ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σύστημα διαχείρισης με χρήση φωνητικών εντολών»

Φοιτητής

Προβατίδης Αναστάσιος 2019143

Επιβλέπων

Δρ. Κυριάκος Τσιακμάκης

Σεπτέμβριος 2025

Σύστημα διαχείρισης με χρήση φωνητικών εντολών

Κωδικός: 25146

Φοιτητής: Αναστάσιος Προβατίδης

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 06-03-2025

Ημερομηνία περάτωσης Π.Ε. 09-09-2025

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

Περίληψη

Η παρούσα Διπλωματική εργασία πραγματεύεται την ανάπτυξη ενός IoT συστήματος φωνητικού ελέγχου για την απομακρυσμένη διαχείριση οικιακών συσκευών. Η αρχιτεκτονική του βασίζεται στην χρήση ενός μικροϋπολογιστή Raspberry Pi 4,ο οποίος εκτελεί το λογισμικό MVF και διασυνδέεται με το πρότυπο κύκλωμα οδήγησης Maestro VF. Μέσω του συστήματος οι χρήστες έχουν την δυνατότητα να αλληλεπιδρούν φωνητικά για την παραμετροποίηση συσκευών με λειτουργίες εισόδου/εξόδου (I/O), τόσο σε πραγματικό χρόνο όσο και προγραμματισμένα μέσω χρονοδιακόπτη. Η διασύνδεση των συσκευών πραγματοποιείται αποκλειστικά μέσω του Maestro VF το οποίο υποστηρίζει την εγκατάσταση τριών φορτίων εκ των οποίων δύο τροφοδοτούνται με εναλλασσόμενη τάση (220 V) και ένα με συνεχή (12 V PWM). Η αρχή λειτουργίας του συστήματος περιλαμβάνει την μετάφραση των φωνητικών εντολών που λαμβάνονται από τον χρήστη σε ηλεκτρονικά σήματα (0 – 3.3 V) τα οποία προωθούνται μέσω των θυρών GPIO του Raspberry Pi στα αρμόδια υποκυκλώματα του Maestro VF για το έλεγχο των εκάστοτε εξόδων. Συνολικά το έργο αποτελεί ένα υβριδικό μοντέλο οικιακού βοηθού και έξυπνου συστήματος το οποίο αφενός αποσκοπεί στην διευκόλυνση της καθημερινότητας και στην υποστήριξη ατόμων με κινητικές δυσκολίες και αφετέρου στην ανάδειξη των δυνατοτήτων συνδυασμού IoT και αναγνώρισης φυσικής ομιλίας ως πρακτική λύση στον οικιακό αυτοματισμό παρέχοντας προσβασιμότητα, αυτονομία και προοπτικές μελλοντικές εξέλιξης.

«Management System Using Voice Commands»

Abstract

The following thesis tackles the development of an IoT voice-controlled system for the remote management of household devices. This architecture is based on the use of a Raspberry Pi 4 microcomputer, which runs the MVF software and is connected to the prototype driving circuit Maestro VF. Through the system, users are able to interact via voice commands for the configuration of devices with input/output (I/O) functionalities, both in real time and through scheduled tasks using a timer. The interconnections of the devices is carried out exclusively via the Maestro VF, which supports the installation of three loads, two powered by alternating current (220 V) and one powered by direct current (12 V PWM).The operating principle of the system involves the translation of user voice commands into electronic signals (0-3.3V),which are then routed through the GPIO ports of the Raspberry Pi to the corresponding control circuits of the Maestro VF in order to drive respective outputs. Overall , this project represents a hybrid model of a home assistant and smart system, aiming on the one hand to facilitate daily life and support people with mobility difficulties, and on the other hand to highlight the potential of combining IoT and natural speech recognition as a practical solution for home automation, offering accessibility ,autonomy and perspectives for future development.

Ευχαριστίες

Θέλω να ευχαριστήσω τους γονείς μου και τους φίλους μου για την αμέριστη στήριξη ,υπομονή και αγάπη τους, καθώς και τον καθηγητή μου Κ. Τσιακμάκη για την πολύτιμη καθοδήγηση, τις γνώσεις και την ενθάρρυνση που μου προσέφερε καθόλη την διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

ΔΕ	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο της Ελλάδος
ΚΜΕ	Κεντρική Μονάδα Επεξεργασίας
ΣΑΕ	Συστήματα Αυτομάτου Ελέγχου
ASR	Automatic Speech Recognition
COAP	Constrained Application Protocol
DIY	Do It Yourself
DRC	Design Rule Check
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GPIO	General Purpose Input/Output
HMM	Hidden Markov Model
IDE	Integrated Development Environment
IoT	Internet of Things
MFCC	Mel-Frequency Cepstral Coefficients
MQTT	Message Queuing Telemetry Transport
NLG	Natural Language Generation
NLU	Natural Language Understanding
NLP	Natural Language Processing
OSI	Open Systems Interconnection

PCB	Printed Circuit Board
PWM	Pulse Width Modulation
STT	Speech To Text
TTS	Text To Speech
VUI	Voice User Interface

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η ραγδαία ανάπτυξη της τεχνολογίας τα τελευταία χρόνια έχει αλλάξει ριζικά τον τρόπο με τον οποίο οι άνθρωποι επικοινωνούν και αλληλεπιδρούν με τα συστήματα και τις συσκευές που τους περιβάλλουν. Η διάδοση του Διαδικτύου τον πραγμάτων (IoT) και η ενσωμάτωση τεχνολογιών τεχνητής νοημοσύνης προσέφεραν νέες δυνατότητες σε τομείς που εκτείνονται από την βιομηχανία και τις μεταφορές έως και την καθημερινή οικιακή χρήση. Στα σύγχρονα σπίτια, οι συσκευές δεν αποτελούν πλέον απομονωμένα στοιχεία αλλά τμήματα ενός ευρύτερου δικτύου που επικοινωνεί συνεχώς και ανταλλάσσει δεδομένα, συμβάλλοντας στην δημιουργία «έξυπνων» περιβαλλόντων με αυξημένη λειτουργικότητα.

Μέσα σε αυτό το πλαίσιο, οι φωνητικοί βοηθοί έχουν αναδειχθεί σε κρίσιμή τεχνολογία καθώς διευκολύνουν την πρόσβαση του χρήστη σε πληθώρα λειτουργιών χωρίς την ανάγκη πολύπλοκων χειριστηρίων ή εξειδικευμένων γνώσεων. Η φωνή ως φυσικός τρόπος επικοινωνίας καθιστά την αλληλεπίδραση πιο άμεση και ανθρώπινη, γεγονός που εξηγεί την εντυπωσιακή εξάπλωση των φωνητικών εφαρμογών σε τομείς όπως η υγεία, η εκπαίδευση, η ψυχαγωγία και η ασφάλεια. Χαρακτηριστικό είναι ότι πολλές οικιακές συσκευές έχουν σχεδιαστεί εξαρχής για να ενσωματώνονται με υπηρεσίες όπως η Amazon Alexa ή Apple Siri.

Η παρούσα ΔΕ επικεντρώνεται στην σχεδίαση και ανάπτυξη ενός υβριδικού συστήματος οικιακού αυτοματισμού που αξιοποιεί την φωνητική αλληλεπίδραση ως κύριο μέσο ελέγχου. Η εφαρμογή βασίζεται στον μικροϋπολογιστή Raspberry Pi 4, ο οποίος έχει αναλάβει την διαχείριση δύο φορτιών εναλλασσόμενου ρεύματος και μίας ταινία RGB LED. Το σύστημα είναι σε θέση να εκτελεί τόσο βασικές λειτουργίες όπως η ενεργοποίηση/απενεργοποίηση συσκευών, όσο και πιο σύνθετες διεργασίες όπως ο προγραμματισμός χρονοδιακοπτών και η ρύθμιση φωτισμού. Με τον τρόπο αυτό προσφέρει ένα πιο ολοκληρωμένο πλαίσιο λειτουργικότητας που μπορεί να ανταποκριθεί σε διαφορετικές απαιτήσεις χρηστών.

Η αρχιτεκτονική του βασίζεται σε συνδυασμό τοπικών υπολογιστικών πόρων και υπηρεσιών νέφους, μέσω των οποίων εκτελούνται οι διαδικασίες φωνητικής αναγνώρισης και σύνθεση ομιλίας. Με αυτόν τον τρόπο εξασφαλίζεται φυσική και άμεση επικοινωνία με τον χρήστη, μικρός χρόνος απόκρισης και δυνατότητα προσαρμογής του συστήματος σε διάφορα σενάρια. Παράλληλα η ανάπτυξη ειδικής πλακέτας τυπωμένου κυκλώματος PCB ενισχύει την ασφάλεια και την αξιοπιστία της διασύνδεσης με τα ηλεκτρικά φορτία, επιτρέποντας την εύκολή μελλοντική επέκταση του συστήματος με επιπλέον λειτουργίες.

1.2 Σύγχρονες εφαρμογές και προκλήσεις

Η έμφυτη ανάγκη του ανθρώπου να αυτοματοποιήσει τις λειτουργίες της καθημερινής του ζωής υπήρξε καθοριστικός παράγοντας στην εξέλιξη της βιομηχανίας και των τεχνολογιών πληροφορικής. Από τις πρώτες βιομηχανικές μηχανές μέχρι τα σύγχρονα πληροφοριακά συστήματα, η αναζήτηση τρόπων μείωσης της ανθρώπινης προσπάθειας οδήγησε σε τεχνολογίες που βελτίωσαν θεαματικά την παραγωγικότητα και την ποιότητα ζωής.

Η φωνή ως το βασικότερο μέσο αλληλεπίδρασης μεταξύ των ανθρώπων αποτέλεσε την βάση για την ανάπτυξη εφαρμογών φωνητικών βοηθών, οι οποίες αξιοποιούν τεχνολογίες αναγνώρισης και κατανόησης φυσικής ομιλίας με στόχο την εκτέλεση ενεργειών. Η αυξανόμενη δημοφιλία των φωνητικών βοηθών σε συνδυασμό με την παράλληλη ανάπτυξη των συστημάτων οικιακού αυτοματισμού, οδήγησε στην δημιουργία «έξυπνων» συστημάτων που δίνουν την δυνατότητα στον χρήστη να αλληλεπιδρά με συσκευές και υπηρεσίες με τρόπο άμεσο, φυσικό και αποτελεσματικό. Οι εξελίξεις αυτές δεν περιορίζονται μόνο στον οικιακό χώρο αλλά εκτείνονται και σε τομείς όπως η υγεία η εκπαίδευσή και η βιομηχανική παραγωγή.

Χαρακτηριστικά παραδείγματα τέτοιων εφαρμογών αποτελούν οι πλατφόρμες Amazon Alexa, Home Assistant και Apple Siri, οι οποίες ενσωματώνονται σε πληθώρα συσκευών όπως τα κινητά τηλέφωνα, οι έξυπνες λάμπες , οι ρομποτικές σκούπες τα συστήματα ασφαλείας κλπ. Μέσω αυτών, ο χρήστης μπορεί να εκτελεί απλές λειτουργίες οικιακού αυτοματισμού όπως η ενεργοποίηση/απενεργοποίηση συσκευών, η ρύθμιση φωτισμού αλλά και πιο σύνθετες όπως ο απομακρυσμένος έλεγχος και ο χρονοπρογραμματισμός ενεργειών.

Αναμφίβολά η συμβολή αυτών των συστημάτων στην βελτιστοποίηση της καθημερινότητας είναι καθοριστική. Ωστόσο η παρουσία τους συνοδεύεται και από ουσιαστικές προκλήσεις. Μεγαλύτερα εξ αυτών αποτελούν τα ζητήματα ασφαλείας και προστασίας προσωπικών δεδομένων, καθώς η συλλογή φωνητικών πληροφοριών μπορεί σε περίπτωση διαρροής να εγκυμονεί σοβαρούς κινδύνους για τον χρήστη. Επιπλέον ανακύπτουν ζητήματα αξιοπιστίας λόγω των γλωσσικών περιορισμών που φέρει κάθε εφαρμογή, χαρακτηριστικό παράδειγμα συνίσταται στην πλειονότητα των εφαρμογών που δεν υποστηρίζουν φωνητικά μοντέλα όπως για παράδειγμα της Ελληνικής γλώσσας περιορίζοντας έτσι την προσβασιμότητα.

1.3 Στόχοι έργου

Στόχος της παρούσας εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου και λειτουργικού συστήματος οικιακού αυτοματισμού, το οποίο θα υποστηρίζει την απομακρυσμένη και φωνητικά καθοδηγούμενη εκτέλεση καθημερινών λειτουργιών. Το όραμα που διέπει την υλοποίηση εστιάζει στη δημιουργία μίας εύκολα προσβάσιμης και ευέλικτη λύσης, η οποία θα διευκολύνει όχι μόνον την συμβατική καθημερινότητα των χρηστών αλλά και θα προσφέρει ουσιαστική υποστήριξη σε άτομα με

περιορισμένες κινητικές δυνατότητες. Με αυτόν τον τρόπο ενισχύεται η αυτονομία, βελτιώνεται η ποιότητα ζωής και αναδεικνύονται νέες προοπτικές για μελλοντικές επεκτάσεις και βελτιώσεις.

1.4 Δομή της εργασίας

Κεφάλαιο 1 – Εισαγωγή

Στο πρώτο κεφάλαιο παρουσιάζεται το γενικό πλαίσιο του θέματος και αναλύονται οι στόχοι στους οποίου αποβλέπει η κατασκευή του έργου. Παράλληλα εξετάζεται ο ρόλος των έξυπνων συστημάτων στην καθημερινότητα αναδεικνύοντας τα πλεονεκτήματα και τις προκλήσεις που θέτουν.

Κεφάλαιο 2 – Βιβλιογραφική Έρευνα

Σε αυτό το κεφάλαιο παρουσιάζεται η βιβλιογραφική ανασκόπηση που πραγματοποιήθηκε σχετικά με τις τεχνολογίες φωνητικής ανάλυσης και παραγωγής. Παράλληλα γίνεται αναφορά σε αντίστοιχες εφαρμογές και υλοποιήσεις οι οποίες αποτέλεσαν πηγή έμπνευσης και καθοδήγησης για την ανάπτυξη του παρόντος έργου.

Κεφάλαιο 3 – Υλικό και λογισμικό

Το κεφάλαιο αυτό περιλαμβάνει την καταγραφή των υλικών στοιχείων και των εργαλείων λογισμικού που αποτέλεσαν βάση για την σχεδίαση ,την ανάπτυξη και τον έλεγχο της λειτουργίας του συστήματος.

Κεφάλαιο 4 – Ανάλυση σχεδίασης και λειτουργιών

Σε αυτό το κεφάλαιο παρουσιάζεται αναλυτικά η λειτουργία του συστήματος, η διαδικασία ανάπτυξης του λογισμικού MVF καθώς και ο σχεδιασμός του τυποποιημένου κυκλώματος Maestro VF. Τέλος, πραγματοποιείται μία ανασκόπηση σχετικά με την αποδοτικότητα και την δαπάνη της κατασκευής

Κεφάλαιο 5 – Συμπεράσματα και μελλοντικές επεκτάσεις

Στο τελευταίο κεφάλαιο αυτό παρουσιάζονται οι βασικές παρατηρήσεις και τα προβλήματα που καταγράφηκαν κατά την ανάπτυξη και την λειτουργία του συστήματος, ενώ παράλληλα προτείνονται βελτιώσεις και μελλοντικές επεκτάσεις. Τέλος γίνεται συνοπτική αποτίμηση του έργου.

Βιβλιογραφία

Παράρτημα Α

Στο παράρτημα αυτό παρατίθεται ο πηγαίος κώδικας που χρησιμοποιήθηκε για την ανάπτυξη του πρότυπου λογισμικού MVF.

Κεφάλαιο 2ο: Βιβλιογραφική έρευνα

Προτού ξεκινήσει η διαδικασία αρχιτεκτονικής σχεδίασης και υλοποίησης του συστήματος, πραγματοποιήθηκε εκτενής μελέτη για την εύρεση συστημάτων που εξυπηρετούν παρόμοιους λειτουργικών σκοπούς.

Σε αυτό το κεφάλαιο παρουσιάζεται συνοπτικά η ιστορική εξέλιξη των συστημάτων φωνητικής αναγνώρισης και των οικιακών βοηθών με στόχο την κατανόηση της δομής του έργου που υλοποιήθηκε. Επιπλέον αναλύονται βασικές τεχνολογίες και αρχιτεκτονικές προσεγγίσεις που συγκροτούν τα εν λόγω συστήματα. Τέλος παρατίθενται ενδεικτικά μοντέλα που αποτέλεσαν πηγή έμπνευσης στην τελική ανάπτυξη.

2.1 Ιστορική αναδρομή

Η ανάγκη για την δημιουργία συστημάτων φωνητικής αναγνώρισης και έξυπνων βοηθών ανάγεται αρκετές δεκαετίες πίσω. Η επιθυμία του ανθρώπου να επικοινωνεί φωνητικά με τις μηχανές παρατηρείται ήδη από τις αρχές του 20^{ου} αιώνα. Ωστόσο η πρακτική υλοποίηση της ιδέας καθυστέρησε σημαντικά καθώς τα πρώτα επιτυχημένα αποτελέσματα καταγράφηκαν μόλις την δεκαετία 1950-1960 [1].

Το πρώτο σύστημα αναγνώρισης φωνητικών εντολών παρουσιάστηκε από την Bell Labs στις αρχές του 1950 και ονομάστηκε “Audrey”. Η βασική του λειτουργία περιοριζόταν στην αναγνώριση μονοψήφιων αριθμών από έναν μόνο ομιλητή .Παρά την απλότητα του αποτέλεσε σημείο κατατεθέν για την καλλιέργεια της αλληλεπίδρασης μεταξύ ανθρώπου – μηχανής. Λίγα χρόνια αργότερα το 1961 η IBM παρουσίασε ένα πιο εξελιγμένο σύστημα, το “Shoebox”, το οποίο μπορούσε να αναγνωρίσει όχι μόνο αριθμούς αλλά και μία λίστα δεκαέξι λέξεων [2].

Στα τέλη της δεκαετίας του 1970 η ανάπτυξη συστημάτων όπως το Harpy προσέφερε σημαντική βελτίωση στην ακρίβεια και την ευελιξία των μοντέλων φωνητικής αναγνώρισης, ξεπερνώντας τα προηγούμενα όρια φτάνοντας ακόμα και την αναγνώριση άνω των χιλίων λέξεων. Η πρόοδος της τεχνητής νοημοσύνης σε συνδυασμό με τις επιστήμες στατιστικής ανάλυσης συνέβαλαν στην σταδιακή βελτιστοποίηση της λειτουργικότητας αυτών των συστημάτων επιτρέποντας την είσοδο τους στην αγορά, αρχικά με περιορισμένες εμπορικές εφαρμογές [1].

Συγκεκριμένα η εταιρεία Dragon Systems εισήγαγε στην αγορά το DragonNaturallySpeaking,ένα προϊόν που χρησιμοποιήθηκε ευρέως σε προσωπικούς υπολογιστές για την υπαγόρευση φωνητικών εντολών με δυνατότητα συνεχούς ομιλίας. Στην σύγχρονη εποχή, η εξέλιξη των συστημάτων φωνητικής αναγνώρισης υπήρξε ραγδαία, με την αγορά να κατακλύζεται από προϊόντα που ενσωματώνουν προηγμένες φωνητικές υπηρεσίες. Σημείο καμπής του 21^{ου} αιώνα αποτέλεσε η κυκλοφορία της ψηφιακές βοηθού Siri της Apple, η οποία ενσωματώθηκε στα κινητά τηλέφωνα της

εταιρείας. Στην συνέχεια η Amazon παρουσίασε την συσκευή Echo, ενσωματώνοντας την ευρέως διαδεδομένη σήμερα ψηφιακή βοηθό Alexa . Οι τεχνολογικές αυτές κατευθύνσεις στις οποίες επέλεξαν να επενδύσουν οι κορυφαίες εταιρείες του κλάδου άνοιξαν τον δρόμο για την σχεδίαση και ανάπτυξη ολοκληρωμένων συστημάτων οικιακών βοηθών [2][3].

Τα σύγχρονα Home assistant συστήματα συνδυάζουν μεγάλο εύρος τεχνολογιών, όπως οι υπηρεσίες Cloud, η τεχνητή νοημοσύνη, οι ψηφιακές διεπαφές, οι αλγόριθμοι μηχανικής μάθησης και η ηλεκτροτεχνία. Οι υπολογιστικές υποδομές που τα υποστηρίζουν επιτρέπουν την παροχή υπηρεσιών υψηλής ποιότητας όπως η δημιουργία αυτοματισμών σε οικιακά συστήματα, η διεξαγωγή σύνθετων διαλόγων και η διαχείριση μεγάλου αριθμού συσκευών IoT. Ωστόσο η χρήση τους εγκυμονεί αρκετούς κινδύνους, όπως η πιθανότητα παραβίασης προσωπικών δεδομένων κατά την ανταλλαγή πληροφοριών με απομακρυσμένους διακομιστές, ενώ η λειτουργικότητα τους εξαρτάται άμεσα από την διαθεσιμότητα σύνδεσης στο διαδίκτυο [3].

Στην πορεία εξέλιξης των εμπορικών εφαρμογών αναπτύχθηκαν παράλληλα φωνητικά συστήματα ανοιχτού κώδικα τα οποία βασίζονται στην χρήση μικροϋπολογιστών όπως το Raspberry Pi σε μορφή DIY(Do-It-Yourself). Η αρχιτεκτονική αυτών των συστημάτων περιλαμβάνει διάφορα offline μοντέλα ψηφιακών βοηθών τα οποία υστερούν σε απόδοση σε σχέση με τα αντίστοιχα εμπορικά, υπερτερούν όμως σε ανεξαρτησία και ευελιξία ως προς τον έλεγχο του λογισμικού και του υλικού. Το χαρακτηριστικό αυτό τα καθιστά ιδιαίτερα κατάλληλα για εφαρμογές πειραματικού και εκπαιδευτικού χαρακτήρα [1].

Η τεχνολογία φωνητικής αναγνώρισης μπορεί να συνεισφέρει ουσιαστικά στην καθημερινή ζωή και την επιστήμη, σε τομείς όπως η εκπαίδευση ,η αυτόνομη πλοήγηση και η υποστήριξη ατόμων με κινητικές δυσκολίες. Ταυτόχρονα η συνεχής πρόοδος της τεχνητής νοημοσύνης βελτιώνει σημαντικά τις διαδικασίες επεξεργασίας φωνητικών εντολών, ιδιαίτερα σε περιβάλλοντα με αυξημένο θόρυβο μειώνοντας την ενεργειακή κατανάλωση και βελτιώνοντας την ακρίβεια αναγνώρισης. Επιπλέον τα σύγχρονα μοντέλα μπορούν να υποστηρίζουν πολλαπλές γλώσσες και διαλέκτους, ενώ η δυνατότητα εξατομίκευσης τους ενισχύει την αλληλεπίδραση ανθρώπου-μηχανής προσδίδοντας μεγαλύτερη φυσικότητα [1][3].

2.2 Τεχνολογίες Home Assistant

Η δημιουργία ενός συστήματος Home Assistant έγκειται κυρίως στην ενσωμάτωση λογισμικών φωνητικής αναγνώρισης. Σε αυτήν την ενότητα παρουσιάζονται οι βασικές τεχνολογίες που χρησιμοποιούνται για την υλοποίηση τέτοιων συστημάτων καθώς και το λογισμικό Voice recognition που επιτρέπει την αλληλεπίδραση χρήστη-συσκευής.

2.2.1 Automatic Speech Recognition (ASR)

Η φωνή είναι το θεμελιώδες και βασικότερο μέσο επικοινωνίας στην ανθρώπινη αλληλεπίδραση. Οι σύγχρονες τεχνολογίες επιτρέπουν στις μηχανές να ανταποκρίνονται κατάλληλα στον άνθρωπο εκτελώντας σημαντικές λειτουργίες. Η αυτόματη αναγνώριση ομιλίας ASR αποτελεί τον βασικότερο μηχανισμό των συστημάτων φωνητικού ελέγχου και των οικιακών βοηθών.

Ο μηχανισμός ASR είναι αναπόσπαστο στοιχείο της δομής των δικτύων κατανόησης φυσικής γλώσσας NLP, αναλαμβάνοντας τον ρόλο του μεταφραστή στην επικοινωνία ανθρώπου – μηχανής μετατρέποντας τα ακουστικά σήματα σε γραπτό κείμενο. Η αρχιτεκτονική του διαμορφώνεται από διαδοχικά στάδια τα οποία είναι :

- Προεπεξεργασία ήχου: Το φωνητικό σήμα διέρχεται από φίλτρα FIR και από μηχανισμούς μείωσης θορύβου ώστε να βελτιωθεί η ποιότητα για περαιτέρω επεξεργασία.
- Εξαγωγή χαρακτηριστικών: Το σήμα αναλύεται ώστε να εντοπιστούν τα βασικά του γνωρίσματα.
- Μοντελοποίηση : Η πληροφορία προσαρμόζεται σε κατάλληλα μαθηματικά μοντέλα
- Αποκωδικοποίησης: Πραγματοποιείται η τελική αναγνώριση του λόγου και η απόδοση του σε κείμενο**.**

Η ακρίβεια του συστήματος δεν εξαρτάται αποκλειστικά από τον εσωτερικό μηχανισμό, άλλα και από την ποιότητα του σήματος που καταγράφεται. Για την εξαγωγή των χαρακτηριστικών του σήματος χρησιμοποιείται η τεχνική MFCC, η οποία προσομοιάζει τον μηχανισμό λειτουργίας του ανθρώπινου αυτιού εστιάζοντας στις συχνότητες που είναι απαραίτητες για την κατανόησης της ομιλίας. Η τεχνική περιλαμβάνει το διαχωρισμό του σήματος σε χρονικά πλαίσια, η ανάλυση του φάσματος των πλαισίων στην συνέχεια ολοκληρώνεται με την μέθοδο FFT, ενώ τέλος το σήμα οδηγείται από την έξοδο των φίλτρων σε διακριτό μετασχηματισμό συνημίτονου [4].

Ακολουθεί η μοντελοποίηση των χαρακτηριστικών η οποία ολοκληρώνεται από τα μοντέλα Hidden Markov Models. Τα HMM βασίζονται σε στατιστικά μοντέλα όπου ανιχνεύουν την πιθανότητα εμφάνισης ορισμένων λέξεων σε μία πρόταση ακόμα και σε σενάρια διαρκούς παύσης. Εναλλακτικά εφαρμόζονται τεχνικές απλοποίησης και συμπίεσης δεδομένων όπως το Vector Quantization ώστε να δημιουργηθούν κατάλληλα μοτίβα φωνής για σύγκριση [5].

Το τελικό στάδιο είναι η αποκωδικοποίηση μέσω του αλγορίθμου Viterbi, βασική λειτουργία του οποίου είναι να υπολογίσει την πιθανότερη ακολουθία λέξεων που ταιριάζει στην ληφθήσα καταγραφή. Τα σύγχρονα ASR αξιοποιούν τεράστιες βάσεις δεδομένων φωνής και τεχνικές μηχανικής μάθησης ώστε να προσαρμόζονται σε πραγματικά σενάρια και να βελτιώνουν διαρκώς την ακρίβεια αναγνώρισης [5].

Η υλοποίηση του συστήματος βασίζεται σε τεχνολογία cloud-based ASR μέσω της διεπαφής του Deepgram, η οποία προσφέρει επεξεργασία σε πραγματικό χρόνο και μειώνει τις υπολογιστικές απαιτήσεις του Raspberry Pi [6].

2.2.2 Natural Language Processing (NLP)

Η επεξεργασίας φυσικής Γλώσσας NLP είναι μία προοδευτικά εξελισσόμενη τεχνική τεχνητής νοημοσύνης, άμεση επιδίωξη της είναι η ερμηνεία των δεδομένων φυσικής γλώσσας σε προφορικό και γραπτό λόγο ώστε να είναι κατανοητά και διαθέσιμα προς επεξεργασία από τα υπολογιστικά συστήματα. Απώτερος σκοπός είναι να γεφυρώσει το χάσμα που αφορά τις επιστήμες της υπολογιστικής γλωσσολογίας και της μηχανικής μάθησης [7].

Για την εκτέλεση των μεθόδων του, το NLP ενσωματώνει μεθοδολογίες από τα μαθηματικά ,την γλωσσολογία και την επιστήμη της πληροφορικής. Η. αρχιτεκτονική του διακρίνεται σε δύο θεμελιώδεις άξονες οι οποίοι είναι :

- **Κατανόηση Φυσικής γλώσσας NLU (Natural Language Understanding)**: Εστιάζει στην ανάλυση και εξαγωγή γλωσσικών δεδομένων για την αναγνώριση οντοτήτων και εννοιών.
- **Παραγωγή Φυσικής Γλώσσας NLG (Natural Language Generation):** Αναφέρεται στην δημιουργία περιεχομένου που μπορεί να προέρχεται από αρχεία ήχου, κειμένων και εικόνας με την μορφή φυσικών απαντήσεων**.**

Η δομή που απεικονίζει το NLP απαρτίζεται από ένα σύνολο διεργασιών και τεχνολογιών τόσο για την κατανόηση και επεξεργασία, όσο και για την παραγωγή περιεχομένου:

1. Προεπεξεργασία κειμένου και υποδιαίρεση μονάδας: Σε αυτό το στάδιο εφαρμόζονται τεχνικές κανονικοποίησης όπως η αφαίρεση σημείων στίξης, περιττών συνδέσμων και συμβόλων. Επιπλέον το κείμενο υποδιαιρείται σε προτάσεις και λέξεις έτσι ώστε να πραγματοποιηθεί εξολοκλήρου μετατροπή σε πεζούς χαρακτήρες για τον επαναορισμό κάθε λέξης στην αρχική της ρίζα.

2. Μορφοσυντακτική ανάλυση: Σε αυτό το επίπεδο οι λέξεις κατατάσσονται γραμματικά με βάση το μέρος του λόγου στο οποίο ανήκουν. Η διεργασία επιτυγχάνεται από μηχανισμούς γλωσσικής επίγνωσης όπου για να καταργήσουν την αμφισημία χρησιμοποιούν μοντέλα πρόβλεψης από τις συμφραζόμενες λέξεις που εμπεριέχονται στο κείμενο.

3. Αναγνώριση οντοτήτων: Η διαδικασία αυτή αναλαμβάνει την ταξινόμηση των λέξεων και την αντιστοίχιση τους σε υφιστάμενες οντότητες όπως πρόσωπα, τοποθεσίες και ημερομηνίες.

4. Ανάλυση συναισθήματος: Πρόκειται για μία διαδικασία που βασίζεται σε μοντέλα τεχνητής νοημοσύνης εκπαιδευμένα εκ των προτέρων (pre-trained models). Τα μοντέλα αυτά μετασχηματίζουν σύνολα λέξεων σε αριθμητικά δεδομένα αναπαράστασης όπως τα βάρη συνάρτησης για την εκτίμηση του συναισθήματος σε μία κλίμακα.

5. Γλωσσική παραγωγή: Σε αυτό το σημείο το NLP χρησιμοποιεί κανόνες μορφοποίησης και μοντέλα πρόβλεψης για την μετατροπή των επεξεργασμένων πληροφορίων σε κατανοητό κείμενο [7][8].

Η τεχνική NLP είναι ευρέως ενσωματωμένη σε πολλές εφαρμογές της καθημερινότητας μας, Χαρακτηριστικά παραδείγματα είναι τα συστήματα ψηφιακών βοηθών όπως τα chatbots και η αυτόματη μετάφραση, ενώ συχνά συναντώνται και σε εφαρμογές εκπαίδευσης και κοινωνικών δικτύων για την ανίχνευση ακατάλληλου περιεχομένου. Επιπλέον το NLP τείνει να χρησιμοποιείται σε βάσεις δεδομένων επαγγελματικού η επιστημονικού χαρακτήρα για την εξαγωγή πληροφορίων [7].

Παρά την υψηλή αποτελεσματικότητα η απόδοση των NLP συστημάτων παρουσιάζει διακυμάνσεις που οφείλονται κυρίως στην ποιότητα εκπαίδευσης των μοντέλων πρόβλεψης. Ωστόσο παραμένει ένα από τα πιο ισχυρά και συνεχώς εξελισσόμενα εργαλεία της τεχνητής νοημοσύνης με άμεση επίδραση στην καθημερινότητα τόσο ως αυτόνομη τεχνολογία όπως οι ψηφιακοί βοηθοί όσο και ως ενσωματωμένη λειτουργία σε σύνθετα συστήματα όπως τα Home Assistant [8].

2.2.3 Internet of Things (IoT) Protocols

Η λειτουργικότητα και απόδοση των σύγχρονων συστημάτων IoT έγκειται στην αξιοπιστία της επικοινωνίας μεταξύ λογισμικού και υλικού. Προκειμένου να διασφαλιστεί η επικοινωνία των ελεγκτών με τα αισθητήρια όργανα και τους ενεργοποιητές ορίστηκαν τα πρωτόκολλα IoT. Η ύπαρξη αυτών των πρωτοκόλλων αποτελεί το «νευρικό σύστημα» των εφαρμογών, εξασφαλίζοντας ασφαλή και ταχεία μεταφορά δεδομένων σε αρχιτεκτονικές ασύρματων, ημιασύρματων και ενσύρματων συνδέσεων [9].

Ένα από τα πιο διαδεδομένα πρωτόκολλα εφαρμογής στα συστήματα Home Assistant είναι το MQTT(Message Queuing Telemetry Transport). Η δομή του οποίου βασίζεται στο πρωτόκολλο δικτύου TCP/IP,κατάλληλο για συστήματα χαμηλής ισχύος όπου δεν απαιτούνται μεγάλοι υπολογιστικοί πόροι ,όπως το Raspberry Pi. Εναλλακτικά το πρωτόκολλο CoAP προσφέρει εξαιρετική ευελιξία και ταχύτητα, καθώς η αρχιτεκτονική του βασίζεται στο πρωτόκολλο UDP, ωστόσο υστερεί έναντι του MQTT σε ζητήματα ασφαλείας και σταθερότητας σύνδεσης.

Η αρχιτεκτονικοί των IoT περιλαμβάνει επίσης πρωτόκολλα φυσικού επιπέδου (Physical layer) και σύνδεσης (Link layer), όπως το Zig Bee και το Wi-Fi. Αν και λειτουργούν σε διαφορετικά επίπεδα του μοντέλου OSI μπορούν να συνυπάρχουν στο ίδιο σύστημα. Η ανάγκη μείωσης της κατανάλωσης και της δέσμευσης πόρων οδήγησε στην ανάπτυξη πρωτοκόλλων όπως το Zig Bee και το Z-Wave .Τα πρωτόκολλα αυτά επιτρέπουν την χωρίς εξάρτηση από κεντρικούς ελεγκτές, προσφέροντας μεγάλη εμβέλεια μετάδοσης και λειτουργία σε συχνότητες που δεν παρεμβάλλονται σε εκείνες του Wi-Fi.

Συχνά τίθενται ζητήματα επικοινωνίας σε υψηλούς ρυθμούς μετάδοσης δεδομένων και εύρους ζώνης, ιδιαίτερα όπου υπάρχουν υποδομές δικτύωσης WLAN, προτιμάται η χρήση Wi-Fi. Ωστόσο, τα

συστήματα που βασίζονται σε αυτήν την τεχνολογία εμφανίζουν αυξημένες απαιτήσεις ισχύος και χαμηλότερη αποδοτικότητα σε πολύπλοκα δίκτυα όπου απαιτείται σταθερότητα.

Η υιοθέτηση του MQTT ως βασικού πρωτοκόλλου επικοινωνίας προσφέρει υψηλή συμβατότητα και ευκολία ενσωμάτωσης σε υποδομές cloud όπως τα APIs.Επιπλέον το MQTT μπορεί να θεωρηθεί βέλτιστή επιλογή για την υλοποίηση μηχανισμών διαχείρισης συσκευών και συστημάτων αυτοματισμού χαμηλής ισχύος όπως τα Home Assistant και τα Voice Recognitioning , υποστηρίζοντας αξιόπιστα την μετάδοση φωνητικών μηνυμάτων τόσο σε λήψη όσο και σε αποστολή [9][6].

2.2.4 Platforms

Οι μικροελεγκτές είναι ολοκληρωμένα κυκλώματα που διαθέτουν υψηλή υπολογιστική ισχύ και μνήμη ,ενώ χρησιμοποιούνται κυρίως σε εφαρμογές όπου απαιτείται έλεγχος αισθητήρων και ενεργοποιητών σε πραγματικό χρόνο. Η επιλογή της κατάλληλης πλατφόρμας εξαρτάται από τις υπολογιστικές απαιτήσεις του συστήματος, τις διαθέσιμες διεπαφές I/O και την κατανάλωσης ενέργειας.

Στην σύγχρονη εποχή, οι πιο διαδεδομένοι ελεγκτές της αγοράς είναι το Raspberry Pi ,ο ESP32 και το Arduino.Τα συγκεκριμένα προϊόντα χρησιμοποιούνται ευρέως σε ακαδημαϊκές, πειραματικές και εμπορικές εφαρμογές καθώς προσφέρουν φιλικό περιβάλλον ανάπτυξης και εύκολη σύζευξη με αισθητήρια όργανα μέσω διαθέσιμων βιβλιοθηκών και την συγγραφή κατάλληλου κώδικα [10].

Η πλατφόρμα Arduino αξιοποιείται κυρίως σε εκπαιδευτικές δραστηριότητες διότι η υπολογιστική της ισχύ υπολείπεται σημαντικά σε σχέση με εκείνες του ESP32 και του Raspberry Pi. Αντίθετα οι δύο τελευταίες μπορούν να ανταποκριθούν με υψηλή αξιοπιστία σε εφαρμογές IoT, συνδυάζοντας αυξημένες δυνατότητες επεξεργασίας με χαμηλή κατανάλωση ενέργειας.

Επιπλέον πολλοί σύγχρονοι μικροελεγκτές ενσωματώνουν υποστήριξη για πρωτόκολλα Wi-Fi και Bluetooth ,επιτρέποντας την άμεση διασύνδεση τους με υπηρεσίες νέφους (cloud) .Η δυνατότητα αυτή καθιστά εφικτή την ανάπτυξη συστημάτων αναγνώρισης και παραγωγής φυσικής γλώσσας, όπως οι φωνητικοί βοηθοί και οι πλατφόρμες έξυπνου αυτοματισμού [10].

2.2.5 Cloud και Edge Computing

Στο πλαίσιο ανάπτυξης και βελτιστοποίησης των εφαρμογών και των διεπαφών που ενσωματώνονται σε έξυπνα συστήματα ,ορίστηκε ως επιτακτική η ανάγκη αναζήτησης τεχνολογιών φυσικής γλώσσας που μπορούν να επεξεργάζονται δεδομένα σε πραγματικό χρόνο. Οι υπολογιστικές αυτές τεχνολογίες του Cloud και Edge Computing θεωρούνται ανταγωνιστικές στα συστήματα φωνητικής αλληλεπίδρασης για την λήψη αποφάσεων με χαμηλή καθυστέρηση [7][11].

Η αρχιτεκτονική Cloud Computing βασίζεται στην αποθήκευση και επεξεργασία των δεδομένων σε συστήματα απομακρυσμένων διακομιστών με υψηλή υπολογιστική ισχύ. Πρόκειται για μία ευρέως

διαδεδομένη υποδομή που χρησιμοποιείται σε εφαρμογές φωνητικής ανάλυσης όπως η αυτόματη μετάφραση και ή σύνθεση ομιλίας, χωρίς να επιβαρύνει τους τοπικούς πόρους του υλικού.

Αντίθετα η αρχιτεκτονική Edge Computing δίνει προτεραιότητα στην επεξεργασία των τοπικών δεδομένων κοντά στην πηγή παραγωγής τους. Οι ακατέργαστες πληροφορίες που συλλέγονται από τους IoT αισθητήρες οδηγούνται προς τα συστήματα επεξεργασίας, εκεί πραγματοποιούνται οι διεργασίες φιλτραρίσματος και ανάλυσης, ενώ η τελική απόφαση που αφορά την εκτέλεση των εντολών λαμβάνεται τοπικά ουτοσώστε να διατηρηθεί χαμηλή η απόκριση και η κατανάλωση ενέργειας. Σε περιπτώσεις όπου απαιτείται αυξημένη υπολογιστική ισχύ το Edge συνεργάζεται με υπηρεσίες Cloud δημιουργώντας ένα συνεργατικό μοντέλο.

Η επιλογή αρχιτεκτονικής διαφοροποιείται ανάλογα με την απαίτηση του εκάστοτε συστήματος, Βασικά κριτήριά για την διαμόρφωση της αποτελούν ο όγκος δεδομένων που τίθεται προς επεξεργασία και η ταχύτητα απόκρισης. Το Edge Computing είναι ιδανικό για εφαρμογές με υψηλές απαιτήσεις σε ταχύτητα και ασφάλεια, καθώς τα δεδομένα δεν αποστέλλονται σε εξωτερικούς διακομιστές μέσω διαδικτύου. Το Cloud Computing αντίθετα συνεργάζεται καλύτερα με συστήματα περιορισμένων πόρων, ενώ είναι καταλληλότερο για περιπτώσεις επεξεργασίας αυξημένου όγκου πληροφορίας, παρέχοντας δυνατότητες επέκτασης και συντήρησης αν και με υψηλότερη καθυστέρηση [11].

Σε δομές όπου απαιτείται φωνητικός έλεγχος ή αυτοματισμοί συσκευών όπως τα Home assistant προτείνεται η σύσταση υβριδικών μοντέλων Cloud -Edge. Σε αυτά, οι βασικές διεργασίες όπως ο έλεγχος υλικού και η προεπεξεργασία σημάτων ολοκληρώνονται τοπικά με αρχιτεκτονική Edge, ενώ οι διεργασίες υψηλού φόρτου ή πρόσβασης σε βάσεις δεδομένων πραγματοποιούνται απομακρυσμένα σε διακομιστές Cloud. Έτσι επιτυγχάνεται ισορροπία μεταξύ αυτονομίας ισχύος, χαμηλής καθυστέρησης και σταθερής ακρίβειας [7][11].

2.2.6 Voice User Interface (VUI)

Η τεχνολογία VUI αποτελεί ένα από σημαντικότερα δομικά στοιχεία της αλληλεπίδρασης ανθρώπου-μηχανής λειτουργώντας ως δίαυλος επικοινωνίας για την ανταλλαγή δεδομένων μέσω φωνητικών εντολών. Παραδοσιακά ο άνθρωπος για να αποκτήσει πρόσβαση σε υπολογιστικά συστήματα και υπηρεσίες τείνει να χρησιμοποιεί οθόνες και χειριστήρια, η πρακτική αυτή είναι λειτουργικά ορθή, εντούτοις δεσμεύει σημαντικούς υλικούς πόρους για την εκτέλεση μίας απλής ενέργειας. Την λύση σε αυτό το προβλήματος έρχεται να δώσει η τεχνολογία VUI η οποία μετατρέπει την φωνή σε βασική πύλη ελέγχου των συστημάτων διαχείρισης, προσφέροντας αφενός μεγαλύτερη φυσικότητα στην επικοινωνία αφετέρου συμβάλλει στην εξοικονόμηση πόρων [12].

Η αρχιτεκτονική των φωνητικών διεπαφών VUI βασίζεται στην συνεργασία πολλαπλών τεχνολογιών, εκ των οποίων οι τρείς βασικοί άξονες είναι οι εξής:

- **Αυτόματη Αναγνώριση Ομιλίας :** Μετατρέπει τις φωνητικές εντολές του χρήστη σε κείμενο.

- **Επεξεργασία φυσικής γλώσσας** :Με την βοήθεια μοντέλων τεχνητής νοημοσύνης και βαθιάς εκμάθησης αναλύει και επεξεργάζεται το παραγόμενο κείμενο ώστε να εξαχθεί το περιεχόμενο της εντολής**.**
- **Σύνθεσης Ομιλίας** :Μετατρέπει την απόκριση του συστήματος εκ νέου σε φωνητική έξοδο.

Η VUI έχει ισχυρή παρουσία στην αγορά προϊόντων έξυπνης διαχείρισης, με χαρακτηριστικά παραδείγματα τα συστήματα της Amazon Alexa , Google και Home Assistant. Η χρήση της αποτελεί σημείο αναφοράς στην βελτιστοποίηση της καθημερινής διαβίωσης καθώς δεν περιορίζεται μόνο στην άνεση και την πολυτέλεια αλλά επεκτείνεται και στην υποστήριξη ατόμων με μειωμένη κινητικότητα προσφέροντας μεγαλύτερη αυτονομία [1][3][12].

Βασικός στόχος της ενσωμάτωσης VUI στα συστήματα οικιακού αυτοματισμού είναι η δημιουργία ενός εξατομικευμένου περιβάλλοντος στο οποίο ο χρήστης μπορεί να επικοινωνεί με την συσκευή με αμεσότητα και φυσικότητα αντίστοιχη μίας πραγματικής συνομιλίας. Η ενσωμάτωση αυτής της τεχνολογίας αποδεικνύεται ιδιαίτερα αποδοτική σε συστήματα χαμηλής ισχύος, ειδικότερα όταν υποστηρίζεται από επιμέρους διεπαφές API για αναγνώρισης STT και TTS [12].

2.3 Έμπνευση και συσχετιζόμενες εργασίες

Σε αυτήν την ενότητα παρουσιάζονται τα βασικά έργα που συνέβαλαν στην διαμόρφωση και οργάνωση της παρούσας Διπλωματικής Εργασίας. Παράλληλα περιλαμβάνονται εφαρμογές που λειτουργήσαν τα οποία συνιστούν πηγή έμπνευσης για μελλοντικές βελτίωσης και επεκτάσεις του συστήματος.

2.3.1 Σύστημα οικιακού αυτοματισμού χαμηλού κόστους.

Η πρώτη σημαντική πηγή εντοπίζεται στην εργασία των G. Prasanna και R. Narayanadass, οπού παρουσιάζεται ένα σύστημα ανεπτυγμένο σε πλατφόρμα μικροϋπολογιστή Raspberry Pi με δυνατότητα ελέγχου ηλεκτρικών συσκευών μέσω φωνητικών εντολών [13].

Σε επίπεδο υλικού ,το έργο αξιοποιεί ένα Raspberry Pi B ως ΚΜΕ για την διαχείριση τεσσάρων ρελέ οδήγησης κυκλωμάτων χαμηλής τάσεως. Η διασύνδεση με τις συσκευές πραγματοποιείται τοπικά ,χωρίς σύνδεση στο δίκτυο ενώ, η αλληλεπίδραση με τον χρήστη υποστηρίζεται με την εγκατάσταση εξωτερικού μικροφώνου.

Η λήψη και επεξεργασία των φωνητικών δειγμάτων, πραγματοποιείται αρχικά μέσω μικροφώνου και στην συνέχεια με την offline διεπαφή αναγνώρισης PocketSphinx. Η διεπαφή είναι ενσωματωμένη στον κύριο κώδικα του έργου, ο οποίος έχει αναπτυχθεί σε γλώσσα προγραμματισμού Python. Η διαδικασία ελέγχου είναι ιδιαίτερα απλή, τα τέσσερα ρελέ αντιστοιχίζονται σε τέσσερις συσκευές:Light, Fan, Tv, Reverse. Ενώ κάθε εντολή ενεργοποίησης και απενεργοποίησης συνοδεύεται από το όνομα της αντίστοιχής συσκευής (Πχ Fan off/Tv On) μεταφέροντας ένα σήμα 0-3.3V μέσω της

GPIO θύρας προς την βάση ενός τρανζίστορ που οδηγεί την επαφή του ρελέ. Επιπλέον παρέχεται δυνατότητα ταυτόχρονου ελέγχου όλων των εξόδων μέσω δύο γενικών εντολών (all on ,all off).

Η μελέτη του συστήματος καταδεικνύει οτι η γλώσσα Python μπορεί να υποστηρίξει πλήρως διεπαφές φωνητικών εντολών με εξαιρετικά απλή σύνταξη, ενώ η συνδεσμολογία του συστήματος παραμένει απλή και αξιόπιστη, καθιστώντας την κατάλληλη για εφαρμογές χαμηλής ισχύος. Παρά τα πλεονεκτήματα, η απουσία διαδικτυακής σύνδεσης περιορίζει την ακρίβεια, καθώς το μοντέλο βασίζεται σε μονολεκτικές εντολές και η τεχνολογία ASR που υποστηρίζει η PocketSphinx διαθέτει περιορισμένο λεξικό. Ωστόσο, η εργασία επιτυγχάνει λειτουργικότητα και αποδοτικότητα ,αποτελώντας έναν αξιόλογο κορμό που με τις κατάλληλες τροποποιήσεις μπορεί να συνθέσει ένα ισχυρό σύστημα οικιακού αυτοματισμού [13].

2.3.2 Σύστημα φωνητικής αναγνώρισης μέσω διαδικτυακών διεπαφών

Η εργασία αυτή παρουσιάζει την ανάπτυξης ενός συστήματος φωνητικής αναγνώρισης, του οποίου η αρχιτεκτονική βασίζεται στην αξιοποίηση διαδικτυακών διεπαφών για την επεξεργασία και παραγωγής δεδομένων μέσω τεχνολογιών STT και TTS. Ως κεντρική μονάδα επεξεργασίας χρησιμοποιείται ο μικροϋπολογιστής Raspberry Pi 3,ο οποίος είναι υπεύθυνος για τον έλεγχο τεσσάρων οικιακών συσκευών που οδηγούνται από ηλεκτρονόμους [14].

Στο τεχνικό επίπεδο το σύστημα περιλαμβάνει δύο συσκευές ήχου USB, μία για την λήψη (μικρόφωνο) και μία για την παραγωγή (ηχείο) ηχητικών σημάτων, καθώς και τέσσερα ρελέ που αντιστοιχούν σε τέσσερις διαφορετικές συσκευές . Η οδήγηση των ρελέ πραγματοποιείται με την χρήση τρανζίστορ, ενώ η επιλογή του Raspberry Pi 3 βασίστηκε στην ενσωματωμένη τεχνολογία Wi-Fi, η οποία περιορίζει την ανάγκη καλωδιώσεων και βελτιώνει την αισθητική και εργονομία της εγκατάστασης.

Η αυτόματη αναγνώριση ομιλίας ASR υλοποιείται αποκλειστικά μέσω υπηρεσιών Cloud Computing και συγκεκριμένα με την διεπαφής Google Speech Recognition. Η φωνητική ανατροφοδότηση προς τον χρήστη πραγματοποιείται επίσης μέσω online API της Google, παρέχοντας ρεαλισμό και φυσικότητα στον τόνο της φωνής του συστήματος . Η συνδεσμολογία του κυκλώματος είναι παρόμοια με εκείνη του συστήματος της υποενότητας 2.3.1, χωρίς ωστόσο να παρουσιάζει σημαντικές διαφοροποιήσεις ως προς την λογική ελέγχου.

Η αξιολόγηση του συστήματος κατέδειξε ότι χρήση διαδικτυακών διεπαφών για την επεξεργασία δεδομένων εισόδου παρέχει υψηλή υπολογιστική ακρίβεια και επεκτασιμότητα, ενώ απαλλάσσει το υλικό από την ανάγκη εκτέλεσης βαριών διεργασιών. Παράλληλα η ενσωμάτωση online APIs αποδείχθηκε πιο αποτελεσματική από την τεχνολογία PocketSphinx καθώς υποστηρίζει μεγαλύτερα λεξικά και πιο σύνθετες εντολές. Παρόλο που η εξάρτηση από το διαδίκτυο εισάγει μία μικρή καθυστέρηση στην απόκριση της τάξης μερικών ms, η καθυστέρηση αυτή δεν επηρεάζει την πρακτική

λειτουργικότητα .Συνολικά, το σύστημα συγκεντρώνει όλα τα απαραίτητα χαρακτηριστικά ώστε με κατάλληλες τροποποιήσεις να αναβαθμιστεί σε ένα ολοκληρωμένο Home Assistant [14].

2.3.3 Amazon Alexa

Το σύστημα οικιακού αυτοματισμού της Amazon Alexa συγκαταλέγεται στα πιο ολοκληρωμένα τεχνολογικά προϊόντα φωνητικού ελέγχου, συνδυάζοντας καινοτόμα τεχνολογικά στοιχεία όπως οι υπηρεσίες Cloud Computing, η τεχνητή νοημοσύνη και το Internet of Things . Η αρχιτεκτονική του επιτρέπει την αναγνώριση πολλαπλών φωνητικών προφίλ με υψηλή αξιοπιστία, παραμερίζοντας διαφορές ύφους ,προφοράς ή ρυθμού ομιλίας [12][15].

Σε τεχνικό επίπεδο, η συσκευή Amazon Echo είναι εξοπλισμένη με ισχυρό τετραπύρηνο επεξεργαστή τελευταίας γενιάς, βελτιστοποιημένο για εφαρμογές χαμηλής κατανάλωσης ενέργειας και ελάχιστης καθυστέρησης απόκρισης. Η λήψη φωνητικών εντολών υποστηρίζεται από συστοιχία έξι έως οκτώ μικροφώνων υψηλής ευαισθησίας, ενώ η έξοδος γίνεται μέσω ηχείων υψηλής ανάλυσης που προσαρμόζουν την ακουστική τους απόδοση ανάλογα με την δομή του χώρου. Επιπλέον, η συσκευή υποστηρίζει συνδεσιμότητα μέσω Wi-Fi, Bluetooth και Zigbee εξασφαλίζοντας μεγάλη διαλειτουργικότητα [15].

Για την ενίσχυση της λειτουργικότητας αλλά και την προστασία του τοπικού εξοπλισμού, η Alexa αξιοποιεί αρχιτεκτονική βασισμένη σε cloud, όπου όλες οι διεργασίες επεξεργασίας πραγματοποιούνται απομακρυσμένα. Η κύρια υπηρεσία που χρησιμοποιεί είναι το Amazon Web Services, το οποίο παρέχει υπολογιστική ισχύ και φιλοξενεί τις λειτουργίες ASR, NLU και TTS .Τα προεκπαιδευμένα μοντέλα τεχνητής νοημοσύνης που εφαρμόζονται στο AWS προσφέρουν υψηλή ακρίβεια, βελτιωμένη προσαρμοστικότητα και δυνατότητα συνεχούς αναβάθμισης, διασφαλίζοντας την εξέλιξη του συστήματος με την πάροδο του χρόνου.

Η ενεργοποίηση, η απενεργοποίηση και η παραμετροποίηση κάθε συνδεδεμένης συσκευής μπορεί να πραγματοποιούνται με σύνθετες φωνητικές εντολές, οι οποίες αναλύονται σε πραγματικό χρόνο μέσω απομακρυσμένων διακομιστών Cloud AI. Παράλληλα παρέχεται ενιαίο περιβάλλον διαχείρισης μέσω λογισμικού για iOS και Android, επιτρέποντας στον χρήστη να διαμορφώνει τις παραμέτρους ακόμα και χωρίς άμεση φωνητική αλληλεπίδραση. Η διπλή αυτή δυνατότητα χρήσης ενισχύει την ευελιξία και την αξιοπιστία του συστήματος ,καθιστώντας το κατάλληλο για διαφορετικά προφίλ χρηστών [12].

Η πλατφόρμα Amazon Alexa συγκαταλέγεται στα ισχυρότερα και πιο εξελιγμένα συστήματα οικιακού αυτοματισμού, καθώς συνδυάζει ένα πολυεπίπεδο μοντέλο υπηρεσιών που μπορεί να υποστηρίξει λειτουργίες ελέγχου με εξαιρετικά μικρή απόκριση και ακρίβεια. Η δυνατότητα διαχείρισης πολλαπλών φωνητικών προφίλ, η συνεχής εκπαίδευση των μοντέλων τεχνητής νοημοσύνης και η συμβατότητα την καθιστούν ιδιαίτερα ανταγωνιστική. Επιπλέον η Alexa μπορεί να διαχειριστεί δεδομένα χωρίς να περιορίζεται στην φωνητική αλληλεπίδραση αλλά μέσω εφαρμογών, ενισχύοντας την συνολική

εμπειρία του χρήστη προσφέροντας ένα πλήρως ολοκληρωμένο πλαίσιο οικιακού αυτοματισμού [12][15].

2.3.4 Αποτίμηση βιβλιογραφικής έρευνας

Η βιβλιογραφικής έρευνας ανέδειξε την ανάγκη υιοθέτησης μίας υβριδική προσέγγισης, η οποία συνδυάζει βασικά τεχνολογικά στοιχεία σε επίπεδο υλικού και λογισμικού ,με στόχο την δημιουργία ενός σταθερού και επεκτάσιμου συστήματος φωνητικής αναγνώρισης και διαχείρισης συσκευών. Σε αντιπαραβολή με αντίστοιχα έργα όπως το "Low Cost Home Automation Using Offline Speech Recognition" και το "Design and Implementation of Voice Recognition Based Smart Home Automation System " τα οποία βασίστηκαν επίσης σε πλατφόρμες Raspberry Pi για τον έλεγχο συσκευών μέσω θυρών GPIO, η παρούσα υλοποίηση παρουσιάζει πιο σύνθετη δομή, αξιοποιώντας επιπλέον δυνατότητες όπως η παραμετροποίηση φωτισμού μέσω τεχνικής PWM.

Η επιλογή του Raspberry Pi κρίθηκε καταλληλότερη για ανάπτυξη συστημάτων αυτού του τύπου, δεδομένου ότι μπορεί να ενσωματώσει πολλαπλές τεχνολογίες με υψηλή επεξεργαστική ισχύ για την ασφαλή και αποδοτική διαχείριση εξωτερικών φορτίων.

Σε επίπεδο λογισμικό η παρούσα εργασία δεν αποκλίνει σημαντικά από τις προηγούμενες εφαρμογές, καθώς η αρχιτεκτονική βασίζεται σε υπηρεσίες υπολογιστικής νέφους για την ανάλυση και επεξεργασία φυσικής γλώσσας. Οι διεπαφές STT και TTS παρέχουν υψηλή ακρίβεια, μειώνοντας παράλληλα τις απαιτήσεις σε τοπικούς υπολογιστικούς πόρους.

Συνολικά η βιβλιογραφική ανασκόπηση συνέβαλε στην διαμόρφωση της αρχιτεκτονικής και της λειτουργικής δομής του συστήματος. Αναμφίβολά προϊόντα όπως η Amazon Alexa αποτελούν αντικείμενα έμπνευσης για μελλοντικές επεκτάσεις του συστήματος, (Π.χ. ανάπτυξη εξατομικευμένων μοντέλων φωνητικής αναγνώρισης),ωστόσο η υλοποίηση λειτουργεί αποτελεσματικά ως γέφυρα μεταξύ ακαδημαϊκών και εμπορικών προτύπων, συνδυάζοντας τεχνολογίες αιχμής με περιορισμένες υλικές απαιτήσεις.

Κεφάλαιο 3ο: Υλικό και λογισμικό

Η παρούσα διπλωματική εργασία διακρίνεται σε δύο βασικές συνιστώσες, αυτές είναι το υλικό και το λογισμικό .Σε αυτό το κεφάλαιο περιγράφονται λεπτομερώς τα δομικά στοιχεία του συστήματος καθώς και τα εργαλεία το οποία χρησιμοποιήθηκαν κατά την διαδικασία σχεδίασης και ανάπτυξης του έργου.

3.1 Υλικά

Η υλοποίηση του έξυπνου συστήματος βασίστηκε στην συνεργασία δύο επιμέρους υποσυστημάτων. Το πρώτο είναι ο μικροϋπολογιστής Raspberry Pi 4 ο οποίος αναλαμβάνει τον ρόλο του «εγκεφάλου» αποφάσεων για την λήψη και την αλληλεπίδραση μεταξύ χρήστη και κυκλώματος, το δεύτερό είναι η πλακέτα Maestro VF που επιτελεί χρέη εκτελεστικού υποσυστήματος, υλοποιώντας με ακρίβεια τις εντολές που λαμβάνει από το Raspberry Pi.

Σε αυτήν την ενότητα περιγράφονται αναλυτικά τα δομικά υλικά που χρησιμοποιήθηκαν για την ανάπτυξη του έργου, με τα χαρακτηριστικά και τις αντίστοιχες λειτουργίες που επιτελούν.

3.1.1 Raspberry Pi

Η επεξεργασία των δεδομένων εισόδου και η μετάφραση τους σε μηχανικές εντολές, προϋποθέτει την ύπαρξη μίας μονάδας επεξεργασίας και αποφάσεων. Τον ρόλο αυτό επιτελεί ο μικροϋπολογιστής Raspberry Pi 4 (Σχήμα 3.1), ο οποίος έχει αναλάβει την ευθύνη του κεντρικού ελέγχου και της διαχείρισης όλων των λειτουργιών [16].

Το Raspberry Pi 4 είναι ένας single-board μικροϋπολογιστής μικρού μεγέθους και χαμηλού κόστους με δυνατότητες που σε μικρότερη κλίμακα, πλησιάζουν εκείνες ενός επιτραπέζιου υπολογιστή. Η σχεδίαση του το καθιστά ιδιαίτερα ευέλικτο επιτρέποντας την αξιοποίηση του σε εκπαιδευτικά περιβάλλοντα και σε βιομηχανικές εφαρμογές.

Αναλυτικότερα , η επιλεγμένη έκδοση του Raspberry Pi 4 διαθέτει:

- **Επεξεργαστή** με συχνότητα λειτουργίας 1.5GHz.
- **Μνήμη RAM** τύπου LPDDR4 σε εκδόσεις 2,4 και 8 GB
- **Λειτουργικό σύστημα** Raspberry Pi OS το οποίο εγκαθίσταται σε κάρτα μνήμης SD μέσω εξωτερικού υπολογιστή.
- **Συνδεσιμότητα** Wi-Fi 802.11ac και Bluetooth 5.0 για ασύρματη και θύρα Ethernet για ενσύρματη πρόσβαση στο διαδίκτυο.
- **Θύρες I/O** Δύο θύρες USB 2.0, δύο θύρες USB 3.0 και δύο micro HDMI για λειτουργίες προβολής.

Το πιο σημαντικό τεχνικό χαρακτηριστικό του είναι το 40-pin GPIO Header , το οποίο μπορεί να λειτουργήσει είτε ως είσοδος είτε ως έξοδος, παρέχοντας την δυνατότητα σύνδεσης με ηλεκτρονικά

στοιχεία όπως αισθητήρες, ρελέ και MOSFET. Ορισμένες ακίδες προσφέρουν σταθερή τάση τροφοδοσίας 3.3V και 5V, γραμμές γείωσης , καθώς και κανάλια εισόδου/εξόδου για ψηφιακά σήματα. Επιπλέον υποστηρίζονται τα πρωτοκόλλα σειριακής επικοινωνίας SPI, UART και I^2C, ενώ είναι δυνατή η παραγωγή σημάτων PWM για έλεγχο ηλεκτρονικών στοιχείων όπως κινητήρες ή MOSFET [16].

Η δικτυακή υποστήριξη του Raspberry Pi 4 ενισχύει τη λειτουργικότητα του συστήματος. Τα πρότυπα Wi-Fi 802.11b/g/n/ac εξασφαλίζουν σταθερή επικοινωνία με υπηρεσίες νέφους, ενώ το Bluetooth 5.0 επιτρέπει τη σύζευξη με μικρόφωνα, ηχεία και άλλες συσκευές, υποβοηθώντας την αλληλεπίδραση .Η σύνδεση συσκευών ήχου μέσω USB προσφέρει καθαρή εγγραφή και ποιοτική αναπαραγωγή, απαραίτητη για τις διεργασίες ASR και TTS [18].

Το Raspberry Pi OS αποτέλεσε βασικό στοιχείο της ανάπτυξης, παρέχοντας βιβλιοθήκες και εργαλεία για πρόσβαση στα GPIO και για την ενσωμάτωση των APIs. Η σταθερότητα και η υποστήριξη συνέβαλαν στην ταχεία ανάπτυξη, ενώ τα ενσωματωμένα πακέτα και οι βιβλιοθήκες υποστήριξαν την ενσωμάτωση των υπηρεσιών φωνητικής επεξεργασίας [17].

Η επιλογή του Raspberry Pi 4 δικαιολογείται από την πρακτική λειτουργικότητα της αρχιτεκτονικής του και την ικανότητα υποστήριξης εφαρμογών σε πραγματικό χρόνο. Η υπολογιστική ισχύς επαρκεί για παράλληλες διεργασίες, όπως ο έλεγχος GPIO, η διαχείριση χρονοδιακοπτών και η φωνητική επεξεργασία, ενώ η πρόσβαση σε Cloud APIs εξασφαλίζει ακρίβεια και αξιοπιστία. Συνολικά, η πλατφόρμα συνθέτει μια σταθερή και ευέλικτη βάση για σύγχρονα συστήματα οικιακού αυτοματισμού με φωνητικό έλεγχο [16][18].

Σχήμα 3.1 Raspberry Pi 4 Model B

[raspberrypi.com/products/raspberry-pi-4-model-b/]

3.1.2 Περιφερειακά

Παρακάτω παρατίθενται τα περιφερειακά μηχανικά εξαρτήματα που εγκαταστάθηκαν στα τερματικά σύνδεσης του Raspberry Pi και του Maestro VF.

3.1.2.1 Ψήκτρα Raspberry Pi

Όπως κάθε είδους υπολογιστικό σύστημα, έτσι και τα κυκλώματα των Raspberry Pi ταλανίζονται από φαινόμενα υπερθέρμανσης τα οποία οφείλονται στην αυξημένη δραστηριότητα του επεξεργαστή. Η υπερβολική αύξηση της θερμοκρασίας μπορεί να οδηγήσει σε μείωση της απόδοσης ή ακόμα και σε καταστροφή του υλικού. Για τον λόγο αυτό, η εγκατάσταση συστημάτων ψύξης αποτελεί σύνηθες πρακτική προκειμένου να διατηρηθεί σταθερή η απόδοση της πλατφόρμας ανεξάρτητα από την αύξηση της θερμότητας [20].

Η παθητική ψήκτρα HS1726A (Σχήμα 3.2) είναι κατασκευασμένη από ελαφρύ κράμα αλουμινίου το οποίο προσαρμόζεται σε όλο το επιφανειακό πλαίσιο του Raspberry μεγιστοποιώντας έτσι την διασπορά της θερμότητας. Η παθητική ψύξη προτιμάται σε μικρές εφαρμογές, καθώς παρότι οι λύσεις ενεργητικής ψύξης επιτυγχάνουν χαμηλότερες θερμοκρασίες συνήθως παράγουν αυξημένο θόρυβο λόγω των κινούμενων μερών (Π.χ. ανεμιστήρας).

Η τοποθέτηση της ψήκτρας στην επιφάνεια του επεξεργαστή είναι καθοριστική για την πρόληψη φαινομένων υπερθέρμανσης. Το μοντέλο HS1726A μπορεί να διατηρήσει την θερμοκρασία της πλατφόρμας ακόμη και σε συνθήκες υψηλού φόρτου, κάτω από τους 75°C, εξασφαλίζοντας αθόρυβη και σταθερή λειτουργία του συστήματος.

Σχήμα 3.2 Θήκη αλουμινίου Raspberry Pi 4 HS1726A

[https://www.hellasdigital.gr/go-create/raspberry-and-accessories/raspberry-pi-4-and-accessories/raspberry-pi-4-aluminum-metal-case/]

3.1.2.2 Κάρτα ήχου Vention

Αναπόσπαστο τεχνικό στοιχείο στα συστήματα φωνητικού ελέγχου είναι η συσκευή εισόδου δεδομένων. Το Raspberry Pi 4 διαθέτει αναλογική θύρα jack 3.5 mm, η οποία υποστηρίζει μόνο έξοδο ήχου και δεν παρέχει δυνατότητα σύνδεσης μικροφώνου. Η απαίτηση αυτή καλύπτεται μέσω εξωτερικών καρτών ήχου USB ,που προσφέρουν στο σύστημα θύρες εισόδου [19].

Η κάρτα ήχου Vention (Σχήμα 3.3) διαθέτει δύο ανεξάρτητες θύρες jack 3.5mm, μία για μικρόφωνο και μία για ακουστική συσκευή, επιτρέποντας ταυτόχρονη είσοδο και έξοδο σήματος υψηλής ποιότητας. Χαρακτηριστικό πλεονέκτημα της είναι η πλήρης συμβατότητα με το λειτουργικό Pi OS, χωρίς να απαιτείται περαιτέρω εγκατάσταση πρόσθετων οδηγών.

Η χρήση εξωτερικής κάρτας ήχου βελτιστοποιεί την ποιότητα καταγραφής, εξασφαλίζοντας ακρίβεια και αξιοπιστία στην διαδικασία αναγνώρισης φωνής, ενώ ταυτόχρονα παρέχει ποιοτικό σήμα εξόδου για την ανατροφοδότηση του χρήστη, απαλλαγμένο από ηλεκτρικά παράσιτα που μπορεί να προέρχονται από την πλακέτα.

Σχήμα 3.3 External Soundcard Vention USB 2.0

[https://ventiontech.com/products/usb-external-sound-card?srsltid=AfmBOoqp5cVL3WQYgVH9qz-fEuGrhqDOy0MWZa3tg75LqFsFpc7a9LEO]

3.1.2.3 Μικρόφωνο Awei MK1

Η προσάρτηση εξωτερικού μικροφώνου στην κάρτα ήχου USB που συνδέεται με το Raspberry Pi αποτελεί το τελικό στάδιο συνδεσμολογίας για την επιτυχή αλληλεπίδραση του χρήστη με το σύστημα.

Το Awei MK1 (Σχήμα 3.4) είναι πυκνωτικό μικρόφωνο πέτου με παντοκατευθυντική λήψη, σχεδιασμένο για αναλογικά συστήματα μέσω θύρας jack 3.5 mm. Στο εξωτερικό του περίβλημα χρησιμοποιεί φίλτρο για την καταστολή θορύβου ενώ η ευαισθησία του ορίζεται στα -36 dB, επιπλέον διαθέτει αντίσταση εισόδου 2,2 kΩ , ενώ η απόκριση συχνότητας για την λήψη δειγμάτων είναι 16kHz.

Η χρήση του Awei MK1 στην παρούσα διπλωματική εργασία συμβάλλει ουσιαστικά στην διαδικασία καταγραφής φωνητικών δεδομένων καθώς αποτελεί το κύριο μέσο εισόδου. Τα ενσωματωμένα φίλτρα

βελτιώνουν την καθαρότητα του σήματος μειώνοντας τον περιβαλλοντικό θόρυβο, ενώ η ευαισθησία του περιορίζει την απόσταση ομιλίας με το μέγιστο όριο να είναι τα 20cm .

Σχήμα 3.4 Awei clipper microphone MK1

[https://www.easytechnology.gr/information-technology-and-tablet/computers/pc-peripherals/pc-headset-microphones/awei-clipper-microphone-mk1]

3.1.2.4 Ηχεία

Η εγκατάσταση εξωτερικών ηχείων κρίθηκε απαραίτητη για την ολοκλήρωση του συστήματος φωνητικής αλληλεπίδρασης καθώς μέσω αυτών παρέχονται σε πραγματικό χρόνο οι ακουστικές απαντήσεις.

Τα Trust Remo 2.0 Speaker (Σχήμα 3.5) αποτελούν ζεύγος εξωτερικών ηχείων κατηγορίας 2.0, σχεδιασμένων για χρήση με υπολογιστικά συστήματα. Διαθέτουν ισχύ εξόδου 8 Watt RMS και τροφοδοτούνται αποκλειστικά μέσω θύρας USB χωρίς εξωτερικό τροφοδοτικό.

Η διασύνδεση τους με το Raspberry Pi πραγματοποιείται μέσω υποδοχής 3.5 mm jack, η οποία συνδέεται στην θύρα της ενσωματωμένης κάρτας ήχου Vention. Η τροφοδοσία τους παρέχεται απευθείας από την ενσωματωμένη θύρα USB 2.0 του συστήματος.

Σχήμα 3.5 Ηχεία Remo 2.0 Speaker Set

[www.trust.com/en/product/17595-remo-20-speaker-set]

3.1.2.5 Ταινία RGB LED

Το σύστημα υποστηρίζει έλεγχο φωτισμού μέσω κατάλληλης θύρας PWM, ο οποίος υλοποιείται πρακτικά με την εγκατάσταση ταινίας RGB LED.

Η Casalux (Σχήμα 3.6) έχει συνολικό μήκος 5 m ,με διάταξη κοινής ανόδου αποτελούμενη από συστοιχία 270 LED τύπου SMD3528 με συνολική κατανάλωση 20 W. Η τροφοδοσία της παρέχεται από εξωτερικό τροφοδοτικό ενώ υποστηρίζεται και ο έλεγχος μέσω τηλεχειρισμού.

Στο παρόν έργο γίνεται χρήση μόνο ενός μέτρου (54 LED / m) από το συνολικό μήκος της , το οποίο ενσωματώνεται απευθείας στην πρότυπη πλακέτα Maestro VF για έλεγχο μέσω PWM.

Σχήμα 3.6 Ταινία RGB LED 12V

[https://wox.gr/collections/casalux/products/globalexpress-a-led-5050-rgbw-10-12v-5a-gl-55339]

3.1.2.6 Πρίζα ράγας

Η προστασία από ηλεκτρολογικούς κινδύνους είναι βασική προτεραιότητα στις κατασκευές συστημάτων που φέρουν υψηλές τάσεις. Για την διασφάλιση της παροχής ρεύματος και της προστασίας των χρηστών κρίθηκε απαραίτητη η τοποθέτηση πρίζας ράγας. Οι πρίζες αυτού του τύπου είναι σχεδιασμένες για απευθείας τοποθέτηση σε ηλεκτρολογικούς πίνακες και επιτρέπουν την εύκολη διανομή ισχύος σε εξωτερικές συσκευές ή βοηθητικά κυκλώματα.

Η πρίζα SGM 6 Schuko (Σχήμα 3.7) αποτελεί ένα εξάρτημα τύπου ράγας DIN, το οποίο χρησιμοποιείται για την ασφαλή παροχή τάσης σε ηλεκτρολογικούς πίνακές και κυκλώματα αυτοματισμού. Υποστηρίζει τυποποιημένο βύσμα Schuko (CEE 7/3) προσφέροντας συμβατότητα με οικιακές και βιομηχανικές συσκευές. Είναι κατάλληλη για τάση δικτύου 230 AC με ονομαστική ένταση έως 16 A εξασφαλίζοντας αξιόπιστη λειτουργία.

Στην παρούσα εφαρμογή τοποθετήθηκε στο σώμα της κατασκευής ηλεκτρολογική ράγα, επάνω στην οποία εγκαταστάθηκαν τρία στοιχεία πρίζας για την σύνδεση των I/O του Maestro VF. Στόχος αυτής της εγκατάστασης ήταν η εύκολη προσάρτηση εξωτερικών συσκευών στα άκρα του συστήματος χωρίς ο χρήστης να έρχεται σε άμεση επαφή με τις κλέμμες του κυκλώματος. Η οδήγηση των εξόδων στα άκρα της πρίζας πραγματοποιήθηκε με την χρήση κατάλληλου ηλεκτρολογικού καλωδίου .

Σχήμα 3.7 Πρίζα ράγας SGM 6 Schuko

[www.maxge.eu/en_GB/shop/sgms6-tomas-schuko-1063#attr=10634]

3.1.3 Maestro VF

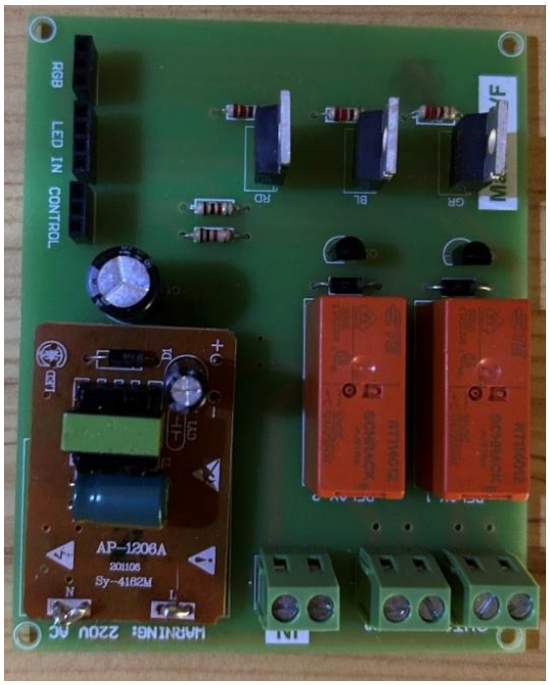
Στο πλαίσιο αρχικής σχεδίασης ορίστηκαν ρητά οι στόχοι του έργου, με κύριο ζητούμενο οι χρήστες να μπορούν να ελέγχουν οικιακές συσκευές από απόσταση με ευκολία και αξιοπιστία. Αφενός ένας μικροϋπολογιστής όπως το Raspberry Pi 4 διαθέτει επαρκή υπολογιστική ισχύ για την διαχείριση της λογικής εκτέλεσης του συστήματος ,αφετέρου πρόκειται για ένα αναπτυξιακό εργαλείο υψηλής ευαισθησίας το οποίο απαιτεί προστασία από φαινόμενα υπερτάσεων και υπερθέρμανσης.

Παρά την υποστήριξη των GPIO pins που παρέχουν δυνατότητα απευθείας διασύνδεσης με ηλεκτρονικά στοιχεία, το Raspberry Pi προσφέρει χαμηλές τάσεις (3.3-5 V) ενώ μπορεί να παρέχει ρεύμα περίπου 16 mA ανά pin.Αυτό καθιστά απαραίτητη την σχεδίαση ενός νέου ηλεκτρονικού κυκλώματος οδήγησης και ελέγχου ,ικανού να χειρίζεται συσκευές υψηλής ισχύος.

Αντίστοιχα σχεδιάστηκε το Maestro VF (Σχήμα 3.8) , ένα πρωτότυπο κύκλωμα οδήγησης συσκευών χαμηλής τάσης ανεπτυγμένο στο περιβάλλον Altium Designer. Το Maestro VF τροφοδοτείται με 220 V AC και διαθέτει τρείς εξόδους , δύο για εναλλασσόμενη τάση 220 V και μία που παρέχει 12 V DC για την ενσωμάτωση RGB κυκλωμάτων.

Για τον έλεγχο κάθε απομονωμένης κλέμμας εξόδου (OUT1-2) και του pin header (RGB LED IN) αναπτύχθηκαν επιμέρους κυκλώματα τα οποία ελέγχονται από τον χρήστη με την βοήθεια λογισμικού.

Η αρχιτεκτονική τους βασίζεται σε ηλεκτρονόμους και τρανζίστορ ισχύος επιτρέποντας την ασφαλή διαχείριση των φορτίων και την απομόνωση του Raspberry Pi από τις υψηλές τάσεις και τα ρεύματα που απαιτούν οι εξωτερικές συσκευές.



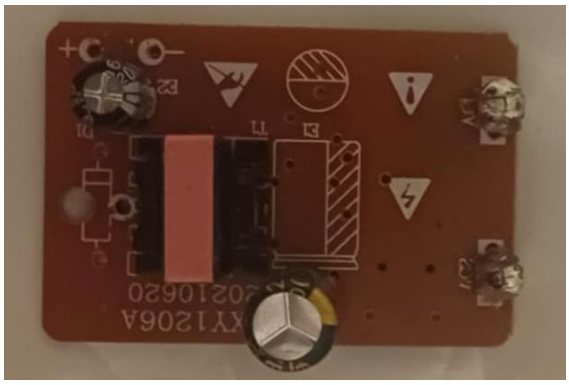
Σχήμα 3.8 Maestro VF

3.1.3.1 Πλακέτα τροφοδοσίας PS

Ο μετασχηματισμός της εναλλασσόμενης τάσης που παρέχεται από το δίκτυο σε χαμηλή συνεχή τάση αποτελεί θεμελιώδη λειτουργία σε όλες τις συμβατικές ηλεκτρονικές συσκευές της βιομηχανίας. Για την τροφοδοσία του κυκλώματος Maestro VF εγκαταστάθηκε στο σώμα της εξωτερική πλακέτα μετατροπής AC–DC.

Η πλακέτα τροφοδοσίας PS (44 mm x 30 mm) (Σχήμα 3.9) είναι ένα ολοκληρωμένο κύκλωμα μετατροπής AC- DC, αποτελούμενο από μετασχηματιστή 220 V/12 V AC, μία γέφυρα ανόρθωσης τεσσάρων διόδων, έναν πυκνωτή εξομάλυνσης για τη μείωση της κυμάτωσης και ένα SMD κύκλωμα σταθεροποίησης τάσης το οποίο διατηρεί την έξοδο σταθερά στα 12 V DC.

Η πλακέτα PS κρίθηκε καταλληλότερη για το Maestro VF, καθώς παρουσιάζει μικρότερες διακυμάνσεις στην τάση εξόδου σε σχέση με το ενσωματωμένο κύκλωμα τροφοδοσίας της πλακέτας και μπορεί να τοποθετηθεί εύκολα στο εσωτερικό της κατασκευής. Επιπλέον, το κύκλωμα διαθέτει συμπαγή σχεδίαση και τοποθετείται εύκολα στο εσωτερικό του Maestro VF, καθώς είναι προσαρμοσμένο στις διαστάσεις της κατασκευής.



Σχήμα 3.9 Πλακέτα τροφοδοσίας PS

3.1.3.2 Ηλεκτρονόμος (Ρελέ) RT114012

Ο ηλεκτρονόμος είναι ένας ηλεκτρομηχανικός διακόπτης ο οποίες χρησιμοποιείται σε εφαρμογές όπου απαιτείται διαχείριση φορτιών υψηλής τάσης και έντασης μέσω κυκλωμάτων χαμηλής τάσης, όπως εκείνα των μικροελεγκτών και των τρανζίστορ.

Το RT114012 (Σχήμα 3.10) είναι ένα ρελέ μονής επαφής με ονομαστική τάση λειτουργίας πηνίου στα 12V DC. Η λειτουργία του βασίζεται στην διάταξη μίας κοινής επαφής C (Common) η οποία μπορεί να μεταβληθεί από την κατάσταση NC-C (Normally Closed) σε NO-C (Normally Open) όταν εφαρμοστεί τάση στα άκρα του εσωτερικού του πηνίου με το ρεύμα διέλευσης να αγγίζει τα 40mA [21].

Η χρήση ρελέ στην παρούσα διπλωματική εργασία κρίθηκε απαραίτητη για τον μηχανικό έλεγχο και την ηλεκτρική απομόνωση των εξόδων υψηλής τάσης από το κύκλωμα του μικροϋπολογιστή. Το Raspberry Pi λόγω των εγγενών ηλεκτρονικών περιορισμών δεν διαθέτει αντίστοιχη ανθεκτικότητα σε συνθήκες υπερφόρτωσης ούτε μπορεί να λειτουργήσει ως διακόπτης σε φορτία υψηλής ισχύος για τον λόγο αυτό η ενσωμάτωση ρελέ κρίνεται κατάλληλη επιλογή.

Σχήμα 3.10 Ηλεκτρονόμος SPDT 12 V

[https://gr.rsdelivers.com/en/product/te-connectivity/rt114012/te-connectivity-pcb-mount-power-relay-12v-dc-coil/1894070?backToResults=1]

3.1.3.3 Δίοδος 1N4007

Η συμπεριφορά των ρελέ όταν η εφαπτόμενη τάση στα άκρα του πηνίου τους διακόπτεται, οδηγεί σε άμεση κατάρρευση του μαγνητικού πεδίου. Το φαινόμενο αυτό έχει ως συνέπεια την παραγωγή ανάστροφης τάσης γνωστή ως flyback. Η υπέρταση που δημιουργείται μπορεί να προκαλέσει βλάβες τόσο στο ίδιο το ρελέ όσο και στα υπόλοιπα στοιχεία του κυκλώματος. Γι' αυτόν τον λόγο χρησιμοποιείται μία προστατευτική δίοδος, γνωστή και ως flyback diode [23].

Η δίοδος 1N4007 (Σχήμα 3.11) είναι ένα ημιαγωγικό εξάρτημα ανόρθωσης σχεδιασμένο για εφαρμογές συνεχούς ρεύματος και χαμηλής τάσης. Η μέγιστη ανάστροφη τάση του ορίζεται στα 1000V ενώ το μέγιστο ρεύμα αγγίζει το 1A. Το στοιχείο χρησιμοποιείται ευρέως σε εφαρμογές προστασίας κυκλωμάτων από υπερτάσεις καθώς τείνει να απορροφά την αποθηκευμένη ενέργεια του πηνίου όταν το μαγνητικό πεδίο καταρρέει.[22]

Η δίοδος 1N4007 χρησιμοποιήθηκε στην παρούσα εργασία αμιγώς για λόγους προστασίας του κυκλώματός οδήγησης των ρελέ, διασφαλίζοντας ομαλή μετάβαση μεταξύ των καταστάσεων διέγερσης και αποδιέγερσης.

Σχήμα 3.11 Δίοδος 1N4007

[https://www.vishay.com/en/product/88503/]

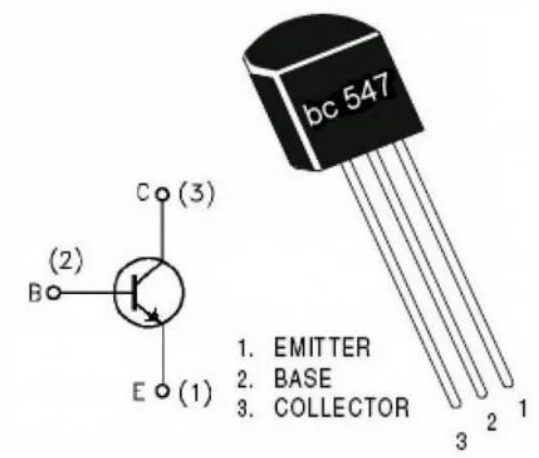
3.1.3.4 Ημιαγωγός(Transistor) BC547

Η παρουσία των τρανζίστορ σε αναπτυξιακά και βιομηχανικά κυκλώματα είναι καίριας σημασίας, καθώς χρησιμοποιούνται σε βασικές εφαρμογές όπως η ενίσχυση σημάτων, ο έλεγχος ψηφιακών διακοπτών και ή οδήγηση επαγωγικών φορτίων.

Το BC547 (Σχήμα 3.12) είναι ένας ημιαγωγός τύπου NPN τριών ακροδεκτών(Συλλέκτης, βάση, εκπομπός),ευρέως διαδεδομένος λόγω του υψηλού συντελεστή κέρδους hFE (110-800) και της χαμηλής κατανάλωσης ισχύος. Επιπλέον απαιτεί ελάχιστο ρεύμα βάσης περίπου $I_b = 3\ mA$ ενώ ο συλλέκτης του μπορεί να διαχειριστεί επαρκώς ρεύματα έως και 100 mA [24].

Η επιλογή του BC547 στην παρούσα εφαρμογή συνίσταται στα μικροκυκλώματα οδήγησης των ρελέ. Παρότι οι θύρες GPIO του Raspberry Pi δεν πληρούν τις προϋποθέσεις για την ενεργοποίηση του

πηνίου, μπορούν να οδηγήσουν να οδηγήσουν αποτελεσματικά την βάση τρανζίστορ χωρίς να απαιτείται υψηλό ρεύμα. Ταυτόχρονα το τρανζίστορ μπορεί να οδηγήσει επαρκώς έως και δύο ρελέ διατεταγμένα σε σειρά με τον συλλέκτη του δεδομένου ότι η απαίτηση ρεύματος κάθε πηνίου δεν υπερβαίνει τα 40mA .



Σχήμα 3.12 Τρανζίστορ NPN BC547

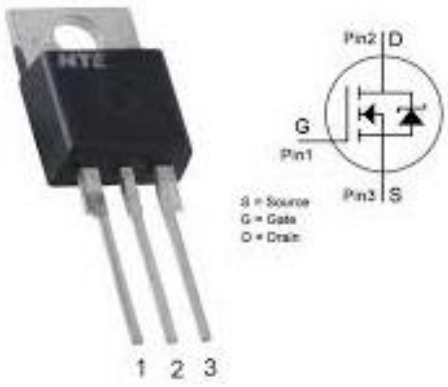
[https://eu.mouser.com/ProductDetail/onsemi-Fairchild/BC547B?qs=UMEuL5FsraB3zD25tclGGQ%3D%3D]

3.1.3.5 Ημιαγωγός ισχύος (MOSFET) IRLZ44

Το IRLZ44 (Σχήμα 3.13) είναι ένα τρανζίστορ ισχύος τύπου N-channel MOSFET, σχεδιασμένο για εφαρμογές χαμηλής τάσης οδήγησης με αντοχές διαχείρισης υψηλών ρευμάτων. Η λειτουργία των MOSFET διακρίνεται σε δύο βασικές κατηγορίες όπου το στοιχείο μπορεί να λειτουργεί ως ενισχυτής ή ηλεκτρονικός διακόπτης. Η δομή του MOSFET περιλαμβάνει τρείς ακροδέκτες :Gate, Source, Drain όπου η δημιουργία αγώγιμου καναλιού μεταξύ Drain και Source επιτυγχάνεται όταν η εφαρμοζόμενη τάση υπερβεί την τάση κατωφλίου.

Το IRLZ44 διαθέτει χαμηλή τάση κατωφλίου 1-2 V η οποία ταιριάζει ιδανικά σε εφαρμογές όπου επιθυμείται έλεγχος από πλατφόρμες όπως το Raspberry Pi καθώς τα σήματα λογικού ελέγχου δεν ξεπερνούν τα 3.3-5 V. Επιπλέον το IRL διαθέτει πολύ χαμηλή εσωτερική αντίσταση αγωγής $R_{DS(ON)}$ =0.022 Ω, γεγονός που σημαίνει ότι οι απώλειες λόγο θερμότητας είναι μειωμένες ακόμα και όταν τα ρεύματα είναι υψηλά [25].

Η επιλογή του IRLZ44 στην παρούσα εργασία θεωρείται κατάλληλη, καθώς μπορεί να διαχειριστεί φορτία RGB LED 12V, επιτρέποντας τον έλεγχο της φωτεινότητας κάθε καναλιού μέσω PWM σημάτων. Αυτό σημαίνει πώς το κύκλωμα του Raspberry Pi μπορεί να ρυθμίσει την φωτεινότητα των LED διατηρώντας χαμηλή την κατανάλωση χωρίς να απαιτείται κάποιο ενδιάμεσο κύκλωμα οδήγησης αφού η εκείνο μπορεί να λειτουργήσει απευθείας με τα σήματα ελέγχου της πλατφόρμας.



Σχήμα 3.13 N-Channel Mosfet IRLZ44N

[https://www.vishay.com/en/product/91328/]

3.1.3.6 Τερματικό σύνδεσης TE_282836-2

Η διασύνδεση των κυκλωμάτων με εξωτερικές πηγές είτε για τροφοδοσία, είτε για λήψη και μετάδοση σημάτων είναι βασική προϋπόθεση για την ομαλή λειτουργία κάθε ηλεκτρονικού συστήματος. Την κρίσιμη αυτή λειτουργία αναλαμβάνουν να επιτελέσουν τα τερματικά σύνδεσης.

Το TE Connectivity 282836-2 (Σχήμα 3.14) είναι διπολικό τερματικό σύνδεσης με βήμα επαφών 5 mm, σχεδιασμένο για την συγκράτηση καλωδίων σε τυποποιημένα κυκλώματα μέσω του μηχανισμού σπειρωτής σύσφιξης. Χάρη στα τεχνικά του χαρακτηριστικά αποτελεί αξιόπιστη λύση για σύνδεση καλωδίων οικιακής παροχής καθώς η ονομαστική τάση λειτουργίας στα 300V, ενώ το μέγιστο ρεύμα ανέρχεται στα 13.5 A [26].

Η επιλογή του TE_282836-2 στην κατασκευή του Maestro VF χρησιμοποιείται τόσο για την παροχή τροφοδοσίας στο βασικό κύκλωμα του PCB όσο και για την σύνδεση των φορτιών εξόδου των 220V. Η υψηλή μηχανική αντοχή και η θερμική σταθερότητα της κατασκευής του προσφέρουν ασφαλή και σταθερή λειτουργία ακόμη και σε περιπτώσεις υψηλών ρευμάτων.

Σχήμα 3.14 Τερματικό σύνδεση δύο επαφών

[https://eu.mouser.com/ProductDetail/TE-Connectivity/282836-2?qs=A%252Bip%252BNCYi6Ohx9vaQ1l4dg%3D%3D]

3.1.3.7 Αντιστάσεις 220 Ω & 1 kΩ

Ο ρόλος των αντιστάσεων στην λειτουργία των κυκλωμάτων είναι θεμελιώδης .Οι αντιστάσεις ,οι οποίες κατασκευάζονται συνήθως από κεραμικά ή σύνθετα υλικά τα οποία περιορίζουν την ροή του ηλεκτρικού ρεύματος προστατεύοντας με αυτόν τον τρόπο τα ευαίσθητα ηλεκτρονικά εξαρτήματα από υπερφορτίσεις ή καταστροφές.

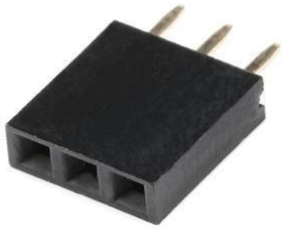
Η συγκεκριμένη εφαρμογή περιλαμβάνει αντιστάσεις 220 Ω και 1 kΩ (Σχήμα 3.15). Η πρώτη είναι τοποθετημένη στο κύκλωμα ελέγχου ρύθμισης φωτισμού RGB και σκοπός της είναι να περιορίσει το ρεύμα φόρτισης/εκφόρτισης της πύλης (GATE) του MOSFET, το οποίο οδηγείται από την ακίδα GPIO του Raspberry Pi. Η δεύτερη τοποθετείται στην βάση του τρανζίστορ ελέγχου προκειμένου να περιορίσει το ρεύμα που διαρρέει το κύκλωμα οδήγησης του ρελέ για την αποφυγή υπερθέρμανσης.

Σχήμα 3.15 Αντίσταση

3.1.3.8 Dupont Female Header

Ένα κύκλωμα είναι απαραίτητο να περιλαμβάνει τερματικά για την σύνδεση της τροφοδοσίας του. Εξίσου σημαντικό όμως είναι να διαθέτει κατάλληλες θύρες για την σύνδεση φορτίων χαμηλής τάσης όπως εξερχόμενα/εισερχόμενα ψηφιακά σήματα. Για την διασύνδεση του Maestro VF με το Raspberry Pi εγκαταστάθηκαν τρείς θηλυκοί Connectors (3-pin, 4-pin) τύπου Dupont.

Το Dupont Female Header (Σχήμα 3.16) είναι ένας τυποποιημένος σύνδεσμος χαμηλής τάσης, κατασκευασμένος από πλαστικό υλικό, στο εσωτερικό του περιλαμβάνει με μονωμένους ακροδέκτες με βήμα 2.54 mm και θηλυκή υποδοχή για την προσάρτηση καλωδίων. Στην συγκεκριμένη εφαρμογή το Maestro VF αξιοποιεί την δομή του για την απευθείας σύνδεση και επικοινωνία με το Raspberry Pi 4.



Σχήμα 3.16 DuPont Connector

[https://protostack.com.au/shop/connectors/dupont-style-connectors/4-pin-dupont-style-connector-housing/]

3.2 Λογισμικό και τεχνολογίες

Σε αυτήν την ενότητα παρουσιάζεται το λογισμικό το οποίο αναπτύχθηκε καθώς και τα επιμέρους δομικά χαρακτηριστικά που χρησιμοποιήθηκαν για την ανάπτυξη του.

3.2.1 MVF Software

Η διαχείριση ενός έξυπνου συστήματος βασίζεται κατά κατεξοχήν στην στενή αλληλεπίδραση λογισμικού και υλικού. Στο πλαίσιο δημιουργία της παρούσας διπλωματικής εργασίας, το λογισμικό λειτουργεί ως «νευρικό σύστημα» αναλαμβάνοντας την επεξεργασία των φωνητικών εντολών που παράγει ο χρήστης και την μετατροπή τους σε εντολές μηχανής. Οι εντολές αυτές κατευθύνονται προς τις εξόδους GPIO του Raspberry Pi 4, οι οποίες με την σειρά τους καθοδηγούν το κύκλωμα εκτέλεσης.

Για την υποστήριξη αυτών των λειτουργιών αναπτύχθηκε το MVF Software. Ένα πρωτότυπο και πλήρως εξατομικευμένο λογισμικό που εκτελείται τοπικά στο Raspberry Pi αποτελώντας τον κεντρικό πυρήνα ελέγχου του συστήματος. Η αρχιτεκτονική του έχει σχεδιαστεί έτσι ώστε να συνεργάζεται αποτελεσματικά με την αναπτυξιακή πλακέτα Maestro VF διασφαλίζοντας σταθερή και αξιόπιστη απόδοση.

Το MVF Software έχει αναπτυχθεί εξ ολοκλήρου στη γλώσσα προγραμματισμού Python προσαρμοσμένο στις λειτουργικές απαιτήσεις του έργου. Ο τομέας ευθύνης του είναι εκτεταμένος, ξεκινώντας από την λήψης των εντολών και καταλήγοντας στην επεξεργασία και τελική εκτέλεση των ενεργειών στο υλικό. Για την αναγνώριση και την επεξεργασία της ομιλίας, το λογισμικό ενσωματώνει τεχνολογίες υπολογιστικής νέφους μέσω APIs, ενώ για την παροχή ανατροφοδότησης χρησιμοποιεί αντίστοιχες υπηρεσίες διεπαφών ούτος ώστε να διασφαλίσει αμφίδρομη ομαλή επικοινωνία ανάμεσα στον χρήστη και το σύστημα.

3.2.2 Python

Η πολύγλωσσή ανάπτυξη λογισμικού είναι μία καθιερωμένη πρακτική σε εφαρμογές IoT και Cloud συστημάτων, καθώς η κάθε γλώσσα μπορεί να προσφέρει διαφορετικά πλεονεκτήματα ανάλογα με υποσύστημα που καλείται να υλοποιήσει. Σημαντικοί παράγοντες για την επιλογή της κατάλληλης γλώσσας είναι η επεκτασιμότητα, η ευέλικτη ανάπτυξη και η υψηλή προσαρμοστικότητα σε διάφορα περιβάλλοντα.

Για την υλοποίηση του παρόντος λογισμικού επιλέχθηκε η Python, μία από τις πλέον διαδεδομένες γλώσσες προγραμματισμού που χρησιμοποιείται εκτενώς σε διάφορους τομείς όπως η τεχνητή νοημοσύνη, τα backend λογισμικά και οι εφαρμογές αυτοματισμού. Χαρακτηριστικά στοιχεία της αποτελούν η απλοϊκή σύνταξη εντολών, η ταχύτητα εκμάθησης και η πλούσια συλλογή βιβλιοθηκών και πακέτων που υποστηρίζει για την ανάπτυξη απλών και σύνθετων υλοποιήσεων.

Στο συγκεκριμένο σύστημα η Python αξιοποιείται για την ανάπτυξη του τμήματος του λογισμικού που ενσωματώνει υπηρεσίες διεπαφών STT και TTS δημιουργώντας ένα πλήρως διαδραστικό περιβάλλον

αλληλεπίδρασης με τον χρήστη. Παράλληλα, παρέχει δυνατότητα απευθείας διασύνδεσης με το υλικό μέσω των θυρών GPIO για τον έλεγχο ηλεκτρονικών στοιχείων, όπως ρελέ και MOSFET καθιστώντας την κατάλληλη για εφαρμογές αυτοματισμού.

Η επιλογή της συνέβαλε καθοριστικά στην ταχεία και αξιόπιστη ανάπτυξη του κώδικα, διασφαλίζοντας την ομαλή συνεργασία όλων των επιμέρους στοιχείων τόσο σε επίπεδο υλικού όσο και λογισμικού. Επιπλέον η αρχιτεκτονική της γλώσσας υποστηρίζει εύκολη συντήρηση με δυνατότητες επεκτάσεων ορίζοντας ένα ευέλικτο και προσαρμοστικό σύστημα.

3.2.3 APIs

Τα API αποτελούν θεμελιώδη δομικά στοιχεία λογισμικού τα οποία προσφέρουν τυποποιημένες διεπαφές επικοινωνίας μεταξύ διαφορετικών εφαρμογών. Μέσω ενός API καθίσταται δυνατή η συνεργασία μεταξύ προγραμμάτων χωρίς να απαιτείται πρόσβαση στον εσωτερικό τους κώδικα ή τροποποίηση της δομής τους.

Η χρήση APIs προσφέρει σημαντικά πλεονεκτήματα κατά την ανάπτυξη ενός προγράμματος. Η ενσωμάτωση τους είναι απλή ενώ σπάνια απαιτούνται εκτενείς τροποποιήσεις στην σύνταξη του κώδικα. Μια διεπαφή συχνά αναπαριστά την σχέση μεταξύ πελάτη και υπηρεσίας, διευκολύνοντας την ανταλλαγή ακατέργαστων και μη δεδομένων. Επιπλέον σημαντικό χαρακτηριστικό των APIs είναι η διαλειτουργικότητα. Οι σύγχρονες εφαρμογές αναπτύσσονται σε διαφορετικά περιβάλλοντα και γλώσσες προγραμματισμού, ένα API επιτρέπει την συνεργασία τους ανεξαρτήτως γλώσσας και πλατφόρμας [27].

Η αφαιρετική ιδιότητα των APIs απλοποιεί την διαδικασία ανάπτυξης, μειώνει την πολυπλοκότητα του κώδικα και αυξάνει την ευελιξία του. Η χρήση τους σημαίνει αξιοποίηση μόνο των λειτουργιών που απαιτούνται για την εκάστοτε εφαρμογή ενώ παράλληλα λειτουργούν ανεξάρτητα χωρίς να επηρεάζουν τα υπόλοιπα υποσυστήματα.

Η δομή της εργασίας αξιοποιεί APIs και στα τρία στάδια λειτουργίας του συστήματος: είσοδο ,επεξεργασία και έξοδο. Στο πρώτο στάδιο τα δεδομένα εισόδου εισάγονται μέσω φωνητικών εντολών, μία διαδικασία ιδιαίτερα απαιτητική καθώς περιλαμβάνει αναγνώρισή, ανάλυση και μετατροπή του προφορικού λόγου σε κείμενο. Η πολυπλοκότητα αυτής της διεργασίας καθιστά αδύνατη την υλοποίηση της από έναν μεμονωμένο προγραμματιστή, γεγονός που αναδεικνύει τον κρίσιμό ρόλο των APIs.

Για την υλοποίηση της φωνητικής διεπαφής ενσωματώθηκαν δύο βασικά APIs. Το Deepgram για την αναγνώριση και μετατροπή των φωνητικών εντολών σε κείμενο και το ElevenLabs για την παραγωγή φυσικών ηχητικών μηνυμάτων προς τον χρήστη. Με αυτόν τον τρόπο εξασφαλίζεται αξιόπιστη επικοινωνία σε πραγματικό χρόνο και φυσικότητα στην αλληλεπίδραση του χρήστη με το σύστημα.

3.2.3.1 Deepgram

Το Deepgram είναι μια πλατφόρμα φωνητικής αναγνώρισης, η οποία παρέχει υπηρεσίες και μοντέλα μέσω API. Η δομή του βασίζεται σε τεχνικές βαθιάς εκμάθησης και στην ανάπτυξη νευρωνικών δικτύων, τα οποία έχουν εκπαιδευτεί με μεγάλο όγκο ηχητικών δεδομένων. Τα δίκτυα αυτά αξιοποιούνται για την ανάλυση φωνητικών σημάτων και την εξαγωγή τους αργότερα σε επεξεργάσιμα δεδομένα έτσι ώστε να χρησιμοποιηθούν από τις διασυνδεμένες εφαρμογές της πλατφόρμας [28].

Βασικές αρχές που διέπουν το σύστημα αποτελούν η αξιοπιστία και η ταχεία απόκριση, στοιχεία απαραίτητα για εφαρμογές αυτοματισμού. Η διεπαφή επιτυγχάνει εξαιρετικά χαμηλή καθυστέρηση μικρότερη των 300 ms, ενώ ταυτόχρονα παρουσιάζει πολύ χαμηλό ποσοστό σφάλματος ακόμα και σε λειτουργίες συνεχούς ροής με τιμές που δεν ξεπερνούν το 10%.

Αξιοσημείωτο χαρακτηριστικό της πλατφόρμας είναι η ενσωμάτωση φίλτρων αναγνώρισης σε υψηλό επίπεδο. Συγκεκριμένα παρέχεται δυνατότητα αναγνώρισης στίξης, εντοπισμού διαφορετικών ομιλητών καθώς και ρύθμιση ευαισθησίας. Επιπλέον υποστηρίζει ανίχνευση γλωσσών, είτε σε μονόγλωσσά περιβάλλοντα είτε με εναλλαγές μέσω του Nova-3 μοντέλου το οποίο καλύπτει έως και τριάντα έξι γλώσσες.

Η επικοινωνία του API με το λογισμικό λαμβάνει χώρα σε πραγματικό χρόνο μέσω Websocket αιτημάτων. Το SDK εξασφαλίζει την υποδομή του Websocket ώστε να ολοκληρωθεί επιτυχώς η επικοινωνία με την πλατφόρμα, ενώ περιέχει ακόμα βιβλιοθήκες για την διαχείριση γεγονότων και τις αντίστοιχες κλάσεις τους [28] .

3.2.3.2 ElevenLabs

Το ElevenLabs είναι μια πλατφόρμα σύνθεσης φυσικής ομιλίας, βασισμένη σε τεχνικές τεχνητής νοημοσύνης η οποία παρέχει υπηρεσίες μέσω API. Σε αντίθεση με το Deepgram που εστιάζει στην φωνητική αναγνώριση, το ElevenLabs εξειδικεύεται στην δημιουργία ρεαλιστικών φωνητικών συνθέσεων αντλώντας δεδομένα από εκτενείς βάσεις ηχητικών δειγμάτων. Μέσω της επεξεργασίας τους, παράγονται ηχητικά μηνύματα με φυσική έκφραση, κατάλληλα για συστήματα αλληλεπίδρασης σε πραγματικό χρόνο [29].

Όσο αφορά τις ομοιότητες με το Deepgram πρόκειται για ακόμη επίσης για μια πλατφόρμα με μικρό χρόνο απόκρισης στις streaming υπηρεσίες, ο οποίος κυμαίνεται από 300 έως 700 ms .Ωστόσο η καθυστέρηση μειώνεται σημαντικά σε σύντομες προτάσεις με χρόνο κάτω από 400 ms. Το API υποστηρίζει ένα είκοσι εννέα διαφορετικές γλώσσες ,οι οποίες διακρίνονται για τον υψηλό ρεαλισμό και την φυσική ροή λόγου με συναισθηματικό ύφος. Οι συνθέσεις αυτές είναι εφικτές χάρη στα νευρωνικά δίκτυα στα οποία στηρίζει την δομή της η πλατφόρμα, στοιχείο το οποίο μπορεί να αξιοποιηθεί μεταγενέστερα για την δημιουργία εξατομικευμένων φωνητικών μοντέλων.

3.3 Εργαλεία ανάπτυξης

Σε αυτήν την ενότητα παρουσιάζονται τα εργαλεία λογισμικού τα οποία χρησιμοποιήθηκα για την συγγραφή του πηγαίου κώδικα του MVF και την σχεδίαση της πρότυπης πλακέτα Maestro VF.

3.3.1 Altium Designer

Η ανάγκη δημιουργίας ενός κυκλώματος που θα λειτουργεί ως εκτελεστικό όργανο για την υλοποίηση των αποφάσεων του μικροϋπολογιστή Raspberry Pi οδήγησε στην ανάπτυξη του Maestro VF. Στο πλαίσιο της σχεδίασης του επιλέχθηκε το Altium designer, ένα από τα πλέον διαδεδομένα και προηγμένα λογισμικά ανάπτυξης PCB, το οποίο πληροί βιομηχανικά πρότυπα και υποστηρίζει ολοκληρωμένα εργαλεία σχεδίασης [30].

Το Altium Designer είναι ένα ολοκληρωμένο λογισμικό σχεδίασης ηλεκτρονικών κυκλωμάτων της εταιρείας Altium Limited που συνδυάζει δυνατότητες δημιουργίας ηλεκτρονικών διαγραμμάτων, τρισδιάστατης μοντελοποίησης και παραγωγής τυποποιημένων κυκλωμάτων. Η έκδοση 20.2.5 που χρησιμοποιήθηκε στην εργασία ενσωματώνει όλες τις απαραίτητες λειτουργίες που απαιτούνται για την ανάπτυξη πρωτότυπων PCB, προσομοιάζοντας σε περιβάλλοντα EDA (Electronic Design Automation).

Η πρώτη φάση σχεδίασης ξεκίνησε με την δημιουργία ηλεκτρονικού διαγράμματος μέσω του εργαλείου Schematic Capture. Στην συνέχεια το σχέδιο μεταφέρθηκε στο περιβάλλον PCB διατηρώντας αυτούσια όλα τα υλικά και τα αποτυπώματα σύνδεσης. Σε τελικό στάδιο αξιοποιήθηκαν ενσωματωμένα εργαλεία προσομοίωσης και μοντελοποίησης για την οπτικοποίηση και τον έλεγχο της σχεδίασης.

Πέρα από την βασική λειτουργία σχεδίασης, το Altium designer παρέχει δυνατότητες ακριβούς τρισδιάστατης απεικόνισης, δημιουργίας νέων βιβλιοθηκών εξαρτημάτων μέσω της πλατφόρμας Altium 365. Επιπλέον το λογισμικό μπορεί να συνεργαστεί τοπικά με άλλα λογισμικά μηχανολογικής σχεδίασης όπως το SolidWorks και το Siemens NX. Η πλατφόρμα του Altium διακρίνεται για την υψηλή αξιοπιστία της, υποστηρίζοντας σχεδίαση κυκλωμάτων έως και τριάντα δύο επιπέδων, ενώ προσφέρει εκτενείς βιβλιοθήκες εξαρτημάτων με πλήρη τεχνικά δεδομένα και αποτυπώματα.

Ιδιαίτερα χρήσιμα αποδείχθηκαν τα εργαλεία αποσφαλμάτωσης που εντόπιζαν ασυμβατότητες και λανθασμένες συνδέσεις, καθώς και οι επιλογές auto-routing για την βελτιστοποίηση της τοπολογίας. Μετά την ολοκλήρωση του σχεδιασμού πραγματοποιήθηκε εξαγωγή των αρχείων Gerber για την τελική παραγωγή του PCB. Η χρήση του Altium designer για την υλοποίηση του Maestro VF κάλυψε επαρκώς τις απαιτήσεις του συστήματος εξασφαλίζοντας ταχύτητα ,αξιοπιστία και επεκτασιμότητα.

3.3.2 VS CODE

Η συγγραφή και ανάπτυξη του λογισμικού MVF πραγματοποιήθηκε με κύριο περιβάλλον επεξεργασίας κώδικα το Visual Studio Code, ένα σύγχρονο και ελαφρύ εργαλείο που αναπτύχθηκε από

την Microsoft. Το VS Code είναι μία εφαρμογή ανοιχτού κώδικα που διατίθεται δωρεάν και υποστηρίζει μία πληθώρα γλωσσών προγραμματισμού ,γεγονός που το καθιστά ιδανικό για ανάπτυξη λογισμικών σε σύγχρονα επαγγελματικά και ακαδημαϊκά πρότυπα [31].

Η αρχιτεκτονική του βασίζεται στην προσθήκη επεκτάσεων, οι οποίες παρέχουν τη δυνατότητα προσαρμογής του περιβάλλοντος στις ανάγκες κάθε έργου, επιτρέποντας την ενσωμάτωση υποστήριξης νέων γλωσσών, εργαλείων αποσφαλμάτωσης και απομακρυσμένης διαχείρισης από εφαρμογές υπηρεσιών cloud.

Σε αντίθεση με πλήρη ολοκληρωμένα περιβάλλοντα ανάπτυξης IDE (Intergrated Development Enviroment) όπως το PyCharm, το VS code δεν διαθέτει εγγενές γραφικό περιβάλλον σχεδίασης ή ενσωματωμένο μεταγλωττιστή. Ωστόσο με την κατάλληλη επιλογή και εγκατάσταση επεκτάσεων μπορεί να λειτουργήσει ως ένα ευέλικτο και αποδοτικό IDE ελαφριού τύπου. Η φιλοσοφία του εστιάζει στην ταχεία εκκίνηση και στην χαμηλή δέσμευση πόρων από το σύστημα. Είναι χαρακτηριστικό ότι το μέγεθος της μνήμης που απαιτείται για την εγκατάσταση του είναι περιορισμένο ,περίπου 300 MB σε αντίθεση με άλλα περιβάλλοντα που απαιτούν μνήμη 1 – 2 GB.

Κατά την ανάπτυξη του MVF ,το VS Code παρείχε αυτοτελώς και μέσω επεκτάσεων όλα τα απαραίτητα εργαλεία για την πλήρη υλοποίηση του λογισμικού .Κατά την συγγραφή του πηγαίου κώδικα η εφαρμογή υποστήριζε χρωματική επισήμανση (syntax highlighting), η οποία διευκόλυνε την ανάγνωση, την διάκριση των εντολών και τον έλεγχο πιθανών σφαλμάτων. Επιπλέον, για την ανάπτυξη σε Python εγκαταστάθηκε η επίσημη επέκταση η οποία ενσωμάτωσε ξεχωριστό μεταγλωττιστή καλύπτοντας τις ανάγκες εκτέλεσης και ελέγχου του προγράμματος [31].

Κεφάλαιο 4ο: Ανάλυση σχεδίασης και λειτουργιών

4.1 Λειτουργία συστήματος

Σε αυτήν την ενότητα περιγράφεται και αναλύεται η εσωτερική δομή λειτουργίας του συστήματος και η σχέση αλληλεπίδρασής του με τον χρήστη.

4.1.1 Έναρξη συστήματος

Όπως συμβαίνει με τις περισσότερες σύγχρονες συσκευές οικιακού αυτοματισμού, το σύστημα απαιτεί αρχικά την σύνδεση της τροφοδοσίας του στην παροχή του δικτύου, καθώς και την ρύθμιση του Raspberry Pi στο τοπικό ασύρματο δίκτυο (Wi-Fi) μέσω δρομολογητή. Μετά την ολοκλήρωση της συνδεσμολογίας, ο χρήστης βρίσκεται σε θέση να ελέγξει πλήρως τρείς εξόδους, εκ των οποίων η μία αναφέρεται σε στοιχείο φωτισμού LED ενώ οι δύο αντιστοιχούν σε συμβατικές συσκευές.

Η εκκίνηση του συστήματος πραγματοποιείται με την ενεργοποίηση του Raspberry Pi μέσω ειδικού διακόπτη τροφοδοσίας .Κατά την διαδικασία εκκίνησης του λειτουργικού ο χρήστης ενημερώνεται ηχητικά με φωνητικό μήνυμα: «Hello Maestro, I am ready to hear you ». Μετά την πάροδο ορισμένων δευτερολέπτων το σύστημα εισέρχεται σε κατάσταση πλήρους λειτουργικότητας και είναι έτοιμο να δεχτεί και να επεξεργαστεί φωνητικές εντολές.

4.1.2 Έλεγχος εξόδων χαμηλής τάσης

Για την επιτυχή αναγνώριση εντολών ,η ελάχιστη απόσταση του χρήστη από το μικρόφωνο δεν θα πρέπει να υπερβαίνει τα 40 cm. Εφόσον πληρείται η προϋπόθεση, καθίσταται δυνατός ο έλεγχος της απενεργοποίησης/ενεργοποίησης μίας εξόδου 220V.Οι φωνητικές εντολές πρέπει να ακολουθούν το εξής πρότυπο:

<Random info> <action > <devices> <models> <Random info>

Enable Actions	Disable Actions	Devices	Models
Turn on	Turn off	Relay	Alpha
Open	Close	Device	Beta
Enable	Disable	System	Both

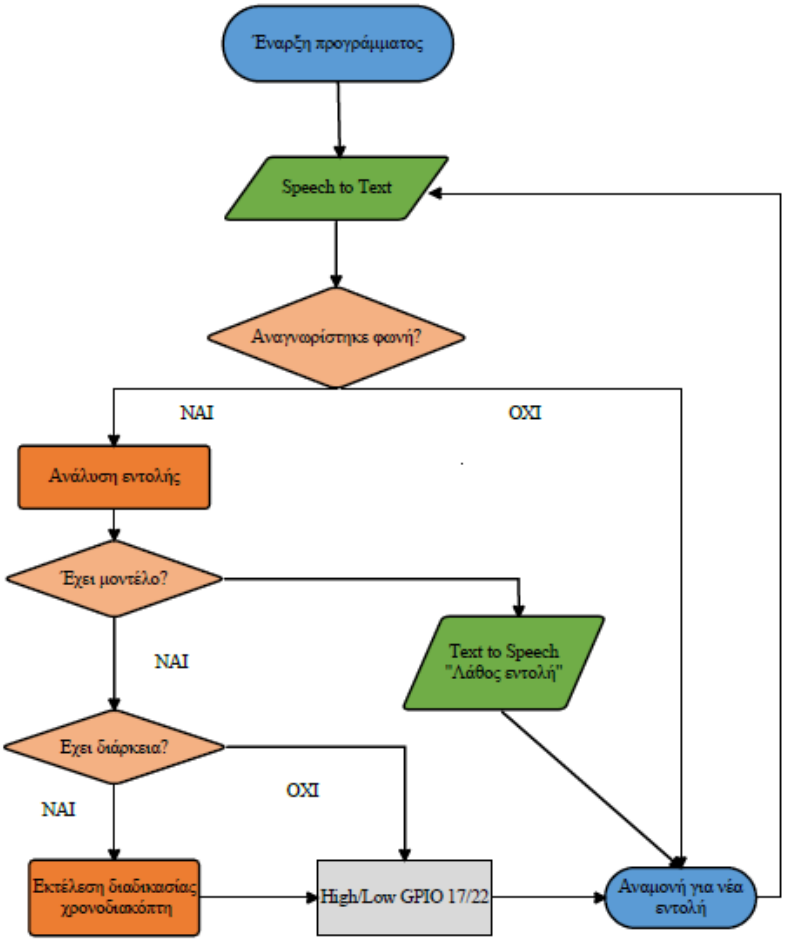
Πίνακας 4.1 Πίνακας φωνητικών ορισμάτων για τον χειρισμό των AC εξόδων

Ο έλεγχος των εξόδων OUT1 και OUT2 μπορεί να επιτευχθεί είτε μεμονωμένα αναφέροντας το αντίστοιχο μοντέλο της συσκευής (Πίνακας 4.1) , είτε ταυτόχρονα ορίζοντας την τιμή Both ως όρισμα του μοντέλου. Οι φωνητικές εντολές μπορούν να περιλαμβάνουν επιπλέον πληροφορίες χωρίς αυστηρή σύνταξη, καθώς ο μηχανισμός φιλτραρίσματος συγκρατεί μόνος τις απαραίτητες τιμές των παραμέτρων.

Για την ενεργοποίηση/απενεργοποίηση μίας εξόδου (Σχήμα 4.1) , αρκεί να δηλωθεί η επιθυμητή ενέργεια, να προσφωνηθεί το σύστημα και να καθοριστεί το μοντέλο. Για παράδειγμα η εντολή «Hello Sir, can you open system Alpha?» ενεργοποιεί το ρελέ της εξόδου OUT1 θέτοντας την συνδεδεμένη συσκευή σε λειτουργία. Αντίστοιχα στην περίπτωση που η έξοδος έχει ήδη ενεργοποιηθεί ο χρήστης μπορεί να την απενεργοποιήσει λέγοντας «Can you turn off Relay Beta?»

Η χρήση της τιμής Both στη μεταβλητή του μοντέλου, δίνει την δυνατότητα ταυτόχρονου ελέγχου και των δύο εξόδων, χωρίς την ανάγκη εκφώνησης δύο ξεχωριστών εντολών. Έτσι για παράδειγμα η φράση <<Hello my friend, I want you to disable both devices >> απενεργοποιεί και τις δύο εξόδους ταυτόχρονα.

Η διαδικασία παραμένει απλή , καθώς υποστηρίζεται από μηχανισμούς περιθωριοποίησης των γραμματικών και των συντακτικών λαθών. Ο χρόνος που διατίθεται στον χρήστη για την εκφώνηση είναι επαρκής, ακόμα και με παύσεις στην ομιλία. Επιπλέον προβλέπεται η δυνατότητα ετεροχρονισμένης εκτέλεσης, ώστε ο χρήστης να μπορεί να προγραμματίσει ενέργειες με λειτουργική καθυστέρηση όπως για παράδειγμα η ενεργοποίηση του Boiler ή του κλιματισμού.



Σχήμα 4.1 Διάγραμμα ροής χειρισμού AC εξόδων

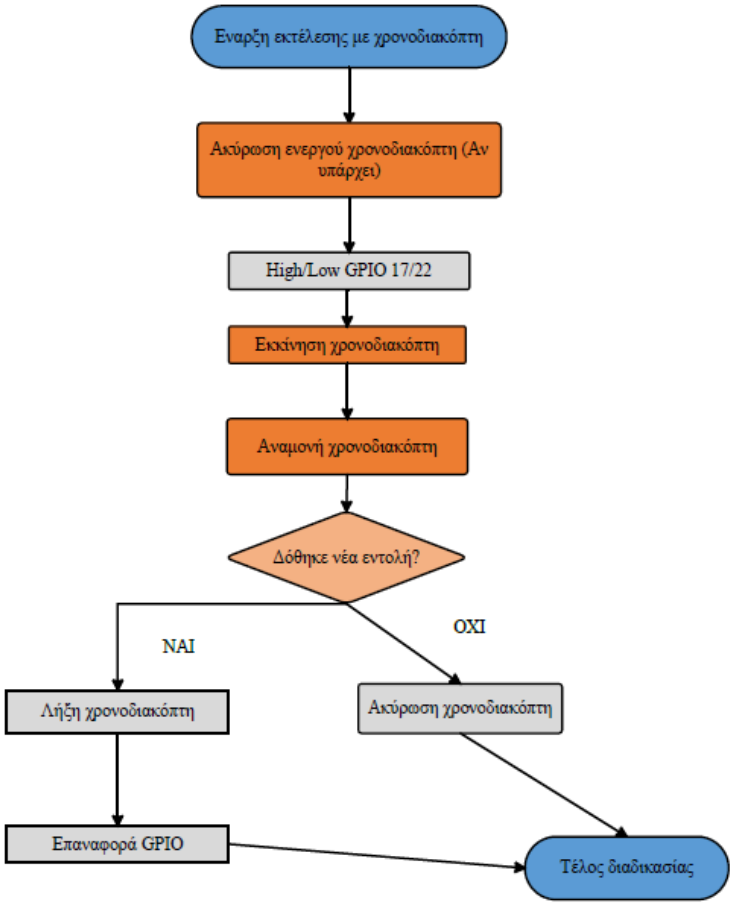
4.1.2.1 Ετεροχρονισμένη εκτέλεση

Για την εκτέλεση ετεροχρονισμένων ενεργειών διατηρείται ίδιο το σώμα της εντολής εκτέλεσης με την προσθήκη δύο στοιχείων μετά την πρόθεση for. Συγκεκριμένα το πρότυπο ετεροχρονισμένης εκτέλεσης έχει την μορφή :

<Random info> <action> <devices> <models> <for> <numbers> <units> <Random info>

Όπως ορίζεται και από τον αντίστοιχο πίνακα ο χρήστης μπορεί να επιλέξει την τιμή της χρονικής διάρκειας εντός του επιτρεπτού εύρους, καθώς και να ορίσει την κατάλληλη μονάδα μέτρησης μεταξύ δευτερολέπτων, λεπτών και ωρών.

Η συμπεριφορά του συστήματος όταν χρησιμοποιούμε χρονοδιακόπτες ακολουθεί την ίδια λογική με εκείνη της διαδικασίας ενεργοποίησης/απενεργοποίησης ,με την διαφορά ότι μετά το πέρας το πέρας της καθορισμένης χρονικής διάρκειας η έξοδος μεταβαίνει στην αντίθετη κατάσταση από αυτή που προηγήθηκε ,ανεξάρτητα από την αρχική της τιμή (Σχήμα 4.2). Για παράδειγμα, όταν η έξοδος OUT1 είναι ενεργή και ο χρήστης εκφωνήσει την εντολή «Close system alpha for three minutes », η έξοδος θα απενεργοποιηθεί για τρία λεπτά και έπειτα θα επανέλθει σε κατάσταση ON. Το ίδιο ακριβώς θα συνέβαινε εάν η εντολή δοθεί όταν το OUT1 είναι ήδη απενεργοποιημένο.



Σχήμα 4.2 Διάγραμμα ροής της διαδικασίας χρονοπρογαμματισμού

Ένα ιδιαίτερο χαρακτηριστικό της λειτουργίας χρονοδιακοπτών είναι η παράλληλη μεταβολή πολλαπλών εξόδων. Με την επιλογή της τιμής both ο χρονοδιακόπτης εφαρμόζεται ταυτόχρονα και στις δύο γραμμές χωρίς η μία να επηρεάζει την συμπεριφορά της άλλης.

Επιπλέον, το σύστημα επιτρέπει την δυνατότητα άμεσης ακύρωσης του ενεργού χρονοδιακόπτη. Αυτή η λειτουργία πραγματοποιείται όταν δοθεί νέα οδηγία για το ίδιο υποσύστημα πρίν την λήξη του τρέχοντος χρονομέτρου. Σε αυτήν την περίπτωση ο υφιστάμενος χρονοδιακόπτης τερματίζεται ακαριαία χωρίς να εκτελεστεί η διαδικασία επαναφοράς, ενώ η καινούργια εντολή εφαρμόζεται άμεσα είτε πρόκειται για άμεση είτε για ετεροχρονισμένη εκτέλεση. Έτσι αποτρέπονται πιθανές συγκρούσεις και διασφαλίζεται οτι υπερισχύει πάντα η πιο πρόσφατη οδηγία του χρήστη.

4.1.3 Έλεγχος RGB φωτισμού

Εκτός από την διαχείριση των δύο εξόδων χαμηλής τάσης, το υλικό του συστήματος υποστηρίζει και τον έλεγχο συσκευών που λειτουργούν με διαμόρφωση εύρους παλμών όπως οι ταινίες LED RGB.

Σε αυτήν την περίπτωση η δομή της φωνητικής εντολής διαφοροποιείται σε σχέση με τις μεταβλητές που χρησιμοποιούνται για τον έλεγχο των ρελέ. Συγκεκριμένα το πρότυπο που ακολουθείται είναι το εξής:

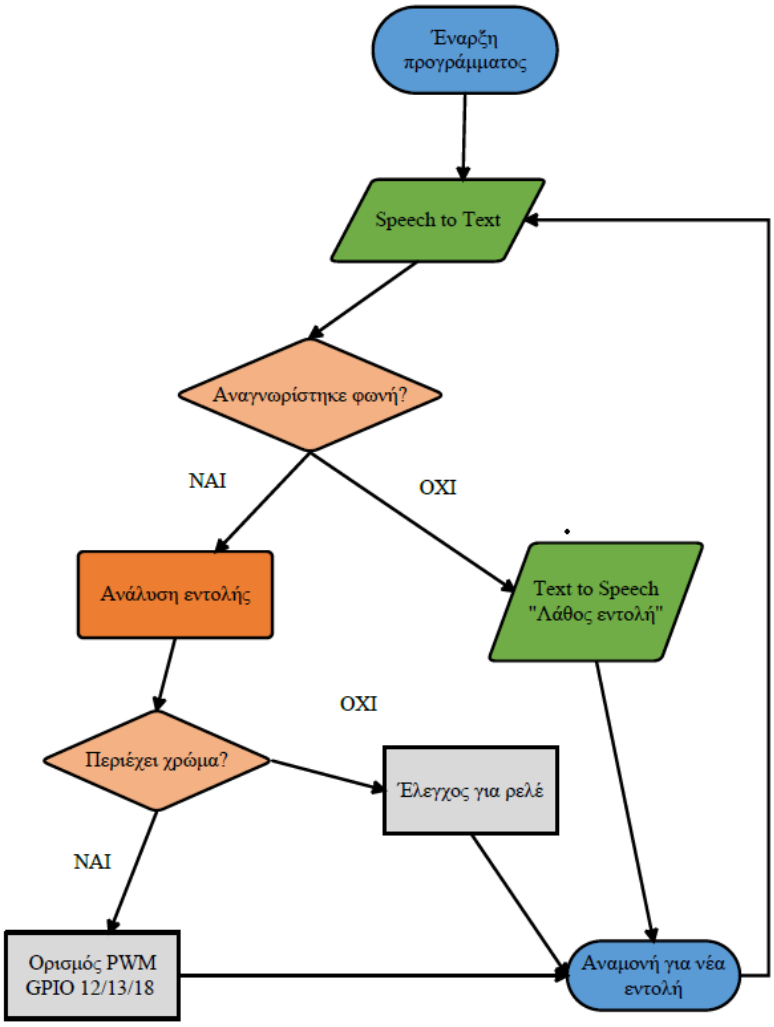
<Random info> <trigger word> <color> <Random info>

Trigger Words		
make	set	change
black		
brown		
red		
orange		
gold		
yellow		
green		
cyan		
blue		
purple		
magenta		
pink		
gray		
silver		
white		

Πίνακας 4.2 Πίνακας χρωματικών ορισμάτων

Η διαδικασία χειρισμού της ταινίας LED είναι ιδιαίτερα απλή, ο χρήστης χρειάζεται μόνο να αναφέρει μία λέξη έναυσης και το επιθυμητό χρώμα (Σχήμα 4.3). Όπως και στις προηγούμενες λειτουργίες η σύνταξη της εκφώνησης δεν περιορίζει το σύστημα καθώς εφαρμόζονται μηχανισμοί Keyword Spotting και Rule-based Parsing. Ωστόσο στην συγκεκριμένη λειτουργία δεν υποστηρίζεται ετεροχρονισμένη εκτέλεση.

Κατά την έναρξη του λογισμικού MVF η ταινία LED παραμένει ανενεργή, αντιστοιχώντας στο χρώμα “black”. Ο χρήστης μπορεί να επιλέξει τον επιθυμητό φωτισμό μέσα από μία παλέτα δεκαπέντε διαθέσιμων χρωμάτων Για παράδειγμα η εντολή «Hello Maestro, make it blue» έχει ως αποτέλεσμα την ενεργοποίηση των μπλέ LED ,ενώ η εντολή «Change color to white» φωτίζει την ταινία με λευκό χρώμα. Αντίστοιχα, για την απενεργοποίηση του φωτισμού, ο χρήστης μπορεί να εκφωνήσει την φράση «Set room to black».



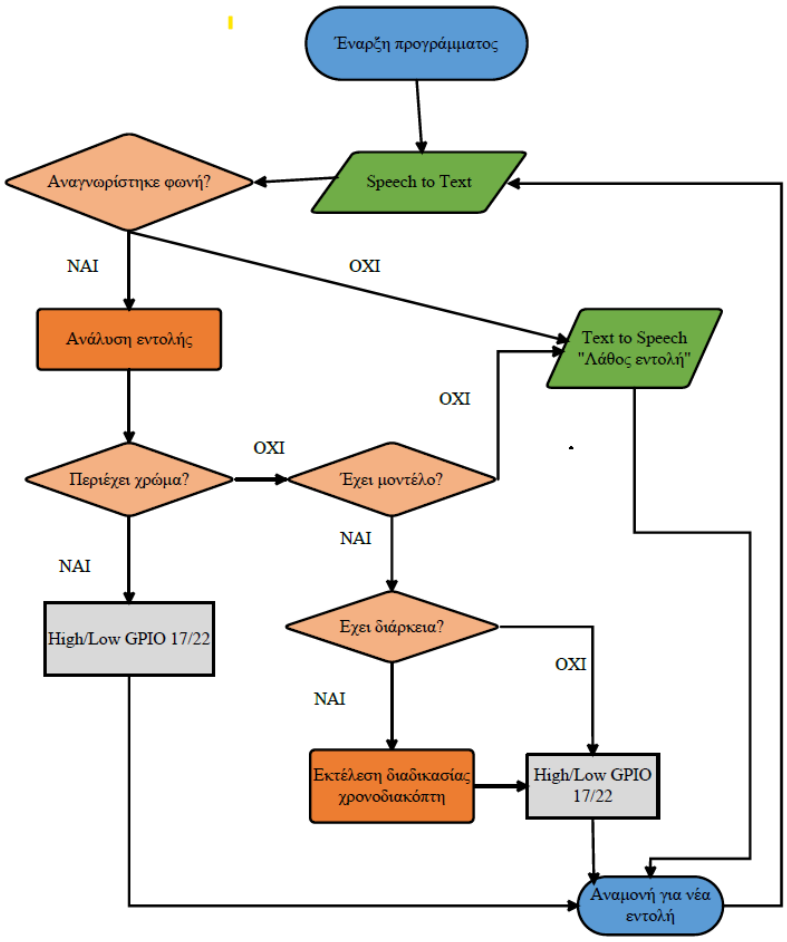
Σχήμα 4.3 Διάγραμμα ροής χειρισμού PWM εξόδου

4.1.4 Γενική λειτουργία

Όπως περιεγράφηκε στις προηγούμενες υποενότητες [4.1] το σύστημα καθίσταται πλήρως λειτουργικό αμέσως μετά την εκφώνηση του μηνύματος καλωσορίσματος. Μετά από κάθε λήψη και αναγνώριση φωνητικής εντολής το λογισμικό παρέχει στον χρήστη κατάλληλα μηνύματα ανατροφοδότησης σχετικά με το αποτέλεσμα της εκτέλεσης. Εάν η εντολή δεν περιλαμβάνει τα απαραίτητα ορίσματα, ζητείται από τον χρήστη να την επαναλάβει, αντίθετα εάν τα ορίσματα είναι επαρκή το λογισμικό παράγει το αντίστοιχο μήνυμα επιβεβαίωσης για την ορθή εκτέλεση.

Σε περιπτώσεις όπου μία εντολή περιλαμβάνει χρονοδιακόπτη ο χρήστης ενημερώνεται σε δύο χρόνους, αρχικά για την εκτέλεση της ενέργειας και ακολούθως για την επαναφορά της εξόδου στην αντίθετη λειτουργία (Σχήμα 4.4).

Συνολικά, το σύστημα παρουσιάζει υψηλή ακρίβεια και αποτελεσματικότητα στην αναγνώριση φωνητικών εντολών για τον έλεγχο των εξόδων, παρέχοντας ταυτόχρονα συνεχή ανατροφοδότηση στον χρήστη. Επιπλέον σημαντικό πλεονέκτημα της εφαρμογής αποτελεί η ανεκτικότητα σε συντακτικές αποκλίσεις και χρονικές παύσεις κατά την εκφώνηση των εντολών.



Σχήμα 4.4 Διάγραμμα ροής κύκλου λειτουργίας του MVF

4.2 Σχεδίαση και υλοποίηση

Σε αυτήν την ενότητα παρουσιάζεται λεπτομερής ανάλυση των διαδικασιών σχεδίασης και ανάπτυξης που αφορούν τόσο το λειτουργικό σύστημα του MVF όσο και του υλικό μέρος της αναπτυξιακής πλακέτας Maestro VF.

4.2.1 Ανάλυση MVF software

Σε αυτήν την υποενότητα αναλύεται το περιεχόμενου του πηγαίο κώδικα του MVF καθώς και οι επιμέρους δομικές και λειτουργικές του ιδιότητες.

Η δομή του κώδικα απαρτίζεται από τις εξής λειτουργίες:

- Φωνητική αλληλεπίδραση μέσω ενσωματωμένων API (TTS & STT), με σκοπό την εισαγωγή δεδομένων από τον χρήστη και την παροχή ανατροφοδότησης σχετικά με το αποτέλεσμα της εκτέλεσης.
- Ανάλυση των ληφθέντων δεδομένων μέσω προσαρμοσμένων φίλτρων, ουτοσώστε να κατηγοριοποιηθεί αρχικά η έξοδος εκτέλεσης και στην συνέχεια να προσδιορισθεί με ακρίβεια το χρονικό πλαίσιο και το είδος την ενέργειας.
- Υποστήριξη ετεροχρονισμένης εκτέλεσης μέσω χρονοδιακοπτών, με δυνατότητα ακύρωσης είτε κατά την εκτέλεσης είτε κατά την φάση αναμονής της διεργασίας.
- Έλεγχο της GPIO ακιδοσειράς του Raspberry Pi για την οδήγηση των ρελέ στην αναπτυξιακή πλακέτα Maestro VF, καθώς και παραγωγή σημάτων PWM για τον χειρισμό της RGB συσκευής.

```
from elevenlabs.client import ElevenLabs            #Χρήση ElevenLabs API για TTS
from difflib import get_close_matches               #Για εύρεση λέξεων με παρόμοια ορθογραφία
from elevenlabs import stream , VoiceSettings       #Για εκτέλεση ανατροφοδότηση σε πραγματικό χρόνο
import threading                                    #Για εκτέλεση χρονοδιακοπτών
from deepgram import (
…….
   LiveTranscriptionEvents,                         #Εντοπισμός ομιλίας
   …….
)
import pigpio                                       #Για τον έλεγχο PWM εξόδων
import RPi.GPIO as GPIO                             #Για τον έλεγχο ψηφιακών εξόδων
```

Ο κώδικας σε πρώτη φάση περιλαμβάνει όλες τις απαραίτητες αρχικοποιήσεις ουτοσώστε να εξασφαλίσει το λειτουργικό σύστημα την εσωτερική του επικοινωνία με τα προσαρτημένα στις GPIO θύρες υλικά και την εξωτερική επικοινωνία του με τις πλατφόρμες των APIs .

Στην συνέχεια συνδέεται με τις πλατφόρμες του ElevenLabs για την δημιουργία streaming (εκτέλεσης σε πραγματικό χρόνο) ανατροφοδότησης όπου ο χρήστης ενημερώνεται άμεσα μετά την εκτέλεση οποιασδήποτε ενέργειας. Η ενσωμάτωση βιβλιοθηκών του Deepgram επιτρέπει την αναγνώριση φωνής σε επίσης πραγματικό χρόνο, με ρυθμίσεις που επιτρέπουν στον κώδικα να χειριστεί γεγονότα μόνο σε διαστήματα όπου ο χρήστης ξεκινά και τελειώνει την ομιλία του.

Για να είναι επιτυχής η εκτέλεση είναι απαραίτητη η αρχικοποίηση βιβλιοθηκών που αφορούν το hardware, η διαδικασία αυτή ολοκληρώνεται ξεχωριστά σε δύο σκέλη όπου το πρώτο αναφέρεται στις θύρες των PWM pins για τον έλεγχο του συστήματος RGB ενώ το δεύτερο αναφέρεται στις ψηφιακές εξόδους για τον έλεγχο ρελέ.

Η εκτέλεση των χρονοδιακοπτών αποτελεί βασική υπηρεσία του λειτουργικού συστήματος ώστε να είναι εφικτή η διαχείριση διεργασιών που απαιτούν ετεροχρονισμένη εκτέλεση. Για τον έλεγχο των εξόδων μέσω ρελέ ενσωματώθηκε μία βιβλιοθήκη που υποστηρίζει την δημιουργία χρονομέτρων και γεγονότων για την εκτέλεση παράλληλων ενεργειών.

```
Client=ElevenLabs(api_key=”API KEY”)                    #Σύνδεση με ElevenLabs μέσω προσωπικού κλειδιού
def text_to_speech(input):
    audio_stream = client.generate(
     text=input,
     stream=True,
     model='eleven_multilingual_v2',                    #Χρήση πολυγλωσσικού μοντέλου
     voice="TX3LPaxmHKxFdv7VOQHJ",
  )
  stream(audio_stream)
```

Στην συνέχεια ορίζονται οι σύνθετες-συναρτήσεις με τα ενσωματωμένα σε αυτές APIs, οι οποίες είναι αρμόδιες για την αλληλεπίδραση μεταξύ του χρήστη και του συστήματος.

Η μεταβλητή client δημιουργεί ένα αντικείμενο σύνδεσης με το API μέσω προσωπικού κλειδιού. Η συνάρτηση text_to_speech αξιοποιεί το API, ορίζοντας το μοντέλο φωνής και γλώσσας για την αναπαραγωγή σε πραγματικό χρόνο του κειμένου που έχει τοποθετηθεί στην μεταβλητή input.

```
def Speech_to_Text():
  is_finals = []                                        #Λίστα για τις τελικές φράσεις που αναγνωρίστηκαν
```

```
   final_text = ""                                                    #Το τελικό κείμενο που θα επιστραφεί
   finish_event = threading.Event()                                   #Δημιουργία "σημαίας" για τον τερματισμό της ομιλίας
try:
    deep gram: DeepgramClient = DeepgramClient("API KEY")   #Σύνδεση με Deepgram μέσω προσωπικού κλειδιού
    dg_connection = deepgram.listen.websocket.v("1")                  #Δημιουργεί Websocket σύνδεση
    ……                                                                #Επιμέρους ρυθμίσεις του API
     return final_text                                                #Επιστροφή του  μεταφρασμένου φωνητικά text
except Exception as e:
          print(f"Could not open socket: {e}")                       #Εμφανίζει το μήνυμα σφάλματος  σύνδεσης
           return None
```

Η συνάρτηση Speech_to_text επιστρέφει μέσω του Deepgram το κείμενο που αναγνωριστικέ μόνο κατά το διάστημα ομιλίας του χρήστη. Η επικοινωνία γίνεται μέσω WebSocket ώστε να πραγματοποιηθεί ζωντανή σύνδεση για την μεταφορά ήχου και την λήψη του final_text .

Λόγω της εξάρτησης από εξωτερική υπηρεσία είναι πιθανές οι αποτυχίες σύνδεσης με την πλατφόρμα. Για τον λόγο αυτό ενσωματώνεται το μπλοκ try..except που εξασφαλίζει την ομαλή λειτουργία και εκτέλεση του κώδικα.

```
enable_actions = ["turn on", "open", "enable"]                  #Λίστα εντολών ενεργοποίησης Ρελέ
disable_actions = ["turn off", "close", "disable"]              #Λίστα εντολών ενεργοποίησης Ρελέ
devices = ["relay", "device", "system"]                         #Λίστα ονομαστικών κλήσεων  Ρελέ
models = ["alpha", "beta","both"]                               #Λίστα κατηγοριοποίησης εξόδων
```

Στην συνέχεια ορίζονται τέσσερις βασικές λίστες που χρησιμοποιούνται στο στάδιο της ανάλυσης των εντολών. Οι enable_actions και disable_actions αφορούν τις ενέργειες ενεργοποίησης και απενεργοποίησης των ψηφιακών εξόδων για τον έλεγχο των ρελέ. Η λίστα devices ορίζει εναλλακτικούς όρους κλήσης, η models κατηγοριοποιεί τις εξόδους σε μοντέλα Alpha και Beta, ενώ το όρισμα both χρησιμοποιείται για ταυτόχρονη εκτέλεση μιας ενέργειας.

```
color_rgb = {                                                   #Λεξικό ορισμάτων για  χρώματα  στα pwm pins
   "red": (255, 0, 0),
   …….
   "silver": (192, 192, 192)
}
```

Στο συγκεκριμένο τμήμα κώδικα ορίζεται το λεξικό το οποίο περιλαμβάνει τα χρώματα που μπορούν να αποδοθούν στις PWM εξόδους για τον έλεγχο των στοιχείων RGB. Κάθε παράμετρος αντιστοιχίζεται σε μία τριπλέτα τιμών που εκφράζουν το Duty Cycle κάθε καναλιού.

Με αυτόν τον τρόπο η φωνητική εντολή του χρήστη μεταφράζεται σε σαφή αριθμητικά δεδομένα τα οποία μπορούν να χρησιμοποιηθούν άμεσα για τον έλεγχο της φωτεινότητας.

```
number_words = {                                    #Λεξικό χρονικών ορισμάτων για χρονοδιακόπτες
   "one": 1, "two": 2, "three": 3, "four": 4, "five": 5,
    ……..
   "fifty": 50, "fifty five": 55, "sixty": 60
}

time_units = {                                      #Λεξικό ορισμάτων χρονικών μονάδων για χρ/τες
   "second": 1, "seconds": 1,
   "minute": 60, "minutes": 60,
   "hour": 3600, "hours": 3600
}
active_timers = {                                   #Λεξικό για αποθήκευση χρο/των στα ορίσματα των ρελέ
   "alpha": None,
   "beta": None
}
```

Κάτι παρόμοιο συμβαίνει όταν ο χρήστης καλεί έναν χρονοδιακόπτη να εκτελεστεί, τότε αντίστοιχα μετατρέπονται οι λεκτικοί αριθμοί σε ακέραιες τιμές. Η time_units αντιστοιχεί τις μονάδες χρόνου σε δευτερόλεπτα ουτοσώστε όλες οι μετρήσεις να ενοποιούνται για τις ρουτίνες χρονισμού. Για την αποθήκευση και παρακολούθηση των ενεργών χρονοδιακοπτών χρησιμοποιείται το λεξικό active_timers.

Έτσι όταν ο χρήστης καλεί έναν χρονοδιακόπτη οι τιμές μετατρέπονται αυτόματα και η κατάσταση της αντίστοιχης εξόδου ενημερώνεται άμεσα για την διασφάλιση της σωστής εκτέλεσης ενέργειας.

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)                            #Ορισμός ρελέ Alpha στο pin 17
GPIO.setup(22, GPIO.OUT)                            #Ορισμός ρελέ  Beta  στο pin 22

GPIO.output(17, GPIO.LOW)                           #Αρχικοποίηση ψηφιακών εξόδων σε θέση low
GPIO.output(22, GPIO.LOW)
output_status = {"Alpha": False, "Beta": False}     #Μεταβλητές παρακολούθησης εξόδων

pi.set_PWM_dutycycle(12, 0)  # Red                  #Ορισμός και μηδενισμός εξόδων  pwm
pi.set_PWM_dutycycle(13, 0)  # Green
pi.set_PWM_dutycycle(18, 0)  # Blue
```

Σε αυτό το σημείο του κώδικα δηλώνονται και ορίζονται αρχικές τιμές για τα pins που αντιστοιχούν στις εξόδους του Raspberry Pi. Ζωτικής σημασίας είναι το σύστημα να εκκινεί με τις εξόδους απενεργοποιημένες και σε κατάσταση αναμονής όπως και συμβαίνει σε κάθε συμβατικό κύκλωμα ώστε να αποφεύγεται η ακούσια ενεργοποίηση συσκευών.

Σε πρώτη φάση επιλέγεται το σύστημα αρίθμησης BCM το οποίο βασίζεται στην εσωτερική αρίθμηση των pins και όχι στην φυσική τους διάταξη. Οι ψηφιακές έξοδοι έχουν οριστεί ως Alpha και Beta, αντιστοιχώντας στα GPIO pins 17 και 22. Οι τιμές τους μηδενίζονται κατά την εκκίνηση και η κατάσταση τους παρακολουθείται μέσω του λεξικού output_status.

Παρόμοια λογική ακολουθείται και για τις PWM εξόδους οι οποίες σχετίζονται με την RGB λίστα χρωμάτων. Τα κανάλια Red, Green και Blue αντιστοιχούν στα GPIO pins 12, 13 και 18 ενώ αντίστοιχα οι έξοδοι αρχικά μηδενίζονται ώστε το σύστημα φωτισμού να ξεκινά σε ανενεργή κατάσταση.

```
def set_color(color_name):
    rgb = color_rgb.get(color_name.lower())
    if rgb:
        r, g, b = rgb
        pi.set_PWM_dutycycle(12, int((r / 255) * 255))  # R                    # Ρύθμιση καναλιών PWM
        pi.set_PWM_dutycycle(13, int((g / 255) * 255))  # G
        pi.set_PWM_dutycycle(18, int((b / 255) * 255))  # B
        print(f"Color set to {color_name}")
    else:
        print("Unknown color")                                                  #Αν το χρώμα δεν υπάρχει στο dictionary
```

Η συνάρτηση set_color χρησιμοποιείται για την εκτέλεση εντολών που αφορούν την έξοδο RGB. Αφού δεχθεί ως όρισμα το όνομα του χρώματος αναζητά τις αντίστοιχες τιμές στο λεξικό color_rgb. Αν το χρώμα βρεθεί οι αριθμητικές τιμές μετατρέπονται σε Duty Cycle και αποστέλλονται στα κανάλια PWM για την απόδοση τους στην ταινία LED. Σε περίπτωση που το όρισμα δεν εντοπιστεί στο λεξικό εμφανίζεται κατάλληλο μήνυμα σφάλματος στην κονσόλα.

```
def detect_color_command(text):
    pi.set_PWM_dutycycle(12, 0)                          #Αρχικοποίηση καναλιών PWM
    pi.set_PWM_dutycycle(13, 0)
    pi.set_PWM_dutycycle(18, 0)
    text = text.lower()
    trigger_words = ["set", "change", "make"]            #Λίστα λέξεων που ενεργοποιούν την αναγνώριση χρώματος
    if any(trigger in text for trigger in trigger_words):
        for color in color_rgb.keys():
            if color in text:
                return color
    return None                                          #Αν δεν βρεθεί κανένα χρώμα, επιστρέφει None
```

Η συνάρτηση detect_color_command αναλαμβάνει την ανίχνευση ορισμάτων που σχετίζονται με την αλλαγή χρώματος.

Αρχικά γίνεται καθαρισμός των PWM καναλιών για να αποφευχθούν ανεπιθύμητες χρωματικές μίξεις. Έπειτα το κείμενο εισόδου μετατρέπεται σε πεζούς χαρακτήρες ώστε να απλοποιηθεί η διαδικασία αναζήτησης Για την ενεργοποίηση της εντολής χρησιμοποιούνται λέξεις εναύσματα, η αναγνώριση μίας τέτοιας λέξης επιτρέπει στην συνάρτηση να ελέγξει αν περιλαμβάνεται κάποιο από τα διαθέσιμα χρώματα στο λεξικό color_rgb.

```
def model(recognized_text):                                    #Ορισμός της  model για parsing μεταβλητών
   command_action = None                                       #Αρχικοποίηση μεταβλητών
   object = None
   model_output = None

   text = recognized_text.lower()                              #Μετατροπή text σε lower case

   for action in sorted(enable_actions + disable_actions, key=lambda x: -len(x)):   //Αναζήτηση μεταβλητών στο text
      if action in text:
         ……

   for word in text.split():
         ……

   return command_action, object, model_output                 #Επιστρέφει μεταβλητές που εντοπίστηκαν στο text
```

Η συνάρτηση model αναλαμβάνει να αναλύσει το κείμενο που προέρχεται από την φωνητική εντολή του χρήστη. Στόχος της είναι να εξάγει τρείς βασικές μεταβλητές:command_action, object, model_output. Αρχικά όλες οι μεταβλητές ορίζονται ως None ώστε να είναι σαφές σε περίπτωση σφάλματος ότι χρήστης δεν παρείχε όλα τα απαραίτητα στοιχεία στο σύστημα.

Η αναζήτηση ξεκινάει με τον έλεγχο σύνθετων εντολών που αφορούν την ενέργεια όπως “turn on” ή “turn off”, στην συνέχεια ο κώδικας εξετάζει το υπόλοιπο κείμενο για λέξεις που ταιριάζουν με αντικείμενα η μοντέλα συσκευών. Με αυτόν τον τρόπο γίνεται διαχωρισμός και αντιστοίχιση της εντολής σε συγκεκριμένα ορίσματα.

Στο τέλος η συνάρτηση επιστρέφει τις μεταβλητές που αναγνωρίστηκαν. Αν κάποια από αυτές λείπει η εντολή θεωρείται μη έγκυρη και η εκτέλεση δεν προχωράει. Έτσι διασφαλίζεται ότι κάθε έξοδος μπορεί να τροποποιηθεί μόνο όταν η εντολή είναι σωστά δομημένη.

```
def extract_total_seconds(text):
   text = text.lower()
   total=0                                          #Αρχικοποίηση χρονοδιακόπτη
   …….
     total = value * multiplier                     #Υπολογισμός της συνολικής διάρκειας σε δευτερόλεπτα
   …….
   …….
      Return total                                  #Επιστροφή της συνολικής διάρκειας σε δευτερόλεπτα
```

Όσο αναφορά την διαχείριση των χρονοδιακοπτών χρησιμοποιείται η συνάρτηση extract_total_seconds για την μετάφραση των χρονικών εκφράσεων από κείμενο σε δευτερόλεπτα. Αρχικά το κείμενο μετατρέπεται σε πεζούς χαρακτήρες και μέσα από λεξικά εντοπίζονται οι αριθμητικές τιμές και οι μονάδες μέτρησης.

Στην συνέχεια η extract total μετατρέπει την μονάδα σε δευτερόλεπτα και την πολλαπλασιάζει με την τιμή value. Το τελικό αποτέλεσμα είναι ο ακριβής χρόνος που λαμβάνει ο χρονοδιακόπτης.

```
def timer_execution(cur_state, model, action, duration):          #Εκτελεί ενέργειες μέσω χρονοδιακόπτη
   pin = 17 if model == "alpha" else 22                           #Ορίζει το GPIO pin εξόδου

   if action in enable_actions:                                   #Ελέγχει αν η εντολή είναι για ενεργοποίηση
      if not cur_state:
         …
         GPIO.output(pin, GPIO.HIGH)                              #Ενεργοποιεί την έξοδο που ζητήθηκε
         …
      else:                                                       #Δεν μεταβάλλεται η κατάσταση εξόδου
         …
     t = threading.Timer(duration, lambda: restore_state(model, pin, False))      //μπαίνει στην ουρά η restore state
      t.start()                                                   #ξεκινάει ο χρονοδιακόπτης
      active_timers[model]=t
      return False                                                #Αποθηκεύεται η νέα κατάσταση εξόδου

   elif action in disable_actions:                 :              #Ελέγχει αν η εντολή είναι για απενεργοποίησή
      if cur_state:
         GPIO.output(pin, GPIO.LOW)                               #Ενεργοποιεί την έξοδο που ζητήθηκε
         ….
      else:                                                       #Δεν μεταβάλλεται η κατάσταση εξόδου
         ….
      return True                                                 #Αποθηκεύεται η νέα κατάσταση εξόδου
```

Η συνάρτηση timer-execution διαχειρίζεται την χρονική συμπεριφορά των ψηφιακών εξόδων με βάση την εντολή και την τρέχουσα κατάσταση τους. Ανάλογα με το αν ζητείται ενεργοποίηση ή απενεργοποίηση η συνάρτηση ελέγχει αν απαιτείται αλλαγή της τρέχουσας κατάστασης. Αν η εντολή ταυτίζεται με την ήδη υπάρχουσα κατάσταση η έξοδος παραμένει σταθερή χωρίς μεταβολή.
Τα σενάρια τα οποία καλείται να χειριστεί η συνάρτηση είναι τέσσερα:

- ON→OFF
- OFF→ON
- ON→ON
- OFF→OFF

Όταν απαιτείται αλλαγή το αντίστοιχο pin μεταβάλλεται σε HIGH ή LOW για διάρκεια ίση με την τιμή του χρονοδιακόπτη. Μετά την λήξη του χρόνου η έξοδος επαναφέρεται στην αντίθετη κατάσταση από την εκτελεσμένη ενέργεια.

```
def restore_state(model, pin, state):                          #Αλλάζει την κατάσταση εξόδου
   GPIO.output(pin, GPIO.HIGH if state else GPIO.LOW)          #Ορίζει το pin εξόδου σε HIGH ή LOW
   output_status[model] = state                                #Ενημερώνει την κατάσταση εξόδου στην μνήμη
```

```
    print(f"{model} restored to {'ON' if state else 'OFF'}\n")              #Ενημέρωση μέσω console
    text_to_speech(f"{model} restored to {'ON' if state else 'OFF'}")       #Ενημέρωση χρήστη
    active_timers[model] = None                                             #Μηδενίζει τον ενεργό timer
```

Η συνάρτηση restore_state εκτελείται αυτόματα με την λήξη του χρονοδιακόπτη που ξεκίνησε στην συνάρτηση timer_execution. Ο ρόλος της είναι να επαναφέρει την έξοδο σε προηγούμενη η αντίθετη κατάσταση. Παράλληλα ενημερώνει το λεξικό output_status,μηδενίζει τον ενεργό χρονοδιακόπτη και παρέχει φωνητική ανατροφοδότηση στον χρήστη σχετικά με το περιεχόμενο της αλλαγής.

```
if __name__ == "__main__":
   output_status = {"alpha": False, "beta": False}                          #Αρχικοποίηση λεξικού για την κατάσταση των εξόδων

print("Hello Maestro, I am ready to hear you. \n")
text_to_speech("Hello Maestro, I am ready to hear you.")                    #Αναπαραγωγή ηχητικού μηνύματος εκκίνησης
```

Για να εκτελεστούν οι βασικές λειτουργίες του συστήματος δημιουργήθηκε η συνάρτηση main..Η main αποτελεί τον κορμό του προγράμματος και καλείται με την εκκίνηση του αρχείου Maestro_mvf.py. Κατά την εκκίνηση οι έξοδοι τίθενται σε κατάσταση απενεργοποίησης για λόγους ασφαλείας. Στην συνέχεια το σύστημα καλωσορίζει τον χρήστη με το μήνυμα “Hello Maestro,i am ready to hear you” σηματοδοτώντας οτι είναι έτοιμο να δεχθεί εντολές.

Η δημιουργία αυτής της συνάρτησης επιτρέπει την ενσωμάτωση ενός βρόχου συνεχούς εκτέλεσης στον οποίο συντονίζονται οι διεργασίες εισόδου και εξόδου με φυσική ροή καθόλη την διάρκεια λειτουργίας του συστήματος.

```
try:                                #Δημιουργεί μπλόκ κώδικα ώστε ο χρήστης  να μπορεί να διακόψει την λειτουργία του προγ/τος
   while True:                      #Δημιουργεί βρόχο για συνεχή ακρόαση εντολών
      text = Speech_to_Text()       #τοποθετεί το κείμενο  που δόθηκε από το API  κατά την καταγραφή
      if text:                      #ελέγχει αν καταγράφηκε κάποια ομιλία
         print(f"Final Recognized Text: {text}\n")     #Εμφανίζει το text που καταγράφηκε στην κονσόλα
         ……
       else:
                                    #Εκτελέσιμος κώδικας
 except KeyboardInterrupt:          #διακόπτει την ροή του προγράμματος όταν ο χρήστης πατήσει Cntrl+C
       print("\n Shutting down...")
   pi.set_PWM_dutycycle(12, 0)      #Απενεργοποιεί τα GPIO PWM pins
   ……
  Pi.stop()
  GPIO.output(17, GPIO.LOW)         #Απενεργοποιεί τις ψηφιακές εξόδους των ρελέ
```

```
……
```

Η σύνταξη του try…except προστατέυει το πρόγραμμα από βίαιο τερματισμό. Σε περίπτωση διακοπής από τον χρήστη η εντολή except οδηγεί σε ομαλό τερματισμό του κύριο προγράμματος απενεργοποιώντας με ασφάλεια τίς εξόδους GPIO τόσο για τα PWM pins (12, 13, 18) όσο και για τα ψηφιακά pins (17,22).

Ο ατέρμων βρόχος while …True εξασφαλίζει ότι το σύστημα παραμένει συνεχώς ενεργό, κρατώντας το μικρόφωνο σε λειτουργία και το πρόγραμμα σε ετοιμότητα για λήψη νέων φωνητικών εντολών. Με αυτόν τον τρόπο επιτυγχάνεται μία φυσική και αδιάλειπτη ροή εκτέλεσης.

Η είσοδος ήχου καταγράφεται μέσω της Speech_to_ text και τοποθετείται στην μεταβλητή text. Στην συνέχεια το λειτουργικό συνεχίζει την ανάλυση για τον εντοπισμό των λέξεων και φράσεων διέγερσης που ενεργοποιούν τις αντίστοιχες εξόδους.

```
  command_action, object, model_output = model(text)
        color_name = detect_color_command(text)
        duration = extract_total_seconds(text)
        print(f"Parsed  action: {command_action}\ndevice: {object}\nmodel: {model_output}\ncolor:
{color_name}\nduration: {duration}\n")
        if color_name:
           set_color(color_name)
           text_to_speech(f"Color set to {color_name}")
           continue
```

Το λειτουργικό σύστημα αναλύει την καταγεγραμμένη φωνητική εντολή σε βασικές παραμέτρους. Η συνάρτηση model διαχωρίζει την πρόταση σε τρία στοιχεία: το είδος της ενέργειας, την οντότητα και το μοντέλου εξόδου. Παράλληλα η συνάρτηση detect_color ελέγχει αν η εντολή περιλαμβάνει χρωματική παράμετρο για την έξοδο RGB, ενώ η extract_total_seconds μετατρέπει ενδεχόμενη χρονική αναφορά σε κατάλληλη τιμή για χρήση από τους χρονοδιακόπτες.

```
    if model_output == "both":                          #Ελεγχος αν η εντολή αφορά και τα δύο συστήματα
       if command_action in enable_actions:
          GPIO.output(17, GPIO.HIGH)
          output_status["alpha"] = True
          …….
          print("Both systems are now ON\n")
          text_to_speech("Alpha and Beta are now ON")
       elif command_action in disable_actions:
         ……..
       for key in ["alpha", "beta"]:                    #Ακύρωση τυχόν ενεργών timers και για τα δύο συστήματα
          if active_timers.get(key):
             active_timers[key].cancel()
```

```
                active_timers[key] = None
        else:
            if active_timers.get(model_output):                    #Αν η εντολή αφορά μόνο ένα από τα δύο συστήματα
                active_timers[model_output].cancel()
                active_timers[model_output] = None
```

Προκειμένου ο χρήστης να μπορεί να δώσει φωνητική οδηγία για παράλληλη εκτέλεση εντολών δύο και στις δύο ψηφιακές εξόδους ορίστηκε η λέξη-έναυσμα both. Όταν η μεταβλητή model_output δεχθεί αυτήν την τιμή το σύστημα μπορεί να είτε απενεργοποιήσει είτε να ενεργοποιήσει ταυτόχρονα τα GPIO pins 17 και 22 που αντιστοιχούν στις εξόδους Alpha και Beta.
Παράλληλα ελέγχεται η δομή active_timers ώστε να ακυρώνονται τυχόν ενεργοί χρονοδιακόπτες πριν την εκτέλεση της νέας εντολής. Με αυτόν τον τρόπο μπορούν να αποφευχθούν σφάλματα ή ανεπιθύμητες ενέργειες που θα μπορούσαν να προκύψουν από παλαιότερες φωνητικές εντολές

```
if duration:
  output_status[model_output] = timer_execution(  output_status[model_output], model_output, command_action,duration)
```

Σε αυτό το τμήμα ο κώδικας ελέγχει αν η φωνητική εντολή που βρίσκεται στην μεταβλητή text περιέχει χρονική διάρκεια. Αν υπάρχει καλείται η συνάρτηση timer_execution ,η οποία εκτελεί την εντολή με καθορισμένο χρονοδιακόπτη, με την ολοκλήρωση της η έξοδος επαναφέρεται στην προηγούμενη κατάσταση.

```
else:
    pin = 17 if model_output == "alpha" else 22 if model_output == "beta" else None    #Επιλογή GPIO pin βασει μοντέλου
    if pin is not None:
        if command_action in enable_actions:
            if output_status[model_output]:
                print(f"System {model_output} is already ON!\n")
                text_to_speech(f"System {model_output} is already ON")
             else:
                GPIO.output(pin, GPIO.HIGH)                                          #Ενεργοποίηση ρελέ
                output_status[model_output] = True
                print(f"{model_output} is now ON\n")
                text_to_speech(f"{model_output} is now ON")

          elif command_action in disable_actions:
             .....                                                                    #Απενεργοποίηση ρελέ
     else:
        print(" Unknown system model!\n")
```

Στο τελικό μέρος της ανάλυσης παρουσιάζεται το σκέλος που αφορά την εκτέλεση ενεργειών που αφορούν τις ψηφιακές εξόδους σε πραγματικό χρόνο, όταν δηλαδή δεν έχουν εντοπιστεί κατά τον έλεγχο της if χρονικές παράμετροι.

Αρχικά το σύστημα ελέγχει αν η επιλεγμένη έξοδος είναι ήδη στην κατάσταση-στόχο που ζητήθηκε, σε αυτήν την περίπτωση ο χρήστης ενημερώνεται φωνητικά πώς η έξοδος είναι ήδη απενεργοποιημένη/ενεργοποιημένη.

Σε διαφορετική περίπτωση που απαιτείται αλλαγή κατάστασης η εντολή εκτελείται ομαλά αλλάζοντας την κατάσταση των αντίστοιχων GPIO pins σε HIGH ή LOW ,ενώ παράλληλα ο χρήστης λαμβάνει φωνητική επιβεβαίωση μέσω TTS.

4.2.2 Σχεδιασμός Maestro VF στο Altium Designer

Σε αυτήν την ενότητα παρουσιάζεται αναλυτικά η μεθοδολογία και η διαδικασία σχεδιασμού που ακολουθήθηκε στο πλαίσιο βιομηχανικών προτύπων για την δημιουργία της πλακέτας Maestro VF.Η υλοποίηση του έργου πραγματοποιήθηκε από την δημιουργία του ηλεκτρονικού σχεδίου μέχρι και τον τελικό σχεδιασμού PCB από το εξελιγμένο και επαγγελματικό εργαλείο CAD Altium designer [32].

Η διαδικασία σχεδιασμού του Maestro VF διαχωρίζεται σε τρείς βασικές φάσεις:

- Πρότυπη σχεδίαση και ορισμός υλικών απαιτήσεων.
- Δημιουργία σχηματικού διαγράμματος.
- Ανάπτυξη τελικού PCB.

4.2.2.1 Φάση πρότυπης σχεδίασης και ορισμός υλικών απαιτήσεων

Η πρώτη φάση περιλαμβάνει τον ορισμό των απαιτήσεων στις οποίες καλείται να ανταποκριθεί το σύστημα. Βασικό μέλημα του πρότυπου σχεδιασμού είναι η ανάπτυξη ενός αξιόπιστου τυπωμένου κυκλώματος ικανού να ανακατευθύνει και να εκτελεί ενέργειες απενεργοποίησης και ενεργοποίησης. Δεδομένου ότι πρόκειται για ένα έξυπνο σύστημα απώτερος στόχος της εφαρμογής είναι η πρότυπη πλακέτα να ελέγχει με ασφάλεια και αξιοπιστία τις συνδεδεμένες συσκευές.

Στο αρχικό στάδιο ορίστηκαν οι βασικές προδιαγραφές του έργου οι οποίες είναι :

- Διαστάσεις PCB: 80 mm x 100 mm.
- Τερματικά I/O: Δύο έξοδοι εναλλασσόμενης τάσης (220 V) και μία έξοδος PWM (12 V).

Το Maestro VF ενσωματώνει έξι τερματικά I/O, εκ των οποίων τα τρία διατίθενται προς στον χρήστη για εξωτερική συνδεσμολογία. Οι δύο κλέμμες εξόδου (OUT1 και OUT2) προορίζονται για την σύνδεση φορτίων 220V AC, ενώ παρέχεται και μία υποδοχή 4-pin Dupont για την σύνδεση ταινιών RGB LED.Η τροφοδοσία του συστήματος επιτυγχάνεται μέσω της κλέμμας IN απευθείας από το δίκτυο.

Για την παροχή χαμηλής συνεχούς τάσης (12 V DC) εξετάστηκαν δύο προοπτικές :

- Η εγκατάσταση εξωτερικής πλακέτας μετασχηματισμού.
- Η συναρμολόγηση εσωτερικού κυκλώματος τροφοδοσίας.

Και οι δύο επιλογές μετατρέπουν την εναλλασσόμενη τάση 220 V σε συνεχή 12 V, ωστόσο η εγκατάσταση εξωτερικής πλακέτας μειώνει αρκετά το κόστος, το βάρος και το φυσικό μέγεθος της συσκευής ενώ εξασφαλίζει μεγαλύτερη σταθερότητα στην τελική έξοδο.

Η παροχή 12 V DC αποτελεί βασικό στοιχείο για την λειτουργία του Maestro VF, καθώς τροφοδοτεί τόσο τα κυκλώματα οδήγησης των ρελέ όσο και το υποσύστημα ελέγχου RGB LED. Με αυτόν τρόπο εξασφαλίζεται πώς οι έξοδοι χαμηλής τάσης και οι έξοδοι φωτισμού μπορούν να λειτουργούν αξιόπιστα χωρίς την ανάγκη πρόσθετης εξωτερικής τροφοδοσίας.

Η οδήγηση φορτίων 220V δεν είναι εφικτή με την απευθείας εφαρμογή σημάτων χαμηλής τάσης από τα GPIO pins του Raspberry Pi , καθώς η ενεργοποίηση του πηνίου του ρελέ απαιτεί ρεύμα της τάξεως των ~40 mA ενώ κάθε GPIO μπορεί να παρέχει μέγιστο 16 mA. Για την επίλυση αυτού του προβλήματος η οδήγηση του ρελέ πραγματοποιείται με την συνδρομή του τρανζίστορ BC547 το οποίο λειτουργεί ως ενισχυτής ρεύματος. Η βάση του πολώνεται μέσω μία αντίστασης 1kΩ από την έξοδο GPIO επιτρέποντας την οδήγηση του πηνίου του ρελέ που βρίσκεται συνδεδεμένο στον συλλέκτη. Με αυτόν τον τρόπο η επαφή COM-NO του ρελέ μεταβαίνει στην επαφή COM-NC μεταφέροντας έτσι την φάση στην έξοδο της κλέμμας OUT1/2 μαζί με την γραμμή του ουδέτερου.

Για την προστασία του κυκλώματος κατά την απότομη αποκοπή τροφοδοσίας τοποθετείται μία δίοδος 1N4007 παράλληλα στο πηνίο απορροφώντας το ανάστροφο ρεύμα που δημιουργείται από την κατάρρευση του μαγνητικού πεδίου του ρελέ RT114012.

Όσον αφορά το υποσύστημα φωτισμού, η ρύθμιση των χρωμάτων αποτελεί βασική διεργασία του συστήματος, ωστόσο το Raspberry Pi δεν είναι κατάλληλο να οδηγήσει απευθείας φορτία υψηλών ρευμάτων ούτε μπορεί να διαχειριστεί τάσεις της τάξεως των 12 V. Για τον σκοπό αυτό σχεδιάστηκε ένα κύκλωμα ισχύος για την οδήγηση της εξόδου LED IN, το οποίο αποτελείται από τρία MOSFET IRLZ44N και τρεις αντιστάσεις των 220 Ω, μια σε κάθε κανάλι (Κόκκινο, πράσινο, μπλέ).

Η χρήση MOSFET για συστήματα διαχείρισης φωτισμού είναι μία καθιερωμένη τεχνική που συναντάται πολύ συχνά στην βιομηχανία. Καθώς προσφέρουν χαμηλή αντίσταση αγωγής ($R_{DS(ON)}$) μειώνοντας τις απώλειες και βελτιώνοντας την ενεργειακή απόδοση, ενώ ταυτόχρονα επιτρέπουν την ασφαλή διαχείριση υψηλών ρευμάτων. Ο έλεγχος των καναλιών RGB επιτυγχάνεται μέσω σημάτων PWM τα οποία εφαρμόζονται μέσω των αντιστάσεων στις πύλες των IRLZ44N επιτρέποντας έτσι την ρύθμιση της έντασης κάθε καναλιού και την δημιουργία σύνθετων χρωματικών αποχρώσεων.

4.2.2.2 Φάση σχεδίασης ηλεκτρονικού σχηματικού

Σε αυτήν την υποενότητα αναλύεται η τοπολογία των συνδέσεων του ηλεκτρονικού σχεδίου. Βασική προϋπόθεση για την ορθολογική σχεδίαση ενός τυποποιημένου κυκλώματος PCB, είτε σε ερασιτεχνικό είτε σε επαγγελματικό επίπεδο είναι η δημιουργία ενός ολοκληρωμένου ηλεκτρονικού σχηματικού. Το σχηματικό αποτελεί τη γραφική αποτύπωση της τοπολογίας των ηλεκτρονικών στοιχείων και είναι

αναγκαίο τόσο για την προετοιμασία και ανάλυση του κυκλώματος όσο και για την διασφάλιση της ηλεκτρικής ορθότητας.

Η λειτουργική σχεδίαση συνιστά θεωρητική αναπαράσταση του κυκλώματος η οποία επιτρέπει τον έλεγχο της σωστής συνδεσμολογίας και αποτελεί το απαραίτητο βήμα πριν από την υλοποίηση του PCB. Για τον εντοπισμό και την διόρθωση πιθανών σφαλμάτων τα σχηματικά υπόκεινται σε διαδικασία προσομοίωσης μέσω λογισμικών όπως το Pspice, όπου πραγματοποιείται ανάλυση των αναπτυσσόμενων ρευμάτων και τάσεων του κυκλώματος [33].

Η δημιουργία του σχηματικού πραγματοποιήθηκε αποκλειστικά στο ενσωματωμένου περιβάλλον Schematic του Altium Designer. Σε πρώτο στάδιο τοποθετήθηκαν όλα τα υλικά που ορίστηκαν κατά την φάση της πρότυπης σχεδίασης. Για λόγους σαφήνειας και εργονομίας τα τερματικά στοιχεία που αφορούν τις εξωτερικές συνδέσεις τοποθετήθηκαν στην περιφέρεια ενώ το κύκλωμα διακρίθηκε σε τρία επιμέρους υποκυκλώματα:

- **Υποσύστημα τροφοδοσίας χαμηλής τάσης**

  Το κύκλωμα τροφοδοσίας απεικονίζεται στο επάνω αριστερό τμήμα του σχεδίου. Η αρχή λειτουργίας του βασίζεται σε μία κλασική υλοποίηση μετατροπής AC-DC, αποτελούμενη από μετασχηματιστή (TRANS), γέφυρα ανόρθωσης (BRIDGE), πυκνωτή εξομάλυνσης (C1) και σταθεροποιητή τάσης (STABILIZER). Η παροχή του δικτύου συμβολίζεται μέσω του τερματικό IN ενώ έξοδος του σταθεροποιητή παρέχει σταθερή τάση +12 V DC η οποία σηματοδοτείται από αντίστοιχη ακίδα. Η οδός των 12 V αποτελεί την κύρια γραμμή τροφοδοσίας των βασικών ηλεκτρονικών στοιχείων ελέγχου και εκτείνεται παράλληλα με την γραμμή της γείωσης (GND) κατά μήκος του οριζόντιου άξονα ώστε να διασφαλιστεί η ευκολία σύνδεσης για την τροφοδοσία των υπολοίπων υποσυστημάτων.

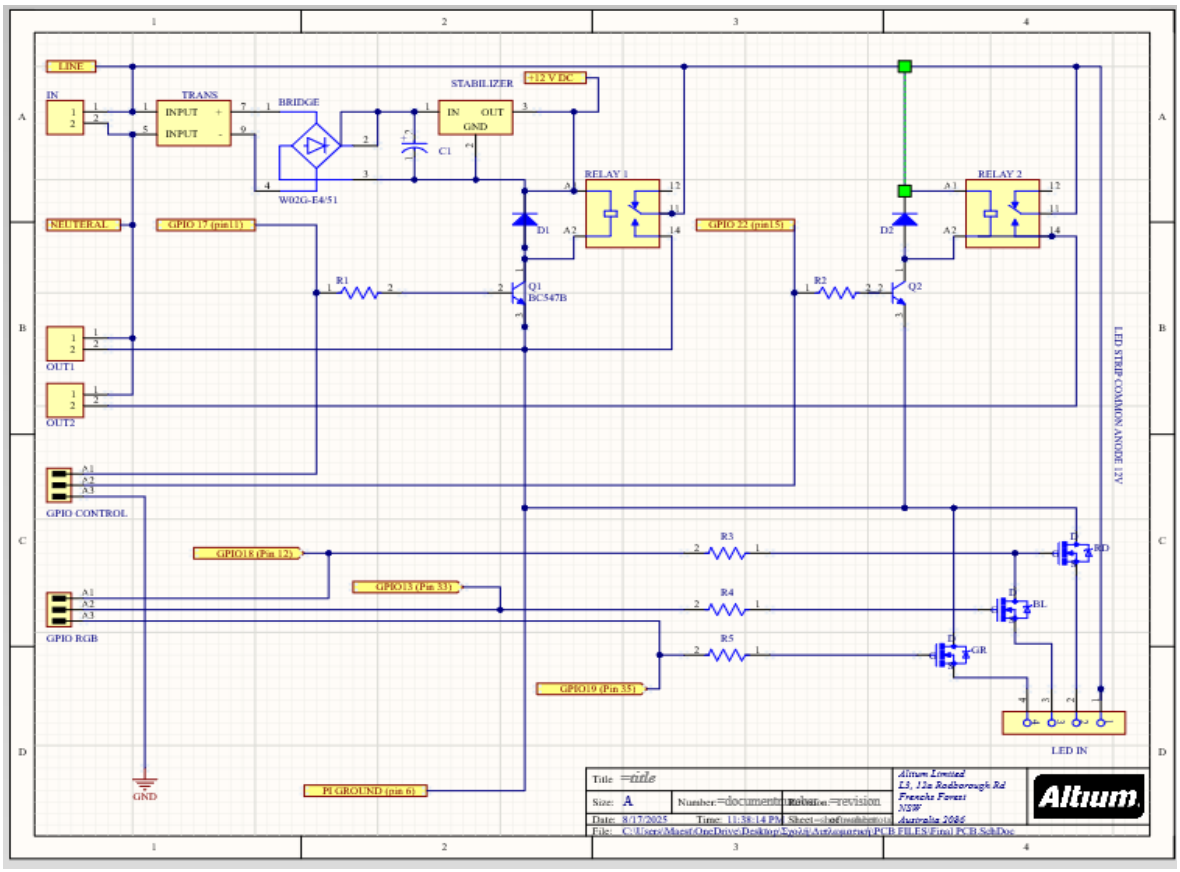
- **Υποσύστημα οδήγησης εξόδων εναλλασσόμενου ρεύματος**

  Στο δεξί επάνω τμήμα του σχεδίου υλοποιείται κύκλωμα οδήγησης φορτιών AC μέσω ρελέ. Κάθε επιμέρους διάταξη περιλαμβάνει έναν ηλεκτρονόμο (RELAY 1,RELAY 2) και μία δίοδο προστασίας (D1,D2) συνδεδεμένη παράλληλα στις επαφές του εσωτερικού του πηνίου (COIL). Τα δύο στοιχεία τοποθετούνται σε σειρά με το άκρο του συλλέκτη του τρανζίστορ (Q1,Q2), ενώ η βάση του συνδέεται μέσω αντίστασης (R1,R2) απευθείας με το τερματικό GPIO CONTROL. Τέλος ο εκπομπός οδηγείται απευθείας στην γείωση.

- **Υποσύστημα οδήγησης RGB LED μέσω PWM**

  Για λόγους ευκρίνειας ,το κύκλωμα τοποθετήθηκε στο χαμηλότερο επίπεδο του σχεδίου. Η διάταξη περιλαμβάνει τρία τρανζίστορ ισχύος MOSFET (RD,BL,GR) για τον έλεγχο της έντασης κάθε καναλιού. Οι πύλες των MOSFET οδηγούνται από την θύρα GPIO RGB του Raspberry Pi μέσω αντιστάσεων (R3,R4,R5). Οι επαφές της αποστράγγισης των τρανζίστορ

συνδέονται σε κοινή γείωση, ενώ οι πηγές του τοποθετούνται στο τερματικό LED IN. Στόχος αυτής της συνδεσμολογίας είναι η οδήγηση της καθόδου των LED στοιχείων στην γείωσης όταν γίνει εφαρμογή σήματος PWM από το Raspberry Pi.



Σχήμα 4.5 Ηλεκτρονικό σχηματικό της πλακέτας Maestro VF

Η δημιουργία του ηλεκτρονικού σχηματικού είναι το πρώτο και καθοριστικό στάδιο στην διαδικασία σχεδίασης ενός τυποποιημένου κυκλώματος (Σχήμα 4.5). Πρόκειται για ένα προϊόν προετοιμασίας όπου τα δεδομένα της αναπαράστασης μετατρέπονται αργότερα σε μηχανικά εξαρτήματα πραγματικών διαστάσεων. Η χρήση ακίδων στις γραμμές των συνδέσεων βελτιώνει την ευαναγνωσιμότητα του σχεδίου, διευκολύνει τον χρήστη, συμβάλλει στην αποφυγή λαθών και επιταχύνει την υλοποίηση του κυκλώματος, ενώ ταυτόχρονα παρέχει σαφείς πληροφορίες στον αναγνώστη για την κατανόηση του.

Μετά την ολοκλήρωση του είναι απαραίτητος ο έλεγχος της ορθότητας των συνδέσεων μέσω των ενσωματωμένων εργαλείων του Altium Designer καθώς επίσης και η προσομοίωση μέσω κατάλληλων λογισμικών ώστε να διασφαλιστεί η ηλεκτρική ευστάθεια του κυκλώματος.

4.2.3 Φάση δημιουργίας τυποποιημένου κυκλώματος

Η δημιουργία ηλεκτρονικού σχηματικού (schematic) είναι υψίστης σημασίας για την σχεδίαση του τελικού PCB. Μετά την ολοκλήρωση του ο χρήστης μπορεί να μεταφράσει το σχέδιο και να μεταφέρει τα ηλεκτρονικά στοιχεία και τις συνδέσεις στο γραφικό περιβάλλον σχεδίασης PCB. Σημαντικό

πλεονέκτημα αυτής της διεργασίας είναι η αμφίδρομη συσχέτιση μεταξύ των δύο layouts που σημαίνει πώς οι αλλαγές αντικατοπτρίζονται αμφότερα σε όλο το έργο.

Πρώτο βήμα για την γραφική σχεδίαση είναι ο καθορισμός των διαστάσεων της πλακέτας βάσει των προδιαγραφών των εξαρτημάτων και του ελάχιστου επιθυμητού μεγέθους (Σχήμα 4.6). Κατόπιν μελέτης στο πλαίσιο της χωροθέτησης των στοιχείων έγινε διάκριση του συνολικού κυκλώματος σε τρία βασικά υποκυκλώματα Η διαδικασία τοποθέτησης των αποτυπωμάτων των στοιχειών είναι απαραίτητη για τον προσδιορισμό του ακριβούς μεγέθους της πλακέτας. Στην συγκεκριμένη εφαρμογή λόγω του μικρού αριθμού εξαρτημάτων σχεδιάστηκε ένα διακριτικό κύκλωμα με λιτή τοπολογία διαστάσεων 100 mm x 80 mm [4.2.2].

Όπως αναφέρθηκε το μέγεθος της πλακέτας είναι μικρό και η συνδεσμολογία είναι αρκετά απλή επομένως δεν απαιτείται δημιουργία διπλής όψης για την υλοποίηση της. Το υπόστρωμα του κυκλώματος αποτελείται από υλικό FR-4 πάχους 1.6 mm με διηλεκτρική σταθερά εr = 4.8. Οι διαδρομές χαλκού βρίσκονται στην κάτω όψης της και προστατεύονται από στρώμα μάσκας ενώ τα pads παραμένουν διαθέσιμα για συγκόλληση. Η πάνω όψη καλύπτεται από ένα στρώμα μεταξοτυπίας και χρησιμοποιείται για την τοποθέτηση των εξαρτημάτων.

Ιδιαίτερη σημασία στην σχεδίαση τυποποιημένων κυκλωμάτων είναι η θέσπιση αποστάσεων για την απομόνωση γραμμών τροφοδοσίας με μεγάλη διαφορά δυναμικού. Το Maestro VF ενσωματώνει δύο διαφορετικές γραμμές τροφοδοσίας 220V AC και 12 V DC. Με δεδομένο το διηλεκτρικό υλικό χρησιμοποιήθηκαν οι πίνακες IPC-2221B/IEC 60664-1/62368-1/61010 για τον υπολογισμό του ελάχιστη επιτρεπτού clearance. Σύμφωνα με τον πίνακα IPC-2221B ορίστηκε ελάχιστη απόσταση 3mm μεταξύ των αγωγών υψηλής και χαμηλής τροφοδοσίας για την αποφυγή φαινομένων ηλεκτρικού τόξου.

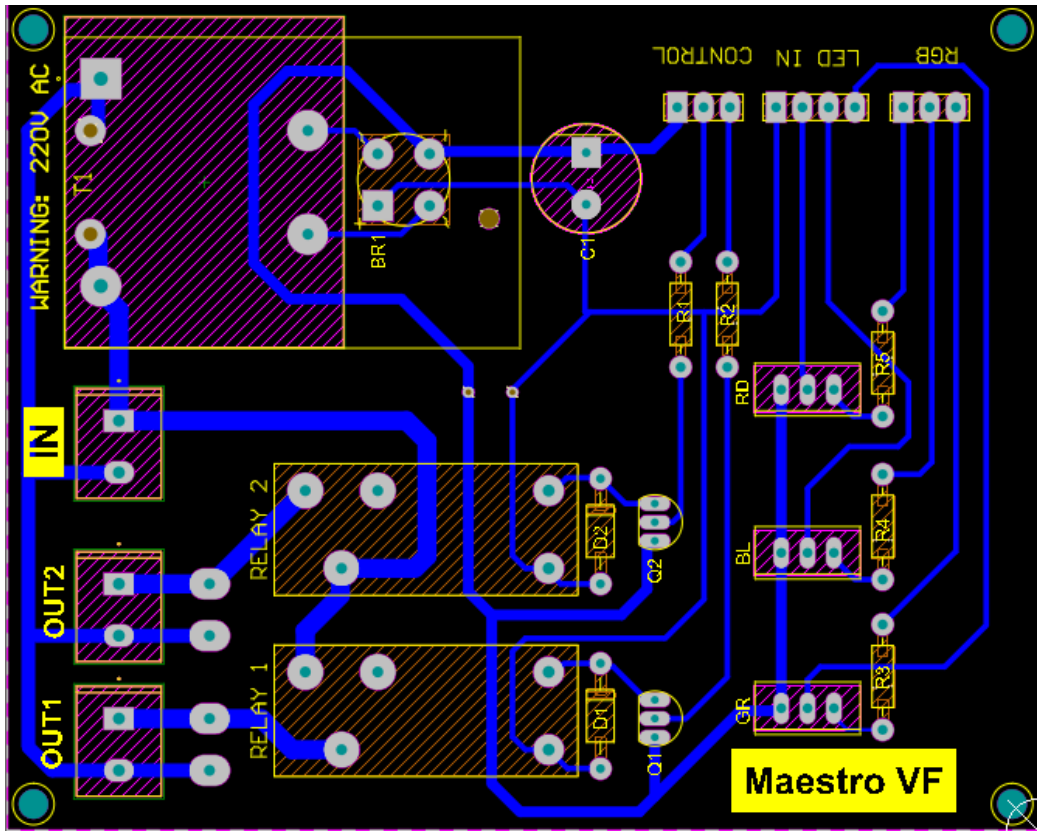
Στην συνέχεια τα βήματα σχεδίασης αφορούν περισσότερο την χωροθέτηση των στοιχειών της πλακέτας. Η αρχική μελέτη για την τοποθέτηση των εξαρτημάτων είναι σημαντική, καθώς παρέχει έναν πρόχειρο υπολογισμό των διαστάσεων του κυκλώματος, ωστόσο δεν αποτελεί τελική κατάσταση. Όπως αναφέρθηκε, η διάσπαση του κυκλώματος σε επιμέρους σημεία έχει ήδη ολοκληρωθεί ωστόσο χρειάζεται περαιτέρω ομαδοποίηση. Σημαντικός παράγοντας στην διαδικασία σχεδίασης είναι η δημιουργία σύντομων και άμεσων βρόχων, ιδιαίτερα όταν πρόκειται για συνδέσεις με τις επαφές γείωσης. Επιπλέον, τα αποτυπώματα των ιστών θα πρέπει να ακολουθούν όσο το δυνατόν πιο απλή συνδεσμολογία, ώστε να αποφεύγονται μεγάλες διαδρομές και ανεπιθύμητες γέφυρες.

Το πιο σύνθετο και απαιτητικό μέρος στην σχεδίαση κυκλωμάτων έγκειται στην αποτύπωση της κατάλληλης τοπολογίας μεταξύ των στοιχείων. Η σωστή δημιουργία του σχηματικού αποτελεί βασική προϋπόθεση για την ολοκλήρωση της διαδικασίας, καθώς η συσχέτιση των εξαρτημάτων αποτυπώνεται πλήρως στο διάγραμμα PCB. Καθοριστικό ρόλο διαδραματίζει η μελέτη του χώρου τοποθέτησης των υλικών με στόχο την εύρεση της βέλτιστη δρομολόγησης. Παρόλα αυτά είναι σύνηθες να απαιτείται

τροποποίηση, όπως η περιστροφή ή μετακίνηση υλικών, ώστε να απλοποιείται η συνολική εικόνα του κυκλώματος.

Για λόγους ομοιομορφίας και συγχρονισμού όταν περιλαμβάνονται πολλαπλά ταυτόσημα στοιχεία σε μία πλακέτα όπως στο κύκλωμα ελέγχου ρελέ , είναι καλή πρακτική η χάραξη ισομηκών διαδρόμων για την τροφοδοσία τους. Παράλληλα εφαρμόζονται τεχνικές εξοικονόμησης χώρου, όπως η δημιουργία σύντομων αγωγών όταν αυτό είναι εφικτό καθώς έτσι μειώνεται η ευαισθησία σε παρεμβολές. Μία ακόμη καθιερωμένη τεχνική είναι η δημιουργία ενός «πιάτου» γείωσης, δηλαδή μίας επιφάνειας χαλκού στην περιφέρεια του κυκλώματος για την μείωση των βρόχων γείωσης.

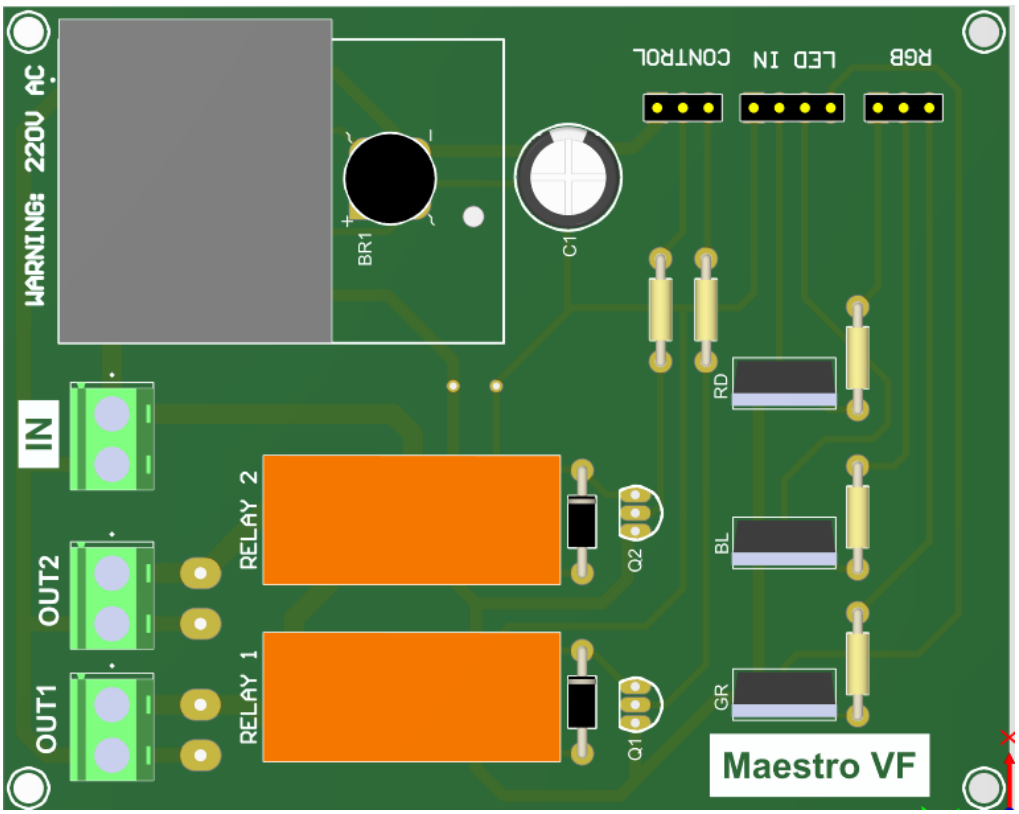
Εξίσου σημαντικοί είναι οι κανόνες που περιορίζουν τις αυθαιρεσίες στην σχεδίαση. Για παράδειγμα η χάραξη αγωγών με γωνία 90° δημιουργεί μεγάλη συγκέντρωση πεδίου, ενισχύοντας τα φαινόμενα των ανακλάσεων γι' αυτό προτιμώνται καμπύλες 45°. Επίσης οι αγωγοί των κυκλωμάτων 220V και 12V δεν αρκεί να διατηρούν την ελάχιστη απαιτούμενη απόσταση, αλλά πρέπει να ακολουθούν και τα πρότυπα διαμόρφωσης κατάλληλου πάχους. Στην συγκεκριμένη εφαρμογή, σύμφωνα με τίς προδιαγραφές χρησιμοποιήθηκαν αγωγοί πάχους 1.5-2 mm για το κύκλωμα τροφοδοσίας. Αντίθετα στο υπόλοιπο κύκλωμα ελέγχου των ρελέ και του PWM, λόγω χαμηλότερης τάσης χρησιμοποιήθηκαν αγωγοί πάχους 1-1.2 mm.



Σχήμα 4.6 Top overlay του Maestro VF

Τελικό βήμα για την ολοκλήρωση του προτύπου αποτελεί ο έλεγχος για τον εντοπισμό παραβάσεων. Το Altium Designer ως βιομηχανικό σχεδίασης διαθέτει ενσωματωμένα εργαλεία με τα οποία ο σχεδιαστής μπορεί και να ελέγξει την ευστάθεια των χαραγμένων διαδρομών και την τήρηση των κανόνων σχεδίασης. Ειδικότερα, το εργαλείο DRC ελέγχει όλες τις παραμέτρους όπως το πάχος αγωγών και οι αποστάσεις απομόνωσης επισημαίνοντας τυχόν παραλείψεις και λάθη που πρέπει να διορθωθούν [32].

Με το πέρας το απαραίτητων ελέγχων ο χρήστης εξάγει από το περιβάλλον του Altium τα απαραίτητα αρχεία σχεδίασης (Gerber files) που χρειάζεται ο κατασκευαστής για να προχωρήσει στην παραγωγή του κυκλώματος (Σχήμα 4.7). Τα αρχεία Gerber περιλαμβάνουν τις διαστάσεις της πλακέτας, την τοπολογία των συνδέσεων ,των τρυπών και όλα τα τεχνικά χαρακτηριστικά που αφορούν το υλικό [33].



Σχήμα 4.7 Τρισδιάστατη απεικόνιση του Maestro VF

4.3 Υπολογισμοί

Σε αυτήν την ενότητα παρουσιάζονται αναλυτικά οι υπολογισμοί στα άκρα των συνδέσεων των επιμέρους ηλεκτρονικών στοιχείων του Maestro VF. Οι υπολογισμοί κρίθηκαν απαραίτητοι για την

επαλήθευση και την αξιολόγηση της λειτουργικότητας του συστήματος ώστε να εγγυηθεί ή ασφάλεια χρήσης του συστήματος .

4.3.1 Κύκλωμα τροφοδοσίας

Σε αυτήν την υποενότητα αναλύεται η μεθοδολογία που ακολουθείται για τον υπολογισμό και την σχεδίαση της ενσωματωμένης διάταξης τροφοδοσίας του Maestro VF.

Βασική προϋπόθεση για τη τροφοδοσία των κυκλωμάτων οδήγησης είναι η μετατροπή της εναλλασσόμενης τάσης του δικτύου V_{ACPWR} = 230 V σε συνεχή τάση 12V .Επιπλέον το τροφοδοτικό κύκλωμα πρέπει να μπορεί να καλύπτει την ολική κατανάλωση του συστήματος P_{TOT}.

Ο μετασχηματιστής επιλέγεται ώστε η έξοδος στο δευτερεύον τύλιγμα να είναι V_{TROUT}~12 V .Η σχέση τάσης μετασχηματισμού δίνεται από την εξίσωση :

$$\frac{V_1}{V_2}=\frac{N_1}{N_2} \qquad \text{(Εξ 4.1)}$$

Όπου V1 η τάση του πρωτεύοντος και V2 η τάση δευτερεύοντος τυλίγματος.

Μετά τον μετασχηματισμό η τάση V_{TRAC} οδηγείται σε γέφυρα ανόρθωσης. Η πλήρη ανόρθωση έχει ως αποτέλεσμα τον διπλασιασμό της συχνότητας του σήματος εξόδου από f=50 Hz σε f=100 Hz. Συνεπώς η ανορθωμένη τάση συνοδεύεται από κυματώσεις (ripple) οι οποίες εξαρτώνται από το ρεύμα του φορτίου και τη χωρητικότητα του πυκνωτή εξομάλυνσης. Αυτό σημαίνει οτι η στιγμιαία τιμή τάσης μεταβάλλεται περιοδικά σε κάποιο εύρος.

Η κορυφή στην έξοδο του ανορθωτή προκύπτει από την σχέση:

$$V_{DCPeak}\sim V_{TRAC}\cdot\sqrt{2}\text{-}2\cdot V_F = 12\cdot 1.41\text{-}1.4=15.8\ V \qquad \text{(Εξ 4.2)}$$

Όπου V_F =0.6-0.7 V η πτώση τάσης σε κάθε δίοδο της γέφυρας.

Έστω συνολικό ρεύμα κατανάλωσης I_{TOT} =0.6-0.8 A

Για πυκνωτή εξομάλυνσης C=2200μF η κυμάτωση δίνεται από την σχέση:

$$\Delta V=\frac{I_{TOT}}{C\cdot f}=\frac{0.8}{0.0022\cdot 100}=2.73\ V_{P\text{-}P} \qquad \text{(Εξ 4.3)}$$

Επομένως η ελάχιστη τιμή τάσης στην έξοδο του πυκνωτή είναι:

$$V_{Ripple}=15.8\text{-}1.4=14.4\ V \qquad \text{(Εξ 4.4)}$$

Για την επίτευξη σταθερής εξόδου 12 V DC το κύκλωμα μπορεί να προσθέσει γραμμικό σταθεροποιητή τύπου LM7812 για την τροφοδοσία του φορτίου με κατανάλωση 0.6 - 0.8 A [34].

4.3.2 Κύκλωμα ελέγχου ρελέ

Σε αυτή την υποενότητα παρατίθενται οι υπολογισμοί που ολοκληρώθηκαν στο μικροκύκλωμα που αφορά την διαχείριση των ρελέ για το έλεγχο των εξόδων OUT1 και OUT2.

Το διπολικό τρανζίστορ BC547 χρησιμοποιείται ως διακόπτης όταν πολωθεί εκτός της ενεργούς περιοχής του. Αυτό σημαίνει πώς η έξοδος μπορεί να φέρει μόνο δύο καταστάσεις: κόρου (ON) και αποκοπής (OFF) (Πίνακας 4.3). Η παροχή τροφοδοσίας από τάσεις μεγαλύτερες από τις τάσεις ελέγχου επιτρέπει την οδήγηση επαγωγικών φορτιών όπως των ρελέ [23][39].

A/A	GPIO 17 (V)	GPIO 22 (V)	Relay A	Relay B	OUT1 (V AC)	OUT2 (V AC)
1	3.3	3.3	ON (12V)	ON (12V)	220	220
2	3.3	0.0	ON (12V)	OFF	220	0
3	0.0	3.3	OFF	ON (12V)	0	220
4	0.0	0.0	OFF	OFF	0	0

Πίνακας 4.3 Πίνακας συσχέτισης καταστάσεων θυρών GPIO και OUT1-2

Σε εφαρμογές όπου το φορτίο έχει αναφορά την γείωση και οδηγείται από τρανζίστορ NPN ο έλεγχος της λειτουργίας του επιτυγχάνεται με την εφαρμογή τάσεως από τις ακίδες GPIO του Raspberry Pi προς την βάση του BC547.Ως είσοδος σήματος ορίζεται η τάση V_{in} από τις ακίδες GPIO 17 και 22 ,ενώ το ρεύμα βάσης I_B αποτελεί το ρεύμα εισόδου του τρανζίστορ. Το ρεύμα που διαρρέει το επαγωγικό φορτίο του ρελέ ισούται με το ρεύμα συλλέκτη δηλαδή:

$$I_C = I_L \qquad \text{(Εξ 4.5)}$$

Σύμφωνα με τα τεχνικά φυλλάδια η τιμή της τάσης V_{in} κυμαίνεται στο διάστημα 0 – 3.3 V , ενώ το μέγιστο ρεύμα που μπορεί να αποδώσει μία θύρα GPIO είναι $I_{in(max)}$=16 mA.

Όταν η τάση $V_{in} = 0$ δηλαδή ισχύει $V_{in} < V_{BE(min)} = 0.7$ V, το τρανζίστορ βρίσκεται σε αποκοπή ,συνεπώς δεν υπάρχει διαρροή ρεύματος I_L στο φορτίο και το έλασμα του ρελέ παραμένει στην επαφή NC-COM με την συσκευή σε κατάσταση OFF αφού :

$$I_C = I_B = 0 \qquad \text{(Εξ 4.6)}$$

Όταν η $V_{in} = 3.3\ V$ δηλαδή ισχύει $V_{in} > V_{BE(min)} = 0.7$,το τρανζίστορ βρίσκεται σε κατάσταση κόρου ,αυτό σημαίνει πώς το επαγωγικό φορτίο του ρελέ διαρρεεται από ρεύμα I_L>0.

Για τον υπολογισμό του ρεύματος I_L ,λήφθηκε υπόψη η χαρακτηριστική τάση κορεσμού συλλέκτη - εκπομπού $V_{CE(SAT)}$ του BC547 επομένως:

$$I_L = \frac{V_{CC} - V_{CE(SAT)}}{R_L} = \frac{12-0.1}{360} = 33 \text{ mA} \qquad \text{(Εξ 4.7)}$$

Στην συνέχεια ρεύμα βάσης και το πραγματικό κέρδος hFE του τρανζίστορ υπολογίζονται ως εξής :

$$I_B=\frac{V_{IN}-V_{BE}}{R_B}=\frac{3.3-0.7}{1000}=2.6\ \text{mA} \quad \text{(Εξ 4.8)}$$

$$\text{και} \quad \beta=\text{hFE}=\frac{I_C}{I_B}=\frac{33}{2.6}=12.69 \quad \text{(Εξ 4.9)}$$

Συμπερασματικά όταν η τάση εισόδου στην βάση του τρανζίστορ V_{IN} υπερβεί την V_{BE} το τρανζίστορ μεταβαίνει σε κατάσταση κόρου, έτσι το έλασμα του ρελέ μετατίθεται από την επαφή NC-COM στην επαφή NO-COM επιτρέποντας την σύνδεση της φάσης στα άκρα του τερματικού στα ενεργοποιώντας την έξοδο [23].

4.3.3 Κύκλωμα ελέγχου MOSFET

Σε αυτήν υποενότητα παρατίθενται οι υπολογισμοί και οι αντιστοιχίες που πραγματοποιήθηκαν στο κύκλωμα ελέγχου παλμικής διαμόρφωσης.

Για την υλοποίηση του έργου χρησιμοποιήθηκε μία ταινία RGB με 54 LED / m και συνδεσμολογία κοινής ανόδου. Αυτό σημαίνει ότι οι τρείς γραμμές χρωμάτων τροφοδοτούνται από κοινή θετική τάση, ενώ οδηγούνται μέσω διάταξης στην γείωση. Το σύστημα υποστηρίζει παλέτα 15 χρωμάτων, με την εναλλαγή τους να επιτυγχάνεται μέσω PWM [35].

Το σήμα PWM παράγεται από τις ακίδες GPIO του Raspberry Pi και κατευθύνεται σε τρία MOSFET IRLZ44N ένα για κάθε χρώμα. Στην πύλη του MOSFET εφαρμόζεται τετραγωνικό σήμα V_{GS} =0-3.3V. Το ποσοστό χρόνου που το σήμα παραμένει ενεργό καθορίζει την μέση φωτεινότητα του αντίστοιχου χρώματος [35].

A/A	Color	DC GPIO 12 (%)	DC GPIO 13 (%)	DC GPIO 18 (%)
1	Black	0	0	0
2	Brown	165	42	42
3	Red	255	0	0
4	Orange	255	165	0
5	Gold	255	215	0
6	Yellow	255	255	0
7	Green	0	255	0
8	Cyan	0	255	255
9	Blue	0	0	255
10	Purple	128	0	128
11	Magenta	255	0	255
12	Pink	255	192	203
13	Gray	128	128	128
14	Silver	192	192	192
15	White	255	255	255

Πίνακας 4.4 Πίνακας συσχέτισης Duty Cycle ανά χρωματική εντολή

Το μέσο ρεύμα κατανάλωσης ανά χρώμα I_{AVG} δίνεται από την σχέση:

$$I_{AVG} = D \cdot I_{max} \quad \text{(Εξ 4.10)}$$

όπου I_{max} το μέγιστο ρεύμα διαρροής ανά χρωματική διάταξη

Η σχετική φωτεινότητα προκύπτει ως:

$$\text{Lum} = \frac{I_{AVG}}{I_{MAX}} \cdot 100\% \quad \text{(Εξ 4.11)}$$

Το IRLZ44N εμφανίζει πολύ χαμηλή αντίσταση αγωγής με τιμή :

$$R_{DS(ON)} = 0.017\ \Omega \quad \text{(Εξ 4.12)}$$

Επομένως η πτώση τάσης όταν το στοιχείο είναι ενεργό ισούται με:

$$V_{DS(ON)} = I_C \cdot R_{DS(ON)} \quad \text{(Εξ 4.13)}$$

και οι απώλειες ισχύος:

$$P_{loss} = I^2 \cdot R_{DS(ON)} = 0.475\ \text{mW} \quad \text{(Εξ 4.14)}$$

Η τιμή αυτή καθιστά τα MOSFET πρακτικά ως ιδανικούς διακόπτες με αμελητέες απώλειες που δεν επηρεάζουν την απόδοση του συστήματος

Σε ταινίες RGB, παρότι τα μπορούσε να θεωρηθεί πώς η ισχύς P_C ανα χρωματική διάταξη κατανέμεται ισόποσα ($P_C = \frac{P_{LED}}{3}$,) ωστόσο αυτό δεν ισχύει καθώς η κατανάλωση διαφοροποιείται λόγω των διαφορετικών τεχνικών χακτηριστικών στοιχειών LED [36].

Σύμφωνα με τα τεχνικά στοιχεία του κατασκευαστή τα SMD LED φέρουν διαφορετικές αντιστάσεις σε σειρά με την άνοδο τους, συγκεκριμένα ισχύει:

$$R_{Red} = 330\Omega \text{ και } R_{Green} = R_{Blue} = 150\ \Omega \quad \text{(Εξ 4.15)}$$

Ενώ η πτώση τάσης ανά στοιχείο είναι:

$$V_{Red} \sim 2\text{V και } V_{Green} = V_{Green} \sim 3\ \text{V} \quad \text{(Εξ 4.16)}$$

Για τον υπολογισμό της συνολικής κατανάλωση P_{LED} χρειάζεται ο υπολογισμός των ρευμάτων κατανάλωσης ανά κανάλι οποίος δίνεται από την σχέση:

$$I_T = 6 \cdot (I_R + I_B + I_G) \quad \text{(Εξ 4.17)}$$

Το ρεύμα ανά κανάλι υπολογίζεται ως:

$$I_c = \frac{12 - 3V_c}{R_c} \quad \text{(Εξ 4.18)}$$

Όπου c={R,B,G)

Έτσι ανά τμήμα:

$$I_R = \frac{12-6}{330} = 18.18\ \text{mA}, \quad \text{(Εξ 4.19)}$$

$$I_{B,G} = \frac{12-6}{330} = 20 \text{ mA} \quad \text{(Εξ 4.20)}$$

Επομένως με έξι τμήματα ανά μέτρο το συνολικό ρεύμα είναι:

$$I_{LED} = 6 \cdot (I_R + I_B + I_G) = 109+120+120 = 329 \text{ mA} \quad \text{(Εξ 4.21)}$$

Τέλος η συνολική ισχύς δίνεται από τον νόμο του Ohm :

$$P=V \cdot I \Leftrightarrow P_{LED} = 12 \cdot 329\, mA = 4.19 \text{ W / m} \quad \text{(Εξ 4.22)}$$

4.4 Αξιολόγηση Συστήματος

Σε αυτήν την υποενότητα γίνεται αξιολόγηση της κατασκευής του συστήματος αναφορικά με το συνολικό κόστος που δαπανήθηκε, την συνολική κατανάλωση και την ηλεκτρική απομόνωση.

4.4.1 Κόστος

Η συγκεκριμένη κατασκευή συνιστά ένα ολοκληρωμένο προϊόν IoT. Η μονάδα επεξεργασίας και επικοινωνίας (ΚΜΕ) δεν ενσωματώνεται στο κύκλωμα αλλά βασίζεται σε εξωτερικό μικροϋπολογιστή. Γεγονός που αυξάνει το κόστος. Ωστόσο η επιλογή αυτή προσφέρει ουσιαστικά οικονομικά πλεονεκτήματα, καθώς η πλατφόρμα είναι φιλική προς τον χρήστη, δεν παρουσιάζει λειτουργικές ασυμβατότητες και επιτρέπει την ανάπτυξη λογισμικού με πολυγλωσσική δομή χωρίς να απαιτείται πρόσθετη δαπάνη για εμπορικές άδειες .

Η σχεδίαση του πρότυπου κυκλώματος Maestro VF αποτελεί καθοριστικό παράγοντα για την μείωση του κόστους παραγωγής το οποίο δεν ξεπερνάει τα 18 €. Από την άλλη ,η εγκατάσταση περιφερειακών εξαρτημάτων για την ασφαλή χρήση και την διασφάλιση της αλληλεπίδρασης μεταξύ χρήστη-συστήματος αυξάνει επιπλέον την συνολική δαπάνη.

Σε σύγκριση με αντίστοιχα εμπορικά συστήματα IoT, το προτεινόμενο παρουσιάζει ιδιαίτερα ανταγωνιστικό κόστος, καθώς η συνολική τιμή του δεν υπερβαίνει τα 150 €. Αν και η χρήση εξωτερικής ΚΜΕ αυξάνει σημαντικά το κόστος ,εξασφαλίζει ταυτόχρονα ευελιξία και επεκτασιμότητα. Συνεπώς η τελική δαπάνη της κατασκευής κρίνεται προσιτή και βιώσιμη σε σχέση με τις παροχές του.

4.4.2 Κατανάλωση και απόδοση

Η αξιολόγηση της κατανάλωση ισχύος είναι κρίσιμο στοιχείο για την εκτίμηση της αποδοτικότητας ενός συστήματος. Το Maestro VF σχεδιάστηκε με στόχο ώστε να συνδυάζει χαμηλή ενεργειακή κατανάλωση με πλήρη λειτουργικότητα .

Οι μετρήσεις που ολοκληρώθηκαν στα μικροκυκλώματα έδειξαν οτι η ταινία RGB καταναλώνει περίπου 7 W / m , ενώ στο κύκλωμα οδήγησης των ρελέ απαιτεί μόλις 1.5 W.

Το Raspberry Pi 4 που λειτουργεί ως κύριος ΚΜΕ του συστήματος τροφοδοτείται με 5 V και καταναλώνει κατά μέσο όρο 1 A ,δηλαδή περίπου 5W ενώ αντίστοιχα τα περιφερειακά εξαρτήματα που το συνοδεύουν απαιτούν 4 W.

Η συνολική κατανάλωση ισχύος P_{SYS} του συστήματος υπολογίζεται ως:

$$P_{SYS} = P_{RPI} + P_{MVF} + P_{LED/M} + P_{PSK} = 5 + 1.5 + 7.2 + 4 = 18.7\ W \qquad (\text{Εξ } 4.12)$$

Η τελική τιμή αυτή δείχνει ότι παρά τον πρωτότυπο σχεδιασμό , το σύστημα του εμφανίζει ικανοποιητική ενεργειακή απόδοση συγκριτικά με τα αντίστοιχα εμπορικά προϊόντα που προορίζονται για παρόμοιες εφαρμογές.

4.4.3 Ηλεκτρική ασφάλεια

Η διασφάλιση της ακεραιότητας των υλικών κατά την λειτουργία ενός συστήματος είναι απαραίτητη προϋπόθεση για την αξιόπιστη και ασφαλή χρήση του. Στην περίπτωση του Maestro VF κρίθηκε ως επιτακτική η ανάγκη απομόνωσης του από τον μικροϋπολογιστή Raspberry προκειμένου να προστατευθεί τόσο το υλικό όσο και ο χρήστης.

Η μόνωση της τάσης του δικτύου από το υπόλοιπο σύστημα προβλέφθηκε αρχικά από την φάση σχεδίασης του Maestro VF και ολοκληρώθηκε κατά την συναρμολόγηση του τελικού προτύπου. Η είσοδος και η έξοδος της εναλλασσόμενης τάσης των 230 V πραγματοποιήθηκε μέσω ηλεκτρολογικού καλωδίου τύπου H03VV-F το οποίο συνδέθηκε σε τερματικές πρίζες Schuko .Η επιλογή αυτή διασφάλισε ότι ο χρήστης δεν εκτίθεται σε άμεση επαφή με αγώγιμα στοιχεία. Παράλληλα στο σώμα του συστήματος ενσωματώθηκαν φωτεινοί ενδείκτες οι οποίοι παρέχουν ενημέρωση στον χρήστη για την παρουσία τάσης στα τερματικά εξόδου [37].

Για την απομόνωση του Raspberry Pi από εξωτερικές τάσεις υψηλού φορτίου τοποθετήθηκαν μηχανικοί αποστάτες στήριξης στην πλακέτα Maestro VF,ώστε να αποτραπεί οποιαδήποτε ηλεκτρική επαφή με αγώγιμα μέρη .Επιπλέον η ψήκτρα αλουμινίου που περιβάλλει το μικροϋπολογιστή τοποθετήθηκε επάνω σε μονωτικό υλικό, ενισχύοντας την θερμική όσο και την ηλεκτρική του προστασία [38].

Κεφάλαιο 5ο: Συμπεράσματα και μελλοντικές Επεκτάσεις

Σε αυτό το κεφάλαιο παρουσιάζονται οι παρατηρήσεις που καταγράφηκαν κατά την διαδικασία ελέγχου και λειτουργίας του συστήματος, καθώς και οι πιθανές επεκτάσεις που θα μπορούσαν να ενισχύσουν τις δυνατότητες του σε μεταγενέστερο στάδιο. Επιπλέον παρατίθενται τα συμπεράσματα που προέκυψαν από την ανάπτυξη έργου καθώς και η συνολική αξιολόγηση των αποτελεσμάτων.

5.1 Παρατηρήσεις

Όπως συμβαίνει σε κάθε διαδικασία σχεδιασμού και ανάπτυξης ενός τεχνολογικού συστήματος, κατά την υλοποίηση του παρόντος έργου προέκυψαν τεχνικές και λειτουργικές προκλήσεις οι οποίες καταγράφηκαν και αξιολογήθηκαν. Η τεκμηρίωση αυτή αποσκοπεί στην αναγνώριση των παραγόντων που επηρέασαν την απόδοση του συστήματος και στην αναγνώριση των σημείων που χρήζουν βελτίωσης. Στην ενότητα αυτή περιγράφονται τα προβλήματα και οι παρατηρήσεις που αναδείχθηκαν κατά την φάση ελέγχου και λειτουργίας καθώς και τα αντίστοιχα μέτρα αντιμετώπισης που εφαρμόστηκαν.

5.1.1 Ταχύτητα και απόδοση συστήματος

Η ταχύτητα απόκρισης της εξόδου ενός ΣΑΕ, καθώς και ο χρόνος επεξεργασίας των δεδομένων εισόδου, αποτελούν βασικούς δείκτες αξιολόγησης της αποδοτικότητας του. Στην παρούσα υλοποίηση ο μικροϋπολογιστής Raspberry Pi 4 διαδραματίζει τον ρόλο της ΚΜΕ του συστήματος μέσω της οποίας δρομολογούνται όλες οι προβλεπόμενες ενέργειες.

Κατά την διαδικασία ελέγχου το σύστημα παρουσίασε καθυστερήσεις στο στάδιο εκτέλεσης των εντολών. Οι καθυστερήσεις αυτές αποδίδονται αμφότερα σε λειτουργικούς και τεχνικούς παράγοντες.

Σε τεχνικό επίπεδο διαπιστώθηκε ότι η ταχύτητα απόκρισης μειωνόταν αισθητά σε τυχαία χρονικά σημεία. Το πρόβλημα αυτό έγκειται στο γεγονός για το οποίο η εσωτερική θερμοκρασία του επεξεργαστή υπερέβαινε τους 80°C όταν ο φόρτος εργασίας ήταν αυξημένος. Για την διασφάλιση της ακεραιότητας του ο επεξεργαστής ενεργοποιούσε μηχανισμό thermal throttling με αποτέλεσμα την μείωση της απόδοση του έως και 90% [40].

Σε λειτουργικό επίπεδο παρατηρήθηκε πώς η ταχύτητα απόκρισης παρουσίαζε διακυμάνσεις, οι οποίες εξαρτώνται από την σταθερότητα της σύνδεσης στο διαδίκτυο, δεδομένου ότι η υλοποίηση βασίζεται σε Cloud-based APIs. Επιπλέον, η καθυστέρηση αυξανόταν σημαντικά όταν οι φωνητικές εντολές ήταν μεγάλου μήκους ή περιλάμβαναν λειτουργίες που σχετίζονταν με χρονοδιακόπτες. Το φαινόμενο αυτό αποδόθηκε στην αρχιτεκτονική του κώδικα, η οποία προβλέπει παύσεις και αναμονές για την ανίχνευση ή την ακύρωση επιμέρους εντολών [41].

Για τον περιορισμό του φαινομένου υπερθέρμανσης, εγκαταστάθηκε παθητική ψήκτρα στον επεξεργαστή. Η αναβάθμιση αυτή διατήρησε την εσωτερική θερμοκρασία σε τιμές κάτω των 75°C ,ακόμα και σε συνθήκες υψηλού φόρτου. Όσον αφορά τους περιορισμούς που σχετίζονται με το λογισμικό, οι δυνατότητες βελτίωσης ήταν περιορισμένες, καθώς η σταθερότητα σύνδεσης δεν εξαρτάται από τον κώδικα. Συνεπώς οι σημαντικότερες αλλαγές επικεντρώθηκαν στην σύμπτυξη του κώδικα σε μικρότερο μέγεθος και στην μείωση του χρόνου αναμονής κατά την ανίχνευση των εντολών εισόδου.

5.1.2 Αναγνώριση εντολών

Κατά την διάρκεια των αρχικών δοκιμών του συστήματος παρατηρήθηκε μειωμένη ακρίβεια στην διαδικασία ανίχνευσης, καθώς η μετατροπή φυσικής ομιλίας σε κείμενο πραγματοποιούταν μέσω εγκατεστημένου offline API.

Η διαδικασία αποσφαλμάτωσης πραγματοποιήθηκε αρχικά ορίζοντας τρία διαφορετικά μέτρα αξιολόγησης:

- Εναλλαγή ύφους
- Εναλλαγή του τόνου
- Παραμετροποίηση απόστασης

Για την εκτίμηση της σταθερότητας και της αξιοπιστίας του συστήματος ορίστηκε ένα σύνολο εντολών, το οποίο εκτελέστηκε δέκα φορές με μικρές παραλλαγές. Η ανάλυση των αποτελεσμάτων κατέδειξε ότι περίπου 60% αναγνωρίζονταν εσφαλμένα [42].

Για τον περιορισμό του σφάλματος υλοποιήθηκαν παρεμβάσεις τόσο στο τεχνικό όσο και το προγραμματιστικό τμήμα του συστήματος. Η ενσωμάτωση online διεπαφών στον κώδικα κατέστησε δυνατή την αξιοποίηση προηγμένων φίλτρα αναγνώρισης και επεξεργασίας σήματος, καθώς και λειτουργιών ανάλυσης και πρόβλεψης των ορθών δεδομένων μέσω προεκπαιδευμένων μοντέλων τεχνητής νοημοσύνης. Ο συνδυασμός αυτής της αναβάθμιση με την βελτίωση του υλικού αποδείχθηκε καταλυτικός παράγοντας για την βελτιστοποίηση της αποδοτικότητας [43].

Συγκεκριμένα στο Raspberry Pi εγκαταστάθηκε εξωτερική κάρτα ήχου με ενσωματωμένα φίλτρα και υψηλής ευαισθησίας πυκνωτικό μικρόφωνο με στόχο την μείωση του θορύβου και την καθαρότερη λήψη σήματος [19].

Οι αναβαθμίσεις αυτές οδήγησαν σε αύξησή του ποσοστού επιτυχούς αναγνώρισης στο 80%, ενώ η ελάχιστη απαιτούμενη απόσταση ομιλίας αυξήθηκε από 20 cm σε 40 cm, διατηρώντας την ποιότητα αναγνώρισης ακόμη και σε περιπτώσεις εναλλαγών ύφους. Επιπλέον, διαπιστώθηκε ότι οι εντολές με ένταση 50-65 dB αναγνωρίζονταν αξιόπιστα και όταν οι άλλοι παράγοντες μεταβάλλονταν ενώ η καταστολή του περιβαλλοντικού θορύβου μειώθηκε σημαντικά.

5.2 Προτάσεις βελτίωσης

Σε αυτήν την ενότητα παρουσιάζονται σχετικές προτάσεις για την τροποποίηση και την επέκταση των λειτουργιών του συστήματος, με σκοπό την βελτίωση αλληλεπίδρασης του με τον χρήστη.

5.2.1 Δημιουργία Application

Στη σύγχρονη εποχή τα περισσότερα ΣΑΕ υποστηρίζουν ασύρματο χειρισμό μέσω ειδικά ανεπτυγμένων εφαρμογών . Η αρχιτεκτονική δομή του παρόντος συστήματος μπορεί να προβλέψει μία τέτοιά επέκταση καθώς η πλακέτα ελέγχου διαθέτει δυνατότητα σύνδεσης με συσκευές μέσω πρωτόκολλων Bluetooth και Wi-Fi.

Κατά την φάση σχεδίασης εκτιμήθηκε οτι το έργο αποτελεί μια μικρογραφία ενός συστήματος Home Assistant. Αντίστοιχα με ολοκληρωμένα συστήματα Home και Voice Assistant, το προτεινόμενο σύστημα θα μπορούσε να ενσωματώνει πλατφόρμα για φορητές συσκευές ή Web εφαρμογή μέσω της οποίας θα παρέχεται δυνατότητα ζωντανής παρακολούθησης των συνδεδεμένων συσκευών και παραμετροποίησης των λειτουργιών τους [45].

Μέσω κατάλληλης διεπαφής το σύστημα θα μπορούσε να υποστηρίζει πέραν του φωνητικού ελέγχου και χειροκίνητη διαχείριση μέσω γραφικού περιβάλλοντος. Ένα τέτοιο περιβάλλον θα επιτρέπει στον χρήστη τον έλεγχο όλων των συνδεδεμένων συσκευών μέσω κατάλληλών πλήκτρων, καθώς και την δυνατότητα τροποποίησης των ετικετών ονομασίας των εξόδων ώστε να εξατομικεύσει τις φωνητικές εντολές σύμφωνα με τις ανάγκες του. Παράλληλα η προσαρμογή του MVF ως backend λογισμικό μπορεί να εξασφαλίσει ένα εξατομικευμένο interface , το οποίο θα παρέχει αποστολή εντολών , οπτική εποπτεία και ανατροφοδότηση βελτιώνοντας συνολικά την εμπειρία του χρήστη [46].

5.2.2 Βελτιστοποίηση ταχύτητας απόκρισης

Η επικοινωνία μεταξύ χρήστη και συστήματος μέσω API είναι πλήρως λειτουργική, ωστόσο η ταχύτητα απόκρισης ιδίως κατά την εκτέλεση της διεργασίας STT χρήζει βελτίωσης. Για την αντιμετώπιση του ζητήματος προτείνεται η ανάπτυξη ενός υβριδικού μοντέλου στο οποίο θα αξιοποιούνται παράλληλα online και offline APIs [6].

Η ταχύτητα σύνδεσης του Raspberry Pi με το δίκτυο επηρεάζει άμεσα την διαδικασία αναγνώρισης καθώς η πλατφόρμα Deepgram βασίζεται σε cloud-based streaming επεξεργασία. Παρότι ένα αμιγώς offline μοντέλου θα εξάλειφε την εξάρτηση από την δικτυακή σύνδεση, η απόδοση του δεν θα μπορούσε να συγκριθεί με την τρέχουσα έκδοση του MVF, δεδομένου οτι το Deepgram προσφέρει ακρίβεια που αγγίζει το 95%. Το κύριο μειονέκτημα είναι ο αυξημένος χρόνος απόκρισης, ο οποίος εξαρτάται σε μεγάλο βαθμό από την σταθερότητα και την ταχύτητα της σύνδεσης [6][32].

Η αρχιτεκτονική ενός υβριδικού συστήματος STT έγκειται στην επιλογή του κατάλληλου API ανάλογα με το μήκος της φωνητικής εντολής εισόδου. Η χρήση τοπικά εγκατεστημένου offline API όπως το VOSK σύμφωνα με μελέτες θεωρείται ιδανική για μικρές εντολές έως και τρείς λέξεις, μειώνοντας τον

χρόνο απόκρισης σε 200-300 ms σε σχέση με τα 600-700 ms του Deepgram.Για εντολές με μεγαλύτερου μήκους άνω των τεσσάρων λέξεων προτείνεται η χρήση online μοντέλων ούτος ώστε να διατηρείται υψηλός ο δείκτης ακριβείας [6].

Σε ό,τι αφορά την παραγωγή φωνής, διαπιστώθηκε ότι η αντικατάσταση της διεπαφής Text to Speech TTS του Elevenlabs από ένα offline μοντέλο όπως το Python API pyttsx3 μπορεί να μειώσει τον χρόνο απόκρισης κατά περίπου 500 ms. Ωστόσο η ποιότητα και η φυσικότητα της φωνητικής ανατροφοδότησης υστερούν αισθητά, γεγονός που δεν καθιστά επιτακτική την ανάγκη αντικατάσταση του τρέχοντος TTS μοντέλου [33].

Παράλληλα σημαντικό ρόλο στην ταχύτητα εκτέλεσης του λογισμικού διαδραματίζει και το υλικό, δεδομένου ότι τα APIs επιτελούν σημαντικό μέρος της επεξεργασίας, η ανάπτυξη ενός υβριδικού συστήματος όπως αναφέρθηκε θα μπορούσε να συνδυαστεί άριστα με έναν ισχυρότερο μικροϋπολογιστή όπως το Raspberry Pi 5. Η αυξημένη ταχύτητα λειτουργίας του επεξεργαστή και η μεγαλύτερη μνήμη RAM του μοντέλου δύνανται να διπλασιάσουν την αποδοτικότητα, αν και το κόστος αναβάθμισης του μοντέλου είναι σημαντικά υψηλότερο [16][20].

5.2.3 Βελτιστοποίηση αρχιτεκτονικής και σχεδιασμού του κυκλώματος

Η ανάπτυξη ενός συστήματος Home Assistant έχει ως στόχο την παροχή αξιόπιστης και ταχείας διαχείρισης όσο το δυνατόν μεγαλύτερο αριθμού συσκευών προσφέροντας ευελιξία στη σύνδεση και στον έλεγχο. Η αρχική υλοποίηση του συστήματος επιτρέπει τον χειρισμό τριών συσκευών μέσω ενσύρματης σύνδεσης στο τοπικό ηλεκτρονικό δίκτυο. Η εξέλιξη του έργου στοχεύει στην δημιουργία μίας ολοκληρωμένης λύσης ικανής να ενσωματώσει πολλαπλές συσκευές με ενσύρματη και ασύρματη συνδεσμολογία.

Στον σχεδιασμό ηλεκτρονικών συστημάτων, η βελτιστοποίηση του μεγέθους του τυποποιημένου κυκλώματος PCB αποτελεί μέλημα πρώτης τάξεως καθώς επηρεάζει άμεσα την αισθητική και την εργονομία της εγκατάστασης. Στην πρώτη εκδοχή του Maestro VF σημαντικό μέρος της πλακέτας δεσμεύτηκε για την υλοποίηση του κυκλώματος ελέγχου RGB ενώ ακόμα το τελικό μέγεθος αυξήθηκε σημαντικά από την προσθήκη μηχανικών οπών στήριξης.

Οι προσομοιώσεις ανασχεδιασμένου έδειξαν οτι η αφαίρεση του κυκλώματος RGB σε συνδυασμό με την αντικατάσταση των ρελέ με αντίστοιχα μικρότερων διαστάσεων μπορεί να μειώσει σημαντικά το πλάτος του Maestro VF έως και 40 mm. Το αποτέλεσμα αυτών των αλλαγών είναι ένα πιο συμπαγές και εργονομικό σύστημα με σαφείς μηχανικές και αισθητικές βελτιώσεις. Παράλληλα ,η δυνατότητα επέκτασης με την προσθήκη επιπλέον εξόδων προσφέρει ουσιαστική αναβάθμιση αν και απαιτεί ανασχεδιασμό της τοπολογίας των συνδέσεων και πιθανή αύξηση του κόστους.

Μια πιο εξελιγμένη εκδοχή μπορεί να υλοποιηθεί με την σχεδίαση ενός ολοκληρωμένου κυκλώματος που θα ενσωματώνει τον μικροελεγκτή ESP32. Η αντικατάσταση του Raspberry Pi από τον ESP32 σε

συνδυασμό με την αύξηση του αριθμού εξόδων αποτελούν σημαντική πρόκληση για την δημιουργία ενός πλήρους αυτόνομου συστήματος Home assistant. Με αυτόν τον τρόπο εξαλείφεται η ανάγκη εξωτερικών συνδέσεων μεταξύ μικροϋπολογιστή και υλικού, ενώ ταυτόχρονα περιορίζεται ο απαιτούμενος χώρος εγκατάστασης καθιστώντας το σύστημα ταχύτερο και εργονομικό [46][47].

5.3 Συμπεράσματα και αποτίμηση του έργου

Η απόπειρα ανάπτυξης ενός συστήματος στα πρότυπα του Home Assistant συνέβαλε ουσιαστικά στην ενίσχυση των δεξιοτήτων στον τομέα της σχεδίασης και του προγραμματισμού. Στο πλαίσιο υλοποίησης της παρούσας διπλωματικής εργασίας ακολουθήθηκε προσέγγιση που προσομοίαζε την διαδικασίας παραγωγής βιομηχανικού προϊόντος, από το στάδιο της σύλληψης της ιδέας έως και την τελική υλοποίηση.

Μέσα από την διαδικασία σχεδίασης πρότυπου τυποποιημένου κυκλώματος PCB , επιτεύχθηκε η εξατομίκευση του τελικού προϊόντος σύμφωνα με τις απαιτήσεις του έργου .Παράλληλα, η ερευνά που ολοκληρώθηκε γύρω από την βιομηχανική σχεδίαση και παραγωγή συνέβαλε στην κατανόηση των βασικών αρχών που διέπουν την ορθή λειτουργία τυποποιημένων κυκλωμάτων, ακολουθώντας θεμελιώδη πρότυπα αρχιτεκτονικής ,ασφάλειας και αισθητικής.

Για την πλήρη υποστήριξη του καινοτόμου υλικού που κατασκευάστηκε και την σύνδεση του με το κύκλωμα του Raspberry Pi αναπτύχθηκε αντίστοιχο πρότυπο λογισμικό. Η συγγραφή του κώδικα σε γλώσσα Python αποδείχθηκε καθοριστικής σημασίας, καθώς η αρχιτεκτονική και η σύνταξη επέτρεψαν την διασύνδεση με το υλικό και την εύκολή ενσωμάτωση των διεπαφών TTS και STT.

Συνολικά η παρούσα ΔΕ επικεντρώθηκε στην μελέτη βασικών τεχνολογιών του IoT, όπως το Cloud Computing, τα πρότυπα χαμηλής κατανάλωσης και τα υβριδικά μοντέλα επεξεργασίας και παραγωγής φωνητικών σημάτων. Η υλοποίηση ανέδειξε την δυνατότητα κατανόησης, σχεδίασης και ανάπτυξης σύνθετων συστημάτων σε όλο το φάσμα τους, τόσο σε κατασκευαστικό όσο και σε προγραμματιστικό επίπεδο.

Κατά τη διαδικασία συγγραφής της παρούσας διπλωματικής εργασίας αξιοποιήθηκαν εργαλεία τεχνητής νοημοσύνης για τη βελτίωση της σύνταξης και της ορθογραφίας, με στόχο την παραγωγή ενός πληρέστερου και ακαδημαϊκά τεκμηριωμένου κειμένου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] F. Jelinek, “Continuous speech recognition by statistical methods,” *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, Apr. 1976.

[2] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, USA: Prentice Hall, 1993, pp. 1–507.

[3] B. H. Juang and L. R. Rabiner, “Automatic speech recognition – A brief history of the technology development,” in *Encyclopedia of Language and Linguistics*, 2nd ed., vol. 1. Oxford, U.K.: Elsevier, 2005, pp. 1–7.

[4] S. B. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[5] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[6] M. Rastogi, “Analysis of ASR systems: Deepgram, Whisper v2, Assembly AI, Azure Speech,” 2023.

[7] J. Bakagianni, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, “A systematic survey of natural language processing for the Internet of Things,” *Future Internet*, vol. 17, no. 3, pp. 1–25, Mar. 2025.

[8] A. Gatt and E. Krahmer, “Survey of the state of the art in natural language generation: Core tasks, applications and evaluation,” *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, Jan. 2018.

[9] N. Naik and P. Jenkins, “IoT protocols and standards: A comprehensive survey,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4722–4745, Jun. 2020, doi: 10.1109/JIOT.2019.2963750.

[10] M. Banzi and M. Shiloh, *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*, 3rd ed. Sebastopol, CA, USA: Maker Media, Inc., 2014, pp. 1–262.

[11] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[12] M. Porcheron, J. E. Fischer, S. Reeves, and S. Sharples, “Voice interfaces in everyday life,” in *Proc. CHI Conf. Human Factors in Computing Systems (CHI ’18)*, 2018, Paper 640, pp. 1–12.

[13] G. Prasanna and R. Narayanadass, “Low cost home automation using offline speech recognition,” *International Journal of Signal Processing Systems*, vol. 2, no. 2, pp. 96–101, Jun. 2014, doi: 10.12720/ijsps.2.2.96-101.

[14] A. Obiad, “Design and implementation of voice recognition based smart home automation system,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 7, no. 6, pp. 1–5, Jun. 2018.

[15] Y. Li, S. Kim, and E. Sy, “A survey on Amazon Alexa attack surfaces,” *arXiv preprint arXiv:2102.09612*, Feb. 2021.

[16] Raspberry Pi Foundation, “Raspberry Pi 4 Model B specifications,” Raspberry Pi Documentation. [Online]. Available: https://www.raspberrypi.com/documentation/computers/raspberry-pi-4.html.

[17] Raspberry Pi Foundation, “Raspberry Pi OS,” Raspberry Pi Documentation. [Online]. Available: https://www.raspberrypi.com/software/.

[18] Wi-Fi Alliance, “Wi-Fi certified 802.11ac,” [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-ac

[19] Vention, “USB external sound card adapter – product specifications,” Vention Official Website. [Online]. Available: https://ventiontech.com/. [Accessed: Aug. 24, 2025].

[20] F. Ke, J. Zhang, Y. Xu, and J. Wang, “Thermal management of single-board computers: A case study on Raspberry Pi 4,” Journal of Thermal Science and Engineering Applications, vol. 15, no. 3, pp. 1–10, Jun. 2023, doi: 10.1115/1.4055923.

[21] TE Connectivity, Power PCB Relay RT1 – Data Sheet, 2018. [Online]. Available: https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FRT1%7F0718%7Fpdf%7FEnglish%7FENG_DS_RT1_0718.pdf

[22] Vishay Semiconductors, 1N4001–1N4007 Standard Recovery Rectifier, 1.0 A – Datasheet, 2016. [Online]. Available: https://www.vishay.com/docs/88503/1n4001.pdf

[23] Texas Instruments, Inductive Load Protection with Flyback Diodes, Application Report SLVA689, 2015. [Online]. Available: https://www.ti.com/lit/an/slva689/slva689.pdf

[24] Fairchild Semiconductor, *BC547 / BC547A / BC547B / BC547C NPN General Purpose Amplifier Transistor – Datasheet*, Rev. 1.1.1 [Online]. Available: https://www.farnell.com/datasheets/59764.pdf

[25] Vishay Siliconix, IRLZ44 Power MOSFET – Datasheet, 2018. [Online]. Available: https://www.vishay.com/docs/91328/irlz44.pdf

[26] TE Connectivity, 282836-2 Terminal Block – Product Datasheet. [Online]. Available: https://www.te.com/usa-en/product-282836-2.html

[27] Python Software Foundation,Python Documentation.[Online].Available: https://docs.python.org/3/

[28] Deepgram Inc., Deepgram Speech-to-Text API Documentation. [Online]. Available: https://developers.deepgram.com

[29] ElevenLabs Inc., ElevenLabs Speech Synthesis API Documentation. [Online]. Available: https://docs.elevenlabs.io

[30]Altium Limited, Altium Designer Documentation. [Online]. Available: https://www.altium.com/documentation/altium-designer

[31] Microsoft, Visual Studio Code – Code Editing. Redefined. [Online]. Available: https://code.visualstudio.com/

[32] Altium Limited, Altium Designer Documentation. [Online]. Available: https://www.altium.com/documentation/altium-designer

[33] Z. Peterson, "The PCB Design Process Methodology and Application," Altium Resources, Nov. 9, 2020. [Online]. Available: https://resources.altium.com/p/pcb-designing-tutorial.

[34] Texas Instruments, LM7812 Voltage Regulator – Datasheet, 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm7812.pdf

[35] K. C. Ho, "Dimming techniques focusing on the improvement in brightness linearity," *Electronics*, vol. 10, no. 2, p. 206, Jan. 2021, doi: 10.3390/electronics10020206.

[36] M. A. Green, "LED fundamentals: Characteristics, efficiency, and applications," *Progress in Photovoltaics: Research and Applications*, vol. 19, no. 7, pp. 901–912, Nov. 2011.

[37] International Electrotechnical Commission, *IEC 60664-1: Insulation coordination for equipment within low-voltage systems – Part 1: Principles, requirements and tests*, IEC Standard, 2020.

[38] Association Connecting Electronics Industries (IPC), *IPC-2221B: Generic Standard on Printed Board Design*, IPC Standard, 2012.

[39] Μ. Ν. Σπάσος και Κ. Θ. Αμοιρίδης, *Σύγχρονες Εφαρμογές Αναλογικών Ηλεκτρονικών: Θεωρία – Ασκήσεις – Πειράματα – Εφαρμογές OrCAD – PSpice*, 2η έκδ. Θεσσαλονίκη, Ελλάδα: Εκδόσεις ΑΙΒΑΖΗ, 2016, σσ. 46–47. ISBN: 978-960-549-026-3.

[40] T. Benoit-Cattin, D. Velasco-Montero, and J. Fernández-Berni, "Impact of thermal throttling on long-term visual inference in a CPU-based edge device," *Electronics*, vol. 9, no. 12, p. 2106, 2020.

[41] B. Cougo, F. Lima, C. Rech, and H. Pinheiro, "Influence of PWM methods on semiconductor losses and thermal cycling of SiC inverter," *Electronics*, vol. 9, no. 11, p. 1871, Nov. 2020.

[42] W. Xiong, J. Droppo, X. Huang, F. Seide, A. Stolcke, D. Yu, and G. Zweig, “The Microsoft 2016 conversational speech recognition system,” in *Proc. Interspeech Conf. on Speech Communication and Technology (ICASS)*, 2017, pp. 5255–5259.

[43] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, … Y. Zhu, “Deep Speech 2: End-to-End Speech Recognition in English and Mandarin,” in *Proc. Int. Conf. on Machine Learning (ICML)*, 2016, pp. 173–182.

[44] Di Leo, S., De Cicco, L. & Mascolo, S. (2025). Real-Time Speech-to-Text on Edge: A Prototype System for Ultra-Low Latency Communication with AI-Powered NLP. Information, 16(8): 685.

[45] Ahmed, D., Hassan, M., & Khan, S. (2024). Design and Implementation of an ESP32-Based Smart Home Automation System With Environmental Monitoring and Automated Controls. IOSR-JECE, 19(6), 23–28.

[46] S. M. H. Anik, X. Gao, H. Zhong, X. Wang, and N. Meng, “Automation configuration in smart home systems: Challenges and opportunities,” *arXiv preprint* arXiv:2401.12345, 2024.

ΠΑΡΑΡΤΗΜΑ Α

Σε αυτό το παράρτημα παρατίθεται ο πλήρες κώδικας που χρησιμοποιήθηκε για την ανάπτυξη του MVF software.

```
from elevenlabs.client import ElevenLabs
from difflib import get_close_matches
from elevenlabs import stream , VoiceSettings
import threading
from dotenv import load_dotenv
import re
import os
from deepgram import (
    DeepgramClient,
    DeepgramClientOptions,
    LiveTranscriptionEvents,
    LiveOptions,
    Microphone,
)
import pigpio
import RPi.GPIO as GPIO

pi = pigpio.pi()

client = ElevenLabs(api_key= "API_KEY")
def text_to_speech(input):
    audio_stream = client.generate(
    text=input,
    stream=True,
    model='eleven_multilingual_v2',
    voice="TX3LPaxmHKxFdv7VOQHJ",
    )
    stream(audio_stream)

def Speech_to_Text():
    is_finals = []
    final_text = ""
    finish_event = threading.Event()
#deepgram initializing
    try:
        deepgram: DeepgramClient = DeepgramClient("API_KEY")
        dg_connection = deepgram.listen.websocket.v("1")
        def on_open(self, open, **kwargs):
            pass

        def on_message(self, result, **kwargs):
            nonlocal is_finals
            nonlocal final_text
            sentence = result.channel.alternatives[0].transcript
            if len(sentence) == 0:
                return
            if result.is_final:
                is_finals.append(sentence)

                if result.speech_final:
                    utterance = " ".join(is_finals)
                    final_text = utterance
                    is_finals = []
                    finish_event.set()


        def on_metadata(self, metadata, **kwargs):
```

```
            pass

        def on_speech_started(self, speech_started, **kwargs):
            pass

        def on_utterance_end(self, utterance_end, **kwargs):
            nonlocal is_finals
            if len(is_finals) > 0:
                utterance = " ".join(is_finals)
                is_finals = []

        def on_close(self, close, **kwargs):
            pass

        def on_error(self, error, **kwargs):
            pass
            #print(f"Handled Error: {error}")

        def on_unhandled(self, unhandled, **kwargs):
            pass
        dg_connection.on(LiveTranscriptionEvents.Open, on_open)
#deepgram live options
        options: LiveOptions = LiveOptions(
            model="nova-3",
            language="en",
            smart_format=True,
            encoding="linear16",
            channels=1,
            sample_rate=16000,
            interim_results=True,
            utterance_end_ms="1000",
            vad_events=True,
            endpointing=200,
        )
        addons = {
            "no_delay": "true"
        }
        if dg_connection.start(options, addons=addons) is False:
            print("Failed to connect to Deepgram")
            return None
        microphone = Microphone(dg_connection.send)
        microphone.start()
        finish_event.wait()
        dg_connection.finish()
        microphone.finish()
        return final_text
    except Exception as e:
        print(f"Could not open socket: {e}")
        return None
#parametrs parssing list
enable_actions = ["turn on", "open", "enable"]
disable_actions = ["turn off", "close", "disable"]
devices = ["relay", "device", "system"]
models = ["alpha", "beta","both"]
#numbers parssing list

number_words = {
    "one": 1, "two": 2, "three": 3, "four": 4, "five": 5,
    "six": 6, "seven": 7, "eight": 8, "nine": 9, "ten": 10,
    "fifteen": 15, "twenty": 20, "twenty five": 25, "thirty": 30,
    "thirty five": 35, "forty": 40, "forty five": 45,
    "fifty": 50, "fifty five": 55, "sixty": 60
}

#time units parssing list
time_units = {
```

```
    "second": 1, "seconds": 1,
    "minute": 60, "minutes": 60,
    "hour": 3600, "hours": 3600
}
#colors parssing list
color_rgb = {
    "red": (255, 0, 0),
    "green": (0, 255, 0),
    "blue": (0, 0, 255),
    "yellow": (255, 255, 0),
    "cyan": (0, 255, 255),
    "magenta": (255, 0, 255),
    "white": (255, 255, 255),
    "black": (0, 0, 0),
    "orange": (255, 165, 0),
    "purple": (128, 0, 128),
    "pink": (255, 192, 203),
    "gray": (128, 128, 128),
    "brown": (165, 42, 42),
    "gold": (255, 215, 0),
    "silver": (192, 192, 192)
}

#timers status for output list
active_timers = {
    "alpha": None,
    "beta": None
}

#setup GPIO  for both outputs
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(22, GPIO.OUT)

GPIO.output(17, GPIO.LOW)  # Relay alpha OFF
GPIO.output(22, GPIO.LOW)  # Relay beta OFF
output_status = {"Alpha": False, "Beta": False}


# Reset all PWM channels (RGB/MOSFET)
pi.set_PWM_dutycycle(12, 0)  # Red
pi.set_PWM_dutycycle(13, 0)  # Green
pi.set_PWM_dutycycle(18, 0)  # Blue

ready_to_listen = True
#setup color parssing function
def set_color(color_name):
    rgb = color_rgb.get(color_name.lower())
    if rgb:
        r, g, b = rgb
        pi.set_PWM_dutycycle(12, int((r / 255) * 255))  # R
        pi.set_PWM_dutycycle(13, int((g / 255) * 255))  # G
        pi.set_PWM_dutycycle(18, int((b / 255) * 255))  # B
        print(f"Color set to {color_name}")
    else:
        print("Unknown color")
#setup trigger prhases about color parssing
def detect_color_command(text):
    pi.set_PWM_dutycycle(12, 0)
    pi.set_PWM_dutycycle(13, 0)
    pi.set_PWM_dutycycle(18, 0)
    text = text.lower()
    trigger_words = ["set", "change", "make"]
    if any(trigger in text for trigger in trigger_words):
```

```
        for color in color_rgb.keys():
            if color in text:
                return color
    return None

def find_closest(word, word_list, cutoff=0.6):
    match = get_close_matches(word, word_list, n=1, cutoff=cutoff)
    return match[0] if match else None

def model(recognized_text):
    command_action = None
    object = None
    model_output = None
    text = recognized_text.lower()
    for action in sorted(enable_actions + disable_actions, key=lambda x: -len(x)):
        if action in text:
            command_action = action
            break
    for word in text.split():
        if not object:
            match = find_closest(word, devices)
            if match:
                object = match
        if not model_output:
            match = find_closest(word,models)
            if match:
                model_output = match
    return command_action, object, model_output

#function for calculating the exactly total seconds for executing commands
def extract_total_seconds(text):
    text = text.lower()
    total = 0
    for unit, multiplier in time_units.items():
        for word, value in number_words.items():
            phrase = f"{word} {unit}"
            if phrase in text:
                total = value * multiplier
                print(f" {value} {unit}  {total} ")
                return total

    match = re.search(r'(\d+)\s*(seconds?|minutes?|hours?)', text)
    if match:
        value = int(match.group(1))
        unit = match.group(2)
        multiplier = time_units.get(unit.rstrip('s'), 1)
        total = value * multiplier
        print(f"kkk: {value} {unit} ? {total} ")
        return total
    return None

def timer_execution(cur_state, model, action, duration):
    pin = 17 if model == "alpha" else 22
    # Case 1 & 2: ENABLE command
    if action in enable_actions:
        if not cur_state:
            # Case 1: already OFF, turn ON
            GPIO.output(pin, GPIO.HIGH)
            output_status[model] = True
            print(f"{model} is ON for {duration} seconds")
            text_to_speech(f"{model} is ON for {duration} seconds")
        else:
            # Case 2: already ON, don't change now
            print(f"{model}  is already ON. Will turn OFF after {duration} seconds")
            text_to_speech(f"{model} is already ON. Will turn it OFF after {duration}
seconds")
```

```
            #turn OFF later

        t = threading.Timer(duration, lambda: restore_state(model, pin, False))
        t.start()
        active_timers[model]=t
        return False  #latest status = OFF

    # Case 3 & 4: DISABLE command
    elif action in disable_actions:
        if cur_state:
            # Case 3: already ON, turn OFF
            GPIO.output(pin, GPIO.LOW)
            output_status[model] = False
            print(f"{model} turned OFF for {duration} seconds\n")
            #text_to_speech(f"{model} turned OFF for {duration} seconds")
        else:
            # Case 4: already OFF, don't change now
            print(f"{model} is already OFF. Will turn ON after {duration} seconds\n")
           {duration} seconds")
            # turn ON later
        t = threading.Timer(duration, lambda: restore_state(model, pin, True))
        t.start()
        active_timers[model] = t
        return True  #latest status = ON
def restore_state(model, pin, state):
    GPIO.output(pin, GPIO.HIGH if state else GPIO.LOW)
    output_status[model] = state
    print(f"{model} restored to {'ON' if state else 'OFF'}\n")
    active_timers[model] = None
if __name__ == "__main__":
   output_status = {"alpha": False, "beta": False}
print("Hello Maestro, I am ready to hear you. \n")
text_to_speech("Hello Maestro, I am ready to hear you.")

try:
    while True:
        text = Speech_to_Text()
        #When the users talk to microphone the function StT will be activated
        if text:
            print(f"Final Recognized Text: {text}\n")
            command_action, object, model_output = model(text)
            color_name = detect_color_command(text)
            duration = extract_total_seconds(text)
            print(f"Parsed  action: {command_action}\ndevice: {object}\nmodel:
{model_output}\ncolor: {color_name}\nduration: {duration}\n")
#parssing color atribute then color setup function is activated
            if color_name:
                set_color(color_name)
                #text_to_speech(f"Color set to {color_name}")
                continue
            if command_action and object and model_output:
                if model_output == "both":
#Trigger case about both systems
                    if command_action in enable_actions:
#Both systems getting high
                        GPIO.output(17, GPIO.HIGH)
                        GPIO.output(22, GPIO.HIGH)
                        output_status["alpha"] = True
                        output_status["beta"] = True
                        print("Both systems are now ON\n")
                        text_to_speech("Alpha and Beta are now ON")
                    elif command_action in disable_actions:
#Both systems getting low
                        GPIO.output(17, GPIO.LOW)
                        GPIO.output(22, GPIO.LOW)
                        output_status["alpha"] = False
```

```
                        output_status["beta"] = False
                        print("Both systems are now OFF\n")
                        text_to_speech("Alpha and Beta are now OFF")
#cancel and retrigger new parse commands
                if model_output == "both":
                    for key in ["alpha", "beta"]:
                        if active_timers.get(key):
                            active_timers[key].cancel()
                            active_timers[key] = None
                else:
                    if active_timers.get(model_output):
                        active_timers[model_output].cancel()
                        active_timers[model_output] = None
#calls timer execute function
                if duration:
                     output_status[model_output] = timer_execution(
                     output_status[model_output],
                     model_output,
                     command_action,
                     duration)
                else:
                    pin = 17 if model_output == "alpha" else 22 if model_output == "beta"
else None
                    if pin is not None:
                        if command_action in enable_actions:
#Check if action variable belong to Enable List
                            if output_status[model_output]:
                                print(f"System {model_output} is already ON!\n")
                                text_to_speech(f"System {model_output} is already ON")
                            else:
                                GPIO.output(pin, GPIO.HIGH)
                                output_status[model_output] = True
                                print(f"{model_output} is now ON\n")
                                text_to_speech(f"{model_output} is now ON")
                        elif command_action in disable_actions:
#Check if action variable belong to Enable List

                            if not output_status[model_output]:
                                print(f"System {model_output} is already OFF\n")
                                text_to_speech(f"System {model_output} is already OFF")
                            else:
                                GPIO.output(pin, GPIO.LOW)
                                output_status[model_output] = False
                                print(f"{model_output} is now OFF\n")
                                text_to_speech(f"{model_output} is now OFF")
                    else:
                        print(" Unknown system model!\n")
            else:
                print(" Unrecognized command!\n")
                text_to_speech("I did not understand the command!")
        else:
            print(" No speech detected!\n")
            text_to_speech("Sorry, I didn't hear anything!")
        print("Ready for next command...\n")
#the file will stop when the Users press Cntrl+C or ^C
except KeyboardInterrupt:
    print("\n Shutting down...")
    pi.set_PWM_dutycycle(12, 0)
    pi.set_PWM_dutycycle(13, 0)
    pi.set_PWM_dutycycle(18, 0)
    pi.stop()
    GPIO.output(17, GPIO.LOW)
    GPIO.output(22, GPIO.LOW)
```