



**Ελληνικό Μεσογειακό Πανεπιστήμιο
(ΕΛ.ΜΕ.ΠΑ)**

Σχολή Μηχανικών

Τμήμα Μηχανολόγων Μηχανικών

Πτυχιακή εργασία

Τίτλος: Ηλεκτρικό Αναπηρικό Αμαξίδιο

Αποδουλιανάκης Παύλος (ΑΜ: ΤΜ6178)

Επιβλέπων εκπαιδευτικός : Φασουλάς Ιωάννης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω, εκ βάθους καρδίας τον καθηγητή μου, Δρ. Κύριο Φασουλά Ιωάννη, για την εμπιστοσύνη του να μου αναθέσει να φέρω εις πέρας, μία τόσο σημαντική πτυχιακή εργασία, καθώς και για την ενέπνευση που μου άσκησε, στο να ασχοληθώ με το κομμάτι της μηχαντρονικής, να το ενστερνιστώ και να υλοποιήσω την παρούσα πτυχιακή εργασία.

Επίσης θα ήθελα να ευχαριστήσω τον φίλο και συμφοιτητή Λοΐζου Αλέξανδρο για την ηθική υποστήριξη που μου πρόσφερε καθ' όλη τη διάρκεια της πτυχιακής εργασίας και τους επίσης φίλους και συμφοιτητές, Καμπουράκη Γεώργιο, Παχάκη Γεώργιο και Σηφάκη Χρήστο για τις πολύτιμες συμβουλές τους.

Κλείνοντας, δεν θα μπορούσα να παραλείψω το στενό μου οικογενειακό περιβάλλον, το οποίο μου έχει σταθεί καθ' όλη τη πορεία της ζωής μου και εξίσου σε αυτό το κομβικό κομμάτι για εμένα, την υλοποίηση της πτυχιακής μου εργασίας.

Abstract

The following undergraduate thesis, is about a fully electronic and digital conversion, of a standard wheelchair. After the conversion, the wheelchair must be able to move a person with maximum mass of 100 kg, on a 30% slope road, at 1.4m/s speed.

The basic prerequisites are, that the manipulation will be done in two different ways, which they are: with a joystick controller, and secondly by a voice command.

In conclusion, all systems to be installed will have to be modified so as not to alter any characteristic of the original condition of the wheelchair and that the user will be able to install all the parts of the conversion without the existence of any specialist technician.

Σύνοψη

Η παρακάτω πτυχιακή εργασία, αφορά μια πλήρως μηχανική μετατροπή ενός συμβατικού αναπηρικού αμαξιδίου, σε ηλεκτρικό. Μετά την μετατροπή, το αναπηρικό αμαξίδιο θα πρέπει να μπορεί να μετακινήσει ένα άτομο με μέγιστη μάζα εκατό κιλών, σε δρόμο με κλίση 30%, με ταχύτητα 1,4m/s.

Ο χειρισμός του αμαξιδίου θα γίνεται με δύο διαφορετικούς τρόπους: πρώτον, μέσω ενός χειριστηρίου τύπου «joystick», και δεύτερον μέσω φωνητικής εντολής.

Εν κατακλείδι, όλα τα συστήματα που πρόκειται να εγκατασταθούν δεν θα πρέπει να αλλάξουν κανένα χαρακτηριστικό της αρχικής κατάστασης του αναπηρικού αμαξιδίου, καθώς και ο ίδιος ο χρήστης να είναι σε θέση να εγκαταστήσει όλα τα απαραίτητα εξαρτήματα για την μετατροπή, χωρίς την ύπαρξη ειδικού τεχνικού.

Πίνακας Περιεχομένων

<i>Ευχαριστίες</i>	III
<i>Abstract</i>	IV
<i>Σύνοψη</i>	V
<i>Πίνακας Περιεχομένων</i>	VI
<i>Λίστα Εικόνων</i>	IX
<i>Λίστα Πινάκων</i>	XIV
Κεφάλαιο 1 <i>Εισαγωγή</i>	1
1.1 <i>Περίληψη</i>	2
1.2 <i>Κίνητρο για την Διεξαγωγή της Εργασίας – Στόχοι</i>	2
1.3 <i>Δομή Εργασίας</i>	3
Κεφάλαιο 2 <i>Η ιστορία του αναπηρικού αμαξιδίου</i>	5
2.1 <i>State of the Art</i>	9
Κεφάλαιο 3 <i>Κόριο μέρος πτυχιακής εργασίας</i>	13
3.1 <i>Ανάλυση Προβλήματος</i>	13
3.2 <i>Είδη αναπηρικών αμαξιδίων</i>	13
3.3 <i>Υπολογισμός κινητήρων</i>	15
3.4 <i>Χειρισμός</i>	17
3.5 <i>Σχεδιάγραμμα του ηλεκτρικού κυκλώματος</i>	18
3.6 <i>Σχεδιάγραμμα του λογικού διαγράμματος</i>	19
3.7 <i>Τα τμήματα του ΚΙΤ</i>	20
3.8 <i>Κονσόλα χειρισμού</i>	22

3.9	Θήκη συσσωρευτών	23
4.1	Βάσεις κινητήρων.....	24
Κεφάλαιο 4 Υλοποίηση		25
4.1	Χειριστήριο (JOYSTICK) (JH-D202X-R4).....	26
4.2	Φωνητική εντολή	30
4.3	Προγραμματισμός και λογισμικό φωνητικής εντολής.....	33
4.4	Συγγραφή κώδικα φωνητικής εντολής.....	39
4.5	Η οθόνη LCD 4884	41
4.6	Προγραμματισμός της οθόνης 4884 και το πρωτόκολλο επικοινωνίας “MASTER – SLAVE”. ..	44
4.7	Η Η-Γέφυρα (IRF3205)	46
4.8	Ο Μετατροπέας τάσης (4R7 DC-DC)	48
4.9	Έλεγχος περιστροφής.....	49
Κεφάλαιο 5 Κατασκευές		50
5.1	Κατασκευή θήκης συσσωρευτών	50
6.3	Κατασκευή βάσεων κινητήρων.....	61
5.3	Προσαρμογή της τροχαλίας στον κινητήρα	70
5.4	Κατασκευή βάσεως συσσωρευτών.....	73
5.5	Κατασκευή κονσόλας χειρισμού.....	82
Κεφάλαιο 6 Αποτελέσματα		93
6.1	Συμπεράσματα.....	94
6.2	Μελλοντική Εργασία και Επεκτάσεις	95
6.3	Βιβλιογραφία	96
Κεφάλαιο 7 Παραρτήματα της Πτυχιακής Εργασίας.....		97
	Ο Κώδικας για το ARDUINO MASTER.....	97
	Ο Κώδικας για το ARDUINO SLAVE.....	106

Λίστα Εικόνων

Εικόνα 1: Το αναπηρικό αμαξίδιο που χρησιμοποιήθηκε.	1
Εικόνα 2: Αριστερά η αναπηρική καρέκλα του 18 ^{ου} αιώνα και δεξιά η καρέκλα μπάθ του Χιθ.	6
Εικόνα 3: Το αναπηρικό αμαξίδιο “YAMAHA JWX-2” με την τεχνολογία PAPA.	8
Εικόνα 4: Στα αριστερά εικονίζεται μία μονοκινητήρια αναπηρική καρέκλα με σύστημα διεύθυνσης και στα δεξιά μία ηλεκτρική αναπηρική καρέκλα δύο κινητήρων με σύστημα μετάδοσης αποτελούμενο από γρανάζια και αλυσίδες.	9
Εικόνα 5: Το ολοκληρωμένο σύστημα απευθείας μετάδοσης της κίνησης της «καρέκλας κλαίν».	10
Εικόνα 6: Αριστερά βλέπουμε τον Τζωρτζ Κλάιν να χειρίζεται την «Καρέκλα Κλάιν» και δεξιά να επιθεωρεί τις μπαταρίες με ένα συνεργάτη του πριν τις τοποθετήσει στο αμαξίδιο.	11
Εικόνα 7: Πέντε διαφορετικοί τύποι αμαξιδίων.	14
Εικόνα 8: Σχεδιάγραμμα ηλεκτρικού κυκλώματος.	18
Εικόνα 9: Σχεδιάγραμμα λογικού διαγράμματος.	19
Εικόνα 10: Τα τμήματα που αποτελούν το ΚΙΤ.	20
Εικόνα 11: Το αμαξίδιο πριν και μετά την εγκατάσταση του ΚΙΤ.	21
Εικόνα 12: Όψεις του αναπηρικού αμαξιδίου με το ΚΙΤ.	21
Εικόνα 13: Κουσόλα χειρισμού.	22
Εικόνα 14: Η θήκη συσσωρευτών.	23
Εικόνα 15: Βάσεις κινητήρων.	24
Εικόνα 16: Αμαξίδιο κατασκευασμένο από εξαρτήματα LEGO.	25
Εικόνα 17: Χειριστήριο δύο αξόνων και οκτώ διαφορετικών κατευθύνσεων (JH-D202X-R4).	27
Εικόνα 18: Οι άξονες X και Y καθώς και οι τιμές των ποτενσιόμετρων στις ακραίες θέσεις.	28
Εικόνα 19: Σύνδεση του χειριστηρίου στην πλατφόρμα «Arduino UNO».	29
Εικόνα 20: Το ΚΙΤ συναρμολόγησης EasyVR3.0.	32
Εικόνα 21: Οπίσθια όψη του πλήρως συναρμολογημένου και συγκολλημένου «EasyVR 3.0».	32
Εικόνα 22: Εμπρόσθια όψη του πλήρως συναρμολογημένου και συγκολλημένου «EasyVR 3.0».	33

Εικόνα 23: Το «Easy VR 3.0» επάνω στην πλατφόρμα «Arduino UNO».....	33
Εικόνα 24: Το πρόγραμμα που γράφτηκε μέσω του λογισμικού της «Arduino».	34
Εικόνα 25: Σύνδεση λογισμικού «EasyVR Commander» με την πλακέτα «EasyVR 3.0».	35
Εικόνα 26: Η επιλογή “Add Command” μέσα στο λογισμικό.	36
Εικόνα 27: Οι πέντε προκαθορισμένες εντολές με λατινικούς χαρακτήρες.....	37
Εικόνα 28: Η επιλογή “Train Command (ENTER)”.	37
Εικόνα 29: Εκκίνηση καταγραφής της εντολής.....	38
Εικόνα 30: Πέρασ της καταγραφής των ηχογραφήσεων για τις πέντε εντολές.	39
Εικόνα 31: Η επιλογή «Διαχείριση βιβλιοθηκών».	40
Εικόνα 32: Ολοκλήρωση εγκατάστασης της βιβλιοθήκης “EasyVR”.	40
Εικόνα 33: Η οθόνη LCD 4884.	41
Εικόνα 34: Διάγραμμα ροής για το μενού της οθόνης.....	42
Εικόνα 35: Το μενού, με τις τρεις διαφορετικές επιλογές διασύνδεσης.	43
Εικόνα 36: Η Σύνδεση των δυο μικροελεγκτών με τη μέθοδο «master - slave».	45
Εικόνα 37: Η Η-Γέφυρα (IRF3205) δύο καναλιών.	46
Εικόνα 38: Η Η-Γέφυρα BTS7960B.....	47
Εικόνα 39: Ο μετατροπέας τάσης (4R7 DC-DC) που χρησιμοποιήθηκε.....	48
Εικόνα 40: Το ποτενσιόμετρο για τον έλεγχο περιστροφής.....	49
Εικόνα 41: Συσσωρευτές των δώδεκα βολτ (12V).	50
Εικόνα 42: Χαρτόνι για την κατασκευή του καλουπιού.	51
Εικόνα 43: Προσαρμογή του χαρτονιού στις επιθυμητές διαστάσεις.....	52
Εικόνα 44: Σκληρό αφρολέξ επενδυμένο με ταινία περιτυλίγματος.	53
Εικόνα 45: Κατασκευή του δεύτερου καλουπιού.	53
Εικόνα 46: Αριστερά δοχείο με ποσότητα πολυεστέρα και δεξιά καταλότης - σκληρυντής για πολυεστέρα.....	54
Εικόνα 47: Άπλωμα πολυεστέρα στο πρώτο στρώσιμο ινών.....	55
Εικόνα 48: Στιγμή από τη κατεργασία στην τελική γεωμετρία.	56

<i>Εικόνα 49: Αφαίρεση του καλουπιού από την θήκη.....</i>	<i>57</i>
<i>Εικόνα 50: Σημάδεμα για το κόψιμο της θήκης στο επιθυμητό ύψος.....</i>	<i>57</i>
<i>Εικόνα 51: Η τελική μορφή της πολυεστερικής θήκης.....</i>	<i>58</i>
<i>Εικόνα 52: Κατασκευή καλουπιού για το κάλυμμα της θήκης.....</i>	<i>59</i>
<i>Εικόνα 53: Στιγμή από την κατεργασία για το κάλυμμα.....</i>	<i>60</i>
<i>Εικόνα 54: Το κάλυμμα έπειτα από το πέρας της κατεργασίας και πριν τη βαφή του.....</i>	<i>60</i>
<i>Εικόνα 55: Η τελική μορφή της θήκης των συσσωρευτών με το κάλυμμα της έπειτα από βαφή με μαύρο ματ σπρέι.....</i>	<i>61</i>
<i>Εικόνα 56: Ράγες στήριξης.....</i>	<i>62</i>
<i>Εικόνα 57: Εσοχή για την προσαρμογή στη δοκό του αμαξιδίου.....</i>	<i>63</i>
<i>Εικόνα 58: Η ράγα στήριξης τοποθετημένη στο αμαξίδιο.....</i>	<i>63</i>
<i>Εικόνα 59: Το κενό που δημιουργούν οι δύο ράγες λόγω της διαφοράς ύψους.....</i>	<i>64</i>
<i>Εικόνα 60: Δέσιμο της κινούμενης ράγας στην μέγγενη και κοπή των «αυτιών» με γωνιακό τροχό.....</i>	<i>65</i>
<i>Εικόνα 61: Η ράγα στήριξης μέσα στην κινούμενη ράγα πριν τη δημιουργία νέων «αυτιών».....</i>	<i>66</i>
<i>Εικόνα 62: Σημάδεμα και κατασκευή νέων «αυτιών».....</i>	<i>66</i>
<i>Εικόνα 63: Το «συρτάρωμα» των δύο ραγών.....</i>	<i>67</i>
<i>Εικόνα 64: Τα εξαρτήματα που θα στερεώσουν τον κινητήρα στη ράγα στήριξης.....</i>	<i>68</i>
<i>Εικόνα 65: Τα εξαρτήματα για την στερέωση του κινητήρα έπειτα από τη βαφή.....</i>	<i>68</i>
<i>Εικόνα 66: Το στερέωμα του κινητήρα επάνω στη ράγα κινήσεως.....</i>	<i>69</i>
<i>Εικόνα 67: Τοποθέτηση του κινητήρα επάνω στο αναπηρικό αμαξίδιο.....</i>	<i>69</i>
<i>Εικόνα 68: Κοχλίας και περικόχλιο σε σχήμα «πεταλούδας για τη σταθεροποίηση του κινητήρα».....</i>	<i>70</i>
<i>Εικόνα 69: Ηλεκτρικός κινητήρας.....</i>	<i>70</i>
<i>Εικόνα 70: Αριστερά διακρίνεται ο μετατροπέας από αριστερό σε δεξιό σπείρωμα και στα δεξιά μαζί με το περικόχλιο που θα ασφαλίσει την τροχαλία στην θέση της.....</i>	<i>71</i>
<i>Εικόνα 71: Τοποθέτηση της τροχαλίας επάνω στον άξονα του κινητήρα.....</i>	<i>72</i>
<i>Εικόνα 72: Διάτρητη λαμαρίνα τριών χιλιοστών.....</i>	<i>73</i>
<i>Εικόνα 73: Βιομηχανικό λαμαρινοφάλιδο πεπιεσμένου αέρα.....</i>	<i>74</i>

<i>Εικόνα 74: Διαδικασία για την κοπή της διάτρητης λαμαρίνας</i>	75
<i>Εικόνα 75: Η θέση στην οποία θα τοποθετηθεί η βάση των συσσωρευτών</i>	76
<i>Εικόνα 76: Στιγμή από τη διαδικασία του «γωνιάσματος» της βάσης</i>	77
<i>Εικόνα 77: Στιγμή από το «στραντζάρισμα» της διάτρητης λαμαρίνας για την κατασκευή ορθής γωνίας</i>	78
<i>Εικόνα 78: Προσαρμογή ακριβώς στις διαστάσεις των κυκλικών δοκών του αμαξιδίου</i>	79
<i>Εικόνα 79: Κοπή διάτρητης λαμαρίνας για την κατασκευή «νεύρου» σε λαμαρινοψάλιδο</i>	80
<i>Εικόνα 80: Η βάση μαζί με τα δύο «νεύρα» μετά από βαφή και αφαίρεσης σκουριάς με γυαλόχαρτο</i>	80
<i>Εικόνα 81: Η βάση μετά τη συναρμολόγηση</i>	81
<i>Εικόνα 82: Το τελικό αποτέλεσμα της βάσης των συσσωρευτών τοποθετημένη στο αμαξίδιο</i>	82
<i>Εικόνα 83: Φόλλο λαμαρίνας, πάχους δύο χιλιοστών</i>	83
<i>Εικόνα 84: Επιφάνεια για την κατασκευή της κονσόλας</i>	83
<i>Εικόνα 85: Η κονσόλα μετά το στραντζάρισμα και το άνοιγμα οπών για τα πριτσίνια και το «πιάσιμο» της επάνω στο μπράτσο</i>	84
<i>Εικόνα 86: Το χειριστήριο που θα προσαρμοστεί επάνω στην κονσόλα</i>	85
<i>Εικόνα 87: Διάνοξη οπών για την τοποθέτηση του χειριστηρίου</i>	85
<i>Εικόνα 88: Ο γενικός διακόπτης του οχήματος</i>	86
<i>Εικόνα 89: Τοποθέτηση της οθόνης επιλογών και του χειριστηρίου επάνω στην κονσόλα</i>	87
<i>Εικόνα 90: Προετοιμασία της κονσόλας για βαφή με μαύρο ματ σπρέι</i>	88
<i>Εικόνα 91: Βαφή της βάσης της οθόνης επιλογών με μαύρο ματ σπρέι και τοποθέτηση αφρώδους μονωτικού υλικού</i>	89
<i>Εικόνα 92: Η κονσόλα με όλα τα εξαρτήματα</i>	90
<i>Εικόνα 93: Προσαρμογή του μικροφώνου επάνω στη βάση της οθόνης</i>	91
<i>Εικόνα 94: Διάτρητη λαμαρίνα έπειτα από διαδικασία αφαίρεσης σκουριάς</i>	91
<i>Εικόνα 95: Αριστερά εικονίζεται το ένα καλάκι πριν τη βαφή και δεξιά έπειτα από τη βαφή</i>	92
<i>Εικόνα 96: Η κονσόλα έπειτα την τοποθέτηση των καλάκιών</i>	92
<i>Εικόνα 97: Το ΚΙΤ συναρμολόγησης</i>	93

Εικόνα 98: Το αναληρικό αμαξίδιο πριν και μετά την εγκατάσταση του «ΚΙΤ». 94

Λίστα Πινάκων

Πίνακας 1: Συγκριτικός πίνακας μεταξύ του state of the art και του ηλεκτρικού αναπηρικού αμαξιδίου. 12

Πίνακας 2: Απαιτούμενη ισχύς για την κίνηση αναπηρικών αμαξιδίων, τριών διαφορετικών τύπων. 16

Κεφάλαιο 1 Εισαγωγή

Το περιεχόμενο της παρούσας πτυχιακής εργασίας είναι, η μετατροπή ενός συμβατικού χειροκίνητου αναπηρικού αμαξιδίου, σε ένα ηλεκτρικά καθοδηγούμενο, με δύο διαφορετικούς τρόπους ελέγχου, με εύκολη εγκατάσταση των εξαρτημάτων, από οποιονδήποτε, χωρίς τη παρέμβαση ενός εξειδικευμένου τεχνικού.

Στόχος είναι η βελτίωση της ποιότητας ζωής των ατόμων με ειδικές ανάγκες, μέσω ενός πολύ χαμηλού κόστους τροποποίησης, του υπάρχοντος αμαξιδίου που διαθέτουν, σε ένα αρκετά πιο ευέλικτο.



Εικόνα 1: Το αναπηρικό αμαξίδιο που χρησιμοποιήθηκε.

1.1 Περίληψη

Η παρούσα πτυχιακή εργασία έχει στόχο, να δημιουργήσει ένα απόλυτα ολοκληρωμένο μηχανητρονικό σύστημα, το οποίο θα είναι ευέλικτο να προσαρμοστεί σε συμβατικά αναπηρικά αμαξίδια, χωρίς να απαιτείται η οποιαδήποτε παρέμβαση στο αμαξίδιο για την εγκατάσταση του συστήματος και θα αποσπάται με ευκολία όποτε το αποφασίσει ο χρήστης να το επαναφέρει στην αρχική του μορφή.

Το μηχανητρονικό σύστημα θα βασίζεται στον μικροελεγκτή ARDUINO UNO, ο οποίος θα μπορεί να δέχεται εντολές από δύο διαφορετικές διεπαφές (interfaces), ανάλογα ποιά επιθυμεί ο χρήστης και θα ελέγχει δύο ηλεκτροκινητήρες, που θα προσαρμοστούν στους οπίσθιους τροχούς του αμαξιδίου, περιστρέφοντας τους κατάλληλα για τις απαραίτητες κινήσεις στο χώρο.

Οι δύο διαφορετικές διεπαφές θα είναι οι εξής: Μέσω χειριστηρίου τύπου joystick, και μέσω φωνητικής εντολής, όπου το όχημα θα κατευθύνεται μονάχα με την φωνή του χρήστη, σε τέσσερις διαφορετικές κατευθύνσεις (μπροστά - πίσω - αριστερά - δεξιά) καθώς και θα σταματάει με την εντολή «ΣΤΟΠ».

Η τροφοδοσία του συστήματος θα εξασφαλίζεται από δύο συσσωρευτές των 12 βολτ, που θα τοποθετηθούν στο όχημα, οι οποίοι θα συνδεθούν εν σειρά δημιουργώντας μια διαφορά δυναμικού των 24 βολτ, καθώς επίσης θα τοποθετηθεί και ένας γενικός διακόπτης που θα διακόπτει την παροχή ρεύματος.

1.2 Κίνητρο για την Διεξαγωγή της Εργασίας - Στόχοι

Τα άτομα με ειδικές ανάγκες αντιμετωπίζουν σοβαρό πρόβλημα στις καθημερινές τους μετακινήσεις. Το κόστος για ένα ηλεκτρικό αναπηρικό αμαξίδιο, που θα ομαλοποιούσε τις καθημερινές τους ανάγκες, είναι αρκετά υψηλό και πολλαπλάσιο σε σχέση με ένα συμβατικό αμαξίδιο, γεγονός που το καθιστά σε αρκετές περιπτώσεις απαγορευτικό για τα άτομα που το έχουν ανάγκη.

Με δεδομένη αυτή την κατάσταση, η παρούσα πτυχιακή εργασία στοχεύει στη μελέτη και κατασκευή ενός ευέλικτου μηχανολογικού συστήματος με αρκετά χαμηλό κόστος, που θα προσαρμόζεται στα απλά και οικονομικά χειροκίνητα αμαξίδια της αγοράς, δίνοντας το πλεονέκτημα στα άτομα που έχουν στην κατοχή τους ένα συμβατικό αμαξίδιο, να το μετατρέψουν σε ηλεκτρικό, εύκολα και γρήγορα χωρίς την ανάγκη κάποιου εξειδικευμένου τεχνικού, αποτρέποντας τους να δαπανήσουν ένα υπέρογκο ποσό.

Στόχος της προσπάθειας αυτής, είναι η κάλυψη των περισσότερων περιπτώσεων ατόμων με ειδικές ανάγκες και η πλήρης αξιοποίηση των χρημάτων που θα δαπανήσει ο αγοραστής, προσφέροντας καινοτόμους τρόπους χειρισμού του αμαξιδίου όπως η φωνητική εντολή.

1.3 Δομή Εργασίας

Ακολουθούν έξι κεφάλαια τα οποία θα αναφερθούν αναλυτικά στη διαδικασία έρευνας, σχεδιασμού και υλοποίησης και αποτελεσμάτων του ηλεκτρικού αναπηρικού αμαξιδίου.

Πιο συγκεκριμένα,

Κεφάλαιο 2: Ιστορική αναδρομή στην εξέλιξη του αναπηρικού αμαξιδίου ανά τους αιώνες και εμπειριστατωμένη αναφορά στο state of the art της απαρχής των ηλεκτροκίνητων αναπηρικών αμαξιδίων.

Κεφάλαιο 3: Αναφέρεται στο κύριο μέρος της εργασίας. Οριοθετείται το πρόβλημα που θα επιλυθεί, μέσα από έρευνα αγοράς και κατόπιν συγκεκριμενοποιείται με ρεαλιστικούς παράγοντες λαμβάνοντας υπ' όψιν κάθε απαραίτητη παράμετρο. Τέλος περιγράφεται ο τρόπος με τον οποίο θα γίνεται και ο χειρισμός του αμαξιδίου που θα κατασκευαστεί.

Κεφάλαιο 4: Στο τέταρτο κεφάλαιο, γίνεται μια εκτενής αναφορά στο ηλεκτρονικό κομμάτι της εργασίας, καθώς περιγράφονται αναλυτικά οι συνδεσμολογίες και ο τρόπος λειτουργίας των βασικών συστημάτων χειρισμού (χειριστήριο, φωνητική εντολή), καθώς και οι συγγραφή του κώδικα που θα χρησιμοποιήσουν οι μικροελεγκτές. Τέλος, μέσω της παρουσίασης του τρόπου λειτουργίας της μεθόδου “Master-slave”, θα γίνει κατανοητός ο τρόπος με τον οποίο φτάνει η πληροφορία από τον χρήστη, στους κινητήρες και ο τρόπος διασύνδεσης της οθόνης LCD.

Κεφάλαιο 5: Στο κεφάλαιο αυτό, παρουσιάζονται αναλυτικά όλες οι κατασκευές που απαιτήθηκαν για την υλοποίηση του μηχανολογικού τομέα. Οι κατασκευές αυτές είναι:

- Θήκη συσσωρευτών
- Βάση συσσωρευτών
- Βάσεις κινητήρα με μεταβλητό χειροκίνητο σύστημα σύμπλεξης τροχών.
- Κονσόλα χειρισμού

Κεφάλαιο 6: Στο έκτο κεφάλαιο αναφέρονται τα συμπεράσματα της εργασίας καθώς και μελλοντικές διορθώσεις και επεκτάσεις. Το κεφάλαιο αυτό κλείνει με την παρουσίαση της βιβλιογραφίας που χρησιμοποιήθηκε.

Κεφάλαιο 7: Στο τελευταίο κεφάλαιο αναγράφονται οι κώδικες που χρησιμοποιήθηκαν για τους δύο μικροελεγκτές καθώς και σχεδιάγραμμα του ηλεκτρικού κυκλώματος.

Κεφάλαιο 2 Η ιστορία του αναπηρικού αμαξιδίου

Η ακριβής χρονολογία στην οποία εφευρέθηκαν οι πρώτες τροχοφόρες καρέκλες, οι οποίες χρησιμοποιήθηκαν από άτομα με ειδικές ανάγκες παραμένει ακόμη και σήμερα άγνωστη. Ωστόσο, σύμφωνα με μελετητές, είναι αρκετά πιθανό η ιστορία της αναπηρικής καρέκλας να ξεκινάει κάποια στιγμή, μεταξύ του 6ου και του 7ου αιώνα π.Χ. , ενδεχομένως με την ανάπτυξη τροχοφόρων επίπλων και καροτσιών.

Η πιθανή εμφάνιση της αναπηρικής καρέκλας στην Ευρώπη, χρονολογείται τον 12^ο αιώνα. Παρόλα αυτά, η πρώτη καταγεγραμμένη χρήση αναπηρικού αμαξιδίου, στην Ευρώπη, από άτομα με ειδικές ανάγκες συναντάται τον 17^ο αιώνα. Στις αρχές του αιώνα εκείνου ο Γερμανός μηχανικός και εφευρέτης Γιόχαν Χάουτς (Johann Hautsch) κατασκεύαζε τροχοφόρες αναπηρικές καρέκλες στην Νυρεμβέργη και το 1655, ο επίσης Γερμανός Στέφαν Φάρφλερ (Stephan Farfler), ο οποίος έπασχε από κινητικά προβλήματα, κατασκεύασε μία τρίτροχη καρέκλα, την οποία κατόρθωνε να κινήσει με τη χρήση μιας περιστροφικής λαβής, προσαρμοσμένης στον εμπρόσθιο τροχό της καρέκλας. Έτσι με την πάροδο του χρόνου, η χρήση της αναπηρικής καρέκλας διαδόθηκε και προς τα τέλη του αιώνα βελτιώθηκε αισθητά, βοηθώντας στην εξάπλωση της σε ευρεία χρήση. (Πηγή: <https://www.britannica.com/topic/history-of-the-wheelchair-1971423>)

Σχεδιάστηκαν ως μέσο μεταφοράς κυρίως για τους πλούσιους. Τον 18ο αιώνα, οι αναπηρικές καρέκλες άρχισαν να εμφανίζονται σε καταλόγους χειρουργικών και ιατρικών καταλόγων, όπου διαφημίζονται ως οχήματα μεταφοράς για ασθενείς. Έμοιαζαν αρκετά με τις πολυθρόνες, ήταν ξύλινες ή ψάθινες με σιδερένια μηχανικά μέρη, με μεγάλους τροχούς στο εμπρόσθιο μέρος και ένα μικρότερο τροχό στο πίσω μέρος που εξασφάλιζε την ισορροπία. Ήταν περίτεχνα, βαριά και αποδείχθηκαν ως δυσκίνητα.

Το 1750 ο Άγγλος εφευρέτης Τζέιμς Χιθ (James Heath) εισήγαγε την καρέκλα μπάθ, προορισμένη για χρήση από κυρίες και αναπήρους. Η καρέκλα μπάθ ήταν ένα δημοφιλές μέσο μεταφοράς, ιδιαίτερα στη Βρετανία, όπου χρησίμευε ως μέσο μεταφοράς για τραυματίες, άρρωστους ή άτομα με αναπηρία και ως τρόπος μεταφοράς για τους πλούσιους. (Πηγή: <https://blog.sciencemuseum.org.uk/history-of-the-wheelchair/>)



Εικόνα 2:Αριστερά η αναπηρική καρέκλα του 18^{ου} αιώνα και δεξιά η καρέκλα μπάθ του Χιθ.

Προς τα μέσα του 19ου αιώνα εισήχθησαν αναπηρικές καρέκλες με ξύλινα πλαίσια και καθίσματα και πλάτες από ζαχαροκάλαμο. Χρησιμοποιήθηκαν ευρέως στις Ηνωμένες Πολιτείες από βετεράνους του εμφυλίου πολέμου. Στα τέλη του 19ου αιώνα, εισήχθησαν και άλλες τροποποιήσεις, όπως οι τροχοί με συρματόσχοινα και τα λάστιχα από καουτσούκ. Ακόμη όμως και με αυτές τις εξελίξεις, η ανεξάρτητη κινητικότητα με τις περισσότερες αναπηρικές καρέκλες παρέμεινε περιορισμένη στα όρια των εσωτερικών χώρων.

Μία από τις σημαντικότερες εξελίξεις στην τεχνολογία των αναπηρικών αμαξιδίων τον 20^ο αιώνα ήταν η εφεύρεση της αναδιπλούμενης αναπηρικής καρέκλας, αρχικά κατασκευασμένης με σωληνωτό χάλυβα, που επέτρεψε άτομα με

ειδικές ανάγκες να χρησιμοποιούν τα αναπηρικά αμαξίδια τους έξω από τα σπίτια τους ή τα Ιατρεία. Τα πρώτα πτυσσόμενα σχέδια και οι σωληνωτές καρέκλες χάλυβα αναπτύχθηκαν κατά την πρώτη δεκαετία του αιώνα. Αργότερα, το 1932, ο Αμερικανός μηχανικός Έρμπερτ Έβερεστ (Herbert A. Everest) και ο Αμερικανός μηχανολόγος μηχανικός Χάρυ Τζένινγκς (Harry C. Jennings) εισήγαγαν την αναπηρική καρέκλα με εγκάρσιο σκελετό, η οποία έγινε ο τυπικός σχεδιασμός για πτυσσόμενες καρέκλες σωληνώσεων. Οι δύο άνδρες αργότερα δημιούργησαν την «Everest & Jennings, Inc.», η οποία έγινε κορυφαίος κατασκευαστής αναπηρικών αμαξιδίων.

Οι μεταγενέστερες εξελίξεις στον σχεδιασμό της αναπηρικής πολυθρόνας επικεντρώθηκαν κυρίως στη μείωση του βάρους και στην αύξηση της αξιοπιστίας και της απόδοσης. Πολλές προόδους προήλθαν από τη χρήση αναπηρικών αμαξιδίων στον αθλητισμό, η οποία ενέπνευσε την ανάπτυξη μοντέλων υπέρ-ελαφρών (ultra-lightweight). Έτσι έκανε την εμφάνιση της η καρέκλα Κουίκι (Quickie), μια υπέρ-ελαφριά αναπηρική πολυθρόνα άκαμπτου πλαισίου που εισήχθη το 1979 από τους Μέριλιν Χάμιλτον (Marilyn Hamilton), Τζιμ Οκαμότο (Jim Okamoto) και Ντόν Χέλμαν (Don Helman). Η αναπηρική πολυθρόνα Quickie ήταν μοναδική τόσο για τη βελτιωμένη της απόδοση όσο και για το ξεχωριστό χρώμα και της αισθητικής.

Μετά τον Δεύτερο Παγκόσμιο Πόλεμο, η ζήτηση αυξήθηκε για τις ηλεκτροκίνητες αναπηρικές καρέκλες. Τα πρώτα ηλεκτρικά αναπηρικά αμαξίδια αποτελούσαν βασικά πρότυπα αναπηρικά καροτσάκια με προσαρμοσμένους κινητήρες μόνο. Αργότερα, εισήχθησαν τα ηλεκτρικά αναπηρικά αμαξίδια στα οποία τοποθετήθηκαν ο κινητήρας και οι μπαταρίες κάτω από το κάθισμα. Επιπλέον βελτιώσεις σε ηλεκτρικά αναπηρικά αμαξίδια, περιελάμβαναν βελτιώσεις σε αναλογικούς ελεγκτές, μικροεπεξεργαστές και άλλες τεχνολογίες υπολογιστών.

Τον 21ο αιώνα η ενσωμάτωση των αναδυόμενων τεχνολογιών έγινε όλο και πιο σημαντικό στοιχείο του σχεδιασμού της αναπηρικής πολυθρόνας. Μεταξύ των νέων σχεδίων που εισήχθησαν στο πρώτο μέρος του αιώνα ήταν η αναπηρική καρέκλα με ένα κινητήρα που ενεργοποιούταν με μυϊκή ώθηση (Pushrim-Activated, Power Assisted Wheelchair - PAPA W). Το PAPA W ήταν ένα υβρίδιο ηλεκτρικών και χειροκίνητων τεχνολογιών για αναπηρικά αμαξίδια που χρησιμοποιούσε κινητήρες για να συμπληρώνουν την ισχύ που εφάρμοζε ο χρήστης κατά τη διάρκεια της επιτάχυνσης ή της πέδησης. (Πηγή: <https://www.britannica.com/topic/history-of-the-wheelchair-1971423> , <https://abilitytools.org/blog/history-of-the-wheelchair/>).



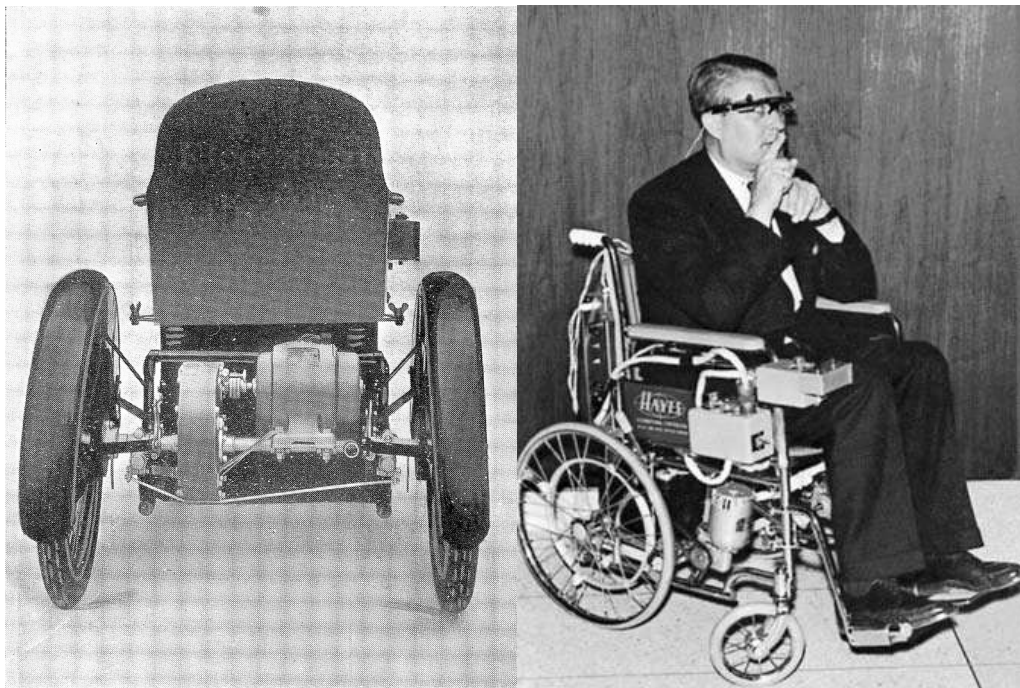
Εικόνα 3: Το αναπηρικό αμαξίδιο "YAMAHA JWX-2" με την τεχνολογία PAPA W.

2.1 State of the Art

Τον 20^ο αιώνα, επιχείρησαν αρκετοί εφευρέτες να προσαρμόσουν, σε υπάρχοντα αναπηρικά αμαξίδια, κινητήρες και συστήματα μετάδοσης που θα ενεργοποιούνταν από ένα χειριστήριο που θα κινούσε ο εκάστοτε χρήστης.

Ορισμένοι χρησιμοποίησαν ένα μόνο κινητήρα που έδινε κίνηση στους δυο οπίσθιους τροχούς μέσω ενός άξονα και προσάρμοσαν ένα σύστημα διεύθυνσης.

Πολλοί εφευρέτες επίσης, χρησιμοποίησαν δύο κινητήρες, έναν σε κάθε από τους δύο κινητήριους τροχούς, οι οποίοι με την διαφορετική ταχύτητα περιστροφής που μπορούσαν να προσφέρουν, κατάφερναν να δώσουν την επιθυμητή διεύθυνση στο χώρο, χωρίς την ύπαρξη κάποιου πολύπλοκου συστήματος διεύθυνσης.



Εικόνα 4: Στα αριστερά εικονίζεται μία μονοκινητήρια αναπηρική καρέκλα με σύστημα διεύθυνσης και στα δεξιά μία ηλεκτρική αναπηρική καρέκλα δύο κινητήρων με σύστημα μετάδοσης αποτελούμενο από γρανάζια και αλυσίδες.

Το 1950 ο καναδός εφευρέτης Τζώρτζ Κλάιν (George Klein), κατοχύρωσε δίπλωμα ευρεσιτεχνίας για ένα μηχανικό εξάρτημα το οποίο επιτρέπει την κίνηση σε συνηθισμένα αναπηρικά καρότσια της εποχής.

Το 1953 παρουσίασε ένα πρωτοποριακό αμαξίδιο με όνομα «Η καρέκλα Κλάιν» (The Klein Chair). Σε ένα συμβατικό αναπηρικό αμαξίδιο, προσάρμοσε δύο από τα συστήματα που είχε κατοχυρώσει, τρία χρόνια νωρίτερα, σε κάθε κινητήριο τροχό ξεχωριστά, δίνοντας τους κίνηση.

Ο τρόπος μετάδοσης τη κίνησης, γινόταν απευθείας από τον κινητήρα στον τροχό, μέσω μιας τροχαλίας, προσαρμοσμένης επάνω στον άξονα του κινητήρα, που ερχόταν σε επαφή με τον τροχό του αμαξιδίου μεταφέροντας του την κίνηση.

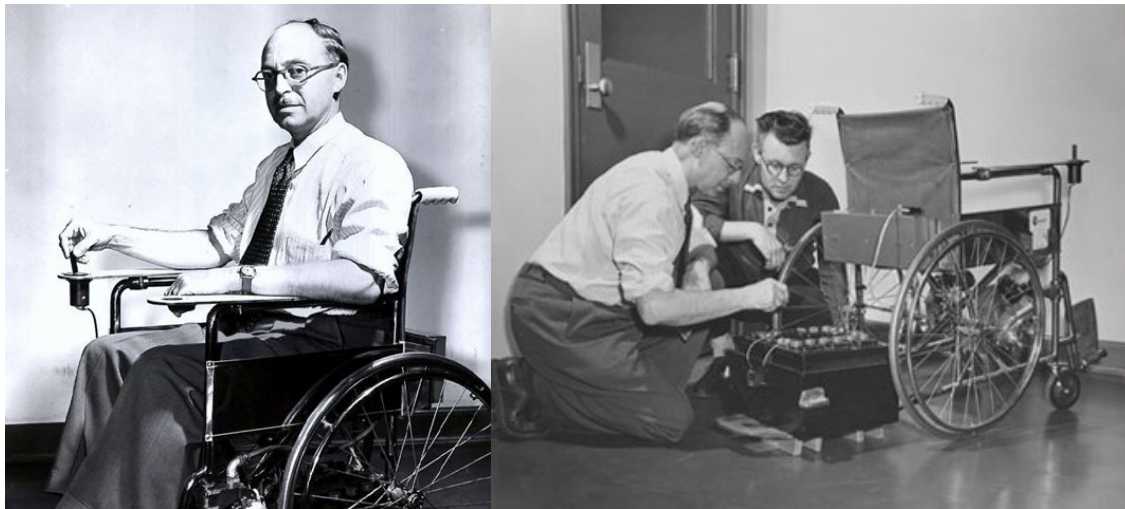
Με αυτό τον τρόπο πετύχαινε την μετάδοση της κίνησης, μέσω των δύο κινητήρων, χωρίς την παρουσία κάποιου περίπλοκου συστήματος μετάδοσης.



Εικόνα 5: Το ολοκληρωμένο σύστημα απευθείας μετάδοσης της κίνησης της «καρέκλας κλαιν».

Ο έλεγχος των δύο συστημάτων γινόταν μέσω ενός χειριστηρίου που βρισκόταν στο δεξιό μπράτσο της καρέκλας και η ισχύς που απαιτούνταν για την κίνηση του οχήματος, προερχόταν από δύο μπαταρίες, που προσάρμοσε στο κάτω μέρος του αμαξιδίου.

Το όλο εγχείρημα ήταν εξαιρετικά καινοτόμο και οι τεχνολογίες που χρησιμοποιήθηκαν πάνω σε αυτό εξακολουθούν να εφαρμόζονται μέχρι σήμερα στα ηλεκτρικά αμαξίδια. (Πηγή: <https://news.engineering.utoronto.ca/maker-george-klein-first-electric-wheelchair/> , [https://en.wikipedia.org/wiki/George_Klein_\(inventor\)](https://en.wikipedia.org/wiki/George_Klein_(inventor)) , <https://www.karmanhealthcare.com/blog/2018/10/22/what-was-the-first-electric-wheelchair/> , <https://www.streetdirectory.com/etoday/george-klein-and-the-electric-wheelchair-upcjoj.html>)



Εικόνα 6: Αριστερά βλέπουμε τον Τζωρτζ Κλάιν να χειρίζεται την «Καρέκλα Κλάιν» και δεξιά να επιθεωρεί τις μπαταρίες με ένα συνεργάτη του πριν τις τοποθετήσει στο αμαξίδιο.

Ακολουθεί ένας πίνακας που συγκρίνει τις ιδιαιτερότητες της «καρέκλας κλαίν» με το ηλεκτρικό αναπηρικό αμαξίδιο της παρούσας πτυχιακής εργασίας.

ΙΔΙΑΙΤΕΡΟΤΗΤΕΣ	«Καρέκλα Κλάιν»	Μηχατρονικό αναπηρικό αμαξίδιο
Εγκατάσταση χωρίς μετατροπή του αμαξιδίου	ΝΑΙ	ΝΑΙ
Απαιτείται ειδικός τεχνικός για την εγκατάσταση	ΝΑΙ	ΟΧΙ
Καθοδήγηση κινητήρων μέσω μικροελεγκτή	ΟΧΙ	ΝΑΙ
Μεγάλη μεταβολή βάρους μετά την εγκατάσταση	ΝΑΙ	ΟΧΙ
Δυνατότητα επέκτασης ψηφιακών συστημάτων π.χ. αισθητήρες	ΟΧΙ	ΝΑΙ
Πολλαπλοί τρόποι χειρισμού	ΟΧΙ	ΝΑΙ
Καθοδήγηση μέσω χειριστηρίου	ΝΑΙ	ΝΑΙ
Χειρισμός μέσω φωνητικής εντολής	ΟΧΙ	ΝΑΙ

Πίνακας 1: Συγκριτικός πίνακας μεταξύ του state of the art και του ηλεκτρικού αναπηρικού αμαξιδίου.

Κεφάλαιο 3 Κόριο μέρος πτυχιακής εργασίας

3.1 *Ανάλυση Προβλήματος*

Ο σκοπός της πτυχιακής εργασίας, είναι να μετατραπεί ένα οποιοδήποτε συμβατικό αναπηρικό αμαξίδιο σε ηλεκτρικό, με στόχο την βελτίωση της ποιότητας ζωής των ατόμων με ειδικές ανάγκες.

Για να γίνει εφικτό, απαιτείται να δημιουργηθεί ένα ολοκληρωμένο μηχανικό σύστημα ελέγχου, το οποίο θα δέχεται εντολές από το χρήστη του αμαξιδίου μέσω ειδικών χειριστηρίων, θα τις επεξεργάζεται μέσω μίας ακολουθίας αλγορίθμων και θα δίνει τις κατάλληλες εντολές στους δύο κινητήρες για να περιστραφούν, ώστε να μετατραπεί η εντολή του χρήστη, σε κίνηση του οχήματος.

Η βασική προϋπόθεση είναι, να μπορεί να μεταφέρει έναν άνθρωπο με μέγιστο βάρος εκατό κιλά και σε ανηφορική κλίση της τάξεως του 30%.

Επίσης, το όλο σύστημα θα είναι ευέλικτο και εύκολο για να προσαρμοστεί σε όλα τα συμβατικά αμαξίδια της αγοράς, χωρίς να υπάρξει καμία μηχανολογική αλλαγή για την εγκατάσταση του και όταν απεγκατασταθεί, το αμαξίδιο θα επιστρέψει στην αρχική του μορφή.

3.2 *Είδη αναπηρικών αμαξιδίων*

Στην αγορά υπάρχουν πέντε διαφορετικά είδη συμβατικών χειροκίνητων αναπηρικών αμαξιδίων. Η κύρια διαφορά μεταξύ τους, είναι στο μέγεθος των οπίσθιων τροχών. Στην κατηγορία με τους μεγάλους τροχούς, ο ίδιος ο χρήστης του αμαξιδίου, μπορεί να το κινήσει χειροκίνητα, ενώ στις υπόλοιπες, απαιτείται η ύπαρξη ενός αρτιμελούς συνοδού και συνήθως χρησιμοποιούνται σε νοσοκομεία, για την εξυπηρέτηση των ασθενών.

Η παρούσα πτυχιακή εργασία θα ασχοληθεί μόνο με την κατηγορία των αμαξιδίων με τους μεγάλους τροχούς, καθώς αποτελούν τη συντριπτική πλειοψηφία που έχουν τεθεί σε κυκλοφορία και είναι και τα πλέον δημοφιλέστερα.

Παρόλα αυτά θα μπορούσε να προσαρμοστεί ένα αντίστοιχο μηχανικό σύστημα και σε αυτά, αλλά θα απαιτηθούν μεγαλύτερης ισχύος κινητήρες, καθώς η σχέση μετάδοσης σε μικρότερους τροχούς, είναι μικρότερη.



Α. Με μεγάλους τροχούς



Β. Με μεσαίους τροχούς και δοχείο.



Γ. Με μικρούς τροχούς



Δ. Με μεσαίους τροχούς



ΣΤ. Ειδικού τύπου

Εικόνα 7: Πέντε διαφορετικοί τύποι αμαξιδίων

3.3 Υπολογισμός κινητήρων

Όπως προαναφέρθηκε, η βασική προϋπόθεση είναι το αναπηρικό αμαξίδιο, μετά την εγκατάσταση του μηχανολογικού συστήματος, να είναι ικανό να μεταφέρει έναν άνθρωπο βάρους εκατό κιλών και να μπορεί να κινηθεί σε έδαφος με ανηφορική κλίση τριάντα τοις εκατό.

Σε όλα τα αναπηρικά αμαξίδια της αγοράς, με μεγάλους τροχούς, η διάμετρος των οπίσθιων τροχών είναι εξακόσια χιλιοστά.

Το αναπηρικό αμαξίδιο που θα χρησιμοποιήσουμε ζυγίζει είκοσι κιλά. Επομένως, θα πρέπει οι κινητήρες που θα τοποθετηθούν επάνω, να είναι ικανοί να μετακινήσουν το όχημα μαζί με τον ασθενή, με συνολική μάζα εκατόν είκοσι κιλά, με επιθυμητή ταχύτητα 1.4m/s στην ακραία συνθήκη ανάβασης.

Για να υπολογίσουμε τη γωνιακή ταχύτητα που θα έχει στην ακραία θέση ο κάθε τροχός ακολουθούμε τη σχέση $v = \omega * r$ συνεπώς, $\omega = v / r \leftrightarrow \omega = 1,4\text{m/s} / 0,3\text{m} \leftrightarrow \underline{\omega = 4,66 \text{ rad/sec}}$

Εφόσον η γωνιακή ταχύτητα του αμαξιδίου θα ισούται με $\omega = 2\pi * N / 60$ τότε οι στροφές ανά λεπτό προκύπτουν: $N = (\omega / 2\pi) * 60 \leftrightarrow N = (4,66 / 2\pi) * 60 \leftrightarrow \underline{N = 44,58 \text{ rpm}}$

Για τον υπολογισμό της ροπής χρειαζόμαστε πρώτα να υπολογίσουμε την ελάχιστη δύναμη που χρειάζεται για να μετακινηθεί το όχημα σε ανηφορική κλίση 30% .

Η κλίση 30% μεταφράζεται σε $\sin\theta = \sin(\arctan 0,3) \leftrightarrow \underline{\sin\theta = 0,28, \cos\theta = 0,96}$.

Ο συντελεστής τριβής προσδιορίζεται σε $\mu = 0,01$.

Ο κάθε κινητήρας θα αναλάβει από 60 κιλά βάρος, οπότε $m = 60\text{kg}$.

Οπότε η τελική εξίσωση για τον υπολογισμό της ελάχιστης δύναμης που απαιτείται είναι:

$$F = m * g * \sin\theta + \mu * m * g * \cos\theta = 60 * 9,806 * 0,28 + 0,01 * 60 * 9,806 * 0,96 \leftrightarrow \underline{F_{\min} = 170,4 \text{ Nt}}$$

Άρα η ελάχιστη ροπή που απαιτείται για να κινηθεί το όχημα είναι $M_{\min} = F_{\min} * r$
 $\leftrightarrow M_{\min} = 170,4 \text{ Nt} * 0,3\text{m} \leftrightarrow \underline{M_{\min} 51,12\text{Nm}}$.

Αντίστοιχα η ελάχιστη ισχύς που θα πρέπει να έχει ο κάθε κινητήρας είναι:

$P_{\min} = M_{\min} * \omega \leftrightarrow P_{\min} = 51,12\text{Nm} * 4,66\text{rad/sec} \leftrightarrow \underline{P_{\min} \approx 240\text{Watt}}$.

Επομένως οι κινητήρες που θα πρέπει να επιλέξουμε πρέπει να αποδίδουν την ισχύ των 250 watt έκαστος.

ΜΕΓΙΣΤΟ ΒΑΡΟΣ (kg)	ΜΕΓΙΣΤΟ ΒΑΡΟΣ ανά κινητήρα (kg)	ΔΙΑΜΕΤΡΟΣ ΤΡΟΧΟΥ (m)	Επιθυμητή μέγιστη ταχύτητα (m/s)	Στροφές ανά λεπτό (rpm)	Γωνιακή ταχύτητα (rad/sec)	ΔΥΝΑΜΗ (Nt)	ΡΟΠΗ (Nm)	ΙΣΧΥΣ (watt)
Αναπηρικά αμαξίδια με διάμετρο τροχού 600mm								
116,5	58,25	0,6	1,4	44,58599	4,666	165,4194	49,62581	231,5871
116	58	0,6	1,4	44,58599	4,666	164,7094	49,41283	230,5932
120	60	0,6	1,4	44,58599	4,666	170,3891	51,11672	238,5447
122	61	0,6	1,4	44,58599	4,666	173,2289	51,96866	242,5204
182	91	0,6	1,4	44,58599	4,666	258,4234	77,52702	361,7928
Αναπηρικά αμαξίδια με διάμετρο τροχού 300mm								
114	57	0,3	1,4	89,17197	9,333	161,8696	24,28044	226,6174
115	57,5	0,3	1,4	89,17197	9,333	163,2895	24,49343	228,6053
138	69	0,3	1,4	89,17197	9,333	195,9474	29,39211	274,3264
Αναπηρικά αμαξίδια με διάμετρο τροχού 200mm								
100	50	0,2	3,4	324,8408	34	141,9909	14,19909	482,769
110	55	0,2	4,4	420,3822	44	156,19	15,619	687,2359

Πίνακας 2: Ελάχιστη απαιτούμενη ισχύς για κάθε κινητήρα, για την κίνηση αναπηρικών αμαξιδίων, τριών διαφορετικών τύπων σε κλίση 30%.

3.4 Χειρισμός

Για να κινηθεί το αμαξιδίο απαιτείται η συνεργασία πολλών διαφορετικών υποσυστημάτων. Τα τρία βασικά υποσυστήματα είναι τα χειριστήρια, ο μικροελεγκτής και οι κινητήρες.

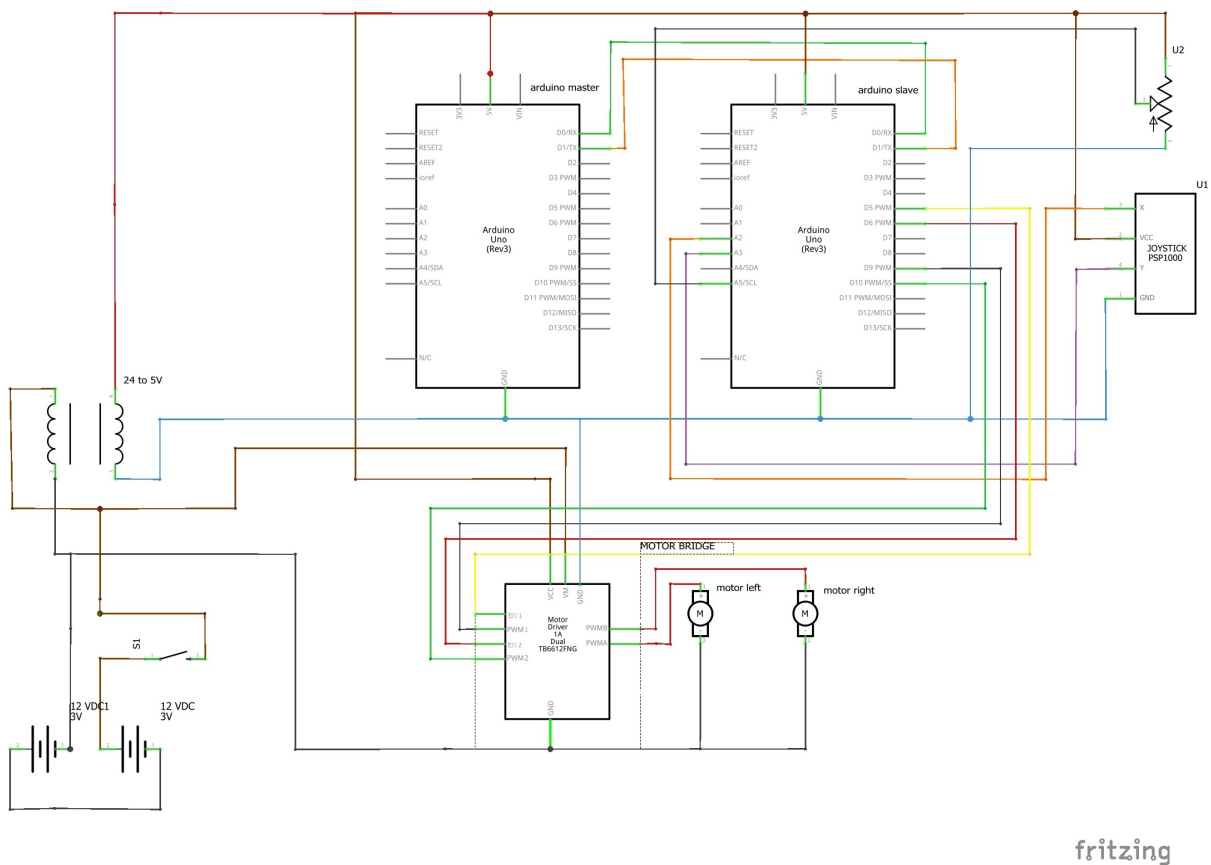
Ο χειρισμός του αμαξιδίου είναι πολύ σημαντικός και ένα από τα κύρια υποσυστήματα. Το χειριστήριο θα πρέπει να έχει μια σχεδίαση αρκετά φιλική προς το χρήστη, για να μπορεί να το χειρίζεται με ευκολία, ειδικά τα άτομα με ειδικές ανάγκες.

Οι πληροφορίες που θα μεταδίδονται από το χειριστήριο πρέπει να είναι ακριβείς για να μην υπάρχουν σφάλματα στην τελική κίνηση του οχήματος.

3.5 Σχεδιάγραμμα του ηλεκτρικού κυκλώματος

Παρακάτω ακολουθεί το σχεδιάγραμμα του ηλεκτρικού κυκλώματος που δημιουργήθηκε με το πρόγραμμα «fritzing».

Διακρίνουμε τους δύο μικροελεγκτές «Arduino UNO» που προσαρμόζονται επάνω τους με μορφή «Shield» η οθόνη LCD και η πλακέτα που αποτελεί την φωνητική εντολή, τους κινητήρες που συνδέονται με την Η-Γέφυρα και τέλος το χειριστήριο, το ποτενσιόμετρο, ο μετατροπέας τάσης, ο γενικός διακόπτης και οι συσσωρευτές.

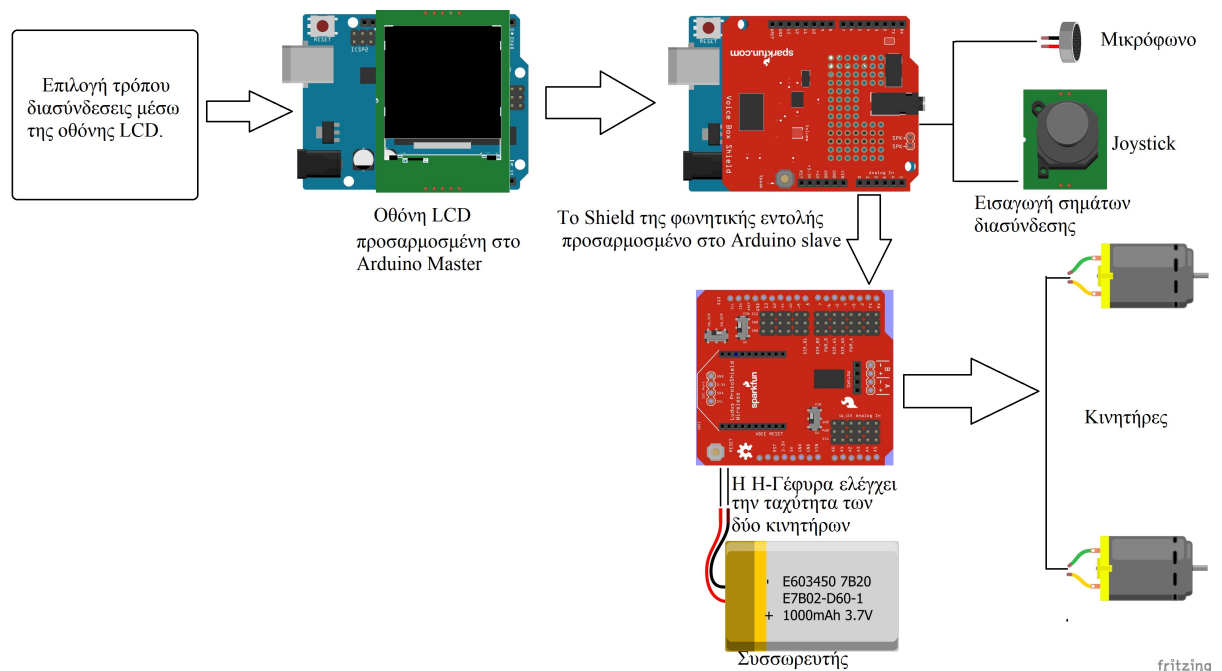


Εικόνα 8: Σχεδιάγραμμα ηλεκτρικού κυκλώματος.

3.6 Σχεδιάγραμμα του λογικού διαγράμματος

Παρακάτω παριστάνεται με ένα λογικό διάγραμμα, ο τρόπος διάδρασης του χρήστη με το μηχανικό σύστημα και πώς η εντολή του χρήστη καταφέρνει να ενεργοποιήσει κατάλληλα, την περιστροφή των κινητήρων. Η επιλογή του τρόπου επικοινωνίας με το χρήστη, γίνεται μέσω της οθόνης LCD που προσαρμόζεται στο «Arduino master».

Κατόπιν το σύστημα διαχειρίζεται την πληροφορία και δίνει εντολή στην Η-Γέφυρα να κινήσει τους δύο κινητήρες.

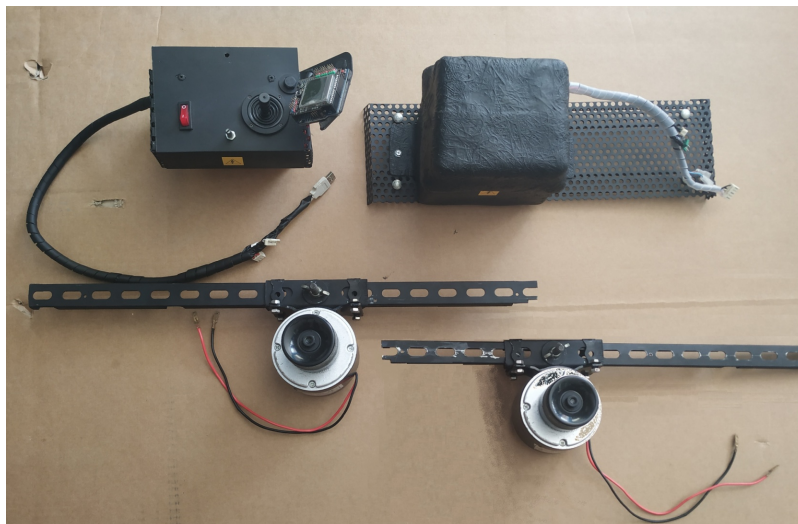


Εικόνα 9: Σχεδιάγραμμα λογικού διαγράμματος.

3.7 Τα τμήματα του KIT

Το KIT συναρμολόγησης αποτελείται από τέσσερα τμήματα τη κονσόλα χειρισμού, την θήκη συσσωρευτών και δύο βάσεις για τους κινητήρες. Τα παραπάνω τμήματα προσαρμόζονται επάνω στο όχημα «κουμπωτά» και απαιτούν μερικά δεματικά καλωδίων (tie wraps), για την στερέωση τους επάνω στο αμαξίδιο.

Ακολουθούν παρακάτω φωτογραφίες όπου φαίνονται τα τμήματα του «KIT» και η λεπτομερής επεξήγηση του καθενός.



Εικόνα 10: Τα τμήματα που αποτελούν το KIT.



Εικόνα 11: Το αμαξίδιο πριν και μετά την εγκατάσταση του ΚΙΤ.



Εικόνα 12: Όψεις του αναπηρικού αμαξιδίου με το ΚΙΤ.

3.8 Κονσόλα χειρισμού

Η κονσόλα χειρισμού είναι ο «εγκέφαλος» του «ΚΙΤ». Από αυτή δίνονται όλες οι εντολές από το χρήστη, είτε μέσω του χειριστηρίου είτε μέσω της φωνητικής εντολής.

Στην κονσόλα δεσπόζει η οθόνη «LCD», όπου είναι προσαρμοσμένη στο «Arduino master», το χειριστήριο, ο γενικός διακόπτης καθώς και το ποτενσιόμετρο που διορθώνει την περιστροφή των κινητήρων για να κινούνται με τις ίδιες στροφές.

Μέσα στην κονσόλα περιλαμβάνονται, ένας ακόμα μικροελεγκτής «Arduino UNO» που έχει προσαρμοστεί επάνω του το Shield φωνητικής εντολής και έχει οριστεί ως «slave», το μικρόφωνο και ένα ηχείο.



Εικόνα 13: Κονσόλα χειρισμού.

3.9 Θήκη συσσωρευτών

Η θήκη συσσωρευτών μπορεί να θεωρηθεί ως « η καρδιά του οχήματος», καθώς μέσα της περικλείει τους δύο συσσωρευτές των 12V έκαστος, που τροφοδοτούν το όχημα με ηλεκτρική ενέργεια. Επίσης μέσα της βρίσκεται και η Η-Γέφυρα που ελέγχει τη περιστροφή των κινητήρων. Ακόμα μέσα στη θήκη περικλείεται ένας μετατροπέας τάσης από 24V σε 5V και τέλος μία ασφάλεια των 10A.



Εικόνα 14: Η θήκη συσσωρευτών.

4.1 Βάσεις κινητήρων

Οι δύο κινητήρες προσαρμόστηκαν επάνω σε «συρταρωτές» βάσεις, που επιτρέπουν την εύκολη σύμπλεξη και αποσύμπλεξη, με τον τροχό του αμαξιδίου.

Η προσαρμογή των βάσεων επάνω στο αμαξίδιο απαιτεί μόνο μερικά δεματικά καλωδίων, “Tie wraps” και έτσι αποφεύγεται η χρήση κοχλιών, που θα καθιστούσαν την εγκατάσταση πιο σύνθετη.

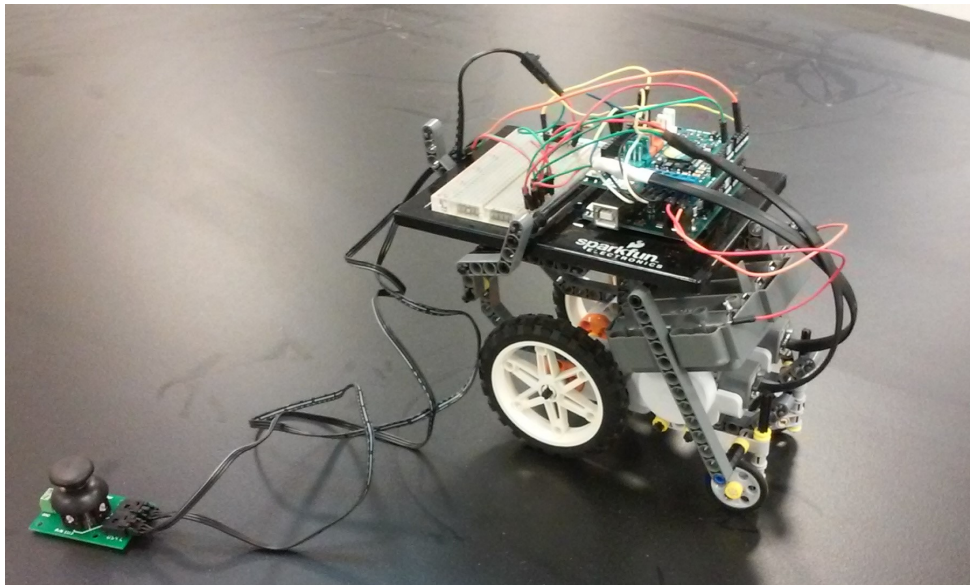


Εικόνα 15: Βάσεις κινητήρων.

Κεφάλαιο 4 Υλοποίηση

Για την υλοποίηση της εργασίας, κρίθηκε αναγκαίο να κατασκευαστεί ένα όχημα σε κλίμακα, για να δοκιμαστούν πάνω σε αυτό τα ηλεκτρονικά συστήματα και να ρυθμιστούν τα χειριστήρια ούτως ώστε να προσομοιωθεί ο χειρισμός του αμαξιδίου και να λυθούν όσα προβλήματα προκύψουν.

Το όχημα αυτό κατασκευάστηκε στο εργαστήριο ρομποτικής του τμήματος μηχανολογίας του Τ.Ε.Ι. Κρήτης και αποτελείται από εξαρτήματα και κινητήρες της LEGO.



Εικόνα 16: Αμαξίδιο κατασκευασμένο από εξαρτήματα LEGO.

4.1 Χειριστήριο (JOYSTICK - JH-D202X-R4)

Ο πρώτος και βασικότερος τρόπος χειρισμού του αναπηρικού οχήματος αποφασίστηκε να είναι ένα χειριστήριο.

Η φιλικότητα και η εξοικείωση της χρήσης του χειριστηρίου, από πλήθος άλλων εφαρμογών, όπως τα ηλεκτρονικά παιχνίδια, τα τηλεκατευθυνόμενα οχήματα, ο χειρισμός μηχανημάτων κ.λπ., που ερχόμαστε καθημερινά σε επαφή μαζί τους, καθιστά το χειριστήριο βασικό τρόπο χειρισμού και στα αναπηρικά αμαξίδια.

Τα συνηθέστερα χειριστήρια αποτελούνται από δύο ποτενσιόμετρα που εκφράζουν τους άξονες X και Y, οι οποίοι είναι βαθμονομημένοι εκ κατασκευής.

Στη μέση των δύο αξόνων βρίσκεται το σημείο αναφοράς ($X=0$, $Y=0$), όπως ακριβώς συμβαίνει στο καρτεσιανό επίπεδο.

Ο άξονας του χειριστηρίου είναι ελεύθερος να κινηθεί και στους δύο άξονες X και Y, μέσω ενός μηχανισμού και ανάλογα σε ποια θέση του κάθε άξονα βρίσκεται διαμορφώνεται η διεύθυνση και η τιμή της ταχύτητας. Για παράδειγμα αν η θέση του χειριστηρίου είναι προς τα πίσω στο τέλος της διαδρομής του άξονα, δηλαδή $X=0$ και $Y=-MAX$ το όχημα θα κινηθεί με τη μέγιστη ταχύτητα προς τα πίσω. Ακολούθως για τη θέση του χειριστηρίου προς τα δεξιά στη μέση της διαδρομής του άξονα X, δηλαδή $X=+MAX/2$ και $Y=0$, τότε το όχημα θα περιστραφεί προς τα δεξιά με την μισή ταχύτητα περιστροφής.

Συνδυάζοντας κάθε πιθανή θέση του χειριστηρίου στους δύο άξονες μπορούμε να έχουμε οκτώ διαφορετικές διευθύνσεις στο χώρο με διαφορετικές ταχύτητες.



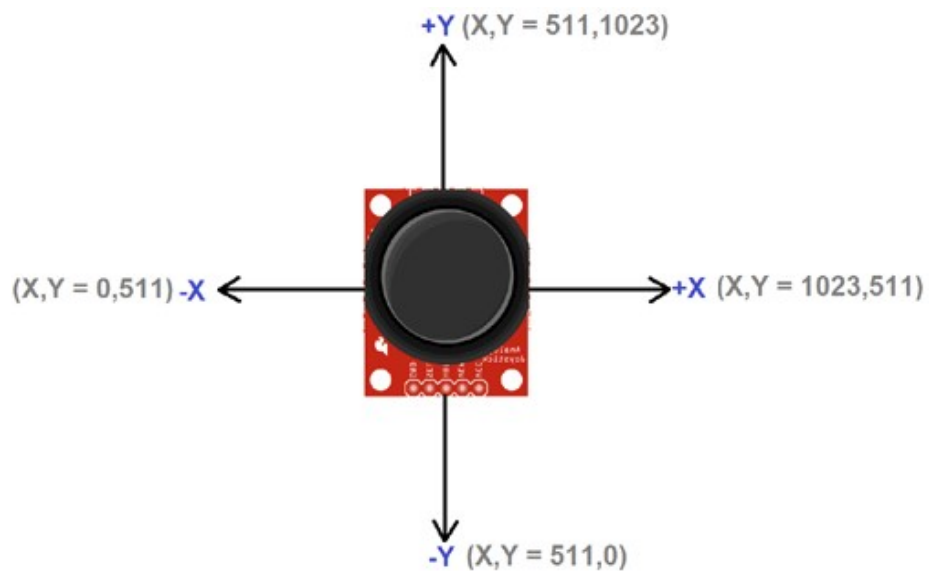
Εικόνα 17: Χειριστήριο δύο αξόνων και οκτώ διαφορετικών κατευθύνσεων (JH-D202X-R4).

Για τον προγραμματισμό και το χειρισμό του δοκιμαστικού αμαξιδίου, χρησιμοποιήθηκε ένα χειριστήριο του εργαστηρίου της ρομποτικής στο Τ.Ε.Ι. ΚΡΗΤΗΣ. Πρόκειται για ένα χειριστήριο δύο αξόνων X,Y και 8 διαφορετικών διευθύνσεων, ίδιων προδιαγραφών με αυτό που θα χρησιμοποιηθεί στο τελικό στάδιο αλλά με σαφώς μικρότερο μέγεθος μη κατάλληλο για άτομα με ειδικές ανάγκες. Πάραυτα ο κώδικας που γράφτηκε πάνω σε αυτό το χειριστήριο θα μεταφερθεί αυτούσιος με ορισμένες τροποποιήσεις στο τελικό σχέδιο του αμαξιδίου.

Τα χειριστήριο αποτελείται από δύο ποτενσιόμετρα που αντιπροσωπεύουν τους δύο άξονες του καρτεσιανού επιπέδου. Η εργοστασιακή τους βαθμονόμηση, ξεκινάει από μια κλίμακα της τάξεως, από 0 έως 1023, όπου αυτοί οι αριθμοί είναι και η τάση στην έξοδο του ποτενσιόμετρου σε mV. Με πιο απλά λόγια, οι μηδενικές και οι μέγιστες τιμές των ποτενσιόμετρων, βρίσκονται στις ακραίες θέσεις του χειριστηρίου και οι ενδιάμεσες, σε όλες τις πιθανές θέσεις που μπορεί να καταλάβει ο άξονας που συνδυάζει τα δύο ποτενσιόμετρα και αποτελεί και τον κέρσορα του χειριστηρίου.

Στο σημείο που τέμνονται οι δύο άξονες, το οποίο αποτελεί και το σημείο αναφοράς του χειριστηρίου, οι διαδρομές και των δύο ποτενσιόμετρων είναι στη μέση, οπότε και η τιμή της τάσης στην έξοδο τους θα είναι η μισή της μέγιστης δηλαδή $V_{\text{μέσης}} = V_{\text{MAX}}/2 = 1023/2 \approx 511$ mV. Έτσι αν η μέγιστη τιμή στον άξονα X και στον άξονα Y, βρίσκεται στο τέρμα της διαδρομής στην θετική κατεύθυνση προκύπτει το παρακάτω σχεδιάγραμμα.

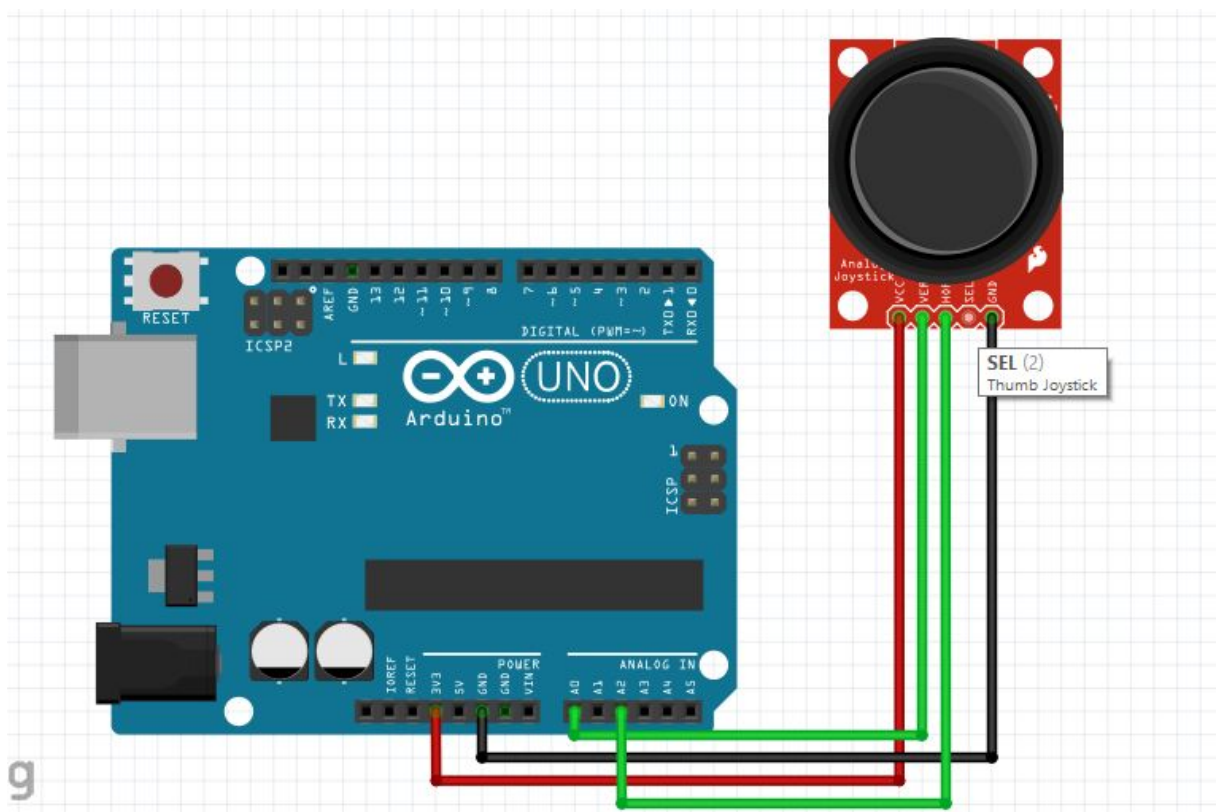
Αντιλαμβάνεται λοιπόν κανείς, ότι κινώντας τον κέρσορα σε οποιαδήποτε πιθανή θέση, μπορούμε να προσδιορίσουμε την κατεύθυνση και την ταχύτητα που θα κινηθεί το όχημα.



Εικόνα 18: Οι άξονες X και Y καθώς και οι τιμές των ποτενσιόμετρων στις ακραίες θέσεις.

Αφού προσδιορίστηκε και κατανοήθηκε ο τρόπος λειτουργίας του χειριστηρίου, έφτασε η ώρα της σύνδεσης του με την πλατφόρμα «Arduino UNO». Το συγκεκριμένο χειριστήριο διαθέτει πέντε επαφές (Pin) οι οποίες είναι:

1. Vin - Με αυτή την επαφή θα τροφοδοτηθεί το χειριστήριο με τάση 5V.
2. Gnd - Σε αυτή την επαφή θα γειώσουμε το χειριστήριο.
3. VRx - Με αυτή την επαφή θα ελέγχεται το ποτενσιόμετρο του άξονα X
4. VRy - Με αυτή την επαφή θα ελέγχεται το ποτενσιόμετρο του άξονα Y
5. SW - Αυτή η επαφή είναι το μπουτόν του χειριστηρίου την οποία δεν θα χρησιμοποιήσουμε



Εικόνα 19: Σύνδεση του χειριστηρίου στην πλατφόρμα «Arduino UNO».

4.2 Φωνητική εντολή

Ένας εκ των τριών βασικών τρόπων χειρισμού, του αναπηρικού οχήματος, κρίθηκε αυτός της φωνητικής εντολής. Η απόφαση αυτή προέκυψε, από την εύκολη και άμεση διασύνδεση που μπορεί να προσφέρει η απλή χρήση της φωνής του χειριστή, με το όχημα.

Αυτός ο τρόπος χειρισμού, καθιστά το τελικό αποτέλεσμα πιο προσιτό και φιλικά αποδεκτό, από εκείνα τα άτομα που τα κινητικά τους προβλήματα, δεν τους επιτρέπουν άλλο τρόπο αλληλεπίδρασης με το όχημα, παρά μόνο με την χρήση της ομιλίας.

Για την υλοποίηση του υποσυστήματος χειρισμού του οχήματος, μέσω της ομιλίας, χρησιμοποιήθηκε ως βασικό στέλεχος της μηχαντρονικής κατασκευής, η εξειδικευμένη πλακέτα (Shield) «EasyVR 3.0», της κατασκευάστριας εταιρίας ψηφιακών και ηλεκτρονικών συσκευών, με την ονομασία «VeeR».

Η πλακέτα αυτή, είναι ικανή να ψηφιοποιήσει την ανθρώπινη ομιλία και να την αποδώσει σε ένα ψηφιακό παλμό, ο οποίος παλμός, αξιοποιείται από το υπόλοιπο κύκλωμα, ώστε να μεταφέρει την επιθυμητή εντολή του χρήστη, προς τους κινητήρες, προκαλώντας την κίνηση και τον έλεγχο του οχήματος.

Βασικά μέρη της ολοκληρωμένης πλακέτας «EasyVR», είναι το κύριο στέλεχος, που ευθύνεται για την καταγραφή και ψηφιοποίηση του ηχητικού σήματος του πομπού, μέσω ενός μικροφώνου, που συνδέεται πάνω στο σώμα της, μέσω ενός ταχυσυνδέσμου. Η παραπάνω πλακέτα, προσαρμόζεται σε μια δεύτερη και μεγαλύτερη πλακέτα, η οποία αποτελεί τη γέφυρα του ειδικού εξαρτήματος, επάνω στην κύρια πλατφόρμα «Arduino UNO».

Εκτός της εισαγωγής δεδομένων μέσω μικροφώνου, η συγκεκριμένη πλακέτα προσφέρει τη δυνατότητα εξαγωγής ηχητικών σημάτων μέσω δύο ηχητικών εξόδων. Η πρώτη είναι μια γραμμή εξόδου, όπου μπορεί να συνδεθεί ένα βύσμα τύπου «jack 3.5mm» των 3,5 χιλιοστών και η δεύτερη ένας ταχυσύνδεσμος, ίδιος

με τον αντίστοιχο για την είσοδο του μικροφώνου, όπου μπορεί να συνδεθεί ένα απλό ηχείο. Και στις δύο περιπτώσεις, η πλακέτα προσφέρει στον χρήστη ακόμα μία άμεση δυνατότητα ανάδρασης, ώστε να καταλαβαίνει ο χρήστης ότι η εντολή που δόθηκε έχει εκτελεστεί ή όχι.

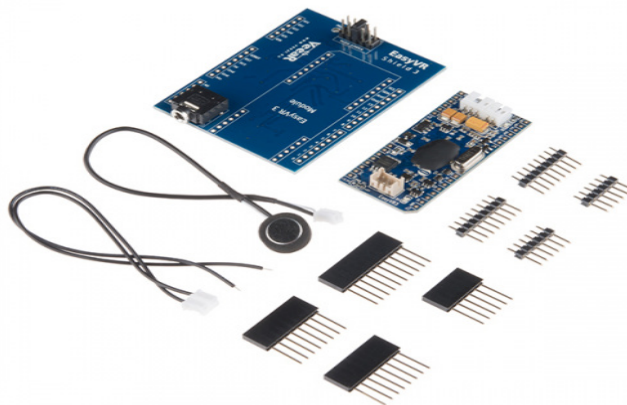
Τα τεχνικά χαρακτηριστικά του EasyVR 3.0 είναι:

- Τύπος Shield: Αισθητήρας
- Τύπος αισθητήρα: Ήχου
- Τυπική τάση εισόδου: 3.3VDC

5 VDC

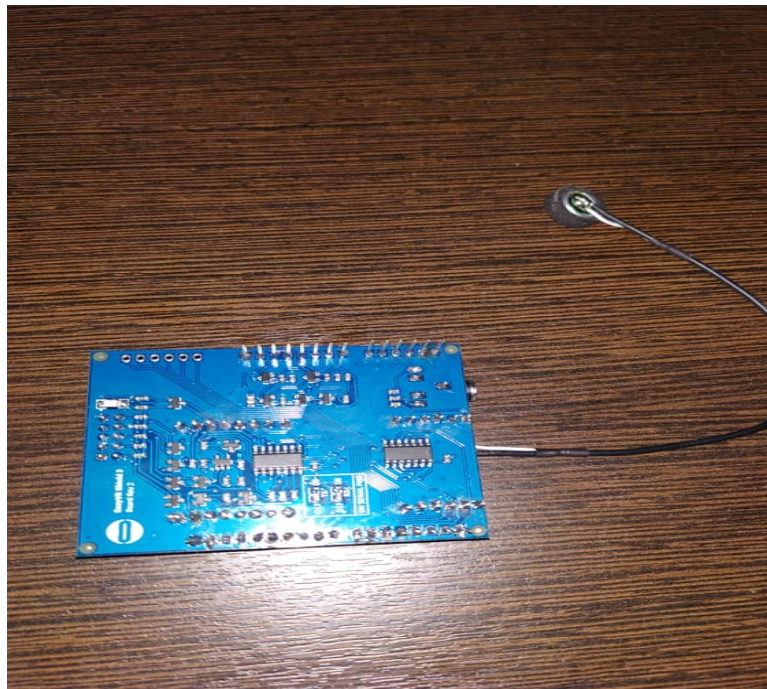
- Ρεύμα λειτουργίας: 25mA
- Διασύνδεση: Ψηφιακή
- Πρωτόκολλο επικοινωνίας: UART

Το ολοκληρωμένο σύστημα της VeeR, «EasyVR 3.0» αποτελείται από δώδεκα (12) εξαρτήματα, τα οποία απαιτείται να συναρμολογηθούν και να συγκολληθούν μεταξύ τους, καθώς η εταιρεία τα διαθέτει ξεχωριστά, ως ένα κιτ συναρμολόγησης και όχι έτοιμα εξ αρχής.

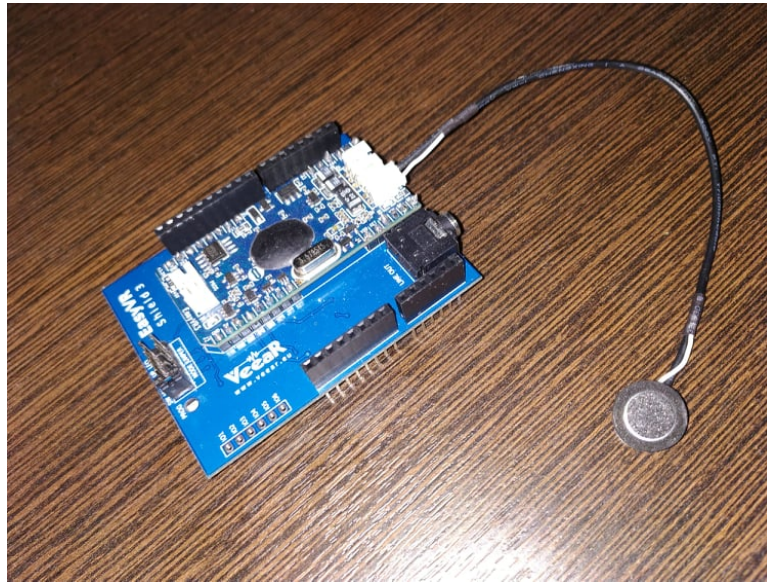


Εικόνα 20: Το KIT συναρμολόγησης EasyVR3.0

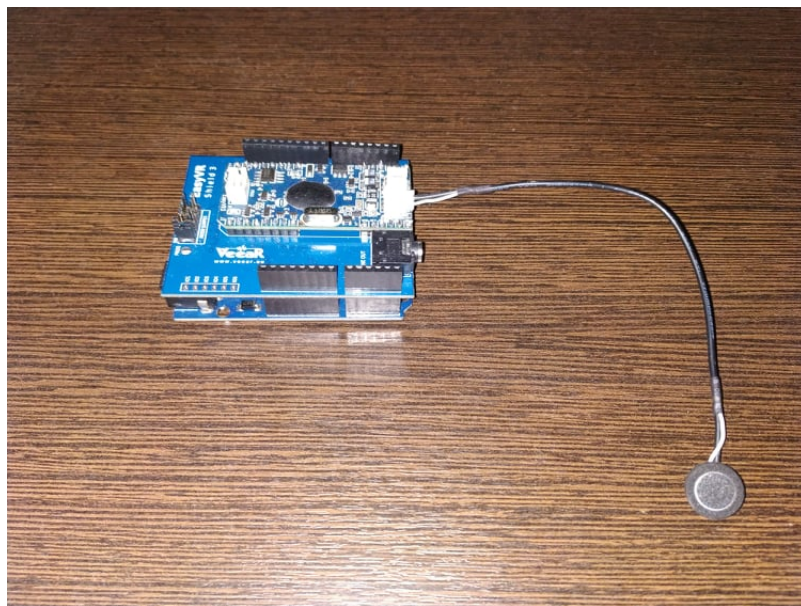
Έπειτα από τη συναρμολόγηση και τις ογδόντα οκτώ (88) συγκολλήσεις τύπου «καλάι» που απαιτούνται για να συγκροτηθεί το «EasyVR 3.0», προσαρμόζεται μέσω των ποδιών (pin) επάνω στην πλατφόρμα της «Arduino» την «UNO» και είναι έτοιμη για να προγραμματιστεί μέσω της εφαρμογής «Arduino» με την βοήθεια ενός ηλεκτρονικού υπολογιστή.



Εικόνα 21: Οπίσθια όψη του πλήρως συναρμολογημένου και συγκολλημένου «EasyVR 3.0».



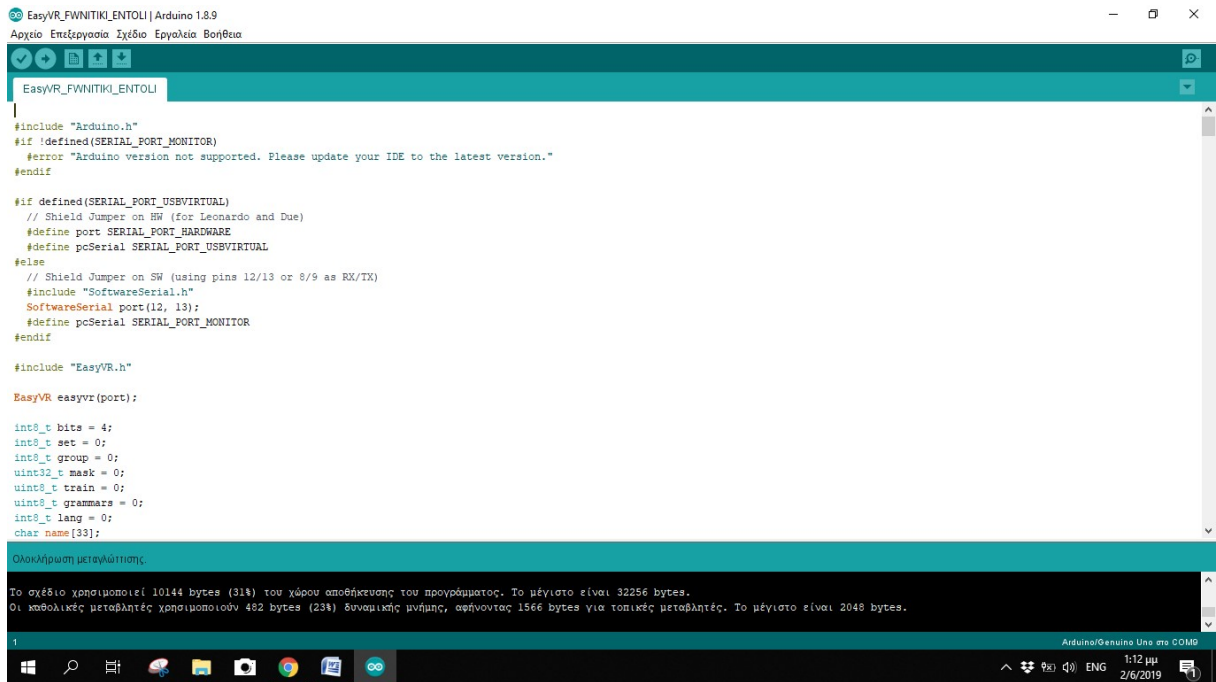
Εικόνα 22: Εμπρόσθια όψη του πλήρως συναρμολογημένου και συγκολλημένου «EasyVR 3.0» .



Εικόνα 23:Το «Easy VR 3.0» επάνω στην πλατφόρμα «Arduino UNO».

4.3 Προγραμματισμός και λογισμικό φωνητικής εντολής

Εφόσον η πλακέτα έχει συγκροτηθεί και τοποθετηθεί, επάνω στο κύριο σώμα του «Arduino UNO», με τη βοήθεια του λογισμικού της «Arduino», είναι έτοιμη να προγραμματιστεί στις ανάγκες της παρούσας εφαρμογής.



```
#include "Arduino.h"
#ifdef SERIAL_PORT_MONITOR
#error "Arduino version not supported. Please update your IDE to the latest version."
#endif

#ifdef SERIAL_PORT_USBVIRTUAL
// Shield Jumper on HW (for Leonardo and Due)
#define port SERIAL_PORT_HARDWARE
#define pcSerial SERIAL_PORT_USBVIRTUAL
#else
// Shield Jumper on SW (using pins 12/13 or 8/9 as RX/TX)
#include "SoftwareSerial.h"
SoftwareSerial port(12, 13);
#define pcSerial SERIAL_PORT_MONITOR
#endif

#include "EasyVR.h"

EasyVR easyvr(port);

int8_t bits = 4;
int8_t set = 0;
int8_t group = 0;
uint32_t mask = 0;
uint8_t train = 0;
uint8_t grammars = 0;
int8_t lang = 0;
char name[33];
```

Ολοκλήρωση μεταγλώττισης.

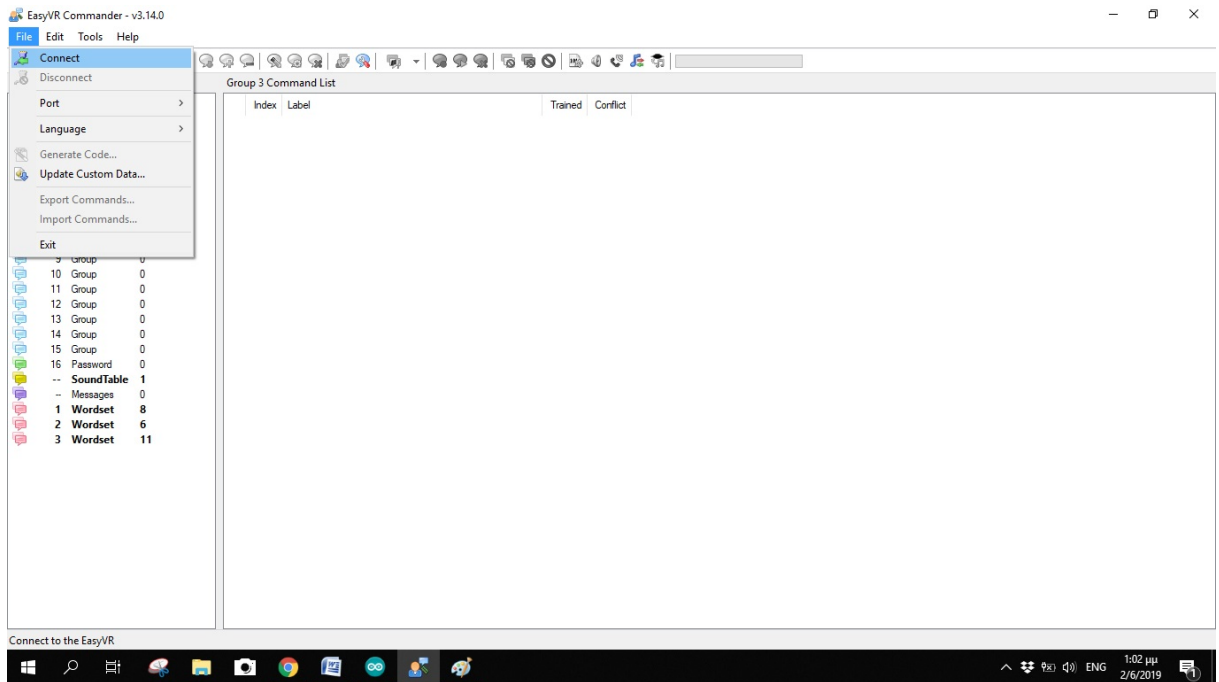
Το σχέδιο χρησιμοποιεί 10144 bytes (31%) του χώρου αποθήκευσης του προγράμματος. Το μέγιστο είναι 32256 bytes.
Οι καθολικές μεταβλητές χρησιμοποιούν 482 bytes (23%) δυναμικής μνήμης, αφήνοντας 1566 bytes για τοπικές μεταβλητές. Το μέγιστο είναι 2048 bytes.

Εικόνα 24: Το πρόγραμμα που γράφτηκε μέσω του λογισμικού της «Arduino».

Όπως γίνεται κατανοητό, για την εκτέλεση των εντολών που θα δώσει ο χρήστης, θα πρέπει να έχουν καθοριστεί και αποθηκευτεί, πρώτα μέσα στο ίδιο το πρόγραμμα. Οι προκαθορισμένες αυτές εντολές, είναι πέντε: «μπροστά», «πίσω», «αριστερά», «δεξιά» και «στόπ». Οι οποίες γίνεται αυτονόητο και το τι πρόκειται να εκτελέσουν.

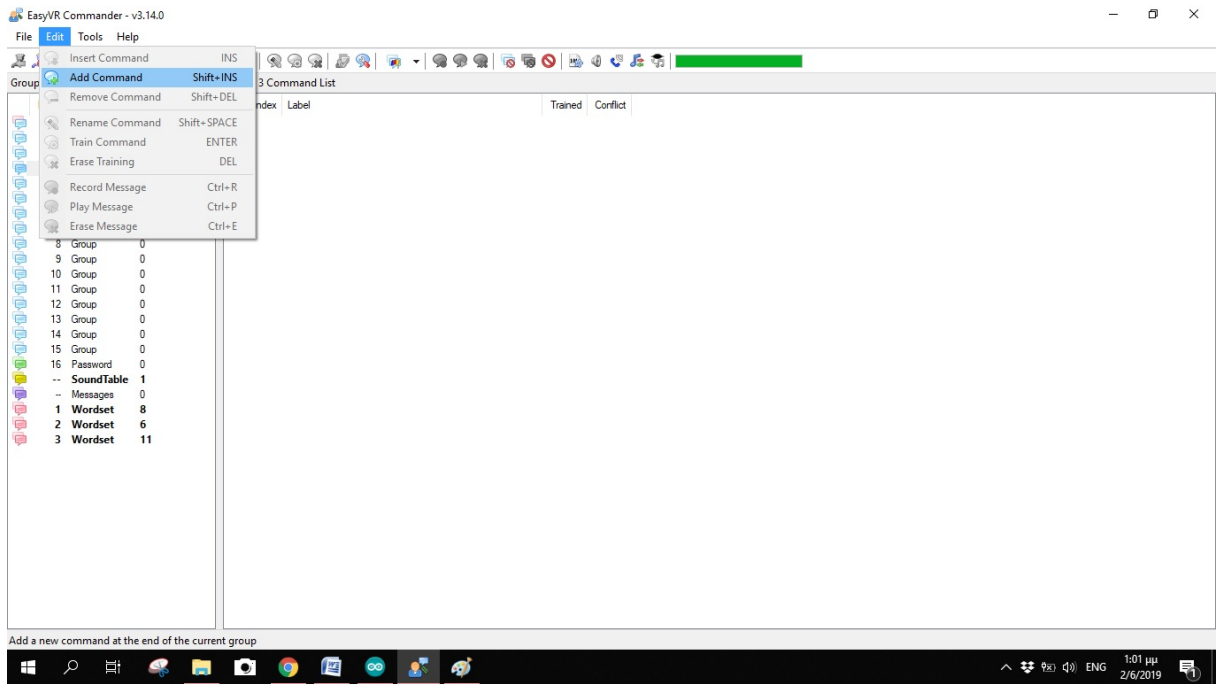
Η εταιρεία «VeeR», προσφέρει δωρεάν μέσα από το διαδικτυακό ιστότοπο της, ένα λογισμικό, το «EasyVR Commander» στην έκδοση «3.14.0», το οποίο επιτρέπει την εισαγωγή προκαθορισμένων εντολών, μέσω του «Easy VR Shield 3», τις οποίες μπορούμε να εισάγουμε μετέπειτα στο πρόγραμμα και όταν θα δίνεται η τελική εντολή από το χρήστη, να την αναγνωρίζει και να την ταυτίζει με τις προκαθορισμένες εντολές και να εκτελεί την σωστή.

Ακολουθούν αναλυτικές εικόνες, που επεξηγούν βήμα - βήμα την παραπάνω διαδικασία. Αρχικά, όπως φαίνεται στην ακόλουθη εικόνα, πρέπει να συνδεθεί η πλακέτα «EasyVR 3.0», με το λογισμικό. Επιλέγοντας το εικονίδιο “FILE”, στην επιλογή “Connect”, το λογισμικό θα συνδεθεί με την πλακέτα και θα αρχίσει η επικοινωνία μεταξύ τους.



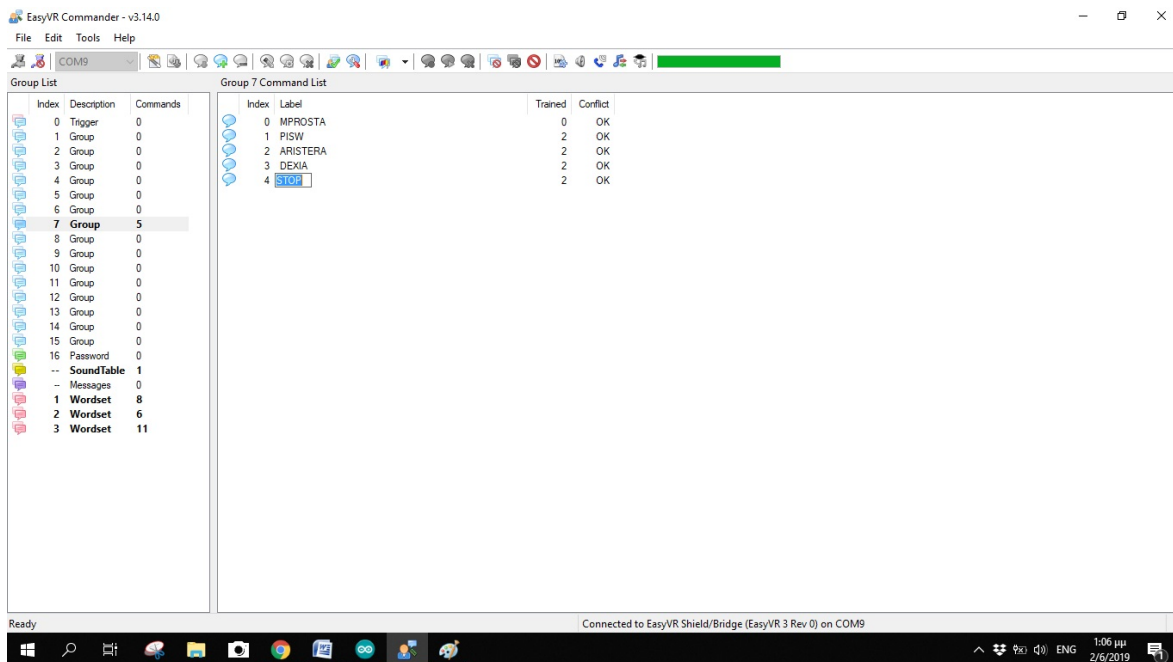
Εικόνα 25: Σύνδεση λογισμικού «EasyVR Commander» με την πλακέτα «EasyVR 3.0».

Επόμενο βήμα, είναι η προσθήκη της επιθυμητής εντολής, που θα προϋπάρχει μέσα στο τελικό πρόγραμμα. Μέσα σε ένα οποιοδήποτε από τα δεκαπέντε «γκρούπ», που βρίσκονται στην αριστερή καρτέλα, κάνοντας αριστερό «κλικ» με το «ποντίκι», στην επιλογή “Edit”, εμφανίζεται η επιλογή “Add Command”. Την επιλέγουμε και είμαστε έτοιμοι να κατασκευάσουμε την πρώτη προκαθορισμένη εντολή.



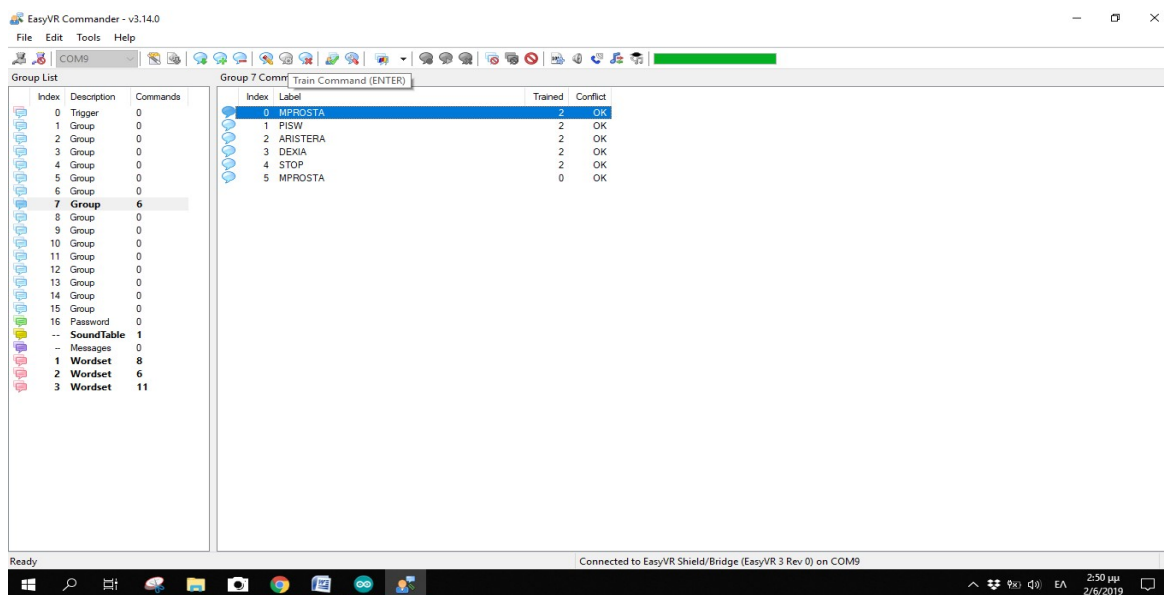
Εικόνα 26: Η επιλογή “Add Command” μέσα στο λογισμικό.

Πρώτο μέλημα, είναι να της προσδώσουμε ένα όνομα με λατινικούς χαρακτήρες, που θα βοηθήσει αργότερα στον προγραμματισμό, καθώς δεν αναγνωρίζονται οι ελληνικοί χαρακτήρες, όπως διακρίνεται στην επόμενη εικόνα.



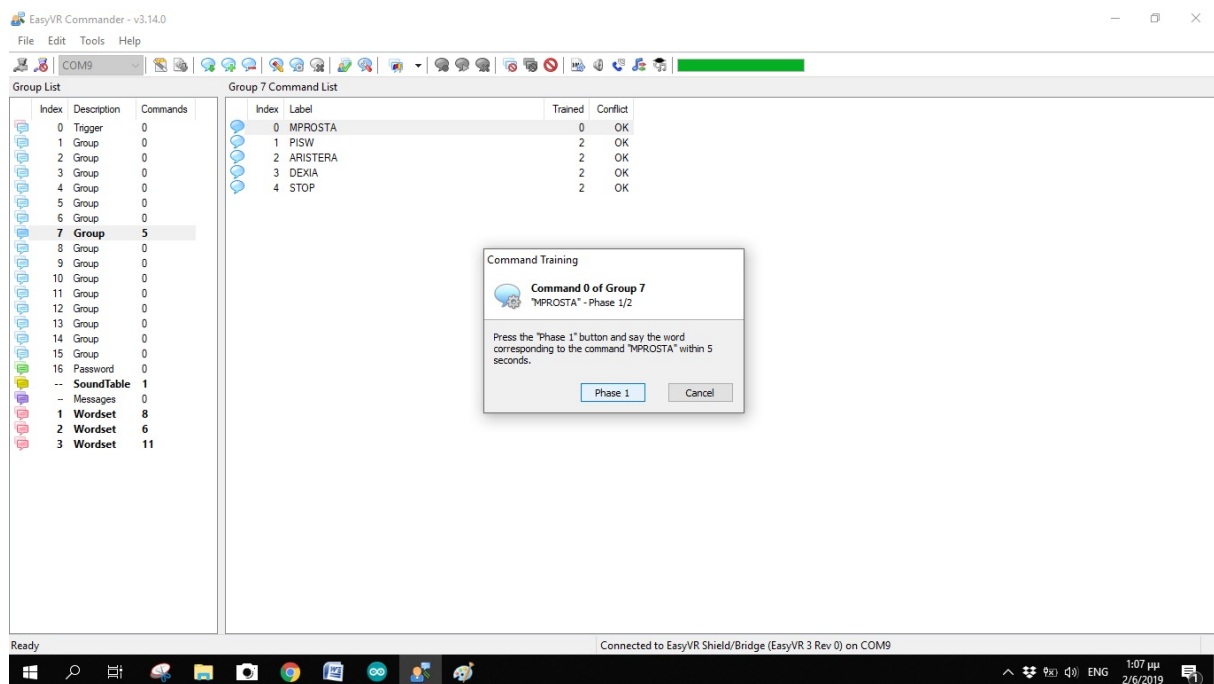
Εικόνα 27: Οι πέντε προκαθορισμένες εντολές με λατινικούς χαρακτήρες.

Έφτασε η ώρα της ηχογράφησης και απομνημόνευσης της πρώτης εντολής. Έχοντας προεπιλέξει την εντολή που θέλουμε να ηχογραφήσουμε, πατάμε από την γραμμή εντολών την επιλογή “Train Command (ENTER)”.



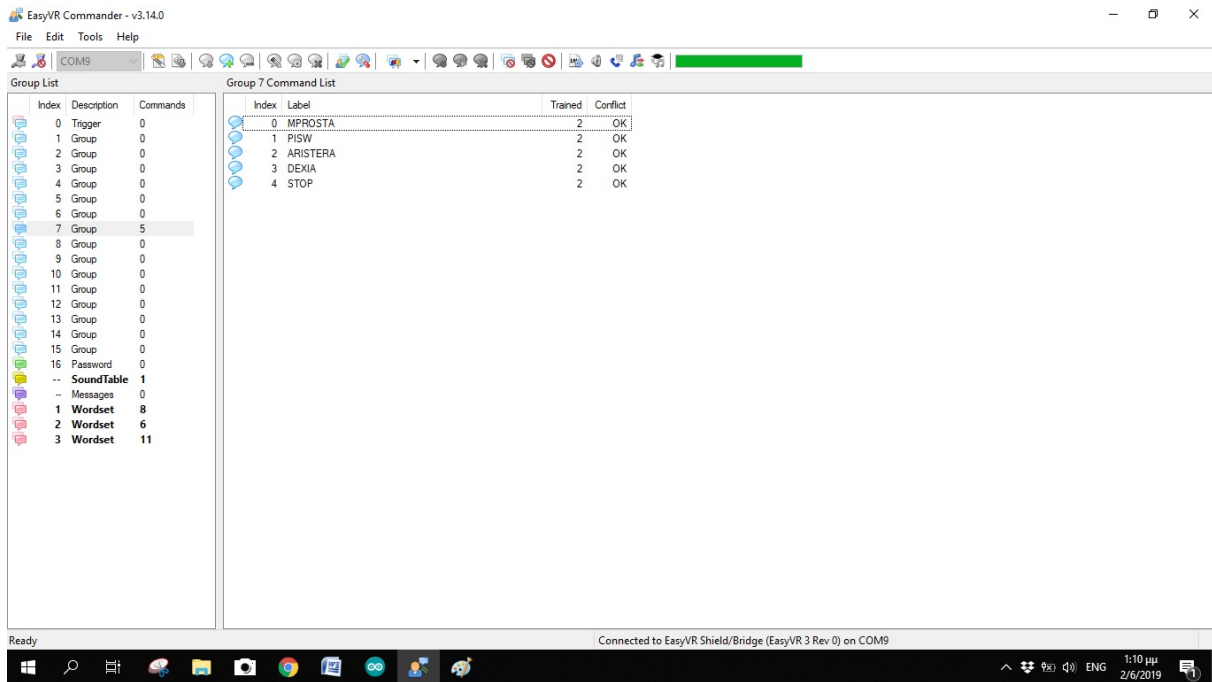
Εικόνα 28: Η επιλογή “Train Command (ENTER)”.

Κατόπιν, εμφανίζεται στην κεντρική οθόνη μία ειδοποίηση, η οποία μας πληροφορεί για την επικείμενη εκκίνηση, της διαδικασίας της καταγραφής. Κάνουμε «κλικ» στην επιλογή “Phase 1” και μας ειδοποιεί το λογισμικό, για την πρώτη ηχογράφηση που θα κάνουμε. Η διαδικασία θα ολοκληρωθεί σε δύο διαδοχικές φάσεις ηχογραφήσεων.



Εικόνα 29: Εκκίνηση καταγραφής της εντολής.

Μόλις τελειώσει η πρώτη καταγραφή, εμφανίζεται ένα αντίστοιχο μήνυμα στο κέντρο της οθόνης, που μας ειδοποιεί για την εκκίνηση, της δεύτερης φάσης ηχογράφησης. Επαναλαμβάνουμε την συγκεκριμένη διαδικασία και για τις υπόλοιπες τέσσερις εντολές και η διαδικασία καταγραφής και απομνημόνευσης των προεπιλεγμένων εντολών, περατώνεται σε αυτό το σημείο.

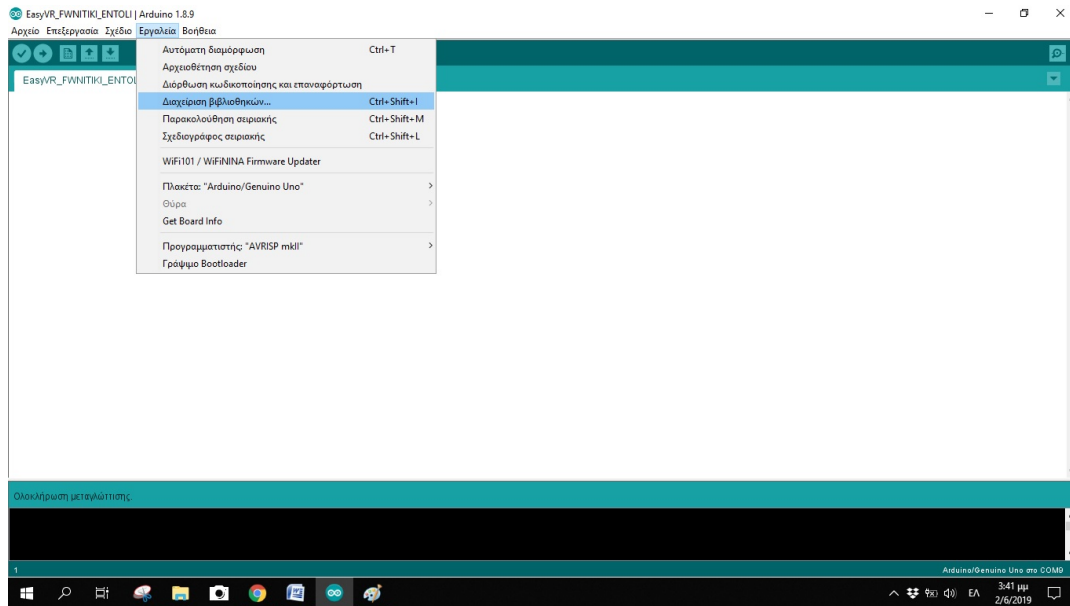


Εικόνα 30: Πέρασ της καταγραφής των ηχογραφήσεων για τις πέντε εντολές.

4.4 Συγγραφή κώδικα φωνητικής εντολής

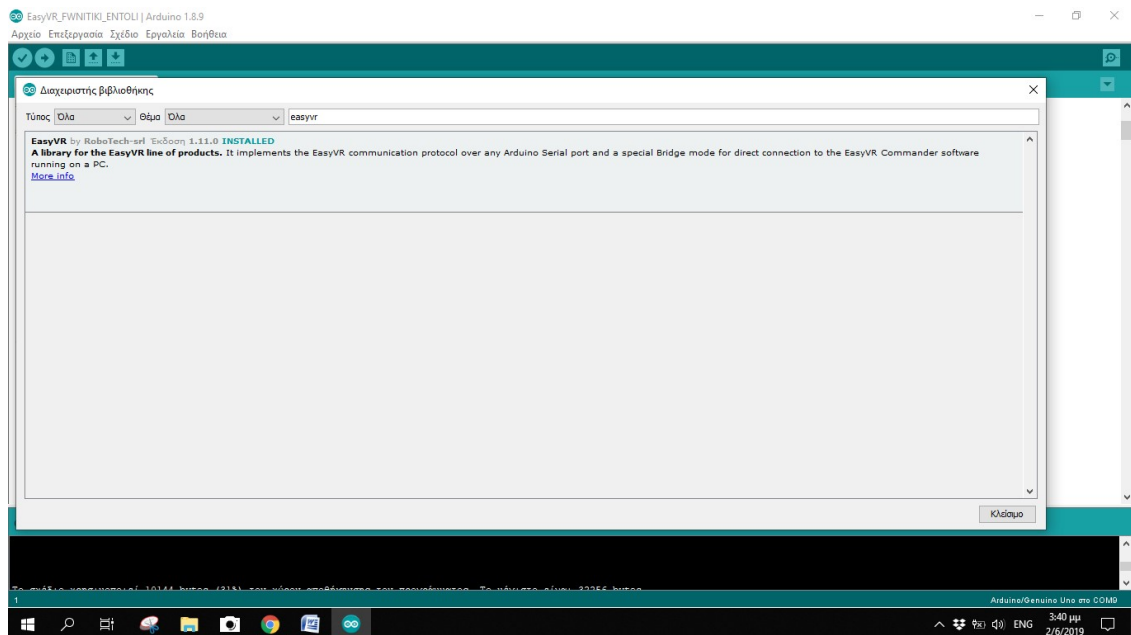
Απαραίτητο εργαλείο για να ξεκινήσει η συγγραφή του κώδικα είναι η εγκατάσταση της βιβλιοθήκης “EasyVR”. Η διαδικασία εγκατάστασης της συγκεκριμένης βιβλιοθήκης, είναι αρκετά απλή και θα περιγραφεί ευθύς αμέσως, στα παρακάτω βήματα.

Αρχικά, εκκινώντας την εφαρμογή “Arduino”, διακρίνεται στην γραμμή εντολών, η επιλογή «εργαλεία». Επιλέγοντας την, εμφανίζεται ένα υπομενού, με την επιλογή «Διαχείριση βιβλιοθηκών». Στη συνέχεια, σε ένα νέο παράθυρο που εμφανίζεται, βρίσκεται μια μπάρα αναζήτησης, στην οποία πληκτρολογούμε το όνομα της βιβλιοθήκης, που θέλουμε να εγκαταστήσουμε.



Εικόνα 31: Η επιλογή «Διαχείριση βιβλιοθηκών».

Το πρόγραμμα την εντοπίζει και μας την εμφανίζει και είμαστε έτοιμοι να την εγκαταστήσουμε επιλέγοντας την επιλογή «εγκατάσταση» από το αντίστοιχο πεδίο. Αφού ακολουθήσουμε τα παραπάνω βήματα, θα έχουμε φτάσει στο σημείο που εμφανίζεται στην επόμενη εικόνα.



Εικόνα 32: Ολοκλήρωση εγκατάστασης της βιβλιοθήκης "EasyVR".

Σε αυτό το σημείο, μπορούμε να ξεκινήσουμε την συγγραφή του κώδικα, που θα αποτελέσει το εργαλείο πραγματοποίησης της ομιλίας, σε μηχανικό έργο.

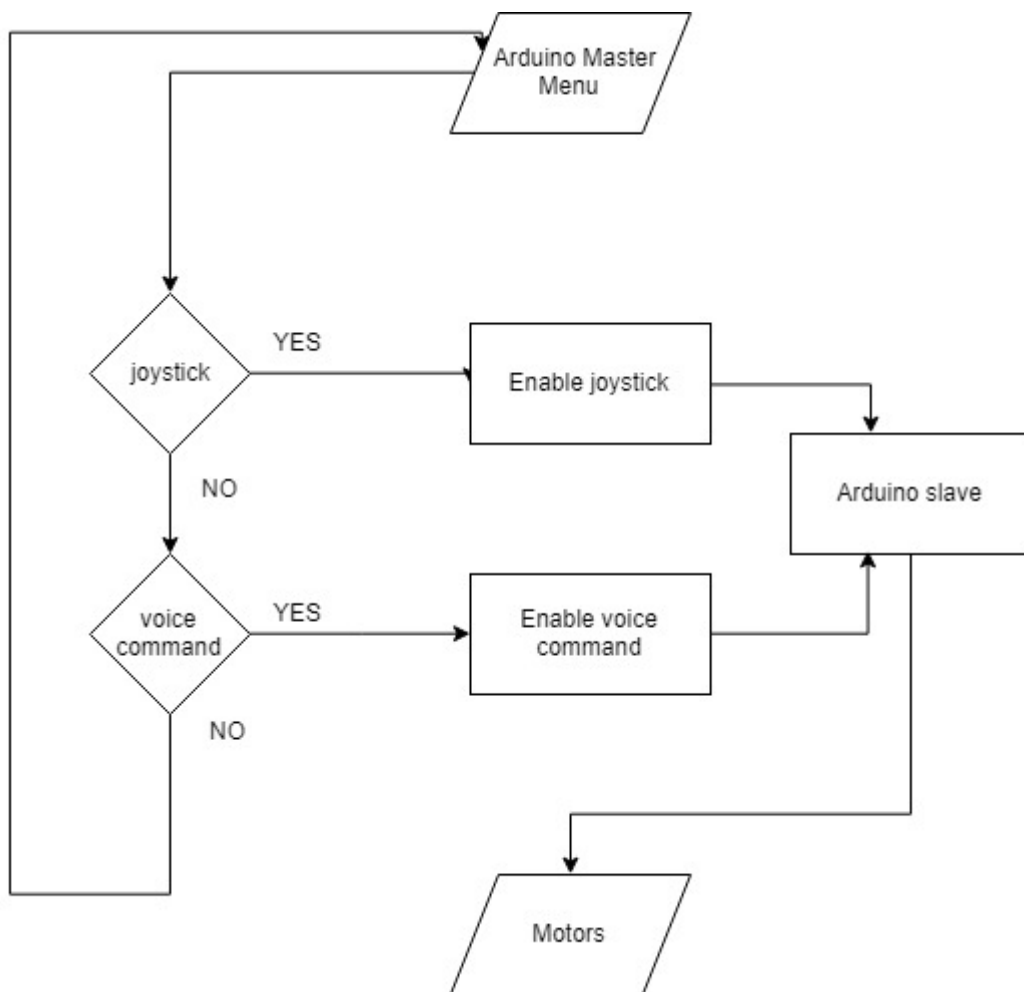
4.5 Η οθόνη LCD 4884

Για λόγους ευχρηστίας, αποφασίστηκε να χρησιμοποιηθεί μια οθόνη LCD, η οποία θα οργανώνει το σύνολο των περιφερειακών εξαρτημάτων και θα λειτουργεί σαν πίνακας έλεγχου του αναπηρικού αμαξιδίου.



Εικόνα 33: Η οθόνη LCD 4884.

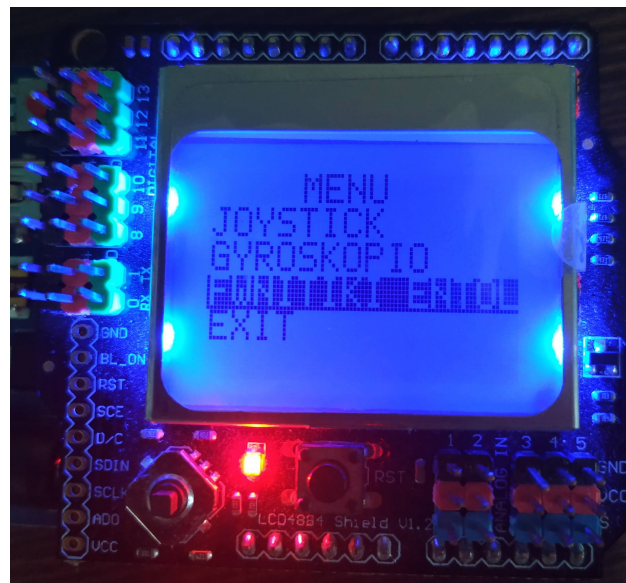
Στην οθόνη θα υπάρχει ένα κύριο μενού, με όλους τους τρόπους χειρισμού του αμαξιδίου, και ο χρήστης, επιλέγοντας έναν από αυτούς μέσω της οθόνης, θα μπορεί να αποφασίσει με ποιο τρόπο θέλει να λειτουργήσει το όχημα. Ακολουθεί ένα διάγραμμα ροής που περιγράφει τη διαδικασία.



Εικόνα 34: Διάγραμμα ροής για το μενού της οθόνης.

Η οθόνη LCD 4884, είναι μια αξιόπιστη οθόνη που κυκλοφόρησε η «NOKIA» με το μοντέλο «5110», με ανάλυση 48 επί 84 σημείων. Στη συγκεκριμένη έκδοση για Arduino, είναι προσαρμοσμένη σε ένα shield το οποίο έχει ενσωματωμένο ένα χειριστήριο (joystick) 5 κατευθύνσεων και ένα μπουτόν για την επαναφορά της (reset). Το Shield διαθέτει έντεκα ακροδέκτες, εκ των οποίων οι έξι είναι ψηφιακοί και οι πέντε αναλογικοί.

Η συγκεκριμένη οθόνη, θα βοηθάει το χρήστη να επιλέγει τον τρόπο διασύνδεσης του, με το αναπηρικό αμαξίδιο. Πιο συγκεκριμένα, στην οθόνη θα προβάλλεται ένα μενού, με τους τρεις διαφορετικούς τρόπους διασύνδεσης (χειριστήριο - γυροσκόπιο - φωνητική εντολή) και ο χρήστης θα επιλέγει έναν από τους τρεις που επιθυμεί να λειτουργήσει. Για συντομία της πτυχιακής εργασίας σε συνεννόηση με τον Κ. Φασουλά, η επιλογή του γυροσκοπίου απορρίφθηκε.



Εικόνα 35: Το μενού, με τις τρεις διαφορετικές επιλογές διασύνδεσης.

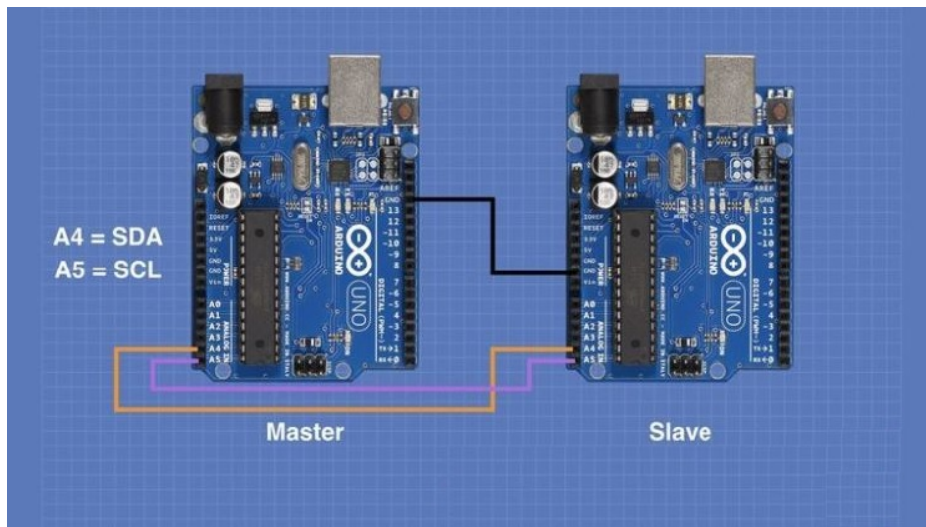
4.6 Προγραμματισμός της οθόνης 4884 και το πρωτόκολλο επικοινωνίας “MASTER - SLAVE”.

Λόγω της παλαιότητας του shield, το οποίο κυκλοφόρησε πριν το 2010, αντιμετωπίστηκαν αρκετά προβλήματα στον προγραμματισμό της οθόνης. Τα δύο πιο σοβαρά προβλήματα από αυτά ήταν: πρώτων, η πλατφόρμα «Arduino IDE» έχει σταματήσει να την υποστηρίζει και να προσφέρει βιβλιοθήκες από την έκδοση «Arduino 1.5» και δεύτερον η έκδοση «1.5» είναι αρκετά παλιά για να υποστηρίξει και να προγραμματίσει τα πιο νέα υποσυστήματα, όπως αυτό της φωνητικής εντολής.

Με αυτά τα δύο καίρια ζητήματα που έπρεπε να επιλυθούν, καταφύγαμε στη λύση της απομόνωσης της οθόνης, με δύο ξεχωριστά «Arduino UNO», με τη μέθοδο «master-slave». Στο πρώτο «Arduino UNO» με το ρόλο «master», θα συνδεθεί το shield της οθόνης, το οποίο θα προγραμματιστεί με την 1.5 έκδοση της «Arduino IDE» και θα επικοινωνεί με το δεύτερο, σε ρόλο «slave», που θα εκτελεί τις υπόλοιπες εντολές, με την πιο πρόσφατη έκδοση της «Arduino IDE».

Η μέθοδος «master - slave», μας βοηθάει να συνδέσουμε δυο ξεχωριστά Arduino. Το Arduino που θα ορίσουμε ως «master» θα είναι το κύριο Arduino όπου θα ελέγχει ο χρήστης και στο οποίο θα είναι συνδεδεμένη η οθόνη. Αφού επιλέξει τον τρόπο που θέλει να χειριστεί το αμαξίδιο μέσω της οθόνης, τότε το «Arduino master» θα δίνει εντολή αυτόματα στο «Arduino slave» να εκτελέσει την προαναφερόμενη εντολή.

Η σύνδεση των δυο Arduino μπορεί να γίνει με δύο ξεχωριστούς τρόπους με ακριβώς ίδιο τρόπο λειτουργίας και αποτελεσμάτων. Ο πρώτος είναι μέσω των επαφών «RX, TX» και «GND» και ο δεύτερος, που είναι και αυτός που επιλέχτηκε, μέσω των επαφών «SDA, SCL» και «GND». Και στις δύο περιπτώσεις, η μια επαφή στέλνει ενώ η άλλη λαμβάνει δεδομένα και η επαφή «GND» είναι η γείωση.



Εικόνα 36: Η Σύνδεση των δυο μικροελεγκτών με τη μέθοδο «master – slave».

Αφού ξεπεράστηκε το πρόβλημα της επικοινωνίας και του προγραμματισμού των δύο μικροελεγκτών, φτάνουμε στο σημείο του προγραμματισμού της οθόνης.

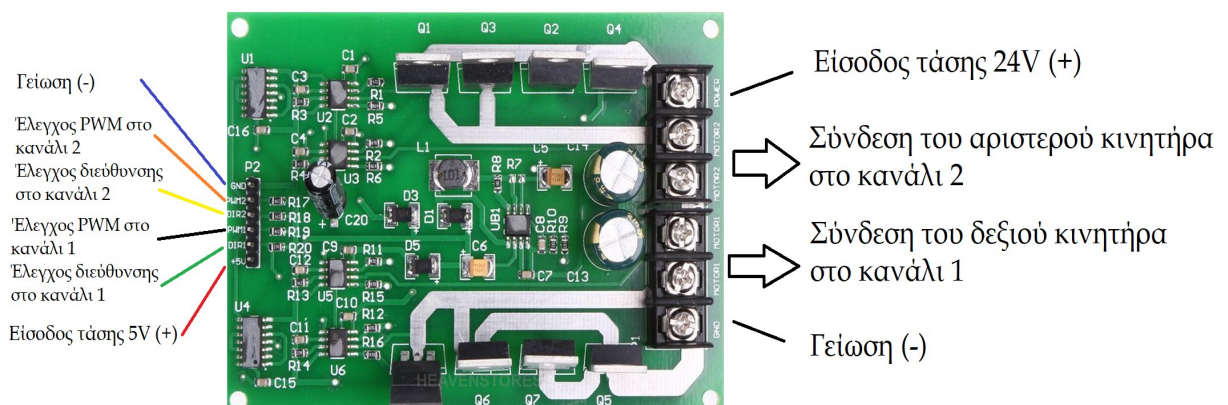
4.7 Η Η-Γέφυρα (BTS7960B)

Μία Η-Γέφυρα μας βοηθάει να εναλλάξουμε την πολικότητα σε ένα ή περισσότερους κινητήρες αυτόματα μέσω του μικροελεγκτή «Arduino», όποτε είναι απαραίτητο. Επίσης ελέγχει την ταχύτητα περιστροφής στον κάθε κινητήρα ανά πάσα στιγμή, επιτυγχάνοντας επιταχύνσεις, επιβραδύνσεις, συνεχόμενη τάση καθώς και σταμάτημα των κινητήρων.

Μία Η-Γέφυρα έχει εισόδους και εξόδους. Στις εισόδους υπάρχουν οι ακροδέκτες της τροφοδοσίας, του ελέγχου της κατεύθυνσης για κάθε κινητήρα και της ταχύτητας περιστροφής μέσω PWM. Στις εξόδους συνδέονται οι ακροδέκτες των κινητήρων καθώς και η εξωτερική τροφοδοσία τους.

ΕΙΣΟΔΟΙ

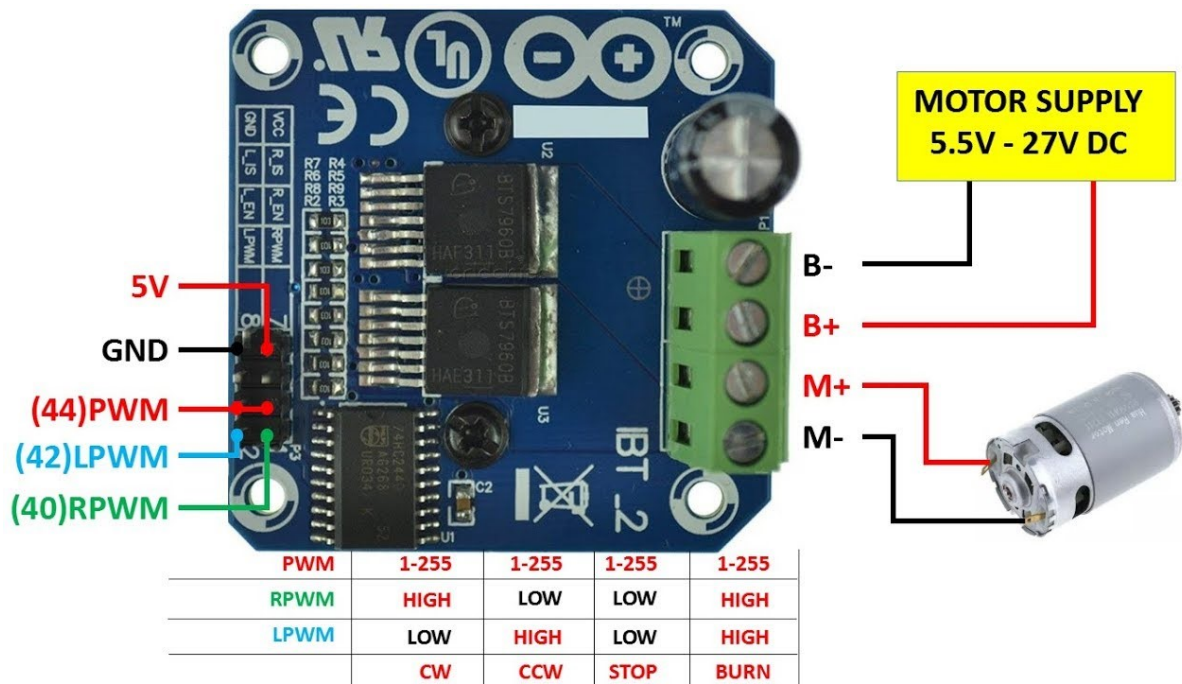
ΕΞΟΔΟΙ



Εικόνα 37: Η Η-Γέφυρα (IRF3205) δύο καναλιών.

Η Συγκεκριμένη Γέφυρα διαθέτει δύο κανάλια και είναι ικανή να ελέγχει μέχρι δύο κινητήρες. Μπορεί να ελέγξει κινητήρες που απαιτούν τάση λειτουργίας 3V μέχρι και 36V. Το μέγιστο ρεύμα λειτουργίας που μπορεί να διαχειριστεί είναι 15A.

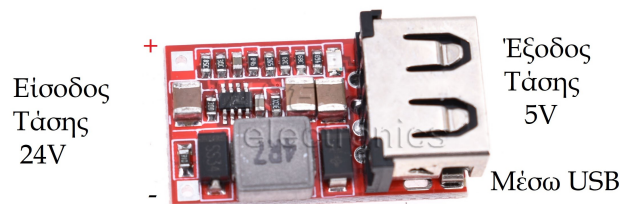
Στην δική μας περίπτωση επιλέχτηκε η Η - Γέφυρα (BTS7960B), η οποία μπορεί να διαχειριστεί υψηλότερη ένταση ρεύματος (43A). Επειδή όμως, μπορεί να τροφοδοτήσει μόνο ένα κινητήρα, θα χρησιμοποιήσουμε δύο.



Εικόνα 38: Η Η-Γέφυρα BTS7960B.

4.8 Ο Μετατροπέας τάσης (4R7 DC-DC)

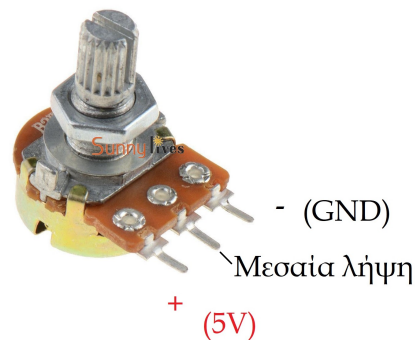
Για την τροφοδοσία των μικροελεγκτών και των υποσυστημάτων που απαιτούν τάση λειτουργίας 5V (Χειριστήριο, ποτενσιόμετρο κ.λ.π), είναι αναγκαία η ύπαρξη ενός μετατροπέα, που θα μετατρέπει την τάση των 24V που παρέχουν οι συσσωρευτές σε 5V. Ο συγκεκριμένος μετατροπέας που χρησιμοποιήθηκε έχει δύο ακροδέκτες (+,-), για την τροφοδοσία του, ενώ η έξοδος της τάσης των 5V εξασφαλίζεται μέσω μίας θύρας USB.



Εικόνα 39: Ο μετατροπέας τάσης (4R7 DC-DC) που χρησιμοποιήθηκε.

4.9 Έλεγχος περιστροφής

Για την εξασφάλιση της ταυτόχρονης περιστροφής των κινητήρων, όταν το όχημα θα κινείται ευθύγραμμα, χρησιμοποιήθηκε ένα ποτενσιόμετρο το οποίο θα μειώνει ή θα αυξάνει την ταχύτητα περιστροφής του κινητήρα, που έχει «χάσει» την προκαθορισμένη ταχύτητα περιστροφής.



Εικόνα 40: Το ποτενσιόμετρο για τον έλεγχο περιστροφής.

Ο τρόπος λειτουργίας είναι απλός. Το ποτενσιόμετρο όταν βρίσκεται στην ακραία θέση, έχει αντίσταση 10 k Ohm, ενώ στην ελάχιστη κοντά στο 0. Όσο λοιπόν ο κέρσορας βρίσκεται στην μεσαία θέση (περίπου 5 k Ohm) οι κινητήρες περιστρέφονται με τον προκαθορισμένο τρόπο.

Μόλις ξεπεράσει τα 5 k Ohm, τότε προοδευτικά ο αριστερός κινητήρας θα αυξάνει την ταχύτητα περιστροφής του, ενώ αντίστοιχα μόλις κατέβει κάτω από τα 5 k Ohm, τότε ο δεξιός κινητήρας θα αυξάνει προοδευτικά την ταχύτητα περιστροφής του.

Κεφάλαιο 5 Κατασκευές

5.1 Κατασκευή θήκης συσσωρευτών

Για τη ασφαλή τοποθέτηση των δύο συσσωρευτών στο όχημα, είναι απαραίτητη η δημιουργία μια βάσης. Αυτή θα μπορούσε να κατασκευαστεί από πλήθος υλικών όπως το ξύλο, πλαστικό και άλλα. Στη συγκεκριμένη περίπτωση θα κατασκευαστεί από πολυεστέρα.



Εικόνα 41: Συσσωρευτές των δώδεκα βολτ (12V).

Ο πολυεστέρας είναι ένα ρευστό υλικό που με την προσθήκη ενός καταλύτη σταθεροποιείται και γίνεται σταθερός και συμπαγής. Ο μόνος τρόπος για να του δώσουμε την μορφή που εμείς θέλουμε, είναι να κατασκευάσουμε ένα καλούπι στο οποίο θα απλώσουμε τον πολυεστέρα και θα πάρουμε την επιθυμητή γεωμετρία.



Εικόνα 42: Χαρτόνι για την κατασκευή του καλούπιού.

Όμως το καλούπι που θα χρησιμοποιήσουμε για την κατασκευή της θήκης, πρέπει να το κατασκευάσουμε και αυτό. Γι' αυτό λοιπόν με ένα χαρτόνι θα περιτυλίξουμε τις μπαταρίες και θα δημιουργήσουμε ένα πρωτότυπο καλούπι στις διαστάσεις που επιθυμούμε.



Εικόνα 43: Προσαρμογή του χαρτονιού στις επιθυμητές διαστάσεις.

Το πρωτότυπο καλούπι είναι έτοιμο και προχωράμε στην κατασκευή του δεύτερου καλουπιού, στο οποίο θα απλωθεί ο πολυεστέρας και θα σχηματίσει τη θήκη των συσσωρευτών. Επειδή όμως ο πολυεστέρας είναι πολύ ισχυρός και ομογενοποιείται σε κάθε επιφάνεια, θα πρέπει μετά την τήξη του να αφαιρέσουμε το καλούπι του, χωρίς αυτό να προσκολληθεί επάνω του.

Γι' αυτό το λόγο, το καλούπι θα είναι κατασκευασμένο από σκληρό αφρολέξ και θα το επενδύσουμε με ταινία περιτυλίγματος, για να γίνει εύκολη η αφαίρεση του, στο πέρας της κατεργασίας.



Εικόνα 44: Σκληρό αφρολέξ επενδυμένο με ταινία περιτολίγματος.

Αφού προετοιμάσουμε το αφρολέξ που θα χρησιμοποιήσουμε, με τη βοήθεια του πρώτου καλουπιού από χαρτόνι, θα κατασκευάσουμε το δεύτερο καλούπι στο οποίο θα δημιουργήσουμε τα τοιχώματα της θήκης.



Εικόνα 45: Κατασκευή του δεύτερου καλουπιού.

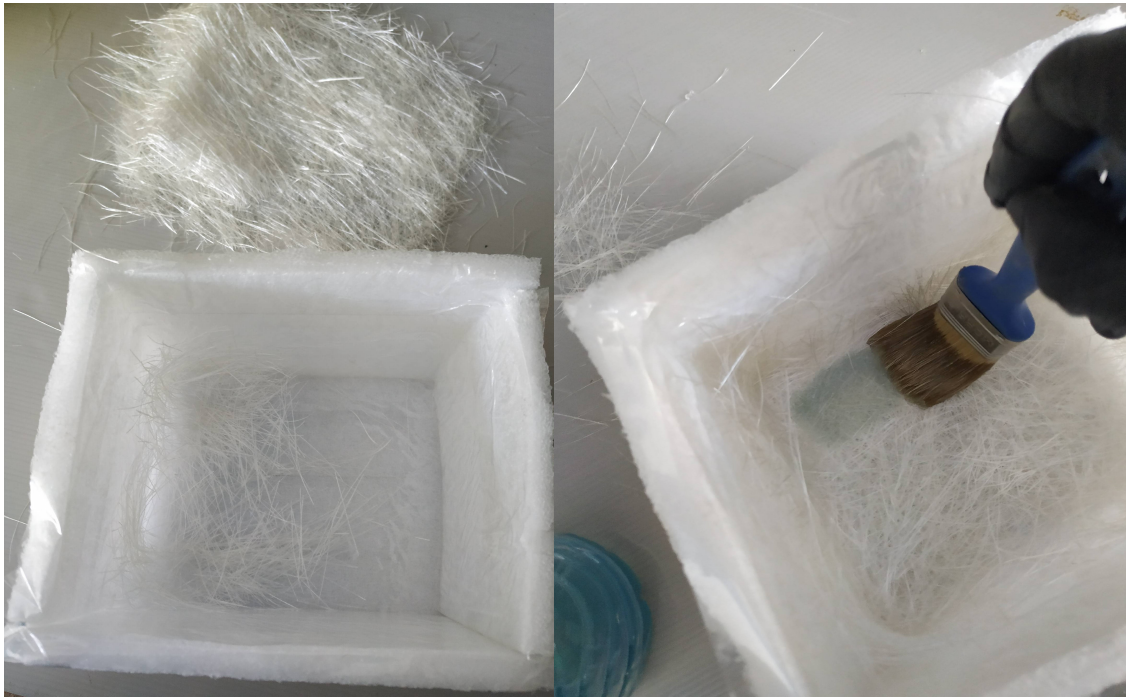
Η συναρμολόγηση των τεμαχίων του αφρολέξ έγινε με νοβοπανόβιδες και χρησιμοποιήθηκε και χαρτοταινία, για τον περιορισμό του υγρού μέσα στο καλούπι.

Σε αυτό το σημείο απλώνουμε μια στρώση από ίνες πολυεστέρα και ετοιμαζόμαστε να απλώσουμε με τη βοήθεια ενός πινέλου τον πολυεστέρα. Η διαδικασία αυτή επαναλαμβάνεται, έως ότου δημιουργηθεί το κατάλληλο πάχος που θέλουμε να αποκτήσει το τοίχωμα της θήκης, το οποίο είναι το πάχος του χαρτονιού που αποτέλεσε το πρώτο καλούπι.

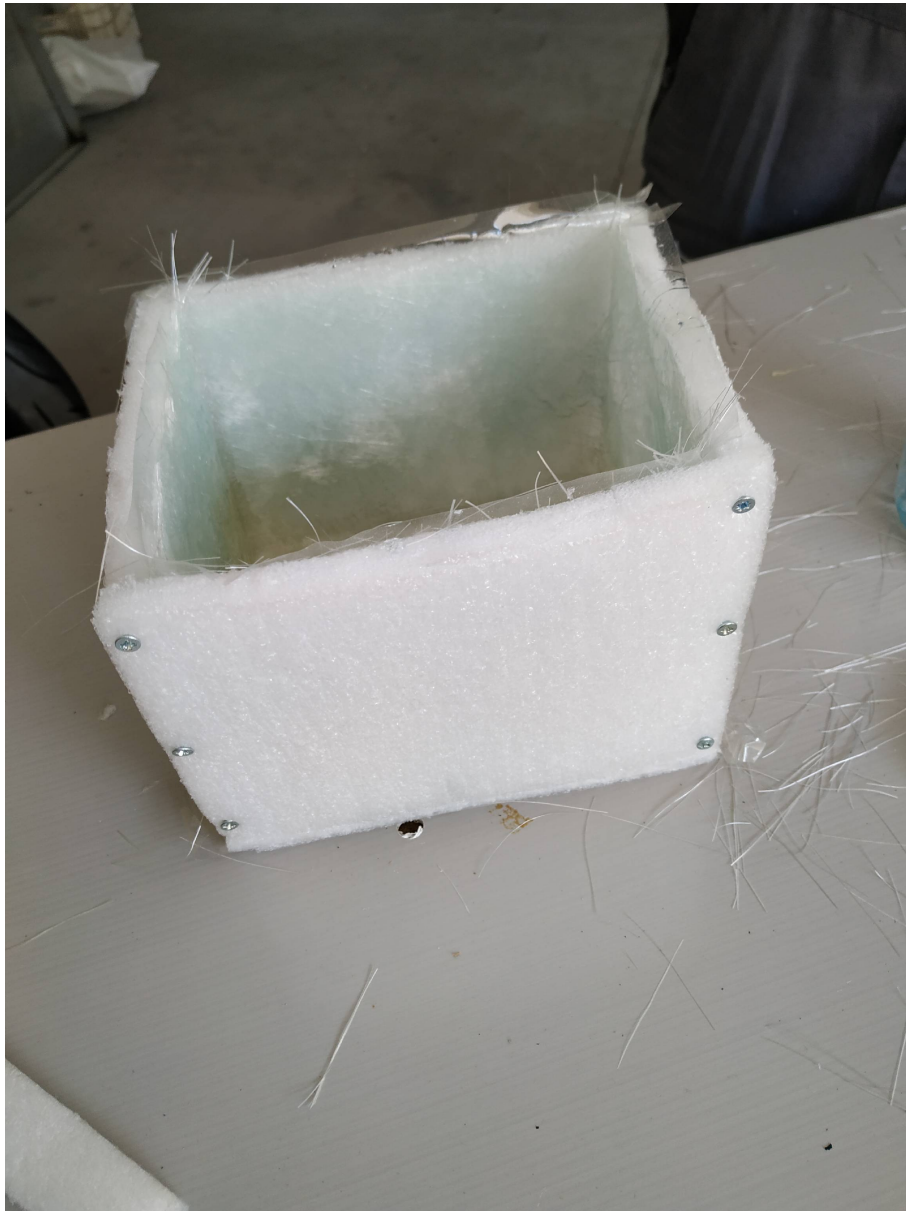


Εικόνα 46: Αριστερά δοχείο με ποσότητα πολυεστέρα και δεξιά καταλύτης – σκληρυντής για πολυεστέρα.

Για την γρήγορη πήξη του πολυεστέρα, χρησιμοποιούμε καταλύτη, με αναλογία 15 γραμμάρια ανά κιλό πολυεστέρα και τα αναμειγνύουμε μέχρι να ομογενοποιηθούν.



Εικόνα 47: Άπλωμα πολυεστέρα στο πρώτο στρώσιμο ινών.



Εικόνα 48: Στιγμή από τη καταργασία στην τελική γεωμετρία.

Μόλις ολοκληρωθεί η διαδικασία του απλώματος του πολυεστέρα, αφήνουμε το καλούπι σε καλά αεριζόμενο χώρο, για εικοσιτέσσερις ώρες ώστε να πήξει. Κατόπιν αφαιρούμε το καλούπι από τη θήκη.



Εικόνα 49: Αφαίρεση του καλουπιού από την θήκη.

Μετά που θα βγάλουμε τη θήκη από το καλούπι, πρέπει να κοπούν τα υπολείμματα τις κατεργασίας, να τριφτεί και έπειτα να βαφτεί.



Εικόνα 50: Σημάδεμα για το κόψιμο της θήκης στο επιθυμητό ύψος.



Εικόνα 51: Η τελική μορφή της πολυεστερικής θήκης.

Αφού ολοκληρώσαμε τη κατασκευή της θήκης, θα πρέπει να δημιουργήσουμε με τον ίδιο τρόπο και ένα κάλυμμα. Στο κάλυμμα που θα κατασκευάσουμε θα πρέπει να του δώσουμε αρκετό ύψος για να συμπεριλάβουμε επάνω του την πλακέτα της γέφυρας που θα συνδεθούν οι κινητήρες καθώς επίσης και ο μετασχηματιστής που θα τροφοδοτήσει τους δύο μικροελεγκτές.

Θα χρησιμοποιήσουμε και σε αυτή τη περίπτωση αφρολέξ, για να κατασκευάσουμε ένα παραλληλόγραμμο μικρότερο από το εμβαδόν που καταλαμβάνει η θήκη, προκειμένου να δημιουργήσουμε μία «πατούρα» στην οποία θα «θυληκώνει» το κάλυμμα επάνω στη θήκη και ύψος αρκετό για να προσαρμοστούν τα εξαρτήματα που θα περιλαμβάνει το κάλυμμα.



Εικόνα 52: Κατασκευή καλουπιού για το κάλυμμα της θήκης.

Σε αυτή τη περίπτωση, ο πολυεστέρας θα απλωθεί ακριβώς επάνω από τη θήκη, για να πάρει την καλύτερη δυνατή μορφή. Για να αφαιρεθεί το κάλυμμα μετά την κατεργασία, απλά τυλίγουμε τη θήκη με νάιλον, το οποίο θα αφαιρεθεί εξαιρετικά εύκολα στο πέρας της κατεργασίας.

Με τα ίδια υλικά και τον τρόπο απλώματος, στρώση-στρώση, του πολυεστέρα, που εξηγήσαμε προηγουμένως, συνεχίζουμε και εδώ στην κατασκευή του καλύμματος.



Εικόνα 53: Στιγμή από την κατεργασία για το κάλυμμα.

Αφού ολοκληρώσαμε το άπλωμα του πολυεστέρα και το αφήσαμε για ένα εικοσιτετράωρο να στερεοποιηθεί και να πήξει, το αφαιρούμε από τη θήκη και εν συνεχεία, θα κόψουμε τα υπολείμματα της κατεργασίας, θα το τρίψουμε με γυαλόχαρτο και θα το βάψουμε.



Εικόνα 54: Το κάλυμμα έπειτα από το πέρας της κατεργασίας και πριν τη βαφή του.



Εικόνα 55: Η τελική μορφή της θήκης των συσσωρευτών με το κάλυμμα της έπειτα από βαφή με μαύρο ματ σπρέι.

6.3 Κατασκευή βάσεων κινητήρων

Μία από τις κυριότερες κατεργασίες, της παρούσας πτυχιακής εργασίας, είναι η κατασκευή και τοποθέτηση στο αναπηρικό αμαξίδιο, μίας πλήρως προσαφαιρούμενης βάσης στήριξης, για κάθε ένα κινητήρα ξεχωριστά. Επιπροσθέτως, θα πρέπει να κατασκευαστεί ένας μηχανισμός που θα αποσυμπλέκει τους κινητήρες καθώς δεν θα πρέπει να βρίσκονται μόνιμα σε σύμπλεξη με τους τροχούς του αμαξιδίου.

Τα κύρια υλικά που χρησιμοποιήθηκαν, για την κατασκευή της κάθε βάσης, είναι ράγες στήριξης από χάλυβα.



Εικόνα 56: Ράγες στήριξης.

Για την προσαρμογή των ραγών στήριξης στις πλαϊνές πλευρές του αμαξιδίου, θα πρέπει να κατασκευαστούν κατάλληλες εσοχές επάνω στις ράγες οι οποίες θα βοηθήσουν στο να «κουμπώσει» η κάθε ράγα πάνω στις δοκούς του αμαξιδίου. Με αυτόν τον τρόπο επιτυγχάνεται η σταθεροποίηση στον άξονα Χ και με τη βοήθεια δεματικών στερεωτικών καλωδίων (tie wraps), εξασφαλίζεται η σταθεροποίηση και στους υπόλοιπους άξονες, Ζ και Ψ.

Η διάνοιξη των εσοχών έγινε με τη βοήθεια γωνιακού τροχού, αφού ασφαλίστηκε η ράγα επάνω σε μία μέγγενη.



Εικόνα 57: Εσοχή για την προσαρμογή στη δοκό του αμαξιδίου.

Αφού επαναλάβουμε τη διαδικασία διάνοιξης των τριών εσοχών που απαιτούνται σε κάθε ράγα στήριξης, μπορούμε να τη προσαρμόσουμε επάνω στο αμαξίδιο και με τη βοήθεια των δεματικών να την ασφαλίσουμε στη θέση της.



Εικόνα 58: Η ράγα στήριξης τοποθετημένη στο αμαξίδιο.

Για την κατασκευή του μηχανισμού κίνησης των κινητήρων, θα χρησιμοποιηθεί ακόμα μία ράγα στήριξης μεγαλύτερου μεγέθους, η οποία θα κινείται «συρταρωτά» επάνω από τη κύρια ράγα στήριξης.

Για τη σταθεροποίηση του κινητήρα, θα χρησιμοποιηθεί ένας κοχλίας και ένα περικόχλιο σε σχήμα «πεταλούδας» για να μπορεί ο χρήστης να τον συμπλέκει και να τον αποσυμπλέκει με το χέρι ανά πάσα στιγμή.

Σε αυτό το σημείο θα χρειαστεί ακόμα μια κατεργασία, καθώς οι δύο ράγες ταιριάζουν για την λειτουργία που θέλουμε να τους αποδώσουμε, μόνο σε δύο από τις τρεις διαστάσεις τους.

Εμείς θέλουμε να «συρταρώνουν» σε κάθε πλευρά τους και επειδή στην ράγα που θα χρησιμοποιηθεί για την κίνηση του κινητήρα, το ύψος είναι μεγαλύτερο από το επιθυμητό, θα κόψουμε τα υπάρχοντα «αυτιά» της και διπλώνοντας το υπόλοιπο υλικό κατά ενενήντα μοίρες θα σχηματίσουμε τα καινούρια αυτιά, στις κατάλληλες διαστάσεις, για την εφαρμογή μας.



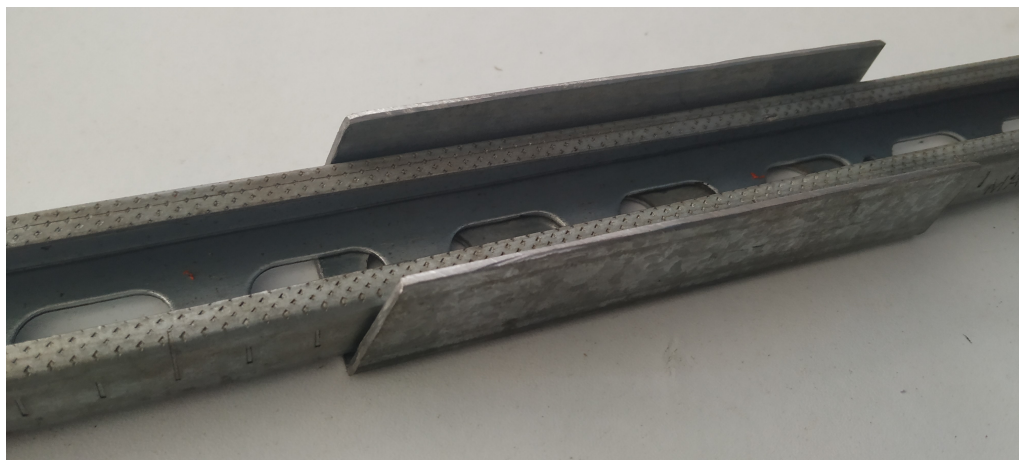
Εικόνα 59: Το κενό που δημιουργούν οι δύο ράγες λόγω της διαφοράς ύψους.

Για την κατασκευή των νέων αυτιών στη ράγα κινήσεως, θα την ασφαλίσουμε πρώτα σε μία μέγγενη και με τη βοήθεια ενός γωνιακού τροχού θα κόψουμε σε όλο το μήκος τους, τα αρχικά «αυτιά».



Εικόνα 60: Δέσιμο της κινούμενης ράγας στην μέγγενη και κοπή των «αυτιών» με γωνιακό τροχό.

Αφού κόψαμε τα αρχικά «αυτιά», θα ασφαλίσουμε ξανά την ράγα στη μέγγενη και με τη βοήθεια μιας βαριοπούλας θα σχηματίσουμε τα καινούρια «αυτιά», στις επιθυμητές διαστάσεις.



Εικόνα 61: Η ράγα στήριξης μέσα στην κινούμενη ράγα πριν τη δημιουργία νέων «αυτιών»



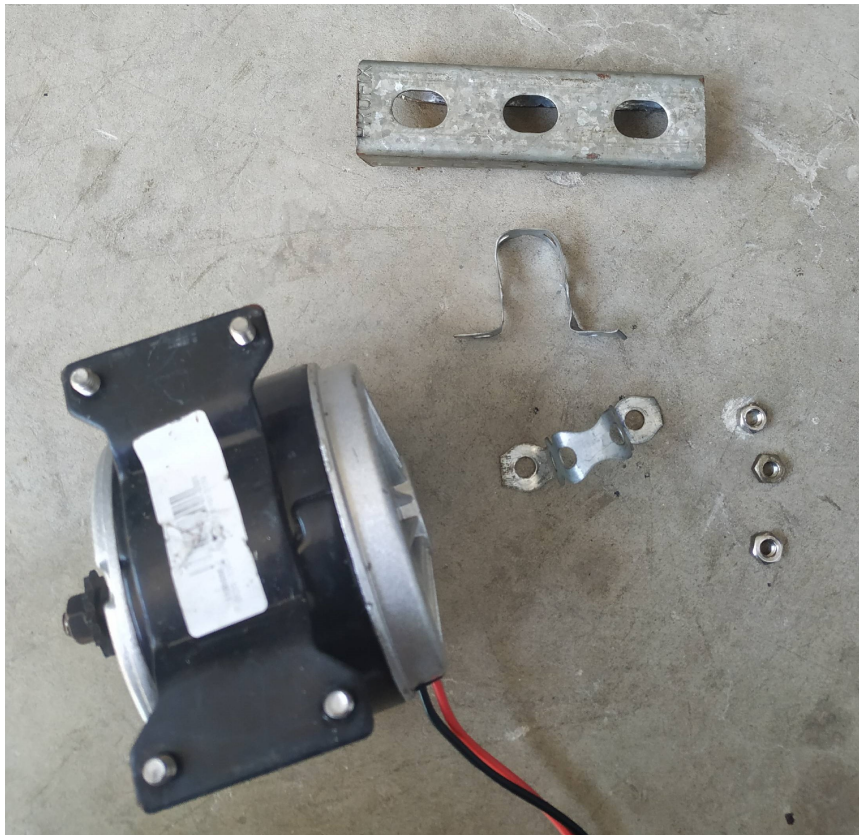
Εικόνα 62: Σημάδεμα και κατασκευή νέων «αυτιών».

Αφού ολοκληρώσαμε την κατασκευή των νέων αυτιών και εξασφαλίσαμε το συρτάρισμα που επιθυμούσαμε μένει μόνο να προσαρμοστεί ο κινητήρας επάνω στη κινούμενη ράγα.



Εικόνα 63: Το «συρτάρισμα» των δύο ραγών.

Καθώς ο κινητήρας των 250W που προμηθευτήκαμε, διαθέτει εξαρχής τη δική του βάση, στην οποία υπάρχουν και τέσσερις οπές με σπείρωμα είναι αρκετά εύκολο προσθέτοντας δύο κομμάτια τσέρκι, να στερεωθεί με τέσσερις κοχλίες M5 και τέσσερα περικόχλια M5 επάνω στη ράγα κινήσεως.



Εικόνα 64: Τα εξαρτήματα που θα στερεώσουν τον κινητήρα στη ράγα στήριξης.

Πριν τη συναρμολόγηση επάνω στη ράγα κινήσεως προηγείται η διαδικασία της βαφής των εξαρτημάτων.



Εικόνα 65: Τα εξαρτήματα για την στερέωση του κινητήρα έπειτα από τη βαφή.



Εικόνα 66: Το στερέωμα του κινητήρα επάνω στη ράγα κινήσεως.



Εικόνα 67: Τοποθέτηση του κινητήρα επάνω στο αναπηρικό αμαξίδιο.



Εικόνα 68: Κοχλίας και περικόχλιο σε σχήμα «πεταλούδας» για τη σταθεροποίηση του κινητήρα».

5.3 Προσαρμογή της τροχαλίας στον κινητήρα



Εικόνα 69: Ηλεκτρικός κινητήρας

Για την προσαρμογή της τροχαλίας επάνω στον άξονα του κινητήρα, πρέπει να γίνουν ορισμένες ενέργειες. Αρχικά, ο άξονας θα πρέπει να επεκταθεί, για να μπορέσει η τροχαλία να βρεθεί στο σημείο επαφής με τον τροχό, ώστε να υπάρξει η μετάδοση της κίνησης.

Όμως ο κατασκευαστής του κινητήρα έχει δημιουργήσει ένα αριστερό σπείρωμα, καθώς ο κινητήρας αυτός κατασκευάστηκε με προορισμό να κινείται μόνο δεξιόστροφα και το αριστερό σπείρωμα αναγκάζει το γρανάζι που θα προσαρμόσει πάνω στον άξονα να μην χαλαρώσει.

Καθώς στην τοπική αγορά δεν διατίθεται κάποιος σύνδεσμος άξονα που να διαθέτει αριστερό σπείρωμα, αγοράστηκε από το εξωτερικό ένας μετατροπέας, από αριστερό σε δεξιό σπείρωμα και ταυτόχρονα από αρσενικό σε θηλυκό.



Εικόνα 70: Αριστερά διακρίνεται ο μετατροπέας από αριστερό σε δεξιό σπείρωμα και στα δεξιά μαζί με το περικόχλιο που θα ασφαλίσει την τροχαλία στην θέση της.

Έφτασε η στιγμή της τοποθέτησης της τροχαλίας επάνω στον άξονα του κινητήρα. Τοποθετούμε και σφίγγουμε με ένα γερμανικό κλειδί τον μετατροπέα, επάνω στον άξονα του κινητήρα, αριστερόστροφα. Έπειτα τοποθετούμε την τροχαλία και τέλος σφίγγουμε το εξωτερικό περικόχλιο, με ένα πολύγωνο δεξιόστροφα.

Τέλος, προσθέτουμε στην εξωτερική επιφάνεια της τροχαλίας μία στρώση αντιολισθητικής ταινίας, για την εξασφάλιση μεγαλύτερης τριβής και αποφυγής κάποιου γλιστρήματος, κατά την μετάδοση της κίνησης



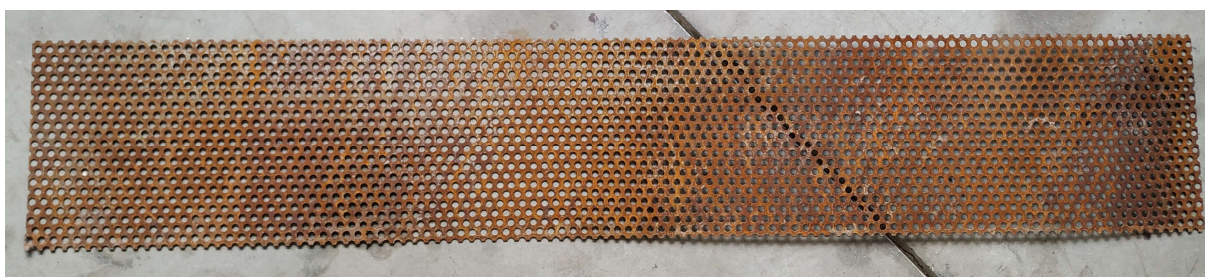
Εικόνα 71: Τοποθέτηση της τροχαλίας επάνω στον άξονα του κινητήρα.

5.4 Κατασκευή βάσεως συσσωρευτών

Για την τοποθέτηση των μπαταριών, ενός μετασχηματιστή και της «Η-γέφυρας», είναι αναγκαία η κατασκευή μίας βάσης στην οποία θα τοποθετηθούν όλα τα προαναφερόμενα επάνω σε αυτήν.

Η βάση αυτή θα πρέπει να προσαρμόζεται επάνω στο αναπηρικό αμαξίδιο, όπως όλα τα πρόσθετα τμήματα της πτυχιακής εργασίας, δηλαδή χωρίς καμία μετατροπή του αμαξιδίου.

Επιπροσθέτως, θα πρέπει να είναι ικανή να αντέχει ένα αρκετά μεγάλο φορτίο, το οποίο φέρουν οι συσσωρευτές και να μην καταλαμβάνει μεγάλο όγκο, αφενός διότι στο σημείο που πρόκειται να τοποθετηθεί ο χώρος είναι περιορισμένος και αφετέρου για αισθητικούς λόγους. Κατά συνέπεια αποφασίστηκε να κατασκευαστεί από μέταλλο και συγκεκριμένα από διάτρητη λαμαρίνα τριών χιλιοστών.



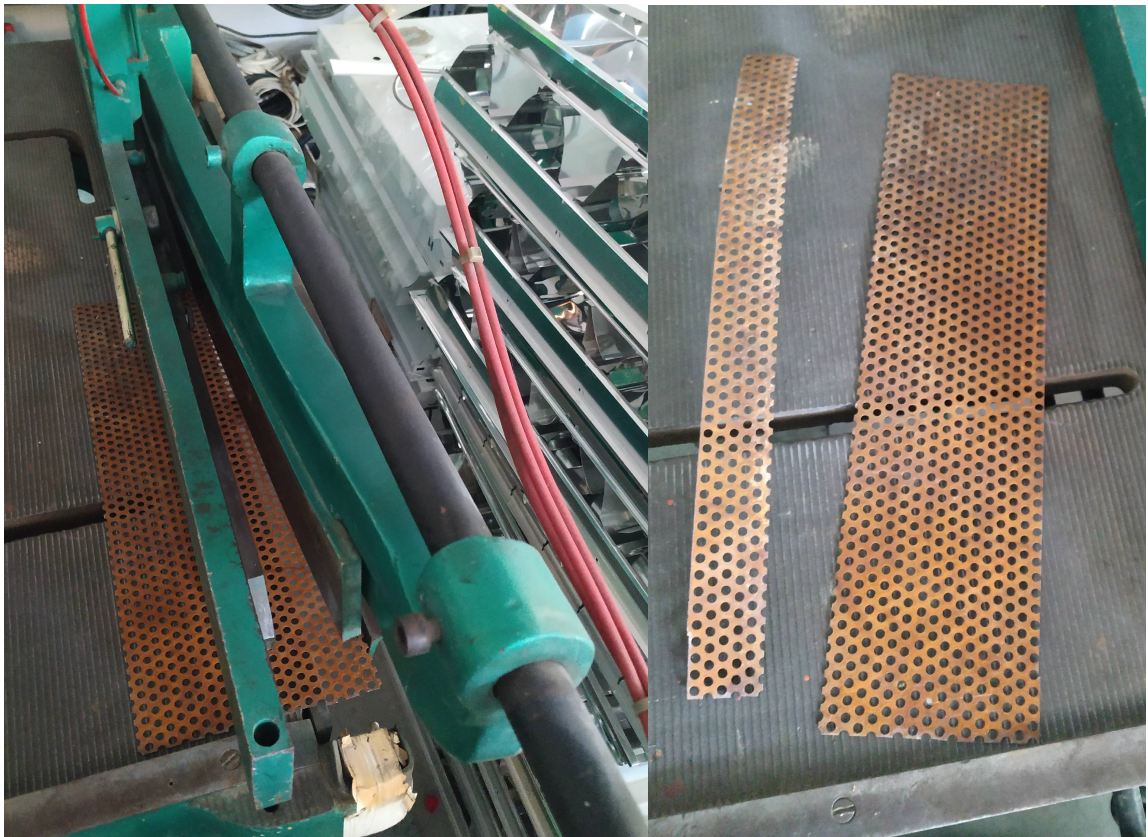
Εικόνα 72: Διάτρητη λαμαρίνα τριών χιλιοστών.

Για την προσαρμογή της λαμαρίνας στις επιθυμητές διαστάσεις, είναι αναγκαία η κοπή της. Υπάρχουν πολλοί τρόποι για να γίνει εφικτή η κοπή. Οι πιο κατάλληλοι τρόποι είναι με ειδικά ψαλίδια τα οποία έχουν σχεδιαστεί και κατασκευαστεί για αυτήν ακριβώς τη χρήση και είναι ευρέως γνωστά ως «λαμαρινοψάλιδα». Φυσικά υπάρχουν σε διάφορα μεγέθη, με τα μεγαλύτερα να είναι αρκετά ογκώδη που χρησιμοποιούνται σε βιομηχανίες και από πολύ μικρά τα επονομαζόμενα λαμαρινοψάλιδα χειρός. Στη συγκεκριμένη περίπτωση επειδή υπήρχε διαθέσιμο επαγγελματικό λαμαρινοψάλιδο η κατεργασία έγινε εκεί για λόγους συντομίας.



Εικόνα 73: Βιομηχανικό λαμαρινοψάλιδο πεπιεσμένου αέρα.

Εφόσον τοποθετήθηκε το τεμάχιο που θέλουμε να κατεργαστούμε στο ψάλιδο, ασφαλίζουμε το φύλλο της λαμαρίνας με τη πλάκα πρέσεως, η οποία μετακινείται στην κατεύθυνση του άξονα Ψ, με τη βοήθεια πεπιεσμένου αέρα, και στη συνέχεια υποβάλλοντας μυϊκή δύναμη στον μοχλό, το ψάλιδο ασκεί πίεση στην λαμαρίνα, την οποία και κόβει.



Εικόνα 74: Διαδικασία για την κοπή της διάτρητης λαμαρίνας.

Αφού λοιπόν η λαμαρίνα κόπηκε στις επιθυμητές διαστάσεις, συνεχίζουμε την κατεργασία της με σκοπό την ενίσχυση της. Στις δύο πλευρές που θα ακουμπάνε πάνω στο πλαίσιο του αμαξιδίου θα δημιουργηθούν δύο «σπασίματα» για να «φωλιάσει» επάνω στο αμαξιδιο και με αυτόν τον τρόπο θα σταθεροποιηθεί σε μεγάλο βαθμό.



Εικόνα 75: Η θέση στην οποία θα τοποθετηθεί η βάση των συσσωρευτών.

Αφού την τοποθετήσουμε επάνω στο αμαξιδίο στη θέση που θέλουμε να τη σταθεροποιήσουμε, προχωράμε στην κατασκευή των δύο «σπασιμάτων». Με τη βοήθεια της γωνιάς σημαδεύουμε τα σημεία τα οποία θα σχηματίσουν την ορθή γωνία που θα αποτελεί το «σπάσιμο» στο φύλλο της λαμαρίνας.



Εικόνα 76: Στιγμή από τη διαδικασία του «γωνιάσματος» της βάσης.

Αφού σημαδέψαμε, με τη βοήθεια της στράντζας θα δημιουργήσουμε το επιθυμητό «σπάσιμο». Τοποθετούμε λοιπόν το φύλλο της λαμαρίνας στη στράντζα και αφού το ασφαλίσουμε μετακινώντας το άνω χείλος στο σημείο που θα «δαγκώσει» την λαμαρίνα, μέσω ενός ποδομοχλού, σηκώνουμε τον ειδικό μοχλό, που περιστρέφει το κάτω χείλος, κατά ενενήντα μοίρες (90°), και έτσι έχουμε δημιουργήσει το αποτέλεσμα που θέλαμε.



Εικόνα 77: Στιγμή από το «στραντζάρισμα» της διάτρητης λαμαρίνας για την κατασκευή ορθής γωνίας.



Εικόνα 78: Προσαρμογή ακριβώς στις διαστάσεις των κυκλικών δοκών του αμαξιδίου.

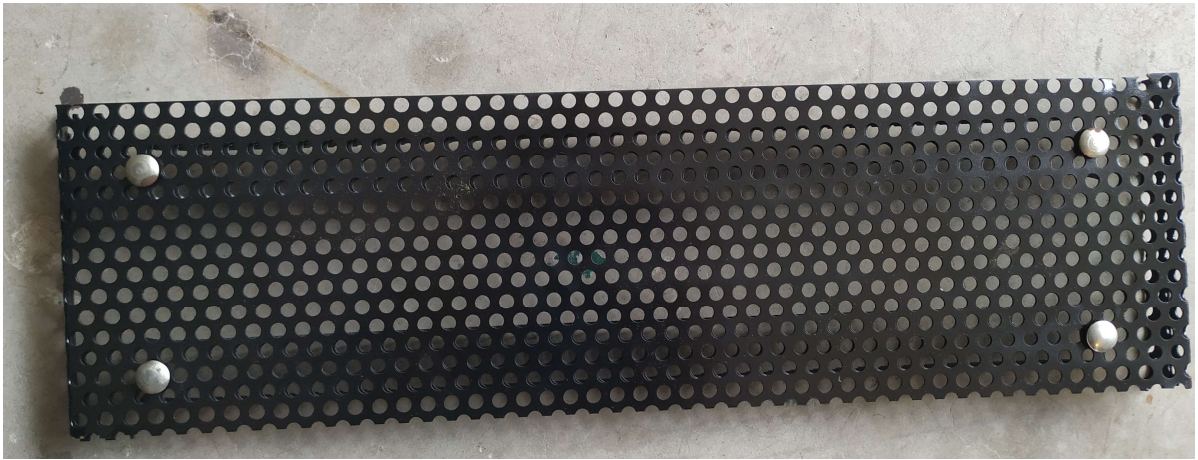
Με τη δημιουργία των δύο πλαϊνών σπασιμάτων εξασφαλίστηκε η σταθερότητα στον άξονα Χ. Μένει ακόμα να δημιουργηθούν δύο «νεύρα», τα οποία θα μας εξασφαλίσουν την σταθερότητα και στον άξονα των Υ. Για τα δύο αυτά «νεύρα» χρησιμοποιήθηκε το ίδιο ακριβώς υλικό και διαμορφώθηκαν σε σχήμα Π, χρησιμοποιώντας τα ίδια εργαλεία και τις ίδιες τεχνικές.



Εικόνα 79: Κοπή διάτρητης λαμαρίνας για την κατασκευή «νεύρου» σε λαμαρινοψάλιδο.

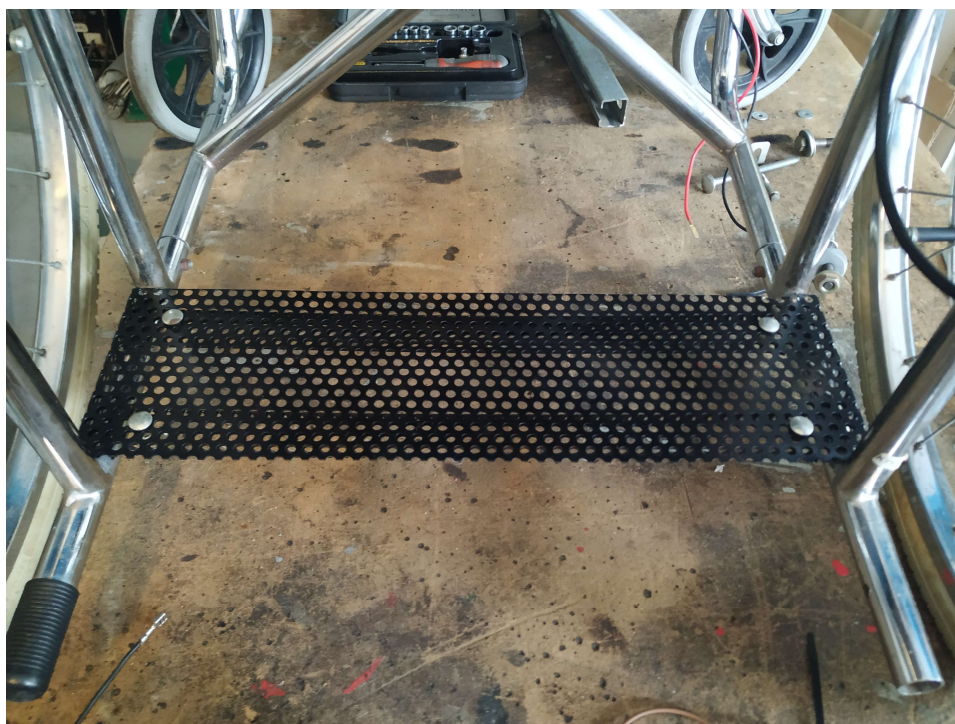


Εικόνα 80: Η βάση μαζί με τα δύο «νεύρα» μετά από βαφή και αφαίρεσης σκουριάς με γυαλόχαρτο.



Εικόνα 81: Η βάση μετά τη συναρμολόγηση.

Αφού κατασκευάστηκαν τα τρία τεμάχια, αφαιρέθηκε η σκουριά και βάφτηκαν με σπρέι, με τη βοήθεια τεσσάρων κοχλιών και τεσσάρων περικοχλίων συναρμολογήθηκαν και η βάση είναι έτοιμη για τοποθέτηση επάνω στο αναπηρικό αμαξίδιο.



Εικόνα 82: Το τελικό αποτέλεσμα της βάσης των συσσωρευτών τοποθετημένη στο αμαξίδιο.

5.5 Κατασκευή κονσόλας χειρισμού

Η κονσόλα χειρισμού πρέπει να περιλαμβάνει το χειριστήριο, ένα γενικό διακόπτη, το μικρόφωνο για την φωνητική εντολή, καθώς επίσης και την οθόνη επιλογών που είναι προσαρμοσμένη στο «ARDUINO-MASTER». Η κονσόλα θα τοποθετηθεί στο δεξί μπράτσο του αναπηρικού αμαξιδίου και όπως και οι υπόλοιπες κατασκευές, έτσι και αυτή θα προσαρμοστεί επάνω στο αμαξίδιο, χωρίς καμία τροποποίηση του. Το υλικό κατασκευής της θα είναι λαμαρίνα, πάχους δύο χιλιοστών.



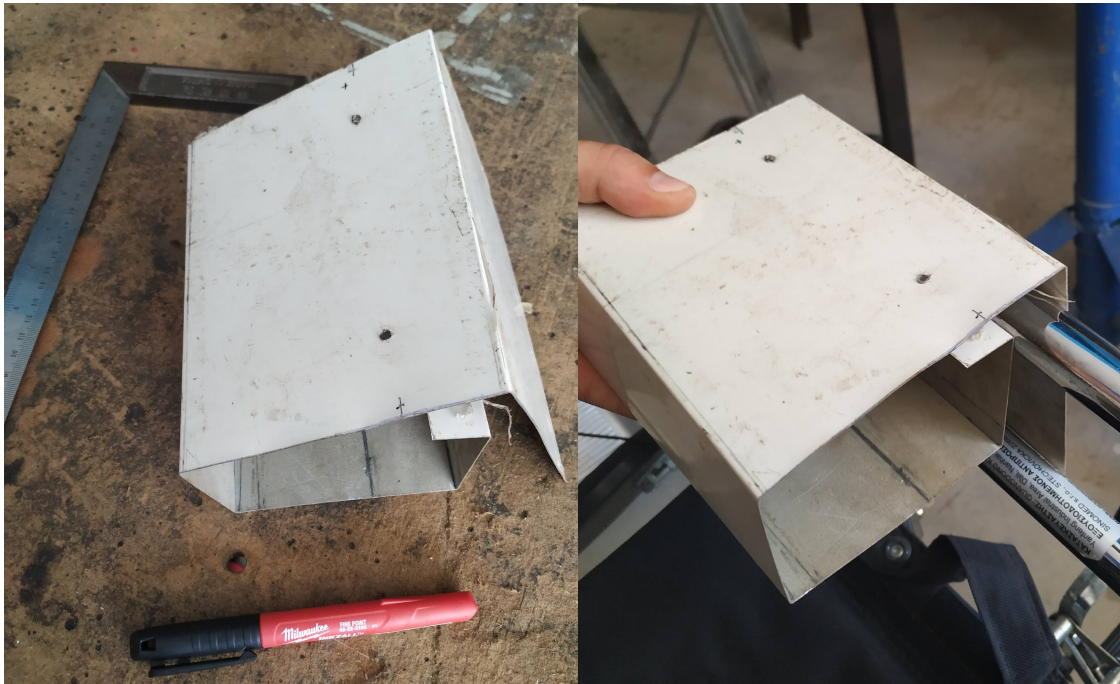
Εικόνα 83: Φύλλο λαμαρίνας, πάχους δύο χιλιοστών.

Οι διαστάσεις της κονσόλας θα είναι 160 χιλιοστά μήκος, 120 χιλιοστά πλάτος και 70 εκατοστά ύψος, κατάλληλες για να προσαρμοστούν όλα τα απαραίτητα στοιχεία πάνω της. Αφού σημαδέψουμε στο αρχικό φύλλο και κόψουμε την επιφάνεια που θα κατεργαστούμε, στο λαμαρινοψάλιδο, όπως ακριβώς περιγράφεται στην ενότητα «ΚΑΤΑΣΚΕΥΗ ΒΑΣΗΣ ΣΥΣΣΩΡΕΥΤΩΝ».



Εικόνα 84: Επιφάνεια για την κατασκευή της κονσόλας.

Με τη βοήθεια της στράντζας θα κατασκευάσουμε το «κουτί» καθώς επίσης και μία «θηλυκή πατούρα», που θα μας βοηθήσει στην στερέωση της κονσόλας επάνω στο μπράτσο του αμαξιδίου. Έπειτα το δέσιμο του κουτιού θα γίνει με δύο πριτσίνια που θα ασφαλίζουν το επάνω μέρος του κουτιού με το υπόλοιπο «σώμα».



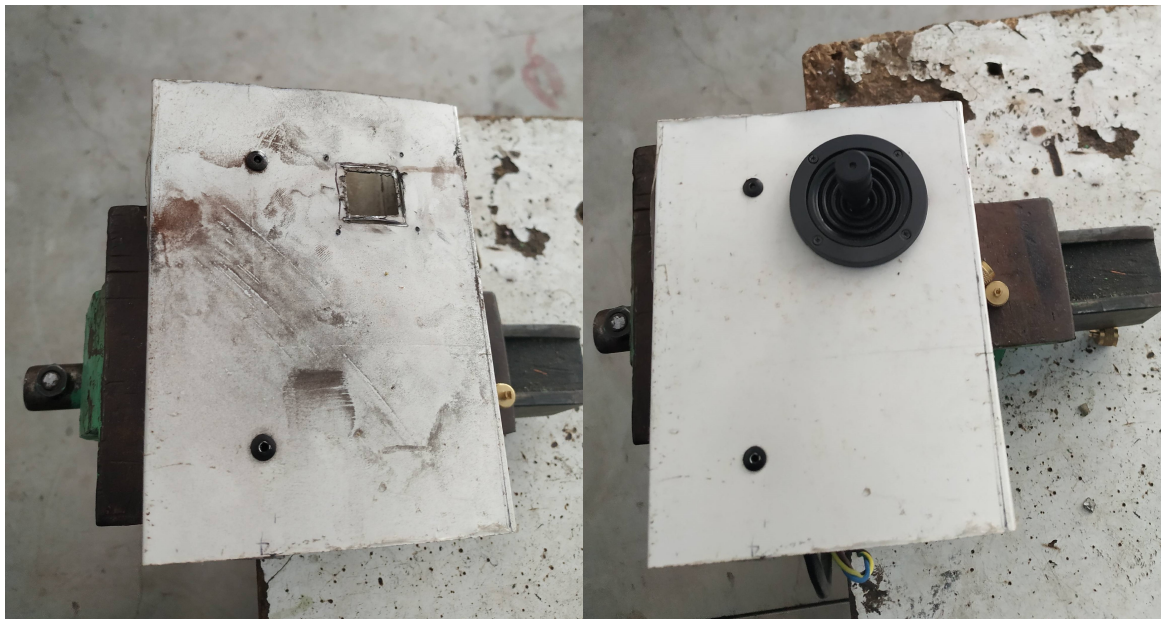
Εικόνα 85: Η κονσόλα μετά το στραντζάρισμα και το άνοιγμα οπών για τα πριτσίνια και το «πιάσιμο» της επάνω στο μπράτσο.

Καθώς η κονσόλα πήρε την μορφή της, ήρθε η ώρα για την προσαρμογή των απαραίτητων στοιχείων που θα περιλαμβάνει. Πρώτα απ' όλα θα ανοιχτούν οπές για την τοποθέτηση του χειριστηρίου.



Εικόνα 86: Το χειριστήριο που θα προσαρμοστεί επάνω στην κονσόλα.

Αφού στερεώσουμε την κονσόλα επάνω στη μέγγενη, με τη βοήθεια ενός πολυεργαλείου για μικρού εύρους εργασίες θα ανοίξουμε τις οπές στα σημεία που έχουμε σημαδέψει για την τοποθέτηση του χειριστηρίου.



Εικόνα 87: Διάνοξη οπών για την τοποθέτηση του χειριστηρίου.

Ακολουθεί η τοποθέτηση της θόνης, η οποία θα προσαρμοστεί σε μία γωνιακή λάμα των πέντε χιλιοστών, και με τη σειρά της θα στερεωθεί επάνω στην κονσόλα με ένα κοχλία και ένα περικόχλιο. Στη συνέχεια θα ανοιχτεί ακόμα μία εσοχή στην οποία θα προσαρμοστεί ο γενικός διακόπτης.



Εικόνα 88: Ο γενικός διακόπτης του οχήματος.



Εικόνα 89: Τοποθέτηση της οθόνης επιλογών και του χειριστηρίου επάνω στην κονσόλα.

Αφού ολοκληρώθηκαν όλες οι κατεργασίες για την κονσόλα, έφτασε η στιγμή για το βάψιμο της. Θα αφαιρέσουμε όλα όσα προσθέσαμε επάνω της για να μείνει μόνο η λαμαρίνα και να την προετοιμάσουμε για το βάψιμο αφού την καθαρίσουμε και τη σκουπίσουμε αρκετά καλά.



Εικόνα 90: Προετοιμασία της κονσόλας για βαφή με μαύρο ματ σπρέι.

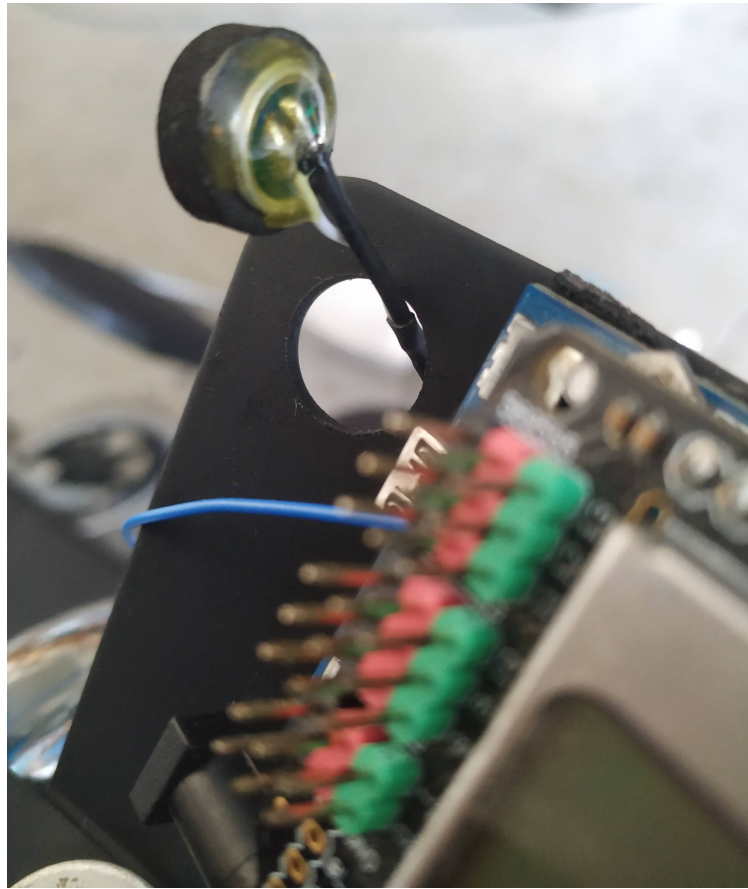


Εικόνα 91: Βαφή της βάσης της οθόνης επιλογών με μαύρο ματ σπρέι και τοποθέτηση αφρώδους μονωτικού υλικού.



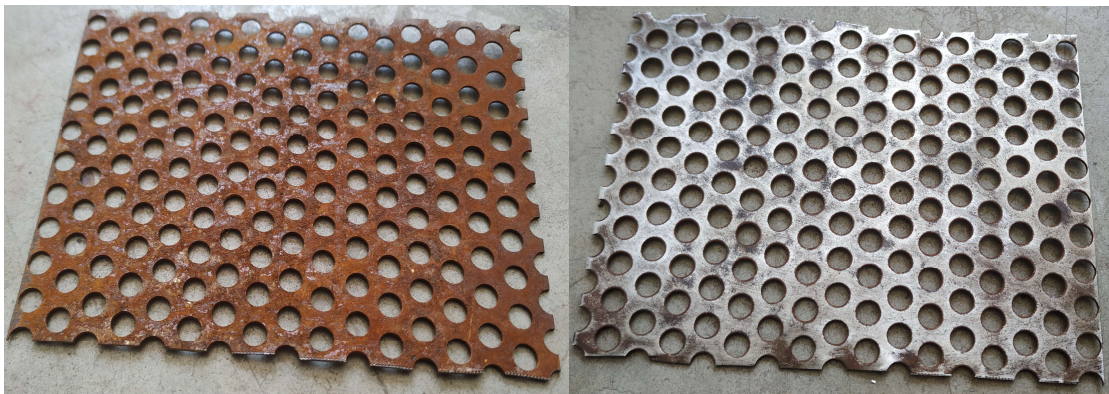
Εικόνα 92: Η κονσόλα με όλα τα εξαρτήματα.

Στην οπή που διανοίξαμε στη βάση της οθόνης θα προσαρμόσουμε το μικρόφωνο για την επιλογή της φωνητικής εντολής. Θα χρησιμοποιηθεί βενζινόκολα γενικής χρήσης για τη στερέωση του.



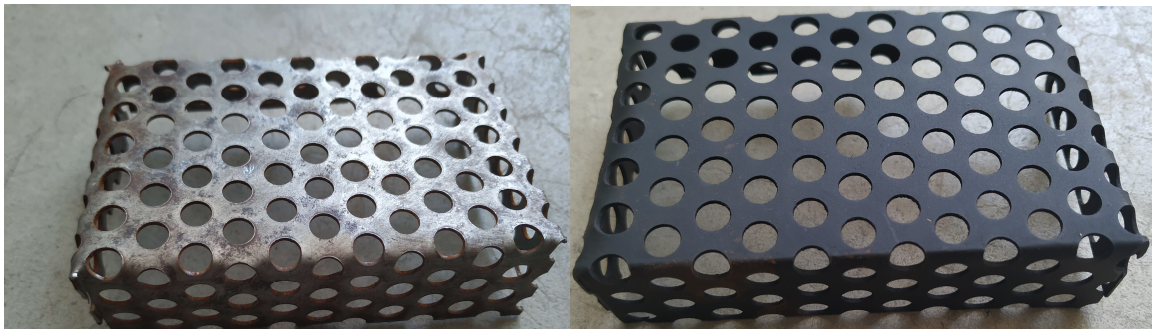
Εικόνα 93: Προσαρμογή του μικροφώνου επάνω στη βάση της οθόνης.

Τέλος, θα κατασκευαστούν δύο καπάκια που θα τοποθετηθούν στις δύο εναπομείναντες εκτεθειμένες πλευρές. Το υλικό που θα χρησιμοποιηθεί, είναι διάτρητη λαμαρίνα, κατάλληλη για την κυκλοφορία αέρα, ώστε να φύγει τα ηλεκτρονικά μέρη που εγκλείονται μέσα στην κονσόλα.

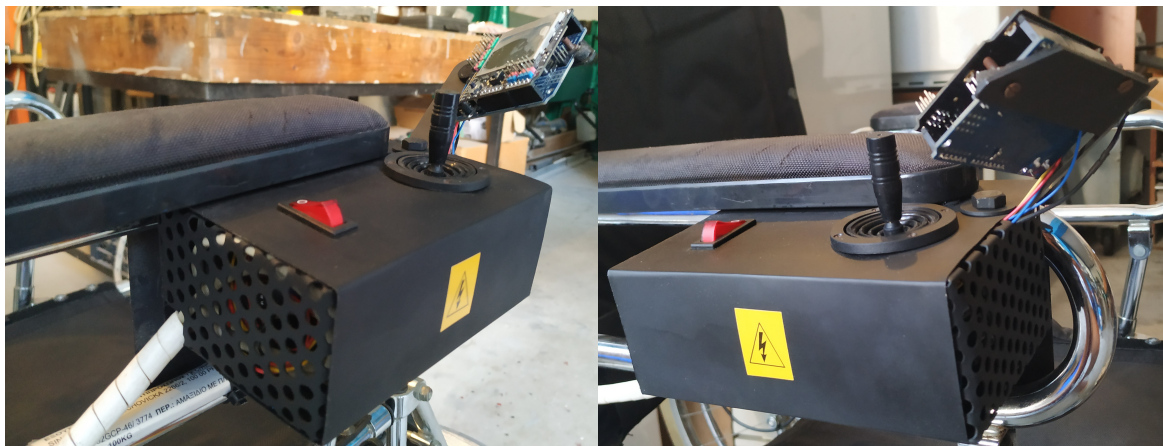


Εικόνα 94: Διάτρητη λαμαρίνα έπειτα από διαδικασία αφαίρεσης σκουριάς.

Όπως και παραπάνω, η διαδικασία για το στραντζάρισμα είναι η ίδια. Ακολουθούν εικόνες που υποδείχνουν συνοπτικά την τελική μορφοποίηση στα καπάκια.



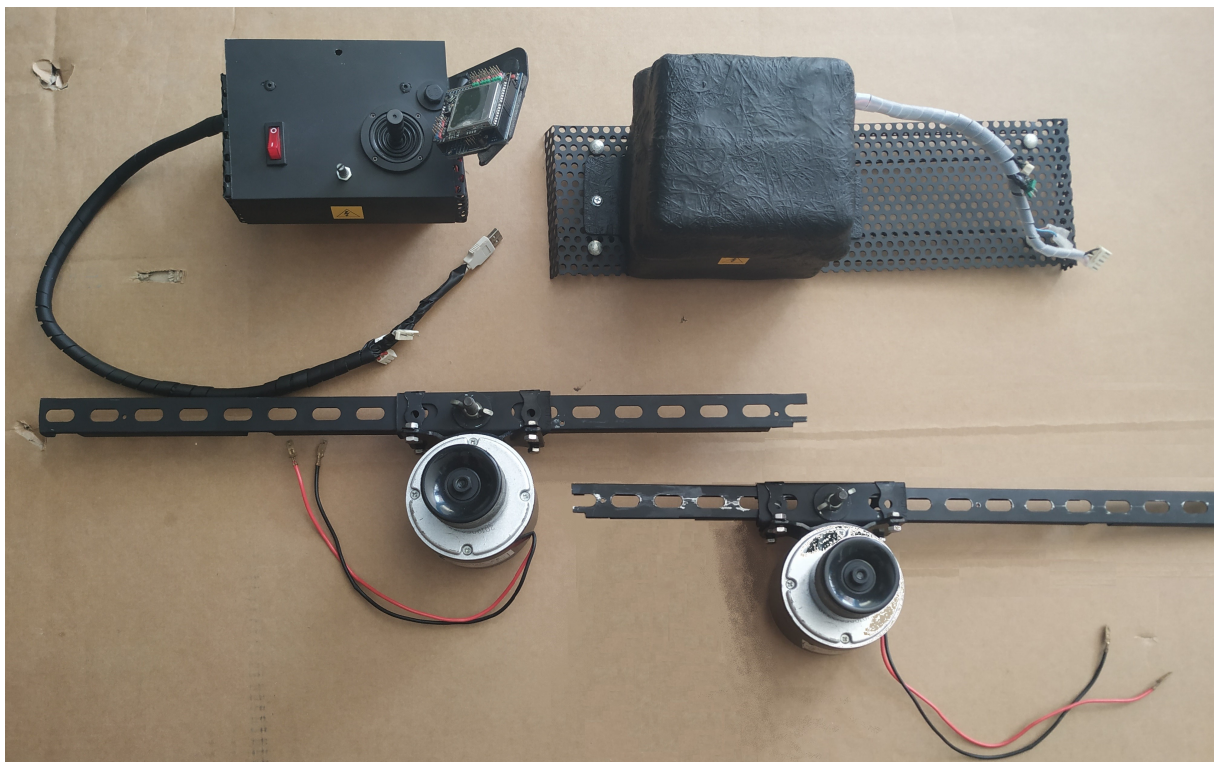
Εικόνα 95: Αριστερά εικονίζεται το ένα καπάκι πριν τη βαφή και δεξιά έπειτα από τη βαφή.



Εικόνα 96: Η κονσόλα έπειτα την τοποθέτηση των καλακιών.

Κεφάλαιο 6 Αποτελέσματα

Στην παρακάτω φωτογραφία, παρουσιάζονται τα τμήματα που κατασκευάστηκαν και αποτελούν το «KIT» συναρμολόγησης για την μετατροπή του συμβατικού αμαξιδίου σε ηλεκτρικό.



Εικόνα 97: Το KIT συναρμολόγησης

Με το παραπάνω «KIT», οποιοσδήποτε διαθέτει ένα συμβατικό αναπηρικό αμαξίδιο, μπορεί εύκολα, γρήγορα, οικονομικά και πάνω απ' όλα με ασφάλεια, να το μετατρέψει μόνος του σε ηλεκτρικό.



Εικόνα 98: Το αναπηρικό αμαξίδιο πριν και μετά την εγκατάσταση του «KIT».

Ο χρόνος που απαιτείται για την εγκατάσταση του «KIT», είναι λιγότερος από πέντε λεπτά για κάποιον που το έχει συναρμολογήσει ξανά και λιγότερος από δέκα λεπτά, για κάποιον που το επιχειρεί πρώτη φορά. Η δε απεγκατάσταση, δεν διαρκεί περισσότερο από δύο λεπτά.

6.1 Συμπεράσματα

Συνοψίζοντας, με ένα κόστος χαμηλότερο από τριακόσια ευρώ (300€), μπορεί κάποιος να μετατρέψει το υπάρχον αμαξίδιο του σε ηλεκτρικό, ειδάλλως, με ένα επιπλέον κόστος διακοσίων ευρώ (200€), μπορεί να προμηθευτεί ένα χειροκίνητο αμαξίδιο και κατόπιν να το μετατρέψει σε ηλεκτρικό, με κόστος χαμηλότερο από πεντακόσια ευρώ (500€) την στιγμή που στην αγορά οι τιμές για την απόκτηση ενός ηλεκτρικού αμαξιδίου, ξεκινάνε από χίλια ευρώ (1000€).

Όπως γίνεται αντιληπτό, η μετατροπή αυτή εκτός από οικονομικά συμφέρουσα, παρέχει και τη δυνατότητα απεριόριστων επεκτάσεων, χάρη στην ύπαρξη του μικροελεγκτή «Arduino UNO».

Στα αρνητικά, η γέφυρα IRF3205 που χρησιμοποιήθηκε, αποδείχθηκε ότι μπορούσε να διαχειριστεί ένταση ρεύματος μέχρι 10A και όχι 15A που ισχυριζόταν ο κατασκευαστής. Έτσι, αναγκαία το ρεύμα που τροφοδοτούσε τους κινητήρες δεν επαρκούσε, με αποτέλεσμα να μπορούμε να εκμεταλλευτούμε μόνο το 70% της απόδοσης τους.

Το αποτέλεσμα ήταν η καταστροφή της γέφυρας, όταν το αμαξίδιο χρειάστηκε παραπάνω ισχύ από αυτή που μπορούσε η γέφυρα να τροφοδοτήσει τους κινητήρες.

6.2 Μελλοντική Εργασία και Επεκτάσεις

Οι επεκτάσεις που μπορεί να επιδεχτεί η παρούσα πτυχιακή εργασία είναι απεριόριστες. Η παρουσία των μικροελεγκτών “Arduino” δίνει τη δυνατότητα ποικίλων τρόπων επέκτασης με χαρακτηριστικά παραδείγματα την ενσωμάτωσης αισθητήρων απόστασης, «GPS», κάμερας και άλλων συστημάτων ανάδρασης.

Βελτίωση μπορεί να επιδεχτεί και ο τρόπος χειρισμού, με ενσωμάτωση ενός κράνους με γυροσκόπιο, που αρχικά είχε επιλεγεί στην παρούσα εργασία, όμως εν τέλει απορρίφθηκε για λόγους εξοικονόμησης χρόνου. Μία πρόταση θα ήταν και ο χειρισμός μέσω ενός κινητού τηλεφώνου, με τη βοήθεια ενός «Bluetooth shield» η μέσω «Wi-Fi». Έτσι ο χρήστης θα μπορεί να χειρίζεται το αμαξίδιο όταν θα βρίσκεται σε πολυσύχναστους χώρους, όπως μια καφετέρια και θα θέλει να μετακινήσει σε ένα ελεύθερο χώρο το αμαξίδιο όση ώρα θα απολαμβάνει το ρόφημα του, μόνος του μέσω του τηλεφώνου του.

Φυσικά οι ιδέες για βελτίωση δεν σταματούν ποτέ. Από την ενσωμάτωση φωτιστικών σωμάτων, για κυκλοφορία την νύχτα αλλά και στον δρόμο, μέχρι εγκατάσταση συστήματος για ασύρματη φόρτιση των μπαταριών και οτιδήποτε άλλο μπορεί να σκεφτεί κάποιος, είναι αρκετά εύκολο μέσω του μικροελεγκτή “Arduino”, να επιτευχθεί.

6.3 Βιβλιογραφία

https://eclass.hmu.gr/main/login_form.php?next=%2Fmain%2Fportfolio.php

<https://blog.sciencemuseum.org.uk/history-of-the-wheelchair/>

<http://news.engineering.utoronto.ca/maker-george-klein-first-electric-wheelchair/>

<http://www.hawking.org.uk/the-computer.html>

<https://www.britannica.com/topic/history-of-the-wheelchair-1971423>

<http://thestuffwebuild.com/projects/diy-electric-motorized-bicycle/>

<https://grobotronics.com/easyvr-shield-3.0-voice-recognition-shield.html>

<https://dronebotworkshop.com/>

Κεφάλαιο 7 Παραρτήματα της Πτοχιακής Εργασίας

Ο Κώδικας για το ARDUINO MASTER

Ο κώδικας που συγγράφηκε για το Arduino, το οποίο έχει το ρόλο «master» και στο οποίο συνδέεται η οθόνη LCD, χρησιμοποιεί τον παρακάτω κώδικα:

```
#include <LCD4884.h>

#include "LCD4884.h"

#include "LOGONEXT.h"

// Include Arduino Wire library for I2C

#include <Wire.h>

// Define Slave I2C Address

#define SLAVE_ADDR 9 // ορισμός διεύθυνσης για τον μικροελεγκτή σε ρόλο Slave

int z=0; //Δήλωση μεταβλητών

int dd=200;

int elegxos=0; // ελεγχος θέσης οθόνης

int elegxos2=0;

int joy=0;

int gyro=0;

int fwni=0;

void setup()

{

    lcd.LCD_init(); // Δήλωση της οθόνης
```

```

lcd.LCD_clear(); // Καθαρισμός της οθόνης

pinMode(7, OUTPUT); //Ορισμός του Pin 7 ως έξοδο

// Initialize I2C communications as Master

Wire.begin();

}

void loop()

{

for (int a=0; a<5; a++) // έλεγχος της οθόνης

{

digitalWrite(7, LOW);

delay(300);

digitalWrite(7, HIGH);

delay(300);

}

lcd.LCD_clear();

lcd.LCD_write_string(10,0,"TEI OF CRETE", MENU_HIGHLIGHT); // εισαγωγικό
μηνυμα

lcd.LCD_write_string(0,2,"Department of", MENU_NORMAL);

lcd.LCD_write_string(0,3,"Mechanical Engineering", MENU_NORMAL);

delay(3000);

lcd.LCD_clear();

lcd.LCD_write_string(0, 0, "Project:", MENU_NORMAL);

lcd.LCD_write_string(0,2,"ELECTRIC", MENU_NORMAL);

lcd.LCD_write_string(0, 3, "WHEELCHAIR", MENU_NORMAL);

```



```

    lcd.LCD_write_string(0,5,"KALWS ORISATE", MENU_HIGHLIGHT);
delay(3000);
lcd.LCD_clear();
delay(500);

lcd.LCD_clear();
lcd.LCD_write_string(25,0,"MENU", MENU_NORMAL); // Κυρίως Μενού
lcd.LCD_write_string(0,1,"JOYSTICK", MENU_HIGHLIGHT);
lcd.LCD_write_string(0,2,"GYROSKOPIO", MENU_NORMAL);
lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_NORMAL);
lcd.LCD_write_string(0,4,"EXIT", MENU_NORMAL);
do
{
    z=analogRead(0); //Μεταβλητή που παίρνει την τιμή των βόλτ κάθε εντολής του
χειριστηρίου
    if (z==0)
    {
        //////////////////////////////////////
    }
    else
        if (z>100 && z<150) // Αν έχει πιεστεί προς τα μέσα το πλήκτρο
        {
            if (elegxos==0){ // Αν βρίσκεται στη θέση του joystick
                lcd.LCD_clear();

```

```

lcd.LCD_write_string(0,0,"epilexthike", MENU_NORMAL);

lcd.LCD_write_string(0,3,"joystick", MENU_NORMAL);

delay(5000);

lcd.LCD_clear();

lcd.LCD_write_string(15,0,"", MENU_NORMAL); // ignore MENU_NORMAL
for now

delay(800);

lcd.LCD_write_string(25,2,"ETOIMO", MENU_NORMAL); // ignore
MENU_NORMAL for now

delay(800);

lcd.LCD_write_string(20,4,"GIA", MENU_NORMAL); // ignore
MENU_NORMAL for now

lcd.LCD_write_string(45,4,"XRISI", MENU_NORMAL); // ignore
MENU_NORMAL for now

delay(800);

//metavasi sto arduino slave

Wire.beginTransmission(SLAVE_ADDR);

joy=1;

Wire.write(joy);

Wire.endTransmission();

}

else if (elegchos==1){ //An βρίσκεται στη θέση του γυροσκόπιου

lcd.LCD_clear();

lcd.LCD_write_string(0,0,"epilexthike", MENU_NORMAL);

lcd.LCD_write_string(0,3,"GYROSKOPIO", MENU_NORMAL);

```

```

        delay(5000);

        lcd.LCD_clear();

        lcd.LCD_write_string(15,0,"",    MENU_NORMAL);    //    ignore
MENU_NORMAL for now

        delay(800);

        lcd.LCD_write_string(25,2,"ETOIMO",    MENU_NORMAL);    //    ignore
MENU_NORMAL for now

        delay(800);

        lcd.LCD_write_string(20,4,"GIA",    MENU_NORMAL);    //    ignore
MENU_NORMAL for now

        lcd.LCD_write_string(45,4,"XRISI",    MENU_NORMAL);    //    ignore
MENU_NORMAL for now

        delay(800);

        //metavasi sto arduino slave

        Wire.beginTransmission(SLAVE_ADDR);

        gyro=2;

        Wire.write(gyro);

        Wire.endTransmission();

    }

    else { // Όταν βρίσκεται στην θέση της φωνητικής εντολής

        lcd.LCD_clear();

        lcd.LCD_write_string(0,0,"epilexthike", MENU_NORMAL);

        lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_NORMAL);

        delay(5000);

        lcd.LCD_clear();

```

```

        lcd.LCD_write_string(15,0,"", MENU_NORMAL); // ignore
MENU_NORMAL for now

        delay(800);

        lcd.LCD_write_string(25,2,"ETOIMO", MENU_NORMAL); // ignore
MENU_NORMAL for now

        delay(800);

        lcd.LCD_write_string(20,4,"GIA", MENU_NORMAL); // ignore
MENU_NORMAL for now

        lcd.LCD_write_string(45,4,"XRISI", MENU_NORMAL); // ignore
MENU_NORMAL for now

        delay(800);

        //metavasi sto arduino slave

        Wire.beginTransmission(SLAVE_ADDR);

        fwni=3;

        Wire.write(fwni);

        Wire.endTransmission();

    }

}

else

    if (z>150 && z<350 && elegxos==0) // Αν έχει πιεστεί προς τα κάτω το χειριστήριο

    {lcd.LCD_clear();

        lcd.LCD_write_string(25,0,"MENU", MENU_NORMAL); //Πηγαίνει στη θέση
του γυροσκόπιου

        lcd.LCD_write_string(0,1,"JOYSTICK", MENU_NORMAL);

```

```

    lcd.LCD_write_string(0,2,"GYROSKOPIO", MENU_HIGHLIGHT);
    lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_NORMAL);
    lcd.LCD_write_string(0,4,"EXIT", MENU_NORMAL);

    elegxos = 1;
    elegxos2=1;
    delay(500);
}
else
    if (z>150 && z<350 && elegxos==1) { // Αν πιεστεί προς τα κάτω και βρίσκεται
στη θέση το γυροσκόπιου
        lcd.LCD_clear();
        lcd.LCD_write_string(25,0,"MENU", MENU_NORMAL); //Πηγαίνει στη θέση
της φωνητικής εντολής
        lcd.LCD_write_string(0,1,"JOYSTICK", MENU_NORMAL);
        lcd.LCD_write_string(0,2,"GYROSKOPIO", MENU_NORMAL);
        lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_HIGHLIGHT);
        lcd.LCD_write_string(0,4,"EXIT", MENU_NORMAL);
        elegxos=2;
        elegxos2=2;
        delay(500);
    }
    else if (z>150 && z<350 && elegxos==2) // Αν πιεστεί προς τα κάτω και βρίσκεται στη
θέση της φωνητικής εντολής
    {
        lcd.LCD_clear();

```



```
lcd.LCD_write_string(25,0,"MENU", MENU_NORMAL);//Πηγαίνει στην αρχική  
θέση του joystick
```

```
lcd.LCD_write_string(0,1,"JOYSTICK", MENU_HIGHLIGHT);
```

```
lcd.LCD_write_string(0,2,"GYROSKOPIO", MENU_NORMAL);
```

```
lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_NORMAL);
```

```
lcd.LCD_write_string(0,4,"EXIT", MENU_NORMAL);
```

```
elegxos=0;
```

```
elegxos2=0;
```

```
delay(500);
```

```
}
```

```
else if (z>510 && z<750 && elegxos2==2) // Ιδιος έλεγχος για τις περιπτώσεις που  
πιεστεί προς τα πάνω το πλήκτρο
```

```
{
```

```
lcd.LCD_clear();
```

```
lcd.LCD_write_string(25,0,"MENU", MENU_NORMAL);
```

```
lcd.LCD_write_string(0,1,"JOYSTICK", MENU_NORMAL);
```

```
lcd.LCD_write_string(0,2,"GYROSKOPIO", MENU_HIGHLIGHT);
```

```
lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_NORMAL);
```

```
lcd.LCD_write_string(0,4,"EXIT", MENU_NORMAL);
```

```
elegxos=1;
```

```
elegxos2=1;
```

```
delay(500);
```

```
}
```

```
else if (z>510 && z<750 && elegxos2==1)
```

```
{
```

```

    lcd.LCD_clear();

    lcd.LCD_write_string(25,0,"MENU", MENU_NORMAL);

    lcd.LCD_write_string(0,1,"JOYSTICK", MENU_HIGHLIGHT);

    lcd.LCD_write_string(0,2,"GYROSKOPIO", MENU_NORMAL);

    lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_NORMAL);

    lcd.LCD_write_string(0,4,"EXIT", MENU_NORMAL);

    elegxos=0;

    elegxos2=0;

    delay(500);

    }

else if (z>510 && z<750 && elegxos2==0)

{

    lcd.LCD_clear();

    lcd.LCD_write_string(25,0,"MENU", MENU_NORMAL);

    lcd.LCD_write_string(0,1,"JOYSTICK", MENU_NORMAL);

    lcd.LCD_write_string(0,2,"GYROSKOPIO", MENU_NORMAL);

    lcd.LCD_write_string(0,3,"FWNITIKI ENTOLI", MENU_HIGHLIGHT);

    lcd.LCD_write_string(0,4,"EXIT", MENU_NORMAL);

    elegxos=2;

    elegxos2=2;

    delay(500);

    }

}

while (1>0);

```

```
}
```

Ο Κώδικας για το ARDUINO SLAVE

```
#include <Wire.h>

#define SLAVE_ADDR 9

#include "Arduino.h"

#if !defined(SERIAL_PORT_MONITOR)

    #error "Arduino version not supported. Please update your IDE to the latest version."

#endif

#if defined(SERIAL_PORT_USBVIRTUAL)

    // Shield Jumper on HW (for Leonardo and Due)

    #define port SERIAL_PORT_HARDWARE

    #define pcSerial SERIAL_PORT_USBVIRTUAL

#else

    // Shield Jumper on SW (using pins 12/13 or 8/9 as RX/TX)

    #include "SoftwareSerial.h"

    SoftwareSerial port(12, 13);

    #define pcSerial SERIAL_PORT_MONITOR

#endif

#include "EasyVR.h"

EasyVR easyvr(port);
```

```
#define SND_BEEP          0
#define SND_Access_denied  1
#define SND_Access_granted 2
#define SND_Hello          3
#define SND_Please_repeat  4
#define SND_Please_say_your_password 5
#define SND_Please_talk_louder 6
#define SND_EPANALAVATE    9
#define SND_KALWSORISATE  10
#define SND_TFE             11
#define SND_TX              12
```

```
int8_t bits = 4;
```

```
int8_t set = 0;
```

```
int8_t group = 0;
```

```
uint32_t mask = 0;
```

```
uint8_t train = 0;
```

```
uint8_t grammars = 0;
```

```
int8_t lang = 0;
```

```
char name[33];
```

```
bool useCommands = true;
```

```
bool useTokens = false;
```

```
bool isSleeping = false;
```

```
const char* entoli="STOP";
```

```

int potvalX = 0; //Δήλωση ποτενσιόμετρου άξονα X
int potvalY = 0; //Δήλωση ποτενσιόμετρου άξονα Y
int speedRight = 0;
int speedLeft = 0;
int potenciometro = A5; // select the input pin for the potentiometer
int motorregulation = 0; // variable to store the value coming from the sensor
int elegxos_ixou = 1;
int elegxos_ixou3 = 1;

// Variable for received data
int rd;

void setup() {

// Initialize I2C communications as Slave
Wire.begin(SLAVE_ADDR);

// Function to run when data received from master
Wire.onReceive(receiveEvent);

// setup PC serial port
pcSerial.begin(9600);

// bridge mode?

```



```

int mode = easyvr.bridgeRequested(pcSerial);

switch (mode)
{
case EasyVR::BRIDGE_NONE:
    // setup EasyVR serial port
    port.begin(9600);

    // run normally
    pcSerial.println(F("---"));
    pcSerial.println(F("Bridge not started!"));
    break;

case EasyVR::BRIDGE_NORMAL:
    // setup EasyVR serial port (low speed)
    port.begin(9600);

    // soft-connect the two serial ports (PC and EasyVR)
    easyvr.bridgeLoop(pcSerial);

    // resume normally if aborted
    pcSerial.println(F("---"));
    pcSerial.println(F("Bridge connection aborted!"));
    break;

case EasyVR::BRIDGE_BOOT:
    // setup EasyVR serial port (high speed)
    port.begin(115200);

```

```

// soft-connect the two serial ports (PC and EasyVR)
easyvr.bridgeLoop(pcSerial);

// resume normally if aborted
pcSerial.println(F("---"));
pcSerial.println(F("Bridge connection aborted!"));
break;
}

easyvr.setPinOutput(EasyVR::IO1, LOW);
easyvr.setLevel(EasyVR::HARDER);
easyvr.playSound(SND_KALWSORISATE, EasyVR::VOL_FULL);
easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);

// initialize EasyVR
while (!easyvr.detect())
{
pcSerial.println(F("EasyVR not detected!"));
delay(10);
}

easyvr.setPinOutput(EasyVR::IO1, LOW);
pcSerial.print(F("EasyVR detected, version "));
pcSerial.println(easyvr.getID());
easyvr.setTimeout(5);
lang = EasyVR::ENGLISH;

```

```
easyvr.setLanguage(lang);

int16_t count = 0;

pcSerial.print(F("Sound table: "));
if (easyvr.dumpSoundTable(name, count))
{
    pcSerial.println(name);
    pcSerial.print(F("Sound entries: "));
    pcSerial.println(count);
}
else
    pcSerial.println(F("n/a"));

pcSerial.print(F("Custom Grammars: "));
grammars = easyvr.getGrammarsCount();
if (grammars > 4)
{
    pcSerial.println(grammars - 4);
    for (set = 4; set < grammars; ++set)
    {
        pcSerial.print(F("Grammar "));
        pcSerial.print(set);
    }
}
```

```

uint8_t flags, num;

if (easyvr.dumpGrammar(set, flags, num))
{
    pcSerial.print(F(" has "));
    pcSerial.print(num);
    if (flags & EasyVR::GF_TRIGGER)
        pcSerial.println(F(" trigger"));
    else
        pcSerial.println(F(" command(s)"));
}
else
    pcSerial.println(F(" error"));

for (int8_t idx = 0; idx < num; ++idx)
{
    pcSerial.print(idx);
    pcSerial.print(F(" = "));
    if (!easyvr.getNextWordLabel(name))
        break;
    pcSerial.println(name);
}
}
}
else

```

```

pcSerial.println(F("n/a"));

if (easyvr.getGroupMask(mask))
{
    uint32_t msk = mask;
    for (group = 0; group <= EasyVR::PASSWORD; ++group, msk >>= 1)
    {
        if (!(msk & 1)) continue;
        if (group == EasyVR::TRIGGER)
            pcSerial.print(F("Trigger: "));
        else if (group == EasyVR::PASSWORD)
            pcSerial.print(F("Password: "));
        else
        {
            pcSerial.print(F("Group "));
            pcSerial.print(group);
            pcSerial.print(F(" has "));
        }
        count = easyvr.getCommandCount(group);
        pcSerial.print(count);
        if (group == 0)
            pcSerial.println(F(" trigger(s)"));
        else
            pcSerial.println(F(" command(s)"));
    }
}

```

```

for (int8_t idx = 0; idx < count; ++idx)
{
    if (easyvr.dumpCommand(group, idx, name, train))
    {
        pcSerial.print(idx);
        pcSerial.print(F(" = "));
        pcSerial.print(name);
        pcSerial.print(F(", Trained "));
        pcSerial.print(train, DEC);
        if (!easyvr.isConflict())
            pcSerial.println(F(" times, OK"));
        else
        {
            int8_t confl = easyvr.getWord();
            if (confl >= 0)
                pcSerial.print(F(" times, Similar to Word "));
            else
            {
                confl = easyvr.getCommand();
                pcSerial.print(F(" times, Similar to Command "));
            }
            pcSerial.println(confl);
        }
    }
}

```



```
    }  
  }  
}  
group = 0;  
useCommands = (mask != 0);  
mask |= 1; // force to use trigger  
isSleeping = false;  
pcSerial.println(F("---"));  
}
```

```
const char* ws0[] =  
{  
  "ROBOT",  
};  
const char* ws1[] =  
{  
  "LEFT",  
  "RIGHT",  
  "UP",  
  "DOWN",  
  "FORWARD",  
  "BACKWARD",  
  "STOP"
```

```

};

const char* ws7[] =
{
    "MPROSTA",
    "PISW",
    "ARISTERA",
    "DEXIA",
    "STOP",
    "STOP2",
    "STOP3"
};

const char** ws[] = { ws0, ws1, ws7};

bool checkMonitorInput()
{
    if (pcSerial.available() <= 0)
        return false;

    // check console commands
    int16_t rx = pcSerial.read();
    if (rx == '?')
    {
        setup();
        return true;
    }
}

```

```

}
if (isSleeping)
{
    // any character received will exit sleep
    isSleeping = false;
    easyvr.stop();
    pcSerial.println(F("Forced wake-up!"));
    return true;
}
if (rx == 'l')
{
    easyvr.stop();
    lang++;
    if (easyvr.setLanguage(lang) || easyvr.setLanguage(lang = 0))
    {
        pcSerial.print(F("Language set to "));
        pcSerial.println(lang);
    }
    else
        pcSerial.println(F("Error while setting language!"));
}
if (rx == 'b')
{
    useTokens = false;

```

```
useCommands = false;

set++;

if (set > 3)
    set = 0;
}

if (rx == 'g' && grammars > 4)
{
    useTokens = false;
    useCommands = false;
    set++;
    if (set >= grammars)
        set = 4;
}

if (rx == 'c')
{
    useTokens = false;
    useCommands = true;
    do
    {
        group++;
        if (group > EasyVR::PASSWORD)
            group = 0;
    }
    while (!((mask >> group) & 1));
```

```
}  
if (rx == 'k')  
{  
    useTokens = true;  
}  
if (rx == '4')  
{  
    bits = 4;  
}  
if (rx == '8')  
{  
    bits = 8;  
}  
if (rx == 'n')  
{  
    int16_t num = 0;  
    delay(5);  
    while ((rx = pcSerial.read()) >= 0)  
    {  
        delay(5);  
        if (isdigit(rx))  
            num = num * 10 + (rx - '0');  
        else  
            break;  
    }  
}
```

```

    }
    pcSerial.print(F("Play token "));
    pcSerial.println(num);
    easyvr.stop();
    easyvr.sendToken(bits, num);
}
if (rx == 's')
{
    int16_t num = 0;
    delay(5);
    while ((rx = pcSerial.read()) >= 0)
    {
        delay(5);
        if (isdigit(rx))
            num = num * 10 + (rx - '0');
        else
            break;
    }
    pcSerial.print(F("Play sound "));
    pcSerial.println(num);
    easyvr.stop();
    easyvr.playSound(num, EasyVR::VOL_DOUBLE);
}
if (rx == 'd')

```



```

{
  easyvr.stop();
  pcSerial.println(F("Play tones:"));
  int16_t num = 0;
  delay(5);
  while ((rx = pcSerial.read()) >= 0)
  {
    delay(5);
    if (isdigit(rx))
      num = rx - '0';
    else if (rx == '*')
      num = 10;
    else if (rx == '#')
      num = 11;
    else if (rx >= 'A' && rx <= 'D')
      num = rx - 'A';
    else if (rx == '_')
      num = -1;
    else
      break;
    pcSerial.print(num);
    if (easyvr.playPhoneTone(num, 3))
      pcSerial.println(F(" OK"));
    else

```

```

    pcSerial.println(F(" ERR"));
}
}
if (rx == 'm')
{
    int16_t num = 0;
    delay(5);
    while ((rx = pcSerial.read()) >= 0)
    {
        delay(5);
        if (isdigit(rx))
            num = num * 10 + (rx - '0');
        else
            break;
    }
    pcSerial.print(F("Mic distance "));
    pcSerial.println(num);
    easyvr.stop();
    easyvr.setMicDistance(num);
}
if (rx == 'w')
{
    int8_t mode = 0;
    delay(5);

```

```

while ((rx = pcSerial.read()) >= 0)
{
    delay(5);
    if (rx == 'w')
        mode = EasyVR::WAKE_ON_WHISTLE;
    if (rx == '2')
        mode = EasyVR::WAKE_ON_2CLAPS;
    if (rx == '3')
        mode = EasyVR::WAKE_ON_3CLAPS;
    if (rx == 'l')
        mode = EasyVR::WAKE_ON_LOUDSOUND;
}
pcSerial.print(F("Sleep mode "));
pcSerial.println(mode);
easyvr.stop();
easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
isSleeping = easyvr.sleep(mode);
return true;
}

if (rx >= 0)
{
    easyvr.stop();
    pcSerial.flush();
}

```

```

    return true;
}

return false;

pinMode(5,OUTPUT); // PWM κινητήρα 1 (R)
pinMode(6,OUTPUT); // Κατεύθυνση κινητήρα 1 για το κανάλι A
pinMode(9,OUTPUT); // PWM κινητήρα 2 (L)
pinMode(10,OUTPUT); // Κατεύθυνση κινητήρα 2 για το κανάλι B
pinMode(A5,INPUT); // Είσοδος ποτενσιομέτρου

}

void receiveEvent() {
    // read one character from the I2C
    rd = Wire.read();
    Serial.println(rd);
}

void loop() {

    motorregulation = analogRead(A5);
    motorregulation= map(motorregulation, 230, 680, 10000, -10000);

```

```

//Τομέας Χειριστηρίου
while (rd==1){

//ηχητικο σημα
  elegxos_ixou3=1;
  if (elegxos_ixou==1){
    easyvr.setPinOutput(EasyVR::IO1, LOW);
    easyvr.setLevel(EasyVR::HARDER);
    easyvr.playSound(SND_TX, EasyVR::VOL_FULL);
    easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);
    elegxos_ixou=0;

  }

//Ποτενσιόμετρα
  potvalX = analogRead(A2);
  potvalX = map(potvalX, 0, 1023, -125, 125); //Καλιμπράρισμα χειριστηρίου

  potvalY = analogRead(A3);
  potvalY = map(potvalY, 0, 1023, -125, 125); //Καλιμπράρισμα χειριστηρίου
  delay(10);

//Ελεγχος κινητήρα
//Σταμάτημα

```

```

if ((potvalX >= -15) && (potvalX <= 15) && (potvalY<=15) && (potvalY >=-15))
{
    digitalWrite(6, LOW);
    digitalWrite(10, LOW);
    analogWrite(5, 0);
    analogWrite(9, 0);
    Serial.println("STOP");
    delay(10);
}
else {
    //Στρίψιμο προς τα ΔΕΞΙΑ
    if (potvalX > 15) {
        speedRight=potvalX;
        speedLeft=potvalY;
        //ΚΙΝΗΤΗΡΑΣ ΔΕΞΙΑ
        digitalWrite(6, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A
        analogWrite(5, speedRight); //Περιστροφή του Κινητήρα στο κανάλι A με ταχύτητα
απο το ποτενσιομετρο

        //kinitiras aristera
        speedLeft = 0;
        digitalWrite(10, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B
        analogWrite(9, speedLeft); //Περιστροφή του Κινητήρα στο κανάλι B
        delay(10);

```



```

}

else if

//Στρίψιμο προς τα αριστερά

(potvalX < -15) {

    speedRight = 0;

    //ΚΙΝΗΤΗΡΑΣ ΔΕΞΙΑ

    digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A

    analogWrite(5, speedRight); //Περιστροφή του Κινητήρα στο κανάλι A με ταχύτητα
απο το ποτενσιομετρο

    //kinitiras aristera

    speedLeft = potvalX * (-1);

    digitalWrite(10, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B

    analogWrite(9, speedLeft); //Περιστροφή του Κινητήρα στο κανάλι B με μέγστη
ταχύτητα

    delay(10);

    Serial.println("ARISTERA");

}

else if

//Οπισθεν

(potvalY < -15) {

    //kinitiras de3ia

    speedRight= potvalY * (-1);

    speedLeft=speedRight;

```

```

if (motorregulation > 2) {
    speedRight = speedRight + (motorregulation/500);
}

digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα πίσω για το κανάλι A
analogWrite(5, speedRight); //Περιστροφή του Κινητήρα στο κανάλι A με μέγιστη
ταχύτητα

//kinitiras aristera
if (motorregulation < -2) {
    speedLeft = speedLeft + ((motorregulation/500)*(-1));
}

digitalWrite(10, LOW); //Καθορισμός κίνησης προς τα πίσω για το κανάλι B
analogWrite(9, speedLeft); //Περιστροφή του Κινητήρα στο κανάλι B με μέγιστη
ταχύτητα

    Serial.println("OPIS8EN");
}

else {

    //kinitiras de3ia

    speedRight= potvalY;

    speedLeft=speedRight;

    if (motorregulation > 2) {

        speedRight = speedRight + (motorregulation/500);

    }

    digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A

```

```
    analogWrite(5, speedRight); //Περιστροφή του Κινητήρα στο κανάλι A με μέγιστη  
    ταχύτητα
```

```
    //kinitiras aristera  
    if (motorregulation < -2) {  
        speedLeft = speedLeft + ((motorregulation/500)*(-1));  
    }  
    digitalWrite(10, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B  
    analogWrite(9, speedLeft); //Περιστροφή του Κινητήρα στο κανάλι B με μέγιστη  
    ταχύτητα  
    Serial.println("empros");  
    delay(10);  
    }  
    }  
    }
```

```
//Τομέας Φωνητικής εντολής
```

```
while (rd==3){
```

```
    checkMonitorInput();
```

```
    //ηχητικο σημα
```

```
    elegxos_ixou=1;
```

```
    if (elegxos_ixou3==1){
```

```
        easyvr.setPinOutput(EasyVR::IO1, LOW);
```

```

easyvr.setLevel(EasyVR::HARDER);

easyvr.playSound(SND_TFE, EasyVR::VOL_FULL);

easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);

elegxos_ixou3=0;
}

if (!isSleeping)
{
    easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
    if (useTokens)
    {
        pcSerial.print(F("Detect a "));
        pcSerial.print(bits);
        pcSerial.println(F(" bit token ..."));
        easyvr.detectToken(bits, EasyVR::REJECTION_AVG, 0);
    }
    else if (useCommands)
    {
        pcSerial.print(F("Say a command in Group "));
        pcSerial.println(7);
        easyvr.recognizeCommand(7);
    }
    else
    {

```

```

    pcSerial.print(F("Say a word in Wordset "));
    pcSerial.println(set);
    easyvr.recognizeWord(set);
}
}
do
{
    if (checkMonitorInput())
        return;
}
while (!easyvr.hasFinished());
isSleeping = false;

easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off

if (easyvr.isAwakened())
{
    pcSerial.println(F("Audio wake-up!"));
    return;
}

int16_t idx;
if (useTokens)
{

```

```

idx = easyvr.getToken();

if (idx >= 0)
{
    pcSerial.print(F("Token: "));
    pcSerial.println(idx);
    easyvr.playSound(0, EasyVR::VOL_FULL);
}
}

// handle voice recognition

idx = easyvr.getWord();

if (idx >= 0)
{
    pcSerial.print(F("Word: "));
    pcSerial.print(easyvr.getWord());
    pcSerial.print(F(" = "));
    if (useCommands)
        pcSerial.println(ws[group][idx]);

    // --- optional: builtin words can be retrieved from the module
    else if (set < 4)
        pcSerial.println(ws[set][idx]);

    // ---
    else
    {

```



```

uint8_t flags, num;

if (easyvr.dumpGrammar(set, flags, num))
    while (idx-- >= 0)
    {
        if (!easyvr.getNextWordLabel(name))
            break;
    }
if (idx < 0)
    pcSerial.println(name);
else
    pcSerial.println();
}

easyvr.playSound(0, EasyVR::VOL_FULL);
}
else
{
    idx = easyvr.getCommand();
    if (idx >= 0)
    {
        pcSerial.print(F("Command: "));
        pcSerial.print(easyvr.getCommand());
        if (easyvr.dumpCommand(group, idx, name, train))

```

```

    {
        pcSerial.print(F(" = "));
        pcSerial.println(name);
    }

    entoli=name;

    pcSerial.println();

    pcSerial.print("TREXON ENTOLI");

    pcSerial.println();

    pcSerial.println(entoli);

if (easyvr.getCommand()==0)
{
    pcSerial.println();

    pcSerial.print("TREXON ENTOLI");

    pcSerial.println();

    pcSerial.println("MPROSTA");

    easyvr.setPinOutput(EasyVR::IO1, LOW);

    easyvr.setLevel(EasyVR::HARDER);

    easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);

    //kinitiras de3ia

    if (motorregulation > 2) {

```

```

digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A
analogWrite(5, 100+(motorregulation/500)); //Περιστροφή του Κινητήρα στο κανάλι A
με PWM απο το ποτενσιομετρο
}
else {
digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A
analogWrite(5, 100); //Περιστροφή του Κινητήρα στο κανάλι A με PWM 100
}
//kinitiras aristera
if (motorregulation < -2) {
digitalWrite(10, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B
analogWrite(9, 100+(motorregulation/500)*(-1)); //Περιστροφή του Κινητήρα στο
κανάλι B με PWM απο το ποτενσιομετρο
}
else {
digitalWrite(10, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B
analogWrite(9, 100); //Περιστροφή του Κινητήρα στο κανάλι B με PWM 100
}
}

if (easyvr.getCommand()==1)

{
pcSerial.println();
pcSerial.print("TREXON ENTOLI");

```

```

pcSerial.println();

pcSerial.println("PISW");

easyvr.setPinOutput(EasyVR::IO1, LOW);

easyvr.setLevel(EasyVR::HARDER);

easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);

//kinitiras de3ia
if (motorregulation > 2) {
digitalWrite(6, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A
analogWrite(5, 100+(motorregulation/500)); //Περιστροφή του Κινητήρα στο κανάλι A
με PWM απο το ποτενσιομετρο
}
else {
digitalWrite(6, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A
analogWrite(5, 100); //Περιστροφή του Κινητήρα στο κανάλι A με PWM 100
}

//kinitiras aristera
if (motorregulation < -2) {
digitalWrite(10, LOW); //Καθορισμός κίνησης προς τα πίσω για το κανάλι B
analogWrite(9, 100+(motorregulation/500)*(-1)); //Περιστροφή του Κινητήρα στο
κανάλι B με PWM απο το ποτενσιομετρο
}
else {

```

```

digitalWrite(10, LOW); //Καθορισμός κίνησης προς τα πίσω για το κανάλι B
analogWrite(9, 100); //Περιστροφή του Κινητήρα στο κανάλι B με PWM 100
}

}

if (easyvr.getCommand()==2)

{
pcSerial.println();
pcSerial.print("TREXON ENTOLI");
pcSerial.println();
pcSerial.println("ARISTERA");

easyvr.setPinOutput(EasyVR::IO1, LOW);
easyvr.setLevel(EasyVR::HARDER);
easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);

//kinitiras de3ia

digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A
analogWrite(5, 100); //Περιστροφή του Κινητήρα στο κανάλι A με PWM 100

//kinitiras aristera

digitalWrite(10, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B

```

```

    analogWrite(9, 0); //Σταμάτημα της περιστροφής του Κινητήρα στο κανάλι B
    delay(10);
}

if (easyvr.getCommand()==3)

{
    pcSerial.println();
    pcSerial.print("TREXON ENTOLI");
    pcSerial.println();
    pcSerial.println("DEXIA");

    easyvr.setPinOutput(EasyVR::IO1, LOW);
    easyvr.setLevel(EasyVR::HARDER);
    easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);

    //kinitiras de3ia
    digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A
    analogWrite(5, 0); //Σταμάτημα της περιστροφής του Κινητήρα στο κανάλι A

    //kinitiras aristera
    digitalWrite(10, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B
    analogWrite(9, 100); //Περιστροφή του Κινητήρα στο κανάλι B με PWM 100
    delay(10);
}

```

```

}

if (easyvr.getCommand()==4 || easyvr.getCommand()==5 || easyvr.getCommand()==6)

{

pcSerial.println();

pcSerial.print("TREXON ENTOLI");

pcSerial.println();

pcSerial.println("STOP");

pcSerial.println();

easyvr.setPinOutput(EasyVR::IO1, LOW);

easyvr.setLevel(EasyVR::HARDER);

easyvr.playSound(SND_BEEP, EasyVR::VOL_FULL);

//kinitiras de3ia

digitalWrite(6, LOW); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι A

analogWrite(5,0); //Σταμάτημα της περιστροφής του Κινητήρα στο κανάλι A

//kinitiras aristera

digitalWrite(10, HIGH); //Καθορισμός κίνησης προς τα εμπρός για το κανάλι B

analogWrite(9,0); //Σταμάτημα της περιστροφής του Κινητήρα στο κανάλι B

}

else{

```



```

    delay(10);
}

}

else // errors or timeout
{
    if (easyvr.isTimeout())
        pcSerial.println(F("Timed out, try again..."));
    int16_t err = easyvr.getError();
    if (err >= 0)
    {
        pcSerial.print(F("SFALMA ANAGNWRISIS"));
        pcSerial.println(err, HEX);
        easyvr.setPinOutput(EasyVR::IO1, LOW);
        easyvr.setLevel(EasyVR::HARDER);
        easyvr.playSound(SND_EPANALAVATE, EasyVR::VOL_FULL);
    }
}
}
}
}
}
}

```