



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΡΟΜΠΟΤΙΚΟΣ ΣΕΡΒΙΤΟΡΟΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Διονύσιου Τσακαρισάνου

Επιβλέπων : Δρ. Καβαλλιεράτου Εργίνα

Μέλη εξεταστικής επιτροπής:

Δρ. Καβαλλιεράτου Εργίνα,

Δρ. Σταματάτος Ευστάθιος,

Δρ. Καπόρης Αλέξιος

Σάμος, Φεβρουάριος 2023

Η σελίδα αυτή είναι σκόπιμα λευκή.

Ευχαριστίες

Η εκπόνηση της παρούσας διπλωματικής εργασίας πραγματοποιήθηκε στο Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου και συγκεκριμένα στο Εργαστήριο Τεχνητής Νοημοσύνης, κατά τη διάρκεια του έτους 2022-23.

Κατά τη διάρκεια της εκπόνησης αυτής, σημαντικό ρόλο έπαιξε η καθοδήγηση της καθηγήτριάς μου, Δρ. Καβαλλιεράτου Εργίνας και την ευχαριστώ για την σημαντική βοήθεια που μου προσέφερε και την εμπιστοσύνη που μου έδειξε αναθέτοντας μου το έργο αυτό.

Επίσης θέλω να εκφράσω τις θερμές μου ευχαριστίες στους υποψήφιους διδάκτορες Νικόλαο Μάνο και Μάριο Βασιλείου για την βοήθεια που μου προσέφεραν στο εργαστήριο κατά την διεξαγωγή της έρευνας.

Τέλος, θα ήθελα να ευχαριστήσω την αγαπημένη μου οικογένεια για την ανεκτίμητη και απεριόριστη στήριξη που μου προσέφεραν, καθ' όλη τη διάρκεια των σπουδών μου.

© 2023

του/της

Διονύσιου Τσακαρισάνου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή	1
1.1	Ρομποτικός Σερβιτόρος.....	1
1.1.1	<i>Line following navigation</i>	1
1.1.2	<i>LiDAR navigation</i>	2
1.1.3	<i>Ανθρωποειδής μορφή και προστασία από το περιβάλλον</i>	2
1.2	Αντικείμενο διπλωματικής.....	2
1.3	Δομή της διπλωματικής	3
2	Λογισμικό.....	4
2.1	Λογισμικό Arduino	4
2.1.1	<i>Λογισμικό Arduino κίνησης</i>	4
2.1.2	<i>Λογισμικό Arduino αισθητήρων έκτακτης ανάγκης</i>	5
2.2	Raspberry Pi.....	6
2.2.1	<i>Flask server</i>	6
3	Σχεδίαση κατασκευής ρομποτικού σερβιτόρου.....	18
3.1	Αποθηκευτικός χώρος.....	18
3.2	Μεταλλικός σκελετός	20
3.3	Ηλεκτρονικά	21
3.3.1	<i>Σχεδιασμός ηλεκτρονικών πίνακα</i>	21
3.3.2	<i>Σχεδιασμός ηλεκτρονικών κεφαλιού</i>	25
4	Υλοποίηση κατασκευής	27
4.1	Σκελετός.....	27
4.2	Πλαστικά εξαρτήματα.....	28
4.2.1	<i>Κατασκευή πρόσοψης συρόμενων πορτών</i>	28
4.2.2	<i>Κατασκευή και εγκατάσταση πλευρικών τμημάτων</i>	30
4.3	Υλοποίηση ηλεκτρονικών.....	30
4.3.1	<i>Κινητήρες, πίνακας ηλεκτρονικών και μπαταρίες</i>	30
4.3.2	<i>Κατασκευή κεφαλιού</i>	33
5	Κίνηση ρομποτικού σερβιτόρου.....	37
5.1	Βηματικοί κινητήρες.....	37
5.2	Οδηγοί βηματικών κινητήρων	39
5.3	Αλγόριθμος κίνησης	40

6	Δοκιμές σε χώρο εστίασης	42
7	Συμπεράσματα και πιθανές βελτιώσεις	45
	Βιβλιογραφία	48
	Παράρτημα I Κώδικας Arduino αισθητήρων	49
	Παράρτημα II Κώδικας Arduino κίνησης	53
	Παράρτημα III Κώδικας Speech API	58

Λίστα Εικόνων

Εικόνα 1: Κύριο λογισμικό ρομποτικού σερβιτόρου που εκτελείται ως υπηρεσία συστήματος	6
Εικόνα 2: Πρόσωπο ρομποτικού σερβιτόρου.....	8
Εικόνα 3: Οθόνη επιλογής γλώσσας	9
Εικόνα 4: Οθόνη επιλογής τραπεζιού.....	10
Εικόνα 5: Οθόνη υποδοχής με το πρόσωπο του ρομποτικού σερβιτόρου.....	10
Εικόνα 6: Οθόνη καταλόγου προϊόντων και παραγγελίας	11
Εικόνα 7: Οθόνη επιβεβαίωσης ολοκλήρωσης της καταχώρισης παραγγελίας	12
Εικόνα 8: Οθόνη προεπισκόπησης παραγγελίας	12
Εικόνα 9: Διαχειριστική διεπαφή προσωπικού	14
Εικόνα 10: Οθόνη εισαγωγής PIN για πρόσβαση στο τεχνικό UI	14
Εικόνα 11: Διεπαφή τεχνικής διαχείρισης.....	15
Εικόνα 12: Διάγραμμα σχεσιακής βάσης δεδομένων.....	16
Εικόνα 13: Διάγραμμα αρχιτεκτονικής λογισμικού	17
Εικόνα 14: Αποθηκευτικός χώρος παραγγελιών	19
Εικόνα 15: Σχέδιο βασικών διαστάσεων μεταλλικού σκελετού	20
Εικόνα 16: Συνοπτικό διάγραμμα ηλεκτρονικών ρομποτικού σερβιτόρου.....	21
Εικόνα 17: Σχέδιο ρομποτικού σερβιτόρου.....	26
Εικόνα 18: Σκελετός ρομποτικού σερβιτόρου, προσαρμογή πλευρικών τμημάτων	28
Εικόνα 19: Μηχανισμός κλειδώματος πόρτας	29
Εικόνα 20: Οι μπαταρίες και η συνδεσμολογία τους	31
Εικόνα 21: Πίνακας ηλεκτρονικών ρομποτικού σερβιτόρου	32
Εικόνα 22: Φόρτιση μπαταριών και πάνελ πίνακα ηλεκτρονικών	33
Εικόνα 23: Εκτύπωση κεφαλιού.....	34
Εικόνα 24: Εσωτερικό κεφαλιού ρομποτικού σερβιτόρου.....	36
Εικόνα 25: Καλωδίωση διασύνδεσης πίνακα ηλεκτρονικών με κεφάλι	36
Εικόνα 26: Βηματικός κινητήρας με ενσωματωμένο μειωτήρα.....	38
Εικόνα 27: Οδηγός βηματικού κινητήρα TMC2209	40
Εικόνα 28: Φυλλάδιο οδηγιών χρήσης ρομποτικού σερβιτόρου.....	43
Εικόνα 29: Δοκιμή ρομποτικού σερβιτόρου σε χώρο εστίασης.....	44

Λίστα Σχημάτων

Σχήμα 1: Σχηματικό διάγραμμα του πίνακα ηλεκτρονικών	24
Σχήμα 2: Διάγραμμα ροής αλγορίθμου βηματισμού.....	41

Λίστα Πινάκων

Πίνακας 1: Εντολές που υποστηρίζονται από την Arduino κίνησης.....	5
Πίνακας 2: Εντολές που υποστηρίζονται από την Arduino αισθητήρων έκτακτης ανάγκης.....	5
Πίνακας 3: Λίστα βασικού ηλεκτρονικού εξοπλισμού ρομποτικού σερβιτόρου	35

Ακρωνύμια

UI	User Interface
API	Application Programming Interface
Estop	Emergency stop
JSON	JavaScript Object Notation
TTS	Text-To-Speech
DB	Database
CNC	Computerized Numerical Control
DC	Direct Current
USB	Universal Serial Bus
RJ45	Registered Jack 45
EDA	Electronic Design Automation
RAM	Random Access Memory
SoC	System on Chip
GPIO	General-Purpose Input/Output
DSI	Display Serial Interface
RPI	Raspberry Pi
UTP	Unshielded Twisted Pair
AWG	American Wire Gauge
SQL	Structured Query Language
PIN	Personal Identification Number
UUID	Universally Unique Identifier
WASM	Web Assembly
PLA	Polylactic Acid
PETG	Polyethylene terephthalate glycol
ABS	Acrylonitrile butadiene styrene

Περίληψη

Σε έναν κόσμο, όπου η ρομποτική διαρκώς κάνει την παρουσία της αισθητή σε διάφορους τομείς της καθημερινότητας, μέσα από την αυτοματοποίηση εργασιών και την γενικότερη αρωγή της στην ανθρωπότητα, είναι φυσικό πως θα έλθει να συμπληρώσει, αν όχι αντικαταστήσει, ένα από τα παλαιότερα και βασικότερα επαγγέλματα όπως αυτό του σερβιτόρου. Ο χώρος της ρομποτικής που στοχεύει στην ανάπτυξη ρομποτικών προσωπικών εξυπηρέτησης έχει ήδη σημειώσει πρόοδο, με τους πρώτους ρομποτικούς σερβιτόρους να κάνουν την εμφάνιση τους σε εστιατόρια σε όλο τον κόσμο. Παρ' όλα αυτά υπάρχουν ακόμη διάφορα ζητήματα που εμποδίζουν την πλήρη υιοθέτηση των ρομποτικών σερβιτόρων, τα οποία χρήζουν επίλυσης. Η πρόταση που παρουσιάζεται στη παρούσα διπλωματική εργασία, αφορά ένα ρομπότ σερβιτόρο, το οποίο προσπαθεί να δώσει λύση σε κάποια από αυτά τα ζητήματα. Πρόκειται για μία κατασκευή στην οποία χρησιμοποιήθηκε ως επί το πλείστον τρισδιάστατη εκτύπωση, αλλά και ορισμένα μεταλλικά και ακρυλικά στοιχεία για την υλοποίηση της, όπως επίσης και ηλεκτρονικός εξοπλισμός και λογισμικό που της δίνουν «ζωή». Ταυτόχρονα δόθηκε μεγάλη έμφαση σε συστήματα που στοχεύουν στην ασφάλεια χρήσης. Ο προτεινόμενος ρομποτικός σερβιτόρος είναι φιλικός, καθώς έχει ανθρώπινα χαρακτηριστικά και είναι σε θέση να λαμβάνει παραγγελίες από την οθόνη αφής του, να τις διαβιβάζει στο προσωπικό του χώρου εστίασης και έπειτα να τις παραδίδει στο επιθυμητό τραπέζι, κατ' εντολή του προσωπικού. Αυτό το επιτυγχάνει μέσω ενός συστήματος καταχώρισης παραγγελιών και μέσω προγραμματισμένων κινήσεων που περιγράφουν την διαδρομή κάθε τραπεζιού. Στη συνέχεια ο ρομποτικός σερβιτόρος δοκιμάστηκε σε πραγματικό χώρο εστίασης, όπου έγινε εξαγωγή ποικίλων αποτελεσμάτων χρήσης, αλλά και συμπερασμάτων. Αυτά με την σειρά τους οδήγησαν σε βελτιώσεις του συστήματος, αλλά και τροφοδότησαν σκέψη για κάποια πιθανώς αναβαθμισμένη μελλοντική έκδοση.

Λέξεις Κλειδιά: ρομποτικός σερβιτόρος, ρομπότ σερβιτόρος, εστίαση, σύστημα παραγγελιών, βάση δεδομένων, τρισδιάστατη εκτύπωση

Abstract

In a world, where robotics field continually makes its presence known in various fields of everyday life, be it through job automation or generally by helping humanity, it's only natural that it will try to supplement, if not replace, one of the oldest and most basic professions such as the waiter's. The branch of robotics that aims at the development of robotic waiting staff has already made progress, which can be observed by the appearance of robot waiters in numerous restaurants across the globe. Nevertheless, some issues still exist that stand in the way of a full adoption of such robot waiters, that require solving. The system proposed in the current thesis, is about a robot waiter, that attempts to solve some of these issues. It's about an implementation, which was constructed mostly using 3D printed, metal and acrylic parts, paired with electronics and software used to give it "life". At the same time, emphasis was given in the development of subsystems that aim to offer safety during operation. The proposed robot waiter is friendly, as a result of its human characteristics, and is able to receive orders from its touchscreen, relay them to the restaurant's staff and then deliver them on command. This is achieved through an order processing system and by using pre-programmed moves that describe each table's route. The robot waiter was then field tested in a real-life restaurant, which led to the extraction of several results and conclusions. Those then assisted in several improvements of the system and also provided food for thought concerning a possible future upgraded version.

Keywords: *robot waiter, order system, catering, database, 3D printing*

1

Εισαγωγή

1.1 Ρομποτικός Σερβιτόρος

Το επάγγελμα του σερβιτόρου αποτελεί ένα από τα παλαιότερα επαγγέλματα της σύγχρονης ανθρώπινης ιστορίας. Τα καθήκοντά του ποικίλουν, από το να δέχεται και να εκτελεί παραγγελίες, μέχρι και να είναι φιλικός και εξυπηρετικός προς τον πελάτη. Στόχος του πάντοτε αποτελεί η όσον το δυνατόν βελτιστοποίηση της εμπειρίας του πελάτη με τον χώρο εστίασης.

Τα τελευταία χρόνια έχουν γίνει πολλές προσπάθειες στην εξομοίωση του ρόλου του σερβιτόρου, με χρήση ρομποτικών συστημάτων. Οι προσπάθειες αυτές γίνονται με σκοπό την συμπλήρωση του επαγγέλματος του σερβιτόρου, ώστε να διευκολύνεται η εκτέλεση των καθηκόντων του. Όπως αναφέρουν και οι Juan Miguel Garcia-Haro *et al.* (2020) ολοένα και περισσότερες εταιρίες επενδύουν στον τομέα της εστίασης, με αποτέλεσμα την δημιουργία ευκαιριών για την ρομποτική να ανθίσει στον τομέα αυτό.

Παρόλα αυτά, όπως άλλωστε και σε κάθε νέο πρόβλημα που η ρομποτική καλείται να δώσει λύση, η εξομοίωση του ρόλου του σερβιτόρου αντιμετωπίζει μια πληθώρα προβλημάτων τα οποία προσπαθούν να αντιμετωπίσουν διάφορες υπάρχουσες προτάσεις.

1.1.1 Line following navigation

Μία από τις μεθόδους πλοήγησης που εφαρμόζεται στον χώρο των ρομποτικών σερβιτόρων για την πλοήγηση τους στα τραπέζια, είναι το line following (ακολουθία γραμμής). Σύμφωνα με τους Thanh *et al.* (2019) μπορούν να χρησιμοποιηθούν αισθητήρες φωτεινότητας για την ανίχνευση και στη συνέχεια ακολουθία μαύρης γραμμής, που είναι εγκατεστημένη στο

δάπεδο του χώρου εστίασης, και αναπαριστά μονοπάτια τα οποία χρησιμοποιεί το ρομπότ για τη πλοήγησή του. Ένα μειονέκτημα της μεθόδου αυτής, είναι ότι συνεπάγεται την εγκατάσταση γραμμής στο δάπεδο του χώρου εστίασης, θυσιάζοντας την αισθητική του και κάνοντας δύσκολη την τροποποίηση ή προσθήκη νέων διαδρομών.

1.1.2 LiDAR navigation

Αρκετοί ρομποτικοί σερβιτόροι χρησιμοποιούν τεχνολογία LiDAR (Light Detection And Ranging), ώστε να εντοπίζουν την θέση τους και να προηγούνται στον χώρο, όπως για παράδειγμα στο σύστημα που προτείνουν οι Wan *et al.* (2019). Έτσι, επιτυγχάνεται μεγάλη ευελιξία στην κίνηση του ρομπότ, καθώς και καθίσταται δυνατή η δυναμική κίνηση του στον χώρο εστίασης. Το μειονέκτημα αυτής της προσέγγισης αποτελεί το υψηλό κόστος των αισθητήρων LiDAR, όπως επίσης και η .

1.1.3 Ανθρωποειδής μορφή και προστασία από το περιβάλλον

Όπως ήδη αναφέρθηκε, ένας σερβιτόρος οφείλει να είναι φιλικός προς τον πελάτη. Για τον λόγο αυτό, και ένας ρομποτικός σερβιτόρος καλείται να μιμηθεί την φιλική εμφάνιση ενός πραγματικού σερβιτόρου. Έτσι, πολλοί ρομποτικοί σερβιτόροι είναι σχεδιασμένοι με ανθρωποειδή χαρακτηριστικά, για να επιτυγχάνεται η θετική εντύπωση του πελάτη. Όμως ένα ανθρωποειδές ρομπότ, συνήθως δεν είναι σχεδιασμένο γνώμονα την εργασία σε αντίξοες συνθήκες, όπως θα ήταν σχεδιασμένο ένα βιομηχανικό ρομπότ. Σύμφωνα και με τους Juan Miguel Garcia-Haro *et al.* (2020), ένας ρομποτικός σερβιτόρος πρέπει να είναι ανθεκτικός στο περιβάλλον ενός εστιατορίου, που μπορεί να περιλαμβάνει διάφορους παράγοντες φθοράς, όπως λάδια, υψηλές θερμοκρασίες, ή υγρασία.

1.2 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική έχει ως στόχο την περιγραφή και ανάλυση του σχεδιασμού και της υλοποίησης ενός προτεινόμενου συστήματος ρομποτικού σερβιτόρου, καθώς και πως αυτό εμπνεύστηκε ως προσπάθεια βελτίωσης ήδη υπάρχοντων λύσεων του χώρου. Επίσης θα προσπαθήσει να δώσει στον αναγνώστη μια εικόνα των πιθανών ζητημάτων που παρουσιάζονται κατά τη διάρκεια των διαδικασιών αυτών, μέσω αποτελεσμάτων που παράχθηκαν από διάφορες δοκιμές, οι οποίες πραγματοποιήθηκαν κατά την διάρκεια, αλλά και μετά το πέρας της υλοποίησης του πρωτότυπου. Ακόμη θα του παρουσιάσει τα συμπεράσματα που εξήχθησαν και θα προτείνει πιθανές βελτιώσεις για μια πιθανή μελλοντική υλοποίηση.

Τα προβλήματα που θα επιχειρήσει να δώσει λύση αναφέρθηκαν στο προηγούμενο κεφάλαιο. Ένα από αυτά αφορά την ανάγκη εγκατάστασης μαύρης γραμμής για την πλοήγηση του ρομπότ στον χώρο εστίασης. Το προτεινόμενο σύστημα αντιμετωπίζει το ζήτημα αυτό, αντικαθιστώντας τη μέθοδο ακολουθίας μαύρης γραμμής, με χρήση προγραμματισμένων κινήσεων για την κίνηση του ρομποτικού σερβιτόρου από και προς τα

τραπέζια του καταστήματος, κάνοντας χρήση βηματικών κινητήρων για την αποφυγή ολίσθησης κινήσεων και συνεπώς παρέκκλισης από τις προκαθορισμένες διαδρομές.

Ακόμη ένα πρόβλημα που έρχεται να αντιμετωπίσει το προτεινόμενο σύστημα είναι η βελτίωση της αισθητικής και ανθρωποειδούς εμφάνισης ενός ρομποτικού σερβιτόρου. Ένας ρομποτικός σερβιτόρος, όπως και ένας πραγματικός σερβιτόρος, πρέπει να είναι φιλικός για την βελτίωση της εμπειρίας του πελάτη κατά την διάρκεια της αλληλεπίδρασής του. Η λύση που προτείνεται είναι η σχεδίαση του συστήματος να περιλαμβάνει ανθρωποειδή χαρακτηριστικά, τα οποία δίνουν στον ρομποτικό σερβιτόρο μια ανθρωποειδή εικόνα, χωρίς όμως να προσπαθούν να αποκρύψουν την ρομποτική του φύση ή να αποτρέπουν την ομαλή εκτέλεση των λειτουργιών του. Παραδείγματα των ανθρώπινων χαρακτηριστικών αυτών είναι το σώμα με τα χέρια, καθώς και το κεφάλι με μάτια και στόμα που απεικονίζονται και κινούνται στην οθόνη, ώστε να αναπαριστούν το πρόσωπο του σερβιτόρου.

Τέλος, οι περισσότεροι ρομποτικοί σερβιτόροι αποτελούνται από εξαρτήματα, τα οποία είναι δύσκολο να αποκτηθούν ή να παραχθούν. Όμως, η κατασκευή του παρόντος συστήματος αποτελείται σχεδόν εξ ολοκλήρου από τρισδιάστατα εκτυπωμένα εξαρτήματα. Το γεγονός αυτό επιτρέπει σε κάποιον που επιθυμεί να αναπαραγάγει την κατασκευή αυτή, να το πραγματοποιήσει με την βοήθεια ενός τρισδιάστατου εκτυπωτή, με σχετικά χαμηλό κόστος.

1.3 Δομή της διπλωματικής

Το Κεφάλαιο 1 (παρόν κεφάλαιο) αποτελεί εισαγωγή της διπλωματικής εργασίας, ενώ αναφέρει τον χώρο που αφορά, τον στόχο, και έπειτα παρουσιάζεται η πρόταση συστήματος της παρούσας διπλωματικής εργασίας, με έμφαση στις βελτιώσεις που φέρνει και κλείνοντας με την περιγραφή της δομής της. Έπειτα στο Κεφάλαιο 2 αναλύεται το λογισμικό που αναπτύχθηκε για το σύστημα, όπως και η αρχιτεκτονική και ο τρόπος λειτουργίας του. Στο Κεφάλαιο 3 περιγράφεται η σχεδίαση του υλικού του συστήματος και αναλύονται τα βασικότερα επιμέρους τμήματα του. Στη συνέχεια στο Κεφάλαιο 4 περιγράφεται η υλοποίηση της σχεδίασης του συστήματος και των επιμέρους κομματιών του. Το Κεφάλαιο 5 αφορά την κίνηση. Στο Κεφάλαιο 6 περιγράφονται οι δοκιμές του συστήματος που έγιναν σε χώρο εστίασης, όπως και αξιολογούνται τα αποτελέσματα χρήσης που προέκυψαν. Τέλος, στο Κεφάλαιο 7 περιγράφονται συμπεράσματα και πιθανές βελτιώσεις του συστήματος.

2

Λογισμικό

2.1 Λογισμικό Arduino

Ο ρομποτικός σερβιτόρος κάνει χρήση δύο πλακετών μικροελεγκτών Arduino UNO R3 για την εκτέλεση χαμηλού επιπέδου λειτουργιών, όπως η κίνηση του και η συλλογή δεδομένων από το περιβάλλον του. Παρακάτω αναλύεται το λογισμικό το οποίο εκτελείται και ελέγχει τις δύο αυτές Arduino, καθώς και περιγράφεται ο τρόπος με τον οποίο επιτυγχάνουν την εκτέλεση των διαφόρων λειτουργιών τους.

2.1.1 Λογισμικό Arduino κίνησης

Η πρώτη από τις δύο Arduino που προαναφέρθηκαν, αναλαμβάνει την κίνηση του ρομπότ. Για τον σκοπό αυτό αναπτύχθηκε λογισμικό το οποίο «ζει» στην πλακέτα αυτή και εκτελεί τις εντολές κίνησης στο χαμηλότερο επίπεδο του υλικού. Οι εντολές που υποστηρίζονται από το λογισμικό της Arduino κίνησης, και κατ' επέκταση υλοποιούν το πρωτόκολλο επικοινωνίας, αφορούν τα διαφορετικά είδη κινήσεων που καλούνται να εκτελέσουν οι δύο βηματικοί κινητήρες του ρομπότ, καθώς και την αποστολή πληροφοριών για αναγνώριση από το σύστημα (βλ. Πίνακας 1).

Το λογισμικό της Arduino κίνησης αποτελείται επίσης και από αλγόριθμο, ο οποίος υλοποιεί την διαδικασία βηματισμού των κινητήρων. Αυτό επιτυγχάνεται μέσω χειρισμού των ψηφιακών εξόδων της πλακέτας, οι οποίες με την σειρά τους στέλνουν σήμα στους οδηγούς των βηματικών κινητήρων. Ο αλγόριθμος κίνησης θα εξηγηθεί πιο αναλυτικά στο κεφάλαιο της κίνησης.

Πίνακας 1: Εντολές που υποστηρίζονται από την Arduino κίνησης.

Εντολή	Παράμετροι	Περιγραφή
move_both	Αριθμός βημάτων, κατεύθυνση	Κίνηση και των δύο κινητήρων προς την επιλεγμένη κατεύθυνση για καθορισμένο αριθμό βημάτων.
move_left	Αριθμός βημάτων, κατεύθυνση	Κίνηση του αριστερού κινητήρα προς την επιλεγμένη κατεύθυνση για καθορισμένο αριθμό βημάτων.
move_right	Αριθμός βημάτων, κατεύθυνση	Κίνηση του δεξιού κινητήρα προς την επιλεγμένη κατεύθυνση για καθορισμένο αριθμό βημάτων.
identify	Καμία	Αποστολή πληροφοριών πλακέτας για αναγνώριση από το σύστημα.

2.1.2 Λογισμικό Arduino αισθητήρων έκτακτης ανάγκης

Εκτός από την Arduino κίνησης, υπάρχει και μια δεύτερη Arduino ή οποία είναι υπεύθυνη για συλλογή δεδομένων. Όπως και η πρώτη, η πλακέτα αυτή περιέχει λογισμικό που υλοποιεί ένα πρωτόκολλο επικοινωνίας, το οποίο χρησιμεύει κυρίως για την μεταφορά δεδομένων των αισθητήρων του ρομποτικού σερβιτόρου στον κεντρικό μικροεπεξεργαστή του. Οι εντολές που υποστηρίζονται από το πρωτόκολλο αυτό αποτελούνται από μία εντολή αιτήματος για μετάδοση δεδομένων αισθητήρων, όπως και από μία εντολή απομακρυσμένης ενεργοποίησης του συστήματος έκτακτης ανάγκης. Για βοήθεια στην αναγνώριση της πλακέτας από το σύστημα, υποστηρίζεται και η εντολή identify, όπως και στην Arduino κίνησης. Είναι σημαντικό να αναφερθεί ότι για το σύστημα έκτακτης ανάγκης (ESTOP) χρησιμοποιήθηκαν hardware interrupts τα οποία εξασφαλίζουν την σχεδόν ακαριαία εμπλοκή του.

Πίνακας 2: Εντολές που υποστηρίζονται από την Arduino αισθητήρων έκτακτης ανάγκης

Εντολή	Παράμετροι	Περιγραφή
update	Καμία	Λήψη δεδομένων (απόσταση αισθητήρων υπερήχων, τάση μπαταριών και κατάσταση έκτακτης ανάγκης).
estop	Καμία	Ενεργοποίηση κατάστασης έκτακτης ανάγκης (χρησιμοποιείται κυρίως για απομακρυσμένη ενεργοποίηση του συστήματος έκτακτης ανάγκης).
identify	Καμία	Αποστολή πληροφοριών πλακέτας για αναγνώριση από το σύστημα.

2.2 Raspberry Pi

Για την εκτέλεση των έργων του, ο ρομποτικός σερβιτόρος χρειάζεται ένα κεντρικό υπολογιστικό σύστημα. Τον ρόλο αυτό, έρχεται να καλύψει ένας μικροεπεξεργαστής Raspberry Pi Model B 4GB. Ως λειτουργικό σύστημα για τον μικροεπεξεργαστή αυτό επιλέχθηκε μία διανομή Linux και πιο συγκεκριμένα Debian, η οποία είναι διαμορφωμένη για εκτέλεση από το υλικό αυτό, και ονομάζεται Raspbian. Κατά την εκκίνηση του συστήματος λαμβάνουν μέρος κάποιες βασικές διαδικασίες για την εξασφάλιση της αδιάκοπης λειτουργίας του λογισμικού του ρομποτικού σερβιτόρου.

Μία από αυτές τις διαδικασίες αποτελεί η εκκίνηση ενός λογισμικού εξυπηρετητή, το οποίο δρα επίσης ως κύριο λογισμικό ελέγχου, σε μορφή υπηρεσίας συστήματος. Αυτό επιτυγχάνεται χάρη στην εγγραφή του λογισμικού αυτού ως υπηρεσία μέσω του εργαλείου διαχείρισης υπηρεσιών των Linux με το όνομα systemd (βλ. Εικόνα 1: Κύριο λογισμικό ρομποτικού σερβιτόρου που εκτελείται ως υπηρεσία συστήματος).

```
pi@RobotWaiter:~$ systemctl status robotwaiterserver.service
● robotwaiterserver.service - Robot Waiter Server
   Loaded: loaded (/etc/systemd/system/robotwaiterserver.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-12-14 18:00:21 EET; 1min 20s ago
     Main PID: 1720 (sudo)
        Tasks: 3 (Limit: 4915)
           CPU: 1.328s
      CGroup: /system.slice/robotwaiterserver.service
              └─1720 /usr/bin/sudo /usr/bin/python /home/pi/user_interface/frontend/run.py
                 └─1721 /usr/bin/python /home/pi/user_interface/frontend/run.py

Dec 14 18:00:30 RobotWaiter sudo[1721]: | $$ | $$| $$$$$$| $$$$$$| $$$$$$| | $$ | $$| \ $$| $$ | $$ /$$$$$ | $$ | $$$$$$| $$ | $$
Dec 14 18:00:30 RobotWaiter sudo[1721]: |_/ |_/ \_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/ |_/
Dec 14 18:00:30 RobotWaiter sudo[1721]:
Dec 14 18:00:30 RobotWaiter sudo[1721]: Dionysios Tsakarisanos 2022
Dec 14 18:00:30 RobotWaiter sudo[1721]:
Dec 14 18:00:30 RobotWaiter sudo[1721]: * Serving Flask app "robot_waiter_app" (lazy loading)
Dec 14 18:00:30 RobotWaiter sudo[1721]: * Environment: production
Dec 14 18:00:30 RobotWaiter sudo[1721]: WARNING: This is a development server. Do not use it in a production deployment.
Dec 14 18:00:30 RobotWaiter sudo[1721]: Use a production WSGI server instead.
Dec 14 18:00:30 RobotWaiter sudo[1721]: * Debug mode: off
pi@RobotWaiter:~$
```

Εικόνα 1: Κύριο λογισμικό ρομποτικού σερβιτόρου που εκτελείται ως υπηρεσία συστήματος

Εκτός από την υπηρεσία συστήματος, κατά την διάρκεια της εκκίνησης, δημιουργείται επίσης και ένα παράθυρο φιλομετρητή Google Chrome το οποίο ξεκινά με εντολή που περιέχει ειδικές παραμέτρους.

2.2.1 Flask server

Το κυρίως λογισμικό του ρομποτικού σερβιτόρου εκτελείται στο Raspberry Pi. Αναπτύχθηκε με χρήση της γλώσσας προγραμματισμού Python 3 και πιο συγκεκριμένα βασίστηκε στο Flask. Το Flask αποτελεί ένα πλαίσιο ανάπτυξης λογισμικού για εφαρμογές ιστού (web framework), για χρήση με την γλώσσα προγραμματισμού Python, στην οποία είναι και

αναπτυγμένο. Το πρόγραμμα που αναπτύχθηκε χωρίζεται σε δύο βασικά κομμάτια, τα οποία είναι άμεσα συνδεδεμένα μεταξύ τους και που θα αναλύσουμε παρακάτω.

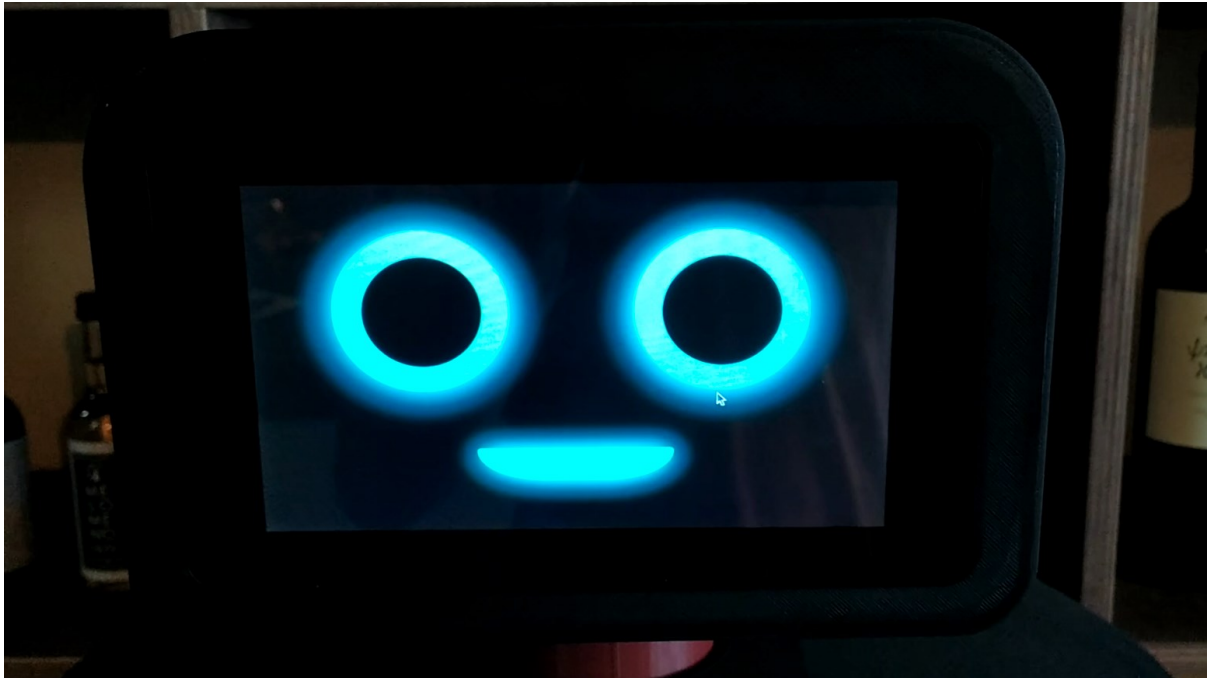
2.2.1.1 RobotWaiter class

Το πρώτο κομμάτι του προγράμματος Flask αποτελεί ο κώδικας διαχείρισης του υλικού του ρομποτικού σερβιτόρου. Ο κώδικας αυτός εκτελείται σε ξεχωριστό νήμα (thread) το οποίο δημιουργείται κατά την εκκίνηση του κύριου νήματος του εξυπηρετητή Flask. Στόχος του είναι η υλοποίηση του πρωτοκόλλου επικοινωνίας με το υλικό του ρομπότ, δηλαδή τις δύο Arduino που αναφέρθηκαν σε προηγούμενο κεφάλαιο, από την πλευρά του Raspberry Pi. Αυτό είναι απαραίτητο για την δημιουργία ενός επιπέδου αφαιρετικότητας για τη διαβίβαση μηνυμάτων από και προς το υλικό, όπως θα γινόταν αν όλο το υλικό του ρομπότ αποτελούσε μία ενιαία νοητή οντότητα, σε μορφή κλάσης της python με όνομα RobotWaiter. Με τον τρόπο αυτό γίνεται πολύ πιο εύκολη η ενσωμάτωση του υλικού στον κώδικα. Πράγματι, η κλάση αυτή χρησιμοποιείται μετέπειτα σε διάφορα σημεία του κύριου προγράμματος Flask για τον έλεγχο του ρομπότ, όπως για παράδειγμα την συλλογή δεδομένων αισθητήρων και την εκτέλεση κινήσεων.

2.2.1.2 Σύστημα παραγγελιών

Άλλο ένα τμήμα του εξυπηρετητή Flask αποτελεί η φιλοξενία του frontend, το οποίο αποτελεί την διεπαφή χρήστη του ρομποτικού σερβιτόρου και του συστήματος παραγγελιών. Το frontend χωρίζεται σε τρία επιμέρους κομμάτια. Το UI του συστήματος παραγγελιών που προορίζεται για χρήση από τους πελάτες, το UI του συστήματος παραγγελιών που προορίζεται για χρήση από το προσωπικό του χώρου εστίασης, και το διαχειριστικό UI του ρομπότ που χρησιμοποιείται για απευθείας πρόσβαση σε πιο τεχνικές λειτουργίες του.

Το UI των πελατών είναι σχεδιασμένο με στόχο την καθοδήγηση τους κατά την δημιουργία και καταχώριση παραγγελιών. Είναι ένα από τα σημαντικότερα υποσυστήματα του ρομπότ καθώς αποτελεί το μόνο λογισμικό με το οποίο αλληλοεπιδρά ο πελάτης με το ρομποτικό σερβιτόρο. Συνεπώς παίζει καθοριστικό ρόλο στην εμπειρία του πελάτη και στην γενικότερη εικόνα του ρομπότ. Για τον λόγο αυτό κατά τον σχεδιασμό του επιλέχθηκε η υλοποίηση οθόνης που αποτελείται από ανθρωποειδή χαρακτηριστικά, ώστε να αποδίδει στον ρομποτικό σερβιτόρο ένα πρόσωπο και να είναι φιλικό προς τον χρήστη. Πιο συγκεκριμένα σχεδιάστηκε ένα πρόσωπο με στόμα και μάτια, τα οποία είναι γαλάζιου χρώματος που διαχέεται σταδιακά σε μαύρο φόντο, για να δίνεται η αίσθηση του βάθους, και να μην ξεχωρίζει από το υπόλοιπο κεφάλι του σερβιτόρου (βλ. Εικόνα 2). Στο πρόσωπο αυτό δόθηκε μια πιο «ζωντανή» εμφάνιση χάρη στη χρήση animation το οποίο ανοιγοκλείνει τα μάτια σε τυχαία χρονικά διαστήματα, καθώς και αλγόριθμος που ανοιγοκλείνει το στόμα κατά την διάρκεια της ομιλίας, η οποία θα αναλυθεί σε επόμενο κεφάλαιο.



Εικόνα 2: Πρόσωπο ρομποτικού σερβιτόρου

Όπως προαναφέρθηκε, κατά τον σχεδιασμό της διεπαφής των πελατών δόθηκε μεγάλη βάση στην αισθητική και την ευχρηστία της ίδιας. Για την διευκόλυνση της διαδικασίας καταχώρισης παραγγελίας, δημιουργήθηκε μια σειρά βημάτων από τα οποία ο πελάτης καλείται να περάσει, ώστε να καταχωρίσει την παραγγελία του. Όμως, θα πρέπει να λάβουμε υπόψιν ότι ο πελάτης ίσως να μην είναι σε θέση να κατανοήσει τις ενέργειες που πρέπει να εκτελέσει σε κάθε ένα από αυτά τα βήματα, καθώς μπορεί να μην κατέχει τεχνικές γνώσεις ή να χρησιμοποιεί το σύστημα για πρώτη φορά. Για τον λόγο αυτό, η σειρά των βημάτων σχεδιάστηκε σε μορφή οδηγού, ο οποίος ακολουθεί συγκεκριμένο σενάριο, και καθ' όλη την διάρκειά του ο ρομποτικός σερβιτόρος δίνει με ευγενικό τρόπο ξεκάθαρες οδηγίες για την ολοκλήρωση του κάθε βήματος. Αυτό επιτυγχάνεται με συγχρονισμένη χρήση ομιλίας και γραπτών μηνυμάτων που εμφανίζονται στην οθόνη του ρομπότ, σε μορφή σύννεφων κειμένου με πεπερασμένη χρονική διάρκεια εμφάνισης. Από πλευράς λογισμικού τα βήματα αυτά υλοποιήθηκαν με την βοήθεια ενός state machine το οποίο υλοποιήθηκε στην πλευρά του frontend με χρήση της γλώσσας προγραμματισμού JavaScript.

Το πρώτο από τα βήματα, αποτελεί η επιλογή γλώσσας διεπαφής. Στο βήμα αυτό ο ρομποτικός σερβιτόρος παρουσιάζει στον χρήστη μια οθόνη με τις υποστηριζόμενες γλώσσες, κάνοντας ξεκάθαρο το γεγονός ότι η οθόνη αποτελεί οθόνη αφής και τον κύριο τρόπο αλληλεπίδρασης. Στη συνέχεια ο χρήστης επιλέγει την γλώσσα του και ο κώδικας του

frontend επικοινωνεί με το API του εξυπηρετητή Flask ώστε να τεθεί η κατάλληλη γλώσσα και να αιτηθεί η σελίδα με την ανάλογη μετάφραση. Έτσι επίσης ξεκινά και μια νέα συνεδρία (session) χρήσης του συστήματος παραγγελιών. Η επιλεγμένη γλώσσα αποθηκεύεται σαν μεταβλητή στον κώδικα JavaScript του frontend, κατά την φόρτωσή του. Έτσι είναι δυνατή η φόρτωση των κατάλληλων μεταφράσεων για όλα τα στοιχεία του UI (π.χ. κατάλογος προϊόντων, ομιλία).



Εικόνα 3: Οθόνη επιλογής γλώσσας

Με την επιλογή γλώσσας και την μετάβαση στην σελίδα παραγγελίας, εμφανίζεται το πρόσωπο του ρομποτικού σερβιτόρου το οποίο και συστήνεται στον πελάτη στην γλώσσα που επιλέχθηκε. Εδώ παρουσιάζεται και η πρώτη οδηγία, η οποία προτρέπει τον πελάτη να ξεκινήσει την διαδικασία παραγγελίας, πατώντας πάνω στην οθόνη. Με την εκτέλεση της ενέργειας αυτής, παρουσιάζεται η οθόνη επιλογής τραπεζιού στην οποία ο πελάτης καλείται να επιλέξει το τραπέζι του για μελλοντική χρήση από το σύστημα. Τα τραπέζια παρουσιάζονται σε μορφή σχεδίου κάτοψης του χώρου του καταστήματος, όπως και η θέση και τον προσανατολισμό του ρομποτικού σερβιτόρου, ώστε να μπορεί ο χρήστης εύκολα να αναγνωρίσει το τραπέζι του και να το επιλέξει πατώντας επάνω του. Με την επιλογή του τραπεζιού στην οθόνη, δημιουργείται ένα κενό αντικείμενο παραγγελίας στο οποίο και αποθηκεύεται ο αριθμός του τραπεζιού.

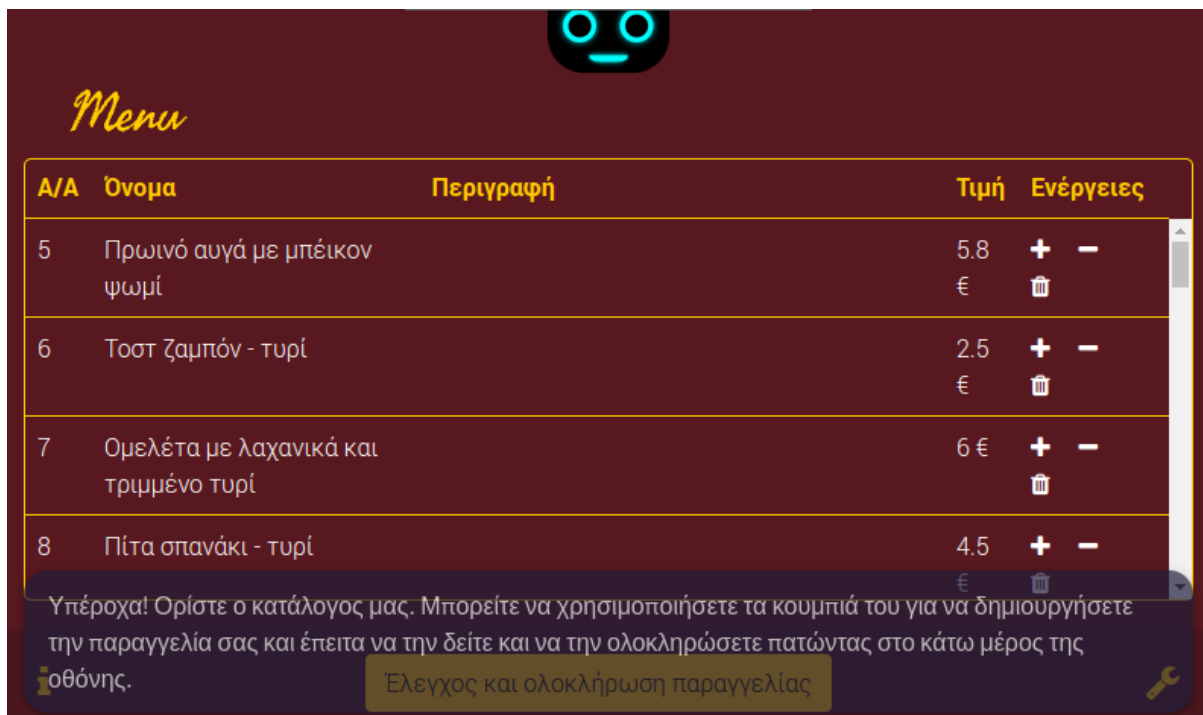


Εικόνα 4: Οθόνη επιλογής τραπεζιού



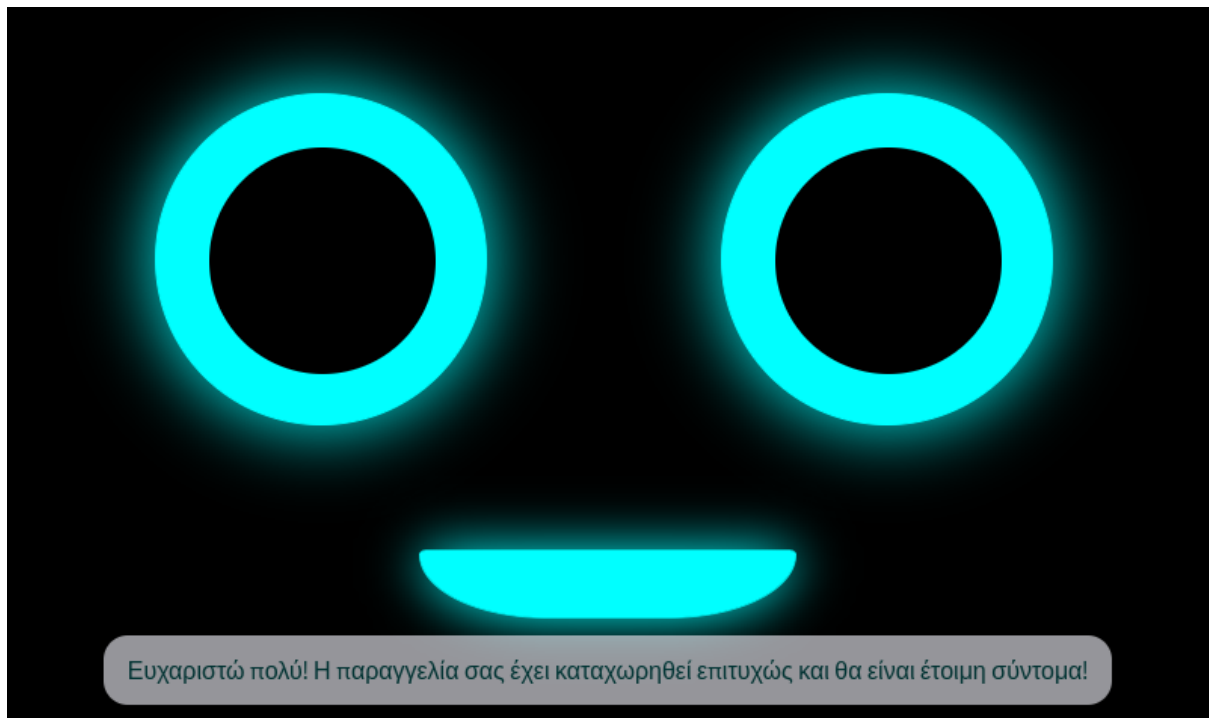
Εικόνα 5: Οθόνη υποδοχής με το πρόσωπο του ρομποτικού σερβιτόρου

Έπειτα το σύστημα εκτελεί αίτημα στο API για τον κατάλογο προϊόντων του καταστήματος και όταν τον λάβει τον εμφανίζει σε αντίστοιχη οθόνη. Η οθόνη αυτή παρέχει επίσης στον χρήστη στοιχεία ελέγχου, για την επεξεργασία των προϊόντων της παραγγελίας του, δίνοντας για ακόμα μια φορά σαφείς οδηγίες για την χρήση τους. Τα στοιχεία αυτά αποτελούνται από πλήκτρα προσθήκης, αφαίρεσης και αλλαγής ποσότητας, για κάθε προϊόν του καταλόγου καθώς και κεντρικό κουμπί για συνέχεια στο επόμενο βήμα. Ο κατάλογος βρίσκεται σε περιοχή την οποία ο χρήστης μπορεί να κυλίσει για τον δει ολόκληρο σε περίπτωση που αυτός δεν χωράει στην οθόνη.

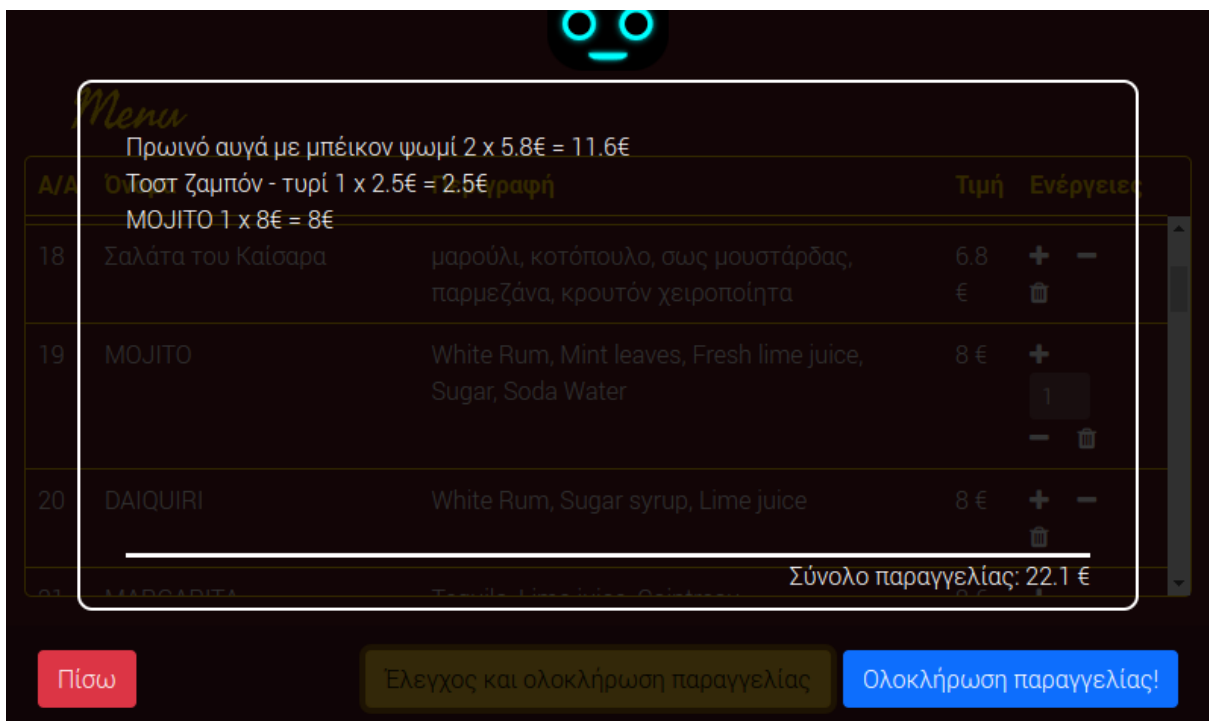


Εικόνα 6: Οθόνη καταλόγου προϊόντων και παραγγελίας

Σε συνέχεια του προηγούμενου βήματος ο χρήστης επιλέγει να δει την παραγγελία του. Έτσι εμφανίζεται μια νέα οθόνη στην οποία φαίνεται αναλυτικά η παραγγελία που έχει δημιουργηθεί και δίνεται η επιλογή της οριστικοποίησης της, ή επιστροφής στην προηγούμενη οθόνη για τροποποίηση της. Ακόμη υπολογίζεται και εμφανίζεται το συνολικό πόσο της παραγγελίας. Έπειτα από την οριστικοποίηση της παραγγελίας, για μια τελευταία φορά εμφανίζεται το πρόσωπο του ρομποτικού σερβιτόρου, που ενημερώνει τον χρήστη ότι η παραγγελία του καταχωρήθηκε επιτυχώς, ή ότι υπήρξε σφάλμα σε περίπτωση αποτυχίας λόγω τεχνικού προβλήματος του συστήματος παραγγελιών. Τέλος, το frontend στέλνει στον Flask server αίτημα καταχώρησης του αντικειμένου της παραγγελίας, ώστε να καταχωρηθεί στην βάση δεδομένων.



Εικόνα 7: Οθόνη επιβεβαίωσης ολοκλήρωσης της καταχώρισης παραγγελίας



Εικόνα 8: Οθόνη προεπισκόπησης παραγγελίας

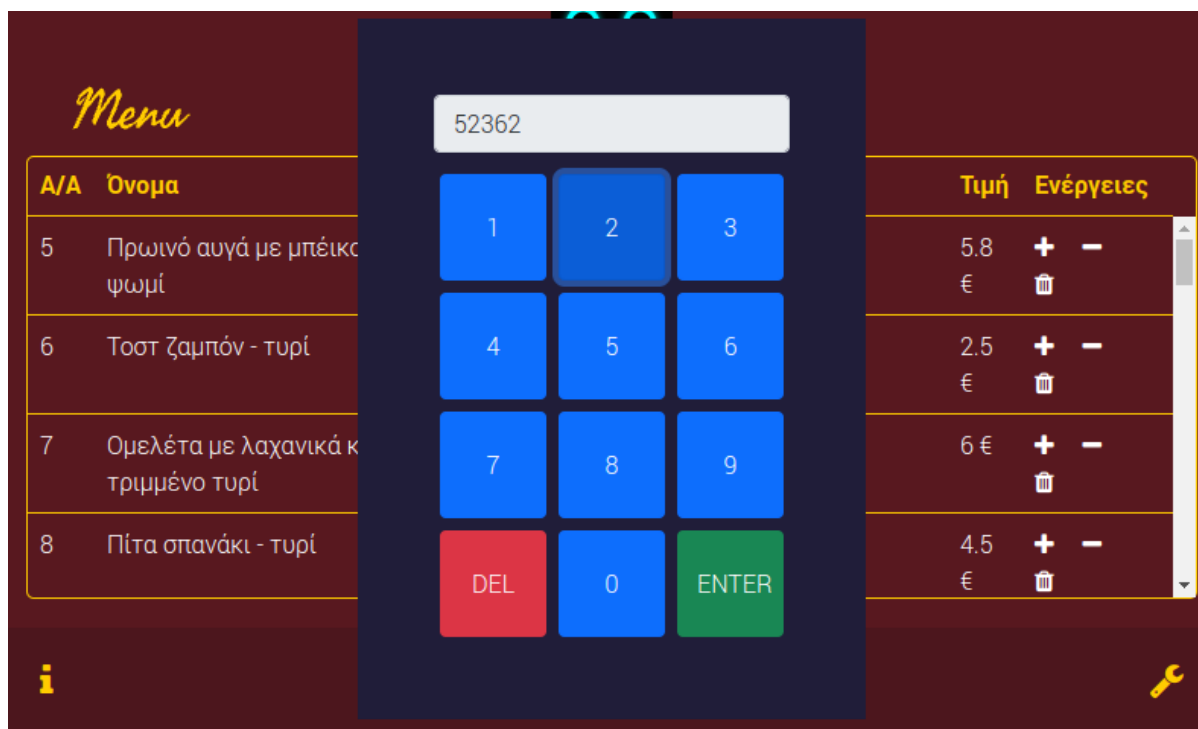
Εκτός από το UI των πελατών, ο εξυπηρετητής φιλοξενεί και ένα UI το οποίο προορίζεται για χρήση από το προσωπικό του χώρου εστίασης. Αυτή η διεπαφή προσφέρει λειτουργίες που αφορούν περισσότερο την διαχείριση των παραγγελιών. Μέσα από αυτή το προσωπικό μπορεί να δει και να ολοκληρώσει τις παραγγελίες που έχουν τοποθετηθεί από τους πελάτες.

Η τρίτη και τελευταία διεπαφή, είναι η διεπαφή διαχείρισης για τεχνικά ζητήματα. Στη διεπαφή αυτή υπάρχουν λειτουργίες που αφορούν την κίνηση και την προβολή της κατάστασης του υλικού του σερβιτόρου και προορίζονται κυρίως για προγραμματισμό κινήσεων και αποσφαλμάτωση. Πιο συγκεκριμένα εμφανίζονται πληροφορίες όπως τιμές απόστασης των αισθητήρων υπερήχων, τάση μπαταριών και αν το σύστημα έκτακτης ανάγκης είναι ενεργοποιημένο. Ταυτόχρονα υπάρχουν κουμπιά για εκτέλεση οποιασδήποτε κίνησης με αριθμό βημάτων, καθώς και κουμπί απομακρυσμένης ενεργοποίησης του συστήματος έκτακτης ανάγκης. Τέλος για την πρόσβαση στη διεπαφή αυτή είναι περιορισμένη, καθώς απαιτείται η χρήση κωδικού PIN που είναι αποθηκευμένος στη βάση δεδομένων και εισάγεται με πληκτρολόγιο αφής από την διεπαφή των πελατών.

Ο εξυπηρετητής Flask εκτός του ότι φιλοξενεί τις διαφορετικές διεπαφές που προαναφέρθηκαν, συνδέεται και σε μία τοπική βάση δεδομένων MariaDB και παρέχει ένα API το οποίο χρησιμοποιείται για εξυπηρέτηση πληθώρας αιτημάτων, όπως για παράδειγμα τα αιτήματα παρασκηνίου που εκτελεί το σύστημα παραγγελιών, ή την εκτέλεση εντολών κίνησης. Το API αυτό ανταλλάσσει δεδομένα σε μορφή JSON τα οποία μετατρέπονται εύκολα σε dictionaries στον κώδικα python για την αποθήκευση ή ανάκτησή τους από την βάση δεδομένων.

Η ομιλία του ρομποτικού σερβιτόρου καθίσταται δυνατή χάρη σε υπηρεσία κειμένου σε ομιλία (TTS). Ο τρόπος με τον οποίο υλοποιείται, παρουσιάζει ενδιαφέρον καθώς κάνει χρήση caching των αρχείων ήχου για βελτίωση ποιότητας. Πιο συγκεκριμένα, όταν δίνεται η εντολή να εκφωνηθεί ένα κομμάτι κειμένου, ακολουθούνται κάποια βήματα για την εκτέλεση της ενέργειας αυτής. Αρχικά, γίνεται αίτημα από το frontend στο API του Flask για αναζήτηση του αρχείου ήχου που αντιστοιχεί στο ζητούμενο κείμενο. Το API αναζητά βάση του κειμένου στην βάση δεδομένων για πλειάδα η οποία, αν υπάρχει, θα περιέχει και την διαδρομή στο αρχείο ήχου, το οποίο αντιστοιχεί στο κείμενο αυτό. Σε περίπτωση που βρεθεί, το API αποστέλλει το αρχείο ήχου, σε δυαδική μορφή κωδικοποιημένη σε Base64, στο frontend το οποίο και το αναπαραγάγει από τα ηχεία του ρομποτικού σερβιτόρου. Στην περίπτωση όμως που δεν βρεθεί εγγραφή στην βάση δεδομένων, το API αναλαμβάνει την δημιουργία αντίστοιχου αρχείου. Για να το πετύχει αυτό, κάνει χρήση ενός δωρεάν API που λέγεται soundoftext, το οποίο χρησιμοποιεί το την μηχανή κειμένου σε ομιλία του Google Translate. Σαν πρώτο βήμα στέλνεται αίτημα στο API σε μορφή JSON που περιέχει δεδομένα όπως το κείμενο προς μετατροπή σε ήχο, καθώς και η γλώσσα στην οποία βρίσκεται. Έπειτα, και με την επιβεβαίωση εκκίνησης της διαδικασίας μετατροπής από το API, ο κώδικας μπαίνει σε βρόγχο κατά την διάρκεια του οποίου στέλνει αιτήματα, ανά τακτά διαστήματα, στο API για ενημέρωση της προόδου της μετατροπής. Μόλις λάβει απάντηση ότι το αρχείο ήχου είναι έτοιμο, ο βρόγχος τερματίζεται και το αρχείο αποθηκεύεται τοπικά. Έπειτα εισάγεται καινούρια εγγραφή στην βάση δεδομένων που αποτελείται από το κείμενο, την τοποθεσία του αρχείου στο σύστημα αρχείων καθώς και

μοναδικός κωδικός του οποίου μάλιστα το όνομα αποκτά και το αρχείο κατά την ολοκλήρωση της λήψης του. Τέλος το αρχείο αποστέλλεται, όπως και στην πρώτη περίπτωση, στο frontend το οποίο το αναπαράγει. Με τον τρόπο αυτό εξασφαλίζεται ο βέλτιστος συγχρονισμός κειμένου και ήχου κατά την ομιλία του ρομποτικού σερβιτόρου, αλλά και η εύκολη και δυναμική δημιουργία εκφώνηση νέων κειμένων σε πραγματικό χρόνο.



Εικόνα 10: Οθόνη εισαγωγής PIN για πρόσβαση στο τεχνικό UI

Διαχείριση				
Εκκρεμείς παραγγελίες				
Ολοκληρωμένες παραγγελίες				
Αριθμός παραγγελίας: 30 - Σύνολο: 20.30 € - Τραπεζί: 3 - Ημ/νία καταχώρησης: 19/11/2022, 4:22:40 μ.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 29 - Σύνολο: 8.50 € - Τραπεζί: 1 - Ημ/νία καταχώρησης: 19/11/2022, 4:19:17 μ.μ. - Ολοκλήρωση παραγγελίας				
Κωδικός	Όνομα	Διαθέσιμο	Τιμή μονάδας	Ποσότητα
6	Τοστ ζαμπόν - τυρί	Ναι	2.50 €	1
7	Ομελέτα με λαχανικά και τριμμένο τυρί	Ναι	6.00 €	1
Αριθμός παραγγελίας: 28 - Σύνολο: 2.50 € - Τραπεζί: 2 - Ημ/νία καταχώρησης: 16/11/2022, 2:37:08 μ.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 27 - Σύνολο: 12.00 € - Τραπεζί: 3 - Ημ/νία καταχώρησης: 16/11/2022, 2:29:02 μ.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 26 - Σύνολο: 4.50 € - Τραπεζί: 1 - Ημ/νία καταχώρησης: 16/11/2022, 2:27:58 μ.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 25 - Σύνολο: 14.00 € - Τραπεζί: 1 - Ημ/νία καταχώρησης: 11/11/2022, 5:33:16 μ.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 24 - Σύνολο: 5.00 € - Τραπεζί: 2 - Ημ/νία καταχώρησης: 11/11/2022, 12:24:02 μ.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 23 - Σύνολο: 4.50 € - Τραπεζί: 3 - Ημ/νία καταχώρησης: 11/11/2022, 11:49:43 π.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 22 - Σύνολο: 2.50 € - Τραπεζί: 1 - Ημ/νία καταχώρησης: 11/11/2022, 11:44:49 π.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 21 - Σύνολο: 8.00 € - Τραπεζί: 2 - Ημ/νία καταχώρησης: 11/11/2022, 11:24:20 π.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 20 - Σύνολο: 6.00 € - Τραπεζί: 1 - Ημ/νία καταχώρησης: 11/11/2022, 11:21:52 π.μ. - Ολοκλήρωση παραγγελίας				
Αριθμός παραγγελίας: 19 - Σύνολο: 12.00 € - Τραπεζί: 1 - Ημ/νία καταχώρησης: 11/11/2022, 11:11:12 π.μ. - Ολοκλήρωση παραγγελίας				

Εικόνα 9: Διαχειριστική διεπαφή προσωπικού



Εικόνα 11: Διεπαφή τεχνικής διαχείρισης

2.2.1.3 Σχεσιακή βάση δεδομένων

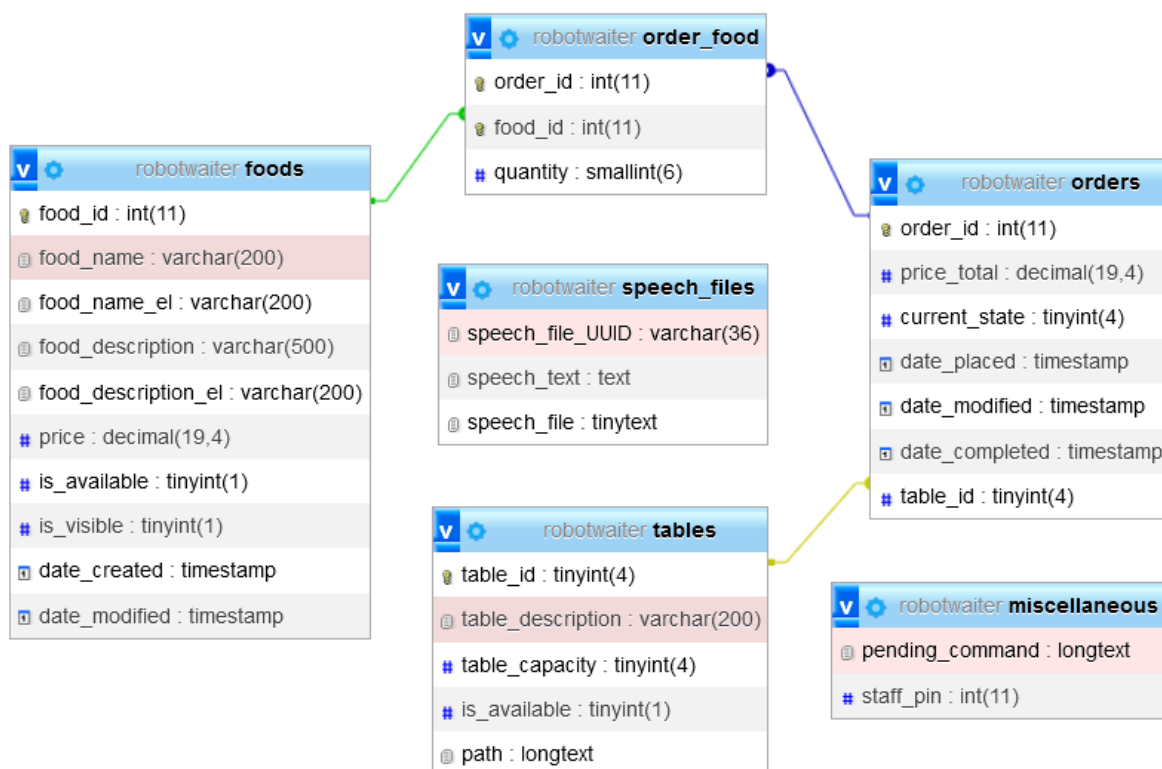
Για την αποθήκευση και οργάνωση των δεδομένων που είναι απαραίτητα για την λειτουργία του ρομποτικού σερβιτόρου, υλοποιήθηκε μία σχεσιακή βάση δεδομένων. Για την ανάπτυξή της χρησιμοποιήθηκε μια βάση δεδομένων βασισμένη σε SQL, που ονομάζεται MariaDB.

Αρχικά, η σχεσιακή βάση δεδομένων οργανώνει τα δεδομένα της με χρήση δομών που ονομάζονται «σχέσεις», οι οποίες είναι στην πραγματικότητα πίνακες με σειρές και στήλες που ονομάζονται πλειάδες και γνωρίσματα αντίστοιχα. Στη συνέχεια αυτοί οι πίνακες συσχετίζονται μεταξύ τους, με χρήση «ξένων κλειδιών». Στην περίπτωση μας, για την εξυπηρέτηση των αναγκών του λογισμικού του ρομποτικού σερβιτόρου, δημιουργήθηκαν έξι διαφορετικοί πίνακες, εκ των οποίων οι τέσσερις συσχετίζονται. Ο πρώτος είναι ο πίνακας των φαγητών, όπου αποθηκεύονται τα διάφορα προϊόντα το καταλόγου ενός καταστήματος. Περιέχει γνωρίσματα όπως το όνομα του φαγητού, ο μοναδικός του κωδικός, η τιμή του κ.α. Ο δεύτερος πίνακας αφορά τις παραγγελίες και περιέχει πληροφορίες όπως ημερομηνία και ώρα καταχώρισης της παραγγελίας, αριθμό τραπεζιού και συνολικό κόστος. Η χρησιμότητα της σχεσιακής βάσης δεδομένων γίνεται φανερή στην διασύνδεση των δεδομένων των δύο προαναφερθέντων πινάκων, καθώς με χρήση ενός τρίτου βοηθητικού πίνακα είναι δυνατή η αντιστοίχιση σειρών των δύο πινάκων, ενώ ταυτόχρονα αποθηκεύονται και επιπλέον δεδομένα για την συσχέτιση αυτή. Πιο συγκεκριμένα γίνεται ο βοηθητικός πίνακας περιέχει σε κάθε του πλειάδα το ξένο κλειδί μιας παραγγελίας, το ξένο κλειδί ενός από τα φαγητά που περιλαμβάνει, καθώς και την ποσότητα αυτού του φαγητού. Για κάθε προϊόν σε μία

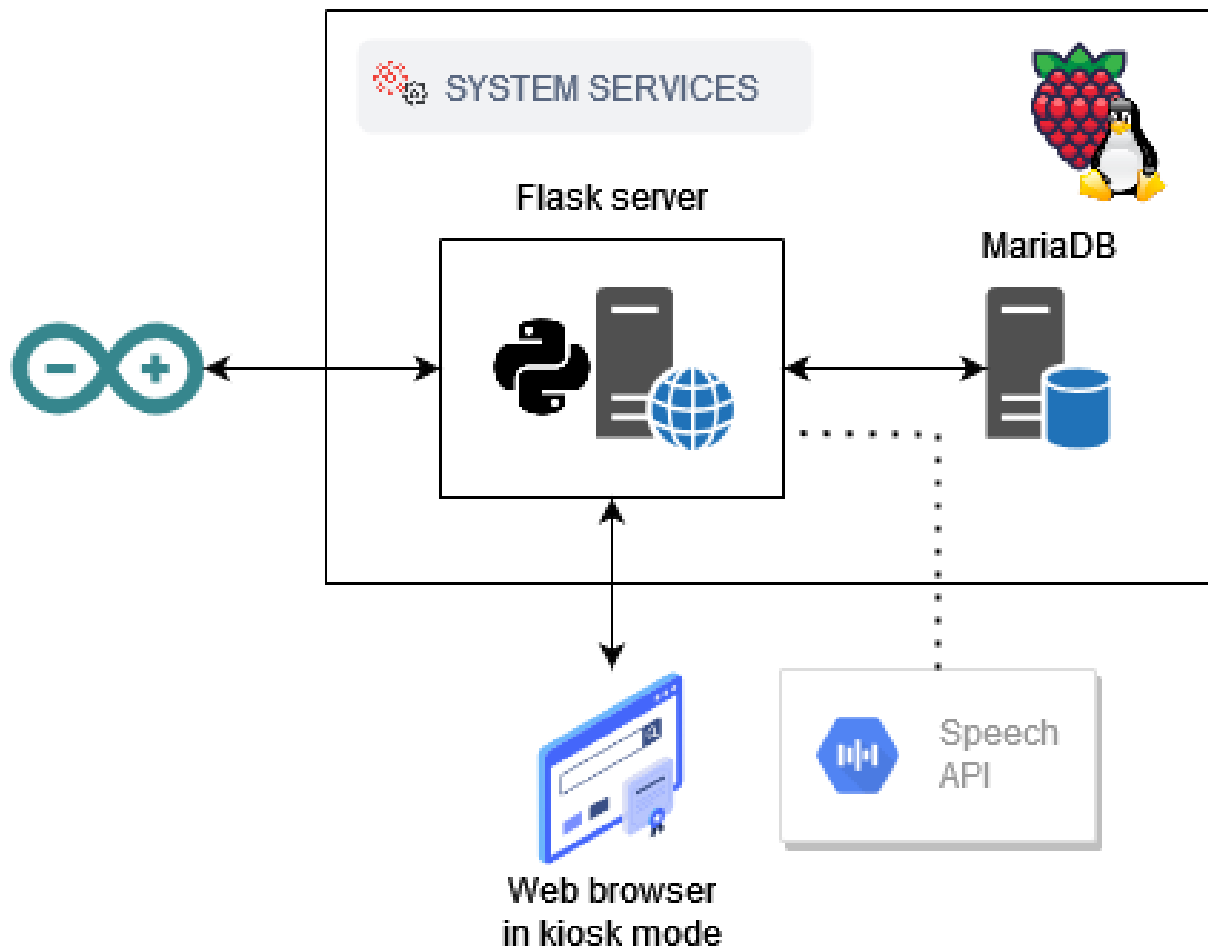
παραγγελία υπάρχει και μια σειρά στον βοηθητικό πίνακα που το συσχετίζει με την παραγγελία αυτή. Γενικότερα οι βοηθητικοί πίνακες χρησιμοποιούνται για να επιλύσουν τις λεγόμενες σχέσεις «πολλά προς πολλά» όπως στην περίπτωση μας, όπου πολλά φαγητά μπορεί να αντιστοιχούν σε πολλές παραγγελίες. Τέλος, υπάρχει και ακόμη ένας πίνακας, ο οποίος περιέχει πληροφορίες για τα τραπέζια και συσχετίζεται με τον πίνακα παραγγελιών. Είναι σημαντικό να αναφερθεί ότι ο συγκεκριμένος πίνακας εκτός από τα γνώρισμα που αφορούν καθαρά το σύστημα παραγγελιών, αποθηκεύει ακόμη ένα γνώρισμα, που είναι η διαδρομή του τραπεζιού. Πρόκειται για τις κινήσεις που πρέπει να κάνει ο ρομποτικός σερβιτόρος για να κινηθεί από και προς το τραπέζι, και είναι κωδικοποιημένο σε μορφή JSON String.

Εκτός από τους πίνακες που ήδη αναφέρθηκαν, υπάρχουν και άλλοι δύο. Ο πρώτος χρησιμοποιείται αποκλειστικά από το speech API και αποθηκεύει ένα UUID για κάθε αρχείο φωνής, το κείμενο στο οποίο αντιστοιχεί το αρχείο αυτό, καθώς και το όνομα του αρχείου όπως αυτό είναι αποθηκευμένο στο τοπικό σύστημα αρχείων. Ο δεύτερος πίνακας δεν ακολουθεί πρακτικές σχεσιακής βάσης και περιέχει μόνο μία πλειάδα με δεδομένα, όπως είναι το PIN εισόδου στη διεπαφή τεχνικής διαχείρισης, αλλά και στοίβα εκκρεμών εντολών (επίσης κωδικοποιημένη σε μορφή JSON) για εκτέλεση από το σύστημα.

Τέλος, η πλήρης δομή της βάσης δεδομένων, καθώς και όσα αναλύθηκαν προηγουμένως, φαίνονται και σε σχετικό διάγραμμα παρακάτω.



Εικόνα 12: Διάγραμμα σχεσιακής βάσης δεδομένων



Εικόνα 13: Διάγραμμα αρχιτεκτονικής λογισμικού

3

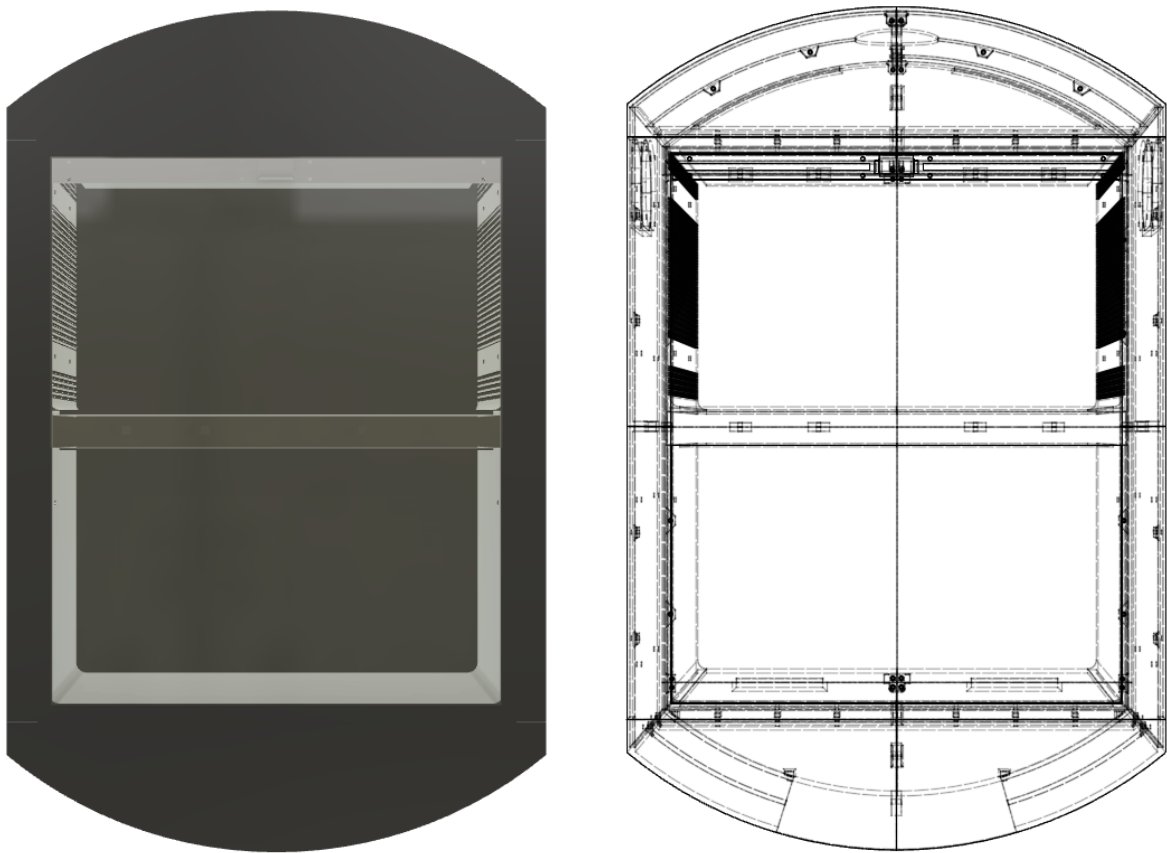
Σχεδίαση κατασκευής ρομποτικού σερβιτόρου

3.1 Αποθηκευτικός χώρος

Για την επιτυχή διακομιδή των παραγγελιών, ο ρομποτικός σερβιτόρος πρέπει να είναι σε θέση να τις αποθηκεύει και να τις προφυλάσσει από το περιβάλλον. Για να επιτευχθεί αυτό είναι απαραίτητη η τοποθέτηση τους σε κάποιο κλειστό χώρο πριν και μετά από την μεταφορά. Ταυτόχρονα όμως, είναι σημαντική η διατήρηση της ανθρωποειδούς εμφάνισης του ρομποτικού σερβιτόρου. Έτσι, είναι απαραίτητο ο κλειστός χώρος αποθήκευσης να είναι σχεδιασμένος κατά τέτοιον τρόπο, ώστε να ενσωματώνεται ομαλά στην συνολική κατασκευή, και να θυμίζει μέρος ενός ανθρώπινου σώματος.

Με γνώμονα τα παραπάνω, τον χώρο στον οποίο αποθηκεύονται οι παραγγελίες, επιλέχθηκε να αναπαριστά το «σώμα» του ανθρωποειδούς ρομπότ. Η επιλογή αυτή έγινε καθώς το τμήμα αυτό αποτελεί, όπως και σε έναν άνθρωπο, την κύρια ενιαία μάζα του σώματος. Το γεγονός αυτό εξασφαλίζει ότι η αναλογία του όγκου του προστατευμένου χώρου αποθήκευσης, και του συνολικού όγκου του σώματος θα είναι η καλύτερη δυνατή. Έτσι σχεδιάστηκε ένας χώρος αποθήκευσης χωρισμένος σε δύο ράφια όπως φαίνεται και στην (βλ. Εικόνα 14: Αποθηκευτικός χώρος παραγγελιών).

Για την μεταφορά αντικειμένων που πιθανώς είναι ζεστά, το κάτω μέρος κάθε ραφιού είναι κατασκευασμένο από πλεξιγκλάς έναντι υλικού εκτύπωσης PLA, το οποίο έχει χαμηλή θερμοκρασία τήξης, για την αποφυγή παραμόρφωσης ή και καταστροφής του τμήματος αυτού.



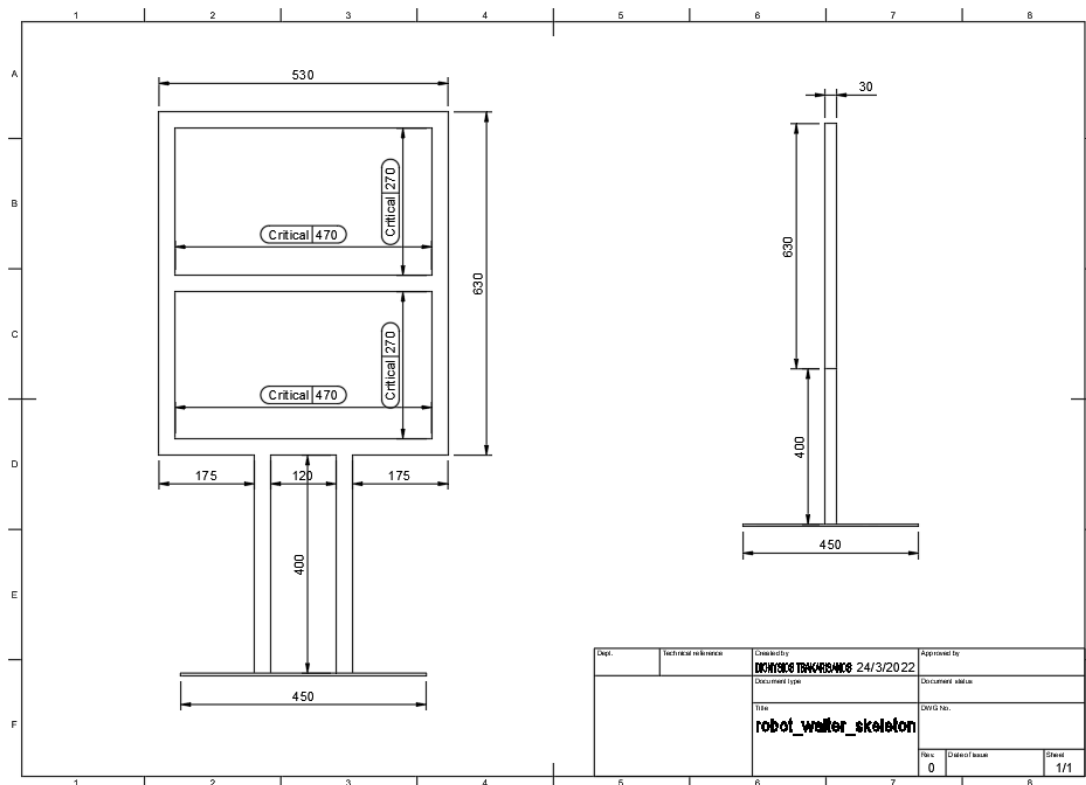
Εικόνα 14: Αποθηκευτικός χώρος παραγγελιών

Τέλος, για την προφύλαξη των παραγγελιών από τον περιβάλλοντα χώρο, σχεδιάστηκε και κατασκευάστηκε μια πρόσοψη για τον χώρο αποθήκευσης, η οποία αποτελείται από δύο συρόμενες πόρτες πλεξιγκλάς. Οι πόρτες αυτές κινούνται σε δύο παράλληλους οδηγούς που βρίσκονται στα πλαϊνά σημεία της πρόσοψης, επιτρέποντας το άνοιγμα μόνο μίας από τις δύο τη φορά. Πάνω στις πόρτες αυτές είναι εγκατεστημένοι τρισδιάστατα εκτυπωμένοι μηχανισμοί κλειδώματος, οι οποίοι επιτυγχάνουν την παγίωση των πορτών κατά την κίνηση του ρομπότ. Οι μηχανισμοί αυτοί λειτουργούν με την χρήση δύο αντικριστών συρτών, οι οποίοι έχουν την τάση να απομακρύνονται από το κέντρο με την βοήθεια ενός ελατηρίου. Στη συνέχεια οι σύρτες εισέρχονται σε ειδικές εγκοπές που βρίσκονται στους οδηγούς της πόρτας, στο σημείο που αυτή είναι στην κλειστή της θέση.

3.2 Μεταλλικός σκελετός

Το ογκώδες μέγεθος του ρομποτικού σερβιτόρου, συνεπάγεται μία κατασκευή η οποία έχει σημαντικά ζητήματα στιβαρότητας καθώς και σταθερότητας. Συνεπώς δημιουργήθηκαν πολλοί προβληματισμοί στον σχεδιασμό αλλά και στην υλοποίησή του.

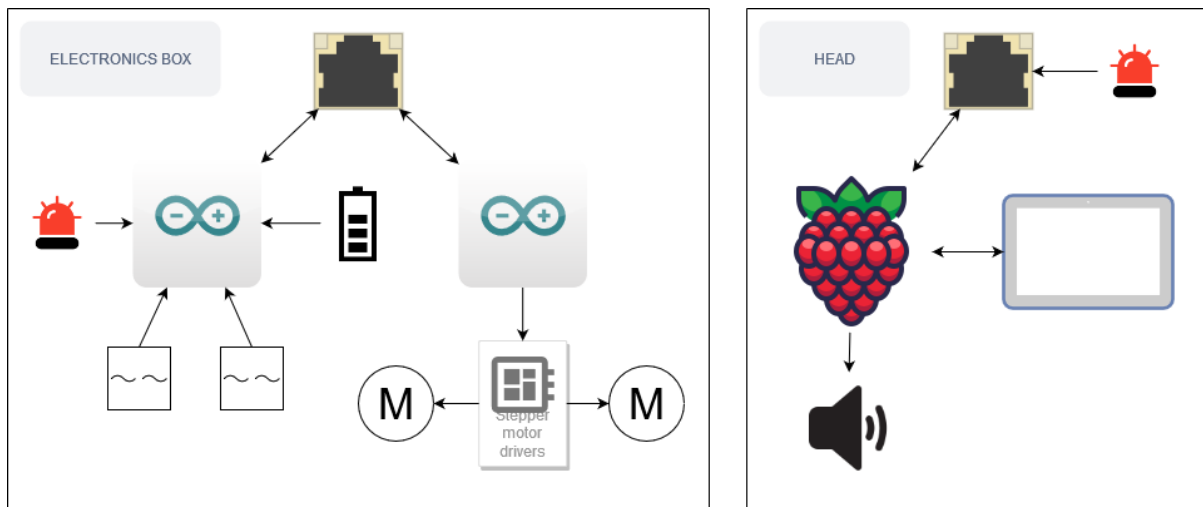
Για την απόδοση στιβαρότητας στην κατασκευή, επιλέχθηκε η χρήση ενός μεταλλικού σκελετού, ο οποίος αποτελεί σημείο στήριξης των διαφόρων τμημάτων του ρομπότ. Πιο συγκεκριμένα, ο σκελετός περικυκλώνει τα ράφια αποθήκευσης του ρομπότ, καθώς πάνω του εγκαθίστανται τα πλαστικά κομμάτια που τα αποτελούν. Επίσης στο πάνω μέρος του βιδώνεται ο λαιμός, ο οποίος με την σειρά του στηρίζει το κεφάλι του ρομπότ. Στην συνέχεια στις εξωτερικές πλευρές του σκελετού προσαρμόζονται τα πλαϊνά πλαστικά τμήματα του κύριου σώματος του ρομποτικού σερβιτόρου, κάνοντας χρήση πλευρικών αντηρίδων που προεξέχουν από κάθε πλευρά του σκελετού.



Εικόνα 15: Σχέδιο βασικών διαστάσεων μεταλλικού σκελετού

3.3 Ηλεκτρονικά

Για την επίτευξη της αυτόνομης αλληλεπίδρασης του με το περιβάλλον, ο ρομποτικός σερβιτόρος διαθέτει ηλεκτρολογική και ηλεκτρονική υποδομή που αποτελείται από διάφορα υποσυστήματα και τρόπους διασύνδεσης τους. Το υλικό αυτό, στην πλειοψηφία του, βάση σχεδίασης είναι τοποθετημένο σε ειδικά διαμορφωμένο πίνακα, αλλά και στο εσωτερικό του κεφαλιού του ρομπότ. Επομένως στο κεφάλαιο αυτό θα αναλυθεί ο ηλεκτρονικός σχεδιασμός του ρομποτικού σερβιτόρου.



Εικόνα 16: Συνοπτικό διάγραμμα ηλεκτρονικών ρομποτικού σερβιτόρου

3.3.1 Σχεδιασμός ηλεκτρονικών πίνακα

Ο πίνακας αυτός είναι εγκατεστημένος στο κάτω μέρος της πίσω πλευράς του σώματος του ρομπότ. Σκοπός του είναι η οργάνωση και προστασία των ηλεκτρονικών ελέγχου του ρομποτικού σερβιτόρου. Στο κεφάλαιο αυτό θα γίνει περιγραφή της σχεδίασης του καθώς και θα αναλυθούν τα περιεχόμενα του και πως αυτά διασυνδέονται, ώστε να συνεργάζονται για την εκτέλεση των λειτουργιών τους.

Αρχικά, μετά από την επιλογή των εξαρτημάτων που θα αποτελούσαν το ηλεκτρονικό υλικό, σειρά είχε η σχεδίαση ενός χώρου ο οποίος θα τα προστάτευε από το περιβάλλον (π.χ. υγρασία, σκόνη) και όπου θα ήταν τακτοποιημένα και οργανωμένα. Έτσι ο πίνακας πρέπει να έχει της κατάλληλες διαστάσεις ώστε να χωρά τα διάφορα ηλεκτρονικά εξαρτήματα, όπως πλακέτες, καλώδια, διακόπτες κτλ. Για τον λόγο αυτό, κατά την σχεδίαση του γίνεται πλάνο κατανομής χώρου, κάνοντας χρήση σχεδίων των περιεχομένων που προβλέπεται να εγκατασταθούν στο εσωτερικό του, τα οποία βρίσκονται στις πραγματικές τους διαστάσεις. Για την στήριξη των πλακετών, χρησιμοποιήθηκαν εκτυπωμένοι πλαστικοί αποστάτες, οι

οποίοι προεξέχουν από τον πάτο του πίνακα και πάνω τους βιδώθηκαν οι πλακέτες. Ποιες όμως είναι οι πλακέτες αυτές και τι λειτουργίες εκτελούν;

Η πρώτη και σημαντικότερη πλακέτα, είναι το Arduino κίνησης. Το Arduino UNO είναι από τους πιο γνωστούς και χρησιμοποιημένους παγκοσμίως μικροελεγκτές. Μέσω προγραμματιστικά ελεγχόμενων εισόδων και εξόδων γενικής χρήσης, επιτρέπει την αλληλεπίδραση και χειρισμό πολλών συσκευών, όπως κινητήρες, λαμπτήρες, αισθητήρες κ.α. Στην περίπτωση μας συνδυάζεται δύο ελεγκτές βηματικών κινητήρων TMC2209 για τον χειρισμό των κινητήρων του ρομπότ. Η σύνδεση των οδηγών με το Arduino γίνεται χάρη σε μια βοηθητική πλακέτα Arduino CNC Shield το οποίο εφαρμόζει πάνω του. Εκτός από τους οδηγούς, στο CNC Shield συνδέεται και η τροφοδοσία των κινητήρων. Το CNC Shield δεσμεύει όλα τα I/O pins του Arduino, συνεπώς χρειάζεται μια δεύτερη Arduino για τις υπόλοιπες λειτουργίες του ρομπότ.

Η δεύτερη Arduino είναι υπεύθυνη για την συλλογή δεδομένων των αισθητήρων του ρομποτικού σερβιτόρου που αφορούν το σύστημα έκτακτης ανάγκης, τους αισθητήρες απόστασης και παρακολούθηση στάθμης μπαταριών. Οι αισθητήρες του συστήματος έκτακτης ανάγκης αποτελούνται από δύο αισθητήρες υπερήχων, που χρησιμοποιούνται για τον εντοπισμό εμποδίων και την απόσταση τους από αυτά, καθώς και από δύο κουμπιά έκτακτης διακοπής λειτουργίας. Για τον υπολογισμό της απόστασης, οι αισθητήρες υπερήχων αποτελούνται από έναν πομπό και ένα δέκτη υπερήχων. Οι πομποί δέχονται από την Arduino ένα σήμα trigger και εκπέμπουν υπέρηχους παλμούς, οι οποίοι αντανακλώνται στην επιφάνεια ενός εμποδίου και επιστρέφουν πίσω στους αισθητήρες και λαμβάνονται από τους αντίστοιχους δέκτες. Οι δέκτες με την σειρά τους στέλνουν σήμα πίσω στην Arduino, η οποία με την σειρά της υπολογίζει το χρονικό διάστημα που πέρασε μεταξύ της αποστολής και της λήψης του παλμού, και στην συνέχεια το χρησιμοποιεί για να υπολογίζει την απόσταση από το αντικείμενο. Στην Arduino αυτή, συνδέεται επίσης custom PCB μέσω του οποίου μετρά την τάση των μπαταριών του ρομπότ. Τέλος, η Arduino αισθητήρων στέλνει μέσω ενός καλωδίου σήμα στην Arduino κίνησης για παύση της κίνησης (σήμα HOLD).

Εκτός από τις δύο Arduino στον πίνακα υπάρχουν και άλλα βοηθητικά εξαρτήματα. Πιο συγκεκριμένα μέσα βρίσκονται δύο από τους συνολικούς τρεις DC-DC converters (μετατροπείς τάσης), οι οποίοι αναλαμβάνουν την μετατροπή της τάσης των μπαταριών σε 5V και 12V αντίστοιχα για τροφοδοσία των ηλεκτρονικών και του ανεμιστήρα ψύξης αντίστοιχως. Οι μετατροπείς αυτοί είναι παίζουν βασικό ρόλο στην διατήρηση της ορθής λειτουργίας του ρομποτικού σερβιτόρου, καθώς σταθεροποιούν την τάση του ρεύματος που προέρχεται από τις μπαταρίες των οποίων η τάση αυξομειώνεται διαρκώς.

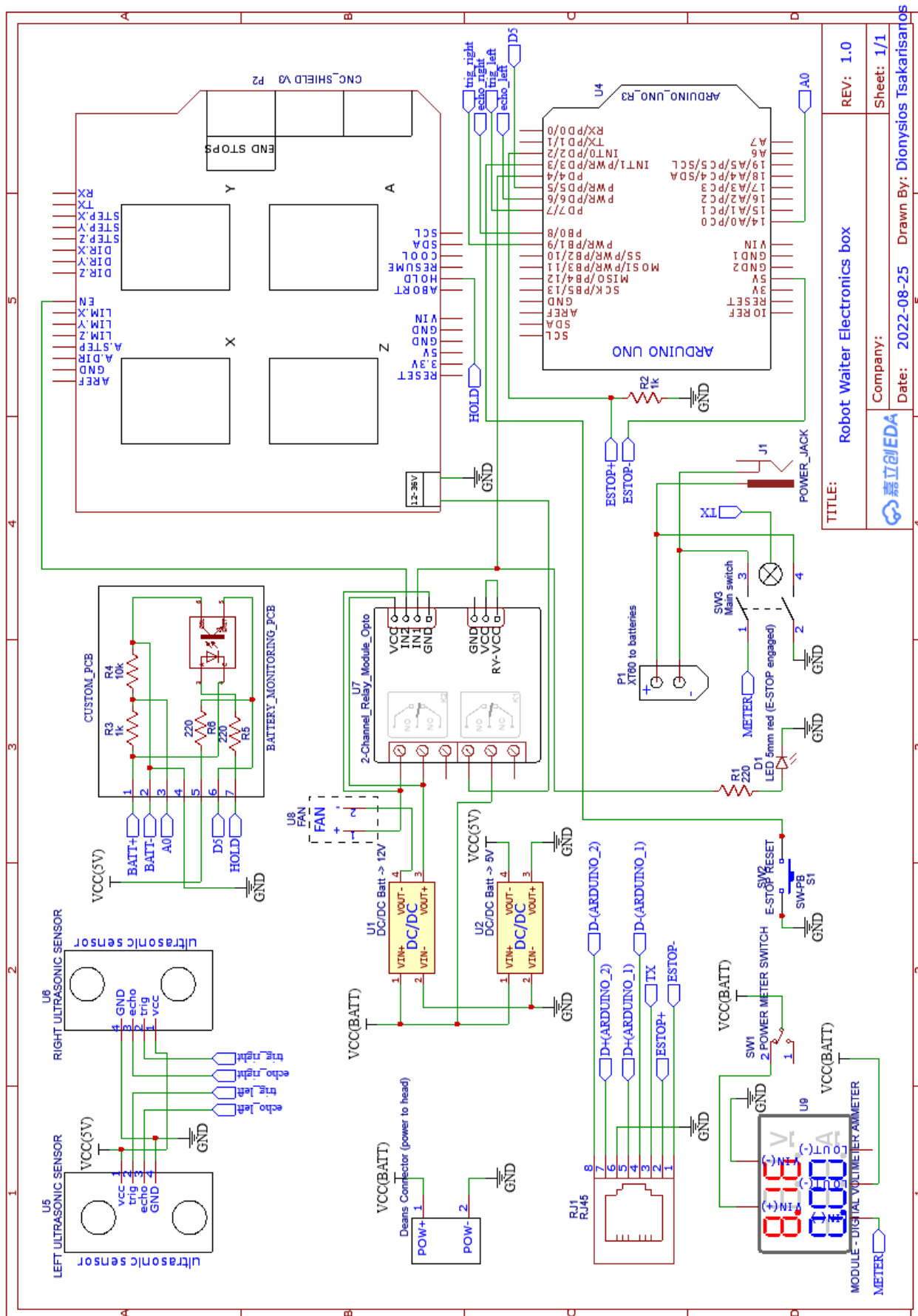
Επίσης, στον πίνακα ηλεκτρονικών υπάρχουν δύο relays (ηλεκτρονόμοι). Το πρώτο παρεμβάλλεται στην τροφοδοσία του CNC Shield, αποτελεί κομμάτι του συστήματος έκτακτης ανάγκης και αναλαμβάνει την διακοπή της τροφοδοσίας αυτής σε περίπτωση ενεργοποίησης του. Έτσι διακόπτεται ακαριαία η ισχύς στους κινητήρες του ρομποτικού σερβιτόρου με αποτέλεσμα την ταχύτερη ακινητοποίηση του. Το δεύτερο relay, αναλαμβάνει την ενεργοποίηση και απενεργοποίηση του ανεμιστήρα ψύξης των βηματικών κινητήρων. Ελέγχεται από το CNC Shield (από το enable pin) ώστε να ενεργοποιείται μόνο κατά την

διάρκεια της κίνησης του ρομπότ, ώστε να ψύχονται οι οδηγοί και να αποφεύγεται η υπερθέρμανση τους.

Το πίσω μέρος του πίνακα καταλαμβάνει ένα πάνελ, το οποίο περιέχει χρήσιμες ενδείξεις της κατάστασης του ρομποτικού σερβιτόρου, καθώς και διάφορα controls. Μία από τις ενδείξεις αυτές, είναι μια οθόνη που αναγράφει την τάση των μπαταριών, αλλά και την κατανάλωση ρεύματος του ρομπότ σε πραγματικό χρόνο, η οποία συνοδεύεται από διακόπτη απενεργοποίησης της για εξοικονόμηση ενέργειας. Για την μέτρηση της συνολικής κατανάλωσης ρεύματος ο μετρητής βρίσκεται ηλεκτρολογικά ενδιάμεσα στην παροχή ρεύματος (μπαταρίες) και στα υπόλοιπα συστήματα. Έπειτα υπάρχει ένα κουμπί επαναφοράς του συστήματος έκτακτης ανάγκης, και πάνω από αυτό υπάρχει αντίστοιχη ενδεικτική λυχνία η οποία φωτίζει όταν το σύστημα αυτό είναι ενεργό. Παρακάτω βρίσκεται ο γενικός διακόπτης του ρομπότ, ο οποίος παρεμβάλλεται μεταξύ των μπαταριών και του υπόλοιπου πίνακα. Έτσι όταν είναι ανοικτός, οι μπαταρίες είναι εντελώς απομονωμένες από το υπόλοιπο ρομπότ. Στον γενικό διακόπτη υπάρχει λυχνία χαμηλής τάσης η οποία είναι σβηστή όταν ο κεντρικός μικροεπεξεργαστής του ρομποτικού σερβιτόρου (Raspberry Pi) είναι σε κατάσταση shutdown. Έτσι αποφεύγεται η πρόωρη χρήση του διακόπτη και η αποκοπή τροφοδοσίας, προτού προλάβει να ολοκληρωθεί ομαλά ο τερματισμός λειτουργίας του μικροεπεξεργαστή και ως εκ τούτου ελαχιστοποιείται ο κίνδυνος διαφθοράς λειτουργικού συστήματος και του συστήματος αρχείων του. Η λυχνία αυτή, για να μπορεί να δείχνει έγκυρα την κατάσταση του Raspberry Pi, συνδέεται απευθείας στο GPIO pin TX της σειριακής θύρας του, καθώς το pin αυτό απενεργοποιείται μόνο όταν η πλακέτα έχει ολοκληρώσει τον τερματισμό λειτουργίας της. Μετά από τον διακόπτη υπάρχει μία ειδική σχάρα που προφυλάσσει την εισαγωγή αέρα που προορίζεται για τον ανεμιστήρα ψύξης. Τέλος, υπάρχει μια υποδοχή βύσματος όπου συνδέεται ο φορτιστής των μπαταριών. Αυτό είναι και το μοναδικό εξάρτημα το οποίο συνδέεται πριν από τον γενικό διακόπτη, εφόσον είναι απαραίτητο για να πραγματοποιηθεί η φόρτιση, να είναι αποσυνδεδεμένες οι μπαταρίες από το υπόλοιπο ρομπότ.

Στον πίνακα ηλεκτρονικών συνδέονται επίσης τα ηλεκτρονικά του κεφαλιού του ρομποτικού σερβιτόρου, οι αισθητήρες απόστασης, οι βηματικοί κινητήρες καθώς και οι μπαταρίες. Όσον αφορά τα τρία τελευταία, αυτά συνδέονται απλώς μέσω καλωδίων που εξέρχονται από τον πίνακα ηλεκτρονικών, και συνεπώς αποτελούν ένα ενιαίο κομμάτι. Αντίθετα, για τις συνδέσεις από και προς το κεφάλι, υπάρχουν ειδικές θύρες και βύσματα για την σύνδεση με τον πίνακα. Οι συνδέσεις USB των Arduino που προορίζονται για σειριακές συνδέσεις μεταξύ του Raspberry Pi και των ίδιων, καθώς και καλώδια για τα κουμπιά έκτακτης ανάγκης, αλλά και για την λυχνία του γενικού διακόπτη, συνδέονται σε θύρα τύπου RJ45 στην οποία συνδέεται καλώδιο UTP το οποίο μεταφέρει τις συνδέσεις αυτές στο κεφάλι του ρομπότ. Το καλώδιο αυτό επίσης συνοδεύεται από δύο μονόκλινα καλώδια τροφοδοσίας τα οποία συνδέονται με τον πίνακα μέσω βύσματος Dean T-style plug, και μεταφέρουν τάση μπαταριών στο κεφάλι.

Τέλος, ολοκληρώνοντας την περιγραφή του πίνακα ηλεκτρονικών, η σχεδίαση του και άρα όσων αναφέρθηκαν παραπάνω, αποτυπώθηκε στο παρακάτω σχηματικό διάγραμμα. Σχεδιάστηκε με την βοήθεια του λογισμικού της EasyEDA.



Σχήμα 1: Σχηματικό διάγραμμα του πίνακα ηλεκτρονικών

3.3.2 Σχεδιασμός ηλεκτρονικών κεφαλιού

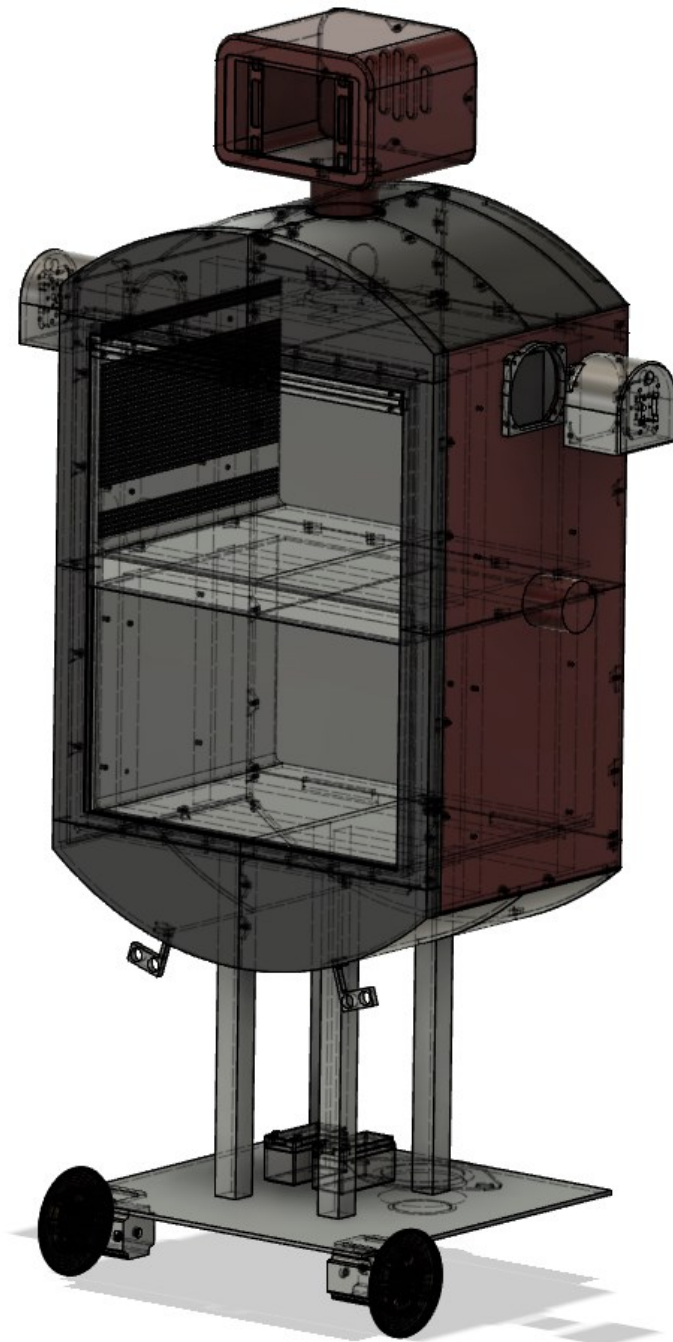
Στο κεφάλαιο «Σύστημα παραγγελιών» πραγματοποιήθηκε, εκτός των άλλων, περιγραφή της διεπαφής με την οποία καθίσταται δυνατή η αλληλεπίδραση του ρομποτικού σερβιτόρου με τον χρήστη. Για την προβολή της διεπαφής στον χρήστη, αλλά και για τον χειρισμό της από τον ίδιο, υπεύθυνη είναι μια οθόνη αφής. Πιο συγκεκριμένα επιλέχθηκε οθόνη αφής τεχνολογίας capacitive (πυκνωτή), με μέγεθος επτά ίντσες διαγώνιο και ανάλυσης 800 x 480 εικονοστοιχείων. Είναι ειδικά σχεδιασμένη για χρήση με τους μικροεπεξεργαστές Raspberry Pi και υποστηρίζει αναγνώριση δέκα ταυτόχρονων αγγιγμάτων. Η οθόνη αυτή βάση σχεδίου εγκαθίσταται σε ειδικά σχεδιασμένη εσοχή, που βρίσκεται στο μπροστινό μέρος του κεφαλιού του ρομπότ και που προστατεύει τα άκρα της.

Στο πίσω μέρος της οθόνης αφής εφαρμόζει και στερεώνεται μία πλακέτα Raspberry Pi 4 model B, μνήμης RAM 4GB και με επεξεργαστή Broadcom BCM2711 SoC. Ο μικροεπεξεργαστής αυτός επιλέχθηκε για την ικανότητα του να ανταπεξέλθει στις απαιτήσεις υπολογιστικής ισχύος που χρειάζεται για να εκτελείται αποδοτικά το λογισμικό του ρομποτικού σερβιτόρου. Ο τρόπος με τον οποίο συνδέεται με την οθόνη αφής, είναι μέσω ενός ribbon cable (καλωδιοταινίας) το οποίο συνδέεται στην θύρα DSI του RPI και στην συνέχεια στην οθόνη, ενώ στην ίδια παρέχεται τροφοδοσία τάσης 5V μέσω δύο καλωδίων (5V, GND) από το GPIO. Στο RPI επίσης συνδέονται οι δύο βύσματα USB που προέρχονται (μέσω του καλωδίου UTP) από τις δύο Arduino του πίνακα ηλεκτρονικών, καθώς επίσης και ένα καλώδιο στο Tx pin του GPIO για την λυχνία ένδειξης λειτουργίας. Ακόμη συνδέεται ένα ζεύγος ηχείων στην θύρα 3.5 mm και τα οποία τροφοδοτούνται με ρεύμα από μία Τρίτη θύρα USB.

Για την τροφοδοσία των ηλεκτρονικών του κεφαλιού (RPI, touchscreen, speakers), όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, δύο μονόκλινα καλώδια μεταφέρουν ρεύμα μπαταριών μέχρι τον λαιμό του ρομπότ. Εκεί βρίσκεται ο τρίτος και τελευταίος μετατροπέας τάσης, ο οποίος βγάζει στην έξοδό του τάση 5V και ρεύμα έως και 3A που είναι αρκετό για να τροφοδοτεί τα ηλεκτρονικά και συνδέεται στο RPI μέσω ενός USB Type-C.

Ακόμη, στο κεφάλι του ρομποτικού σερβιτόρου είναι τοποθετημένα ειδικά κουμπιά για την ενεργοποίηση του συστήματος έκτακτης ανάγκης. Πρόκειται για ογκώδη κόκκινα κουμπιά τα οποία βρίσκονται στην δεξιά και αριστερή πλευρά του κεφαλιού. Για την αποτελεσματικότερη χρήση του συστήματος Estop, πρέπει τα κουμπιά αυτά να είναι ευδιάκριτα και να γίνεται γρήγορα αντιληπτός ο τρόπος χρήσης τους. Για τον λόγο αυτό εκτός από την ογκώδη εμφάνιση τους και το κόκκινο χρώμα, σχεδιάστηκε ειδικός δακτύλιος κίτρινου χρώματος που εφαρμόζει γύρω από τα κουμπιά και αναγράφει «EMERGENCY STOP», κάνοντας ξεκάθαρο τον σκοπό τους. Από πλευράς συνδεσμολογίας τα πλήκτρα αυτά είναι συνδεδεμένα παράλληλα ώστε το πάτημα μόνο ενός από τα δύο να είναι αρκετό για την ενεργοποίηση του συστήματος Estop.

Έτσι ολοκληρώνεται η περιγραφή της σχεδίασης των ηλεκτρονικών του κεφαλιού του ρομποτικού σερβιτόρου. Να αναφερθεί ότι τα ηχεία που συνδέονται στο RPI, βρίσκονται στερεωμένα στο πάνω μέρος του βασικού σώματος του ρομπότ και συνεπώς δεν αποτελούν τμήμα του κεφαλιού.



Εικόνα 17: Σχέδιο ρομποτικού σερβιτόρου

4

Υλοποίηση κατασκευής

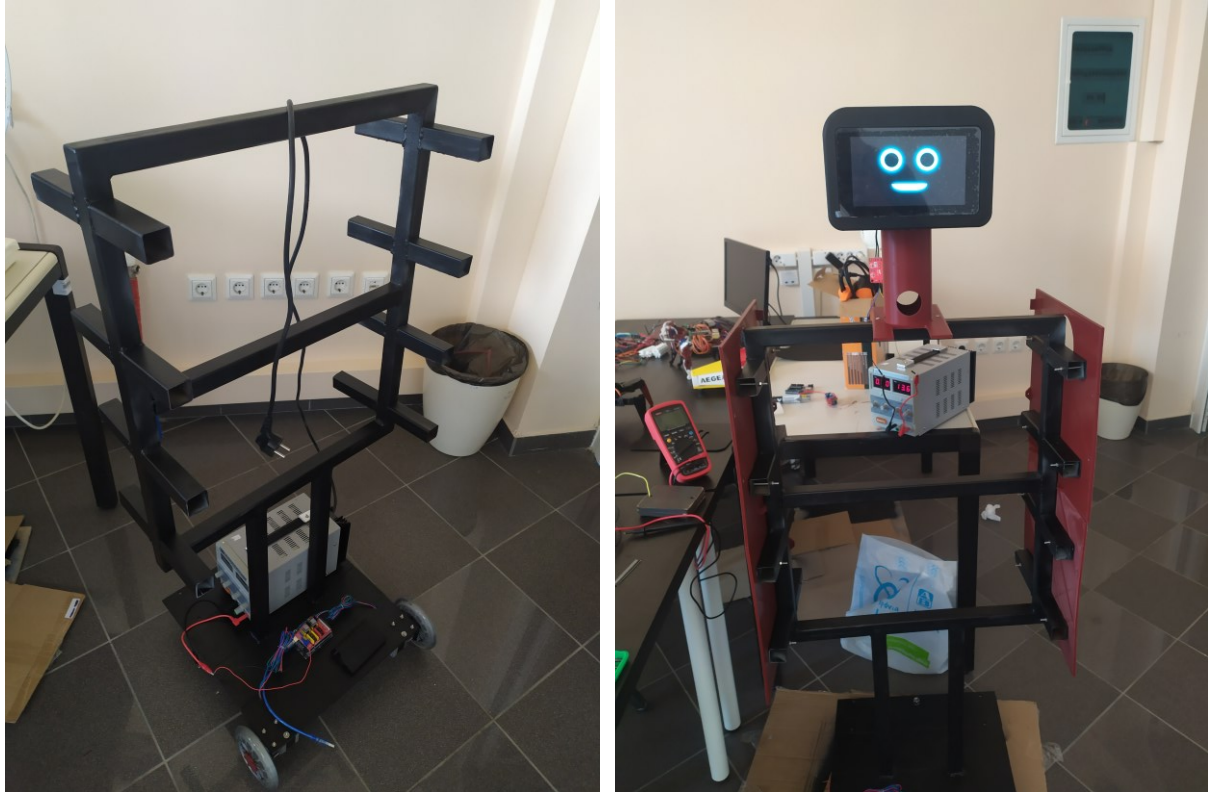
4.1 Σκελετός

Η υλοποίηση της κατασκευής του ρομποτικού σερβιτόρου χωρίζεται σε Ν βασικές φάσεις. Η πρώτη φάση ήταν η συναρμολόγηση του μεταλλικού σκελετού. Ο λόγος για τον οποίο επιλέχθηκε αυτή ως αρχική φάση, ήταν ώστε να δημιουργηθεί μια πλατφόρμα πάνω στην οποία θα μπορούσαν να γίνουν κάποιες αρχικές δοκιμές, όπως για παράδειγμα στην κίνηση, αλλά και ώστε να αποτελέσει βάση στήριξης για τις επόμενες φάσεις της κατασκευής.

Όπως ήδη αναφέρθηκε και στο κεφάλαιο της σχεδίασης, ο μεταλλικός σκελετός του ρομποτικού σερβιτόρου αποτελείται από μία μεταλλική βάση στην οποία στηρίχθηκε όλη η κατασκευή, και από το κύριο μέρος. Πιο αναλυτικά, χρησιμοποιείται τετράγωνη σωλήνα χάλυβα πάχους 30 x 30 mm τα οποία είναι συγκολλημένα μεταξύ τους με χρήση ηλεκτροκόλλησης, σχηματίζοντας τη γεωμετρία του σκελετού που σχεδιάστηκε. Οι διαστάσεις της τετράγωνης σωλήνας επιλέχθηκαν με σκοπό την επίτευξη της απαραίτητης στιβαρότητας του σκελετού. Ακόμη χάρη στις διαστάσεις αυτές, καθίσταται δυνατή η δημιουργία τρυπών σε στρατηγικά σημεία του σκελετού, οι οποίες χρειάζονται για την προσάρτηση των πλαστικών τμημάτων και συνεπώς της υπόλοιπης κατασκευής. Όσο για την βάση χρησιμοποιούνται τέσσερα κολλημένα μεταξύ τους κομμάτια λαμαρίνας πάχους 5 mm. Το πάχος αυτό είναι αρκετό για να βεβαιώνεται η σταθερότητα και η στιβαρότητα της βάσης υπό το βάρος της συνολικής κατασκευής, συν του φορτίου που αυτή μπορεί να μεταφέρει.

Τέλος, καθώς ο μεταλλικός σκελετός είναι φτιαγμένος από σίδηρο, πρέπει με κάποιο τρόπο να προστατευτεί από την οξείδωση. Γι' αυτό, μετά την οριστικοποίηση της υλοποίησης του, βάφτηκε με αντισκωριακό γκρι χρώμα σε μορφή σπρέι. Έπειτα, ως δεύτερη

επίστρωση χρησιμοποιήθηκε μαύρο ματ χρώμα, για να προστατευθεί η πρώτη στρώση, αλλά και για να ταιριάζει αισθητικά με την υπόλοιπη κατασκευή.



Εικόνα 18: Σκελετός ρομποτικού σερβιτόρου, προσαρμογή πλευρικών τμημάτων

4.2 Πλαστικά εξαρτήματα

Η δεύτερη και πιο σύνθετη φάση υλοποίησης της κατασκευής του ρομποτικού σερβιτόρου ήταν η εκτύπωση, συναρμολόγηση και προσάρτηση των πλαστικών τμημάτων. Η διαδικασία αυτή διήρκεσε κατέλαβε ίσως την μεγαλύτερη χρονική περίοδο.

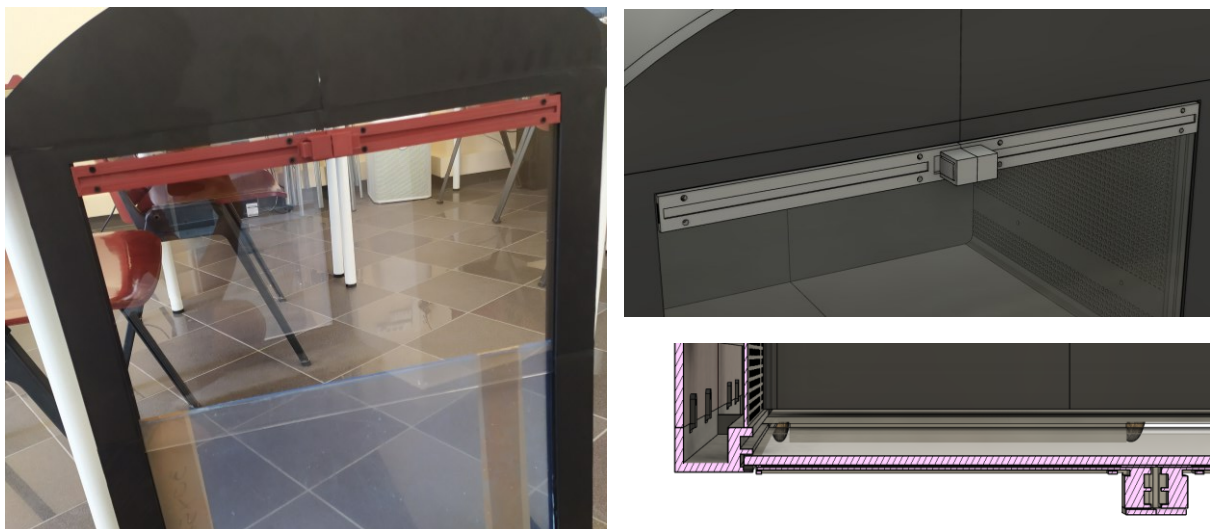
4.2.1 Κατασκευή πρόσοψης συρόμενων πορτών

Αφετηρία συναρμολόγησης του σώματος του ρομποτικού σερβιτόρου, αποτέλεσε η εκτύπωση και συναρμολόγηση της πρόσοψης του αποθηκευτικού χώρου με τις συρόμενες πόρτες από πλεξιγκλάς. Το τρισδιάστατο μοντέλο της πρόσοψης χωρίστηκε σε οχτώ επιμέρους τμήματα, καθώς οι συνολικές διαστάσεις του ξεπερνούσαν σε μέγεθος αυτές της επιφάνειας εκτύπωσης του τρισδιάστατου εκτυπωτή. Κατά την τμηματοποίηση της πρόσοψης τοποθετήθηκαν σε στρατηγικά σημεία ειδικές υποδοχές για την ένωση των εκτυπωμένων τμημάτων, με χρήση εμφωλευμένων παξιμαδιών, βιδών και εγκοπών.

Η τρισδιάστατη εκτύπωση των τμημάτων έγινε σειριακά για την εξάλειψη της πιθανότητας επανάληψης τους, σε περίπτωση αποτυχίας, εφόσον τα τμήματα αναμένονταν να καταναλώσουν μεγάλη ποσότητα υλικού. Ακόμα, επιλέχθηκε συγκεκριμένος προσανατολισμός εκτύπωσης για την επίτευξη της βέλτιστης αντοχής των οδηγών των πορτών, αλλά και η καλύτερη δυνατή υφή, και συνεπώς αισθητική, της εξωτερικής (εμφανούς) πλευράς των τμημάτων.

Μετά την ολοκλήρωση της εκτύπωσης των τμημάτων, ξεκίνησε η διαδικασία της συναρμολόγησης τους. Ως πρώτο βήμα τοποθετήθηκαν παξιμάδια στις προβλεπόμενες θέσεις των ενώσεων και στην συνέχεια έγινε η συναρμολόγηση των τμημάτων του μισού πάνω μέρους της πρόσοψης. Έπειτα, πραγματοποιήθηκε η κοπή των πορτών πλεξιγκλάς στο κατάλληλο μέγεθος, πάνω στις οποίες έγιναν οι κατάλληλες τρύπες όπου στηρίζονται οι μηχανισμοί κλειδώματος. Έπειτα οι πόρτες εισάγονται συρταρωτά στους οδηγούς του πάνω μισού μέρους της πρόσοψης, ώστε με την συναρμολόγηση και ένωση του κάτω μέρους της πρόσοψης με το πάνω μέρος, να «εγκλωβίζονται» με ασφάλεια οι πόρτες στους οδηγούς, χωρίς να είναι δυνατή η αφαίρεση τους.

Έτσι ολοκληρώνεται η κατασκευή της πρόσοψης του αποθηκευτικού χώρου του ρομποτικού σερβιτόρου. Επόμενο βήμα είναι η εγκατάστασή της στο υπόλοιπο σώμα του ρομπότ μέσω των ειδικά σχεδιασμένων υποδοχών που βρίσκονται στο εσωτερικό μέρος της.



Εικόνα 19: Μηχανισμός κλειδώματος πόρτας

4.2.2 Κατασκευή και εγκατάσταση πλευρικών τμημάτων

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, η στήριξη του σώματος του ρομποτικού σερβιτόρου γίνεται χάρη στον μεταλλικό σκελετό. Για να επιτευχθεί όμως αυτό, πρέπει με κάποιον τρόπο να επιτευχθεί η στερέωση των τρισδιάστατα εκτυπωμένων τμημάτων, επάνω στις αντηρίδες του μεταλλικού σκελετού. Τον σκοπό αυτό, εξυπηρετούν τέσσερα πλευρικά πλαστικά τμήματα. Τα τμήματα αυτά διαθέτουν τέσσερις υποδοχές για βίδες το καθένα, οι οποίες αντιστοιχούν σε υποδοχές του μεταλλικού σκελετού. Ο λόγος για τον οποίο σχεδιάστηκαν τέσσερα τμήματα, είναι για ακόμη μια φορά οι διαστάσεις εκτύπωσης του τρισδιάστατου εκτυπωτή. Αυτή τη φορά μάλιστα κατέλαβαν όλο τον διαθέσιμο χώρο στους άξονες X και Y καθώς είναι τετράγωνου σχήματος και συνεπώς οι εκτυπώσεις, παρά το γεγονός ότι τα τμήματα ήταν μικρού πάχους, διήρκεσαν αρκετό χρόνο. Μετά την εκτύπωσή τους τα τμήματα εγκαταστάθηκαν στον μεταλλικό σκελετό (βλ. Εικόνα 183).

4.3 Υλοποίηση ηλεκτρονικών

Η κατασκευή του ηλεκτρονικού σκέλους του ρομποτικού σερβιτόρου, έγινε με βάση την σχεδίαση που αναλύθηκε στο προηγούμενο κεφάλαιο. Ιδιαίτερη έμφαση κατά την κατασκευή δόθηκε στη χρήση μεθόδων και πρακτικών, που διασφαλίζουν την ποιότητα, την αξιοπιστία (σε βαθμό που επιτρέπει ένα πρωτότυπο) και συνεπώς την ασφάλεια των ηλεκτρονικών συστημάτων του ρομποτικού σερβιτόρου. Παρακάτω θα αναλυθεί η υλοποίηση των επιμέρους ηλεκτρονικών τμημάτων, ενώ στο τέλος του κεφαλαίου παρατίθεται πίνακας με λίστα των βασικών υλικών που χρειάστηκαν.

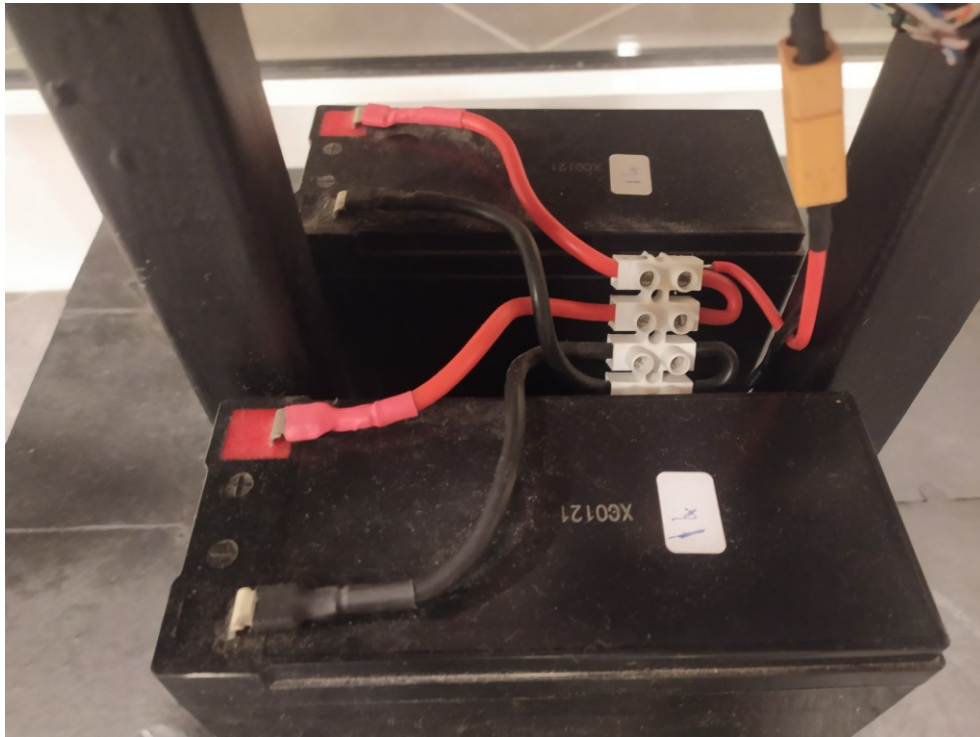
4.3.1 Κινητήρες, πίνακας ηλεκτρονικών και μπαταρίες

Το μεγαλύτερο μέρος των ηλεκτρονικών του ρομπότ, βρίσκεται κοντά στην βάση του. Αποτελείται, με σειρά ύψους τοποθέτησης, από τους βηματικούς κινητήρες, τις μπαταρίες και από τον πίνακα ηλεκτρονικών (ο οποίος βρίσκεται στο ίδιο ύψος με τους αισθητήρες απόστασης).

Οι κινητήρες του ρομποτικού σερβιτόρου, είναι δύο βηματικοί κινητήρες μοντέλου 42BYGHW615AG5 οι οποίοι έχουν ενσωματωμένο κιβώτιο γραναζιών. Για την στερέωση τους στη βάση του ρομπότ χρησιμοποιήθηκαν L-brackets στα οποία βιδώθηκαν με μεγάλου μήκους βίδες οι κινητήρες. Τα καλώδια των κινητήρων στη συνέχεια εξέρχονται από ειδικές υποδοχές της μεταλλικής βάσης και φθάνουν στον πίνακα ηλεκτρονικών, όπου συνδέονται σε επέκταση, η οποία καταλήγει στο CNC Shield που βρίσκεται στο εσωτερικό του πίνακα.

Οι μπαταρίες του ρομποτικού σερβιτόρου είναι στερεωμένες πάνω στην βάση του μεταλλικού σκελετού. Πρόκειται για ένα ζεύγος επαναφορτιζόμενων μπαταριών μολύβδου-οξέος κλειστού τύπου, τάσης 12V και 7,2 Ah η μία. Οι μπαταρίες αυτές συνδέθηκαν παράλληλα με χρήση ηλεκτρολογικής κλέμας ώστε να προσομοιώνουν μία μπαταρία τάσης 12V και συνολικής χωρητικότητας 14,4 Ah. Για την μεταξύ τους σύνδεση, αλλά και για την

παροχή στον πίνακα, χρησιμοποιήθηκε καλώδιο 12 AWG (διατομής αγωγού 2,06 mm), ενώ συνδέονται στον πίνακα με χρήση συνδέσμου τύπου XT60. Τα υλικά αυτά επιλέχθηκαν ώστε να υπάρχει αυξημένη αντοχή σε συνθήκες υψηλής κατανάλωσης ρεύματος.

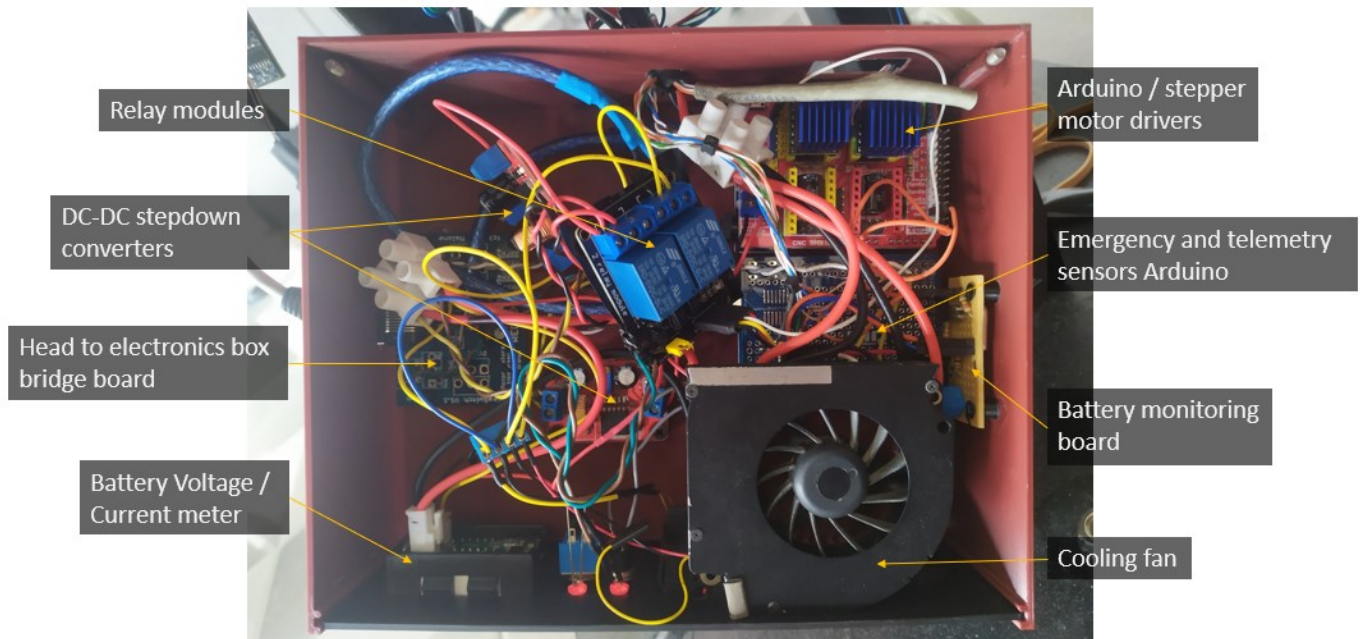


Εικόνα 20: Οι μπαταρίες και η συνδεσμολογία τους

Αρχικά, πριν από την κατασκευή του πίνακα, τα ηλεκτρονικά (όπως Arduino, DC-DC converters κ.α.) συνδέθηκαν με πρόχειρο τρόπο και εξωτερικά του, με σκοπό την πραγματοποίηση δοκιμών για τυχόν αλλαγές ή βελτιώσεις στα εξαρτήματα αυτά. Μετά από τις δοκιμές αυτές ξεκίνησε η υλοποίηση της τελικής έκδοσης του πίνακα, με πρώτο βήμα την εκτύπωση των εξαρτημάτων του. Πιο συγκεκριμένα τυπώθηκαν το καπάκι και το κύριο κοντί του πίνακα, το πάνελ ελέγχου μαζί με το προστατευτικό του κάλυμμα, όπως και αποστάτες για στερέωση των διάφορων πλακετών στο εσωτερικό του πίνακα. Ο πίνακας εκτυπώθηκε από PLA χρώματος μπορντό, ενώ το πίσω πάνελ εκτυπώθηκε με μαύρο, δημιουργώντας μια αντίθεση για καλύτερη εμφάνιση των οργάνων του πάνελ.

Στη συνέχεια αποσυνδέθηκαν οι πλακέτες από την πρόχειρη διάταξη, ώστε να ξεκινήσει η εγκατάστασή τους στο εσωτερικό του πίνακα. Αφού οι πλαστικοί αποστάτες βιδώθηκαν σε κάθε πλακέτα, στη συνέχεια κολλήθηκαν στον πάτο του πίνακα με κόλλα στιγμής, στις προβλεπόμενες θέσεις τους βάση του σχεδίου. Ύστερα προσαρμόστηκαν τα όργανα ελέγχου στο πάνελ του πίνακα.

Τέλος, έγιναν όλες οι απαραίτητες συνδεσμολογίες με χρήση κολλήσεων αλλά και βιδωτών τερματικών. Εδώ δόθηκε ιδιαίτερη έμφαση στην ασφάλεια της καλωδίωσης, για την αποφυγή πιθανών βλαβών, όπως αποσυνδέσεις ή βραχυκυκλώματα από επαφή αγωγών. Για τον λόγο αυτό χρησιμοποιήθηκαν διάφορα μέσα, όπως για παράδειγμα μόνωση κολλήσεων με θερμοσυστελλόμενα σωληνάρια. Η τελική μορφή του πίνακα ηλεκτρονικών φαίνεται σε παρακάτω φωτογραφία, όπου επιδεικνύονται και τα βασικότερα εξαρτήματά του.



Εικόνα 21: Πίνακας ηλεκτρονικών ρομποτικού σερβιτόρου

Η φόρτιση των μπαταριών γίνεται με ειδικό φορτιστή ρεύματος, ο οποίος συνδέεται σε θύρα που είναι τοποθετημένη στο πάνελ του πίνακα ηλεκτρονικών. Ο φορτιστής αυτός παρακολουθεί συνεχώς την τάση των μπαταριών ώστε να τις φορτίζει «έξυπνα». Για τον λόγο αυτό δεν συνιστάται η φόρτιση ενώ το ρομπότ είναι ενεργό, καθώς επηρεάζεται η τάση των μπαταριών.

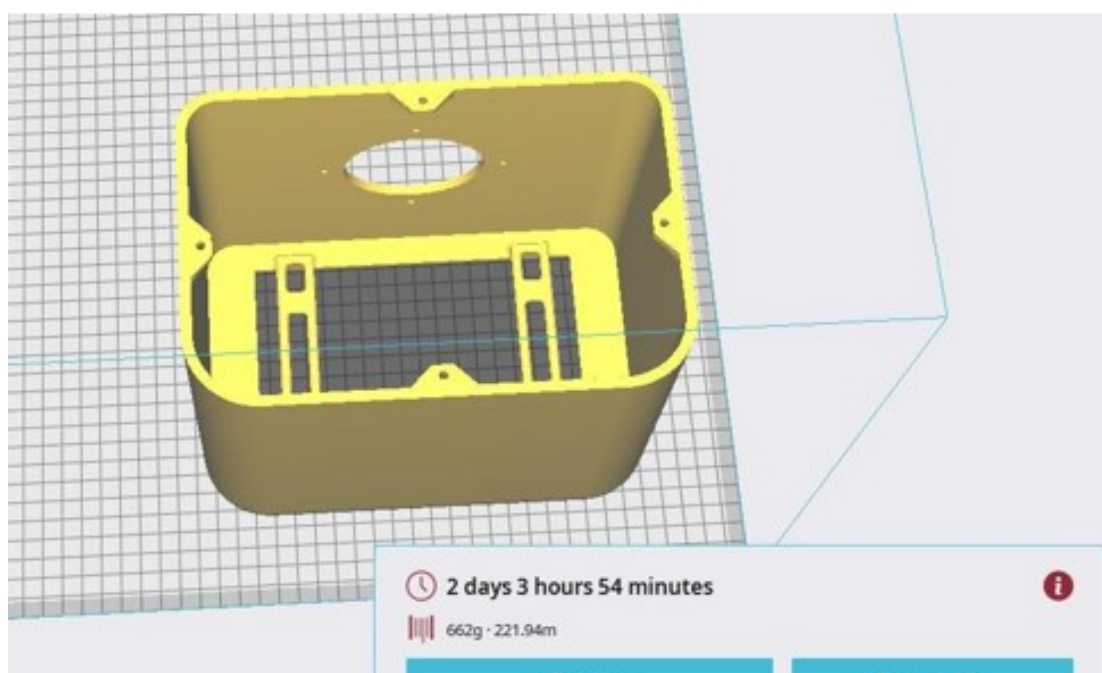


Εικόνα 22: Φόρτιση μπαταριών και πάνελ πίνακα ηλεκτρονικών

4.3.2 Κατασκευή κεφαλιού

Η υλοποίηση του κεφαλιού του ρομποτικού σερβιτόρου έγινε σε μία φάση, η οποία συμπεριλάμβανε την εκτύπωση του κεφαλιού και την εγκατάσταση των ηλεκτρονικών του. Μετά την εκτύπωσή του, και την προσαρμογή του RPI επάνω στην οθόνη αφής, η οθόνη αφής εγκαταστάθηκε στο πρόσωπο του κεφαλιού, με χρήση βιδών σε ειδικά προσχεδιασμένες αντηρίδες στο εσωτερικό του κεφαλιού.

Έπειτα το κεφάλι στερεώθηκε στο πάνω μέρος του λαιμού, ο οποίος με την σειρά του βιδώθηκε στον μεταλλικό σκελετό. Στο εσωτερικό του λαιμού εγκαταστάθηκε ο μετατροπέας τάσης που είναι υπεύθυνος για την τροφοδοσία του RPI. Στη συνέχεια κατασκευάστηκε η καλωδίωση διασύνδεσης πίνακα ηλεκτρονικών και κεφάλι. Η καλωδίωση αυτή (βλ. Εικόνα 25) αποτελείται όπως προαναφέρθηκε από ένα καλώδιο UTP τεσσάρων συνεστραμμένων ζευγών και από ένα εξωτερικό ζεύγος το οποίο μεταφέρει τάση μπαταριών και συνδέεται στον μετατροπέα τάσης.



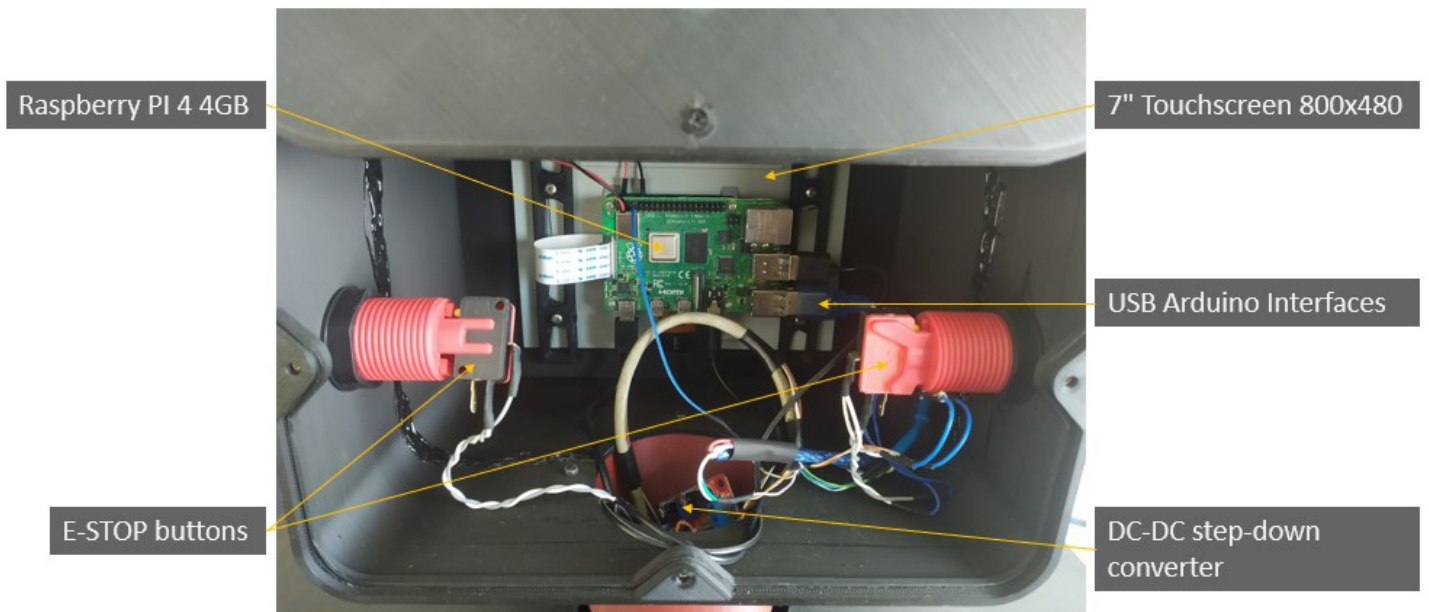
Εικόνα 23: Εκτύπωση κεφαλιού

Πίνακας 3: Λίστα βασικού ηλεκτρονικού εξοπλισμού ρομποτικού σερβιτόρου

ΕΙΔΟΣ	ΠΟΣΟΤΗΤΑ
Stepper motor with gearbox 42BYGHW615AG5	2
Arduino CNC Shield V3	1
Arduino Proto Shield	1
TMC2209 Stepper Motor Driver	2
Raspberry Pi 7" Touchscreen Display (Official)	1
Raspberry Pi 4 Model B 4GB	1
Arduino Uno	2
Dual channel relay module	1
Rubber Wheel 49	2
DC-DC step down module	3
Power Switch 220V Red	1
Caster ball	1
Stepper motor L mounting bracket	2
Lead Acid Battery 12V 7Ah	2
Battery Charger B6AC	1
Voltage-Current meter	1
Push switch	1
Big red tactile push buttons	2
DC barrel jack socket	1
XT60 male	1
XT60 female	1
Dean connector male - female	1
RJ45 breakout board	1
USB Type A to Type B cable	2
Cat5e UTP cable	1
Cooling fan	1
Battery monitoring custom pcb	1
Ultrasonic distance sensor	2



Εικόνα 25: Καλωδίωση διασύνδεσης πίνακα ηλεκτρονικών με κεφάλι



Εικόνα 24: Εσωτερικό κεφαλιού ρομποτικού σερβιτόρου

5

Κίνηση ρομποτικού σερβιτόρου

Όπως και ένας πραγματικός σερβιτόρος, έτσι και ένας ρομποτικός σερβιτόρος πρέπει να μπορεί να κινείται στον χώρο ώστε να εκτελεί τα καθήκοντά του. Πιο συγκεκριμένα πρέπει να είναι σε θέση να εκτελεί διεργασίες, όπως το να πηγαίνει στα τραπέζια και να λαμβάνει παραγγελίες από τους πελάτες, να μεταφέρει παραγγελίες από την κουζίνα στα τραπέζια και να αποφεύγει πιθανά εμπόδια που βρίσκονται στο πέρασμά του. Στο κεφάλαιο αυτό θα αναλυθεί η κίνηση του ρομποτικού σερβιτόρου.

5.1 Βηματικοί κινητήρες

Για να μπορεί ο ρομποτικός σερβιτόρος να εκτελεί τις διαδρομές κάθε τραπεζιού, χρειάζεται μεγάλη ακρίβεια στις κινήσεις του, ώστε να ακολουθεί πιστά και επαναλαμβανόμενα τα βήματα ενός προκαθορισμένου μονοπατιού. Όμως, ένας απλός ηλεκτρικός κινητήρας, όπως για παράδειγμα ένα κινητήρας DC, χαρακτηρίζεται από έλλειψη ακρίβειας στην κίνηση του (τουλάχιστον χωρίς την προσθήκη επιπλέον αισθητήρων ανατροφοδότησης, όπως περιστροφικός κωδικοποιητής ή rotary encoder). Αυτό συμβαίνει λόγω αδράνειας κατά την εκκίνηση ή σταμάτημα της κίνησης, όπου παρατηρείται τυχαία ολίσθηση ή από αδυναμία εντοπισμού της ολίσθησης αυτής με σκοπό την διόρθωση της. Ως αποτέλεσμα, ο ρομποτικός σερβιτόρος δεν θα ήταν ικανός να σταματά και να ξεκινά σε μία επιθυμητή θέση με ακρίβεια, γεγονός που θα οδηγούσε σε γρήγορη και σταδιακή απόκλιση από το προκαθορισμένο μονοπάτι.

Τα παραπάνω προβλήματα οδήγησαν στην ανάγκη αναζήτησης ενός εναλλακτικού είδους κινητήρα, ο οποίος θα είναι σε θέση να κάνει σταδιακές και ακριβείς κινήσεις, κρατώντας ταυτοχρόνως την επιθυμητή του θέση. Ιδανικός για τις προδιαγραφές αυτές κρίθηκε ένας κινητήρας που λέγεται βηματικός. Όπως μπορεί κανείς να συμπεράνει και από το όνομά του, ο κινητήρας αυτός λειτουργεί εναλλάσσοντας το μαγνητικό πεδίο των εσωτερικών του πηνίων καταφέροντας να κάνει σταδιακές και μικρές περιστροφικές κινήσεις που λέγονται steps (βήματα). Έτσι έχει πλήρη έλεγχο της θέσης του ανά πάσα στιγμή, καθώς και ικανότητα αντίστασης σε εξωτερικές δυνάμεις που προσπαθούν να αλλάξουν την θέση αυτή. Συνεπώς στην περίπτωση του ρομποτικού σερβιτόρου, μπορεί να αντισταθεί στην αδράνεια που συνεπάγεται το μεγάλο βάρος της κατασκευής.

Στην υλοποίηση του ρομποτικού σερβιτόρου επιλέχθηκαν δύο βηματικοί κινητήρες μοντέλου 42BYGHW615AG5 οι οποίοι έχουν ενσωματωμένο κιβώτιο γραναζιών. Το κιβώτιο αυτό κρίθηκε κατάλληλο, καθώς χάρη σε μείωση σχέσης, αυξάνει σημαντικά την τελική ροπή του κινητήρα, ώστε να ανταπεξέρχεται στο βάρος της συνολικής κατασκευής κατά την κίνηση.



Εικόνα 26: Βηματικός κινητήρας με ενσωματωμένο μειωτήρα

5.2 Οδηγοί βηματικών κινητήρων

Σε αντίθεση με έναν κλασικό ηλεκτρικό κινητήρα, οι βηματικοί κινητήρες δεν μπορούν να λειτουργήσουν με απλή τροφοδοσία ρεύματος. Χρειάζονται ειδικά κυκλώματα, που λέγονται οδηγοί, τα οποία τροφοδοτούν τα πηνία τους με κατάλληλα διαμορφωμένους χρονικά παλμούς, οι οποίοι καθορίζουν τον βηματισμό του ρότορα και κατ' επέκταση την φορά περιστροφής και την ταχύτητα το κινητήρα. Τα κυκλώματα αυτά υπάρχουν τις περισσότερες φορές σε μορφή ολοκληρωμένου κυκλώματος, το οποίο χειρίζεται από εξωτερικό κύκλωμα, όπως είναι ένας μικροελεγκτής.

Ο ρομποτικός σερβιτόρος χρησιμοποιεί δύο από αυτούς τους οδηγούς (έναν για κάθε βηματικό κινητήρα), μοντέλου TMC2209. Το μοντέλο αυτό είναι από τα πιο σύγχρονα που κυκλοφορούν στην αγορά, και επιλέχθηκε καθώς φέρει ορισμένα πλεονεκτήματα σε σχέση με υπόλοιπα. Ένα από αυτά είναι η ικανότητα του για αθόρυβη οδήγηση (silent drive), όπου το επιτυγχάνει αξιοποιώντας μια προχωρημένη τεχνική που λέγεται μικροβηματισμός (microstepping), κατά την οποία ο οδηγός κινεί τον κινητήρα σε υποδιαιρέσεις των φυσικών του βημάτων. Αυτό το καταφέρνει ρυθμίζοντας την ισχύ του μαγνητικού πεδίου κάθε πηνίου, «ισορροπώντας» έτσι τον ρότορα μεταξύ δύο κανονικών βημάτων, πολλαπλασιάζοντας την ακρίβεια περιστροφής και σε ορισμένες περιπτώσεις αυξάνοντας την τελική ροπή του βηματικού κινητήρα. Επίσης, οι οδηγοί αυτοί αντέχουν περισσότερο ρεύμα οδήγησης και είναι πιο αποδοτικοί, κάτι που συνεπάγεται μεγαλύτερη θερμική αντοχή και αποφυγή υπερθέρμανσης. Αυτό κρίθηκε απαραίτητο κατά τις δοκιμές κίνησης του ρομποτικού σερβιτόρου, με λιγότερο αποδοτικούς οδηγούς.

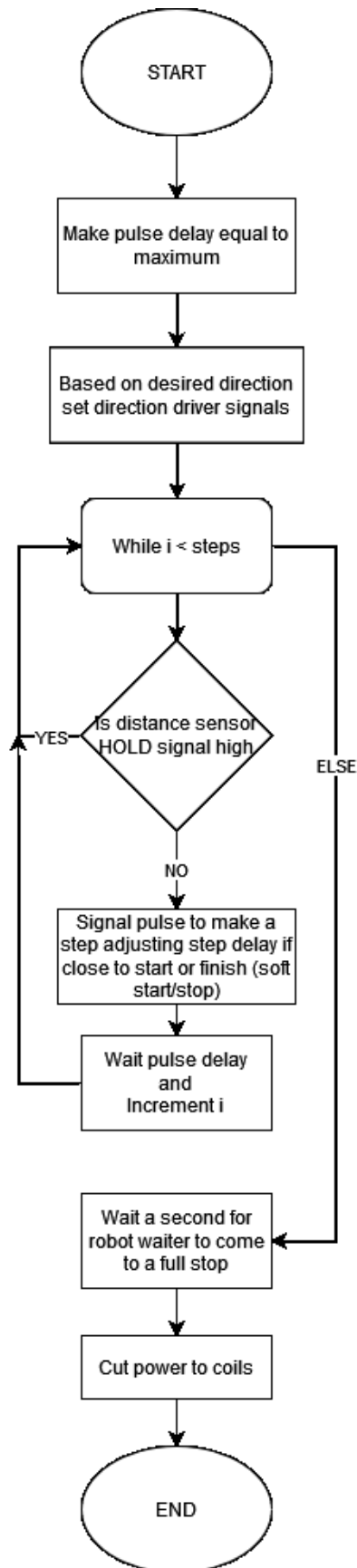
Οι οδηγοί αυτοί ελέγχονται απευθείας από την Arduino κίνησης, η οποία στέλνει σήμα για κάθε ένα από τα βήματα, αλλά και για την κατεύθυνση περιστροφής και το σήμα ενεργοποίησης. Το σήμα ενεργοποίησης είναι ένα ειδικό σήμα, που ανάλογα την λογική του τιμή ελέγχει το αν οι οδηγοί θα τροφοδοτούν τα πηνία των κινητήρων με ρεύμα, και συνεπώς το αν οι κινητήρες θα κρατούν την θέση τους. Όμως κατά την διάρκεια της τροφοδοσίας των πηνίων, καταναλώνεται πολύ ενέργεια και ανεβαίνει η θερμοκρασία τους. Για τον λόγο αυτό, ο κώδικας βηματισμού, μετά την ολοκλήρωση μιας κίνησης, περιμένει μερικά δευτερόλεπτα για την πλήρη ακινητοποίηση του ρομποτικού σερβιτόρου (κατά την οποία οι κινητήρες αντιστέκονται στην αδράνεια του) και στην συνέχεια κάνοντας χρήση του σήματος ενεργοποίησης (enable) απενεργοποιούν τους κινητήρες.



Εικόνα 27: Οδηγός βηματικού κινητήρα TMC2209

5.3 Αλγόριθμος κίνησης

Για την εκτέλεση κινήσεων μέσω ελέγχου των οδηγών των βηματικών κινητήρων από την Arduino, αναπτύχθηκε αλγόριθμος, ο οποίος αναλαμβάνει τον βηματισμό, ενώ λαμβάνει υπόψη όλες τις ανάγκες της κίνησης του ρομποτικού σερβιτόρου. Ο αλγόριθμος αυτός έχει τρεις ελαφρώς διαφορετικές εκδόσεις αναλόγως με την κατεύθυνση και την σηματοδότηση των οδηγών ανάλογα με το είδος της κίνησης και το πως συμμετέχει ο κάθε κινητήρας σε αυτή. Όμως η βασική δομή του παραμένει ίδια. Αρχικά λαμβάνουμε ως παραμέτρους την κατεύθυνση της κίνησης και τον αριθμό βημάτων. Βάση της κατεύθυνσης ρυθμίζουμε τις εξόδους της Arduino που συνδέονται στις εισόδους κατεύθυνσης των οδηγών για να ορίσουμε την επιθυμητή κατεύθυνση της κίνησης. Έπειτα αρχικοποιείται η μεταβλητή καθυστέρησης παλμού βήματος στο μέγιστο. Η μεταβλητή αυτή αυξομειώνεται σταδιακά, ανάλογα με το αν βρισκόμαστε στα πρώτα ή στα τελευταία βήματα της κίνησης, ώστε να γίνεται ομαλό ξεκίνημα (με μεγαλύτερη ροπή), ή σταμάτημα. Στη συνέχεια ο αλγόριθμος εισέρχεται σε βρόγχο ο οποίος εκτελείται όσο εκκρεμούν βήματα, τα οποία παρακολουθούνται με βοηθητική μεταβλητή ακεραίου. Στην αρχή κάθε επανάληψης, γίνεται έλεγχος του σήματος HOLD ώστε να μην συνεχίζεται η κίνηση αν κάποιος αισθητήρας απόστασης εντοπίζει εμπόδιο. Το σήμα αυτό προέρχεται από την Arduino αισθητήρων. Στις πρώτες και τελευταίες επαναλήψεις αυξομειώνεται η καθυστέρηση παλμού, ενώ στην συνέχεια εκτελείται το βήμα πραγματοποιώντας ανάλογη σηματοδότηση στους οδηγούς μέσω των εξόδων της Arduino. Μετά από την εκτέλεση του βήματος αυξάνουμε την μεταβλητή βημάτων κατά ένα. Τέλος όταν ολοκληρωθούν όλα τα βήματα της κίνησης, ο αλγόριθμος περιμένει κάποια δευτερόλεπτα να σταματήσει εντελώς ο ρομποτικός σερβιτόρος, και στην συνέχεια απενεργοποιεί (με χρήση του σήματος ENABLE των οδηγών) την τροφοδοσία των κινητήρων.



Σχήμα 2: Διάγραμμα ροής αλγορίθμου βηματισμού

6

Δοκιμές σε χώρο εστίασης

Με την ολοκλήρωση της κατασκευής του αρχικού πρωτότυπου του συστήματος και τον προγραμματισμό του, ακολούθησαν δοκιμές σε πραγματικό χώρο εστίασης. Για την πραγματοποίηση των δοκιμών ο ρομποτικός σερβιτόρος μεταφέρθηκε σε μαγαζί, το οποίο βρίσκεται στο λιμάνι Καρλοβάσιου Σάμου και λέγεται «Βάλσαμο». Κατά την διάρκεια της παραμονής του στον χώρο αυτό, έγιναν διάφορες μετατροπές και βελτιώσεις, καθώς και εκπαίδευση του προσωπικού πάνω στη χρήση του συστήματος.

Αρχικά, ο κατάλογος προϊόντων του καταστήματος περάστηκε στο σύστημα παραγγελιών, ώστε να είναι δυνατή η χρήση του από τους πελάτες. Στη συνέχεια ο ρομποτικός σερβιτόρος τοποθετήθηκε σε ειδικά επιλεγμένη θέση, που αναπαριστά το «σπίτι» του. Βάση αυτής της θέσης και των θέσεων τεσσάρων δοκιμαστικών τραπεζιών, σχεδιάστηκε ένα πλάνο που αναπαριστά την κάτοψη του καταστήματος, στο οποίο απεικονίζονται οι θέσεις αυτές. Το πλάνο αυτό, φορτώθηκε στην οθόνη επιλογής τραπεζιού του συστήματος παραγγελιών, για χρήση από τους πελάτες. Εκτός από τις προσθήκες αυτές, έγινε γενικότερη βελτίωση και τελειοποίηση του συστήματος παραγγελιών. Μια από τις σημαντικότερες βελτιώσεις, αποτέλεσε η εκ νέου υλοποίηση του συστήματος ομιλίας στην τελική μορφή που αναλύθηκε σε προηγούμενο κεφάλαιο.

Έπειτα, ξεκίνησαν οι δοκιμές της κίνησης για την παράδοση παραγγελιών στα τραπέζια. Για τον λόγο αυτό και κάνοντας χρήση της λειτουργίας χειροκίνητου χειρισμού της διεπαφής τεχνικής διαχείρισης, έγινε σταδιακά ο προγραμματισμός κινήσεων για κάθε ένα από τα τέσσερα δοκιμαστικά τραπέζια. Για την ολοκλήρωση αυτού του βήματος, πραγματοποιούνταν κινήσεις κατ' εντολή με συγκεκριμένο αριθμό βημάτων. Με την εύρεση του ιδανικού αριθμού βημάτων για κάθε κίνηση της διαδρομής, και συνεπώς οι απαραίτητες κινήσεις για την ολοκλήρωση της, αυτές αποθηκεύονταν στην βάση δεδομένων σαν

ακολουθία κινήσεων για τον αντίστοιχο τραπέζι. Κατά την διάρκεια αυτής της διαδικασίας και γενικότερα της κίνησης παρατηρήθηκαν διάφορα προβλήματα που μαζί με τον τρόπο επίλυσης τους θα αναλυθούν στο κεφάλαιο των συμπερασμάτων και βελτιώσεων.

Ύστερα από τον προγραμματισμό κινήσεων, ακολούθησε η εκπαίδευση του προσωπικού, πάνω στην χρήση και χειρισμό του ρομποτικού σερβιτόρου. Για τη διευκόλυνση της διαδικασίας αυτής, έγινε συγγραφή και εκτύπωση ενός βασικού φυλλαδίου οδηγιών χρήσης, το οποίο εξηγεί τον τρόπο χειρισμού των βασικών λειτουργιών του ρομπότ, όπως είναι η φόρτιση, η διαδικασία ενεργοποίησης και η απενεργοποίησης, η χρήση του συστήματος έκτακτης ανάγκης, κ.α.

ΓΡΗΓΟΡΟΣ ΟΔΗΓΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΡΟΜΠΟΤ-ΣΕΡΒΙΤΟΡΟΥ	
<u>Διαδικασία φόρτισης:</u> - Βεβαιωθείτε ότι το ρομπότ είναι πλήρως απενεργοποιημένο και ότι ο κόκκινος γενικός διακόπτης είναι στην θέση (Ο). - Συνδέετε τον φορτιστή στην πρίζα και περιμένετε μέχρι να ανάψει. - Συνδέετε το βύσμα του φορτιστή στην θύρα φόρτισης στο πίσω κάτω μέρος του ρομπότ, ελέγχοντας ότι έχει εισέλθει ολόκληρο και όχι μερικώς στην θύρα. - Κρατάτε πατημένο το κουμπί "START" του φορτιστή για περίπου 3 δευτερόλεπτα έως να ακούσετε τον χαρακτηριστικό ήχο. - Αν εμφανιστεί η ένδειξη "Charging" στην οθόνη του φορτιστή σημαίνει ότι η φόρτιση έχει ξεκινήσει επιτυχώς. Σε διαφορετική περίπτωση (π.χ. ακούγονται παραπάνω ήχοι συνοδευόμενοι από μηνύματα σφάλματος στην οθόνη του φορτιστή) αποσυνδέετε τον φορτιστή από το ρεύμα και από το ρομπότ και επικοινωνείτε με τον κατασκευαστή. - Η διάρκεια φόρτισης διαρκεί 3-4 ώρες. Μετά τον χρόνο αυτό ο φορτιστής θα ειδοποιήσει με ηχητικό σήμα. Όταν συμβεί αυτό αποσυνδέετε τον φορτιστή από το ρεύμα και το ρομπότ. Στην συνέχεια ελέγχετε με μια ενεργοποίηση του ρομπότ, ότι η στάθμη της μπαταρίας (βλέπε ενότητα διαδικασία ενεργοποίησης) δεν βρίσκεται κάτω από τα 12.7 V. <u>Διαδικασία ενεργοποίησης:</u> - Γυρνάτε τον γενικό κόκκινο διακόπτη στην θέση Ι, ώστε να εισέλθει το ρομπότ σε διαδικασία εκκίνησης. - Βεβαιώσετε ότι η τάση των μπαταριών, όπως αναγράφεται στην πάνω μπλέ (V) ένδειξη του μετρητή ισχύος, δεν βρίσκεται κάτω από τα 12.3V. Στην περίπτωση που η τάση βρίσκεται κάτω από τα 12.3V, τερματίζετε την λειτουργία ακολουθώντας την ενότητα «Διαδικασία τερματισμού λειτουργίας» και φορτίζετε άμεσα, καθώς ενδέχεται να προκληθεί βλάβη στις μπαταρίες! - Περιμένετε έως ότου φανεί στην μπροστινή οθόνη το πρόσωπο του ρομπότ. Στην περίπτωση που η οθόνη μετά από χρονική διάρκεια 1 λεπτού δεν έχει ενεργοποιηθεί επικοινωνείτε με τον κατασκευαστή. - Η διαδικασία ενεργοποίησης έχει ολοκληρωθεί και το ρομπότ είναι έτοιμο για χρήση. <u>Διαδικασία τερματισμού λειτουργίας:</u> - Επιλέγεται από την οθόνη του ρομπότ το εικονίδιο κλειδιού στο κάτω δεξιά μέρος της οθόνης παραγγελιών, στην συνέχεια εισάγετε τον κωδικό PIN διαχείρισης και πατάτε ENTER. Αν ο κωδικός ήταν σωστός θα πρέπει πλέον να βρίσκεται στην οθόνη διαχείρισης του ρομπότ, σε διαφορετική περίπτωση θα παραμείνετε στην κύρια οθόνη.	- Επιλέγεται SHUTDOWN και στην συνέχεια επιβεβαιώνετε την επιλογή σας για τερματισμό λειτουργίας στο αντίστοιχο μήνυμα. - Το ρομπότ έχει πλέον εισέλθει σε κατάσταση ασφαλούς τερματισμού λειτουργίας. - Περιμένετε έως ότου η κόκκινη λυχνία στον κόκκινο γενικό διακόπτη στο πίσω μέρος του ρομπότ σβήσει και μετά από 1 λεπτό γυρνάτε τον διακόπτη στην θέση (Ο), ώστε να διακοπεί η παροχή ενέργειας και να επιτευχθεί εξοικονόμηση μπαταρίας. <u>Σημαντικές πληροφορίες ασφαλείας:</u> ΑΠΟΦΕΥΓΕΤΕ ΑΥΣΤΗΡΩΣ ΤΗΝ ΕΚΘΕΣΗ ΤΟΥ ΡΟΜΠΟΤ ΣΤΟΝ ΗΛΙΟ! - Το ρομπότ διαθέτει σύστημα διακοπής κίνησης σε περίπτωση έκτακτης ανάγκης. Για να τεθεί σε λειτουργία πιέζετε οποιοδήποτε από τα κόκκινα κουμπιά με κίτρινο πλαίσιο, τα οποία είναι τοποθετημένα στην κεφαλή του ρομπότ. Όταν συμβεί αυτό, το ρομπότ θα ακινητοποιηθεί πλήρως και θα ανάψει στον πίσω πίνακα η λυχνία έκτακτης ανάγκης. Για να επανέλθει το ρομπότ σε κανονική λειτουργία, θα πρέπει να πατήσετε το κουμπί που βρίσκεται κάτω από την λυχνία. - Το ρομπότ διαθέτει σύστημα παρακολούθησης στάθμης μπαταριών και απενεργοποιείται αυτόματα στην περίπτωση χαμηλής στάθμης. Παρόλα αυτά βεβαιώσετε πάντα σε τακτά χρονικά διαστήματα ότι η τάση των μπαταριών του ρομπότ (βλέπε ενότητα διαδικασία ενεργοποίησης) δεν βρίσκεται κάτω από τα 12.3 V, καθώς ενδέχεται να δημιουργηθεί μόνιμη βλάβη στις μπαταρίες του ρομπότ. <u>Σημειώσεις:</u>

Εικόνα 28: Φυλλάδιο οδηγιών χρήσης ρομποτικού σερβιτόρου



Εικόνα 29: Δοκιμή ρομποτικού σερβιτόρου σε χώρο εστίασης

7

Συμπεράσματα και πιθανές βελτιώσεις

Κατά την διάρκεια της εκπόνησης της παρούσας διπλωματικής, προέκυψαν διάφορα ζητήματα που δημιούργησαν προβληματισμό και τελικά διαμόρφωσαν τις αποφάσεις που λήφθηκαν κατά την υλοποίηση του συστήματος, αλλά και το τελικό αποτέλεσμα.

Μία από τις σημαντικότερες προκλήσεις που παρουσιάστηκαν, αποτέλεσε εκείνη της επίτευξης της ομαλής και αδιάκοπης κίνησης του ρομποτικού σερβιτόρου. Από τις πρώτες κιόλας δοκιμές, έγινε αντιληπτό ότι το μέγεθος και συνεπώς το βάρος της κατασκευής, θα δυσκόλευε σημαντικά την επιλογή εξαρτημάτων (και κυρίως κινητήρων), τα οποία θα ήταν σε θέση να συνδυαστούν για την υλοποίηση ενός συστήματος κίνησης, που θα ανταπεξέρχονταν στις απαιτήσεις και θα πετύχαιναν μια ακριβή κίνηση. Στις πρώτες δοκιμές, χρησιμοποιήθηκαν διάφοροι συνδυασμοί βηματικών κινητήρων και οδηγών, όπου κριτήρια τέθηκαν κυρίως η κατανάλωση ρεύματος, η τελική ροπή, η ευκολία οδήγησης και τέλος η ταχύτητα περιστροφής του κινητήρα. Τελικώς, αποφασίστηκε η υιοθέτηση των προαναφερθέντων κινητήρων και η χρήση των βηματικών κινητήρων DRV8825. Όμως, ακόμη και μετά από την φάση αυτή, η κίνηση συνέχισε να χαρακτηρίζεται από προβλήματα, όπως υψηλά επίπεδα θορύβου κατά την λειτουργία των κινητήρων, υπερθέρμανση των οδηγών (καθώς λειτουργούσαν στα όρια ρεύματος τροφοδοσίας) και αυτό κατά συνέπεια προκαλούσε τη διακοπή της κίνησης, αλλά και ολίσθηση βημάτων, γεγονός που απέτρεπε την ακρίβεια των κινήσεων. Ακόμη, το λογισμικό που υλοποιεί τον αλγόριθμο κίνησης, κατέληξε να είναι αρκετά πολύπλοκο και μη αποδοτικό, εξαιτίας της ανάγκης για συνεχή παραμετροποίηση και επιστράτευσης μη ιδανικών τεχνικών, ως προσπάθεια αντιστάθμισης των προβλημάτων υλικού.

Τα παραπάνω προβλήματα, έγιναν ακόμη πιο φανερά κατά την διεξαγωγή των δοκιμών στον πραγματικό χώρο εστίασης, γεγονός που αποδίδεται κυρίως στην αυξημένη

ανομοιομορφία του δαπέδου σε σχέση με αυτό του εργαστηρίου ρομποτικής (Το δάπεδο του χώρου εστίασης αποτελούνταν από ξύλινες σανίδες οι οποίες λύγιζαν με το βάρος του ρομπότ και προκαλούσαν ανωμαλίες), καθώς και στην αυξημένη θερμοκρασία περιβάλλοντος, λόγω καλοκαιρινής εποχής. Για τον λόγο αυτό και έπειτα από πολλή αναζήτηση, η λύση δόθηκε με την αναβάθμιση των οδηγών των βηματικών κινητήρων, καθώς και η εγκατάσταση ανεμιστήρα ψύξης στο εσωτερικό του πίνακα ηλεκτρονικών. Ως αντικαταστάτες επιλέχθηκαν οι οδηγοί του μοντέλου TMC2209, οι οποίοι είναι αθόρυβοι, ενώ αντέχουν περισσότερο ρεύμα, χάρη σε πιο αποδοτικό σχεδιασμό και σε μεγαλύτερου μεγέθους ψήκτρες αλουμινίου, οι οποίες σε συνδυασμό με τον ανεμιστήρα που ενεργοποιείται κατά την διάρκεια της κίνησης, είναι ικανές να τους ψύχουν επαρκώς. Η επίλυση των προβλημάτων αυτών, κατέστησε δυνατή την ανάπτυξη πιο αποδοτικού λογισμικού κίνησης.

Εκτός από τα ηλεκτρονικά κίνησης, ένας ακόμη παράγοντας που δυσκόλεψε στα αρχικά στάδια της υλοποίησης την αποτελεσματική κίνηση, ήταν το βάρος του μεταλλικού σκελετού. Αυτό οφείλονταν στο γεγονός ότι η πρώτη σχεδίαση και υλοποίηση του αφορούσε έναν διπλό πλαίσιο, το οποίο αποδείχθηκε υπέρβαρο. Το πρόβλημα αυτό επιλύθηκε μετά από ανασχεδιασμό του που οδήγησε στην τελική του μορφή.

Ακόμη ένα ζήτημα που χρειάστηκε επίλυση, αποτέλεσε η μη ανταπόκριση του αρχικού συστήματος ομιλίας στις προσδοκίες. Πιο συγκεκριμένα, ο ήχος του συστήματος ομιλίας χαρακτηριζόταν από πολύ κακή ποιότητα, ενώ με δυσκολία κανείς διέκρινε το κείμενο που καλούταν να εκφωνήσει. Αυτό αποδίδεται κυρίως στην χρήση ενός σχετικά απαρχαιωμένου λογισμικού σύνθεσης φωνής με το όνομα *espeak*, το οποίο ήταν ενσωματωμένο μέσω WASM σε βιβλιοθήκη JavaScript, ώστε να είναι δυνατή η εκτέλεση του στον browser του frontend. Η λύση που βρέθηκε, ήταν η αντικατάστασή του με εντελώς διαφορετικό λογισμικό, που χρησιμοποιεί online speech API και caching και αναλύεται στο κεφάλαιο του συστήματος παραγγελιών.

Κλείνοντας, υπάρχουν κάποιες ιδέες από τις οποίες θα μπορούσε να επωφεληθεί η ανάπτυξη μια πιθανής μελλοντικής υλοποίησης. Για παράδειγμα, ένα πρόβλημα που παρουσιάζει η υπάρχουσα κατασκευή, είναι η αδυναμία έκθεσής της στο φως του ηλίου. Το υλικό από το οποίο είναι εκτυπωμένα τα κομμάτια του ρομποτικού σερβιτόρου (PLA), λειώνει στους ~200 °C και γίνεται μαλακό στους ~60 °C. Έτσι, η έκθεσης των πλαστικών εξαρτημάτων στον ήλιο θα οδηγούσε στην παραμόρφωσή τους και συνεπώς όλης της κατασκευής. Το πρόβλημα αυτό, ίσως να αντιμετωπιζόταν με χρήση εναλλακτικών, πιο ανθεκτικών σε ψηλές θερμοκρασίες υλικών, όπως είναι το PETG, ή το ABS. Επίσης, ακόμη μια ιδέα για βελτίωση θα ήταν η χρήση αλουμινίου ως υλικό κατασκευής του μεταλλικού σκελετού, σε αντίθεση με σίδηρο. Αυτό θα οδηγούσε σε σημαντική μείωση του συνολικού βάρους της κατασκευής και συνεπώς στη βελτίωση της απόδοσης αλλά και της ακρίβειας της κίνησης. Ακόμη μια αναβάθμιση, θα ήταν η προθήκη τερματικού POS στο σώμα του ρομπότ, ώστε ο πελάτης να μπορεί να εξοφλήσει την παραγγελία του επιτόπου. Τέλος, η πλοήγηση του ρομποτικού σερβιτόρου θα επωφελούταν σημαντικά από την ύπαρξη κάποιας μεθόδου *localization*, η οποία ίσως να βασιζόταν σε εντοπισμό με χρήση πομπών, οι οποίοι θα βρίσκονταν εγκατεστημένοι σε κάθε τραπέζι. Αυτό θα καθιστούσε δυνατή την δυναμική

πλοήγηση από τραπέζι σε τραπέζι, μέσω του εντοπισμού τους, όπως επίσης θα επέτρεπε διόρθωση σε τυχόν αποκλίσεις των κινήσεων.

Βιβλιογραφία

- [1] Thanh, V. N., Vinh, D. P., & Nghi, N. T. (2019, August). Restaurant serving robot with double line sensors following approach. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)* (pp. 235-239). IEEE.
- [2] Wan, A., Foo, E., Lai, Z., Chen, H., & Lau, W. (2019). Waiter bots for casual restaurants. *Int. J. Robot. Eng*, 4, 018.
- [3] Garcia-Haro, J. M., Oña, E. D., Hernandez-Vicen, J., Martinez, S., & Balaguer, C. (2020). Service robots in catering applications: A review and future challenges. *Electronics*, 10(1), 47.
- [4] <https://soundoftext.com/#about>

Παράρτημα I Κώδικας Arduino αισθητήρων

```
/**
 * Emergency sensors for RobotWaiter
 * Dionysios Tsakarisanos
 */

#define ESTOP_PIN 2
#define RELAY_PIN 4
#define RESET_PIN 3

boolean is_triggered; // Emergency stop flag

/* Voltmeter */
int meter_pin = A0;
float value = 0.0;
float v = 0.0;

/* Ultrasonics */
int usonicl_trig_pin = 9;
int usonicl_echo_pin = 8;
int usonicro_trig_pin = 7;
int usonicro_echo_pin = 6;
int distance_l = 100;
int distance_r = 100;

/* Hold CNC */
int hold_cnc_pin = 5;
int distance_threshold = 50;

void emergency_ISR();
void reset_ISR();
void parseCommand(String);
int measureDistanceLeft();
```

```

int measureDistanceRight();

void setup() {
  pinMode(ESTOP_PIN, INPUT); // Set the estop button
  //pinMode(ESTOP_PIN, INPUT_PULLUP); // Set the estop button pull up resistor
  pinMode(RESET_PIN, INPUT_PULLUP); // Set the emergency reset button
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(meter_pin, INPUT);

  pinMode(sonicl_trig_pin, OUTPUT);
  pinMode(sonicl_echo_pin, INPUT);
  pinMode(sonicr_trig_pin, OUTPUT);
  pinMode(sonicr_echo_pin, INPUT);

  pinMode(hold_cnc_pin, OUTPUT);
  digitalWrite(hold_cnc_pin, HIGH);

  digitalWrite(RELAY_PIN, HIGH);
  is_triggered = false; // Initialize estop flag

  // Attach interrupts
  attachInterrupt(digitalPinToInterrupt(ESTOP_PIN), emergency_ISR, RISING);
  attachInterrupt(digitalPinToInterrupt(RESET_PIN), reset_ISR, FALLING);

  Serial.begin(115200); // Begin serial communication
}

String command;

void loop() {
  while(Serial.available() > 0) {
    char c = Serial.read();

    if (c == '\n') {
      parseCommand(command);
      command = "";
    } else {
      command += c;
    }
  }

  /*-----Begin Ultrasonic code-----*/
  int suml = 0;

```

```

int sumr = 0;
for (int i=0; i<5; i++) {
    suml += measureDistanceLeft();
    sumr += measureDistanceRight();
}
distance_l = suml/5;
distance_r = sumr/5;
if (distance_l <= distance_threshold || distance_r <= distance_threshold) {
    digitalWrite(hold_cnc_pin, LOW);
} else {
    digitalWrite(hold_cnc_pin, HIGH);
}
/*-----End Ultrasonic code-----*/
}

void parseCommand(String command) {
    String cmd;
    String arg1;
    String arg2;

    int index1 = command.indexOf(",");
    int index2 = command.indexOf(",", index1 + 1);
    int index3 = command.indexOf(",", index2 + 1);

    cmd = command.substring(0, index1);
    arg1 = command.substring(index1 + 1, index2);
    arg2 = command.substring(index2 + 1, index3);

    if (cmd.equalsIgnoreCase("identify")) {
        Serial.println("ARDUINO_EMERGENCY_SENSORS");
    } else if (cmd.equalsIgnoreCase("estop")) {
        emergency_ISR();
    } else if (cmd.equalsIgnoreCase("update")) {
        float maxV = 0.0;
        for (int i=0; i<5; i++) {
            value = analogRead(meter_pin);
            v = ((value*0.5)/1024.0)/0.00088;
            if (v >= maxV) { maxV = v; }
        }
        Serial.print("v="+String(maxV)+",e="+(is_triggered ? "1" : "0")+",ul="+distance_l+",ur="+distance_r+"\n");
    }
}

```

```
void emergency_ISR() {
    digitalWrite(RELAY_PIN, LOW); // Cut power
    is_triggered = true;
}

void reset_ISR() {
    digitalWrite(RELAY_PIN, HIGH); // resume power
    is_triggered = false;
}

int measureDistanceLeft() {
    digitalWrite(sonicl_trig_pin, LOW);
    delayMicroseconds(2);
    digitalWrite(sonicl_trig_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(sonicl_trig_pin, LOW);

    return pulseIn(sonicl_echo_pin, HIGH)*0.034/2;
}

int measureDistanceRight() {
    digitalWrite(sonicr_trig_pin, LOW);
    delayMicroseconds(2);
    digitalWrite(sonicr_trig_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(sonicr_trig_pin, LOW);

    return pulseIn(sonicr_echo_pin, HIGH)*0.034/2;
}
```


Παράρτημα II Κώδικας Arduino κίνησης

```
const int STEP_R = 2;
const int DIR_R = 5;

const int STEP_L = 3;
const int DIR_L = 6;

const int EN_PIN = 8;

const int HOLD_PIN = A1;
//const int pulseDelay = 1100;

const int maxStepSpeed = 800;
const int minStepSpeed = 400;
int currStepSpeed = minStepSpeed;
const int delayPerStep = 900;
const int targetPosPlus = 1000;
const int targetPosMinus = 0;

int minPulseDelay = 300; //600 1000
int maxPulseDelay = 300; //1300
int curPulseDelay = maxPulseDelay;
int pulseDelayStep = 2;
int pulseDelayRange = (maxPulseDelay - minPulseDelay)/pulseDelayStep;
int pulseDelay = 1100; //1100

void setup() {
  pinMode(STEP_R, OUTPUT);
  pinMode(DIR_R, OUTPUT);
  pinMode(STEP_L, OUTPUT);
  pinMode(DIR_L, OUTPUT);
  pinMode(EN_PIN, OUTPUT);
  pinMode(HOLD_PIN, INPUT);

  Serial.begin(115200);

  digitalWrite(EN_PIN, HIGH);
```

```

digitalWrite(DIR_R, LOW);
digitalWrite(DIR_L, HIGH);
}

String command;

void loop() {
  if(Serial.available()) {
    char c = Serial.read();

    if (c == '\n') {
      parseCommand(command);
      command = "";
    } else {
      command += c;
    }
  }
}

void parseCommand(String command) {
  String cmd;
  String arg1;
  String arg2;

  int index1 = command.indexOf(",");
  int index2 = command.indexOf(",", index1 + 1);
  int index3 = command.indexOf(",", index2 + 1);

  cmd = command.substring(0, index1);
  arg1 = command.substring(index1 + 1, index2);
  arg2 = command.substring(index2 + 1, index3);

  if (cmd.equalsIgnoreCase("move_both")) {
    moveBoth(arg1.toInt(), arg2.toInt());
  } else if (cmd.equalsIgnoreCase("move_left")) {
    moveLeft(arg1.toInt(), arg2.toInt());
  } else if (cmd.equalsIgnoreCase("move_right")) {
    moveRight(arg1.toInt(), arg2.toInt());
  } else if (cmd.equalsIgnoreCase("identify")) {
    Serial.println("ARDUINO_STEPPER_DRIVER");
  }

  Serial.println("Command: " + cmd);
  Serial.println("arg1: " + arg1);
  Serial.println("arg2: " + arg2);
}

void moveBoth(int direction, int steps) {

```

```

curPulseDelay = maxPulseDelay;

if (direction == 0) {
    digitalWrite(DIR_R, LOW);
    digitalWrite(DIR_L, LOW);
} else if (direction == 1) {
    digitalWrite(DIR_R, HIGH);
    digitalWrite(DIR_L, HIGH);
}

int i = 0;
while(i<steps) {
    if (digitalRead(HOLD_PIN) == LOW || direction == 1) {
        digitalWrite(EN_PIN, LOW);
        if (i < pulseDelayRange && curPulseDelay > minPulseDelay) {
            curPulseDelay -= pulseDelayStep;
        } else if (steps - i < pulseDelayRange && curPulseDelay < maxPulseDelay) {
            curPulseDelay += pulseDelayStep;
        }

        digitalWrite(STEP_R, HIGH);
        digitalWrite(STEP_L, HIGH);
        delayMicroseconds(curPulseDelay);
        digitalWrite(STEP_R, LOW);
        digitalWrite(STEP_L, LOW);
        delayMicroseconds(curPulseDelay);

        Serial.print('-');
        i++;
    } else {
        digitalWrite(EN_PIN, HIGH);
        delay(1000);
    }
}
Serial.print('.');
delay(500);
digitalWrite(EN_PIN, HIGH);
}

void moveRight(int direction, int steps) {
    curPulseDelay = maxPulseDelay;

    if (direction == 0) {
        digitalWrite(DIR_R, LOW);
        digitalWrite(DIR_L, HIGH);
    } else if (direction == 1) {
        digitalWrite(DIR_R, HIGH);
        digitalWrite(DIR_L, LOW);
    }
}

```

```

}

int i = 0;
while(i<steps) {
  if (digitalRead(HOLD_PIN) == LOW) {
    digitalWrite(EN_PIN, LOW);
    if (i < pulseDelayRange && curPulseDelay > minPulseDelay) {
      curPulseDelay -= pulseDelayStep;
    } else if (steps - i < pulseDelayRange && curPulseDelay < maxPulseDelay) {
      curPulseDelay += pulseDelayStep;
    }

    digitalWrite(STEP_R, HIGH);
    digitalWrite(STEP_L, HIGH);
    delayMicroseconds(curPulseDelay);
    digitalWrite(STEP_R, LOW);
    digitalWrite(STEP_L, LOW);
    delayMicroseconds(curPulseDelay);

    Serial.print('-');
    i++;
  } else {
    digitalWrite(EN_PIN, HIGH);
    delay(1000);
  }
}
Serial.print('.');
delay(500);
digitalWrite(EN_PIN, HIGH);
}

void moveLeft(int direction, int steps) {
  curPulseDelay = maxPulseDelay;

  if (direction == 0) {
    digitalWrite(DIR_L, LOW);
    digitalWrite(DIR_R, HIGH);
  } else if (direction == 1) {
    digitalWrite(DIR_L, HIGH);
    digitalWrite(DIR_L, LOW);
  }

  int i = 0;
  while(i<steps) {
    if (digitalRead(HOLD_PIN) == LOW) {
      digitalWrite(EN_PIN, LOW);
      if (i < pulseDelayRange && curPulseDelay > minPulseDelay) {
        curPulseDelay -= pulseDelayStep;

```

```
} else if (steps - i < pulseDelayRange && curPulseDelay < maxPulseDelay) {  
    curPulseDelay += pulseDelayStep;  
}  
  
digitalWrite(STEP_L, HIGH);  
digitalWrite(STEP_R, HIGH);  
delayMicroseconds(curPulseDelay);  
digitalWrite(STEP_L, LOW);  
digitalWrite(STEP_R, LOW);  
delayMicroseconds(curPulseDelay);  
  
Serial.print('-');  
i++;  
} else {  
    digitalWrite(EN_PIN, HIGH);  
    delay(1000);  
}  
}  
Serial.print('.');  
delay(500);  
digitalWrite(EN_PIN, HIGH);  
}
```

Παράρτημα III Κώδικας *Speech API*

```
def getSpeechFile(self, text, lang=None):
    conn = mariadb.connect(**self.config)
    cur = conn.cursor()
    speech_file = None

    try:
        sql = "SELECT speech_file FROM speech_files WHERE speech_text=?"
        cur.execute(sql, (text,))
        row = cur.fetchone()

        if row:
            print("Speech file found in cache!")
            speech_file = row[0]
        else:
            print("Speech file was not found in cache! Proceeding to
download.")
            speech_file = self.cacheSpeechFile(text, lang)
            print("Downloaded speech file is:", speech_file)

    except Exception:
        return jsonify({'status': 'ERROR', 'message': 'An unknown error
occurred!'})
    finally:
        cur.close()
        conn.close()

    pwd = '/home/pi/user_interface/frontend/'
    # pwd = ''
    speech_file_contents = None
    with open(pwd + 'robot_waiter_app/static/audio/speech/' + speech_file,
'rb') as sf:
        speech_file_contents = base64.b64encode(sf.read()).decode('ascii')

    return jsonify({'status': 'SUCCESS', 'speech_file_name': speech_file,
'speech_file': speech_file_contents})

def cacheSpeechFile(self, text, lang):
    conn = mariadb.connect(**self.config)
    cur = conn.cursor()

    voice = None
    if lang == "el":
        voice = "el-GR"
    elif lang == "en":
        voice = "en-US"
```

```
# Send request to soundoftext API
api_url = 'https://api.soundoftext.com/sounds/'
data = {
    "engine": "Google",
    "data": {
        "text": text,
        "voice": voice
    }
}
headers = {'Content-type': 'application/json', 'Accept': '*/*'}
print("Requesting speech file generation.")
resp = requests.post(api_url, data=json.dumps(data), headers=headers)
resp_dict = resp.json()
file_id = resp_dict["id"]
print(file_id)

# Loop until soundoftext API has generated mp3
location = None
generated = False
while not generated:
    pending_url = api_url + file_id
    r = requests.get(pending_url)
    r_dict = r.json()
    status = r_dict["status"]
    print("Speech file generation pending... Status: ", status)
    if status == "Done":
        location = r_dict["location"]
        generated = True
    else:
        sleep(.1)

print("Speech file generation complete! Downloading...")

# Download mp3 file and save it
filename = file_id + '.mp3'
file_path = './robot_waiter_app/static/audio/speech/' + filename
with requests.get(location, stream=True) as r:
    r.raise_for_status()
    with open(file_path, 'wb') as f:
        for chunk in r.iter_content(chunk_size=8192):
            f.write(chunk)

# Save path of mp3 file in database
print("Inserting speech file database entry.")
try:
    sql = "INSERT INTO speech_files (`speech_file_UUID`, `speech_text`, `speech_file`) VALUES (?, ?, ?)"
    cur.execute(sql, (file_id, text, filename))
    conn.commit()
except Exception:
    return None
finally:
    cur.close()
    conn.close()

print("Done!")
# Return mp3 file name
return filename
```