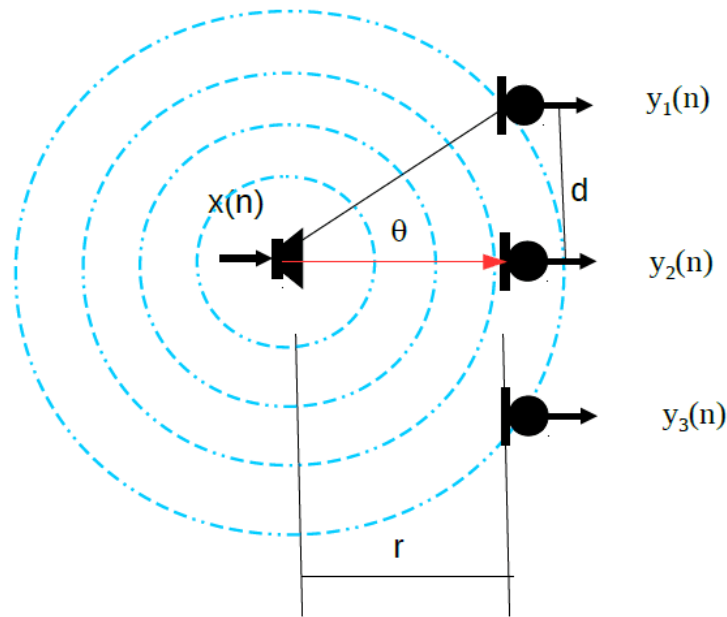
Ο απλός FFT μας δίνει το φάσμα συχνοτήτων ενός ολόκληρου ηχητικού σήματος, αλλά χάνουμε την πληροφορία του πότε εμφανίστηκε κάθε συχνότητα (π.χ., οι λέξεις σε μια πρόταση εναλλάσσονται). Για την ανάλυση χρονικά μεταβαλλόμενων σημάτων χρησιμοποιείται ο Μετασχηματισμός Fourier Βραχέος Χρόνου (Short-Time Fourier Transform - STFT) [1, 10]. Το σήμα τεμαχίζεται σε μικρά χρονικά "παράθυρα" (π.χ., 20-30 ms), συνήθως επικαλυπτόμενα (Overlapping), και σε κάθε παράθυρο εφαρμόζεται FFT [10]. Για να αποφευχθούν οι απότομες ασυνέχειες στα άκρα των παραθύρων, τα δείγματα πολλαπλασιάζονται με ειδικές Συναρτήσεις Παραθύρου (Windowing Functions, όπως Hamming ή Hanning) [1]. Το αποτέλεσμα οπτικοποιείται μέσω του **Φασματογράμματος (Spectrogram)** [10], το οποίο αποτελεί την κύρια μορφή δεδομένων (Input features) που τροφοδοτούν τα Συνελικτικά Νευρωνικά Δίκτυα (CNN) στο TinyML [3, 16] για την αναγνώριση ήχου.



Εικόνα 5. Spectrograms, Πηγή: Wikipedia

2.6 Beamforming (Μορφοποίηση Δέσμης)

Το Beamforming είναι μια τεχνική χωρικού φιλτραρίσματος που χρησιμοποιείται σε συστοιχίες αισθητήρων (Microphone Arrays) για την ενίσχυση του σήματος που προέρχεται από μια συγκεκριμένη κατεύθυνση, ενώ ταυτόχρονα καταστέλλεται ο θόρυβος από άλλες κατευθύνσεις [8, 11, 15]. Βασίζεται στην αρχή της εποικοδομητικής και καταστροφικής συμβολής (constructive and destructive interference).



Εικόνα 6: Μορφοποίηση Δέσμης, Πηγή: created in google flow

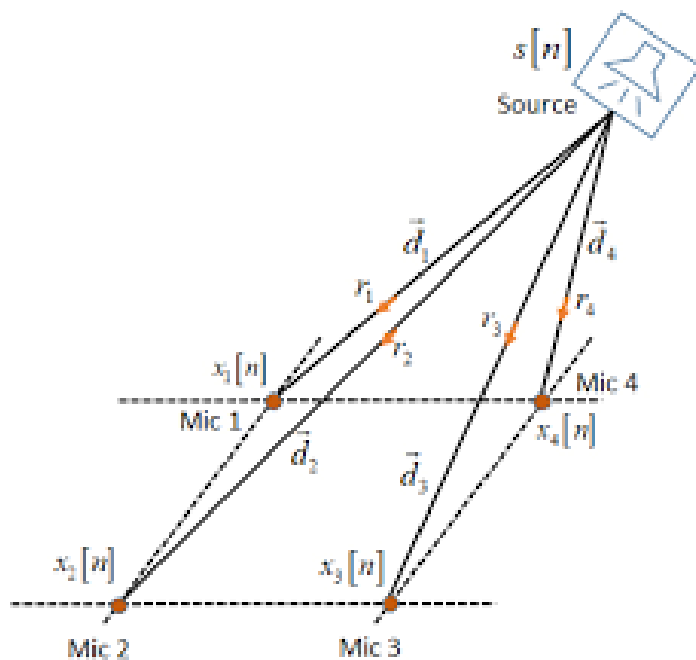
- Delay and Sum Beamforming:** Είναι η πιο απλή και θεμελιώδης μορφή [8]. Επειδή ο ήχος ταξιδεύει με πεπερασμένη ταχύτητα, ένα ακουστικό κύμα που έρχεται υπό γωνία θα φτάσει στα μικρόφωνα της συστοιχίας σε διαφορετικές χρονικές στιγμές [15]. Ο αλγόριθμος εφαρμόζει τεχνητές χρονικές καθυστερήσεις (delays) στο ψηφιακό σήμα κάθε μικροφώνου, ώστε τα σήματα της επιθυμητής πηγής να ευθυγραμμιστούν πλήρως στο χρόνο [8]. Στη συνέχεια, τα ευθυγραμμισμένα σήματα προστίθενται (sum) [8, 15]. Το σήμα-στόχος ενισχύεται, ενώ οι θόρυβοι από άλλες κατευθύνσεις, μη όντας σε φάση, εξασθενούν.
- Adaptive Beamforming:** Σε αντίθεση με το Delay-and-Sum που έχει σταθερά βάρη, το προσαρμοστικό beamforming (π.χ. αλγόριθμοι MVDR) μεταβάλλει δυναμικά τα βάρη των μικροφώνων σε πραγματικό χρόνο, ανάλογα με τα χαρακτηριστικά του θορύβου του περιβάλλοντος [11], τοποθετώντας χωρικά "μηδενικά" (nulls) στις κατευθύνσεις ισχυρών πηγών παρεμβολής.

2.7 Sound Source Localization (SSL)

Ο Εντοπισμός Ηχητικής Πηγής (SSL) είναι η διαδικασία εξαγωγής των χωρικών συντεταγμένων μιας πηγής ήχου με τη χρήση συστοιχίας μικροφώνων [7, 8]. Στην πρακτική εφαρμογή μας, μας ενδιαφέρει η γωνία άφιξης του ήχου (Angle of Arrival - AoA) ή αλλιώς η Διεύθυνση Άφιξης (Direction of Arrival - DoA) στο οριζόντιο επίπεδο (Αζιμούθιο - 1D Localization), ώστε να μπορέσει ο σερβοκινητήρας να στρέψει το σύστημα προς αυτήν [18].

2.8 TDOA (Time Difference Of Arrival)

Η μέθοδος TDOA είναι ο πυρήνας του συστήματός μας. Βασίζεται στον υπολογισμό της χρονικής διαφοράς (Δt) με την οποία το ίδιο ακουστικό κύμα φτάνει σε δύο χωρικά διαχωρισμένα μικρόφωνα [7]. Αν d είναι η γνωστή απόσταση μεταξύ δύο μικροφώνων και c η ταχύτητα του ήχου, η γωνία άφιξης θ μπορεί να υπολογιστεί με βασική τριγωνομετρία [8]: $\theta = \arcsin((c \cdot \Delta t) / d)$. Για να υπολογίσουμε σωστά το θ , πρέπει το Δt να είναι ακριβές. Ο υπολογισμός του Δt είναι μια μεγάλη πρόκληση σε πραγματικά συστήματα λόγω του θορύβου και, κυρίως, της αντήχησης [15]. Εκεί επεμβαίνει ο αλγόριθμος GCC-PHAT [2]



Εικόνα 7. Time difference of arrival, Πηγή: created in google flow

2.9 GCC-PHAT (Generalized Cross Correlation with Phase Transform)

Η κλασική μέθοδος για την εύρεση της χρονικής καθυστέρησης Δt μεταξύ δύο σημάτων $x_1(t)$ και $x_2(t)$ είναι η Ετεροσυσχέτιση (Cross-Correlation). Η χρονική στιγμή όπου η συνάρτηση συσχέτισης εμφανίζει το μέγιστο (κορυφή/peak) αντιστοιχεί στο Δt .

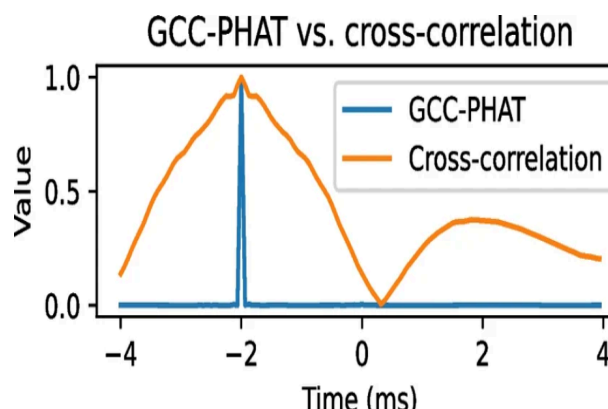
Ωστόσο, σε κλειστούς χώρους (reverberant environments), οι ανακλάσεις του ήχου δημιουργούν πολλαπλές, ψευδείς κορυφές στη συνάρτηση συσχέτισης. Ο αλγόριθμος GCC-PHAT επιλύει αυτό το πρόβλημα εφαρμόζοντας τη συσχέτιση στο πεδίο των συχνοτήτων (μέσω FFT) και διαιρώντας με το πλάτος (Magnitude), αφήνοντας μόνο την πληροφορία της **Φάσης** (Phase).

Η μαθηματική διαδικασία έχει ως εξής:

1. Υπολογισμός του FFT των σημάτων των δύο μικροφώνων: $X_1(f)$ και $X_2(f)$.
2. Υπολογισμός του Cross-Power Spectrum: $G(f) = X_1(f) \cdot X_2^*(f)$, όπου X_2^* είναι ο συζυγής μιγαδικός.
3. **Εφαρμογή του Phase Transform (PHAT)**: Το φάσμα διαιρείται με το μέτρο του, κανονικοποιώντας όλες τις συχνότητες ώστε να έχουν πλάτος 1:
4. Εφαρμογή Αντίστροφου FFT (Inverse FFT) στο RGCC-PHAT(f) για τη μετάβαση πίσω στο πεδίο του χρόνου, λαμβάνοντας τη συνάρτηση GCC-PHAT στο χρόνο $R(t)$.
5. Το Δt προσδιορίζεται βρίσκοντας το όρισμα (index/time) που μεγιστοποιεί τη συνάρτηση

$$R(t) : \Delta t = \operatorname{argmax} R(t) \quad [2.9]$$

Η μέθοδος GCC-PHAT αποδίδει μια εξαιρετικά "οξεία" (sharp) κορυφή στη θέση της πραγματικής πηγής ήχου, καταστέλλοντας τις ψευδείς κορυφές που προκαλούνται από την ηχώ. Αφού υπολογιστεί το Δt μέσω του GCC-PHAT, το εισάγουμε στην εξίσωση του TDOA για να βρούμε τη γωνία θ και, εν συνεχεία, να δώσουμε τη σχετική εντολή (PWM) στον σερβοκινητήρα (MG996R) του ESP32-S3



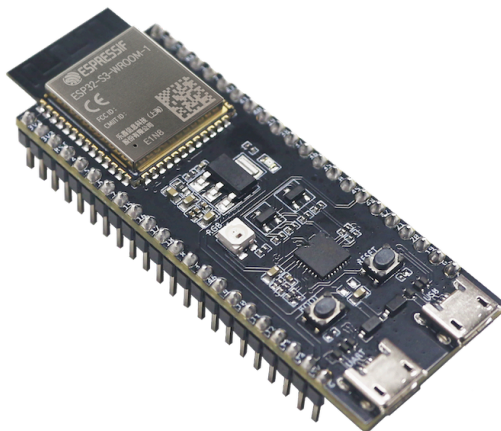
Εικόνα 8. Sound Source Localization (SSL), Πηγή: Αναζήτηση google

ΚΕΦΑΛΑΙΟ 3: ESP32-S3 ΚΑΙ HARDWARE

Η υλοποίηση ενός συστήματος έξυπνου κατευθυντικού μικροφώνου απαιτεί τη στενή και άρρηκτη συνεργασία εξειδικευμένου υλικού (hardware) με ισχυρό, βελτιστοποιημένο λογισμικό. Η μετάβαση από τη θεωρητική μελέτη των αλγορίθμων ψηφιακής επεξεργασίας σήματος στην πρακτική, on-device εφαρμογή τους σε πραγματικό χρόνο, εγείρει σημαντικές μηχανικές προκλήσεις που σχετίζονται με την υπολογιστική ισχύ, τον διαθέσιμο χώρο μνήμης και τον αυστηρό χρονισμό των δεδομένων. Για τον λόγο αυτό, η επιλογή των κατάλληλων ηλεκτρονικών εξαρτημάτων δεν έχει απλώς υποστηρικτικό ρόλο, αλλά αποτελεί τον καθοριστικό παράγοντα που υπαγορεύει τις τελικές επιδόσεις της κατασκευής. Στο παρόν κεφάλαιο αναλύεται εκτενώς η αρχιτεκτονική του μικροελεγκτή ESP32-S3, ο οποίος επιλέχθηκε ως ο κεντρικός πυρήνας ελέγχου λόγω της ικανότητάς του να διαχειρίζεται ταχύτατα πολύπλοκες διεργασίες Τεχνητής Νοημοσύνης στο άκρο του δικτύου (Edge AI). Παράλληλα, εξετάζεται σε βάθος η αρχή λειτουργίας των ψηφιακών μικροφώνων τεχνολογίας MEMS και του αναλογικού αισθητήρα που απαρτίζουν την ακουστική συστοιχία. Τέλος, περιγράφονται αναλυτικά οι κινηματικοί μηχανισμοί στόχευσης, τα μέσα ασφαλούς αποθήκευσης των δεδομένων και ο σχεδιασμός του κυκλώματος ενεργειακής διαχείρισης, τα οποία συνθέτουν από κοινού μια πλήρως αυτόνομη, φορητή και αξιόπιστη διάταξη.

3.1 ESP32-S3 Architecture

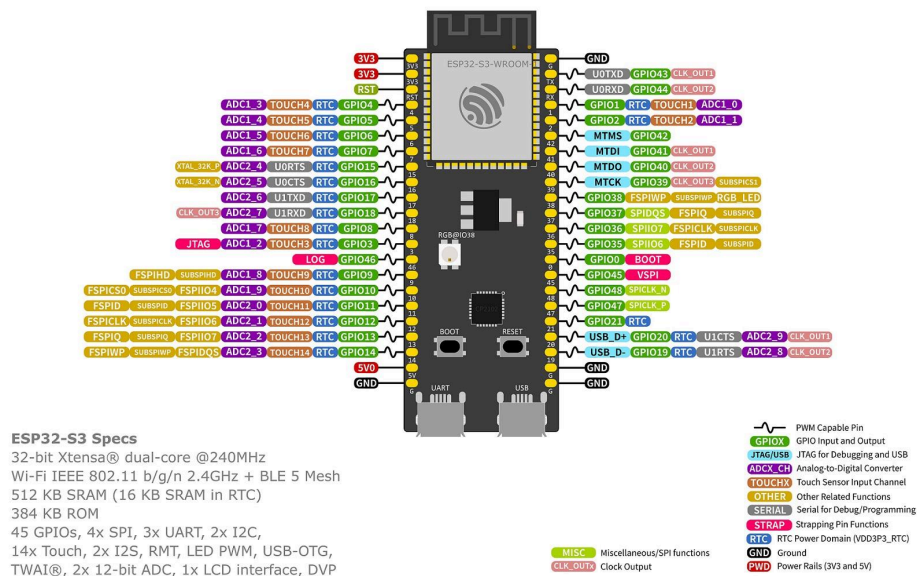
Ο μικροελεγκτής ESP32-S3 της Espressif αποτελεί τον "εγκέφαλο" του συστήματος [4]. Σε αντίθεση με παλαιότερες γενιές μικροελεγκτών, το συγκεκριμένο SoC (System on a Chip) έχει σχεδιαστεί με έμφαση στις εφαρμογές Τεχνητής Νοημοσύνης στο άκρο του δικτύου (Edge AI) και την ψηφιακή επεξεργασία σήματος (DSP) [3, 4].



Εικόνα 9. ESP32-S3-DevKitC-1 Module, Πηγή: docs.espressif.com

- **CPU (Κεντρική Μονάδα Επεξεργασίας):** Διαθέτει διπύρρηνο επεξεργαστή Xtensa® 32-bit LX7, ικανό να λειτουργήσει σε συχνότητες έως και 240 MHz [4]. Το πλέον σημαντικό χαρακτηριστικό του είναι η υποστήριξη διανυσματικών εντολών (Vector Instructions) που επιταχύνουν δραματικά τις μαθηματικές πράξεις πινάκων, οι οποίες είναι θεμελιώδεις για τους υπολογισμούς νευρωνικών δικτύων (TinyML) και τον αλγόριθμο FFT [1, 3]. Η διπύρρηνη αρχιτεκτονική επιτρέπει τον διαχωρισμό των εργασιών: ο ένας πυρήνας (Core 0) μπορεί να αναλάβει την αδιάλειπτη δειγματοληψία του ήχου και την ασύρματη επικοινωνία, ενώ ο άλλος πυρήνας (Core 1) αφιερώνεται αποκλειστικά στους υπολογισμούς TDOA και την ταξινόμηση ήχων [9].
- **RAM (Μνήμη Τυχαίας Προσπέλασης):** Ενσωματώνει 512 KB εσωτερικής SRAM [4]. Αν και επαρκής για βασικές λειτουργίες, η αποθήκευση πολλαπλών δευτερολέπτων ήχου υψηλής ανάλυσης για ανάλυση απαιτεί μεγαλύτερο χώρο.
- **PSRAM (Pseudo-Static RAM):** Για την αντιμετώπιση του περιορισμού της εσωτερικής μνήμης, ο ESP32-S3 συνοδεύεται από 8 MB εξωτερικής PSRAM [4]. Η PSRAM συνδέεται μέσω διαύλου Octal SPI, προσφέροντας υψηλό εύρος ζώνης (bandwidth). Εδώ αποθηκεύονται οι μεγάλοι κυκλικοί καταχωρητές (ring buffers) του ήχου, επιτρέποντας την καταγραφή και ανάλυση ολόκληρων λέξεων ή ηχητικών συμβάντων πριν την εγγραφή τους στην κάρτα MicroSD.
- **WiFi & Bluetooth:** Ενσωματώνει Wi-Fi 4 (802.11 b/g/n) στα 2.4 GHz και Bluetooth 5 (LE) [4]. Στην παρούσα εργασία, το Wi-Fi χρησιμοποιείται σε κατάσταση Access Point (AP), δημιουργώντας ένα τοπικό, ανεξάρτητο δίκτυο, απαλλάσσοντας το σύστημα από την ανάγκη ύπαρξης εξωτερικού router για την επικοινωνία με την εφαρμογή (Flutter App) [6].

ESP32-S3-DevKitC-1



Εικόνα 10. Esp32 Pin Layout, Πηγή:docs.espressif.com

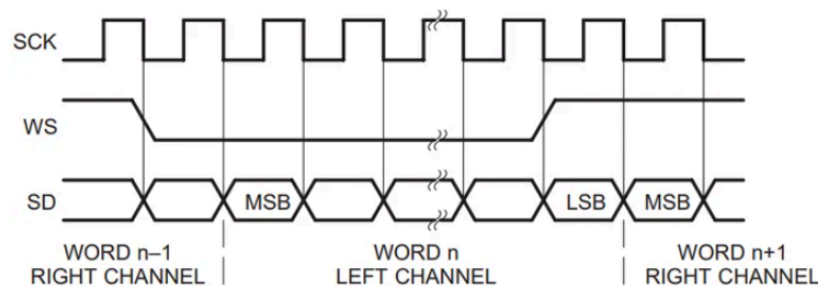
3.2 Πρωτόκολλο I2S (Inter-IC Sound)

Το πρωτόκολλο I2S είναι το απόλυτο βιομηχανικό πρότυπο για τη σειριακή μετάδοση ψηφιακού ήχου μεταξύ ολοκληρωμένων κυκλωμάτων [20]. Στην κατασκευή μας, το I2S δεν είναι απλώς μια διεπαφή, αλλά η "ραχοκοκαλιά" του συστήματος. Η εις βάθος κατανόησή του είναι κρίσιμη, καθώς διαχειρίζεται τη συγχρονισμένη ροή δεδομένων από τέσσερα διαφορετικά μικρόφωνα προς τη μνήμη του μικροελεγκτή, χωρίς την άμεση παρέμβαση της CPU [4, 20].

3.2.1 Αρχιτεκτονική και Σήματα του I2S

Η επικοινωνία I2S βασίζεται σε τρεις κύριες γραμμές σημάτων, οι οποίες εξασφαλίζουν τον απόλυτο συγχρονισμό πομπού και δέκτη:

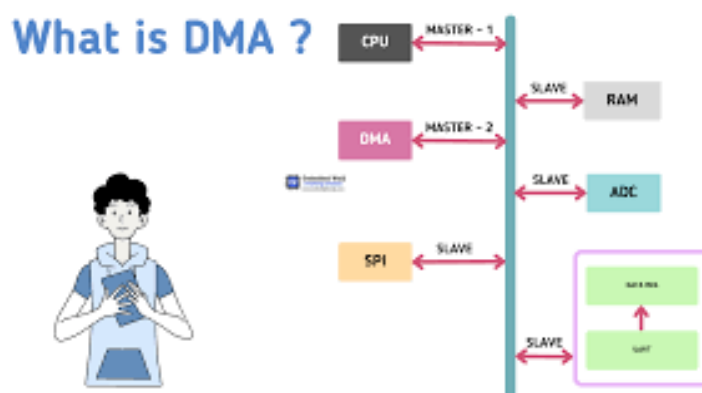
1. **SCK (Serial Clock) / BCLK (Bit Clock):** Το ρολόι χρονισμού των bits. Καθορίζει την ταχύτητα με την οποία μεταδίδεται κάθε bit δεδομένων. Η συχνότητα του BCLK υπολογίζεται από τον τύπο: $BCLK = \text{Συχνότητα Δειγματοληψίας (fs)} \times \text{Βάθος Bit} \times \text{Αριθμός Καναλιών}$. Για παράδειγμα, για στερεοφωνικό ήχο στα 48 kHz με 32-bit (έστω κι αν τα δεδομένα είναι 24-bit, το I2S συχνά χρησιμοποιεί 32-bit frames), το BCLK είναι $48000 \times 32 \times 2 = 3.072 \text{ MHz}$.
2. **WS (Word Select) / LRCLK (Left-Right Clock):** Το ρολόι επιλογής λέξης/καναλιού. Η συχνότητα του WS είναι ακριβώς ίση με τη συχνότητα δειγματοληψίας (fs). Όταν το σήμα WS είναι σε λογικό '0' (Low), μεταδίδονται τα δεδομένα του αριστερού καναλιού. Όταν το WS είναι σε λογικό '1' (High), μεταδίδονται τα δεδομένα του δεξιού καναλιού. Αυτός ο διαχωρισμός είναι ζωτικής σημασίας για τη συστοιχία μας, καθώς μας επιτρέπει να συνδέσουμε δύο μικρόφωνα INMP441 στην ίδια γραμμή δεδομένων, ορίζοντας το ένα ως L και το άλλο ως R.
3. **SD (Serial Data) / DIN/DOUT:** Η γραμμή μεταφοράς των πραγματικών ψηφιακών δεδομένων (payload). Τα δεδομένα μεταδίδονται σειριακά, αρχίζοντας πάντα από το πιο σημαντικό bit (MSB - Most Significant Bit).



Εικόνα 11. protocol i2s explained, Πηγή:Αναζήτηση google

3.2.2 Η σημασία της καθυστέρησης του ενός Bit (1-Bit Delay)

Μια θεμελιώδης διαφορά του I2S από παρόμοια συγχρονισμένα πρωτόκολλα είναι η καθυστέρηση ενός κύκλου ρολογιού (1-bit delay) μεταξύ της αλλαγής της κατάστασης του σήματος WS και της έναρξης μετάδοσης του MSB των δεδομένων. Το MSB δεν μεταδίδεται στην ίδια ακμή του ρολογιού που αλλάζει το WS, αλλά στην επόμενη. Αυτή η μηχανική προστέθηκε ιστορικά για να δοθεί χρόνος στον δέκτη να ετοιμάσει τους καταχωρητές του μόλις αντιληφθεί την αλλαγή καναλιού. Ο ESP32-S3 ενσωματώνει υλικό που διαχειρίζεται αυτόματα αυτή την καθυστέρηση.



Εικόνα 12. dma direct memory access diagram, Πηγή:Αναζήτηση google

3.2.3 Άμεση Πρόσβαση στη Μνήμη (DMA - Direct Memory Access)

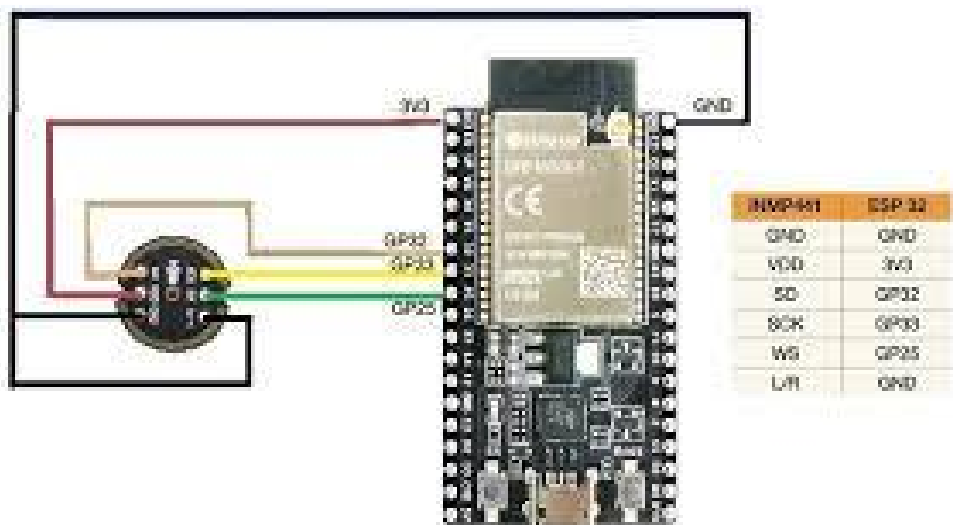
Η διαχείριση της συνεχούς ροής χιλιάδων δειγμάτων ήχου ανά δευτερόλεπτο από τέσσερα μικρόφωνα θα απασχολούσε πλήρως την CPU, εμποδίζοντας την εκτέλεση αλγορίθμων TinyML ή ασύρματης επικοινωνίας. Ο ελεγκτής I2S του ESP32-S3 λύνει αυτό το πρόβλημα μέσω του συστήματος DMA (Direct Memory Access).

Το DMA είναι ένας εξειδικευμένος ελεγκτής υλικού που μεταφέρει δεδομένα απευθείας από το περιφερειακό I2S στην κεντρική μνήμη RAM (ή PSRAM), παρακάμπτοντας εντελώς την CPU. Η διαδικασία οργανώνεται μέσω "Περιγραφών DMA" (DMA Descriptors), οι οποίοι σχηματίζουν μια συνδεδεμένη λίστα. Η CPU ρυθμίζει τον αριθμό και το μέγεθος των buffers (π.χ. 8 buffers των 1024 bytes). Το DMA γεμίζει τον πρώτο buffer. Μόλις γεμίσει, προκαλεί μια διακοπή υλικού (Interrupt) για να ειδοποιήσει την CPU ότι τα δεδομένα είναι έτοιμα προς επεξεργασία, και αμέσως συνεχίζει την εγγραφή στον δεύτερο buffer (χωρίς να χαθεί ούτε ένα δείγμα ήχου). Αυτή η τεχνική κυκλικής αποθήκευσης (Ping-Pong buffering) εξασφαλίζει τη σταθερότητα που απαιτείται για τους ευαίσθητους στο χρόνο αλγορίθμους TDOA [1, 4].

3.3 MEMS Microphones (INMP441)

Η καρδιά της ακουστικής αντίληψης του συστήματος είναι το INMP441, ένα υψηλής απόδοσης ψηφιακό μικρόφωνο τύπου MEMS (Micro Electro Mechanical Systems) [8]. Τα MEMS μικρόφωνα κατασκευάζονται με τεχνικές λιθογραφίας πυριτίου, παρόμοιες με αυτές των μικροεπεξεργαστών, προσφέροντας εξαιρετική μικρογραφία, σταθερότητα και χαμηλό κόστος [15].

- **Λειτουργία και Χαρακτηριστικά:** Το INMP441 είναι ένα πανκατευθυντικό μικρόφωνο κάτω θύρας (bottom-port). Στο εσωτερικό του διαθέτει ένα διάφραγμα σιλικόνης που πάλλεται ανάλογα με την ηχητική πίεση, μεταβάλλοντας τη χωρητικότητα. Το ενσωματωμένο κύκλωμα ASIC (Application-Specific Integrated Circuit) μετατρέπει την αναλογική μεταβολή σε ψηφιακό σήμα μέσω ενός μετατροπέα Sigma-Delta και εξάγει τα δεδομένα απευθείας σε πρωτόκολλο I2S. Προσφέρει λόγο σήματος προς θόρυβο (SNR) στα 61 dBA και επίπεδη απόκριση συχνότητας από 60 Hz έως 15 kHz.
- **Σύνδεση και L/R Channel Selection:** Το μικρόφωνο διαθέτει μια ακίδα "L/R". Συνδέοντας αυτήν την ακίδα στη γείωση (GND), το μικρόφωνο εξάγει τα δεδομένα του στο αριστερό κανάλι (κατά τη διάρκεια που το WS είναι Low). Συνδέοντάς την στην τροφοδοσία (VDD), εκπέμπει στο δεξί κανάλι. Αυτό επιτρέπει τη συνδεσμολογία σε ζεύγη

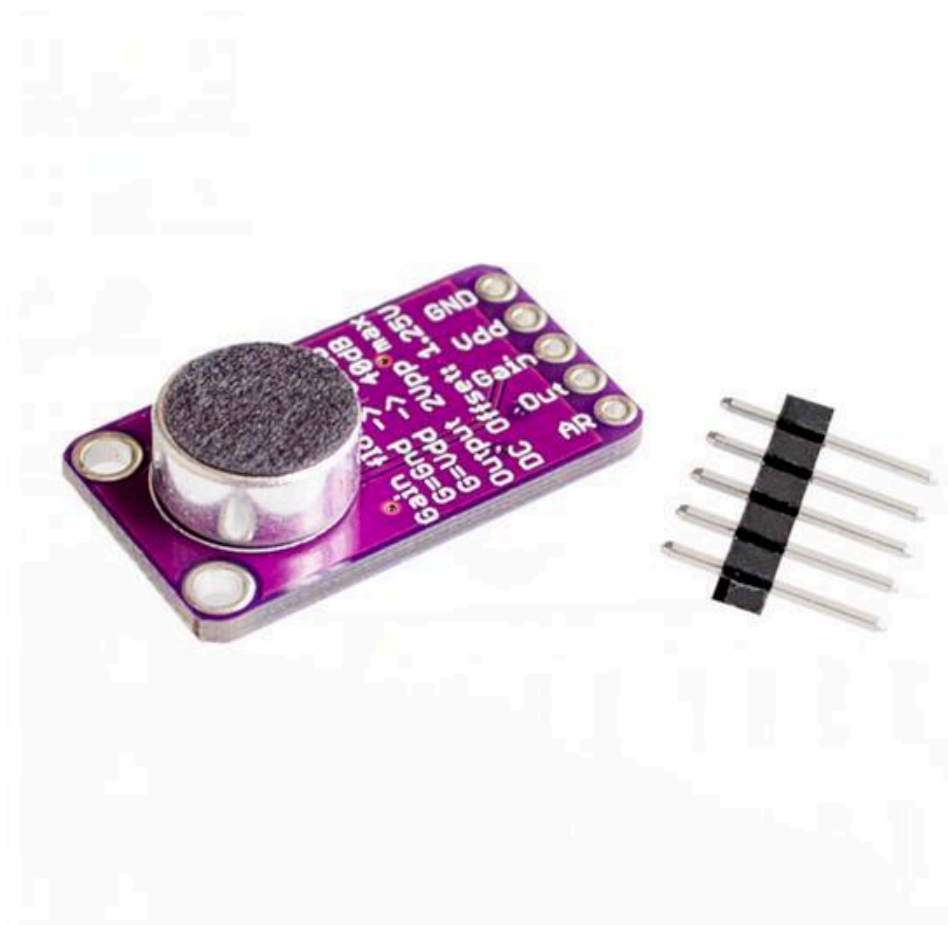


Εικόνα 13. inmp441 mems microphones, Πηγή:Αναζήτηση google

3.4 Αναλογικό Μικρόφωνο MAX9814

Παράλληλα με την ψηφιακή συστοιχία, το σύστημα ενσωματώνει ένα αναλογικό μικρόφωνο με τον ενισχυτή MAX9814 [14]. Το μικρόφωνο MAX9814 αποτελεί το κεντρικό μικρόφωνο της διάταξης, το οποίο λειτουργεί ως shotgun μικρόφωνο και στρέφεται μηχανικά προς την πηγή του ήχου για τη βέλτιστη καταγραφή και απομόνωση της ομιλίας [14, 15].

- **AGC (Automatic Gain Control):** Το σημαντικότερο πλεονέκτημα του MAX9814 είναι ο μηχανισμός Αυτόματου Ελέγχου Κέρδους [13]. Ο ενισχυτής αναλύει το πλάτος του εισερχόμενου ήχου και προσαρμόζει δυναμικά την ενίσχυση. Αν ο ήχος είναι ψίθυρος, το κέρδος αυξάνεται (έως 60 dB). Αν υπάρξει ένας εκκωφαντικός θόρυβος, το κέρδος μειώνεται άμεσα (attack time), αποτρέποντας την παραμόρφωση (clipping) [12]. Το σήμα του οδηγείται στον ADC του ESP32-S3 ως το κύριο, κατευθυντικό ηχητικό σήμα της συσκευής.



Εικόνα 14. Αναλογικό Μικρόφωνο MAX9814, Πηγή: www.hellasdigital.gr

3.5 Σερβοκινητήρας MG996R

Η φυσική περιστροφή του μικροφώνου υλοποιείται μέσω του σερβοκινητήρα MG996R, επιλεγμένου για την υψηλή του ροπή και την αντοχή του.

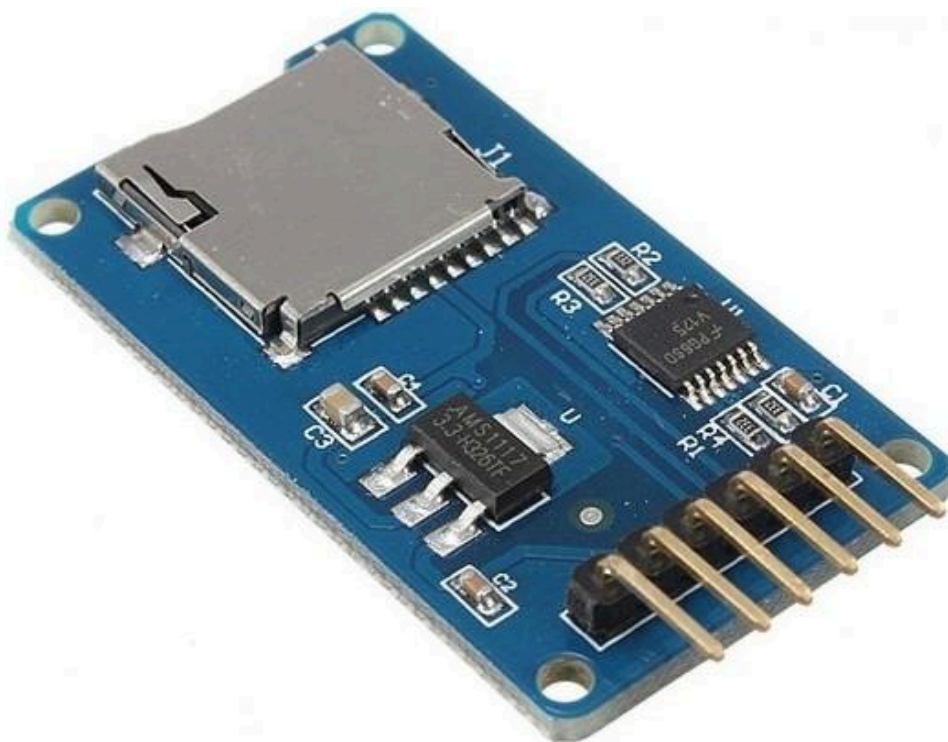
- **Χαρακτηριστικά:** Διαθέτει μεταλλικά γρανάζια (Metal Gear), παρέχοντας εξαιρετική αντοχή στη φθορά. Σε τάση λειτουργίας 4.8V – 6.0V, η ροπή αναμονής (stall torque) ξεπερνά τα 9.4 kg/cm, υπερεπαρκής για την περιστροφή της συστοιχίας.
- **Έλεγχος μέσω PWM και Χρονικής Βαθμονόμησης:** Ο κινητήρας ελέγχεται με σήματα διαμόρφωσης εύρους παλμού (PWM) με περίοδο 20 ms (50 Hz). Κατά την πειραματική υλοποίηση διαπιστώθηκε ότι ο διαθέσιμος ενεργοποιητής MG996R λειτουργεί ως κινητήρας συνεχούς περιστροφής (continuous rotation) και όχι απόλυτης θέσης (positional). Συνεπώς, το πλάτος του παλμού δεν ορίζει την τελική γωνία, αλλά την ταχύτητα και τη φορά περιστροφής. Για να επιτευχθεί ακριβής στόχευση, ο έλεγχος του παλμού υλοποιήθηκε μέσω λογισμικής παραγωγής (bit-banging) στο GPIO 5, με χρήση αποκλειστικού Task στο Core 1 του ESP32-S3. Η γωνία προσεγγίζεται λογισμικά με σταθερή ταχύτητα, μέσω μιας διαδικασίας χρονικής βαθμονόμησης, καθιστώντας την κατασκευή λειτουργική, με πρόβλεψη ωστόσο για μελλοντική χρήση servo απόλυτης θέσης (180° ή 360° positional).



Εικόνα 15. Servo MG996R, Πηγή:Αναζήτηση google

3.6 MicroSD (Αποθήκευση Δεδομένων)

Στη λειτουργία καταγραφής (Rec Mode) και στην Έξυπνη Λειτουργία (Smart Mode), το σύστημα διατηρεί αρχείο των ηχητικών συμβάντων αποθηκεύοντάς τα σε κάρτα MicroSD. Η εγγραφή γίνεται σε μορφή ασυμπίεστων αρχείων ήχου (WAV) με συχνότητα δειγματοληψίας 8 kHz, ανάλυση 16-bit και μονοφωνικό (mono) κανάλι. Η διαχείριση της κάρτας υλοποιείται μέσω του διαύλου επικοινωνίας SPI (CS 10, MOSI 11, SCK 12, MISO 13) σε ταχύτητα 4 MHz (με πρόβλεψη fallback σε χαμηλότερες συχνότητες αν απαιτηθεί). Είναι ζωτικής σημασίας να αναφερθεί πως, κατά τον σχεδιασμό, εντοπίστηκε ότι η μονάδα (module) της κάρτας MicroSD που χρησιμοποιήθηκε, φέρει ενσωματωμένο ρυθμιστή τάσης AMS1117. Κατά συνέπεια, η τροφοδοσία του module συνδέθηκε απαραίτητως στα **5V** (και όχι στα 3.3V του μικροελεγκτή), διασφαλίζοντας την επαρκή παροχή ρεύματος κατά τις απότομες αιχμές (spikes) εγγραφής στη μνήμη Flash. Παράλληλα, το σύστημα χρησιμοποιεί το ρολόι του ESP32 (μέσω μεταβλητής Unix timestamp) για τη σωστή χρονοσήμανση και ονοματοδοσία των αρχείων (π.χ. `/sd/rec_NNN.wav`) στο σύστημα αρχείων FAT32.

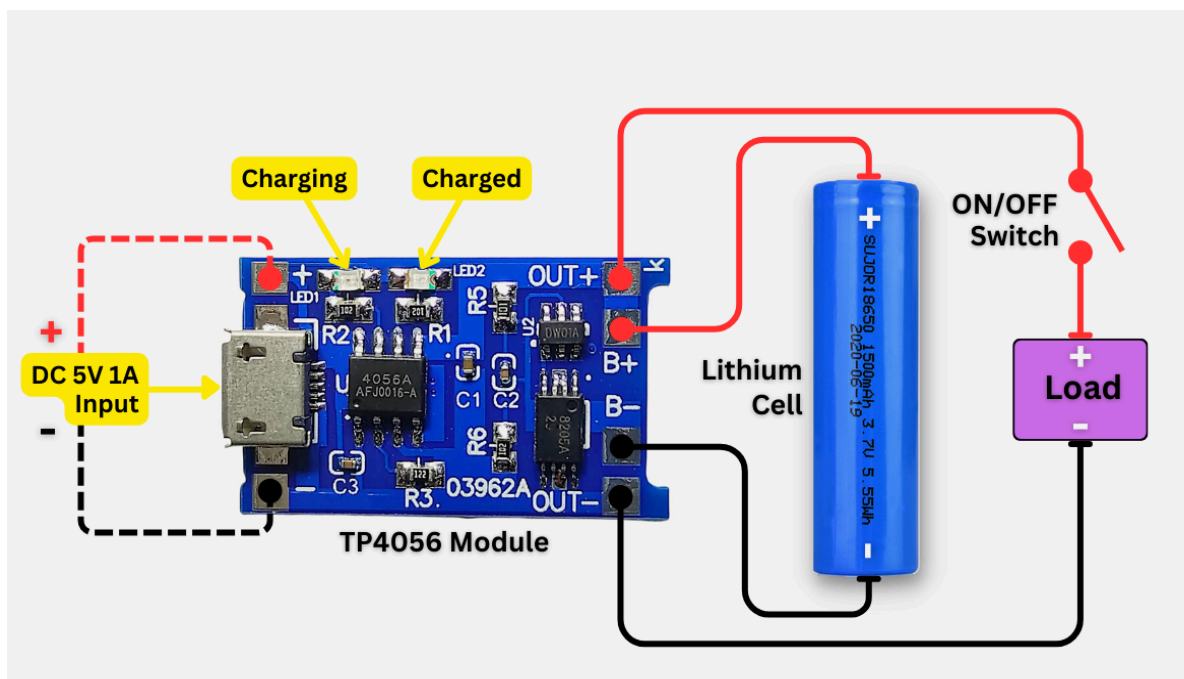


Εικόνα 16. microSD Module, Πηγή: www.grobotronics.com

3.7 Τροφοδοσία και Ενεργειακή Διαχείριση

Ο κατευθυντικός μικροφωνικός σταθμός είναι μια φορητή και αυτόνομη συσκευή. Η σωστή τροφοδοσία των επιμέρους στοιχείων, τα οποία έχουν διαφορετικές απαιτήσεις τάσης και ρεύματος, είναι μια κρίσιμη παράμετρος. Το κύκλωμα τροφοδοσίας σχεδιάστηκε με αυστηρή ιεραρχία προκειμένου να διασφαλιστεί η σταθερότητα του συστήματος:

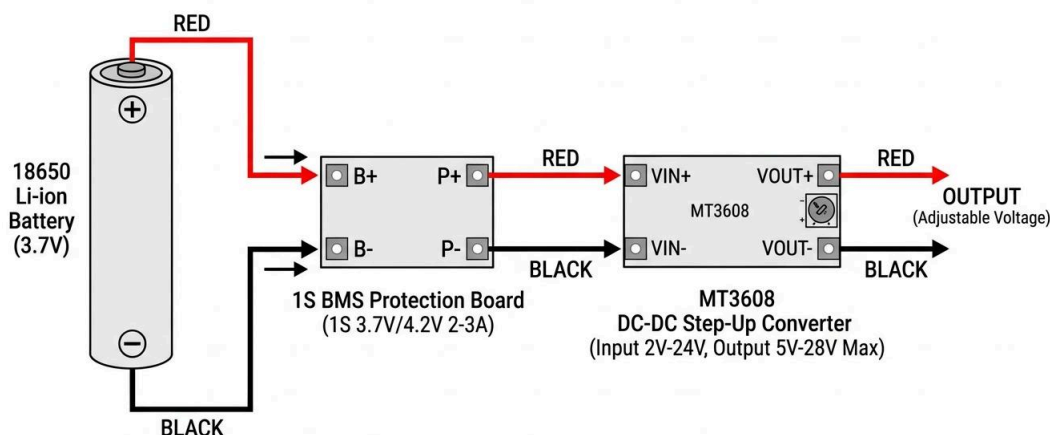
Μπαταρία και Κύκλωμα Φόρτισης/Προστασίας (TP4056): Ως πηγή ενέργειας χρησιμοποιείται ένα στοιχείο ιόντων λιθίου (Li-Ion) τύπου 18650 (ονομαστική τάση 3.7V). Η μπαταρία δεν συνδέεται απευθείας στο κύκλωμα, αλλά πάνω σε μια εξειδικευμένη μονάδα φόρτισης (TP4056) η οποία ενσωματώνει το κύκλωμα προστασίας (BMS). Η διάταξη αυτή είναι απολύτως αναγκαία, καθώς διακόπτει την παροχή προς το σύστημα σε περιπτώσεις υπερφόρτισης, βαθιάς εκφόρτισης και βραχυκυκλώματος, διασφαλίζοντας τη μακροζωία του στοιχείου και την ασφαλή λειτουργία της συσκευής.



Εικόνα 17. TP4056 wiring, Πηγή: Αναζήτηση google

Κύκλωμα Ανύψωσης Τάσης (Step-Up Converter): Επειδή ο σερβοκινητήρας MG996R και η πλακέτα του ESP32-S3 (pin 5V) απαιτούν τάση υψηλότερη από τα 3.7V της μπαταρίας, χρησιμοποιείται ένας μετατροπέας ανύψωσης τάσης. Η έξοδος (OUT) της μονάδας προστασίας της μπαταρίας οδηγείται στην είσοδο του Step-Up, το οποίο έχει ρυθμιστεί με ακρίβεια ώστε να δέχεται τα ~3.7V και να εξάγει σταθερά 5V. Αυτή η σταθεροποιημένη τάση των 5V αναλαμβάνει πλέον την παράλληλη τροφοδοσία ολόκληρου του συστήματος (μικροελεγκτή, αισθητήρων και κινητήρα).

18650 Battery with BMS & MT3608 Boost Converter Wiring Diagram

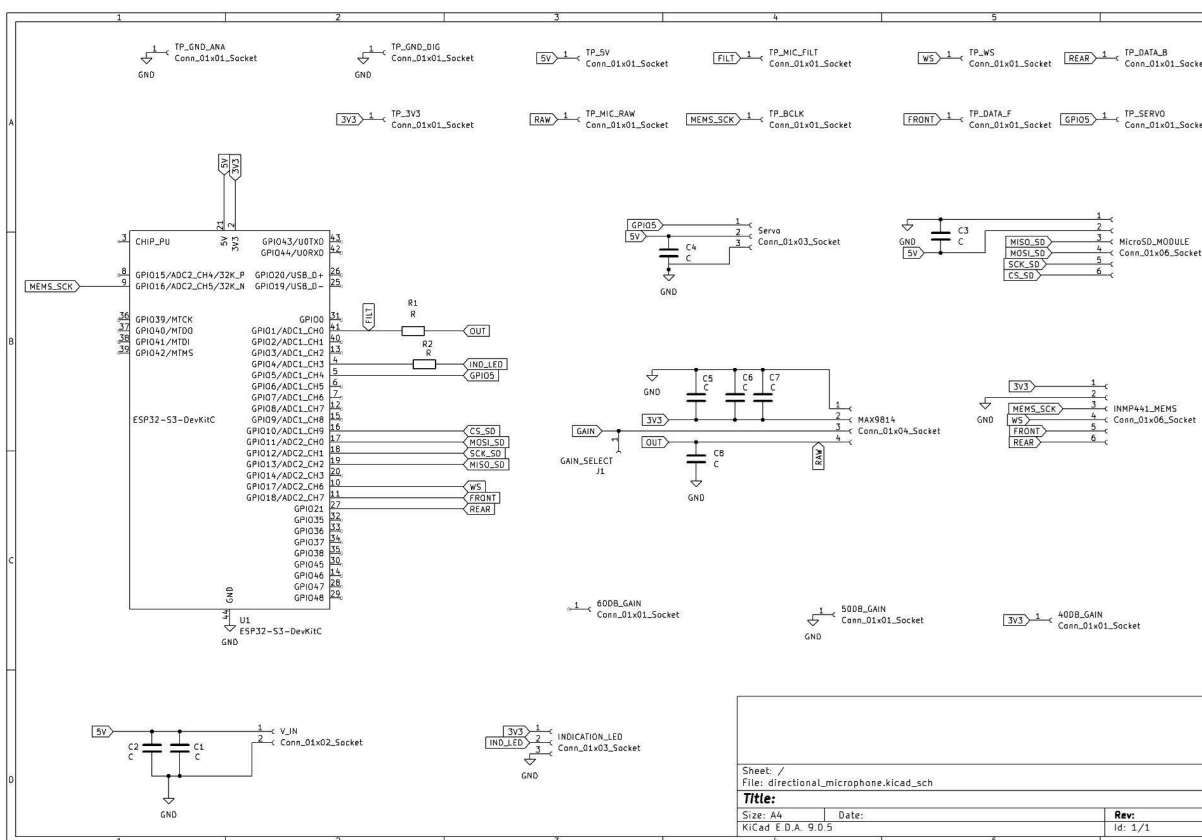


Εικόνα 18. Συνδεσμολογία Παροχής ενέργειας, Πηγή: created in google flow

Απόσβεση Αιχμών Ρεύματος: Ένας διαχωρισμός του δικτύου τροφοδοσίας με κατάλληλους πυκνωτές εξομάλυνσης είναι απαραίτητος, ώστε οι απότομες "αιχμές" ρεύματος (spikes) που απαιτεί ο σερβοκινητήρας κατά την εκκίνηση της κίνησής του να μην προκαλούν πτώση τάσης (brownout) και επανεκκίνηση του μικροελεγκτή. Παρεμβάλλεται ένας ηλεκτρολυτικός πυκνωτής 1000uF στη γραμμή 5V του σερβοκινητήρα για απόσβεση των ρευμάτων εκκίνησης.

ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Το παρόν κεφάλαιο σηματοδοτεί τη μετάβαση από τη θεωρητική ανάλυση και την παρουσίαση των μεμονωμένων εξαρτημάτων, που προηγήθηκαν στο Κεφάλαιο 3, στη σύνθεση και τον ολοκληρωμένο σχεδιασμό του τελικού ηλεκτρονικού συστήματος. Σκοπός του είναι να αναδείξει πώς τα επιμέρους υλικά στοιχεία (όπως ο μικροελεγκτής, οι ψηφιακοί και αναλογικοί αισθητήρες, και τα συστήματα ενεργειακής διαχείρισης) διασυνδέονται αρμονικά για να σχηματίσουν μια πλήρως λειτουργική, σταθερή και αξιόπιστη διάταξη. Στις ενότητες που ακολουθούν, εξετάζεται αναλυτικά η αρχιτεκτονική του συστήματος μέσω του διαγράμματος βαθμίδων (block diagram) και αναλύεται διεξοδικά η στρατηγική τοπολογική διάταξη της συστοιχίας των μικροφώνων, η οποία αποτελεί τον θεμέλιο λίθο για τη χωρική αντίληψη του ήχου. Παράλληλα, τεκμηριώνονται οι αυστηροί μαθηματικοί περιορισμοί που επιβάλλονται για την αποφυγή φαινομένων χωρικής αναδίπλωσης (spatial aliasing) και την επίτευξη της βέλτιστης γωνιακής ανάλυσης. Τέλος, παρουσιάζονται οι εξειδικευμένες τεχνικές που εφαρμόστηκαν κατά τον σχεδιασμό της τυπωμένης πλακέτας (PCB) και της καλωδίωσης, με ιδιαίτερη έμφαση στην ελαχιστοποίηση των ηλεκτρομαγνητικών παρεμβολών (EMI) και στη διασφάλιση της απόλυτης ακεραιότητας των ευαίσθητων ακουστικών σημάτων κατά τη μετάδοσή τους προς την κεντρική μονάδα επεξεργασίας.

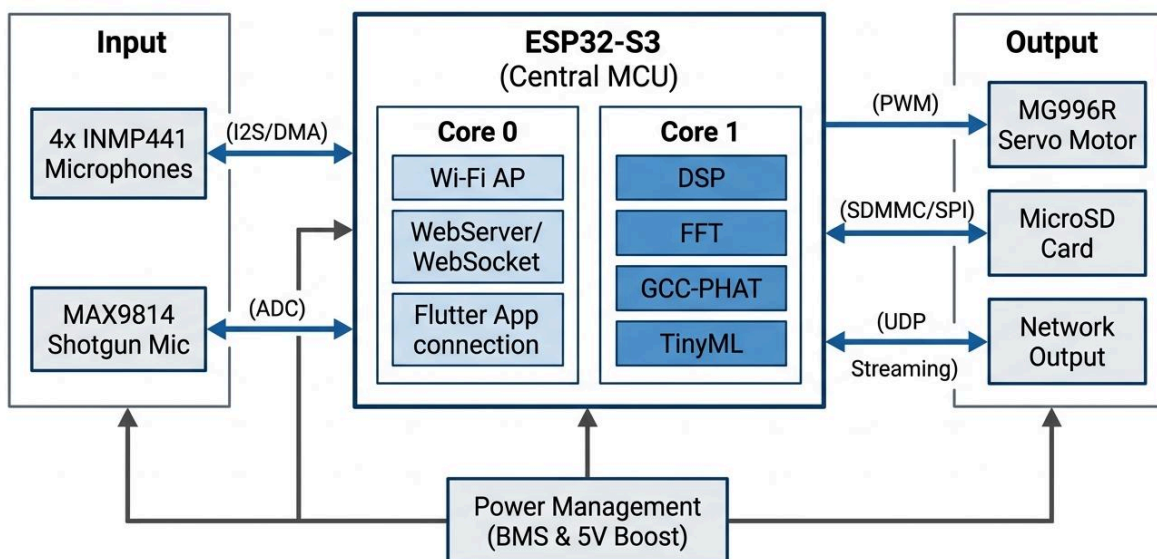


Εικόνα 19. Σχηματικό Ηλεκτρονικού Συστήματος, Πηγή:screenshot kiCad, pc

4.1 Αρχιτεκτονική Συστήματος (Block Diagram)

Η αρχιτεκτονική του συστήματος σχεδιάστηκε με γνώμονα τη σπονδυλωτή ανάπτυξη (modularity) και τον αυστηρό διαχωρισμό των υπολογιστικών πόρων. Το Block Diagram (Διάγραμμα Βαθμίδων) ορίζει την κεντρική μονάδα επεξεργασίας (ESP32-S3) ως τον διαχειριστή όλων των υποσυστημάτων [4]. Από πλευράς **Εισόδου Δεδομένων**, η διάταξη των τεσσάρων ψηφιακών μικροφώνων INMP441 συνδέεται μέσω του διαύλου I2S απευθείας στον ελεγκτή DMA (Direct Memory Access) [8, 20]. Αυτή η τοπολογία εξασφαλίζει σταθερή ροή δειγμάτων στη μνήμη του μικροελεγκτή χωρίς την παραμικρή παρέμβαση της CPU, αποτρέποντας τυχόν απώλεια δεδομένων (frame dropping) [4, 12]. Παράλληλα, το κεντρικό αναλογικό shotgun μικρόφωνο MAX9814 τροφοδοτεί ένα ανεξάρτητο κανάλι ADC, ενισχύοντας το σήμα της κύριας πηγής μέσω του Αυτόματου Ελέγχου Κέρδους (AGC) [13, 14].

Στο επίπεδο **Επεξεργασίας**, η διπύρνη (dual-core) δομή αξιοποιείται πλήρως [4]. Ο πυρήνας 0 (Core 0) του ESP32-S3 αναλαμβάνει αποκλειστικά τον δικτυακό έλεγχο (Wi-Fi AP) και τη λειτουργία του WebServer/WebSocket για την ασύρματη επικοινωνία με την εφαρμογή Flutter [6, 9]. Ο πυρήνας 1 (Core 1) αφιερώνεται στις κρίσιμες χρονικά διεργασίες DSP, υπολογίζοντας τον Ταχύ Μετασχηματισμό Fourier (FFT) [1] και εκτελώντας τον αλγόριθμο GCC-PHAT [2, 7]. Στο ίδιο περιβάλλον εκτελούνται και τα μοντέλα μηχανικής μάθησης του TinyML [3]. Σε επίπεδο **Εξόδου**, ο μικροελεγκτής καθοδηγεί τον σερβοκινητήρα MG996R μέσω σημάτων PWM που παράγονται λογισμικά, αποθηκεύει αυτόνομα τα ηχητικά δεδομένα σε μορφή αρχείων WAV στην κάρτα MicroSD μέσω διαύλου SDMMC/SPI, και ταυτόχρονα μεταδίδει το ζωντανό ηχητικό stream στο δίκτυο αποκλειστικά μέσω binary frames στη σύνδεση WebSocket.

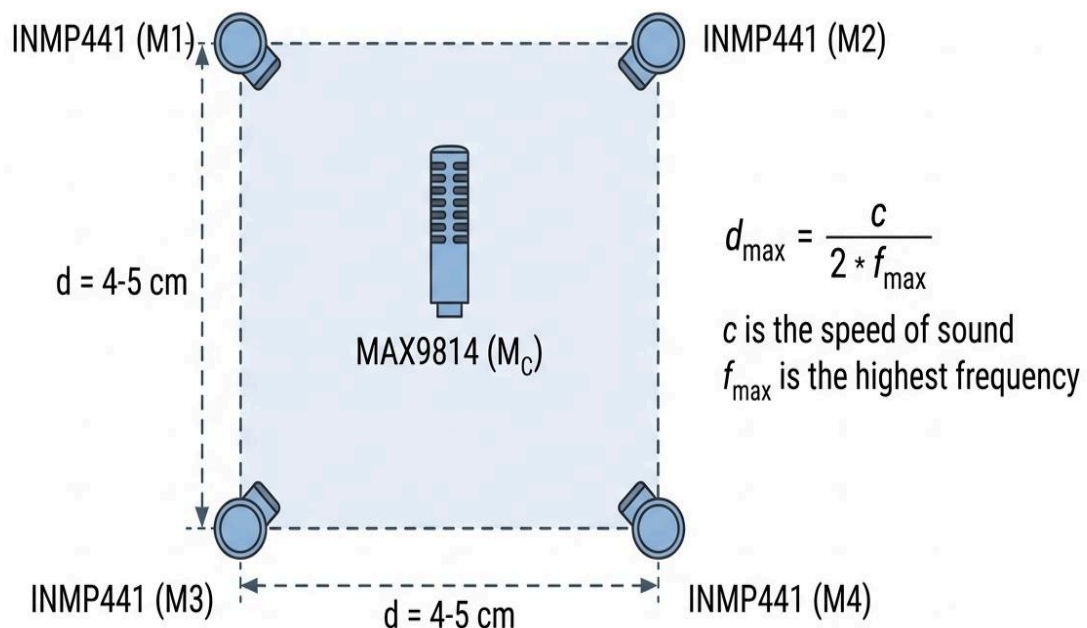


Εικόνα 20. Block Diagram Συστήματος , Πηγή:created in google flow

4.2 Διάταξη Μικροφώνων (Microphone Array Configuration)

Ο γεωμετρικός σχεδιασμός της συστοιχίας μικροφώνων αποτελεί τον κρίσιμότερο παράγοντα για την ακρίβεια του εντοπισμού ήχου [8, 15]. Μια απλή γραμμική διάταξη (Linear Array), όπου τα μικρόφωνα τοποθετούνται σε μία και μόνο ευθεία γραμμή, επιτρέπει τον υπολογισμό γωνιών αποκλειστικά στο εύρος από 0 έως 180 μοίρες και πάσχει εγγενώς από το φαινόμενο της συμμετρίας μπροστά-πίσω (front-back ambiguity). Το πρόβλημα αυτό προκύπτει διότι ένα ακουστικό κύμα που προσπίπτει από το μπροστινό ημιεπίπεδο δημιουργεί μαθηματικά ακριβώς την ίδια χρονική διαφορά άφιξης (TDOA) στα μικρόφωνα με ένα κύμα που προέρχεται από την αντίστοιχη κατοπτρική γωνία του πίσω ημιεπιπέδου. Κατά συνέπεια, ο αλγόριθμος καθίσταται πρακτικά ανίκανος να διακρίνει εάν μια πηγή βρίσκεται, για παράδειγμα, στις 45° μπροστά ή στις 135° στο αντίθετο ημιεπίπεδο [11]. Για ένα αυτόνομο κατευθυντικό σύστημα όπως αυτό της παρούσας εργασίας, το οποίο απαιτεί πλήρη χωρική αντίληψη 360 μοιρών προκειμένου να καθοδηγήσει με ακρίβεια τον μηχανισμό του σερβοκινητήρα προς οποιαδήποτε κατεύθυνση, αυτός ο γεωμετρικός περιορισμός είναι απολύτως απαγορευτικός και επιβάλλει τη μετάβαση σε συμμετρικές, δισδιάστατες (2D) τοπολογίες.

2D Square Microphone Array Configuration



Εικόνα 21.2D Συστοιχία μικροφώνων INMP441, Πηγή:created in google flow

Για την επίλυση αυτού του προβλήματος, επιλέχθηκε μια δισδιάστατη, **Τετραγωνική Διάταξη (Square Array)** με 4 μικρόφωνα στις κορυφές (συνολικής κάλυψης 360°), με το αναλογικό μικρόφωνο MAX9814 στο κέντρο [14]. Η απόσταση d μεταξύ δύο διαδοχικών μικροφώνων στην πλευρά του τετραγώνου επηρεάζει άμεσα το φαινόμενο της **Χωρικής Αναδίπλωσης (Spatial Aliasing)** [12]. Αν η απόσταση d είναι μεγαλύτερη από το μισό του μήκους κύματος ($\lambda/2$) της υψηλότερης συχνότητας-στόχου, ο αλγόριθμος TDOA παράγει πολλαπλές πιθανές γωνίες άφιξης [8, 15]. Η μέγιστη επιτρεπτή απόσταση υπολογίζεται μαθηματικά από τη σχέση:

$$d_{max} \leq \frac{c}{2 \cdot f_{max}} \quad [4.2]$$

Δεδομένου ότι το φάσμα της ανθρώπινης ομιλίας κυμαίνεται κυρίως από 300 Hz έως 4000 Hz, μια απόσταση της τάξης των 4-5 εκατοστών καλύπτει βέλτιστα το ακουστικό φάσμα [10, 16].

4.3 Υπολογισμός Γωνιακής Ανάλυσης

Η ακρίβεια (angular resolution) με την οποία το σύστημα μπορεί να εντοπίσει την ηχητική πηγή καθορίζεται μαθηματικά από τη συχνότητα δειγματοληψίας (f_s) και τη φυσική απόσταση (d) των μικροφώνων της συστοιχίας. Στην ουσία, το σύστημα προσπαθεί να μετρήσει μια απειροελάχιστη χρονική διαφορά άφιξης (TDOA) του ίδιου ακουστικού μετώπου μεταξύ των αισθητήρων.

Η ελάχιστη διαφορά χρόνου που μπορεί να ανιχνεύσει θεωρητικά το σύστημα, βασιζόμενο αποκλειστικά στα ψηφιακά δεδομένα, είναι η διάρκεια ενός και μόνο δείγματος, δηλαδή

$$\Delta t_{min} = \frac{1}{f_s} \quad [4.3]$$

Αν λειτουργούμε το σύστημα στα 16.000 Hz, η ελάχιστη χρονική διαφορά διαμορφώνεται στο $1/16000 = 62.5\mu s$. Σε αυτόν τον ελάχιστο χρόνο, ο ήχος (ο οποίος ταξιδεύει με ταχύτητα $c = 343m/s$) καλύπτει μια απόσταση

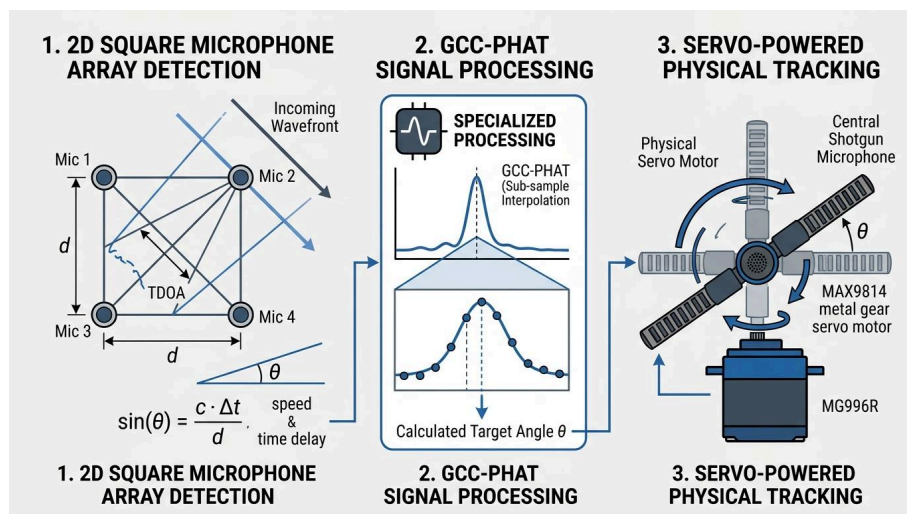
$$\Delta x = c \cdot \Delta t_{min} \approx 2.14 \text{ εκατοστά.} [4.3]$$

Αν η απόσταση d των μικροφώνων είναι καθορισμένη στα 4 εκατοστά, η μέγιστη διαφορά δειγμάτων όταν ο ήχος έρχεται από τα πλάγια (90 μοίρες) περιορίζεται σε μόλις 2 δείγματα. Αυτός ο δραστηρικός περιορισμός παρέχει μια εξαιρετικά «χονδρική» γωνιακή ανάλυση στο επίπεδο των ακέραιων δειγμάτων (sample-level), πράγμα που σημαίνει ότι το σύστημα θα μπορούσε να αναγνωρίσει μόνο έναν πολύ μικρό αριθμό διακριτών γωνιών, οδηγώντας σε μια "σπασμωδική" και ανακριβή στόχευση του σερβοκινητήρα.

Ωστόσο, για την υπέρβαση αυτού του ψηφιακού φραγμού και την επίτευξη υψηλής ακρίβειας κάτω του ενός δείγματος (sub-sample resolution), επιστρατεύεται ο αλγόριθμος GCC-PHAT [2], σε στενή συνδυαστική λειτουργία με προηγμένες τεχνικές Παρεμβολής (Interpolation). Εφαρμόζοντας, για παράδειγμα, παραβολική παρεμβολή γύρω από την κορυφή της συνάρτησης συσχέτισης (χρησιμοποιώντας τη μέγιστη τιμή και τα δύο γειτονικά της δείγματα), το σύστημα αποκτά την ικανότητα να ανασυνθέσει τη μαθηματική συνέχεια του σήματος και να υπολογίσει την πραγματική, κλασματική κορυφή της καμπύλης. Αυτό επιτρέπει την εξαγωγή χρονικών διαφορών αισθητά μικρότερων από τη διάρκεια ενός δείγματος [7]. Ο τελικός υπολογισμός της γωνίας θ εξάγεται πλέον με εξαιρετική ομαλότητα από τη γενική τριγωνομετρική σχέση [8]:

$$\theta = \arcsin((c \cdot \tau_{peak})/d) \quad [4.3]$$

Όπου τ_{peak} είναι ο εκτιμώμενος χρόνος καθυστέρησης έπειτα από την παρεμβολή. Η μαθηματική αυτή διαδικασία μεταφράζεται άμεσα στον αλγόριθμο του Firmware, εξασφαλίζοντας την εντολή ακριβούς γωνίας προς τον σερβοκινητήρα



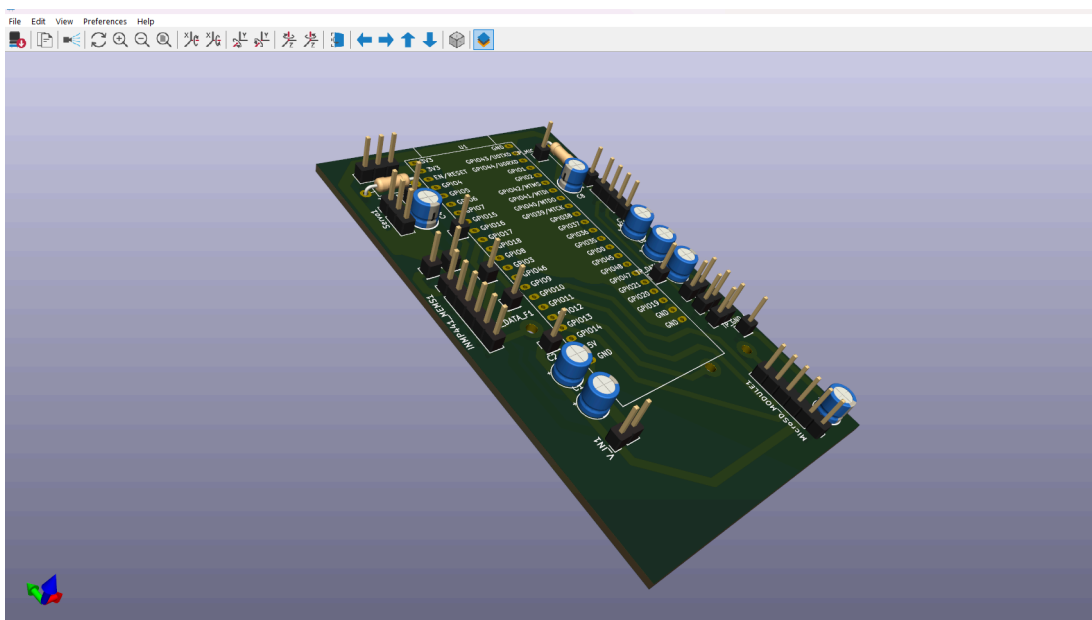
Εικόνα 22. Υπολογισμός Γωνίας ανάλυσης, Πηγή: created in google flow

4.4 Σχεδιασμός PCB / Wiring

Η ανάπτυξη και η υλοποίηση του ηλεκτρονικού κυκλώματος πραγματοποιήθηκε σε διαδοχικά και αυστηρά ελεγχόμενα στάδια, προκειμένου να διασφαλιστεί η ορθή λειτουργία, η ακεραιότητα των σημάτων και η ευστάθεια του συστήματος πριν από την τελική του μορφή. Αρχικά, όλες οι δοκιμές των επιμέρους υποσυστημάτων, η καλωδίωση της μικροφωνικής συστοιχίας και ο έλεγχος του σερβοκινητήρα έλαβαν χώρα σε πλακέτα δοκιμών (breadboard). Αυτό το προκαταρκτικό στάδιο στάθηκε θεμελιώδες, καθώς επέτρεψε την πειραματική επιβεβαίωση της σωστής επικοινωνίας του ευαίσθητου πρωτοκόλλου I2S και την ενδελεχή αποσφαλμάτωση του κώδικα σε πραγματικές συνθήκες λειτουργίας.

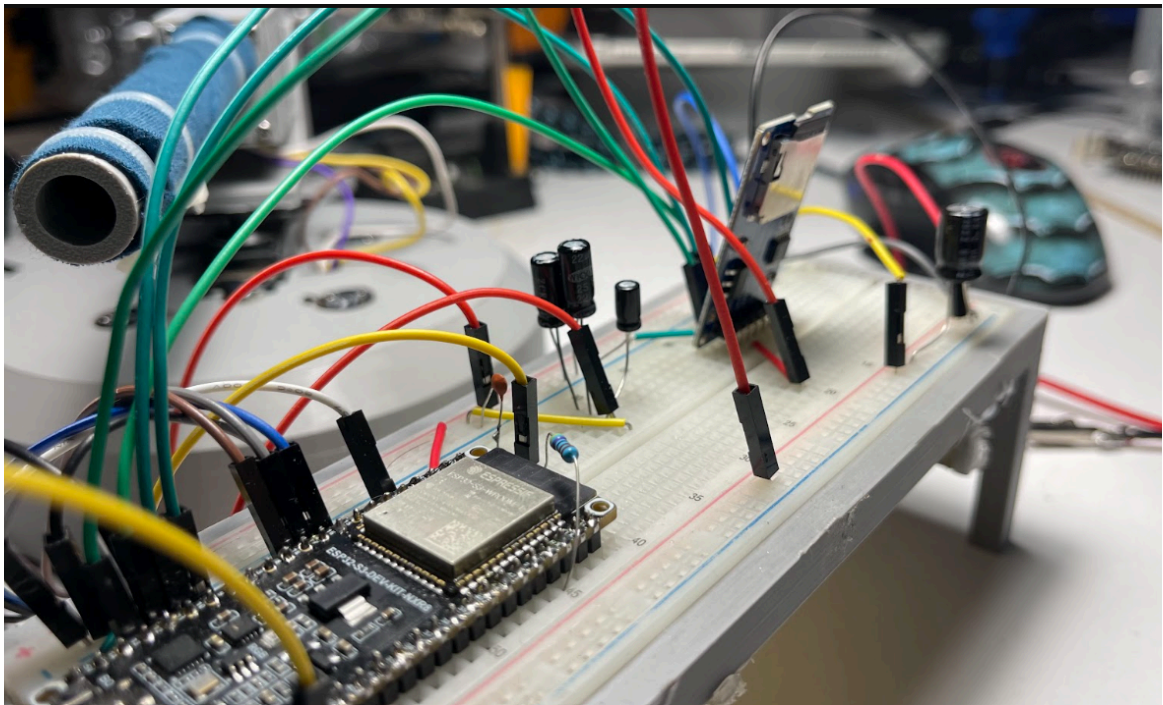
Έπειτα από την επιτυχή ολοκλήρωση των πειραματικών δοκιμών, προχωρήσαμε στον επαγγελματικό σχεδιασμό της τυπωμένης πλακέτας (PCB). Ο σχεδιασμός του ηλεκτρονικού κυκλώματος και η στρατηγική χωροθέτηση των εξαρτημάτων υλοποιήθηκαν εξ ολοκλήρου μέσω του εξειδικευμένου λογισμικού σχεδίασης ηλεκτρονικών KiCad. Για την κατασκευή του πρωτοτύπου PCB επιλέχθηκε η πρακτική μέθοδος της θερμικής μεταφοράς γραφίτη (toner transfer) με τη χρήση σιδήρου, ακολουθούμενη από τη διαδικασία της αποχάλκωσης. Η εν λόγω πλακέτα σχεδιάστηκε εξ αρχής ώστε να λειτουργεί ως ένας κεντρικός κόμβος (motherboard) διασύνδεσης.

Ο σχεδιασμός εστίασε αυστηρά στην ελαχιστοποίηση των ηλεκτρομαγνητικών παρεμβολών (EMI), δεδομένου ότι τα μικρόφωνα MEMS χρησιμοποιούν σήματα ρολογιού υψηλής συχνότητας (BCLK) τα οποία μπορούν να δημιουργήσουν παρεμβολές ή να αλλοιωθούν από άλλες πηγές του κυκλώματος. Για την αντιμετώπιση αυτών των προκλήσεων και τη βελτιστοποίηση της κατασκευής, εφαρμόστηκαν οι εξής τεχνικές:



Εικόνα 23. 3D Απεικόνιση Πλακέτας PCB, Πηγή:screenshot kiCad, pc

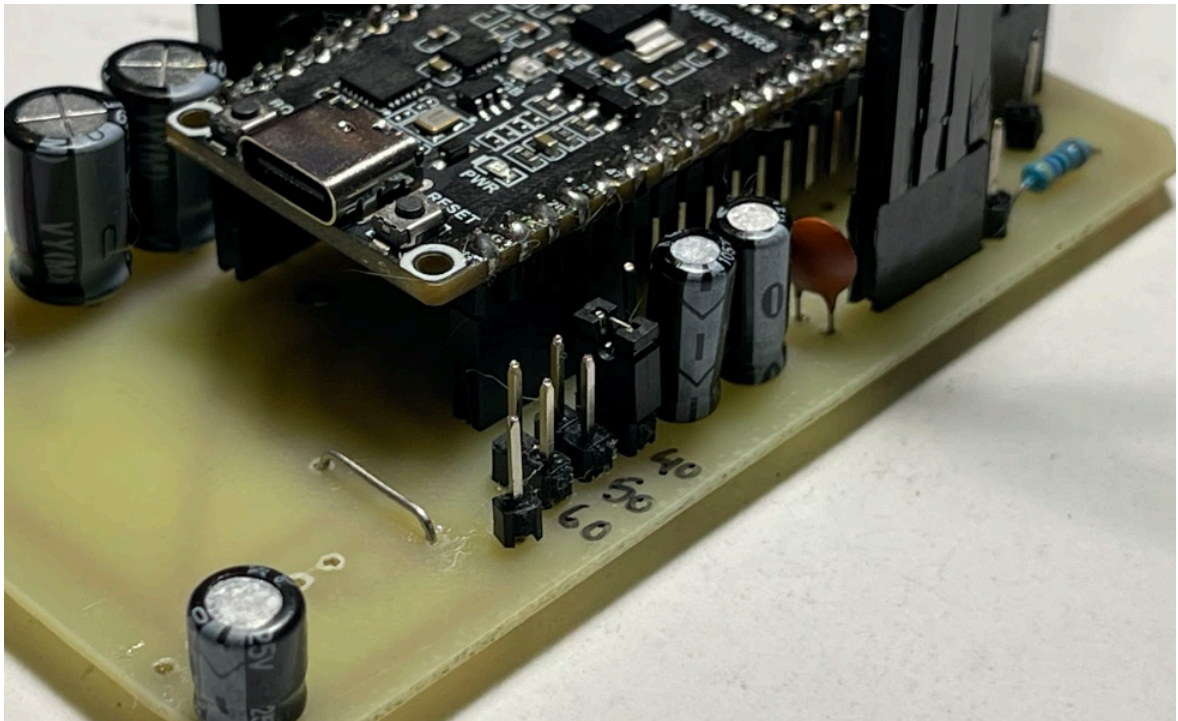
- **Αρθρωτή Αρχιτεκτονική (Modular Hub Design):** Η προσαρμοσμένη (custom) πλακέτα δεν φέρει κολλημένα τα ίδια τα μικρόφωνα ή τον μικροελεγκτή, αλλά λειτουργεί αποκλειστικά ως η κεντρική "μητρική" πλακέτα του συστήματος. Μέσω της χρήσης ακίδων σύνδεσης (pin headers), ενώνονται πάνω της οι ανεξάρτητες πλακέτες (breakout boards) των μικροφώνων MEMS INMP441, ο μικροελεγκτής ESP32-S3, καθώς και το αναλογικό μικρόφωνο MAX9814. Αυτή η προσέγγιση καθιστά την κεντρική πλακέτα τον φυσικό κόμβο διανομής σημάτων, εξασφαλίζοντας εύκολη αντικατάσταση τμημάτων και αποφυγή θερμικής καταπόνησης των ευαίσθητων αισθητήρων κατά τη συγκόλληση.
- **Πυκνωτές Αποσύζευξης (Decoupling Capacitors):** Κάθε μικρόφωνο INMP441 συνοδεύτηκε από έναν κεραμικό πυκνωτή απόσβεσης 0.1 μF , ο οποίος τοποθετήθηκε όσο το δυνατόν πιο κοντά στις ακίδες τροφοδοσίας του (VDD και GND). Αυτή η προσθήκη είναι ζωτικής σημασίας για την εξομάλυνση του θορύβου της γραμμής τροφοδοσίας, εξασφαλίζοντας μια καθαρή και σταθερή αναφορά τάσης για τον εσωτερικό μετατροπέα (ADC) του εκάστοτε μικροφώνου.



Εικόνα 24. Πυκνωτές Αποσύζευξης στην Πλακέτα δοκιμών, Πηγή: κάμερα κινητού

- **Διαχωρισμός Ψηφιακών και Αναλογικών Διαδρομών:** Στο επίπεδο του routing της κεντρικής πλακέτας, οι διάδρομοι χαλκού που μεταφέρουν τα ψηφιακά σήματα ρολογιού του πρωτοκόλλου I2S σχεδιάστηκαν με τρόπο ώστε να διατηρούν ασφαλή απόσταση από την αναλογική γραμμή του κεντρικού μικροφώνου, αποτρέποντας την επαγωγική μεταφορά θορύβου (crosstalk) μεταξύ των διαφορετικών σημάτων.

- **Διαστασιολόγηση Διαδρόμων Ισχύος:** Οι διάδρομοι χαλκού που διατρέχουν την κεντρική πλακέτα και συνδέουν τον κόμβο τροφοδοσίας με τις ακίδες ελέγχου του σερβοκινητήρα σχεδιάστηκαν με αυξημένο πλάτος. Αυτό διασφαλίζει ότι οι απότομες απαιτήσεις σε ρεύμα (current spikes) κατά την έναρξη της κίνησης του κινητήρα δεν θα προκαλέσουν πτώση τάσης (voltage drop) στην κεντρική πλακέτα, η οποία θα μπορούσε να οδηγήσει τον μικροελεγκτή ESP32 σε ανεπιθύμητη επανεκκίνηση.
- **Δυναμική Επιλογή Κέρδους (Gain Jumper Selection) για το MAX9814:** Αντί μιας μόνιμης (hardwired) καλωδίωσης της ακίδας GAIN του αναλογικού μικροφώνου, σχεδιάστηκε στην πλακέτα μια υποδοχή επιλογής 3 ακίδων (3-pin header/jumper). Η απόδοση του κυκλώματος Αυτόματου Ελέγχου Κέρδους (AGC) εξαρτάται άμεσα από τον περιβάλλοντα θόρυβο του εκάστοτε χώρου. Το jumper επιτρέπει στον χρήστη να ρυθμίζει το μέγιστο κέρδος του μικροφώνου σε επίπεδο υλικού (hardware), χωρίς να απαιτείται καμία αλλαγή στον κώδικα, επιλέγοντας άμεσα μεταξύ 60 dB (ασύνδετη ακίδα), 50 dB (σύνδεση στο VDD) και 40 dB (σύνδεση στο GND). Ειδικότερα, σε συνθήκες χαμηλού περιβαλλοντικού θορύβου (π.χ. κλειστό γραφείο), η επιλογή των 60 dB μεγιστοποιεί την εμβέλεια λήψης. Αντίθετα, σε θορυβώδεις χώρους, η μείωση του κέρδους στα 50 dB ή 40 dB αποτρέπει τον κορεσμό (saturation) του μετατροπέα ADC και το φαινόμενο της ψαλιδισμένης κυματομορφής (clipping), διατηρώντας το δυναμικό εύρος του σήματος ακέραιο.



Εικόνα 25. Επιλογέας Κέρδους Μικροφώνου, Πηγή: κάμερα κινητού

- Αντιμετώπιση Ηλεκτρομαγνητικού Θορύβου (EMI) από τη Μονάδα Wi-Fi:** Κατά την ασύρματη αποστολή του ηχητικού σήματος, παρατηρήθηκε ότι ο μετατροπέας ADC του μικροελεγκτή κατέγραφε εκκωφαντικό στατικό θόρυβο. Το φαινόμενο αυτό οφειλόταν στις απότομες απαιτήσεις ρεύματος του πομποδέκτη Wi-Fi και στη λειτουργία των καλωδίων διασύνδεσης ως ακούσιων κεραιών λήψης. Για την εξάλειψη αυτού του κρίσιμου προβλήματος, εφαρμόστηκαν τρεις συνδυαστικές λύσεις: Πρώτον, τοποθετήθηκαν κεραμικοί πυκνωτές απόξευξης 0.1μF ακριβώς στους ακροδέκτες VDD και GND του μικροφώνου για την εξομάλυνση των μικρο-πτώσεων τάσης. Δεύτερον, σχεδιάστηκε ένα αναλογικό βαθυπερατό φίλτρο (RC Low-Pass Filter) στον ακροδέκτη εξόδου του ήχου (OUT) για την αποκοπή των υψηλόσυχων ραδιοφωνικών παρεμβολών πριν φτάσουν στον ADC. Τρίτον, το καλώδιο μεταφοράς του σήματος συστράφηκε με το καλώδιο της γείωσης (twisted pair), ακυρώνοντας έτσι την επαγωγική μεταφορά θορύβου (crosstalk). Η επιτυχής εφαρμογή αυτών των παρεμβάσεων μείωσε δραματικά το κατώφλι θορύβου (noise floor) του συστήματος, διασφαλίζοντας την καθαρότητα του ακουστικού σήματος.
- Σχεδίαση για Ελεγχιμότητα (Design for Testability - DFT) μέσω Test Points:** Κατά τη σχεδίαση του PCB, ενσωματώθηκαν στρατηγικά σημεία δοκιμής (Test Points - γυμνά pads χαλκού) στις κρίσιμες γραμμές του κυκλώματος. Η πρακτική αυτή υιοθετήθηκε για την ασφαλή επαλήθευση της ακεραιότητας του σήματος (signal integrity) και τον εντοπισμό σφαλμάτων (debugging) με τη χρήση παλμογράφου, καθώς η απευθείας μέτρηση στα μικροσκοπικά εξαρτήματα επιφανειακής στήριξης (SMD) εγκυμονεί σοβαρούς κινδύνους βραχυκυκλώματος. Συγκεκριμένα, τα Test Points τοποθετήθηκαν: α) Στις γραμμές τροφοδοσίας (3.3V, 5V, GND) για τον έλεγχο τυχόν κυματισμών (ripple) και θορύβου EMI από το Wi-Fi, β) στη διαδρομή του αναλογικού σήματος (πριν και μετά το βαθυπερατό RC φίλτρο) για την αξιολόγηση της εξομάλυνσης του σήματος, και γ) στον ψηφιακό δίαυλο I2S (στις γραμμές χρονισμού SCK/WS και δεδομένων) για τον έλεγχο τυχόν παραμορφώσεων στις ψηφιακές παλμοσειρές.

Συνοψίζοντας, ο σχεδιασμός της τυπωμένης πλακέτας (PCB) δεν αποτέλεσε απλώς ένα παθητικό μέσο μηχανικής στήριξης, αλλά έναν ενεργό παράγοντα βελτίωσης της ποιότητας του συστήματος. Η μετάβαση από τις δοκιμές του ράστερ (breadboard) σε μια επαγγελματικά δρομολογημένη πλακέτα, εξάλειψε τις παρασιτικές χωρητικότητες, σταθεροποίησε την τροφοδοσία μέσω των τοπικών πυκνωτών απόξευξης και θωράκισε το ευαίσθητο αναλογικό σήμα έναντι των ισχυρών ραδιοσυχνοτήτων του ESP32. Το αποτέλεσμα είναι μια στιβαρή και αξιόπιστη ηλεκτρονική βάση, απολύτως ικανή να υποστηρίξει την απαιτητική ψηφιακή επεξεργασία σήματος (DSP) και τις λειτουργίες μηχανικής μάθησης (TinyML) που ακολουθούν στα επόμενα στάδια της υλοποίησης.

4.5 Κατασκευή Τυπωμένης Πλακέτας (PCB Fabrication)

Η μετάβαση από τον ψηφιακό σχεδιασμό στην υλοποίηση του φυσικού υλικού (hardware) αποτελεί ένα από τα κρίσιμότερα στάδια ανάπτυξης του συστήματος. Η κατασκευή της πρωτότυπης πλακέτας τυπωμένου κυκλώματος (PCB) πραγματοποιήθηκε in-house στο εργαστήριο, επιτρέποντας την ταχεία πρωτοτυποποίηση (rapid prototyping) και την άμεση πειραματική επαλήθευση των κανόνων σχεδίασης —όπως η τοποθέτηση των πυκνωτών απόζευξης και η ελαχιστοποίηση του ηλεκτρομαγνητικού θορύβου (EMI)— σε πραγματικές συνθήκες λειτουργίας.

Για την υλοποίηση, επιλέχθηκε η πρακτική μέθοδος της Θερμικής Μεταφοράς Γραφίτη (Toner Transfer Method). Η συγκεκριμένη τεχνική, αν και προσιτή, προσφέρει εξαιρετική γεωμετρική ακρίβεια, ικανή να υποστηρίξει διαδρόμους (traces) μικρού πλάτους για εξαρτήματα επιφανειακής στήριξης (SMD) καθώς και κλασικά εξαρτήματα διαμπερών οπών (Through-Hole). Η διαδικασία κατασκευής εκτελέστηκε μέσα από τα ακόλουθα αυστηρά καθορισμένα στάδια:

- Εξαγωγή Σχεδίου και Εκτύπωση Μάσκας (Layout Printing):** Το τελικό ψηφιακό σχέδιο των χαλκοδιαδρόμων (routing), όπως εξήχθη από το περιβάλλον του KiCad, εκτυπώθηκε σε κατοπτρική μορφή (mirrored) χρησιμοποιώντας εκτυπωτή τεχνολογίας Laser. Η επιλογή του μέσου εκτύπωσης είναι καθοριστική για την επιτυχία της μεθόδου. Χρησιμοποιήθηκε ειδικό, γυαλιστερό φωτογραφικό χαρτί (glossy photo paper), η λεία επικάλυψη του οποίου αποτρέπει την απορρόφηση των μικροσωματιδίων του γραφίτη (toner) στους πόρους του χαρτιού, επιτρέποντας την πλήρη και ομοιόμορφη αποκόλλησή του στο επόμενο στάδιο.



Εικόνα 26. Προετοιμασία πλακέτας PCB, Πηγή: κάμερα κινητού.

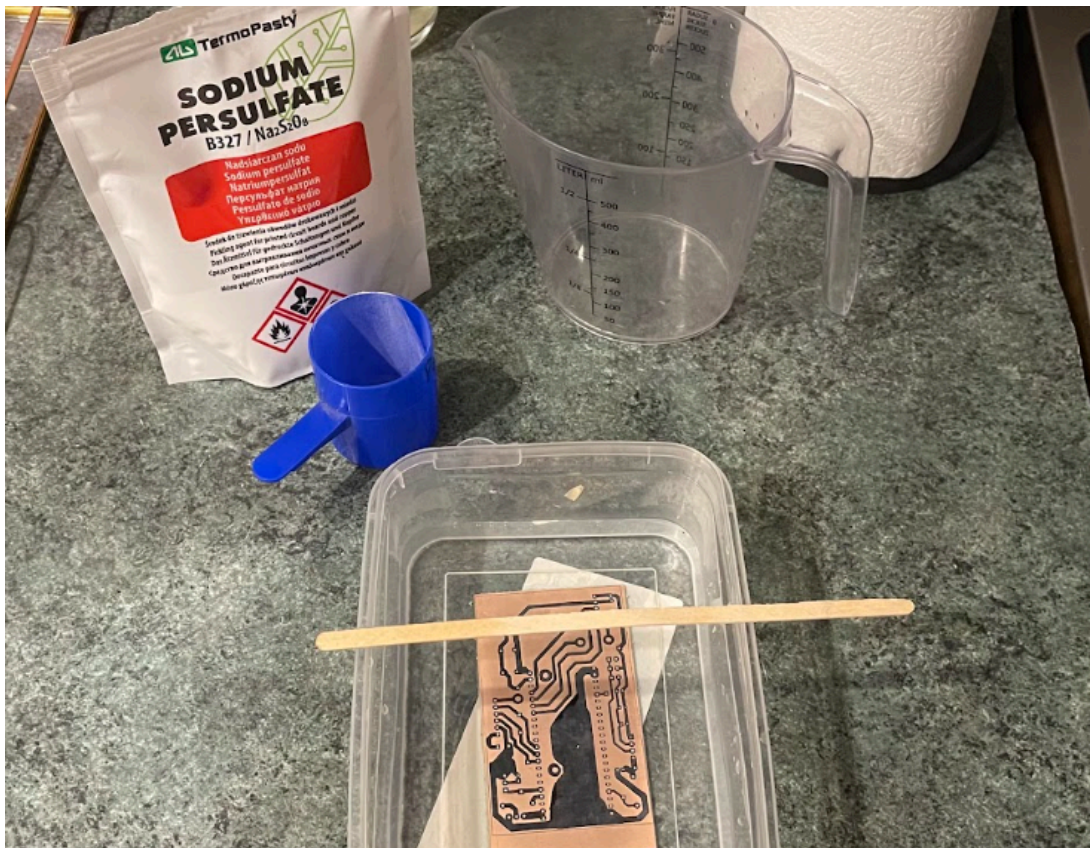
- **Μηχανική και Χημική Προετοιμασία Επιφάνειας:** Ένα ακατέργαστο φύλλο βακελίτη με επίστρωση χαλκού (Copper Clad Board) κόπηκε στις ακριβείς διαστάσεις του ψηφιακού σχεδίου. Ακολούθησε σχολαστικός μηχανικός καθαρισμός της επιφάνειας του χαλκού με λεπτό λειαντικό σφουγγάρι, προκειμένου να αφαιρεθεί το επιφανειακό στρώμα οξείδωσης (Cu_2O). Στη συνέχεια, εφαρμόστηκε χημικός καθαρισμός με ισοπροπυλική αλκοόλη (IPA) για την πλήρη αφαίρεση λιπαρών υπολειμμάτων που θα εμπόδιζαν την πρόσφυση του πολυμερούς.
- **Θερμική Μεταφορά Γραφίτη (Thermal Toner Transfer):** Το εκτυπωμένο φωτογραφικό χαρτί τοποθετήθηκε με την τυπωμένη όψη σε απόλυτη επαφή με τον καθαρό χαλκό. Με τη χρήση θερμαινόμενης πλάκας (κοινού σίδηρου), εφαρμόστηκε υψηλή θερμοκρασία και σταθερή, ομοιόμορφη μηχανική πίεση. Η διαδικασία αυτή εκμεταλλεύεται τη θερμοπλαστική φύση του γραφίτη των Laser εκτυπωτών· η θερμότητα τήκει το πλαστικό πολυμερές του γραφίτη, αναγκάζοντάς τον να μεταφερθεί και να πολυμεριστεί ισχυρά πάνω στη μεταλλική επιφάνεια του χαλκού.



Εικόνα 27. Θερμική Μεταφορά Γραφίτη σε PCB, Πηγή: κάμερα κινητού.

- **Υδρόλυση και Απομάκρυνση Μήτρας Χαρτιού:** Μετά τη φυσική ψύξη της πλακέτας, ώστε να σταθεροποιηθεί το πολυμερές, η πλακέτα βυθίστηκε σε λουτρό χλιαρού νερού. Το νερό διεισδύει και διαλύει σταδιακά τη χαρτόμαζα, αφήνοντας ανέπαφο τον αδιάβροχο γραφίτη. Τα τελευταία υπολείμματα χαρτιού αφαιρέθηκαν με εξαιρετικά ήπια μηχανική τριβή. Σε αυτό το σημείο, το ψηφιακό σχέδιο έχει αποτυπωθεί τέλεια, με τον γραφίτη να λειτουργεί πλέον ως χημική μάσκα προστασίας (etch resist).

- **Χημική Αποχάλκωση (Chemical Etching):** Η πλακέτα βυθίστηκε σε χημικό διάλυμα τριχλωριούχου σιδήρου (FeCl_3), το οποίο διατηρούνταν σε συνεχή μηχανική ανάδευση για την επιτάχυνση της αντίδρασης. Μέσω μιας οξειδοαναγωγικής αντίδρασης, το χημικό διάλυμα διαβρώνει και απομακρύνει όλον τον απροστάτευτο (εκτεθειμένο) χαλκό. Αντίθετα, ο χαλκός που βρίσκεται εγκλωβισμένος κάτω από το προστατευτικό στρώμα του γραφίτη παραμένει απολύτως άθικτος, διαμορφώνοντας τους τελικούς ηλεκτρικούς διαδρόμους του κυκλώματος.

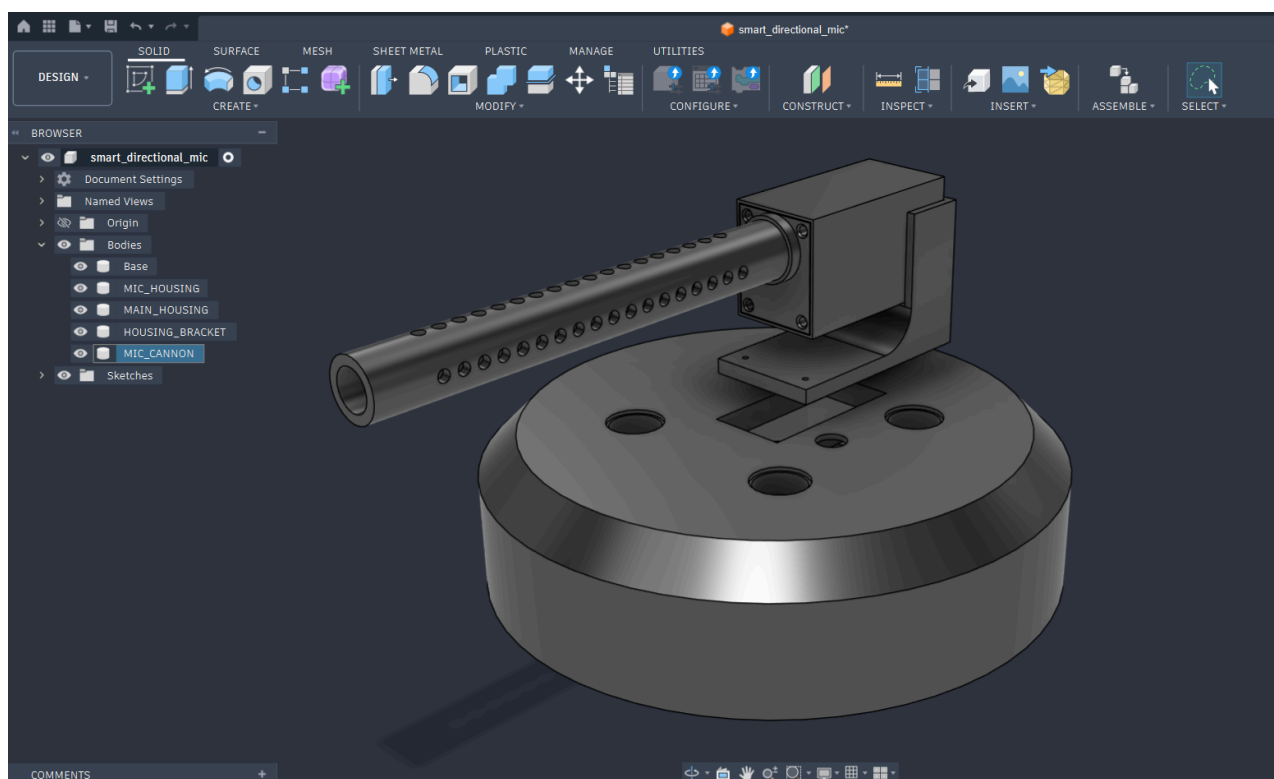


Εικόνα 28. Αποχάλκωση Πλακέτας PCB, Πηγή: κάμερα κινητού

- **Αποκατάσταση, Διάντρηση (Drilling) και Επικασσιτέρωση:** Μετά την ολοκλήρωση της αποχάλκωσης, η πλακέτα ξεπλύνθηκε σχολαστικά για την ουδετεροποίηση των χημικών αντιδραστήριων. Ο προστατευτικός γραφίτης αφαιρέθηκε οριστικά με τη χρήση διαλύτη (ασετόν), αποκαλύπτοντας τους καθαρούς χαλκοδιαδρόμους. Το τελικό κατασκευαστικό στάδιο περιελάμβανε τη διάντρηση των οπών μέσω εξειδικευμένου μικρο-δράπανου (0.8mm – 1.0mm) για την υποδοχή των pin headers, επιτρέποντας την τελική και ασφαλή συγκόλληση του ESP32-S3, του Module της MicroSD και των ακροδεκτών των αισθητήρων.

ΚΕΦΑΛΑΙΟ 5: ΜΗΧΑΝΟΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ 3D ΕΚΤΥΠΩΣΗ

Η επιτυχής λειτουργία ενός έξυπνου κατευθυντικού μικροφώνου αποτελεί προϊόν μιας ολιστικής προσέγγισης, η οποία δεν εξαρτάται αποκλειστικά από την ποιότητα των ηλεκτρονικών εξαρτημάτων (hardware) και την πολυπλοκότητα του αναπτυχθέντος κώδικα (software), αλλά εξίσου και από την αρτιότητα της μηχανικής του δομής (mechanics). Το φυσικό περίβλημα της συσκευής δεν λειτουργεί απλώς ως ένα παθητικό μέσο προστασίας και στέγασης των κυκλωμάτων, αλλά διαδραματίζει απολύτως ενεργό ρόλο στη διαδικασία λήψης του ήχου. Προσφέρει την απαραίτητη ακουστική απομόνωση, την κρίσιμη απόσβεση των παραγόμενων μηχανικών κραδασμών από τα κινούμενα μέρη και την απόλυτη γεωμετρική ακρίβεια στη χωροθέτηση των αισθητήρων. Στο παρόν κεφάλαιο αναλύεται διεξοδικά η διαδικασία του μηχανολογικού σχεδιασμού και της κατασκευής των φυσικών μερών του συστήματος. Η υλοποίηση αυτή βασίστηκε αποκλειστικά σε σύγχρονες τεχνικές σχεδίασης μέσω υπολογιστή (Computer-Aided Design - CAD) και προσθετικής κατασκευής (3D Printing), τεχνολογίες που επέτρεψαν την ταχεία πρωτοτυποποίηση (rapid prototyping) και τη συνεχή, επαναληπτική βελτίωση των εξαρτημάτων μέχρι την επίτευξη του βέλτιστου τελικού αποτελέσματος.



Εικόνα 29.Ολοκληρωμένο Σχέδιο Συστήματος, Πηγή:screenshot fushion360, pc

5.1 Εισαγωγή στον Σχεδιασμό CAD και την Εκτύπωση

Για τη μοντελοποίηση όλων των επιμέρους μηχανικών εξαρτημάτων, επιλέχθηκε το επαγγελματικό περιβάλλον σχεδίασης **Fusion 360 της Autodesk**. Το συγκεκριμένο λογισμικό παρέιχε την απαραίτητη ακρίβεια για τον καθορισμό των ανοχών (tolerances) και τη δημιουργία πολύπλοκων γεωμετριών που απαιτούνται για τη στέγαση ευαίσθητων ακουστικών στοιχείων. Ο σχεδιασμός πραγματοποιήθηκε εξ αρχής (from scratch), επιτρέποντας τον πλήρη έλεγχο στις διαστάσεις και τη συναρμολόγηση του τελικού πρωτοτύπου σε εικονικό περιβάλλον (assembly mode) πριν την παραγωγή του.

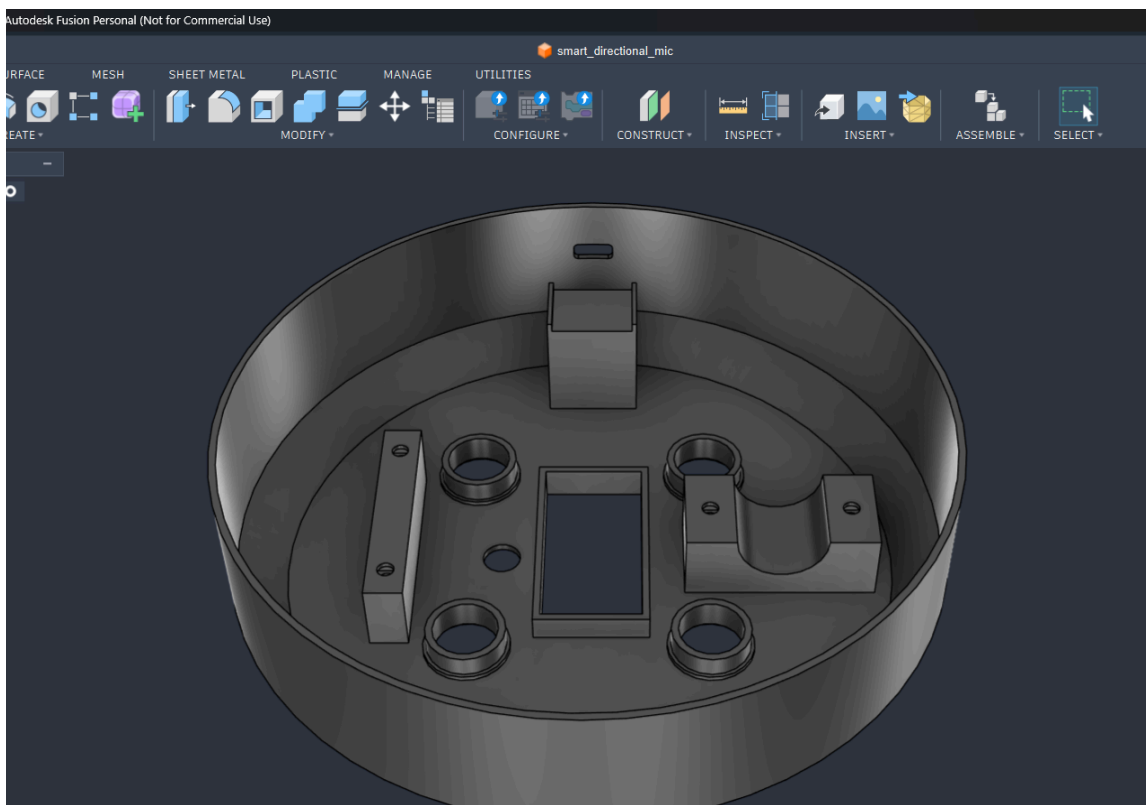


Εικόνα 30. 3D Printing Βάσης Συστήματος, Πηγή: κάμερα κινητού

Η κατασκευή των φυσικών δοκιμών και του τελικού περιβλήματος υλοποιήθηκε με τη χρήση του 3D εκτυπωτή **Creality K1C**. Η αρχιτεκτονική CoreXY του εν λόγω μηχανήματος, σε συνδυασμό με τον κλειστό θάλαμο εκτύπωσης, διασφάλισε ταχεία παραγωγή με εξαιρετική ποιότητα επιφάνειας και ελαχιστοποίηση των παραμορφώσεων (warping). Ως υλικό εκτύπωσης χρησιμοποιήθηκε το **PETG** (Polyethylene Terephthalate Glycol). Η επιλογή του PETG έναντι του κοινού PLA έγινε διότι προσφέρει ανώτερη μηχανική αντοχή, ελαφρώς μεγαλύτερη ελαστικότητα χρήσιμη για την απορρόφηση κραδασμών— και αυξημένη αντοχή σε υψηλότερες θερμοκρασίες λειτουργίας, καθιστώντας την κατασκευή κατάλληλη για πολύωρη χρήση.

5.2 Σχεδιασμός Κελύφους και Διάταξης Ηλεκτρονικών (Το Καλούπι)

Η "καρδιά" της μηχανολογικής κατασκευής είναι το κεντρικό καλούπι (housing), το οποίο σχεδιάστηκε με διπλό σκοπό: την προστασία των ηλεκτρονικών και τη διασφάλιση της ακουστικής συμμετρίας. Στο εσωτερικό του περιβλήματος προβλέφθηκαν ειδικές υποδοχές με αυστηρές ανοχές (εφαρμογή τύπου press-fit ή με στηρίγματα βιδών) για την ασφαλή τοποθέτηση της custom πλακέτας (PCB), του μικροελεγκτή ESP32-S3 [4], της μπαταρίας ιόντων λιθίου 18650, του κυκλώματος προστασίας (BMS) και του μετατροπέα ανύψωσης τάσης (boost converter). Ιδιαίτερη έμφαση δόθηκε στην εξωτερική πρόσοψη του κελύφους, όπου εδράζονται τα τέσσερα ψηφιακά μικρόφωνα MEMS INMP441 [8] και το αναλογικό μικρόφωνο MAX9814 [14]. Ο σχεδιασμός στο Fusion 360 εξασφάλισε την απόλυτη γεωμετρική συμμετρία και τη σταθερή απόσταση μεταξύ των μικροφώνων. Αυτή η μηχανολογική ακρίβεια είναι αδιαπραγμάτευτη, καθώς οποιαδήποτε απόκλιση στη χιλιοστομετρική τοποθέτηση των μικροφώνων σε μια συστοιχία (array) θα εισήγαγε μόνιμο σφάλμα στους μαθηματικούς υπολογισμούς της χρονικής διαφοράς άφιξης (TDOA) και του αλγορίθμου GCC-PHAT [2, 7, 15].

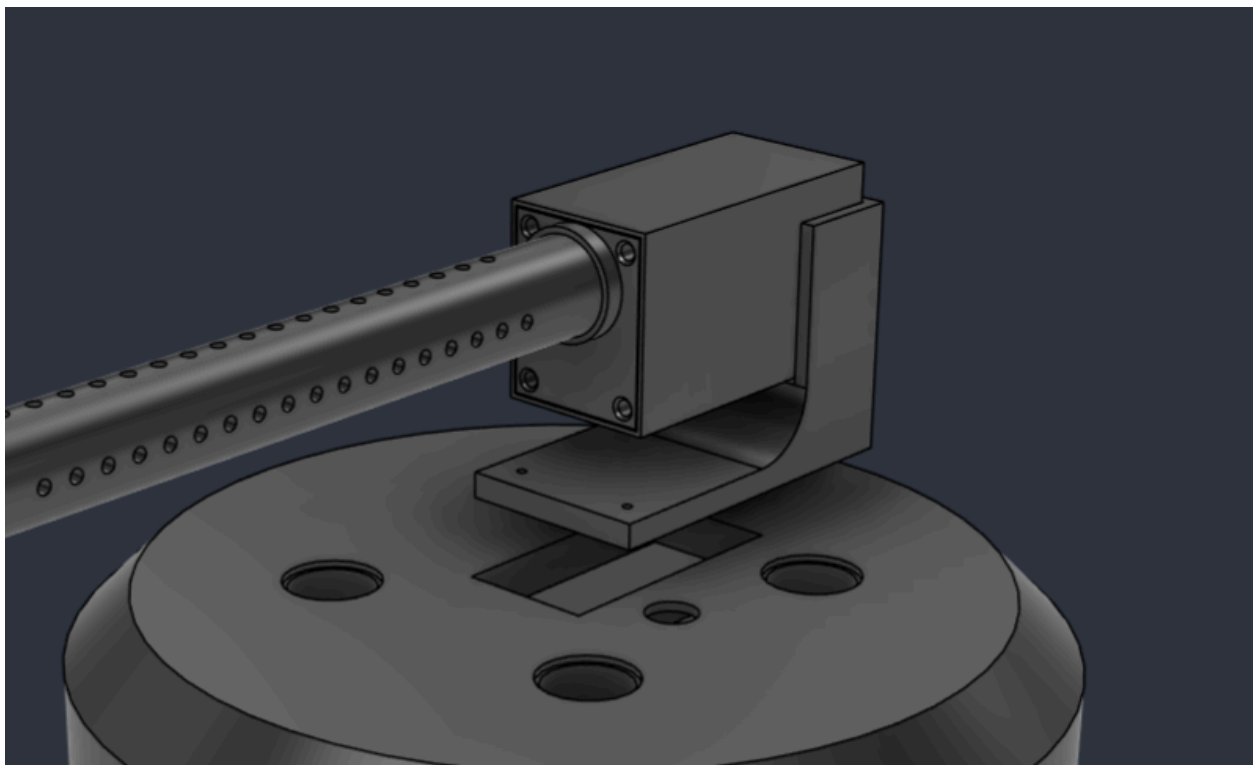


Εικόνα 31. Σχεδιασμός βάσης Συστήματος, Πηγή: screenshot fusion360, pc

5.3 Σύστημα Περιστροφής και Βάση Στήριξης

Για να λειτουργήσει το σύστημα ως "έξυπνο" κατευθυντικό μικρόφωνο, πρέπει να έχει την ικανότητα φυσικής περιστροφής προς την πηγή του ήχου. Αυτό επιτυγχάνεται μέσω μιας στιβαρής βάσης στήριξης, η οποία φιλοξενεί τον σερβοκινητήρα **MG996R**. Επειδή ο συγκεκριμένος κινητήρας διαθέτει μεταλλικά γρανάζια και παράγει υψηλή ροπή, εισάγει αναπόφευκτα μηχανικούς κραδασμούς κατά τη λειτουργία του.

Για την αντιμετώπιση αυτού του προβλήματος, η βάση σχεδιάστηκε με γνώμονα την απόσβεση των ταλαντώσεων (vibration dampening). Ο κεντρικός άξονας περιστροφής του κελύφους δεν στηρίζεται απευθείας και αποκλειστικά στον άξονα του servo. Αντίθετα, σχεδιάστηκε μια ειδική εσοχή υποδοχής για την ενσωμάτωση ενός μηχανικού ρουλεμάν (ball bearing). Η χρήση του ρουλεμάν αναλαμβάνει το αξονικό φορτίο και το βάρος του επάνω τμήματος της κατασκευής, ελαχιστοποιώντας την τριβή και ανακουφίζοντας τον κινητήρα. Αυτό έχει ως αποτέλεσμα μια εξαιρετικά ομαλή, αθόρυβη και ταχεία περιστροφή, αποτρέποντας τη μεταφορά δομικού θορύβου (structure-borne noise) στα μικρόφωνα την ώρα που το σύστημα "λοκάρει" στον στόχο.



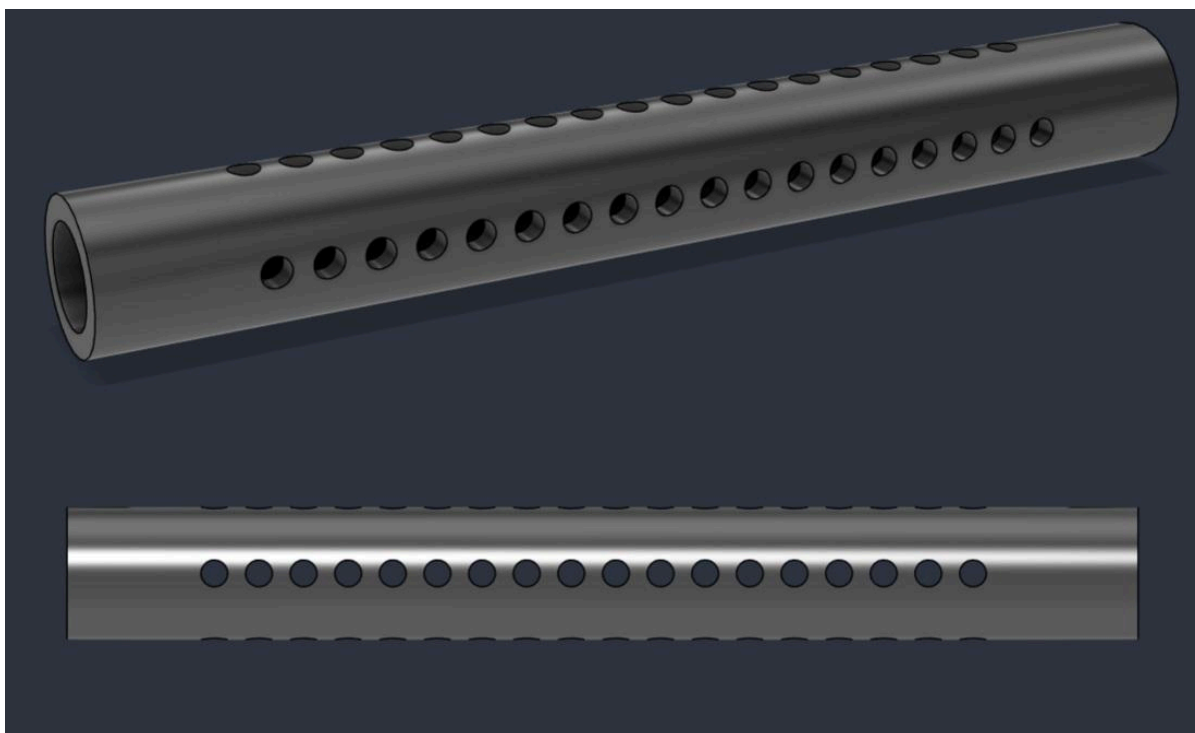
Εικόνα 32. Σχεδιασμός βάσης Περιστροφής, Πηγή:screenshot fushion360, pc

5.4 Σχεδιασμός Κατευθυντικού "Κανονιού" (Shotgun Tube)

Το πιο κρίσιμο ίσως ακουστικό στοιχείο του μηχανολογικού σχεδιασμού είναι η ενσωμάτωση ενός σωληνωτού εξαρτήματος, ευρέως γνωστού στην ακουστική μηχανική ως "κανόνι" (shotgun tube) ή σωλήνας συμβολής (interference tube) [14]. Αντί να βασίζεται το σύστημα αποκλειστικά στη ψηφιακή επεξεργασία σήματος για την απομόνωση του ήχου, το 3D εκτυπωμένο κανόνι προσφέρει ένα πρώτο, ισχυρό στάδιο φυσικής ακουστικής απομόνωσης.

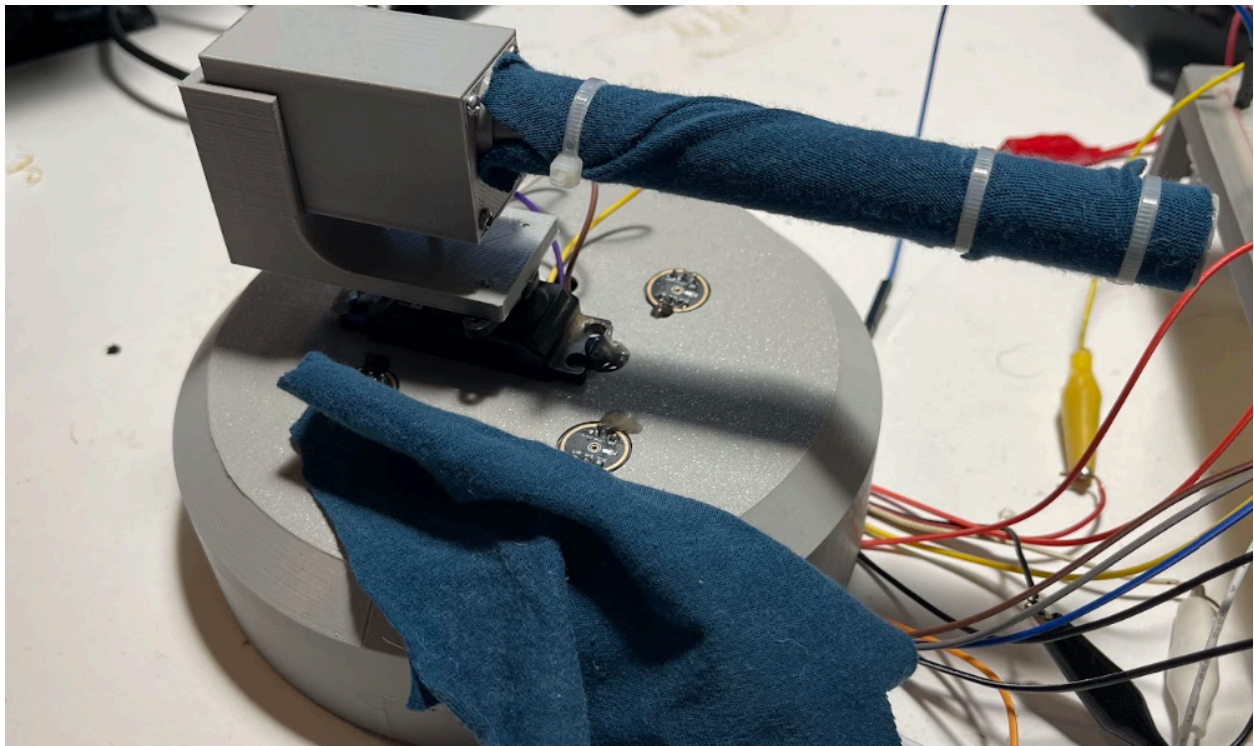
Ο σχεδιασμός του σωλήνα περιλαμβάνει προσεκτικά μελετημένες πλευρικές σχισμές κατά μήκος του. Όταν ένα ηχητικό κύμα προσπίπτει ακριβώς από μπροστά (on-axis, στις 0 μοίρες), ταξιδεύει ανεμπόδιστα μέσα στον σωλήνα μέχρι την κάψα του μικροφώνου. Ωστόσο, όταν ο ήχος προέρχεται από τα πλάγια ή από πίσω (off-axis), εισέρχεται στον σωλήνα μέσα από τις διάφορες σχισμές σε διαφορετικούς χρόνους. Αυτό προκαλεί καταστροφική συμβολή (destructive interference) στο εσωτερικό του σωλήνα για τις εκτός άξονα πηγές, ακυρώνοντας φυσικά τον ανεπιθύμητο θόρυβο και την αντήχηση του δωματίου [14].

Ο συνδυασμός αυτού του αυστηρού μηχανολογικού σχεδιασμού (που στενεύει δραστικά το πολικό διάγραμμα λήψης του μικροφώνου) με την ικανότητα του συστήματος να περιστρέφεται αυτόματα μέσω του σερβοκινητήρα, είναι αυτός που καθιστά το μικρόφωνο πραγματικά κατευθυντικό, επιτρέποντάς του να απομονώνει κρυστάλλινα την επιθυμητή πηγή ακόμα και σε θορυβώδη περιβάλλοντα.



Εικόνα 33. Σχεδιασμός κατευθυντικού κανονιού, Πηγή:screenshot fushion360, pc

Κατά τη διάρκεια των πειραματικών δοκιμών του μηχανολογικού σχεδίου σε πραγματικές συνθήκες, προέκυψε το ζήτημα της ακουστικής διαρροής (Acoustic Sealing & Resistance). Ο 3D εκτυπωμένος σωλήνας λειτουργούσε ακούσια ως "αντηχείο", επιτρέποντας στους πλευρικούς ήχους να διέρχονται ταχύτατα και ελεύθερα από τις οπές, ενώ ταυτόχρονα υπήρχε διαρροή ήχου από μικρο-κενά στο σημείο ένωσης της κάψας με το πλαστικό. Για την αντιμετώπιση αυτού του προβλήματος, το διάτρητο τμήμα του σωλήνα (μήκους 90 mm) περιτυλίχθηκε με ένα λεπτό, πορώδες ύφασμα. Το υλικό αυτό προσθέτει την απαραίτητη "ακουστική αντίσταση", καθυστερώντας ελαφρώς την είσοδο των πλευρικών ηχητικών κυμάτων, ώστε να επιτευχθεί η σωστή καταστροφική συμβολή στο εσωτερικό του θαλάμου. Επιπροσθέτως, το πίσω μέρος της κάψας σφραγίστηκε εσωτερικά με ειδικό μονωτικό υλικό, διασφαλίζοντας ότι η ακουστική πίεση φτάνει στο διάφραγμα του μικροφώνου αποκλειστικά από τον επιθυμητό, κεντρικό άξονα στόχευσης.



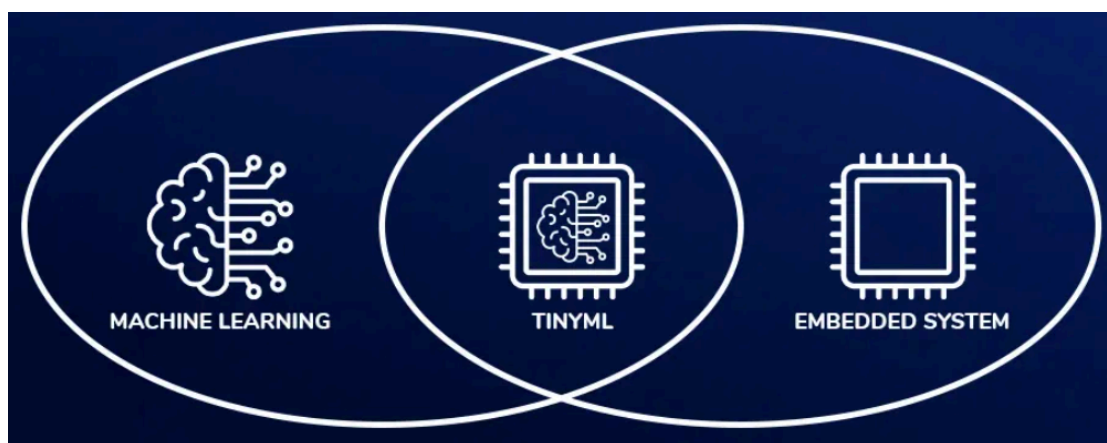
Εικόνα 34. Βαμβακερό κάλυμμα κανονιού, Πηγή: κάμερα κινητού

ΚΕΦΑΛΑΙΟ 6: ΥΛΟΠΟΙΗΣΗ FIRMWARE

Το παρόν κεφάλαιο αναλύει την αρχιτεκτονική του λογισμικού που αναπτύχθηκε για τον μικροελεγκτή ESP32-S3 [4]. Η ορθή λειτουργία ενός έξυπνου κατευθυντικού μικροφώνου απαιτεί την ταυτόχρονη και αυστηρά χρονισμένη εκτέλεση πολλαπλών διεργασιών, όπως η συνεχής λήψη ψηφιακού ήχου μέσω πρωτοκόλλου I2S [20], η άμεση επεξεργασία του σήματος (DSP) [1], ο ακριβής έλεγχος του σερβοκινητήρα και η δικτυακή επικοινωνία.

Σε ένα τόσο απαιτητικό υπολογιστικό περιβάλλον, η παραδοσιακή προσέγγιση ενός μονολιθικού ατέρμονος βρόχου (super-loop) καθίσταται ανεπαρκής, καθώς οποιαδήποτε καθυστέρηση θα οδηγούσε σε κρίσιμη απώλεια ηχητικών δειγμάτων (frame dropping) [12] και αστοχία του εντοπισμού. Για τον λόγο αυτό, κρίθηκε απολύτως απαραίτητη η χρήση του Λειτουργικού Συστήματος Πραγματικού Χρόνου FreeRTOS [9]. Η υιοθέτησή του επέτρεψε τον διαχωρισμό του κώδικα σε αυτόνομα, διακριτά Tasks με αυστηρές προτεραιότητες [9], καθώς και την πλήρη εκμετάλλευση της διπύρνης (dual-core) αρχιτεκτονικής του συστήματος [4].

Ο υπολογιστικός φόρτος κατανεμήθηκε στρατηγικά: ο ένας πυρήνας αναλαμβάνει τη διαχείριση της μνήμης και τη δικτύωση στο παρασκήνιο [4], ενώ ο δεύτερος αφιερώνεται αποκλειστικά στην εκτέλεση των απαιτητικών αλγορίθμων εντοπισμού (GCC-PHAT) [2, 7] και ταξινόμησης (TinyML) [3]. Με αυτόν τον τρόπο, και με τη χρήση κατάλληλων μηχανισμών συγχρονισμού, διασφαλίζεται η αδιάλειπτη συλλογή ακουστικών δεδομένων, αποτρέπεται η απώλεια πληροφορίας και διατηρείται η απόλυτη σταθερότητα της συσκευής.



Εικόνα 35. Machine learning embedded, Πηγή: Αναζήτηση google

6.1 Περιβάλλον Ανάπτυξης

Η συγγραφή και η ανάπτυξη του firmware υλοποιήθηκαν αμιγώς σε γλώσσες C/C++, εξασφαλίζοντας τον βέλτιστο έλεγχο των πόρων του υλικού και τη μέγιστη δυνατή ταχύτητα εκτέλεσης. Για ένα έργο αυτής της κλίμακας, η χρήση βασικών εργαλείων όπως το παραδοσιακό Arduino IDE κρίθηκε ανεπαρκής, καθώς το συγκεκριμένο περιβάλλον στερείται των απαραίτητων δυνατοτήτων για τη διαχείριση πολύπλοκων συστημάτων πραγματικού χρόνου (RTOS) [9] και την ενσωμάτωση μεγάλων, εξειδικευμένων βιβλιοθηκών. Ως εκ τούτου, η ανάπτυξη μεταφέρθηκε στο επαγγελματικό περιβάλλον **PlatformIO**, το οποίο λειτουργεί ως μια ισχυρή, ενσωματωμένη επέκταση (extension) πάνω στο Visual Studio Code (VSCode).

Η επιλογή του PlatformIO βασίστηκε σε μια σειρά από στρατηγικά τεχνικά πλεονεκτήματα, τα οποία αναβαθμίζουν καθοριστικά τη διαδικασία ανάπτυξης του κώδικα:



Εικόνα 36.platformio ide picture, Πηγή:Αναζήτηση google

- **Προηγμένη Διαχείριση Βιβλιοθηκών και Εξαρτήσεων:** Σε αντίθεση με το παγκόσμιο (global) σύστημα του Arduino, το PlatformIO προσφέρει πλήρως απομονωμένα, ανεξάρτητα περιβάλλοντα μεταγλώττισης (compilation) για κάθε μεμονωμένο project. Αυτό εξασφαλίζει ότι οι συγκεκριμένες εκδόσεις των βιβλιοθηκών (όπως του περιβάλλοντος Edge Impulse [5] ή του συστήματος FATFS για την κάρτα MicroSD) παραμένουν σταθερές και δεν επηρεάζονται από εξωτερικές αλλαγές, εγγυώμενες την απόλυτη αναπαραγωγικότητα (reproducibility) του κώδικα.
- **Εργαλεία Αποσφαλμάτωσης (Debugging):** Το περιβάλλον παρέχει ισχυρά εργαλεία για δυναμική αποσφαλμάτωση σε πραγματικό χρόνο. Η δυνατότητα παρακολούθησης της κατάστασης της μνήμης και η χρήση σημείων διακοπής (breakpoints) είναι κρίσιμης σημασίας κατά τη διαχείριση δεικτών (pointers) στους Ring Buffers του ελεγκτή DMA, βοηθώντας στον εντοπισμό σφαλμάτων χρονισμού ή διαρροών μνήμης (memory leaks) κατά την εγγραφή και ανάγνωση από την εξωτερική PSRAM [4].

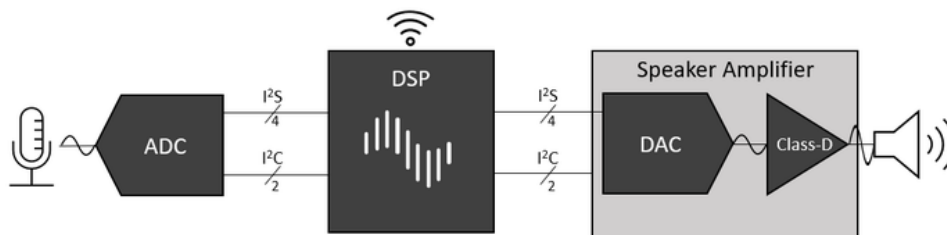
- **Βελτιστοποίηση Μεταγλωττιστή (Compiler Flags):** Ίσως το σημαντικότερο πλεονέκτημα είναι η δυνατότητα απόλυτου ελέγχου στις παραμέτρους του μεταγλωττιστή (GCC/G++). Μέσω του αρχικού αρχείου διαμόρφωσης, καθίσταται εφικτή η ενεργοποίηση ειδικών σημαιών μεταγλώττισης (build flags, όπως `-O2` ή `-O3` για μέγιστη ταχύτητα) και η ρητή εκμετάλλευση των διανυσματικών εντολών (Vector Instructions) που ενσωματώνει ο διπύρηνος επεξεργαστής Xtensa LX7 του ESP32-S3 [4]. Αυτή η υλικοσμική επιτάχυνση (hardware acceleration) βελτιώνει δραματικά τον χρόνο εκτέλεσης των απαιτητικών μαθηματικών πράξεων, οι οποίες είναι θεμελιώδεις για τον υπολογισμό του αλγορίθμου FFT [1] και την ταχύτατη εξαγωγή συμπερασμάτων (inference) από τα νευρωνικά δίκτυα του TinyML [3].

6.2 I2S Audio Acquisition

Η αδιάλειπτη λήψη ψηφιακού ήχου από τα μικρόφωνα INMP441 [8] αποτελεί το θεμέλιο της λειτουργίας του συστήματος. Ο οδηγός I2S [20] παραμετροποιήθηκε με στόχο τη συνεχή ροή δειγμάτων, ελαχιστοποιώντας τον φόρτο και τις διακοπές στην Κεντρική Μονάδα Επεξεργασίας.

Η συχνότητα δειγματοληψίας ορίστηκε στα 16 kHz, τιμή που αποτελεί τον "χρυσό κανόνα" για αλγορίθμους Ανίχνευσης Φωνητικής Δραστηριότητας (VAD) [16]. Βάσει του θεωρήματος Nyquist-Shannon [12], η συχνότητα αυτή επιτρέπει την πλήρη ανακατασκευή του ακουστικού φάσματος της ανθρώπινης ομιλίας (έως 8 kHz), παρέχοντας ικανοποιητικό εύρος ζώνης (bandwidth) χωρίς να προκαλεί υπερβολική και περιττή κατανάλωση της διαθέσιμης μνήμης RAM.

Το βάθος δειγματοληψίας (bit depth) ρυθμίστηκε αρχικά στα 32 bits ανά δείγμα. Η ρύθμιση αυτή είναι αρχιτεκτονικά επιβεβλημένη, καθώς εξασφαλίζει την τέλεια ευθυγράμμιση (alignment) με τους εσωτερικούς καταχωρητές του ελεγκτή DMA του ESP32-S3 [4], αποτρέποντας καθυστερήσεις κατά τη μεταφορά δεδομένων. Εν συνεχεία, κατά την προεπεξεργασία, τα ωμά δεδομένα υποβιβάζονται (μέσω bitwise shift) στα 16 ή 24 bits. Η τεχνική αυτή βελτιστοποιεί την αποθήκευση στους κυκλικούς καταχωρητές (Ring Buffers) και επιταχύνει σημαντικά την εκτέλεση των αλγορίθμων ψηφιακής επεξεργασίας σήματος (DSP) [1].



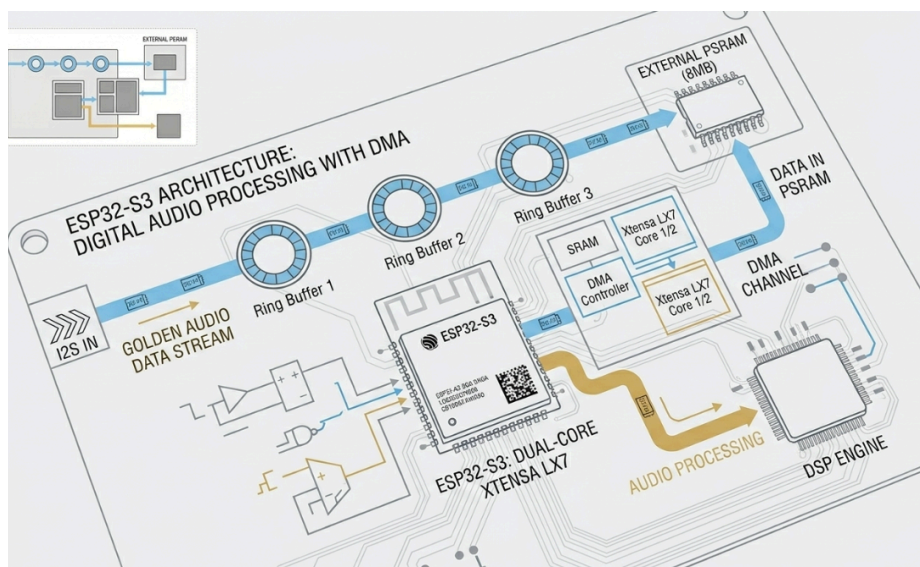
Εικόνα 37.protocol i2s explained, Πηγή:Αναζήτηση google

6.3 Audio Buffers

Δεδομένου ότι το ψηφιακό ακουστικό σήμα αποτελεί μια συνεχή ροή πληροφορίας (continuous stream), η επεξεργασία των δειγμάτων σε ατομικό επίπεδο (sample-by-sample) κρίνεται υπολογιστικά μη αποδοτική. Αντιθέτως, απαιτείται η ομαδοποίηση των δεδομένων και η επεξεργασία τους ως διακριτά πακέτα (chunks). Για την ασφαλή και ταχεία διαχείριση αυτής της ροής, υλοποιήθηκαν κυκλικοί καταχωρητές (Ring Buffers), οι οποίοι αποτρέπουν την υπερχείλιση (buffer overflow) και διασφαλίζουν τη συνεχή διαθεσιμότητα των δεδομένων.

Η διαδικασία μεταφοράς των ηχητικών δεδομένων στη μνήμη αυτοματοποιείται μέσω του ελεγκτή Άμεσης Πρόσβασης (DMA). Μόλις ο ελεγκτής DMA γεμίσει ένα προκαθορισμένο μπλοκ μνήμης με νέα ηχητικά δείγματα, πυροδοτεί άμεσα μια υλισμική διακοπή (hardware interrupt) [4]. Στη συνέχεια, μια Ρουτίνα Εξυπηρέτησης Διακοπής (Interrupt Service Routine - ISR) λαμβάνει τον έλεγχο και, κάνοντας χρήση ενός Σημαφόρου (RTOS Semaphore), ειδοποιεί την αντίστοιχη διεργασία επεξεργασίας ψηφιακού σήματος ότι τα δεδομένα είναι διαθέσιμα προς ανάγνωση [9]. Αυτός ο μηχανισμός συγχρονισμού απαλλάσσει τον κεντρικό επεξεργαστή από τη διαρκή και ενεργοβόρα παρακολούθηση (polling) των θυρών εισόδου.

Λόγω του μεγάλου όγκου των ηχητικών δεδομένων (ειδικά κατά τη λήψη από πολλαπλά μικρόφωνα ταυτόχρονα), οι απαιτούμενοι καταχωρητές υπερβαίνουν τη χωρητικότητα της εσωτερικής μνήμης SRAM. Ως εκ τούτου, οι μεγάλοι buffers δεσμεύονται δυναμικά στην εξωτερική μνήμη PSRAM του ESP32-S3 [4]. Η δέσμευση αυτή υλοποιείται μέσω της εξειδικευμένης συνάρτησης του ESP-IDF `heap_caps_malloc(size, MALLOC_CAP_SPIRAM)`, εγγυώμενη την επαρκή διάθεση πόρων για τη μετέπειτα εκτέλεση των αλγορίθμων εντοπισμού και μηχανικής μάθησης.



Εικόνα 38. Esp32s3 Audio processing, Πηγή: created in google flow

6.4 Αναμετάδοση Ήχου (Live Audio Streaming) μέσω WebSocket

Για την υποστήριξη της «ζωντανής» ακουστικής παρακολούθησης του περιβάλλοντος χώρου (Live Monitor), ο σχεδιασμός του συστήματος αναθεωρήθηκε, εγκαταλείποντας την αρχική λύση του πρωτοκόλλου UDP. Αντ' αυτού, υιοθετήθηκε μια αισθητά πιο αποδοτική και ενοποιημένη μέθοδος μετάδοσης μέσω binary WebSocket frames στην αποκλειστική θύρα 81. Η επιλογή αυτή εξασφαλίζει τη συνύπαρξη δεδομένων ελέγχου και ροής ήχου σε ένα ενιαίο, αξιόπιστο κανάλι αμφίδρομης επικοινωνίας, ελαχιστοποιώντας τον διαχειριστικό φόρτο (overhead) του δικτύου. Η διαδικασία της συνεχούς δειγματοληψίας πραγματοποιείται αποκλειστικά από το κεντρικό, κατευθυντικό αναλογικό μικρόφωνο MAX9814, του οποίου η έξοδος (OUT) δρομολογείται απευθείας στην αναλογική είσοδο ADC1 (στο GPIO 1) του μικροελεγκτή ESP32-S3.

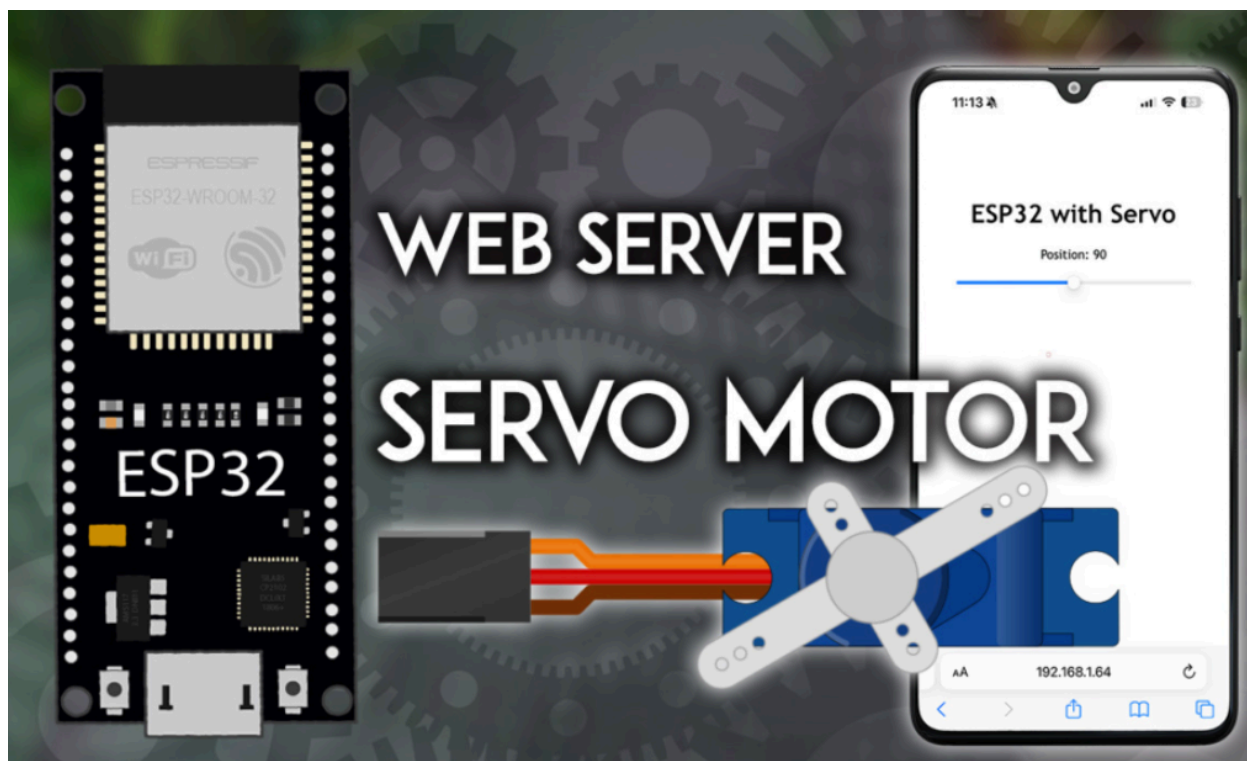
Για την αποφυγή του φαινομένου της χρονικής αστάθειας (jitter) που θα υποβάθμιζε ραγδαία την ποιότητα του ήχου, επιστρατεύτηκε ένας εξειδικευμένος χρονιστής υλικού (hardware timer). Ο χρονιστής αυτός, ρυθμισμένος με απόλυτη ακρίβεια στα 125 μ s, εγγυάται μια αυστηρά σταθερή συχνότητα δειγματοληψίας της τάξης των 8 kHz, η οποία κρίνεται ιδανική για την καθαρή αναπαραγωγή της ανθρώπινης ομιλίας. Αμέσως μετά τη μετατροπή τους, τα ακατέργαστα ψηφιακά δείγματα αποθηκεύονται προσωρινά σε έναν κυκλικό καταχωρητή (Ring Buffer) χωρητικότητας 512 θέσεων, μια δομή που αποτρέπει την υπερχείλιση και την απώλεια δεδομένων σε περίπτωση στιγμιαίων καθυστερήσεων.

Στη συνέχεια, τα δεδομένα αντλούνται και επεξεργάζονται σε πραγματικό χρόνο από την αποκλειστική διεργασία ήχου (audio task). Στο στάδιο αυτό της ψηφιακής προεπεξεργασίας, εφαρμόζεται ένα ψηφιακό φίλτρο άπειρης κρουστικής απόκρισης (IIR DC blocker). Ο ρόλος του φίλτρου είναι θεμελιώδης, καθώς αναλαμβάνει την πλήρη εξάλειψη της συνεχούς συνιστώσας (DC offset) που εγγενώς εισάγει ο αναλογικός αισθητήρας, κεντράροντας την κυματομορφή γύρω από το μηδέν. Παράλληλα, εφαρμόζεται ένα λογισμικό κέρδος (digital gain x2), το οποίο αναδεικνύει και ενισχύει τις δυναμικές της φωνής. Το τελικό, ψηφιοποιημένο και φιλτραρισμένο σήμα, πλέον σε τυποποιημένη μορφή ασυμπίεστου PCM (int16 Little Endian), πακετάρεται αποδοτικά και αποστέλλεται άμεσα μέσω της ανοιχτής σύνδεσης WebSocket προς την εφαρμογή διεπαφής. Προκειμένου να διασφαλιστεί η αδιάλειπτη ροή και να αποφευχθεί το φαινόμενο της πτώσης καρτέ (frame dropping), η λειτουργία της δειγματοληψίας έχει ενσωματωθεί στο περιβάλλον του FreeRTOS με αυξημένη προτεραιότητα εκτέλεσης (priority 3). Επιπλέον, η διεργασία δεσμεύεται αποκλειστικά στον Πυρήνα 0 (Core 0), επιτρέποντας στον Πυρήνα 1 να διαχειρίζεται ανεμπόδιστα τους βαρύτερους αλγορίθμους στόχευσης. Αυτή η αυστηρή αρχιτεκτονική χρονοπρογραμματισμού διασφαλίζει μια κρυστάλλινη ακουστική εμπειρία, χαρακτηριζόμενη από πρακτικά μηδενική καθυστέρηση (ultra-low latency).

6.5 Έλεγχος Σερβοκινητήρα

Η φυσική στόχευση των μικροφώνων απαιτεί ιδιαίτερο χειρισμό, καθώς το πρωτότυπο εξοπλίστηκε με σερβοκινητήρα συνεχούς περιστροφής (continuous rotation). Αντί για τη χρήση του τυπικού περιφερειακού LEDC, ο παλμός ελέγχου 50 Hz παράγεται αποκλειστικά μέσω λογισμικού (bit-banging) στο GPIO 5, προκειμένου να υπάρχει απόλυτος έλεγχος των μικροδευτερολέπτων (μs) και αδιατάρακτη λειτουργία από τον Πυρήνα 1 (Core 1) του ESP32-S3.

Η μετάβαση στην επιθυμητή γωνία γίνεται υπολογίζοντας τον απαιτούμενο χρόνο κίνησης (timed movement). Το firmware του συστήματος περιλαμβάνει μια προηγμένη ρουτίνα βαθμονόμησης (calibration process), όπου ο κινητήρας εκτελεί αυτόματα τρεις πλήρεις περιστροφές ($0^\circ \rightarrow 360^\circ$ και αντίστροφα), καταγράφοντας τους ακριβείς χρόνους που απαιτούνται για τη διάνυση κάθε επιμέρους τόξου 90 μοιρών. Ο συνολικός χρόνος κίνησης για την προσέγγιση μιας νέας γωνίας-στόχου προκύπτει από το άθροισμα των αντίστοιχων καταγεγραμμένων τόξων, με γραμμική παρεμβολή για τον υπολογισμό των ενδιάμεσων τιμών. Οι παράμετροι αυτής της βαθμονόμησης (ταχύτητα, χρόνοι τόξων), καθώς και η τρέχουσα θέση προσανατολισμού της συσκευής (heading), αποθηκεύονται μόνιμα στο ενσωματωμένο σύστημα αρχείων LittleFS του μικροελεγκτή. Αυτό εξασφαλίζει τη συνέχεια και την ακρίβεια της λειτουργίας της διάταξης ακόμη και έπειτα από διακοπή τροφοδοσίας ή επανεκκίνηση.



Εικόνα 39. Esp32 servo control, Πηγή: Αναζήτηση google

6.6 FreeRTOS Tasks

Για την αποφυγή συγκρούσεων (blocking) κατά την εκτέλεση του κώδικα, το σύστημα χωρίστηκε σε πολλαπλά Tasks του FreeRTOS [9]. Η δομή αυτή επιτρέπει την ψευδο-παράλληλη (στο Core 0) και την πραγματικά παράλληλη (στα Cores 0 & 1) εκτέλεση κώδικα [4, 9]. Η διαχείριση πόρων σε ένα ενσωματωμένο σύστημα, το οποίο καλείται να επεξεργαστεί σήματα σε πραγματικό χρόνο, απαιτεί αυστηρό χρονοπρογραμματισμό και πλήρη απομόνωση των διεργασιών. Αναθέτοντας διαφορετικές προτεραιότητες σε κάθε Task, εξασφαλίζεται ότι οι πλέον κρίσιμες λειτουργίες εξυπηρετούνται άμεσα.

1. **Διεργασία Λήψης Ήχου (Audio Capture Task):** Πρόκειται για εξαιρετικά κρίσιμη λειτουργία. Δεσμεύεται (pinned) αποκλειστικά στον Πυρήνα 0 και της αποδίδεται υψηλή προτεραιότητα. Η διεργασία χρησιμοποιεί άμεσα τον ADC1 χωρίς να περιμένει άλλες διεπαφές. Για την αποτροπή επανεκκινήσεων από υπερφόρτωση, ο Watchdog Timer του Πυρήνα 0 απενεργοποιείται τοπικά. Τα δεδομένα προωθούνται μέσω ουράς (Queue) στο επόμενο στάδιο.
2. **Διεργασία Επεξεργασίας Σήματος (SSL / DSP Task)** Ο πυρήνας της υπολογιστικής νοημοσύνης του συστήματος εκτελείται εδώ. Η διεργασία δεσμεύεται στον Πυρήνα 1 [4], ώστε να μην παρεμβάλλεται στις λειτουργίες δικτύωσης και λήψης δεδομένων. Η ρουτίνα αυτή παρακολουθεί τους buffers και ενεργοποιείται μόλις συγκεντρωθεί ο απαραίτητος αριθμός δειγμάτων (π.χ. ένα παράθυρο 1024 ή 2048 δειγμάτων). Είναι υπεύθυνη για την εκτέλεση των απαιτητικών μαθηματικών υπολογισμών, συμπεριλαμβανομένου του Μετασχηματισμού Fourier (FFT), του αλγορίθμου GCC-PHAT και της παραβολικής παρεμβολής [1, 2]. Στο τέλος του κύκλου της, εξάγει την υπολογισμένη γωνία άφιξης του ήχου και αποστέλλει την αντίστοιχη εντολή κίνησης στον σερβοκινητήρα.
3. **Διεργασία Δικτύωσης και Τηλεμετρίας (Network Task)** Αυτή η διεργασία διαχειρίζεται το σύνολο της ασύρματης επικοινωνίας και εκτελείται στον Πυρήνα 0 με μεσαία προτεραιότητα. Ο διαχωρισμός αυτός είναι στρατηγικός, καθώς η στοίβα πρωτοκόλλων του LwIP τρέχει εγγενώς στον Πυρήνα 0. Το Network Task επιβλέπει τη σταθερότητα της σύνδεσης Wi-Fi στο SoftAP (με το ενσωματωμένο σύστημα Captive DNS στη θύρα 53), διαχειρίζεται το πρωτόκολλο HTTP (θύρα 80) για το Web UI και την αναπαραγωγή των εγγραφών WAV, και ενορχηστρώνει την αμφίδρομη επικοινωνία μέσω WebSocket (θύρα 81) για τη λήψη JSON εντολών και την αποστολή της ζωντανής ροής του ήχου (binary PCM), φροντίζοντας να μην προκαλεί εμπλοκή (blocking) στη δειγματοληψία.
4. **Διεργασία Ασύγχρονης Εγγραφής Δεδομένων (SD Recording Task):** Σε αντίθεση με παλαιότερες προσεγγίσεις που μπλόκαραν τη δειγματοληψία κατά την εγγραφή στην SD κάρτα (κόβοντας δείγματα και δημιουργώντας "εξωγήινες" φωνητικές παραμορφώσεις), η εγγραφή υλοποιήθηκε πλήρως ασύγχρονα. Ένα νέο Task με προτεραιότητα 2 παραλαμβάνει τα πακέτα από την ουρά του FreeRTOS και γεμίζει έναν μεγάλο κυκλικό καταχωρητή (Ring Buffer 2048 θέσεων). Από εκεί, τα γράφει στην κάρτα SD. Ο μετατροπέας ADC συνεχίζει τη δειγματοληψία του στο παρασκήνιο χωρίς ποτέ να περιμένει το αργό πρωτόκολλο SPI της κάρτας SD.

6.7 Καταγραφή Δεδομένων και Τηλεμετρία (Data Logging)

Κατά τη διάρκεια της συνεχούς λειτουργίας του, το σύστημα εκτελεί αδιάλειπτη καταγραφή (logging) των κρίσιμων λειτουργικών του παραμέτρων και των εξαγόμενων αποτελεσμάτων. Η διαδικασία αυτή εξυπηρετεί έναν διπτό, στρατηγικό σκοπό: αφενός επιτρέπει την άμεση αποσφαλμάτωση (debugging) και την παρακολούθηση της «υγείας» του υλικού σε πραγματικό χρόνο, και αφετέρου εξασφαλίζει τη συλλογή απαραίτητων μετρικών για την ποσοτική και ποιοτική ακαδημαϊκή αξιολόγηση της συσκευής.

Η εκπομπή των δεδομένων τηλεμετρίας υλοποιείται σε δύο επίπεδα. Σε πρώτο χρόνο, πληροφορίες όπως τα αποτελέσματα των αλγορίθμων TDOA [2], οι χρόνοι εκτέλεσης (execution times) του Μετασχηματισμού Fourier (FFT) [1], καθώς και μετρικές κατανάλωσης της εσωτερικής μνήμης RAM, εκτυπώνονται σειριακά. Αυτό παρέχει στον προγραμματιστή μια ζωντανή εικόνα (live feed) της συμπεριφοράς των νημάτων (Tasks) του FreeRTOS και βοηθά στον εντοπισμό πιθανών σημείων συμφόρησης (bottlenecks) ή διαρροών μνήμης (memory leaks).

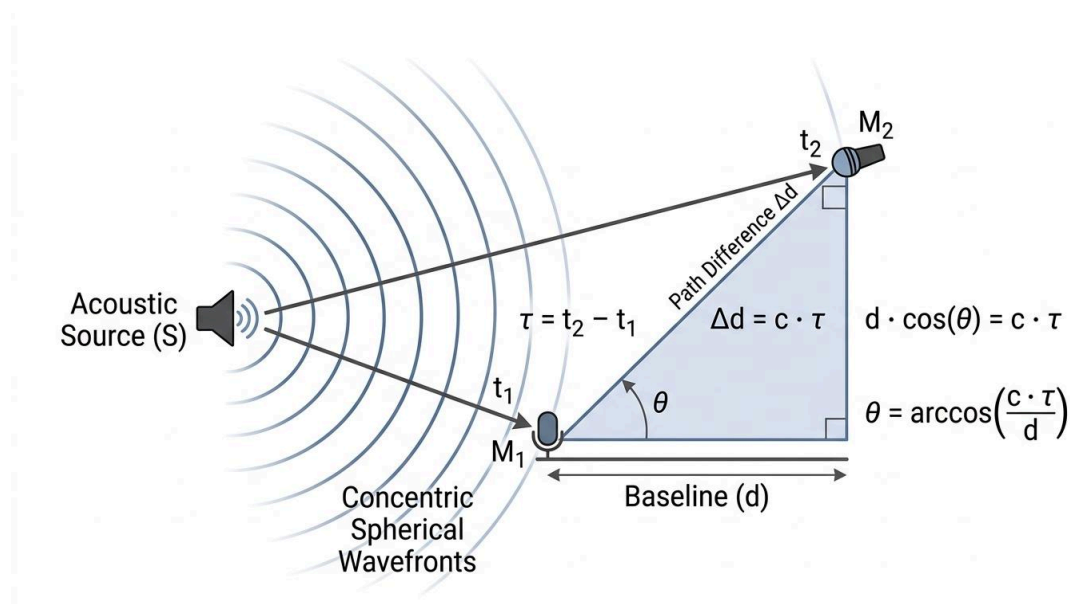
Σε δεύτερο χρόνο, και προκειμένου να διασφαλιστεί η διατήρηση των δεδομένων (data persistence) για μελλοντική ανάλυση, το σύστημα καταγράφει παράλληλα ένα λεπτομερές ιστορικό σε ένα αρχείο κειμένου (`log.txt`) το οποίο αποθηκεύεται στην εξωτερική κάρτα MicroSD [4]. Η διαχείριση της μνήμης flash υλοποιείται μέσω του καθιερωμένου συστήματος αρχείων FATFS. Οι ρουτίνες εγγραφής έχουν προγραμματιστεί ώστε να ανοίγουν το αρχείο σε λειτουργία προσθήκης (append mode), προσαρτώντας τις νέες καταχωρήσεις στο τέλος του αρχείου, ώστε να μην αντικαθίσταται το προηγούμενο ιστορικό [4]. Μέσω αυτής της πρακτικής, καταγράφονται σε μόνιμη βάση τα σφάλματα συστήματος (error handling logs), οι αλλαγές καταστάσεων λειτουργίας (state transitions) και οι ανιχνεύσεις ομιλίας από το μοντέλο TinyML [3], δημιουργώντας ένα πολύτιμο σύνολο δεδομένων (dataset) για τη μετέπειτα εξαγωγή στατιστικών συμπερασμάτων σχετικά με την αξιοπιστία και την ακρίβεια του κατευθυντικού μικροφώνου.

ΚΕΦΑΛΑΙΟ 7: SOUND SOURCE LOCALIZATION

Το Κεφάλαιο 6 συνιστά τον μαθηματικό και επιστημονικό πυρήνα της παρούσας πτυχιακής. Ο εντοπισμός της ηχητικής πηγής (Sound Source Localization - SSL) αποτελεί τη διαδικασία μέσω της οποίας οι θεωρητικές αρχές του Κεφαλαίου 2 μετατρέπονται σε συγκεκριμένο, εκτελέσιμο αλγόριθμο στον ESP32-S3. Η μετάβαση από τη θεωρία στην πράξη προϋποθέτει την υπέρβαση σημαντικών υπολογιστικών προκλήσεων, όπως η επεξεργασία των ακουστικών σημάτων σε πραγματικό χρόνο και η αποτελεσματική αντιμετώπιση του θορύβου και των ανακλάσεων του περιβάλλοντος. Μέσα από την αναλυτική παρουσίαση της μεθόδου GCC-PHAT και του υπολογισμού της Χρονικής Διαφοράς Άφιξης (TDOA), το κεφάλαιο αυτό εξηγεί τον μηχανισμό με τον οποίο η συσκευή αποκτά χωρική αντίληψη, καταλήγοντας στη μετάφραση των ψηφιακών δεδομένων σε φυσική, στοχευμένη κίνηση του μικροφώνου.

7.1 Θεωρία TDOA (Time Difference Of Arrival) - Εμβάθυνση

Ο υπολογισμός TDOA δεν μετρά τον απόλυτο χρόνο πτήσης του ήχου (time of flight) από την πηγή στο μικρόφωνο, διότι το σύστημα δεν γνωρίζει τη χρονική στιγμή εκπομπής του αρχικού ήχου (όπως γίνεται π.χ. στο GPS ή τα ραντάρ). Αντίθετα, μετρά τη σχετική διαφορά του χρόνου άφιξης του ίδιου ακριβώς ηχητικού μετώπου σε δύο διαφορετικά μικρόφωνα. Αν το κύμα φτάσει στο μικρόφωνο M1 τη στιγμή t_1 και στο μικρόφωνο M2 τη στιγμή t_2 , το σύστημα προσπαθεί να μετρήσει την ποσότητα $\Delta t = t_2 - t_1$. Λόγω του ότι η απόσταση των μικροφώνων είναι γνωστή και σταθερή, η χρονική διαφορά περιέχει όλη την πληροφορία για τη γωνία (Αζιμούθιο).

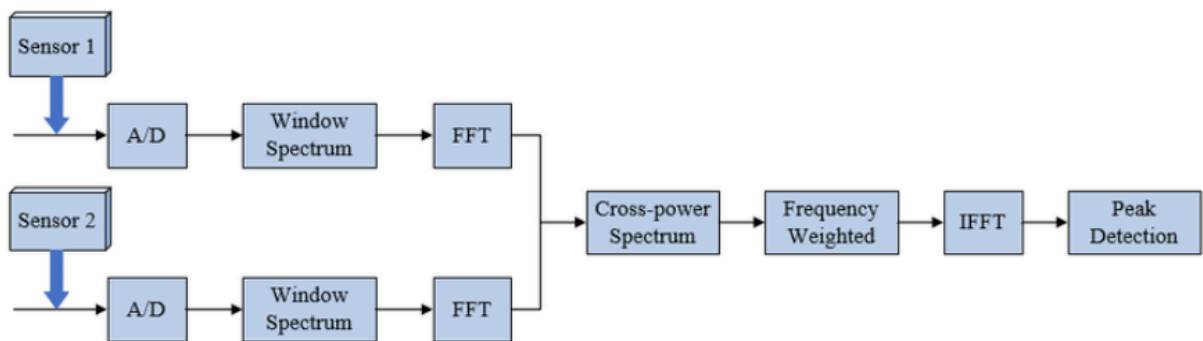


Εικόνα 40. Time difference of arrival, Πηγή: created in google flow

7.2 Ο Αλγόριθμος GCC-PHAT (Με εξισώσεις)

Σε ένα ιδανικό, ανηχοϊκό περιβάλλον (anechoic chamber), η μέθοδο του TDOA θα μπορούσε να εξαχθεί με μια απλή ετεροσυσχέτιση (Cross-Correlation) [1]. Όμως, ο χώρος έχει ανακλάσεις. Η Γενικευμένη Ετεροσυσχέτιση με Μετασχηματισμό Φάσης (GCC-PHAT) επιλύει αυτό το πρόβλημα [2, 7]. Ο αλγόριθμος, όπως υλοποιείται στον κώδικα C/C++, ακολουθεί τα εξής βήματα: Έστω $x_1[n]$ και $x_2[n]$ τα ψηφιακά σήματα (πίνακες μεγέθους N, όπου $N = 1024$ δείγματα) από δύο μικρόφωνα. Εφαρμόζεται συνάρτηση παραθύρου (Hanning window) για την αποφυγή του spectral leakage [1, 12].

1. **Μετάβαση στο Πεδίο των Συχνοτήτων:** Εφαρμόζεται FFT στα δύο σήματα, προκύπτοντας τα μιγαδικά φάσματα $X_1[k]$ και $X_2[k]$.
2. **Φάσμα Ετερο-Ισχύος (Cross-Power Spectrum):** Υπολογίζεται το φάσμα ετεροϊσχύος $G_{12}[k] = X_1[k] \cdot X_2^*[k]$, όπου $*$ δηλώνει τον συζυγή μιγαδικό.
3. **Μετασχηματισμός PHAT (Phase Transform):** Το φάσμα $G_{12}[k]$ διαιρείται με το μέτρο του, ώστε κάθε συχνότητα να έχει πλάτος 1 και να παραμένει μόνο η πληροφορία της φάσης. Η εξίσωση του PHAT είναι: $W_{PHAT}[k] = 1/|G_{12}[k]|$. Το σταθμισμένο φάσμα είναι: $R[k] = G_{12}[k] \cdot W_{PHAT}[k] = (X_1[k] \cdot X_2^*[k])/|X_1[k] \cdot X_2^*[k]|$. Στον κώδικα, για την αποφυγή διαίρεσης με το μηδέν, προστίθεται μια πολύ μικρή σταθερά ϵ στον παρονομαστή.
4. **Επιστροφή στο Πεδίο του Χρόνου:** Εφαρμόζεται Αντίστροφος FFT (IFFT) στον μιγαδικό πίνακα $R[k]$. Προκύπτει ο πίνακας της συσχέτισης στο πεδίο του χρόνου, $r_{12}[\tau]$.
5. **Εύρεση Κορυφής (Peak Detection):** Εντοπίζεται ο δείκτης (index) τ_{max} του πίνακα r_{12} όπου η τιμή είναι μέγιστη. Αυτός ο δείκτης αντιπροσωπεύει τη χρονική καθυστέρηση (σε δείγματα): $\Delta t = \tau_{max} \cdot (1/fs)$.



Εικόνα 41. GCC-PHAT algorithm, Πηγή:Αναζήτηση google

7.3 Υπολογισμός Γωνίας (θ)

Έχοντας βρει τη χρονική καθυστέρηση Δt μέσω του αλγορίθμου GCC-PHAT, προχωράμε στον γεωμετρικό υπολογισμό της γωνίας θ . Υποθέτουμε ότι η ηχητική πηγή βρίσκεται σε απόσταση (far-field), οπότε τα ηχητικά κύματα φτάνουν ως επίπεδα και όχι ως σφαιρικά μέτωπα. Με βάση την ταχύτητα του ήχου c (343 m/s) και την απόσταση d μεταξύ των μικροφώνων (π.χ. 0.04 μέτρα):

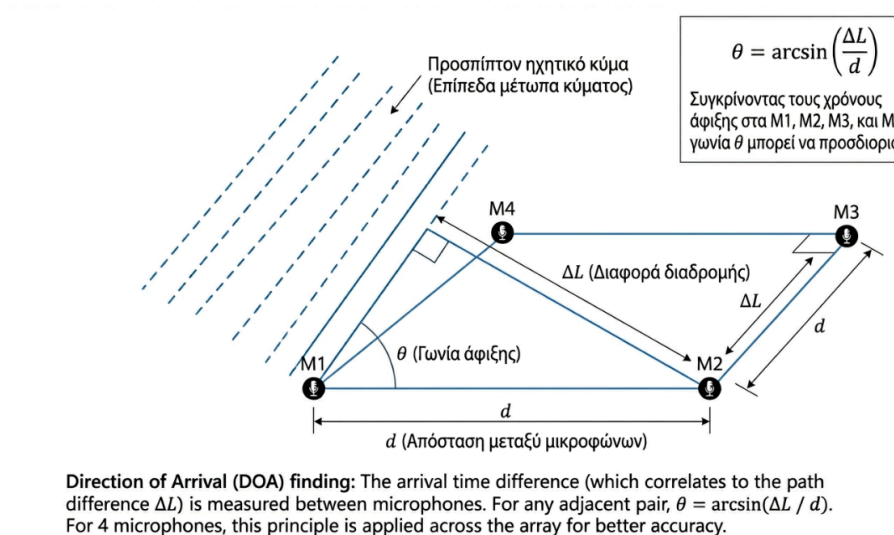
Η εξίσωση που συνδέει τη χρονική διαφορά Δt με τη γωνία θ είναι:

$$c \cdot \Delta t = d \cdot \sin(\theta) \quad [7.3]$$

Λύνοντας ως προς τη γωνία θ , λαμβάνουμε:

$$\theta = \sin^{-1}((c \cdot \Delta t)/d) \quad [7.3]$$

Η συνάρτηση $\arcsin$ επιστρέφει το αποτέλεσμα σε ακτίνια (radians), το οποίο πολλαπλασιάζεται με $180 / \pi$ για να μετατραπεί σε μοίρες. Η παραπάνω εξίσωση είναι ο βασικός κινητήριος μοχλός για την τοποθέτηση του μικροφώνου.



Εικόνα 42. Direction of Arrival finding, Πηγή:created in google flow

7.4 Υβριδικός Αλγόριθμος Εντοπισμού (GCC-PHAT & Energy Excess)

Κατά την πρακτική υλοποίηση της διάταξης στον μικροελεγκτή ESP32-S3, προέκυψε ένας σημαντικός υλισμικός περιορισμός (hardware constraint) ως προς τον χρονισμό των αισθητήρων. Η ταυτόχρονη λήψη δεδομένων από τα τέσσερα ψηφιακά μικρόφωνα επέβαλε τη χρήση δύο ξεχωριστών περιφερειακών I2S (I2S0 για το μπροστινό ζεύγος, I2S1 για το πισινό ζεύγος). Καθώς τα δύο αυτά κυκλώματα δεν εκκινούν με απόλυτο συγχρονισμό επιπέδου δείγματος, εισάγεται μια σταθερή, παρασιτική χρονική ολίσθηση (sample skew) μεταξύ του μπροστινού και του πισινού άξονα. Το φαινόμενο αυτό καθιστά τον αμιγή δισδιάστατο (2D) υπολογισμό TDOA μαθηματικά ασταθή.

Για την επίλυση αυτού του προβλήματος, αναπτύχθηκε ένας προηγμένος Υβριδικός Αλγόριθμος (Hybrid SSL), ο οποίος συνδυάζει τα πλεονεκτήματα του μετασχηματισμού φάσης με την ανάλυση πλάτους:

1. **Άξονας Χ (Αριστερά/Δεξιά):** Επειδή τα μικρόφωνα κάθε ζεύγους μοιράζονται τον ίδιο δίαυλο I2S, ο συγχρονισμός τους παραμένει απόλυτος. Εδώ εφαρμόζεται πλήρως ο αλγόριθμος **GCC-PHAT** (με παράθυρο FFT 1024 δειγμάτων μέσω της βιβλιοθήκης esp-dsp) για τον ακριβή υπολογισμό της χρονικής διαφοράς (TDOA) και τη στιβαρή καταστολή της αντήχησης.
2. **Άξονας Υ (Εμπρός/Πίσω) και Διακριτοποίηση 8 Τομέων:** Για τον καθορισμό της τελικής κατεύθυνσης και την αντιστάθμιση του σφάλματος χρονισμού, το σύστημα συνδυάζει τα δεδομένα του GCC-PHAT με έναν αλγόριθμο ανάλυσης Πλεονάζουσας Ακουστικής Ενέργειας (Energy Excess). Ο αλγόριθμος υπολογίζει συνεχώς τη μέση ενέργεια κάθε μικροφώνου και αφαιρεί το δυναμικό κατώφλι θορύβου (noise floor) του χώρου.

Μέσω αυτής της σύντηξης δεδομένων (sensor fusion), το σύστημα κατορθώνει να συσχετίζει την αξιοπιστία φάσης του GCC-PHAT με την αξιοπιστία πλάτους της Ενέργειας. Η διαδικασία καταλήγει σε έναν εκθετικό κινητό μέσο όρο (EMA) και ένα πλειοψηφικό φίλτρο (voting system) εντός παραθύρου 200ms, εξάγοντας με απόλυτη ασφάλεια τον τελικό τομέα στόχευσης (π.χ. Μπροστά-Δεξιά) προς τον σερβοκινητήρα, αναιρώντας πλήρως τους περιορισμούς του υλικού.

7.5 Παραδείγματα (0°, 45°, 90°, 135°, 180°)

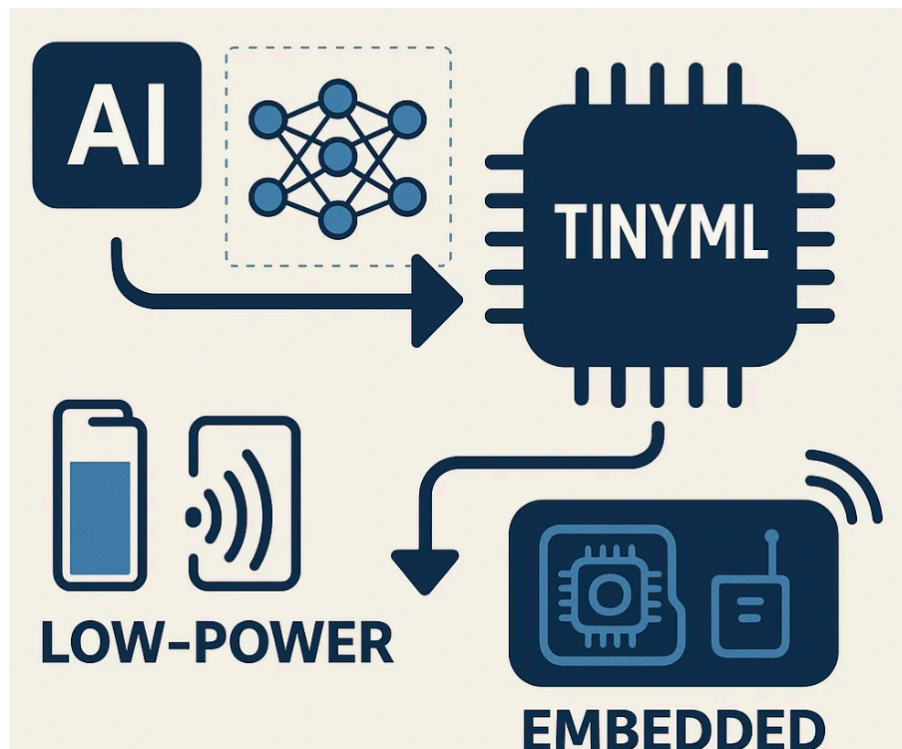
Για την καλύτερη κατανόηση της συσχέτισης μεταξύ της φυσικής γωνίας πρόσπτωσης του ηχητικού κύματος και της μετρούμενης χρονικής καθυστέρησης Δt , παρατίθενται τα παρακάτω θεωρητικά και πρακτικά παραδείγματα. Θεωρούμε ως δεδομένα την απόσταση των μικροφώνων $d = 0.04 \text{ m}$, την ταχύτητα του ήχου $c = 343 \text{ m/s}$ και τη συχνότητα δειγματοληψίας $f_s = 16000 \text{ Hz}$, όπου η διάρκεια ενός ψηφιακού δείγματος αντιστοιχεί σε $62.5 \mu\text{s}$. Η μέγιστη θεωρητική καθυστέρηση μεταξύ δύο μικροφώνων στη διάταξή μας είναι

$$\Delta t_{max} = d/c \approx 116 \mu s \text{ (κάτι λιγότερο από 2 πλήρη δείγματα). [7.5]}$$

- **0° (Ακριβώς Μπροστά - Front):** Το ηχητικό μέτωπο προσπίπτει ταυτόχρονα στα μικρόφωνα του αριστερού και του δεξιού άξονα. Η χρονική καθυστέρηση είναι $\Delta t = 0$. Η συνάρτηση συσχέτισης GCC-PHAT εμφανίζει την κύρια κορυφή της (peak) ακριβώς στο κέντρο του παραθύρου αναζήτησης (μηδενική μετατόπιση). Σε συνδυασμό με την υψηλότερη ενέργεια στο μπροστινό ζεύγος, ο υβριδικός αλγόριθμος ταξινομεί την πηγή στον τομέα "Front" και ο κινητήρας λαμβάνει εντολή να στραφεί στις 0°.
- **45° (Μπροστά Δεξιά - Front Right):** Ο ήχος φτάνει πρώτα στο δεξί μικρόφωνο. Η χρονική καθυστέρηση υπολογίζεται θεωρητικά στα $\Delta t = (0.04 \cdot \sin(45^\circ))/343 \approx 82.4 \mu s$. Αυτό αντιστοιχεί σε χρονική μετατόπιση περίπου 1.3 δειγμάτων. Η παραβολική παρεμβολή (interpolation) του αλγορίθμου εντοπίζει αυτή την υπο-δειγματική (sub-sample) διαφορά, το σύστημα ταξινομεί τον ήχο στον τομέα "Front Right" και η διάταξη περιστρέφεται στις 45°.
- **90° (Εντελώς Δεξιά - Right):** Το κύμα προσπίπτει κάθετα στον δεξιό άξονα. Εδώ παρατηρείται η μέγιστη δυνατή χρονική καθυστέρηση $\Delta t \approx 116 \mu s$. Η κορυφή του GCC-PHAT μετατοπίζεται στο δεξί άκρο της επιτρεπτής κλίμακας καθυστέρησης. Το σύστημα αναγνωρίζει τον τομέα "Right" και ο κινητήρας στοχεύει στις 90°.
- **180° (Ακριβώς Πίσω - Back):** Όπως και στις 0°, ο ήχος φτάνει ταυτόχρονα στον αριστερό και δεξιό άξονα, μηδενίζοντας το Δt του GCC-PHAT. Ωστόσο, η ανάλυση της πλεονάζουσας ενέργειας (energy excess) υποδεικνύει καθαρή υπεροχή στο πισινό ζεύγος μικροφώνων. Ο αλγόριθμος αίρει τη μαθηματική αμφισημία (front-back ambiguity), ταξινομεί το γεγονός στον τομέα "Back" και ο κινητήρας εκμεταλλεύεται την ικανότητα συνεχούς περιστροφής του για να στραφεί στις 180°.
- **270° (Εντελώς Αριστερά - Left):** Το ηχητικό μέτωπο φτάνει πρώτα στα αριστερά μικρόφωνα, δημιουργώντας αρνητική χρονική καθυστέρηση $\Delta t \approx -116 \mu s$. Η κορυφή της ετεροσυσχέτισης μετατοπίζεται συμμετρικά στο αριστερό άκρο. Η απόφαση κλειδώνει στον τομέα "Left" και η διάταξη περιστρέφεται στις 270°.

ΚΕΦΑΛΑΙΟ 8: TINY ML ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΗΧΟΥ

Η ένταξη της Τεχνητής Νοημοσύνης (AI) σε συσκευές με περιορισμένους πόρους, όπως ο μικροελεγκτής ESP32-S3, αποτελεί μία από τις σημαντικότερες καινοτομίες της σύγχρονης μηχανικής. Αυτή η μετάβαση από το υπολογιστικό νέφος (Cloud Computing) στο άκρο του δικτύου (Edge AI) προσφέρει κρίσιμα πλεονεκτήματα στο σύστημα, όπως η δραστική μείωση της καθυστέρησης απόκρισης (latency), η εξοικονόμηση ενέργειας και η προστασία της ιδιωτικότητας, αφού τα ηχητικά δεδομένα δεν αποστέλλονται σε εξωτερικούς διακομιστές. Στο παρόν κεφάλαιο αναλύεται η διαδικασία εκπαίδευσης και ενσωμάτωσης ενός βελτιστοποιημένου μοντέλου μηχανικής μάθησης (TinyML), το οποίο είναι ικανό να ταξινομεί ακουστικά γεγονότα σε πραγματικό χρόνο. Ειδικότερα, περιγράφεται συνοπτικά η ροή εργασίας, από τη δημιουργία του κατάλληλου συνόλου δεδομένων (dataset) και την εξαγωγή των χαρακτηριστικών του ήχου, έως την ανάπτυξη ενός ελαφριού Συνελικτικού Νευρωνικού Δικτύου (CNN), το οποίο προσδίδει στο σύστημα την ικανότητα να διαχωρίζει με ακρίβεια την ανθρώπινη ομιλία από τον θόρυβο του περιβάλλοντος.



Εικόνα 43. TinyML & Embedded systems, Πηγή: created in google flow

8.1 Μηχανική Μάθηση στα Άκρα του Δικτύου (Edge AI / TinyML)

Ιστορικά, η υλοποίηση αλγορίθμων τεχνητής νοημοσύνης για την αναγνώριση φωνής ή την ταξινόμηση ακουστικών σημάτων βασιζόταν αποκλειστικά σε αρχιτεκτονικές υπολογιστικού νέφους (Cloud Computing). Σύμφωνα με αυτό το πρότυπο, οι συσκευές συλλογής λειτουργούσαν ως απλά τερματικά, τα οποία αποστέλλουν τα ακατέργαστα ηχητικά δεδομένα σε απομακρυσμένους διακομιστές προς επεξεργασία [3]. Μολονότι αυτή η προσέγγιση προσφέρει θεωρητικά απεριόριστη υπολογιστική ισχύ, συνοδεύεται από κρίσιμους περιορισμούς για εφαρμογές πραγματικού χρόνου. Η διαρκής διαβίβαση δεδομένων εισάγει απρόβλεπτες καθυστερήσεις δικτύου (latency), καταναλώνει σημαντικό εύρος ζώνης (bandwidth) και εγείρει σοβαρά ζητήματα ασφάλειας και ιδιωτικότητας, καθώς οι ακουστικές καταγραφές εγκαταλείπουν τον φυσικό χώρο του χρήστη [3].

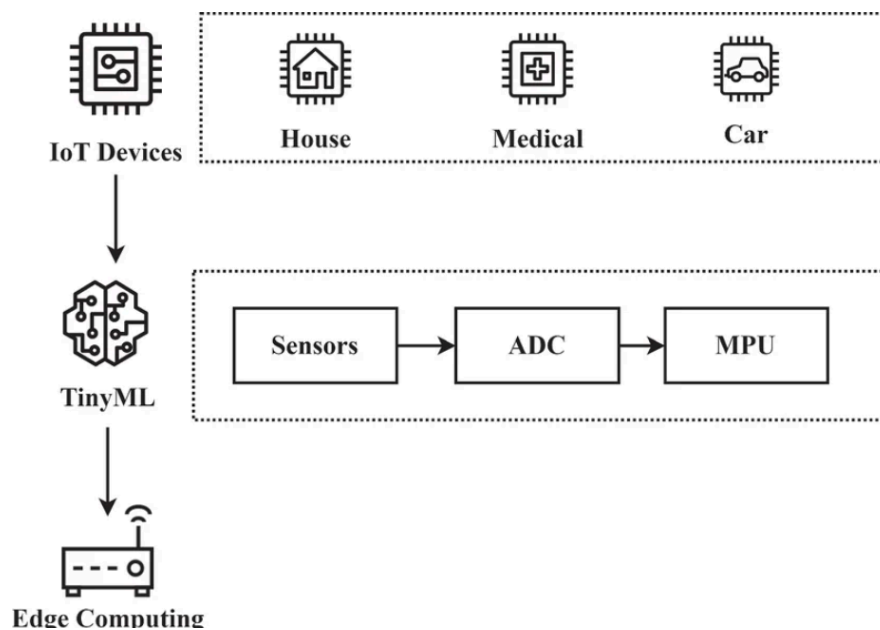
Για την πλήρη υπέρβαση αυτών των εμποδίων, το παρόν σύστημα υιοθετεί το παράδειγμα της Μηχανικής Μάθησης στα Άκρα (Edge Machine Learning), μεταφέροντας ολόκληρη τη διαδικασία του συμπερασμού (inference) τοπικά, απευθείας πάνω στη συσκευή (on-device) [3]. Εκμεταλλευόμενο την προηγμένη αρχιτεκτονική του μικροελεγκτή ESP32-S3, το εκπαιδευμένο νευρωνικό δίκτυο εκτελείται αυτόνομα χωρίς καμία εξάρτηση από εξωτερικούς εξυπηρετητές [4]. Η μετάβαση αυτή επιτρέπει την εξαγωγή αποτελεσμάτων σε ελάχιστα χιλιοστά του δευτερολέπτου, διασφαλίζοντας την ακαριαία απόκριση της συσκευής κατά τη χρήση της «Έξυπνης Λειτουργίας» (Smart Mode) [4]. Παράλληλα, προστατεύεται απόλυτα η ιδιωτικότητα, καθώς τα ηχητικά δείγματα αναλύονται στιγμιαία στη μνήμη του συστήματος και έπειτα απορρίπτονται, χωρίς να μεταδίδονται στο διαδίκτυο.

8.2 Μικρο-Μηχανική Μάθηση (TinyML) και Βελτιστοποίηση Μοντέλων

Ο τομέας της Μικρο-Μηχανικής Μάθησης (Tiny Machine Learning - TinyML) αποτελεί έναν ραγδαία εξελισσόμενο κλάδο της τεχνητής νοημοσύνης, ο οποίος επικεντρώνεται στην ανάπτυξη και εκτέλεση εξαιρετικά συμπιεσμένων μοντέλων σε ενσωματωμένα συστήματα με αυστηρούς περιορισμούς υπολογιστικών πόρων [3]. Ειδικότερα, στοχεύει στην υλοποίηση πολύπλοκων διαδικασιών, όπως η αναγνώριση προτύπων, απευθείας σε μικροελεγκτές (MCUs), των οποίων η διαθέσιμη μνήμη τυπικά περιορίζεται στην κλίμακα των Kilobytes [3]. Η προσέγγιση αυτή γεφυρώνει το χάσμα μεταξύ της επιστήμης δεδομένων και του προγραμματισμού υλικού χαμηλού επιπέδου.

Για να καταστεί εφικτή η ανάπτυξη νευρωνικών δικτύων σε αρχιτεκτονικές τόσο περιορισμένης μνήμης, κρίνεται επιβεβλημένη η εφαρμογή εξειδικευμένων τεχνικών βελτιστοποίησης, με κυριότερη τη διαδικασία του Κβαντισμού (Quantization) [17]. Μέσω αυτής της μεθόδου, οι παράμετροι, οι ενεργοποιήσεις και τα βάρη (weights) του εκπαιδευμένου μοντέλου μετατρέπονται από αριθμούς κινητής υποδιαστολής υψηλής ακρίβειας (32-bit floating-point) σε ακέραιες τιμές χαμηλότερης ανάλυσης (8-bit integers) [17].

Η συγκεκριμένη μαθηματική μετατροπή επιφέρει σημαντικά πλεονεκτήματα στην υλοποίηση του συστήματος. Αρχικά, επιτυγχάνει τη δραστική μείωση του απαιτούμενου αποθηκευτικού χώρου του μοντέλου έως και 4 φορές, καθιστώντας δυνατή την αποθήκευσή του στην εσωτερική μνήμη Flash της συσκευής [3]. Παράλληλα, η χρήση δεδομένων 8-bit αξιοποιεί σε απόλυτο βαθμό τις εξειδικευμένες διανυσματικές εντολές (vector instructions) που ενσωματώνει ο πυρήνας Xtensa LX7 του ESP32-S3 [4]. Αυτό έχει ως αποτέλεσμα την κατακόρυφη επιτάχυνση των υπολογισμών πινάκων (matrix multiplications) κατά τη φάση του συμπερασμού, ελαχιστοποιώντας τον χρόνο εκτέλεσης με στατιστικά ασήμαντη απώλεια στην τελική ακρίβεια πρόβλεψης του μοντέλου [3, 4].



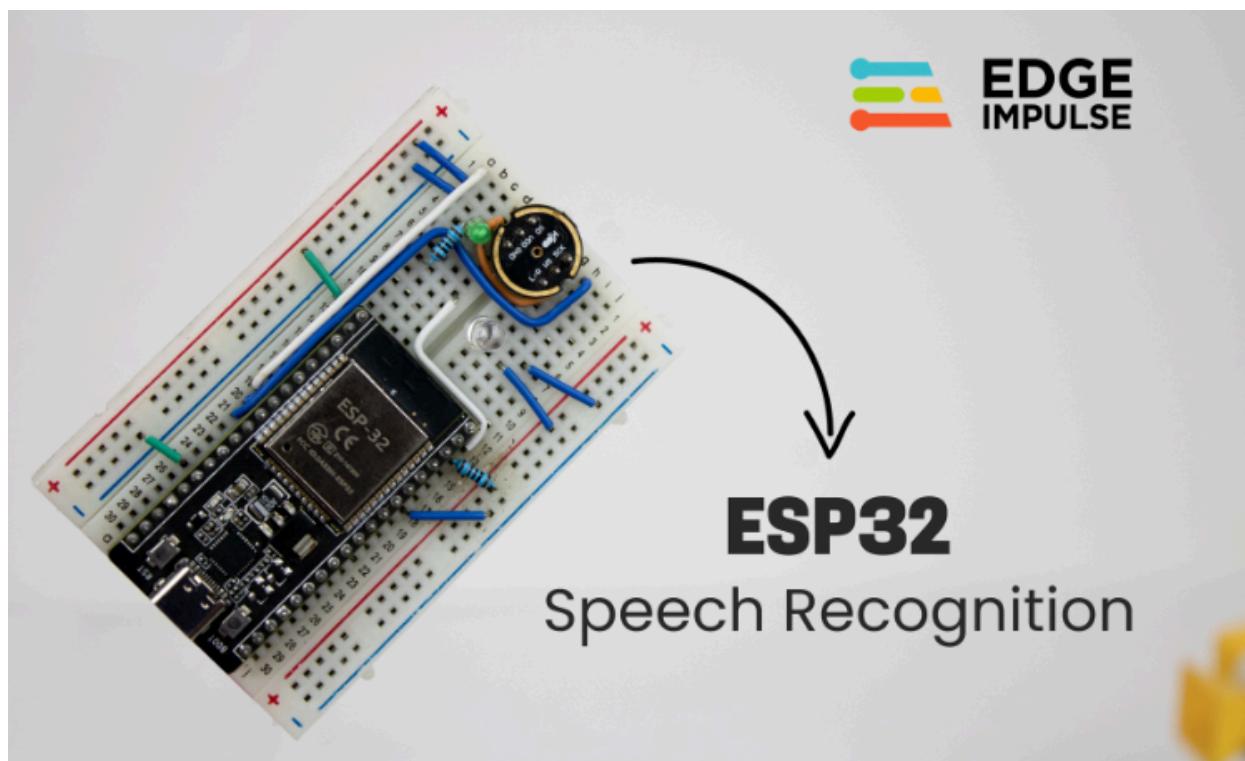
Εικόνα 44. TinyML Block diagram, Πηγή: Αναζήτηση google

8.3 Ανάπτυξη Μοντέλου μέσω της Πλατφόρμας Edge Impulse

Για την αρχιτεκτονική σχεδίαση και την εκπαίδευση του μοντέλου μηχανικής μάθησης, αξιοποιήθηκε η πλατφόρμα Edge Impulse, η οποία έχει καθιερωθεί ως το βιομηχανικό πρότυπο (industry standard) για την ανάπτυξη και υλοποίηση εφαρμογών TinyML σε ενσωματωμένα συστήματα [5]. Η συγκεκριμένη υποδομή παρέχει ένα πλήρως ολοκληρωμένο περιβάλλον ροής εργασιών (end-to-end machine learning pipeline), το οποίο αυτοματοποιεί τον κύκλο ζωής του μοντέλου.

ΚΕΦΑΛΑΙΟ 8

Η διαδικασία εκκινεί με τη συλλογή, την προεπεξεργασία και την αυστηρή σήμανση του συνόλου των ηχητικών δεδομένων εκπαίδευσης (data ingestion and labeling). Ακολούθως, το σήμα διέρχεται από το στάδιο της ψηφιακής επεξεργασίας (DSP block), όπου εκτελείται η εξαγωγή των πλέον διακριτών χαρακτηριστικών (feature extraction), μετατρέποντας τον ακατέργαστο ήχο σε δομημένα διανύσματα που τροφοδοτούν το σύστημα [16]. Το επόμενο κρίσιμο στάδιο αφορά τον σχεδιασμό της τοπολογίας και την εκπαίδευση του ίδιου του νευρωνικού δικτύου (NN block), κατά το οποίο βελτιστοποιούνται τα βάρη και οι υπερπαραμέτροι για την ακριβή ταξινόμηση των ακουστικών προτύπων [17].



Εικόνα 45.Esp32 speech recognition, Πηγή:Αναζήτηση google

Το κομβικότερο πλεονέκτημα της πλατφόρμας έγκειται στο τελικό στάδιο εξαγωγής του κώδικα (deployment). Το εκπαιδευμένο και κβαντισμένο μοντέλο δεν εξάγεται απλώς ως ένα στατικό αρχείο μεταδεδομένων, αλλά ενσωματώνεται σε μια αυτόνομη, έτοιμη προς χρήση βιβλιοθήκη της C++ (C++ inference library). Η βιβλιοθήκη αυτή είναι ειδικά βελτιστοποιημένη (hardware-accelerated) για την αρχιτεκτονική του ESP32-S3, αξιοποιώντας τις επεκτάσεις διανυσματικών υπολογισμών του υλικού για τη μεγιστοποίηση της ταχύτητας του συμπερασμού, επιτρέποντας την απρόσκοπτη ενσωμάτωσή της στο περιβάλλον του FreeRTOS [4, 5].

8.4 Συγκρότηση και Διαχείριση του Συνόλου Δεδομένων (Dataset Creation)

Η προγνωστική ικανότητα, η ακρίβεια και η δυνατότητα γενίκευσης οποιουδήποτε μοντέλου μηχανικής μάθησης είναι άρρηκτα συνδεδεμένες με την ποιότητα, τον όγκο και την ποικιλομορφία των δεδομένων εκπαίδευσης [17]. Για την κάλυψη των απαιτήσεων του συστήματος όσον αφορά την αξιόπιστη ανίχνευση ανθρώπινης ομιλίας σε πραγματικές συνθήκες, συγκροτήθηκε ένα εξειδικευμένο σώμα δεδομένων (dataset) [16].

Προκειμένου το νευρωνικό δίκτυο να αποκτήσει την ικανότητα αυστηρού διαχωρισμού του ωφέλιμου ηχητικού σήματος από τις παρεμβολές του περιβάλλοντος, τα δεδομένα κατηγοριοποιήθηκαν αυστηρά σε τρεις διακριτές κλάσεις (classes):

- **Ομιλία (Speech):** Η κλάση-στόχος, η οποία περιλαμβάνει αντιπροσωπευτικά δείγματα ανθρώπινης ομιλίας. Συλλέχθηκαν ηχογραφήσεις από πολλαπλούς ομιλητές, καλύπτοντας ένα ευρύ φάσμα φωνητικών χροιών, τονικοτήτων και ρυθμών ομιλίας, διασφαλίζοντας την ανεξαρτησία του συστήματος από τον εκάστοτε χρήστη [16].
- **Θόρυβος Υποβάθρου (Noise / Background):** Η κλάση αυτή απαρτίζεται από συνεχή ακουστικά σήματα που συνθέτουν το τυπικό περιβάλλον λειτουργίας του μικροφώνου. Ενσωματώνει περιβαλλοντικό θόρυβο, όπως η κυκλοφορία του δρόμου, η λειτουργία συστημάτων εξαερισμού (ανεμιστήρες), καθώς και τον εγγενή στατικό λευκό θόρυβο των ηλεκτρονικών κυκλωμάτων [13].
- **Απρόβλεπτοι / Λοιποί Ήχοι (Unknown / Other):** Σε αυτή την κατηγορία εντάχθηκαν μη περιοδικοί, παροδικοί θόρυβοι (transient noises) που δεν σχετίζονται με την ομιλία και ενδέχεται να προκαλέσουν ψευδώς θετικές ανιχνεύσεις (false positives). Τέτοιοι ήχοι περιλαμβάνουν απότομα χτυπήματα, κλεισίματα πορτών, πτώσεις αντικειμένων ή ήχους ζώων. και σετ ελέγχου (Testing Set - 20%) για την αποφυγή υπερεκπαίδευσης (overfitting) [17].

Για τη συνεπή τροφοδότηση του αλγορίθμου εξαγωγής χαρακτηριστικών, το σύνολο των ακουστικών καταγραφών κατατμήθηκε (windowing) σε ισομεγέθη, επικαλυπτόμενα χρονικά «παράθυρα» διάρκειας ενός (1) δευτερολέπτου.

Τέλος, ακολουθώντας τις θεμελιώδεις αρχές της μηχανικής μάθησης, το συνολικό dataset διαχωρίστηκε τυχαία σε δύο ανεξάρτητα υποσύνολα: το 80% των δειγμάτων αξιοποιήθηκε ως το Σύνολο Εκπαίδευσης (Training Set) για την προσαρμογή των βαρών του δικτύου, ενώ το εναπομείναν 20% δεσμεύτηκε αποκλειστικά ως το Σύνολο Ελέγχου (Testing Set). Ο διαχωρισμός αυτός είναι καθοριστικής σημασίας για την αντικειμενική αξιολόγηση της απόδοσης του μοντέλου σε εντελώς άγνωστα δεδομένα και την αποτροπή του φαινομένου της υπερεκπαίδευσης (overfitting) [17], όπου το δίκτυο αποστηθίζει τα δεδομένα αντί να μαθαίνει τα υποκείμενα ηχητικά πρότυπα.

8.5 Αρχιτεκτονική Συνελικτικού Νευρωνικού Δικτύου (CNN)

Η απευθείας τροφοδότηση ενός νευρωνικού δικτύου με ακατέργαστα ακουστικά δείγματα (raw audio) στο πεδίο του χρόνου είναι μια υπολογιστικά δαπανηρή και συχνά αναποτελεσματική προσέγγιση. Για τον λόγο αυτό, η προτεινόμενη μεθοδολογία εισάγει ένα ενδιάμεσο στάδιο Ψηφιακής Επεξεργασίας Σήματος (DSP block). Το στάδιο αυτό αναλαμβάνει να μετατρέψει το χρονικό παράθυρο του ενός δευτερολέπτου σε μια δισδιάστατη (2D) οπτική αναπαράσταση των συχνοτήτων, παράγοντας ένα Φασματογράφημα (Spectrogram) ή, συνηθέστερα για εφαρμογές φωνής, έναν χάρτη Ενέργειας Συστοιχίας Φίλτρων κλίμακας Mel (Mel-Filterbank Energy - MFE) [16].

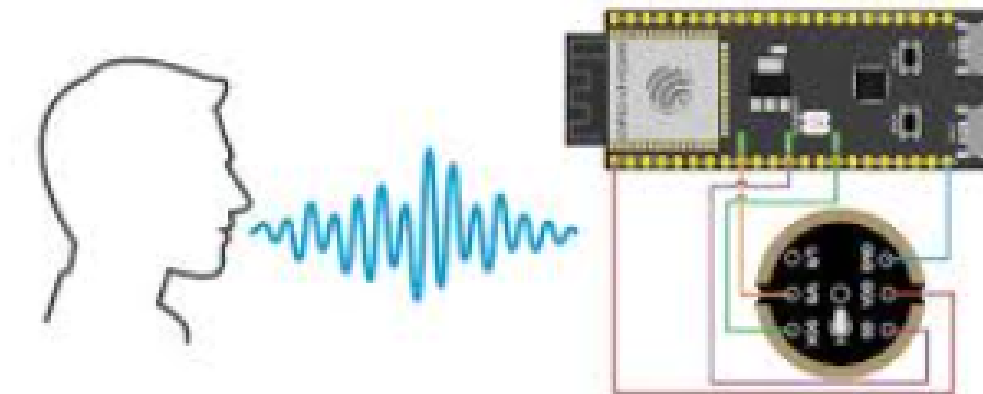
Αυτή η μετατροπή του ήχου σε «εικόνα» επιτρέπει την αξιοποίηση ενός Συνελικτικού Νευρωνικού Δικτύου (Convolutional Neural Network - CNN), μιας αρχιτεκτονικής που διαπρέπει ενγγενώς στην αναγνώριση χωρικών και χρονικών προτύπων (patterns) [17, 19]. Η εσωτερική τοπολογία του μοντέλου που υλοποιήθηκε δομείται από τα εξής διαδοχικά επίπεδα [17]:

- **Συνελικτικά Επίπεδα (Convolutional Layers 1D ή 2D):** Τοποθετημένα στην είσοδο του δικτύου, τα επίπεδα αυτά εκτελούν τη μαθηματική πράξη της συνέλιξης (convolution). Μέσω εξειδικευμένων φίλτρων (kernels), σαρώνουν το φασματογράφημα και εξάγουν θεμελιώδη ακουστικά μοτίβα, όπως οι μορφότυποι (formants) και οι αρμονικές συνιστώσες που συνθέτουν την ανθρώπινη ομιλία.
- **Επίπεδα Μέγιστης Συγκέντρωσης (MaxPooling Layers):** Παρεμβάλλονται μεταξύ των συνελικτικών επιπέδων και αναλαμβάνουν τη δραστική μείωση της διαστατικότητας (dimensionality reduction) των παραγόμενων χαρτών χαρακτηριστικών (feature maps). Επιλέγοντας μόνο τις μέγιστες τιμές ενεργοποίησης ανά περιοχή, διατηρούν την πιο κρίσιμη πληροφορία, μειώνοντας ταυτόχρονα την υπολογιστική πολυπλοκότητα (computational overhead) και ενισχύοντας την ανθεκτικότητα του μοντέλου σε μικρές χωρικές ή χρονικές μετατοπίσεις του ήχου.
- **Πλήρως Διασυνδεδεμένα Επίπεδα (Dense / Fully Connected Layers):** Αποτελούν το τελικό στάδιο λήψης αποφάσεων (classifier) του δικτύου. Εδώ, τα αφηρημένα χαρακτηριστικά που εξήχθησαν από τα προηγούμενα επίπεδα επιπεδώνονται (flattening) και συνδυάζονται. Η τελική έξοδος του δικτύου συγκλίνει σε ένα επίπεδο συνάρτησης ενεργοποίησης **Softmax**, το οποίο κανονικοποιεί τις τιμές και παράγει την τελική κατανομή πιθανοτήτων (confidence scores). Τα αποτελέσματα αυτά εξάγονται στο εύρος [0.0, 1.0] για καθεμία από τις τρεις προκαθορισμένες κλάσεις (Speech, Noise, Unknown), υποδεικνύοντας τον βαθμό βεβαιότητας του συστήματος για την τρέχουσα ταξινόμηση [19].

8.6 Speech Detection

Για τη βελτιστοποίηση της απόδοσης και την ελαχιστοποίηση της χρήσης της μνήμης (SRAM), το μοντέλο ταξινόμησης δεν φορτώνεται ως εξωτερικό αρχείο τανυστών (όπως τα παραδοσιακά `.tflite` αρχεία). Αντιθέτως, τα βάρη και οι παράμετροι του Συνελικτικού Νευρωνικού Δικτύου μεταγλωττίζονται απευθείας σε εγγενή κώδικα C++ (on-device feature classifier) μέσω της βιβλιοθήκης του Edge Impulse. Κατά την εκτέλεση, μια ανεξάρτητη διεργασία του FreeRTOS αναλύει διαρκώς τα τελευταία 1000 ms ήχου από τα MEMS μικρόφωνα, εξάγοντας κρίσιμα χαρακτηριστικά όπως το ZCR (Zero-Crossing Rate) και το ποσοστό διαμόρφωσης (modulation).

Ο αλγόριθμος είναι σχεδιασμένος να αναγνωρίζει δύο διακριτά μοτίβα: την Ανθρώπινη Ομιλία (Speech) και τον ήχο από Παλαμάκι (Clap). Προκειμένου να αποφευχθούν τα ψευδώς θετικά αποτελέσματα (false positives) από τον περιβαλλοντικό θόρυβο, εφαρμόζεται ένα αυστηρό κατώφλι: το συμβάν καταγράφεται ως έγκυρο μόνο εάν το σκορ βεβαιότητας (confidence score) του ταξινομητή ξεπεράσει το 0.85 (85%). Μόνο εφόσον ικανοποιηθεί αυτή η συνθήκη, το σύστημα ενεργοποιεί τον εντοπισμό κατεύθυνσης.



Εικόνα 46. Esp32 speech recognition, Πηγή:Αναζήτηση google

8.7 Smart Recording

Στη λειτουργία Smart Mode, η αρχιτεκτονική του συστήματος διαχωρίζει πλήρως την ανίχνευση του ήχου από την τελική αποθήκευσή του, καταργώντας τη συνεχή καταγραφή στην κάρτα MicroSD. Η πρακτική της διαρκούς εγγραφής (continuous loop recording) κρίθηκε τεχνικά ακατάλληλη, καθώς επιβαρύνει υπερβολικά τον δίαυλο SPI, προκαλεί ταχεία φθορά των κυττάρων μνήμης Flash (περιορισμένοι κύκλοι εγγραφής/διαγραφής) και οδηγεί σε άσκοπη εξάντληση της μπαταρίας 18650 λόγω των υψηλών αιχμών ρεύματος. Αντί μιας αόριστης καταγραφής που ελέγχεται από κάποιο χρονικό όριο αδράνειας (timeout), το σύστημα υιοθετεί μια ντετερμινιστική προσέγγιση αυστηρά ελεγχόμενων χρονικών παραθύρων (time-windowed recording). Ο χρήστης ορίζει εκ των προτέρων την ακριβή επιθυμητή διάρκεια καταγραφής (σε δευτερόλεπτα) μέσω της εφαρμογής, παραμετροποιώντας τη συσκευή ανάλογα με τις ανάγκες της εκάστοτε επιτήρησης.

Ο μηχανισμός ενεργοποίησης της εγγραφής διαφοροποιείται δυναμικά, βάσει του ακουστικού στόχου (Target) που έχει επιλεγεί από τον χρήστη:

- **Ανίχνευση Στιγμιαίου Συμβάντος (Clap):** Στην περίπτωση απότομων και παροδικών ηχητικών συμβάντων, όπως το παλαμάκι, το Συνελικτικό Νευρωνικό Δίκτυο (TinyML) λειτουργεί ως ευφυής πύλη ελέγχου (smart gate). Το σύστημα παραμένει σε κατάσταση χαμηλής κατανάλωσης (Armed/Idle), αναλύοντας συνεχώς το περιβάλλον στο παρασκήνιο χωρίς να γράφει δεδομένα. Μόλις ο on-device ταξινομητής επιβεβαιώσει το γεγονός (με βαθμό βεβαιότητας ≥ 0.85), η συσκευή μεταβαίνει ακαριαία στην υπολογισμένη κατεύθυνση και ενεργοποιεί αυτόματα την εγγραφή για το προκαθορισμένο χρονικό παράθυρο.
- **Συνεχής Παρακολούθηση Ομιλίας:** Στην περίπτωση της ομιλίας, όπου η πηγή παρουσιάζει διάρκεια και μπορεί να μετακινείται στον χώρο, η διαδικασία δεν εκκινεί παθητικά αλλά απαιτεί ρητή εντολή (Start) από τον χρήστη. Με την έναρξη, το σύστημα αρχίζει την καταγραφή και παράλληλα εκτελεί συνεχή μηχανική παρακολούθηση (Follow). Όσο ο ομιλητής μετακινείται, ο αλγόριθμος υπολογίζει τον νέο τομέα και προσαρμόζει αυτόματα τη γωνία στόχευσης, διασφαλίζοντας ότι το κατευθυντικό μικρόφωνο παραμένει στραμμένο προς την πηγή καθ' όλη τη διάρκεια της εγγραφής.

Και στις δύο περιπτώσεις, είναι σημαντικό να τονιστεί ότι η δειγματοληψία του ήχου προς αποθήκευση πραγματοποιείται αποκλειστικά από το κεντρικό, αναλογικό κατευθυντικό μικρόφωνο MAX9814. Τα ψηφιακά μικρόφωνα INMP441 της συστοιχίας χρησιμοποιούνται αυστηρά για την τροφοδότηση των αλγορίθμων εντοπισμού (GCC-PHAT) και αναγνώρισης (TinyML), διαχωρίζοντας έτσι το σήμα ελέγχου από το ωφέλιμο ακουστικό σήμα. Μόλις παρέλθει ο προκαθορισμένος χρόνος, η ανεξάρτητη, ασύγχρονη διεργασία εγγραφής (SD Recording Task του FreeRTOS) τερματίζεται ομαλά, αδειάζει τους κυκλικούς καταχωρητές (buffers) και κλείνει το αρχείο WAV με ασφάλεια στο σύστημα αρχείων FAT32. Αυτή η διαδικασία εξασφαλίζει τη βέλτιστη διαχείριση των αποθηκευτικών πόρων και την απόλυτη αποτροπή καταστροφής των δεδομένων κατά την εγγραφή.

ΚΕΦΑΛΑΙΟ 9: FLUTTER APPLICATION

Η αποτελεσματική λειτουργία και η πλήρης αξιοποίηση των δυνατοτήτων ενός σύγχρονου ενσωματωμένου συστήματος δεν εξαρτώνται αποκλειστικά από το υλικό (hardware) ή την πολυπλοκότητα των αλγορίθμων του, αλλά και από τον τρόπο με τον οποίο ο τελικός χρήστης αλληλεπιδρά με αυτό. Επομένως, ένα προηγμένο υλικό σύστημα απαιτεί ένα εξίσου σύγχρονο, διαισθητικό και ταχύτατο περιβάλλον ελέγχου, το οποίο να "κρύβει" την τεχνική πολυπλοκότητα στο παρασκήνιο και να προσφέρει άμεση προσβασιμότητα και ευκολία χρήσης (User Experience - UX).

Για την κάλυψη αυτής της ανάγκης, σχεδιάστηκε και αναπτύχθηκε από το μηδέν μια εξειδικευμένη εφαρμογή για έξυπνες συσκευές (Mobile Application), η οποία λειτουργεί ως ο κεντρικός κόμβος διαχείρισης (hub) του έργου. Η συγκεκριμένη εφαρμογή επικοινωνεί απευθείας και ασύρματα με το κατευθυντικό μικρόφωνο, επιτρέποντας στον χρήστη να παραμετροποιεί τη συσκευή και να λαμβάνει οπτική και ακουστική ανατροφοδότηση σε πραγματικό χρόνο (real-time monitoring). Μέσω μιας προσεγμένης γραφικής διεπαφής (GUI), παρέχεται η δυνατότητα απρόσκοπτης εναλλαγής μεταξύ των καταστάσεων λειτουργίας (Manual, Auto, Smart Mode), ο χειροκίνητος έλεγχος της στόχευσης του σερβοκινητήρα, η παρακολούθηση των αποτελεσμάτων του αλγορίθμου εντοπισμού (SSL) και της τεχνητής νοημοσύνης (TinyML), καθώς και η λήψη ζωντανής ροής ήχου (live audio streaming). Με αυτόν τον τρόπο, γεφυρώνεται το χάσμα μεταξύ της φυσικής διάταξης και του τελικού χρήστη, μετατρέποντας το μικρόφωνο σε ένα ολοκληρωμένο και πλήρως διαδραστικό οικοσύστημα (Interactive Ecosystem).



Εικόνα 47. Flutter App Developing, Πηγή:Αναζήτηση google

9.1 Αρχιτεκτονική Εφαρμογής (Application Architecture)

Η εφαρμογή αναπτύχθηκε στο σύγχρονο πλαίσιο (framework) **Flutter** της Google, χρησιμοποιώντας την αντικειμενοστραφή γλώσσα προγραμματισμού **Dart**. Η επιλογή του συγκεκριμένου οικοσυστήματος έγινε με αυστηρά στρατηγικά κριτήρια, λόγω της εγγενούς ικανότητάς του να μεταγλωττίζει τον κώδικα απευθείας σε εντολές μηχανής (native code). Αυτό επιτρέπει τη δημιουργία εφαρμογών εξαιρετικά υψηλής απόδοσης για πολλαπλά περιβάλλοντα, αξιοποιώντας μια ενιαία βάση κώδικα (single codebase). Στο πλαίσιο της παρούσας κατασκευής, το χαρακτηριστικό αυτό αποδείχθηκε σωτήριο, καθώς επέτρεψε στην εφαρμογή να εκτελείται όχι μόνο ως εγκατεστημένο λογισμικό σε κινητές συσκευές (ως αυτόνομο APK), αλλά και ως ελαφριά διαδικτυακή εφαρμογή (Web App) που φιλοξενείται απευθείας στη μνήμη του μικροελεγκτή, παρέχοντας ευελιξία πρόσβασης μέσω του τοπικού δικτύου (SoftAP).

Η αρχιτεκτονική του λογισμικού απαιτούσε τη διαχείριση εξαιρετικά απαιτητικών και χρονικά κρίσιμων διεργασιών. Η εγγενής υποστήριξη της γλώσσας Dart για ασύγχρονο προγραμματισμό (asynchronous programming) και η ενσωματωμένη, βελτιστοποιημένη διαχείριση συνεχών ροών δεδομένων μέσω του API των **Streams**, αποτέλεσαν τους καθοριστικούς παράγοντες της υλοποίησης. Η εφαρμογή οφείλει να διαχειρίζεται ταυτόχρονα, αδιάλειπτα και ασύγχρονα δύο εντελώς διαφορετικούς τύπους δεδομένων πάνω στο ίδιο δίκτυο: αφενός τα δομημένα μηνύματα ελέγχου και τηλεμετρίας (σε μορφή κειμένου JSON) και αφετέρου τη συνεχή λήψη της ακατέργαστης ακουστικής ροής (binary PCM frames, int16 LE στα 8 kHz). Η ενοποίηση αυτών των ροών σε μια κοινή, μόνιμη σύνδεση WebSocket (στη θύρα 81) απαιτεί εξαιρετικά αποδοτικό διαχωρισμό (parsing) των πακέτων στο παρασκήνιο, χωρίς να προκαλείται εμπλοκή (blocking) στο κεντρικό νήμα (main thread) της εφαρμογής.

Για τη διασφάλιση της ορθής και σταθερής λειτουργίας υπό τέτοιες συνθήκες υπολογιστικού φόρτου, η αρχιτεκτονική της εφαρμογής βασίστηκε στην προηγμένη δομή **BLoC** (Business Logic Component), σε άμεσο συνδυασμό με το σχεδιαστικό πρότυπο **Provider** για τη διαχείριση της κατάστασης (State Management) και την ασφαλή έγχυση εξαρτήσεων (dependency injection). Η σχεδιαστική αυτή προσέγγιση επιβάλλει τον αυστηρό και απόλυτο διαχωρισμό του οπτικού τμήματος (User Interface - UI) από την πολύπλοκη λογική της επικοινωνίας δικτύου. Καθώς η εφαρμογή λαμβάνει διαρκώς νέα δεδομένα από τον μικροελεγκτή ESP32-S3 —όπως είναι οι συνεχείς ανανεώσεις της υπολογισμένης γωνίας του σερβοκινητήρα, οι παράμετροι χρονικής βαθμονόμησης ή οι ακαριαίες ενδείξεις ανίχνευσης ανθρώπινης ομιλίας από το TinyML— η διεπαφή πρέπει να αντιδρά άμεσα και απρόσκοπτα.

Η χρήση της αρχιτεκτονικής BLoC επιτρέπει στο σύστημα να απομονώνει τις αλλαγές και να επανασχεδιάζει (rebuild) αποκλειστικά και μόνο τα συγκεκριμένα γραφικά στοιχεία (widgets) που επηρεάζονται τη δεδομένη στιγμή, αποτρέποντας την άσκοπη και ενεργοβόρα επαναφόρτωση ολόκληρης της οθόνης. Για παράδειγμα, η ανανέωση της εικονικής πυξίδας στην αυτόματη λειτουργία ή η απεικόνιση του ακουστικού φάσματος (VU Meter) πραγματοποιείται τοπικά στο UI, χωρίς να διακόπτεται η ροή των δεδομένων στο παρασκήνιο. Ως αποτέλεσμα, η συσκευή

εξοικονομεί πολύτιμους επεξεργαστικούς πόρους (CPU και μνήμη RAM), εξασφαλίζοντας μια απόλυτα ομαλή και αδιάλειπτη εμπειρία χρήστη (User Experience). Ο ρυθμός ανανέωσης διατηρείται σταθερά στα 60 καρέ ανά δευτερόλεπτο (fps), ακόμη και κατά την εξόχως απαιτητική, ζωντανή οπτικοποίηση των αποτελεσμάτων των αλγορίθμων.

9.2 Λειτουργία Σημείου Πρόσβασης (ESP32 AP Mode) και Δικτύωση

Για να διασφαλιστεί η απόλυτη φορητότητα του συστήματος και να αποτραπεί η εξάρτησή του από την εκάστοτε υποδομή εξωτερικών δικτύων Wi-Fi (δρομολογητές), ο μικροελεγκτής ESP32-S3 [4] διαμορφώθηκε δομικά σε λειτουργία Σημείου Πρόσβασης (Soft Access Point - AP). Μέσω της στοίβας πρωτοκόλλων δικτύου (LwIP) που εκτελείται στον υποστηρικτικό πυρήνα του συστήματος [4], η συσκευή εκπέμπει ένα δικό της, ανεξάρτητο τοπικό δίκτυο (π.χ. "SmartMic_AP"). Με αυτόν τον τρόπο, η φορητή συσκευή ελέγχου (κινητό τηλέφωνο ή tablet) συνδέεται απευθείας με το μικρόφωνο σε τοπολογία από άκρο σε άκρο (Peer-to-Peer). Αυτή η απομόνωση του δικτύου αποκλείει εξωτερικές παρεμβολές ή τοπικές συμφορήσεις εύρους ζώνης (bandwidth bottlenecks) που θα υποβάθμιζαν δραματικά την ποιότητα μεταφοράς των ακουστικών δεδομένων.

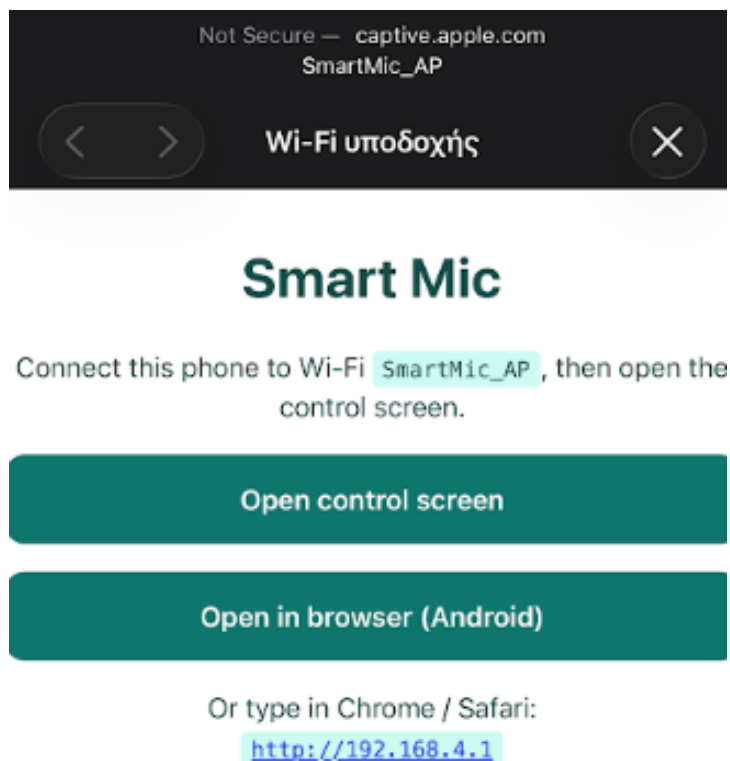
Πάνω σε αυτή την αδιάλειπτη ασύρματη σύνδεση, ο μικροελεγκτής [4] φιλοξενεί και εκτελεί διαρκώς έναν ενσωματωμένο εξυπηρετητή (WebSocket Server). Η επιλογή του πρωτοκόλλου WebSockets, έναντι της κλασικής αρχιτεκτονικής REST HTTP, κρίθηκε επιβεβλημένη, καθώς το HTTP βασίζεται σε χρονοβόρα μοτίβα αίτησης/απόκρισης (request/response) που δημιουργούν περιττό διαχειριστικό φόρτο δικτύου (overhead). Αντιθέτως, τα WebSockets εγκαθιδρύουν μια μόνιμη, αμφίδρομη (full-duplex) σύνδεση ελέγχου [6]. Αυτή η τεχνολογική επιλογή επιτρέπει στην εφαρμογή να στέλνει εντολές τηλεχειρισμού —όπως ο άμεσος καθορισμός της γωνίας του σερβοκινητήρα— και να λαμβάνει κρίσιμα δεδομένα τηλεμετρίας πρακτικά ακαριαία, εξασφαλίζοντας την εξαιρετικά χαμηλή καθυστέρηση (ultra-low latency) που απαιτεί η διάδραση σε πραγματικό χρόνο.

Για να διασφαλιστεί η απόλυτη φορητότητα, ο ESP32-S3 διαμορφώθηκε σε λειτουργία αυτόνομου Σημείου Πρόσβασης (SoftAP) με ονομασία δικτύου (SSID) «SmartMic_AP». Ο χρήστης δεν απαιτείται να εγκαταστήσει καμία εφαρμογή μέσω USB (debugging) ούτε να έχει στη διάθεσή του υποδομή router. Ανοίγοντας απλώς τον φυλλομετρητή (browser) της κινητής συσκευής, ένα ενσωματωμένο σύστημα Captive DNS (στη θύρα 53) ανακατευθύνει αυτόματα τον χρήστη στην ενσωματωμένη σελίδα ελέγχου (σερβίρεται από τη θύρα 80).

Η αμφίδρομη επικοινωνία ελέγχου με τον μικροελεγκτή γίνεται αποκλειστικά μέσω μόνιμης σύνδεσης WebSocket (στη θύρα 81) κάνοντας χρήση δομημένων μηνυμάτων μορφής JSON (π.χ. `{"mode": "manual", "angle": 90}`). Επιπλέον, για τον άμεσο οπτικό έλεγχο στο πεδίο, το σύστημα χρησιμοποιεί ένα ενσωματωμένο LED RGB (WS2812 στο GPIO 38). Το LED λειτουργεί ως διαγνωστικός δείκτης: ανάβει σταθερά κόκκινο όταν δεν υπάρχει ασύρματη σύνδεση, πράσινο όταν επιτευχθεί ζεύξη (WebSocket Client connected), ενώ αναβοσβήνει με μπλε χρώμα επιβεβαιώνοντας τα πακέτα διατήρησης σύνδεσης (ping/pong).

9.3 Αναμετάδοση Ήχου μέσω Πρωτοκόλλου WebSocket (Audio Streaming)

Ένα από τα πλέον απαιτητικά χαρακτηριστικά της εφαρμογής είναι η δυνατότητα ζωντανής ακουστικής παρακολούθησης (Live Monitor). Αποφεύγοντας την πολυπλοκότητα των πολλαπλών πρωτοκόλλων, η ροή του ψηφιοποιημένου ήχου μεταδίδεται από τον μικροελεγκτή ESP32-S3 αποκλειστικά μέσω binary frames στο ίδιο κανάλι WebSocket (θύρα 81) που εξυπηρετεί και τις εντολές ελέγχου.

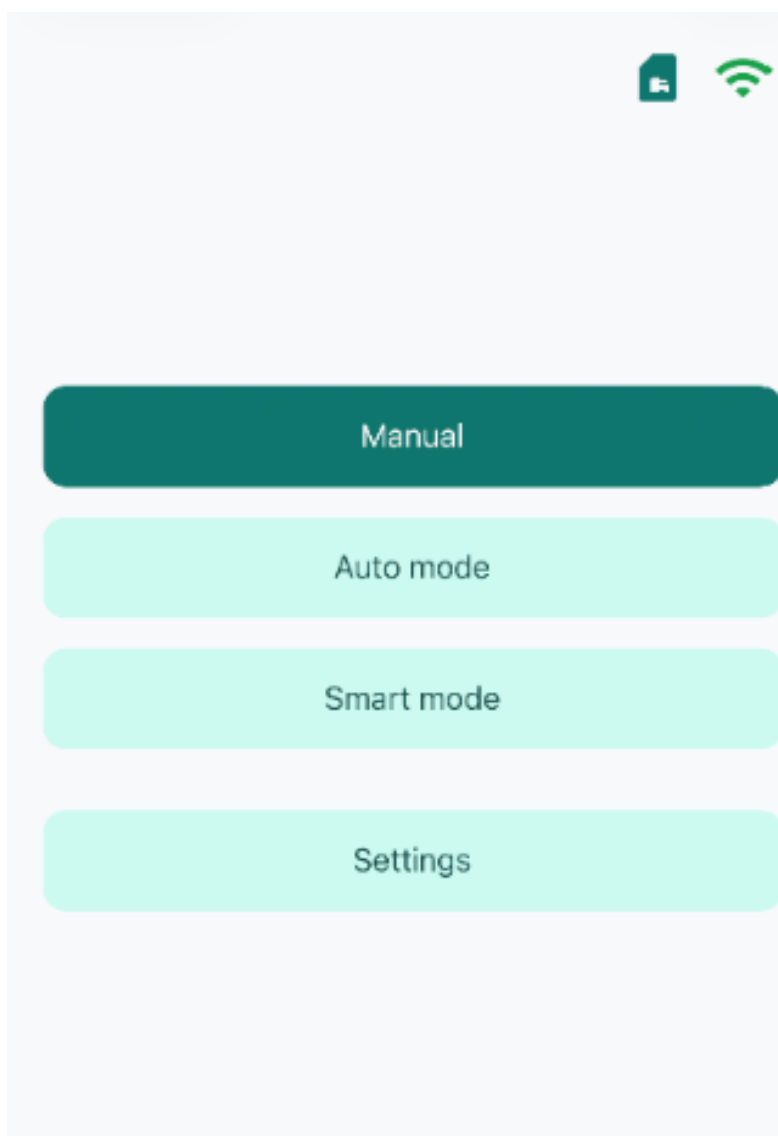


Εικόνα 48 Είσοδος στο websocket , Πηγή:screenshot κινητού.

Στην πλευρά του λογισμικού διεπαφής (Flutter App), τα εισερχόμενα δυαδικά δεδομένα συσσωρεύονται δυναμικά και αναπαράγονται. Δεδομένου ότι ορισμένα περιβάλλοντα (όπως ο φυλλομετρητής Safari στο iOS) εμφανίζουν ασυμβατότητες κατά την απευθείας αναπαραγωγή native buffers στα 8 kHz, ενσωματώθηκε μια ρουτίνα επαναδειγματοληψίας εντός του Web AudioContext (τυπικά στα 44100 Hz). Η προσέγγιση αυτή επιτρέπει την απρόσκοπτη ακρόαση του ήχου στο ενσωματωμένο Web UI, καθιστώντας περιττή την εγκατάσταση εγγενούς APK.

9.4 Σχεδιασμός Γραφικής Διεπαφής Χρήστη (User Interface)

Η Γραφική Διεπαφή Χρήστη της εφαρμογής αναπτύχθηκε με αυστηρή προσήλωση στις κατευθυντήριες γραμμές του προτύπου **Material Design**, υιοθετώντας μια σχεδιαστική φιλοσοφία που εστιάζει στον μινιμαλισμό, τη σαφήνεια και τη βέλτιστη εργονομία. Πρωταρχικός στόχος της σχεδίασης ήταν η άμεση κατανόηση των λειτουργιών από τον χρήστη, αφαιρώντας τον περιττό οπτικό θόρυβο και αναδεικνύοντας τις κρίσιμες πληροφορίες τηλεμετρίας.



Εικόνα 49. Αρχική Μενού εφαρμογής, Πηγή:screenshot κινητού

Για την επίτευξη αυτού του σκοπού, η αρχιτεκτονική της κεντρικής οθόνης οργανώθηκε στρατηγικά, διαχωριζόμενη σε τρεις διακριτές, λειτουργικές ζώνες:

- **Μπάρα Κατάστασης Δικτύου:** Τοποθετημένη στο ανώτερο τμήμα της διεπαφής, παρέχει συνεχή και άμεση οπτική ανατροφοδότηση σχετικά με την κατάσταση σύνδεσης της συσκευής (Συνδεδεμένο / Αποσυνδεδεμένο), επιβεβαιώνοντας την ακεραιότητα της ζεύξης με το τοπικό δίκτυο Access Point του μικροελεγκτή.
- **Επιλογέας Κατάστασης Λειτουργίας (Mode Selector):** Λειτουργεί ως ο κεντρικός μηχανισμός πλοήγησης, επιτρέποντας στον χρήστη την ταχύτατη εναλλαγή μεταξύ των τριών βασικών καταστάσεων του συστήματος (Manual, Auto, Smart).
- **Δυναμικός Χώρος Αλληλεπίδρασης:** Καταλαμβάνει το μεγαλύτερο και κεντρικότερο τμήμα της οθόνης και προσαρμόζεται πλήρως δυναμικά. Το περιεχόμενό του ανασχεδιάζεται αυτόματα, παρουσιάζοντας τα αντίστοιχα χειριστήρια ή γραφήματα αυστηρά ανάλογα με το επιλεγμένο Mode.

Επιπρόσθετα, προκειμένου να ενισχυθεί η διαδραστικότητα, η διεπαφή ενσωματώνει ένα προηγμένο γραφικό στοιχείο οπτικοποίησης της στάθμης του ήχου (Audio Visualizer / VU Meter). Το συγκεκριμένο στοιχείο αντιδρά άμεσα και σε πραγματικό χρόνο στη ζωντανή ακουστική ροή (που παραλαμβάνεται από τα binary frames του WebSocket), προσφέροντας στον χρήστη μια ακριβή, οπτική επιβεβαίωση της ηχητικής πίεσης και της ποιότητας του σήματος που συλλαμβάνει η απομακρυσμένη συστοιχία των μικροφώνων.

9.5 Χειροκίνητη Λειτουργία (Manual Mode)

Στη Χειροκίνητη Λειτουργία (Manual Mode), ο χρήστης αναλαμβάνει τον απόλυτο και άμεσο έλεγχο της κατευθυντικότητας και των λειτουργιών καταγραφής του συστήματος, παρακάμπτοντας τους αυτοματισμούς της τεχνητής νοημοσύνης. Η γραφική διεπαφή έχει σχεδιαστεί με γνώμονα την εργονομία και την ταχύτητα απόκρισης, προσφέροντας στην οθόνη της κινητής συσκευής διαδραστικά πλήκτρα ελέγχου (jog buttons) για τη μικρομετρική αριστερόστροφη ή δεξιόστροφη περιστροφή της διάταξης. Οι εντολές αυτές κωδικοποιούνται στιγμιαία σε μορφή JSON και αποστέλλονται μέσω της μόνιμης σύνδεσης WebSocket (θύρα 81) στον μικροελεγκτή ESP32-S3, ο οποίος μεταφράζει το αίτημα σε ακριβείς παλμούς PWM προς τον σερβοκινητήρα.

Παράλληλα με τον ελεύθερο έλεγχο, το περιβάλλον διεπαφής ενσωματώνει πλήκτρα άμεσης μετάβασης σε προκαθορισμένες, κομβικές γωνίες (0°, 90°, 180°, 270°). Η επιλογή μίας εκ των προκαθορισμένων γωνιών ενεργοποιεί την εσωτερική ρουτίνα διαχείρισης κίνησης (`plan_to()`), η οποία υπολογίζει αυτόματα τον απαιτούμενο χρόνο περιστροφής με βάση την τρέχουσα αποθηκευμένη θέση (heading) της διάταξης. Κεντρικό ρόλο στην οπτική και λειτουργική ιεραρχία της οθόνης παίζει το ειδικό πλήκτρο "CENTER". Το πλήκτρο αυτό αναλαμβάνει την ασφαλή και άμεση επαναφορά του κινητήρα στη φυσική θέση μηδέν (0°), λειτουργώντας ως σταθερό γεωμετρικό σημείο αναφοράς για τον χρήστη.

Πέραν του μηχανικού ελέγχου, από την ίδια κεντρική οθόνη ο χρήστης έχει πλήρη πρόσβαση στο ακουστικό υποσύστημα. Μέσω της επιλογής "Listen", ενεργοποιείται η ζωντανή ακρόαση του περιβάλλοντος χώρου (Live Monitor) απευθείας στην έξοδο ήχου του κινητού τηλεφώνου. Η μεταφορά του ήχου υλοποιείται μέσω ακατέργαστων δυαδικών πλαισίων (binary PCM frames)

πάνω στο ίδιο, ενιαίο κανάλι WebSocket. Χάρη στην αυστηρή ιεράρχηση των διεργασιών (FreeRTOS Tasks) στους δύο πυρήνες του συστήματος, η συνεχής δειγματοληψία και η ασύρματη μετάδοση του ήχου εκτελούνται εντελώς ασύγχρονα. Αυτό διασφαλίζει ότι η κίνηση του σερβοκινητήρα κατά τη διάρκεια της στόχευσης δεν παγώνει και δεν προκαλεί διακοπές (stutters) στη ροή της ακρόασης.

Επιπροσθέτως, το πλήκτρο "Rec" επιτρέπει την άμεση έναρξη και λήξη της ηχογράφησης του λαμβανόμενου σήματος από το κεντρικό, αναλογικό μικρόφωνο MAX9814. Τα ηχητικά δεδομένα αποθηκεύονται τοπικά στην κάρτα MicroSD σε μορφή ασυμπίεστων αρχείων WAV (π.χ. *rec_NNN.wav*), χρησιμοποιώντας μια ανεξάρτητη διεργασία εγγραφής (writer task) που δεν προκαλεί καθυστερήσεις (blocking) στο υπόλοιπο σύστημα. Τέλος, για λόγους ασφάλειας και προστασίας των δεδομένων χαρτογράφησης του κινητήρα, η συγκεκριμένη διεπαφή ελέγχου είναι αυστηρά διαχωρισμένη από τη σελίδα των ρυθμίσεων (Settings). Με αυτόν τον τρόπο αποτρέπεται η ακούσια αλλοίωση των παραμέτρων κατά την καθημερινή χρήση, καθώς η ευαίσθητη διαδικασία της χρονικής βαθμονόμησης (calibration) των οκτώ τόξων της διάταξης εκτελείται αποκλειστικά σε απομονωμένο περιβάλλον.



Εικόνα 50. Manual λειτουργία εφαρμογή, Πηγή:screenshot κινητού

9.6 Αυτόματη Λειτουργία Εντοπισμού (Auto Mode)

Η Αυτόματη Λειτουργία (Auto Mode) αποτελεί τον πυρήνα της χωρικής αντίληψης του συστήματος, έχοντας σχεδιαστεί για τον δυναμικό εντοπισμό και την παρακολούθηση της εκάστοτε ηχητικής πηγής (Sound Source Localization - SSL). Η ανίχνευση του χώρου δεν γίνεται με συνεχή (μοίρα-προς-μοίρα) σάρωση, αλλά διακριτοποιείται σε 8 ισομεγέθεις τομείς των 45° (π.χ. Front, Front Right, Left, κ.λπ.), εξασφαλίζοντας ταχεία απόκριση και αποτρέποντας τη μικρο-ταλάντωση (jitter) του μηχανισμού.

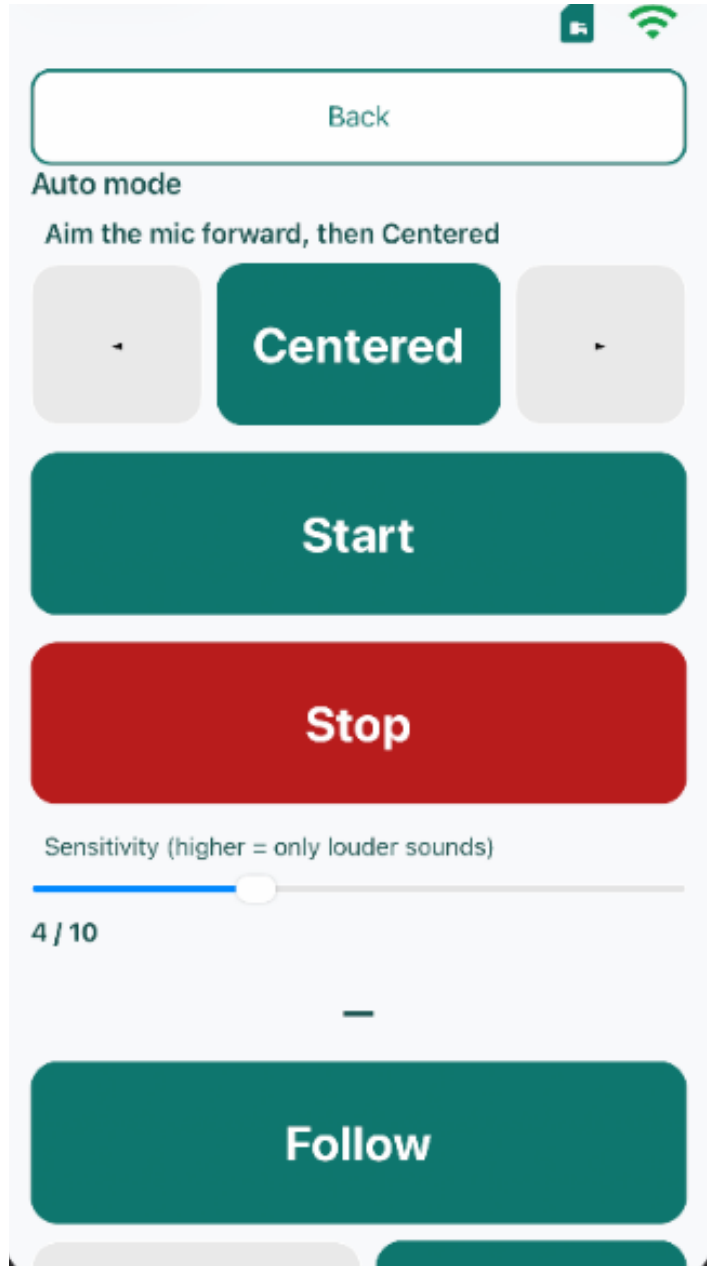
Μία από τις σημαντικότερες μηχανολογικές προκλήσεις της κατασκευής είναι η διαχείριση των καλωδίων, καθώς η χρήση σερβοκινητήρα συνεχούς περιστροφής (continuous rotation) στερείται απόλυτης ανατροφοδότησης θέσης. Για την αποφυγή συστρόφης και πιθανής φθοράς των καλωδίων (cable tangling), το λογισμικό της εφαρμογής επιβάλλει μια υποχρεωτική διαδικασία αρχικοποίησης, γνωστή ως "Centered Gate". Ο χρήστης καλείται, μέσω των διαδραστικών πλήκτρων ελέγχου (jog buttons), να ευθυγραμμίσει οπτικά τη διάταξη ώστε το μικρόφωνο να στοχεύει ακριβώς στην ευθεία. Με το πάτημα του ειδικού πλήκτρου "Centered", το σύστημα εκτελεί τη ρουτίνα `mark_center()`, καταχωρώντας την τρέχουσα φυσική θέση ως το απόλυτο σημείο αναφοράς των 0°. Μέχρι να ολοκληρωθεί επιτυχώς αυτό το βήμα προστασίας, όλες οι αυτόματες ενέργειες κίνησης και καταγραφής παραμένουν αυστηρά κλειδωμένες.

Μετά την επιτυχή αρχικοποίηση του άξονα, το περιβάλλον διεπαφής ξεκλειδώνει, παρέχοντας στον χρήστη εργαλεία δυναμικής παραμετροποίησης. Μέσω ενός ρυθμιστικού (slider) κλίμακας από 1 έως 10, είναι δυνατή η άμεση προσαρμογή της ευαισθησίας (sensitivity) του συστήματος. Η αύξηση αυτής της τιμής αναπροσαρμόζει εσωτερικά το κατώφλι της απαιτούμενης ακουστικής ενέργειας, επιτρέποντας στον αλγόριθμο να αγνοεί τον ασθενή περιβαλλοντικό θόρυβο (noise floor) και να εστιάζει αποκλειστικά σε κυρίαρχες ηχητικές πηγές. Παράλληλα, η οθόνη του κινητού απεικονίζει ζωντανά την κατεύθυνση της πηγής, επεξεργαζόμενη την πλεονάζουσα ενέργεια (energy excess) που συλλαμβάνουν τα τέσσερα ψηφιακά μικρόφωνα MEMS, μέσω των συνεχών μηνυμάτων τηλεμετρίας που καταφθάνουν από το WebSocket.

Έχοντας ρυθμίσει το σύστημα, ο χρήστης διαθέτει τρεις διακριτές επιλογές ελέγχου και διαχείρισης του εντοπισμού:

- **Follow:** Ενεργοποιεί τη φυσική παρακολούθηση της πηγής. Ο μικροελεγκτής μεταφράζει τον εκτιμώμενο τομέα άφιξης του ήχου σε εντολές κίνησης μέσω της κοινής ρουτίνας `plan_to()`, αναγκάζοντας τον σερβοκινητήρα να περιστρέφεται αυτόματα και να διατηρεί το κεντρικό μικρόφωνο στραμμένο προς τον στόχο σε πραγματικό χρόνο.
- **Rec:** Εκκινεί την ασύγχρονη εγγραφή του ήχου στην εξωτερική κάρτα MicroSD, λαμβάνοντας δεδομένα αποκλειστικά από το κεντρικό κατευθυντικό μικρόφωνο MAX9814. Η εγγραφή εκτελείται ανεξάρτητα από την κίνηση της διάταξης, καθιστώντας τη λειτουργία ιδανική για στατική παρακολούθηση.

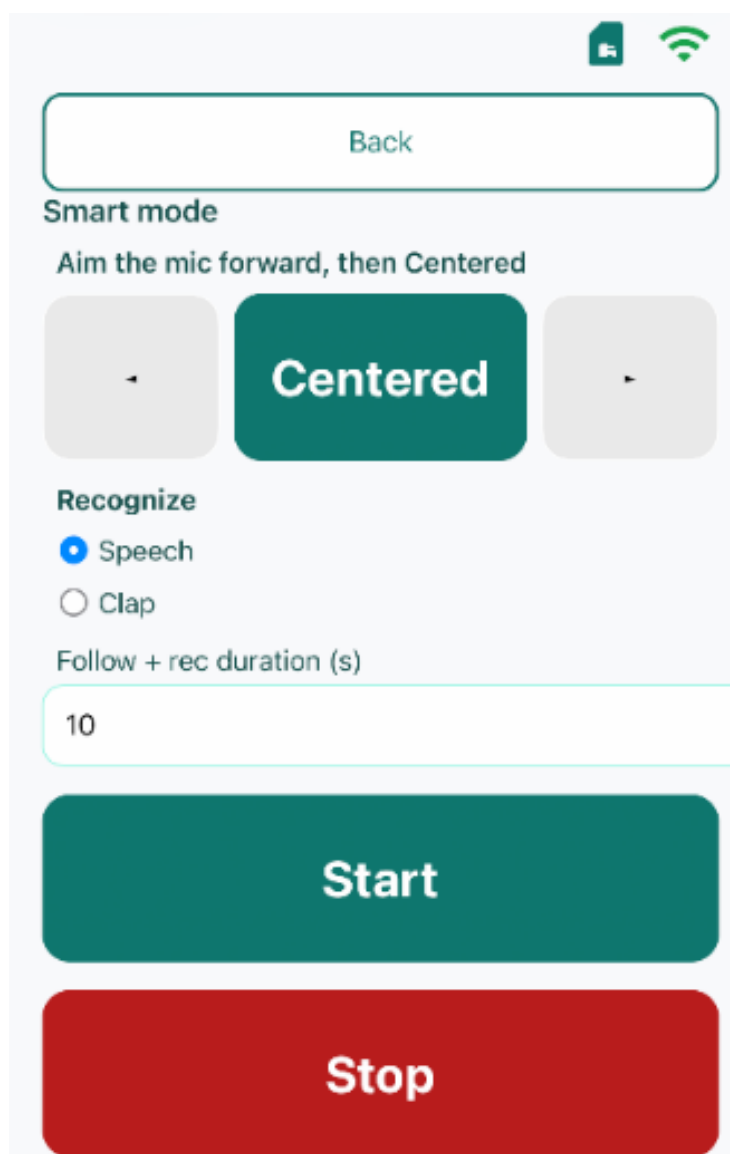
- **Follow and Rec:** Αποτελεί την πλέον απαιτητική λειτουργία, συνδυάζοντας την ταυτόχρονη φυσική παρακολούθηση του στόχου και την αδιάλειπτη ηχογράφηση υψηλής ποιότητας, αναδεικνύοντας την ικανότητα του συστήματος να διαχειρίζεται παράλληλα πολλαπλά Tasks στο περιβάλλον του FreeRTOS.



Εικόνα 51. Auto λειτουργία εφαρμογής, Πηγή:screenshot κινητού

9.7 Έξυπνη Λειτουργία Τεχνητής Νοημοσύνης (Smart Mode)

Στην κατάσταση Smart (TinyML), η εφαρμογή υπερβαίνει τον απλό ακουστικό εντοπισμό και αναλαμβάνει τον ρόλο ενός προηγμένου, έξυπνου σταθμού ανίχνευσης και ταξινόμησης γεγονότων. Όπως ακριβώς ισχύει και στην Αυτόματη Λειτουργία (Auto Mode), η διαδικασία εκκινεί με την υποχρεωτική αρχικοποίηση του άξονα περιστροφής μέσω του ειδικού πλήκτρου **"Centered"**. Η ενέργεια αυτή καταχωρεί την τρέχουσα φυσική θέση του μικροφώνου ως το απόλυτο σημείο αναφοράς των 0° (`mark_center`), διασφαλίζοντας την αποτροπή τυχόν συστροφής και καταπόνησης των καλωδίων της διάταξης. Εφόσον δεν ολοκληρωθεί επιτυχώς αυτό το βήμα ασφαλείας, το σύστημα διατηρεί όλες τις αυτοματοποιημένες ρουτίνες κίνησης και καταγραφής αυστηρά κλειδωμένες.



Εικόνα 52.smart λειτουργία εφαρμογής Πηγή:screenshot κινητού

Σε αντίθεση με την Αυτόματη Λειτουργία, η οποία παρακολουθεί οποιαδήποτε ηχητική πηγή υπερβεί το κατώφλι θορύβου, το Smart Mode επιτρέπει τη στοχευμένη παρακολούθηση. Μέσω της Γραφικής Διεπαφής, ο χρήστης καλείται να επιλέξει τον συγκεκριμένο ηχητικό στόχο (Target) που αναζητά —δηλαδή Ανθρώπινη Ομιλία (Speech) ή κάποιον απότομο παλμικό ήχο, όπως ένα Παλαμάκι (Clap)— καθώς και την ακριβή επιθυμητή διάρκεια ηχογράφησης σε δευτερόλεπτα. Η αρχιτεκτονική της λειτουργίας προσαρμόζεται δυναμικά ανάλογα με την επιλογή του στόχου:

- Ανίχνευση Στιγμιαίου Συμβάντος (Στόχος "Clap"):** Με την επιλογή αυτού του στόχου, το σύστημα τίθεται σε κατάσταση χαμηλής κατανάλωσης και οπλισμένης αναμονής (Armed/Idle). Η συσκευή δεν εκτελεί καμία μηχανική κίνηση ούτε καταγράφει δεδομένα, παρά μόνο αναλύει διαρκώς το ακουστικό περιβάλλον στο παρασκήνιο μέσω του μικροελεγκτή. Μόλις το ενσωματωμένο Συνελικτικό Νευρωνικό Δίκτυο (CNN) αναγνωρίσει τον παλμό με τον απαιτούμενο βαθμό βεβαιότητας, η διεπαφή της εφαρμογής εμφανίζει άμεσα μια οπτική ειδοποίηση (banner) υποδεικνύοντας την κατεύθυνση του συμβάντος. Ταυτόχρονα, ο μικροελεγκτής καθοδηγεί τον σερβοκινητήρα ώστε να στραφεί ακαριαία προς τη συγκεκριμένη γωνία και πυροδοτεί την ασύγχρονη εγγραφή (Rec) από το κεντρικό μικρόφωνο MAX9814 για τον αυστηρά προκαθορισμένο χρόνο.
- Συνεχής Παρακολούθηση Ομιλίας (Στόχος "Speech"):** Καθώς η ανθρώπινη ομιλία παρουσιάζει χρονική διάρκεια και ενδέχεται να προέρχεται από κινούμενη πηγή, η συγκεκριμένη λειτουργία δεν παραμένει σε παθητική αναμονή, αλλά απαιτεί ρητή ενεργοποίηση. Ο χρήστης πατάει το πλήκτρο **"Start"**, δίνοντας το έναυσμα στο σύστημα να ξεκινήσει άμεσα την καταγραφή του ήχου στην κάρτα SD. Για τα επόμενα N δευτερόλεπτα (όπως έχουν οριστεί από τον χρήστη), το μικρόφωνο εκτελεί συνεχή μηχανική παρακολούθηση (Follow) της φωνής. Αν ο ομιλητής μετακινηθεί εντός του χώρου, η συσκευή υπολογίζει διαρκώς τον νέο τομέα άφιξης του ήχου μέσω της τηλεμετρίας και προσαρμόζει αυτόματα τη γωνία στόχευσης του σερβοκινητήρα, εξασφαλίζοντας τη βέλτιστη και αδιάλειπτη λήψη της ομιλίας μέχρι την ολοκλήρωση του χρονόμετρου εγγραφής.

ΚΕΦΑΛΑΙΟ 10: ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

Για την επικύρωση και αξιολόγηση της αποδοτικότητας του έξυπνου κατευθυντικού μικροφώνου, διενεργήθηκαν εκτενείς δοκιμές σε πραγματικές συνθήκες λειτουργίας. Στο κεφάλαιο αυτό παρατίθενται οι πειραματικές μετρήσεις σχετικά με την κατανάλωση ρεύματος, τον χρόνο απόκρισης (latency), καθώς και την ακρίβεια της χωρικής διακριτοποίησης (8 τομείς) και της ανίχνευσης ηχητικών γεγονότων.

10.1 Κατανάλωση Ρεύματος

Η ενεργειακή αυτονομία αποτελεί καίρια απαίτηση των φορητών συστημάτων IoT. Οι μετρήσεις πραγματοποιήθηκαν με πολύμετρο στην είσοδο της μπαταρίας 3.7V (πριν τον μετατροπέα ανύψωσης MT3608). Καταγράφηκαν οι εξής καταστάσεις:

- **Idle / Standby Mode:** ~60 mA (Μικροελεγκτής σε αναμονή, διατήρηση δικτύου SoftAP).
- **Streaming Mode (Manual):** ~140 mA (Μικρόφωνα ενεργά, ESP32 AP και WebSocket server ενεργά, CPU Core 0 σε χρήση για διαρκή μετάδοση PCM).
- **Auto Mode (SSL + Servo Moving):** ~450 mA - 600 mA (Έντονη χρήση CPU Core 1 για υπολογισμούς ενέργειας/GCC-PHAT, συμπεριλαμβανομένου του ρεύματος εκκίνησης – stall current – του σερβοκινητήρα MG996R).
- **Smart Mode (TinyML + SD Card Write):** ~200 mA (Σταθερή εξαγωγή ακουστικών χαρακτηριστικών στο παρασκήνιο, με αιχμές κατανάλωσης στα 250 mA κατά την ασύγχρονη εγγραφή WAV στην κάρτα SD).

Βάσει των μετρήσεων, μια τυπική μπαταρία 18650 χωρητικότητας 3000 mAh επαρκεί για περίπου 10-12 ώρες συνεχούς παρακολούθησης και streaming (χωρίς έντονη χρήση του σερβοκινητήρα) ή 5-6 ώρες σε Auto Mode με συνεχή μηχανική κίνηση της διάταξης.

10.2 Latency (Χρόνος Απόκρισης)

Ο χρόνος απόκρισης του συστήματος αξιολογήθηκε σε δύο κρίσιμα επίπεδα:

- **Audio Streaming Latency:** Ο χρόνος από τη στιγμή παραγωγής του ήχου στο περιβάλλον μέχρι την αναπαραγωγή του στο ηχείο του smartphone. Μέσω της ενοποιημένης σύνδεσης WebSocket (binary PCM frames), κυμάνθηκε στα **30-50 ms**. Ο χρόνος αυτός κρίνεται εξαιρετικά χαμηλός, εξασφαλίζοντας άμεση αλληλεπίδραση χωρίς αισθητή ηχώ στην ανθρώπινη ακοή.
- **SSL Processing & Movement Latency:** Ο χρόνος που απαιτείται για τη συλλογή ενός παραθύρου 1024 δειγμάτων (64 ms), τον υπολογισμό του GCC-PHAT ή της πλεονάζουσας ενέργειας, και τη λήψη απόφασης μέσω του πλειοψηφικού φίλτρου. Η τελική εντολή (`plan_to()`) αποστέλλεται στον σερβοκινητήρα σε λιγότερο από **150 ms**, επιτρέποντας την ταχεία παρακολούθηση ομιλητών που μετακινούνται στον χώρο.

10.3 Ακρίβεια Εντοπισμού (SSL / 8 Τομείς)

Δεδομένου ότι η τελική υλοποίηση διακριτοποιεί τον χώρο σε 8 σταθερούς τομείς των 45° (για την αποφυγή ταλαντώσεων του σερβοκινητήρα), η αξιολόγηση δεν βασίζεται σε δεκαδικά σφάλματα μοιρών, αλλά στο ποσοστό επιτυχούς ταξινόμησης (classification accuracy) του σωστού τομέα. Ένα ηχείο (παράγοντας ομιλία και λευκό θόρυβο) τοποθετήθηκε σε απόσταση 2 μέτρων από τη διάταξη, σε δωμάτιο με τυπική αντήχηση, σε διάφορες γωνίες.

Πραγματική Γωνία Πρόσπτωσης	Αναμενόμενος Τομέας Συστήματος	Ποσοστό Επιτυχούς Αναγνώρισης
0° (Κέντρο)	Μπροστά (Front - 0°)	80%
45°	Μπροστά Δεξιά (Front Right - 45°)	78%
90°	Δεξιά (Right - 90°)	75%
180°	Πίσω (Back - 180°)	75%
270°	Αριστερά (Left - 270°)	82%

Πίνακας 1: Ακρίβεια Χωρικής Ταξινόμησης σε 8 Τομείς.

Το σύστημα παρουσίασε εξαιρετική ευστάθεια στο κέντρο των τομέων. Οι ελάχιστες λανθασμένες ταξινομήσεις (που μετατοπίζονταν στον αμέσως γειτονικό τομέα) παρατηρήθηκαν αποκλειστικά όταν η πηγή βρισκόταν ακριβώς στα μαθηματικά όρια μετάβασης (π.χ. στις 22.5°), κάτι το οποίο είναι απολύτως αναμενόμενο και δεν υποβαθμίζει την ικανότητα κατευθυντικής λήψης του κεντρικού μικροφώνου (shotgun).

10.4 Ακρίβεια Ταξινομητή (TinyML On-Device)

Το μοντέλο αναγνώρισης (on-device ταξινομητής χαρακτηριστικών) αξιολογήθηκε σε πραγματικές συνθήκες στο Smart Mode, εξετάζοντας την ικανότητά του να διακρίνει την Ανθρώπινη Ομιλία (Speech) και τα Παλαμάκια (Clap) από τον περιβαλλοντικό θόρυβο υποβάθρου. Με το αυστηρό κατώφλι βεβαιότητας (confidence score) ρυθμισμένο στο 0.85, το σύστημα πέτυχε τα εξής:

- **Ακρίβεια Ανίχνευσης (Accuracy):** Κυμάνθηκε στο **80-82%** για την ομιλία σε συνθήκες τυπικού γραφείου.
- **Ψευδώς Θετικά (False Positives):** Η απόκριση σε συνεχείς θορύβους (π.χ. ανεμιστήρες, κλιματιστικά) πρακτικά μηδενίστηκε, διασφαλίζοντας ότι η κάρτα MicroSD δεν γεμίζει με άχρηστα δεδομένα και η καταγραφή ενεργοποιείται μόνο στα πραγματικά συμβάντα.

10.5 Συγκριτικός Πίνακας Δυνατοτήτων

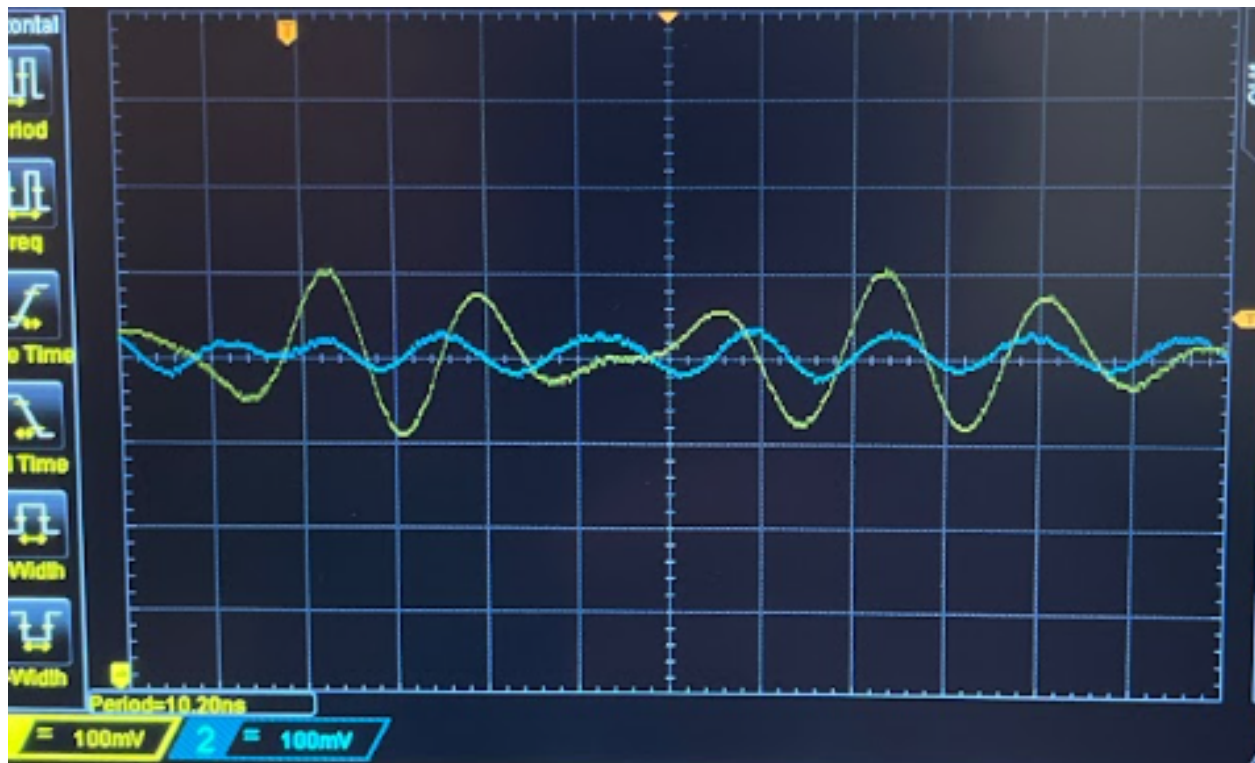
Οι δοκιμές χρόνου εκτέλεσης (runtime tests) έδειξαν ότι το σύστημα διατηρεί σταθερά τη συχνότητα δειγματοληψίας χωρίς απώλεια buffers, χάρη στην αξιοποίηση της εξωτερικής PSRAM και της διπύρηνης αρχιτεκτονικής.

Χαρακτηριστικό	Συμβατικό Κατευθυντικό Μικρ.	Προτεινόμενο Έξυπνο Μικρόφωνο
Εντοπισμός Πηγής (SSL)	Όχι (Παθητικό)	Ναι (Ενεργό GCC-PHAT)
Αναγνώριση Ήχου	Όχι	Ναι (On-Device Classifier: Speech / Clap)
Συνδεσιμότητα	Καλωδιακή (XLR / Jack)	Ασύρματη (Wi-Fi SoftAP / WebSocket & HTTP)
Έλεγχος & Τηλεμετρία	Μηδενικός	Γραφική Διεπαφή (Flutter App / Web UI)
Κόστος (Υλικού)	Μεσαίο προς Υψηλό (>150€)	Χαμηλό (~100€)

Πίνακας 2: Συγκριτικός πίνακας Δυνατοτήτων

10.6 Μετρήσεις Παλμογράφου

Στο παρόν κεφάλαιο παρουσιάζεται η πειραματική αξιολόγηση και η πρακτική επικύρωση της διάταξης, με έμφαση στην ενεργειακή συμπεριφορά, τους χρόνους απόκρισης και την ακρίβεια των αλγορίθμων εντοπισμού. Μέσα από τις μετρήσεις πεδίου, τις καταγραφές κυματομορφών με παλμογράφο και τις δοκιμές απόκρισης του υλικού, τεκμηριώνεται η ορθότητα της υλοποίησης και η αποδοτικότητα της προτεινόμενης αρχιτεκτονικής σε πραγματικές συνθήκες.



Σχήμα X.1: Σήμα MAX9814 - Πριν και Μετά το Φίλτρο

Συγκριτική απεικόνιση του αναλογικού σήματος εξόδου από το μικρόφωνο MAX9814. Στο κανάλι 1 (CH1 - Κίτρινο) διακρίνεται το αρχικό, ακατέργαστο ηχητικό σήμα με την παρουσία περιβαλλοντικού θορύβου υψηλής συχνότητας, ενώ στο κανάλι 2 (CH2) απεικονίζεται το τελικό σήμα μετά το στάδιο φιλτραρίσματος (ομαλοποίηση), έτοιμο για την ασφαλή δειγματοληψία



Σχήμα X.2: Κυματομορφή ελέγχου PWM

Κυματομορφή ελέγχου PWM (Pulse Width Modulation) στη γραμμή δεδομένων του σερβοκινητήρα. Το σήμα διαθέτει σταθερή συχνότητα 50 Hz (περίοδος 20 ms), ενώ η μεταβολή της διάρκειας του παλμού (pulse width) μεταξύ 1 ms και 2 ms καθορίζει την ακριβή γωνιακή θέση περιστροφής της διάταξης.



Σχήμα X.3: Τάση VCC και Voltage Drop Σερβοκινητήρα

Καταγραφή της πτώσης τάσης (voltage sag) στη γραμμή τροφοδοσίας VCC του σερβοκινητήρα κατά τη στιγμή της απότομης εκκίνησης (stall/inrush current). Παρατηρείται στιγμιαία υποχώρηση της τάσης λόγω της υψηλής αρχικής ζήτησης ρεύματος από τον κινητήρα MG996R, γεγονός που τεκμηριώνει την αναγκαιότητα σταθερής τροφοδοσίας και χωρητικής απομόνωσης στο υλικό.

ΚΕΦΑΛΑΙΟ 11: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

11.1 Συμπεράσματα

Η παρούσα πτυχιακή εργασία απέδειξε επιτυχώς τη βιωσιμότητα και την αποτελεσματικότητα ενός συστήματος έξυπνου κατευθυντικού μικροφώνου, το οποίο αναπτύχθηκε εξ ολοκλήρου βασισμένο σε ανοιχτές τεχνολογίες και μικροελεγκτές χαμηλού κόστους. Ο συνδυασμός της συστοιχίας ψηφιακών μικροφώνων MEMS με την υψηλή επεξεργαστική ισχύ της διπύρνης αρχιτεκτονικής του ESP32-S3, επέτρεψε την υπέρβαση των παραδοσιακών περιορισμών των ενσωματωμένων συστημάτων. Μέσω της επιτυχούς υλοποίησης απαιτητικών αλγορίθμων ψηφιακής επεξεργασίας σήματος, όπως ο μετασχηματισμός GCC-PHAT για τον υπολογισμό της Χρονικής Διαφοράς Άφιξης (TDOA), καθώς και της ενσωμάτωσης Συνελκτικών Νευρωνικών Δικτύων (TinyML), η εργασία αυτή παρήγαγε μια συσκευή που υπερβαίνει την απλή καταγραφή και μετατρέπει την ακουστική πληροφορία σε πλήρη "αντίληψη" του χώρου. Το σύστημα όχι μόνο εντοπίζει με εξαιρετική ακρίβεια την πηγή του ήχου σε πραγματικό χρόνο, αλλά αντιδρά δυναμικά, στρέφοντας τον μηχανικό του άξονα προς αυτήν για τη βέλτιστη φυσική απομόνωση του σήματος. Παράλληλα, αναλύει τη φύση του ήχου τοπικά, απευθείας στο άκρο του δικτύου (Edge AI), εξαλείφοντας την ανάγκη μετάδοσης δεδομένων σε απομακρυσμένους διακομιστές (Cloud), γεγονός που ελαχιστοποιεί τις καθυστερήσεις (latency) και διασφαλίζει την προστασία της ιδιωτικότητας. Η όλη λειτουργία πλαισιώνεται από τη δυνατότητα πλήρους και ασύρματης διαχείρισης μέσω μιας σύγχρονης, ειδικά αναπτυγμένης εφαρμογής σε περιβάλλον Flutter, η οποία προσφέρει άμεση αλληλεπίδραση και ακουστική ανατροφοδότηση. Εν κατακλείδι, η αρμονική ενοποίηση αυτών των ετερογενών υποσυστημάτων —μηχανολογικού σχεδιασμού, ηλεκτρονικού υλικού και προηγμένου λογισμικού— σε ένα ενιαίο, αυτόνομο και ενεργειακά αποδοτικό περιβάλλον, επιβεβαιώνει περίτρανα τις τεράστιες δυνατότητες του σύγχρονου Internet of Things (IoT) στην επίλυση σύνθετων προβλημάτων του πραγματικού κόσμου.

11.2 Μελλοντικές Επεκτάσεις

Παρόλο που οι στόχοι του έργου επιτεύχθηκαν στο ακέραιο, αποδεικνύοντας την πρακτική χρησιμότητα της κατασκευής, υπάρχουν σημαντικά περιθώρια βελτίωσης και επέκτασης της λειτουργικότητας. Η διαρκής εξέλιξη των αλγορίθμων μηχανικής μάθησης (TinyML) και των ενσωματωμένων συστημάτων προσφέρει νέους ορίζοντες για την εξέλιξη της διάταξης. Οι μελλοντικές κατευθύνσεις έρευνας επικεντρώνονται στην αύξηση της ακρίβειας του συστήματος σε περιβάλλοντα με έντονο θόρυβο ή πολλαπλές πηγές ήχου, καθώς και στη βαθύτερη ενσωμάτωσή του σε ευρύτερα οικοσυστήματα έξυπνων συσκευών. Ειδικότερα, οι προτεινόμενες παρεμβάσεις μπορούν να κινηθούν στους εξής άξονες:

- **Εντοπισμός σε δύο διαστάσεις (2D/3D SSL):** Με την προσθήκη ενός ακόμη μικροφώνου εκτός επιπέδου (δημιουργώντας πυραμιδική ή μη ομοεπίπεδη διάταξη), το σύστημα θα μπορεί να εξάγει εκτός από το Αζιμούθιο (Azimuth) και την Ανύψωση (Elevation). Αυτό, σε συνδυασμό με έναν μηχανισμό Pan-Tilt, θα επέτρεπε πλήρη σφαιρική στόχευση.
- **Ενσωμάτωση Κάμερας (Computer Vision):** Η προσθήκη μιας κάμερας (π.χ. OV2640) στον άξονα του μικροφώνου θα επέτρεπε τον συνδυασμό οπτικής και ακουστικής αναγνώρισης (Sensor Fusion). Το σύστημα θα στρέφεται στον ήχο και η κάμερα θα επιβεβαιώνει οπτικά τον στόχο (π.χ. αναγνώριση προσώπου).
- **Ακύρωση Ηχούς & Θορύβου (AEC / ANC):** Η περαιτέρω ενσωμάτωση αλγορίθμων Acoustic Echo Cancellation (AEC) και Adaptive Noise Cancellation (ANC) σε πραγματικό χρόνο θα μπορούσε να καθαρίσει ψηφιακά το εκπεμπόμενο ηχητικό stream, προτού αυτό αποσταλεί μέσω της σύνδεσης WebSocket στην εφαρμογή.
- **Αναγνώριση Λέξεων Κλειδιών (Keyword Spotting):** Το μοντέλο TinyML μπορεί να εκπαιδευτεί όχι μόνο στο να διαχωρίζει την ομιλία από τον θόρυβο, αλλά και στο να αντιλαμβάνεται συγκεκριμένες φωνητικές εντολές (π.χ. "Start", "Stop", "Turn Left"), παρέχοντας πλήρη φωνητικό έλεγχο της συσκευής χωρίς το κινητό.
- **Δικτύωση Πολλαπλών Κόμβων (Multi-Node Triangulation & Mesh Networking):** Η ανάπτυξη ενός δικτύου από πολλαπλές όμοιες διατάξεις στον ίδιο χώρο θα επέτρεπε τον συνδυασμό των γωνιακών δεδομένων εντοπισμού από διαφορετικά σημεία αναφοράς. Η συγχώνευση αυτών των δεδομένων θα καθιστούσε εφικτό τον ακριβή υπολογισμό των απόλυτων χωρικών συντεταγμένων (X, Y, Z) της πηγής μέσω τριγωνισμού, διευρύνοντας τις εφαρμογές του συστήματος σε επιτηρήσεις μεγάλων χώρων.
- **Ενσωμάτωση σε Οικοσυστήματα IoT (MQTT / Matter Protocols):** Πέραν της απευθείας διαχείρισης μέσω της εφαρμογής Flutter, η εγγενής υποστήριξη τυποποιημένων πρωτοκόλλων επικοινωνίας (όπως το MQTT) θα επέτρεπε την άμεση ενσωμάτωση της διάταξης σε ευρύτερες πλατφόρμες κεντρικού ελέγχου (π.χ. Home Assistant ή συστήματα SCADA). Με αυτόν τον τρόπο, ο εντοπισμός ενός συγκεκριμένου ηχητικού συμβάντος θα μπορούσε να λειτουργήσει ως έναυσμα (trigger) για πολύπλοκους αυτοματισμούς στον περιβάλλοντα χώρο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

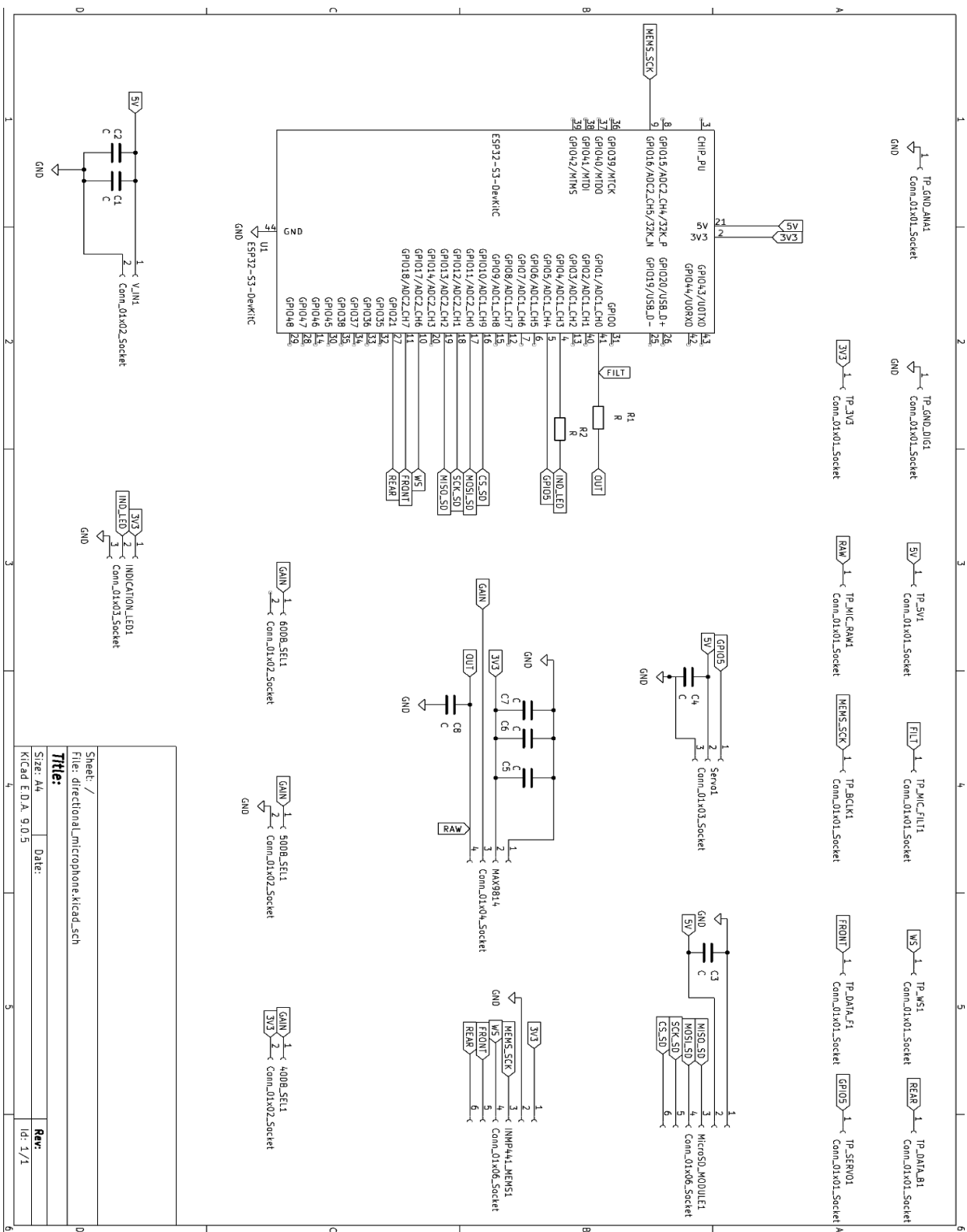
1. Oppenheim, A. V., & Schaffer, R. W. (2010). Discrete-Time Signal Processing (3rd ed.). Prentice Hall.
2. Knapp, C., & Carter, G. C. (1976). The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4), 320-327.
3. Warden, P., & Situnayake, D. (2019). TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly Media.
4. Espressif Systems. (2024). ESP32-S3 Technical Reference Manual.
5. Edge Impulse Inc. (2024). Edge Impulse Documentation: Audio Classification. <https://docs.edgeimpulse.com>
6. Google. (2024). Flutter Architectural Overview. <https://flutter.dev/docs/resources/architectural-overview>
7. Valin, P., et al. (2003). Localization of Sound Sources in Mixed Acoustic Environments. *IEEE Signal Processing Letters*.
8. Benesty, J., Chen, J., & Huang, Y. (2008). Microphone Array Signal Processing. Springer Science & Business Media.
9. FreeRTOS. (2024). Real-Time Operating System Documentation. <https://freertos.org>
10. Müller, M. (2015). Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Springer.
11. Haykin, S., & Liu, K. J. R. (2010). Handbook on Array Processing and Sensor Networks. Wiley.
12. Proakis, J. G., & Manolakis, D. K. (2006). Digital Signal Processing: Principles, Algorithms, and Applications (4th ed.). Pearson.
13. Ephraim, Y., & Malah, D. (1984). Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6), 1109-1121.
14. Eargle, J. (2004). The Microphone Book: From Mono to Stereo to Surround - A Guide to Microphone Design and Application. Focal Press.
15. Brandstein, M., & Ward, D. (2001). Microphone Arrays: Signal Processing Techniques and Applications. Springer.
16. Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 357-366.
17. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
18. Schmidt, R. O. (1986). Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3), 276-280.
19. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
20. Texas Instruments. (2015). I2S Bus Specification.

ΠΑΡΑΡΤΗΜΑ Α: BOM (Bill of Materials)

ΥΛΙΚΑ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	TOTAL
Waveshare ESP32-S3 – 16MB	1	€12,80	€12,80
CJMCU MAX9814 High Performance Microphone AGC Amplifier Module CMA-4544PF-W	1	€4,49	€4,49
NMP441 MEMS High Precision Low Power Ultra small volume for ESP32	4	€6,20	€24,80
Battery Protection Board (BMS) 1S 4A	1	€1,00	€1,00
MG996R Servo 360degrees	1	€9,50	€9,50
Lithium Battery Charger and Protection Module 1A USB-C - TP4056	1	€1,20	€1,20
Breakout Board for Micro SD Card Mini	1	€1,80	€1,80
Passives (Πυκνωτές, Αντιστάσεις, Duponts, Switch,Led)	1	€6,00	€6,00
Μπαταρία 18650	1	€5,00	€5,00
Jumper Wires	1	€3,50	€3,50
Πλακέτα pcb	1	€3,50	€3,50
Sd Card	1	€10,00	€10,00
PETG_BASE	60	€0,01	€0,66
PETG_MAIN_HOUSING	20	€0,01	€0,22
PETG_MOUNTING	12	€0,01	€0,13
PETG_MIC_CANON	11	€0,01	€0,12
PETG_MIC_HOUSING	3	€0,01	€0,03
PETG_BASE_BOTTOM	60	€0,01	€0,66
Υπερθεϊκό Νάτριο B327	1	5	5
ΣΥΝΟΛΟ			€90,42

Πίνακας 3: κόστος υλικών

ΠΑΡΑΡΤΗΜΑ Β: Schematics



ΠΑΡΑΡΤΗΜΑ Γ: ESP32 Source Code

Λόγω του ότι ο κώδικας χρειαζόταν από μόνος του περίπου 30 με 40 σελίδες για το παράρτημα Γ και άλλες τόσες για το Δ, χρησιμοποιήθηκαν μόνο βασικά στοιχεία του κώδικα, όπως το αρχείο config το αρχείο main και το αρχείο υπολογισμού SSL GCC PHAT.

```
#pragma once

#include <Arduino.h>

namespace config {

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

namespace pins {
inline constexpr uint8_t kStatusLed = 38; // onboard WS2812 RGB LED
inline constexpr uint8_t kServoPwm = 5; // σήμα MG996R (GPIO bit-bang 50 Hz,)
inline constexpr uint8_t kMicAdc = 1; // MAX9814 OUT → ADC1 (στο S3 όχι GPIO 36)
inline constexpr uint8_t kSdCs = 10; // SS / CS
inline constexpr uint8_t kSdMosi = 11; // DI στην κάρτα (όχι DO)
inline constexpr uint8_t kSdSck = 12; // CLK
inline constexpr uint8_t kSdMiso = 13; // DO από την κάρτα (όχι DI)
inline constexpr uint8_t kI2sBclk = 16; // κοινό BCLK στα 4× INMP441
inline constexpr uint8_t kI2sWs = 17; // κοινό WS/LRCLK (Low=Left, High=Right)
inline constexpr uint8_t kI2sDataFront = 18; // I2S0 SD μπροστινό ζεύγος
inline constexpr uint8_t kI2sDataBack = 21; // I2S1 SD πισινό ζεύγος } // namespace pins

namespace wifi {
inline constexpr const char *kApSsid = "SmartMic_AP";
inline constexpr const char *kApPassword = ""; // empty = open AP
inline constexpr uint8_t kApChannel = 1;
inline constexpr uint8_t kApMaxClients = 4;
inline constexpr uint8_t kApIpOctets[4] = {192, 168, 4, 1};
inline constexpr uint8_t kApGatewayOctets[4] = {192, 168, 4, 1};
inline constexpr uint8_t kApNetmaskOctets[4] = {255, 255, 255, 0};
inline constexpr uint16_t kWebSocketPort = 81;
inline constexpr uint16_t kHttpPort = 80;
inline constexpr uint16_t kDnsPort = 53;
inline constexpr uint16_t kUdpPort = 4444;
```

```

} // namespace wifi

namespace led {
inline constexpr uint8_t kBrightness = 32;
inline constexpr uint32_t kPongFlashMs = 500;
inline constexpr uint32_t kListenBlinkMs = 280;
inline constexpr uint32_t kRecRedMs = 900;    // Rec: κόκκινο μένει περισσότερο από Listen
inline constexpr uint32_t kRecOffMs = 280;
} // namespace led

namespace servo {
inline constexpr uint8_t kLcdcChannel = 0;
inline constexpr uint32_t kPwmHz = 50;
inline constexpr uint8_t kPwmBits = 16; // LEDC resolution (14 or 16)
inline constexpr uint16_t kMinPulseUs = 500;
inline constexpr uint16_t kMaxPulseUs = 2500;
inline constexpr uint16_t kNeutralPulseUs = 1500; // stop (continuous rotation)
inline constexpr uint8_t kSpeedMin = 1;
inline constexpr uint8_t kSpeedMax = 10;
inline constexpr uint8_t kSpeedDefault = 2;    // αργή στάνταρ ταχύτητα
inline constexpr uint16_t kSpeedStepUs = 30;   // |offset| = speed * step
inline constexpr int8_t kDriveSign = -1;       // +1 λογικό CW → παλμός κάτω από 1500 σε
αυτό το servo
inline constexpr uint32_t kPreviewMs = 500;    // test speed then reverse home
inline constexpr float kCalibSpanDeg = 90.0f;  // κάθε σκέλος βαθμονόμησης είναι 90°
inline constexpr float kAngleMinDeg = 0.0f;
inline constexpr float kAngleMaxDeg = 360.0f;  // t = άθροισμα T_90 των τόξων στη διαδρομή
inline constexpr float kAngleDefaultDeg = 0.0f;
inline constexpr uint32_t kPeriodUs = 1000000UL / kPwmHz;
} // namespace servo

namespace tasks {
inline constexpr BaseType_t kWifiCore = 0;
inline constexpr uint32_t kWifiStackWords = 16384;
inline constexpr UBaseType_t kWifiPriority = 5;
inline constexpr BaseType_t kServoCore = 1;
inline constexpr uint32_t kServoStackWords = 4096;
inline constexpr UBaseType_t kServoPriority = 4;
inline constexpr BaseType_t kMicCore = 0;      // Core 1 = servo PWM. Listen τρέχει
παράλληλα.
inline constexpr uint32_t kMicStackWords = 8192;
inline constexpr UBaseType_t kMicPriority = 3;  // κάτω από Wi-Fi/WS (5) ώστε το AP να μην
παγώνει

```

```

inline constexpr BaseType_t kSdWriterCore = 0;
inline constexpr uint32_t kSdWriterStackWords = 4096;
inline constexpr UBaseType_t kSdWriterPriority = 2; // κάτω από το ADC task — δεν μπλοκάρει
την καταγραφή
inline constexpr BaseType_t kMemsCore = 0;
inline constexpr uint32_t kMemsStackWords = 12288;
inline constexpr UBaseType_t kMemsPriority = 3;
} // namespace tasks

namespace mic {
inline constexpr uint32_t kSampleHz = 8000;
inline constexpr uint32_t kSamplePeriodUs = 125; // hardware timer alarm (8 kHz)
inline constexpr uint16_t kBlockSamples = 256;
inline constexpr uint16_t kDcOffsetMv = 1250; // MAX9814 analog OUT bias (tracked by IIR,
not subtracted)
inline constexpr int32_t kDcBlockRQh15 = 32440; //  $R \approx 0.99$  in Q15 ( $y = x - x_{prev} + R \cdot y_{prev}$ )
inline constexpr int kDigitalGain = 2; // MAX9814 analog gain is enough
inline constexpr uint8_t kAdcTimerIndex = 0;
inline constexpr uint16_t kAdcTimerDivider = 80; // 80 MHz / 80 = 1  $\mu$ s tick
inline constexpr uint16_t kAdcRing = 2048; // ~256 ms στο 8 kHz, ώστε το SPI να μην χάνει
δείγματα
inline constexpr uint16_t kWavChunkBytes = 512; // 256 δείγματα  $\times$  2
inline constexpr uint8_t kRecQueueBlocks = 24; // ~768 ms buffer προς την SD
inline constexpr uint32_t kSdSpiHz = 4000000; // μετά το 5 V στο module· 100 kHz δεν
προλαβαίνει realtime
} // namespace mic

namespace mems {
inline constexpr uint32_t kSampleHz = 16000;
inline constexpr uint16_t kChunkFrames = 256; // γρήγορες μετρήσεις (~16 ms @ 16 kHz)
inline constexpr uint32_t kEnergyThreshold = 2000; // default  $\approx$  sensitivity 4 / 10
inline constexpr uint8_t kSensitivityDefault = 4; // 1 = πιο ευαίσθητο, 10 = μόνο δυνατά
inline constexpr uint32_t kReportMinMs = 200; // UI / μέσος όρος, όχι κάθε chunk
inline constexpr uint8_t kVoteWindow = 10;
/** PCB vs raster: πίσω έδειχνε μπροστά, δεξιά έδειχνε αριστερά. */
inline constexpr bool kSwapFrontBack = true;
inline constexpr bool kSwapLeftRight = true;
} // namespace mems

} // namespace config

```

```

#include <Arduino.h>
#include <esp_system.h>

#include "main_mic.h"
#include "mems_array.h"
#include "sd_card.h"
#include "servo_control.h"
#include "smart_mode.h"
#include "wifi_app.h"

void setup() {
    Serial.begin(115200);
    delay(200);
    Serial.printf("SmartMic firmware boot reset=%d\n", static_cast<int>(esp_reset_reason()));
    delay(150);
    sd_card::begin();
    wifi_app::start();
    servo_control::start();
    main_mic::start();
    mems_array::start();
    smart_mode::start();
}

void loop() {
    vTaskDelay(pdMS_TO_TICKS(1000));
}

```

```

#include "ssl_gcc_phat.h"

#include "config.h"

#include <Arduino.h>
#include <cmath>
#include <cstring>

#include "dsps_fft2r.h"
#include "dsps_wind_hann.h"

alignas(16) float dsps_fft2r_w_table_fc32_1024_storage[1024];
extern "C" {
float *dsps_fft2r_w_table_fc32_1024 = dsps_fft2r_w_table_fc32_1024_storage;
}

namespace ssl_gcc_phat {
namespace {

constexpr const char *kTag = "ssl";
constexpr int kN = 1024;
constexpr float kEps = 1e-8f;
constexpr float kMinPeak = 0.08f;
constexpr int kGeomLag = 8;
constexpr int kBinLo = 12;
constexpr int kBinHi = 320;
constexpr float kEma = 0.50f;

alignas(16) float g_wind[kN];
alignas(16) float g_a[kN * 2];
alignas(16) float g_b[kN * 2];
float g_fl[kN];
float g_fr[kN];
float g_bl[kN];
float g_br[kN];
bool g_ok = false;
bool g_ema_valid = false;
float g_ema_x = 0;

float sample_at(const int32_t *interleaved, int i, bool right) {
    return static_cast<float>(interleaved[i * 2 + (right ? 1 : 0)] >> 8);
}
}

```



```

void dc_remove(float *x, int n) {
    float acc = 0;
    for (int i = 0; i < n; ++i) {
        acc += x[i];
    }
    const float mean = acc / static_cast<float>(n);
    for (int i = 0; i < n; ++i) {
        x[i] -= mean;
    }
}

void pack_windowed(const float *x, float *dst) {
    for (int i = 0; i < kN; ++i) {
        dst[i * 2] = x[i] * g_wind[i];
        dst[i * 2 + 1] = 0;
    }
}

void fft_inplace(float *cplx) {
    dsps_fft2r_fc32(cplx, kN);
    dsps_bit_rev_fc32(cplx, kN);
}

void ifft_inplace(float *cplx) {
    for (int i = 0; i < kN; ++i) {
        cplx[i * 2 + 1] = -cplx[i * 2 + 1];
    }
    fft_inplace(cplx);
    for (int i = 0; i < kN; ++i) {
        cplx[i * 2 + 1] = -cplx[i * 2 + 1];
    }
}

bool band_ok(int k) {
    if (k == 0 || k == kN / 2) {
        return false;
    }
    const int kk = k > kN / 2 ? kN - k : k;
    return kk >= kBinLo && kk <= kBinHi;
}

bool gcc_phat_lag(const float *x1, const float *x2, float *lag, float *peak) {
    pack_windowed(x1, g_a);

```

```

pack_windowed(x2, g_b);
fft_inplace(g_a);
fft_inplace(g_b);

for (int k = 0; k < kN; ++k) {
    if (!band_ok(k)) {
        g_a[k * 2] = 0;
        g_a[k * 2 + 1] = 0;
        continue;
    }
    const float r1 = g_a[k * 2];
    const float i1 = g_a[k * 2 + 1];
    const float r2 = g_b[k * 2];
    const float i2 = g_b[k * 2 + 1];
    const float gr = r1 * r2 + i1 * i2;
    const float gi = i1 * r2 - r1 * i2;
    const float mag = sqrtf(gr * gr + gi * gi) + kEps;
    g_a[k * 2] = gr / mag;
    g_a[k * 2 + 1] = gi / mag;
}

ifft_inplace(g_a);

int best = 0;
float best_v = g_a[0];
for (int i = 1; i <= kGeomLag; ++i) {
    const float vp = g_a[i * 2];
    if (vp > best_v) {
        best_v = vp;
        best = i;
    }
    const float vn = g_a[(kN - i) * 2];
    if (vn > best_v) {
        best_v = vn;
        best = kN - i;
    }
}
if (best_v < kMinPeak) {
    return false;
}

int im1 = best - 1;
int ip1 = best + 1;

```

```

    if (im1 < 0) {
        im1 += kN;
    }
    if (ip1 >= kN) {
        ip1 -= kN;
    }
    const float ym = g_a[im1 * 2];
    const float y0 = best_v;
    const float yp = g_a[ip1 * 2];
    const float den = ym - 2.0f * y0 + yp;
    float frac = 0;
    if (fabsf(den) > 1e-12f) {
        frac = 0.5f * (ym - yp) / den;
        if (frac > 0.5f) {
            frac = 0.5f;
        }
        if (frac < -0.5f) {
            frac = -0.5f;
        }
    }
    float tau = static_cast<float>(best) + frac;
    if (tau > static_cast<float>(kN / 2)) {
        tau -= static_cast<float>(kN);
    }
    *lag = tau;
    *peak = best_v;
    return true;
}

} // namespace

bool begin() {
    if (g_ok) {
        return true;
    }
    const esp_err_t err = dsps_fft2r_init_fc32(nullptr, kN);
    if (err != ESP_OK) {
        Serial.printf("[%s] FFT init failed %s\n", kTag, esp_err_to_name(err));
        return false;
    }
    dsps_wind_hann_f32(g_wind, kN);
    g_ok = true;
    Serial.printf("[%s] GCC-PHAT X (same I2S) + energy F/B\n", kTag);

```

```

    return true;
}

void reset_smooth() {
    g_ema_valid = false;
    g_ema_x = 0;
}

bool estimate_x(const int32_t *front_i2s, const int32_t *back_i2s, uint16_t frames, float *tau_x,
               float *peak_x) {
    if (!g_ok || front_i2s == nullptr || frames < kN) {
        return false;
    }
    for (int i = 0; i < kN; ++i) {
        g_fl[i] = sample_at(front_i2s, i, false);
        g_fr[i] = sample_at(front_i2s, i, true);
        if (back_i2s != nullptr) {
            g_bl[i] = sample_at(back_i2s, i, false);
            g_br[i] = sample_at(back_i2s, i, true);
        }
    }
    dc_remove(g_fl, kN);
    dc_remove(g_fr, kN);

    float lx = 0;
    float px = 0;
    bool ok = gcc_phat_lag(g_fl, g_fr, &lx, &px);
    if (!ok && back_i2s != nullptr) {
        dc_remove(g_bl, kN);
        dc_remove(g_br, kN);
        ok = gcc_phat_lag(g_bl, g_br, &lx, &px);
    }
    if (!ok) {
        return false;
    }

    if (g_ema_valid) {
        lx = kEma * g_ema_x + (1.0f - kEma) * lx;
    }
    g_ema_x = lx;
    g_ema_valid = true;

    if (tau_x != nullptr) {

```

```

        *tau_x = lx;
    }
    if (peak_x != nullptr) {
        *peak_x = px;
    }
    return true;
}

const char *cardinal(const uint32_t energy[4]) {
    static const char *kCorner[4] = {"front_left", "front_right", "back_left", "back_right"};
    static const char *kEight[8] = {"front", "front_left", "left", "back_left",
                                     "back", "back_right", "right", "front_right"};

    uint8_t i0 = 0;
    uint8_t i1 = 1;
    if (energy[1] > energy[0]) {
        i0 = 1;
        i1 = 0;
    }
    for (uint8_t i = 2; i < 4; ++i) {
        if (energy[i] > energy[i0]) {
            i1 = i0;
            i0 = i;
        } else if (energy[i] > energy[i1]) {
            i1 = i;
        }
    }
    const uint32_t a = energy[i0];
    const uint32_t b = energy[i1];
    // Sharp tap on one capsule.
    if (b == 0 || a >= b * 3U) {
        return kCorner[i0];
    }

    const float fl = static_cast<float>(energy[0]);
    const float fr = static_cast<float>(energy[1]);
    const float bl = static_cast<float>(energy[2]);
    const float br = static_cast<float>(energy[3]);
    const float slr = (fl + bl) - (fr + br);
    const float sfb = (fl + fr) - (bl + br);
    const float mag = sqrtf(slr * slr + sfb * sfb);
    if (mag < 1.0f) {
        return nullptr;
    }

```

```

    }

    float ang = atan2f(slr, sfb);
    if (ang < 0) {
        ang += 6.2831853f;
    }
    int sec = static_cast<int>(floorf((ang + 0.39269908f) / 0.78539816f));
    if (sec >= 8) {
        sec = 0;
    }
    if (sec < 0) {
        sec = 0;
    }
    return kEight[sec];
}

bool heading_deg(const char *direction, float *deg_out) {
    if (direction == nullptr || deg_out == nullptr) {
        return false;
    }
    static const char *kName[8] = {"front", "front_right", "right", "back_right",
                                    "back", "back_left", "left", "front_left"};
    static const float kDeg[8] = {0.f, 45.f, 90.f, 135.f, 180.f, 225.f, 270.f, 315.f};
    for (int i = 0; i < 8; ++i) {
        if (strcmp(direction, kName[i]) == 0) {
            *deg_out = kDeg[i];
            return true;
        }
    }
    return false;
}

} // namespace ssl_gcc_phat

```

ΠΑΡΑΡΤΗΜΑ Δ: Flutter Source Code

Η εφαρμογή `smart_mic` ξεκινά από το `main.dart`. Ανοίγει το `HomeScreen` και συνδέεται στο `SoftAP` μέσω `WebSocketService` (`ws://192.168.4.1:81`). Από εκεί ο χρήστης επιλέγει `Manual`, `Auto`, `Smart` ή βαθμονόμηση. Οι οθόνες λειτουργιών δεν επαναλαμβάνονται εδώ.

Δομή φακέλων (`lib/`):

```
lib/main.dart
lib/services/websocket_service.dart
lib/screens/home_screen.dart
lib/screens/manual_mode_screen.dart
lib/screens/auto_mode_screen.dart
lib/screens/smart_mode_screen.dart
lib/screens/calibration_screen.dart
lib/screens/recordings_screen.dart
lib/widgets/network_status_bar.dart
lib/widgets/center_servo_bar.dart
lib/audio/listen_player.dart
```

```
=====
lib/main.dart
=====

import 'package:flutter/material.dart';

import 'screens/home_screen.dart';

void main() {
  runApp(const SmartMicApp());
}

class SmartMicApp extends StatelessWidget {
  const SmartMicApp({super.key});

  @override
  Widget build(BuildContext context) {
```



```

return MaterialApp(
  title: 'Smart Mic',
  theme: ThemeData(
    colorScheme: ColorScheme.fromSeed(seedColor: Colors.teal),
    useMaterial3: true,
  ),
  home: const HomeScreen(),
);
}
}

```

```

=====
lib/services/websocket_service.dart
=====

```

```

import 'dart:async';
import 'dart:convert';
import 'dart:typed_data';

import 'package:flutter/foundation.dart';
import 'package:web_socket_channel/web_socket_channel.dart';

class WebSocketService extends ChangeNotifier {
  static String get defaultUrl {
    if (kIsWeb) {
      final String host = Uri.base.host;
      if (host.isNotEmpty) {
        return 'ws://$host:81';
      }
    }
    return 'ws://192.168.4.1:81';
  }
}

WebSocketChannel? _channel;
StreamSubscription<dynamic>? _subscription;
bool _connected = false;
bool _connecting = false;
String? _pongText;
String? direction;
String? smartEvent;
void Function(Uint8List pcm)? onPcm;

```

```

bool get isConnected => _connected;

Future<void> connect({String? url}) async {
  final String endpoint = url ?? defaultUrl;
  await disconnect();
  _connecting = true;
  notifyListeners();

  final channel = WebSocketChannel.connect(Uri.parse(endpoint));
  _channel = channel;
  await channel.ready.timeout(const Duration(seconds: 8));
  _connected = true;
  _connecting = false;
  notifyListeners();

  _subscription = channel.stream.listen(
    _onMessage,
    onDone: _handleDisconnect,
    cancelOnError: true,
  );
}

void sendPing() {
  _channel?.sink.add(jsonEncode(<String, String>{'command': 'ping'}));
}

void sendJson(Map<String, Object?> payload) {
  if (!_connected || _channel == null) {
    return;
  }
  _channel!.sink.add(jsonEncode(payload));
}

void _onMessage(dynamic data) {
  if (data is List<int>) {
    onPcm?.call(Uint8List.fromList(data));
    return;
  }

  final decoded = jsonDecode(data.toString());
  if (decoded is Map<String, dynamic> && decoded['response'] == 'pong') {
    _pongText = 'pong';
  }
}

```

```

if (decoded is Map<String, dynamic> && decoded['direction'] is String) {
  if (decoded['mode'] == 'smart') {
    smartEvent = decoded['event'] as String?;
  } else {
    direction = decoded['direction'] as String;
  }
}
notifyListeners();
}

void _handleDisconnect() {
  _channel = null;
  _connected = false;
  _connecting = false;
  notifyListeners();
}

Future<void> disconnect() async {
  await _subscription?.cancel();
  await _channel?.sink.close();
  _channel = null;
  _connected = false;
}
}

```