

# Ανάπτυξη Συστήματος Τηλεχειρισμού Οχήματος Μέσω Smart Συσκευής



Εργασία που υποβλήθηκε στο  
Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική,  
του Διεθνούς Πανεπιστημίου της Ελλάδος,  
για τη μερική εκπλήρωση υποχρεώσεων  
για το Δίπλωμα Ειδίκευσης στη Ρομποτική

Εκπονητής: **Γεώργιος Καμνιάτσος**

Επιβλέπων Καθηγητής: **Δημήτριος Σαγρής**

## Ανάπτυξη Συστήματος Τηλεχειρισμού Οχήματος Μέσω Smart Συσκευής

Εργασία που υποβλήθηκε στο Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική, του Διεθνούς Πανεπιστημίου της Ελλάδος, για τη μερική εκπλήρωση υποχρεώσεων για το Δίπλωμα Ειδίκευσης στη Ρομποτική

**Εκπονητής: Γεώργιος Καμνιάτσος**

Επιβλέπων Καθηγητής: Δημήτριος Σααργής

Σέρρες, 23-4-2024

### **Υπεύθυνη Δήλωση Φοιτητών:**

Ο κάτωθι υπογεγραμμένος φοιτητής, έχοντας επίγνωση των συνεπειών του Νόμου περί λογοκλοπής, δηλώνει υπεύθυνα ότι είναι συγγραφέας αυτής της Μεταπτυχιακής Εργασίας, αναλαμβάνοντας την ευθύνη επί ολοκλήρου του κειμένου εξ ίσου, έχοντας δε αναφέρει στην Βιβλιογραφία όλες τις πηγές τις οποίες χρησιμοποίησε. Δηλώνει επίσης ότι, οποιοδήποτε στοιχείο ή κείμενο το οποίο έχει ενσωματώσει στην εργασία του προερχόμενο από βιβλία, άλλες εργασίες ή το διαδίκτυο, γραμμένο επακριβώς ή παραφρασμένο, το έχει πλήρως αναγνωρίσει ως πνευματικό έργο άλλου συγγραφέα και έχει αναφέρει ανελλιπώς το όνομά του και την πηγή προέλευσης.

Ο Φοιτητής:

Γεώργιος Καμνιάτσος

## **ΕΥΧΑΡΙΣΤΙΕΣ...**

*Με την ολοκλήρωση της παρούσας δηπλωματικής εργασίας, θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν στην ολοκλήρωσή της.*

*Ευχαριστώ ιδιαίτερω τον επιβλέπωντα καθηγητή κ. Σαγρή Δημήτριο για την επίβλεψη της εργασίας μου και τις πολύτιμες συμβουλές του καθ' όλη τη διάρκεια της διαδικασίας, καθώς η εμπειρία του και η εμπιστοσύνη του συνέβαλαν στην ανάπτυξη μου ως επιστήμονας και ως άνθρωπος.*

*Επίσης, θέλω να ευχαριστήσω και όλους τους καθηγητές του μεταπτυχιακού προγράμματος για την πληθώρα γνώσεων που μετέδωσαν κατά την διάρκεια αυτού, αλλά και για την υπομονή που έδειξαν ώστε μέσα σε ένα τόσο σύντομο χρονικό διάστημα καταφέραμε να φτάσουμε σε ένα αξιοπρεπές επίπεδο.*

*Επίσης, δεν μπορώ παρά μόνο να εκφράσω τις ευγνωμοσύνες μου προς την οικογένειά μου, για την συνεχή υποστήριξη και κατανόηση κατά την διάρκεια αυτής της πορείας.*

*Ευχαριστώ θερμά όλους τους φίλους, συμφοιτητές και συναδέλφους μηχανικούς καθώς και αυτοί συνέβαλαν με τον τρόπο τους.*

*Σέρρες, Απρίλιος 2024*

**Γεώργιος Καμνιάτσος**



## ΠΕΡΙΕΧΟΜΕΝΑ

|                                                               |    |
|---------------------------------------------------------------|----|
| 1. Εισαγωγή .....                                             | 7  |
| 2. Στάθμη γνώσεων .....                                       | 8  |
| 2.1 Ρομποτικά οχήματα .....                                   | 9  |
| 2.2. Οχήματα καλλωπισμού κήπου .....                          | 10 |
| 2.3. Ρομποτικά Οχήματα Κήπων .....                            | 13 |
| 3. Λογισμικά που χρησιμοποιήθηκαν .....                       | 17 |
| 3.1. Solidworks. ....                                         | 17 |
| 3.2. JavaScript .....                                         | 20 |
| 3.3. HTML .....                                               | 22 |
| 3.4. Περιβάλλον και λογισμικό smart συσκευής.....             | 25 |
| 3.5. Creality Slicer.....                                     | 30 |
| 4. Σκοπός εργασίας.....                                       | 31 |
| 5.Σχεδιασμός Hardware και Software οχήματος .....             | 32 |
| 5.1. Σχεδιασμός Hardware χειρισμού και επικοινωνίας.....      | 32 |
| 5.2. Σχεδιασμός software επικοινωνίας και χειρισμού .....     | 35 |
| 5.3. Συναρμολόγηση και έλεγχος κατασκευής .....               | 38 |
| 6. Σχεδιασμός διάταξης.....                                   | 39 |
| 6.1. Σχεδιασμός γεωμετρίας και εκλογή οχήματος εφαρμογής..... | 39 |
| 6.2. Μετατροπές οχήματος .....                                | 40 |
| 6.3. Σχεδίαση εξαρτημάτων με 3DPrinting .....                 | 50 |
| 7. Ανάπτυξη κώδικα .....                                      | 59 |
| 7.1. Κώδικας επικοινωνίας .....                               | 59 |
| 7.2. Κώδικας ελέγχου οχήματος.....                            | 60 |
| 7.3. Κώδικας χειρισμού και αυτοματισμών .....                 | 61 |
| 8. Ανάπτυξη παραδειγμάτων χρήσης .....                        | 63 |
| 9. Αξιολόγηση αποτελεσμάτων – Συμπεράσματα .....              | 64 |
| 10. Βιβλιογραφία .....                                        | 65 |
| 11. ΠΑΡΑΡΤΗΜΑ - Α (Σχέδια Εξαρτημάτων) .....                  | 67 |
| 11. ΠΑΡΑΡΤΗΜΑ - Β (Ηλεκτρολογικό Διάγραμμα) .....             | 70 |
| 11. ΠΑΡΑΡΤΗΜΑ - Γ (Κώδικας) .....                             | 71 |
| Σύντομο Βιογραφικό Σημείωμα.....                              | 84 |



# 1. Εισαγωγή

Από την εποχή της βιομηχανικής επανάστασης και έπειτα, άρχισε για τον άνθρωπο μια συνεχής εξέλιξη όσο αναφορά την τεχνολογία. Έτσι με την πάροδο των χρόνων ανακαλύφθηκαν τα ρομπότ.

Τι σημαίνει όμως ρομπότ;

Ρομπότ ονομάζεται οποιαδήποτε μηχανική συσκευή, μπορεί και υποκαθιστά τον άνθρωπο σε διάφορες εργασίες. Ένα ρομπότ μπορεί να δράσει κάτω από τον απευθείας έλεγχο ενός ανθρώπου ή αυτόνομα κάτω από τον έλεγχο ενός προγραμματισμένου υπολογιστή.

Τα ρομπότ μπορούν να χρησιμοποιηθούν ώστε να κάνουν εργασίες, οι οποίες είτε είναι δύσκολες είτε επικύνδινες για να γίνουν απευθείας από έναν άνθρωπο. Σε άλλες περιπτώσεις, χρησιμοποιούνται για να εκτελεστούν εργασίες ταχύτερα ή φθηνότερα απ' ό τι ο άνθρωπος. Έτσι μπορούν να χρησιμοποιηθούν στην αυτόματη παραγωγή μεγάλων ποσοτήτων κάποιου προϊόντος και με χαμηλότερο κόστος. (πχ. Στις παραγωγικές διαδικασίες όπου είναι τυποποιημένες.)

Η λέξη ρομπότ προέρχεται από την τσέχικη λέξη *robota* που σημαίνει εργασία. Καθιερώθηκε ως όρος με την σημερινή του έννοια το 1920 από τον Τσέχο θεατρικό συγγραφέα Κάρελ Τσάπεκ στο έργο του "R. U. R" (*Rossum's Universal Robots*), όπου σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ) της τεχνολογικής εξέλιξης και που τελικά εξοντώνουν τους δημιουργούς τους.

Ως αυτόνομο ρομπότ (*Autonomous Mobile Robot* ή *ARM*) ορίζεται ένα ρομποτικό σύστημα το οποίο μπορεί να πλοηγηθεί αυτόνομα σε ένα γνωστό ή ακόμα και άγνωστο περιβάλλον, ενώ πραγματοποιεί ταυτόχρονα διάφορες εργασίες για την επίτευξη ενός δεδομένου στόχου. Τα αυτόνομα ρομπότ διακρίνονται σε κατηγορίες, ανάλογα με τον χώρο εργασίας τους, σε αυτόνομα οχήματα εδάφους (*Autonomous Ground Vehicles* ή *AGVs*), νερού (*Autonomous Water/Underwater Vehicles* ή *AWV/AUV*) και αέρα (*Autonomous Aerial Vehicles* ή *AAVs*), ενώ παράλληλα υπάρχουν και αυτόνομα ανθρωποειδή ρομπότ (*Humanoid Robots*). Στην παρούσα διπλωματική εργασία εστιάζομαστε στα αυτόνομα ρομποτικά οχήματα εδάφους, με έμφαση στα ρομποτικά οχήματα που λειτουργούν όμοια με το συμβατικό αυτοκίνητο (*Car-like Robots*).

## 2. Στάθμη γνώσεων

Τα ρομποτικά οχήματα αντιπροσωπεύουν μια καινοτόμο εξέλιξη στον χώρο της τεχνολογίας, προσφέροντας αυτονομία και ακρίβεια σε διάφορους τομείς. Από τη γεωργία μέχρι τη βιομηχανία και την υγεία, αυτά τα οχήματα έχουν επαναπροσδιορίσει τους τρόπους παραγωγής, επιτρέποντας την αποδοτικότερη χρήση των πόρων και τη βελτίωση της απόδοσης σε πολλούς τομείς όπου αναλύονται παρακάτω.

Στην γεωργία, τα ρομποτικά οχήματα μπορούν να εκτελούν διάφορες εργασίες όπως τη σπορά, άρδευση, τη συλλογή καλλιεργειών και ακόμα και την εκτέλεση αγροτικών εργασιών. Αυτό μειώνει το κόστος και την ανάγκη για ανθρώπινη εργασία.

Στον τομέα των μεταφορών και των ταχυδρομικών υπηρεσιών, τα ρομποτικά οχήματα μπορούν να παρέχουν αυτόνομες και ασφαλείς μεταφορικές λύσεις.

Στη βιομηχανία, τα ρομποτικά οχήματα χρησιμοποιούνται για την αυτοματοποίηση και την εκτέλεση εργασιών σε περιβάλλον που μπορεί να είναι επικίνδυνα για τους ανθρώπους. Από τη μεταφορά εμπορευμάτων έως τη συντήρηση εγκαταστάσεων, τα ρομποτικά οχήματα εξασφαλίζουν ασφάλεια και αποδοτικότητα στις διαδικασίες παραγωγής.

Στον τομέα της υγείας, χρησιμοποιούνται για τη μεταφορά φαρμάκων, την παράδοση εξοπλισμού και την εκτέλεση χειρουργικών επεμβάσεων. Η ακρίβεια και η αξιοπιστία των ρομποτικών οχημάτων συμβάλλουν στη βελτίωση της ποιότητας της φροντίδας των ασθενών και τη μείωση του κινδύνου λανθασμένων διαδικασιών.

Με την ανάπτυξη πιο εξελιγμένων αισθητήρων και αλγορίθμων, καθώς και την ενσωμάτωση τεχνητής νοημοσύνης και μηχανικής μάθησης, το μέλλον των ρομποτικών οχημάτων φαίνεται λαμπρό. Συνολικά, αυτά τα οχήματα επιφέρουν σημαντικές αλλαγές σε πολλούς τομείς, βελτιώνοντας την αποτελεσματικότητα, την ασφάλεια στις διάφορες εφαρμογές τους.

Τα ρομποτικά οχήματα κήπου αποτελούν μια καινοτόμο λύση στη φροντίδα του κήπου, συνδυάζοντας την τεχνολογία με την ανάγκη για αποτελεσματική και βιώσιμη φροντίδα του περιβάλλοντος. Με πλεονεκτήματα όπως η αυτονομία, η ακρίβεια, ο βιώσιμος σχεδιασμός και η εξοικονόμηση πόρων, τα ρομποτικά οχήματα κάνουν «παιχνίδι» την καθημερινή φροντίδα του κήπου. Εφαρμόζονται σε διάφορους τομείς όπως η κοπή του γκαζόν, το αυτόματο πότισμα, η εφαρμογή λιπασμάτων και φυτοφαρμάκων, και η εξάλειψη άγριων φυτών. Παρά τις προκλήσεις όπως το κόστος και η αναγνώριση εμποδίων, οι μελλοντικές εξελίξεις προβλέπουν την ανάπτυξη πιο εξελιγμένων



τεχνολογιών για ακόμα βελτιωμένη απόδοση και λειτουργία. Συνολικά, τα ρομποτικά οχήματα κήπου συμβάλλουν προς τη διατήρηση του περιβάλλοντος και τη βελτίωση της ποιότητας ζωής.

## **2.1 Ρομποτικά οχήματα**

Η ρομποτική τεχνολογία έχει εξελιχθεί σημαντικά τα τελευταία χρόνια και έχει εφαρμογές σε πολλούς τομείς, όπως έχει ήδη αναφερθεί, στην βιομηχανία, στη γεωργία, στην αεροναυπηγική, στην εξερεύνηση του διαστήματος, στην ιατρική, κ.α.

Οι πρώτες αναφορές σε ρομποτικά οχήματα χρονολογούνται από τη δεκαετία του 1950, με την ανάπτυξη των πρώτων αυτόνομων ρομποτικών οχημάτων στα πλαίσια ερευνητικών προγραμμάτων στον τομέα της άμυνας και της εξερεύνησης του διαστήματος. Στη διάρκεια των δεκαετιών που ακολούθησαν, η τεχνολογία εξελίχθηκε σημαντικά, με την εμφάνιση ρομποτικών οχημάτων προορισμένα να επέμβουν όπου ήταν αδύνατο ή επικίνδυνο για τον άνθρωπο. Ανάλογα με την χρήση και τον σκοπό τα ρομποτικά οχήματα διακρίνονται σε διάφορες κατηγορίες.

Μία κατηγορία είναι τα αυτοκίνητα αυτόνομης οδήγησης. Αυτά τα οχήματα χρησιμοποιούνται για αυτόνομη μεταφορά επιβατών ή εμπορευμάτων. Παραδείγματα περιλαμβάνουν τα αυτόνομα οχήματα της Waymo και της Tesla.

Άλλου είδους οχημάτων αποτελούν τα υποβρύχια ρομπότ που χρησιμοποιούνται για εξερεύνηση και έρευνα στα υποβρύχια βάθη, καθώς και για εφαρμογές όπως η υποθαλάσσια επιδιόρθωση και η ανακάλυψη πόρων όπως τα ρομποτικά υποβρύχια της Bluefin Robotics.

Μία άλλη κατηγορία είναι τα αεροπλάνα UAV (Unmanned Aerial Vehicles) (πχ της DJI και της Lockheedartin) που χρησιμοποιούνται για αεροφωτογραφία, επιτήρηση, εξερεύνηση και στρατιωτικές εφαρμογές.

Επίσης μια πολύ σημαντική κατηγορία αποτελούν τα ρομποτικά οχήματα που χρησιμοποιούνται για εργασία σε επικίνδυνα περιβάλλοντα όπως η αποτέφρωση σε εργοστάσια, η καθαριότητα σε ραδιενεργά περιβάλλοντα και οι διαστημικές έρευνες κλπ. πλανήτες.

Τα τροχήλατα ρομποτικά οχήματα διακρίνονται σε 3 βασικές κατηγορίες.

Η 1<sup>η</sup> κατηγορία είναι τα δίτροχα ρομπότ όπου αποτελούν μια κατηγορία με μεγάλη ευελιξία και κινητικότητα. Μερικά παραδείγματα αποτελούν, τα ρομποτικά ποδήλατα, τα ηλεκτρικά σκούτερ, τα ρομποτικά πατίνια κ.α.

Η 2<sup>η</sup> κατηγορία περιλαμβάνει τα τετράτροχα ρομπότ τα οποία προσφέρουν μεγαλύτερη σταθερότητα και φορτηγότητα από τα δίτροχα. Τέτοια τετράτροχα ρομπότ είναι τα αυτόνομα ρομποτικά οχήματα, τα ρομποτικά αγροτικά οχήματα, τα στρατιωτικά κλπ. Η 3<sup>η</sup> κατηγορία είναι ρομπότ 6 τροχών, τα οποία είναι ιδανικά για αναπόσπαστες εργασίες και δύσκολες συνθήκες εδάφους. Σε αυτού του είδους της κατηγορίας των ρομπότ ανήκουν τα εξερευνητικά ρομποτικά οχήματα, τα πολυτροχικά ρομποτικά αγροτικά οχήματα, τα αυτόνομα ρομποτικά οχήματα κ.α.

Στην παρούσα διπλωματική εργασία θα κατασκευαστεί ένα τετράτροχο αυτόνομο ρομποτικό όχημα έχοντας ως στόχο την μελλοντική του εξέλιξη του για εργασίες στην γεωργία.

### **Σύντομη ιστορική αναδρομή των ρομποτικών οχημάτων.**

Το 1950, εμφανίστηκαν οι πρώτες αναφορές σε αυτόνομα ρομποτικά οχήματα, συνήθως σε πλαίσια ερευνητικών προγραμμάτων για την άμυνα και την εξερεύνηση του διαστήματος. Το 1961 Η NASA απέστειλε το πρώτο αυτόνομο ρομποτικό όχημα, το Rover, στο φεγγάρι, όπου έλαβε δεδομένα για το περιβάλλον της επιφάνειας του φεγγαριού. Κατά τη δεκαετία του 1970, εμφανίστηκαν τα πρώτα πειραματικά ρομποτικά οχήματα με δυνατότητα αυτόνομης οδήγησης χρησιμοποιώντας αισθητήρες και συστήματα ελέγχου αποστάσεων τα οποία αναπτύχθηκαν κατά την δεκαετία του 1990 με την χρήση προηγμένων αλγορίθμων και αισθητήρων.

Την δεκαετία του 1980 έκαναν την εμφάνιση τους τα πρώτα εμπορικά ρομποτικά αυτοκίνητα στην βιομηχανία. Η κύρια χρήση τους ήταν η μεταφορά εμπορευμάτων στα εργοστάσια.

Οι πρώτες εμπορικές εφαρμογές αυτόνομων οχημάτων εμφανίστηκαν στον τομέα της αυτοκινητοβιομηχανίας κατά την δεκαετία του 2000 με την εμπορική κυκλοφορία των οχημάτων όπου διαθέτουν προηγμένα συστήματα αυτόνομης οδήγησης.

## **2.2. Οχήματα καλλωπισμού κήπου**

Τα οχήματα καλλωπισμού κήπων είναι εξοπλισμένα οχήματα που χρησιμοποιούνται για τη διατήρηση και τον καλλωπισμό των κήπων και των πρασίνων χώρων. Αυτά τα οχήματα μπορεί να περιλαμβάνουν μικρά εργαλεία κήπου, όπως χλοοκοπτικά, θαμνοκοπτικά, κλαδευτικές μηχανές, φορτηγάκια μεταφοράς χώματος και φυτών, αλλά και πιο εξειδικευμένα οχήματα όπως ψεκαστικά, αεροβολιστές λιπάσματος και

απολιπαντικού, αναρτώμενα ή ενσωματωμένα συστήματα άρδευσης, και ακόμα και μηχανήματα για την ανακύκλωση των κλαδιών και του χλοοτάπητα.

Τα οχήματα αυτά συνήθως είναι σχεδιασμένα για να είναι ευέλικτα και να μπορούν να προσαρμοστούν σε διάφορες εργασίες και συνθήκες. Μερικά από αυτά μπορεί να έχουν εξοπλισμό για την ανακύκλωση απορριμμάτων ή τη διαχείριση του νερού, καθιστώντας φιλικά προς το περιβάλλον.

Οι εταιρίες κήπου και οι επαγγελματίες κηπουροί συχνά επενδύουν σε αυτά τα οχήματα για να αυξήσουν την αποτελεσματικότητα και την παραγωγικότητα στις εργασίες φροντίδας κήπων. Επιπλέον, η χρήση των οχημάτων καλλωπισμού κήπων μπορεί να βοηθήσει στη μείωση του χρόνου και της φυσικής εργασίας που απαιτείται για τη συντήρηση ενός κήπου ή ενός πρασίνου χώρου.

Τα τρακτεράκια είναι επίσης σημαντικά οχήματα καλλωπισμού κήπων και πράσινων χώρων. Αυτά τα μικρά τρακτεράκια συνήθως είναι εξοπλισμένα με διάφορα εργαλεία και εξαρτήματα που τους επιτρέπουν να εκτελούν ποικίλες εργασίες κήπου, όπως κοπή γκαζόν, αροτράρισμα, σπορά, άρδευση, καθαρισμός και άλλες εργασίες φροντίδας του εδάφους.

Τα τρακτεράκια είναι πολύ δημοφιλή στον τομέα του κήπου και της γεωργίας λόγω της πολυχρηστικότητάς τους και της ικανότητάς τους να εφαρμόζουν εργαλεία και εξαρτήματα για διάφορες εργασίες. Συχνά χρησιμοποιούνται σε μεγάλα καταστήματα, αγροκτήματα, δημόσιους χώρους και ιδιωτικούς κήπους για τη διαχείριση και τη φροντίδα των εκτάσεων. Το μέγεθος και οι δυνατότητες των τρακτέρ μπορεί να ποικίλουν ανάλογα με τις ανάγκες και τον προορισμό τους. Πιο συγκεκριμένα παρέχουν:

1. Πολυχρηστικότητα και ευελιξία, διότι μπορούν να εξυπηρετήσουν διάφορες ανάγκες καλλωπισμού και φροντίδας του κήπου. Με την κατάλληλη επιλογή εργαλείων και εξαρτημάτων, μπορούν και εκτελέσουν διάφορες εργασίες όπως κοπή γκαζόν, αροτράρισμα, σπορά, άρδευση και άλλες εργασίες φροντίδας.
2. Δημοφιλία στον κήπο και τη γεωργία διότι έχουν την ικανότητά να χρησιμοποιούνται για διάφορες εργασίες. Η ευκολία στην εφαρμογή εργαλείων τους τα καθιστούν απαραίτητα εργαλεία για τους επαγγελματίες και τους ερασιτέχνες.
3. Χρήση σε διάφορα περιβάλλοντα όπως σε μεγάλες κατοικίες, αγροκτήματα, δημόσιους χώρους και άλλα περιβάλλοντα. Η δυνατότητα προσαρμογής τους σε διάφορες ανάγκες κάνει τα τρακτεράκια ιδανικά για διαφορετικά είδη εργασιών και χρήσεων.

4. Ποικιλία μεγεθών και δυνατοτήτων ανάλογα με τις ανάγκες και τον προορισμό τους. Αυτό επιτρέπει στους χρήστες να επιλέξουν το κατάλληλο μοντέλο που θα ανταποκρίνεται στις ειδικές απαιτήσεις τους.
5. Τεχνολογικές καινοτομίες με ενσωματωμένα συστήματα GPS, αυτοματοποιημένες λειτουργίες και άλλες καινοτομίες που βελτιώνουν την απόδοση και την ακρίβεια των εργασιών.
6. Βιωτικότητα και περιβαλλοντική συνείδηση μια τάση συνεχώς αυξανόμενη. Οι κατασκευαστές εργάζονται προς την κατεύθυνση της μείωσης των εκπομπών και της ενεργειακής αποδοτικότητας στα τρακτεράκια.
7. Εκπαίδευση και επιμόρφωση: όντας σημαντικές για την ασφάλεια και την αποτελεσματικότητα στη χρήση των τρακτέρ. Προγράμματα εκπαίδευσης σχετικά με τη λειτουργία και τη συντήρηση μπορούν να βοηθήσουν στην επίτευξη αυτών των στόχων.
8. Οικονομικές πτυχές καθώς το κόστος απόκτησης, λειτουργίας και συντήρησης των τρακτέρ είναι σημαντικά θέματα που επηρεάζουν τις αποφάσεις αγοράς. Η αξιολόγηση της οικονομικής απόδοσης και της αποτελεσματικότητας των τρακτέρ είναι σημαντική για τους επαγγελματίες και τους ιδιώτες που ενδιαφέρονται για την αγορά τους.

Σε αυτά τα οχήματα καλλωπισμού κήπου εφαρμόζονται επίσης και τεχνολογίες όπως η αυτοματοποίηση των τρακτέρ και οχημάτων καλλωπισμού κήπων μέσω ρομποτικής τεχνολογίας. Τα οχήματα αυτά μπορούν να εφοδιαστούν με αισθητήρες και συστήματα πλοήγησης που τους επιτρέπουν να λειτουργούν αυτόνομα ή ημιαυτόνομα, αναλαμβάνοντας εργασίες καλλωπισμού και φροντίδας χωρίς τη συνεχή παρέμβαση ανθρώπινου προσωπικού.

Άλλη τεχνολογία που εφαρμόζεται είναι η συνδεσιμότητα και δικτύωση καθώς επιτρέπουν την σύνδεση των τρακτέρ και των οχημάτων καλλωπισμού κήπων με διάφορες πλατφόρμες διαχείρισης και διασύνδεσης δεδομένων. Αυτό μπορεί να βελτιώσει τη διαχείριση των εργασιών, την αποδοτικότητα και την παρακολούθηση της απόδοσης.

Σημαντική τεχνολογία αποτελεί επίσης η τεχνολογία λογισμικού ελέγχου. Μέσω αυτής της τεχνολογίας τα τρακτεράκια και τα οχήματα καλλωπισμού κήπων μπορούν να εφοδιαστούν με λογισμικό ελέγχου που επιτρέπει την προγραμματισμένη εκτέλεση εργασιών, την ανίχνευση προβλημάτων και την παρακολούθηση των επιδόσεων. Η τεχνολογία ελέγχου ενσωματώνει την τεχνολογία των αισθητήρων και των επιταχυνσιόμετρων επειδή χρησιμοποιούνται για τη μέτρηση του εδάφους, την

παρακολούθηση της υγρασίας, την ανίχνευση ανωμαλιών και άλλων παραμέτρων που μπορούν να επηρεάσουν την απόδοση του οχήματος.

Όλες αυτές οι τεχνολογικές εξελίξεις επιδρούν θετικά στην απόδοση, την ασφάλεια και την αποτελεσματικότητα των τρακτέρ και των οχημάτων καλλωπισμού κήπων, επιτρέποντάς τους να ανταποκρίνονται καλύτερα στις σύγχρονες ανάγκες της γεωργίας και του κήπου.

### **2.3. Ρομποτικά Οχήματα Κήπων**

Τα ρομποτικά οχήματα κήπων έχουν φέρει την επανάσταση στην φροντίδα του κήπου. Η τεχνολογία έχει εισβάλλει σε κάθε πτυχή της ζωής μας, αλλά η παρέμβασή της στον κόσμο της αγροτικής εργασίας και της φύσης έχει επαναπροσδιορίσει τον τρόπο με τον οποίο διατηρούμε και φροντίζουμε τους κήπους μας. Τα ρομποτικά οχήματα κήπων είναι μια καινοτόμος λύση που συνδυάζει την τεχνολογία με την ανάγκη για αποτελεσματική και βιώσιμη φροντίδα του περιβάλλοντος.

#### **Εισαγωγή στα Ρομποτικά Οχήματα Κήπων**

Τα ρομποτικά οχήματα κήπων αποτελούν μια εξέλιξη στην τεχνολογία των γεωργικών μηχανημάτων. Αντί να βασίζονται στην ανθρώπινη εργασία ή στις παραδοσιακές βενζινοκίνητες μηχανές, αυτά τα οχήματα λειτουργούν εντελώς αυτόνομα ή με ελάχιστη ανθρώπινη επίβλεψη προσφέροντας μια σειρά από πλεονεκτήματα:

##### **1. Αυτονομία:**

Τα ρομποτικά οχήματα κήπων μπορούν να λειτουργούν αυτόνομα για αρκετές ώρες, εξοικονομώντας χρόνο και ενέργεια στους χρήστες τους.

##### **2. Ακρίβεια:**

Εξοπλισμένα με προηγμένους αλγόριθμους και αισθητήρες, τα ρομποτικά οχήματα μπορούν να εκτελούν εργασίες με μεγάλη ακρίβεια, χωρίς να αφήνουν περιθώρια λάθους.

### 3. Βιώσιμος Σχεδιασμός:

Πολλά ρομποτικά οχήματα κήπων λειτουργούν με ηλεκτρική ενέργεια ή ακόμα και με φωτοβολταϊκούς συλλέκτες, μειώνοντας την εξάρτηση από τους ορυκτούς πόρους και τις εκπομπές αερίων του θερμοκηπίου.

### 4. Εξοικονόμηση Πόρων:

Λόγω της ακρίβειάς τους, τα ρομποτικά οχήματα καταναλώνουν λιγότερους πόρους όπως νερό, λιπάσματα και φυτοφάρμακα, βοηθώντας στην εξοικονόμηση και την αειφορία του περιβάλλοντος.

## **Εφαρμογή**

Τα ρομποτικά οχήματα κήπου βρίσκουν εφαρμογή σε ποικίλους τομείς, συμβάλλοντας στην βελτίωση της ποιότητας του περιβάλλοντος και στην εξοικονόμηση χρόνου και πόρων των χρηστών τους. Μερικούς από αυτούς τους τομείς περιγράφονται παρακάτω:

#### 1. Κοπή Γκαζόν:

Τα ρομποτικά οχήματα μπορούν να εκτελούν ακριβείς και τακτικές κοπές γκαζόν, χωρίς την ανάγκη για ανθρώπινη παρέμβαση.

#### 2. Αυτόματο Πότισμα:

Ορισμένα μοντέλα είναι εξοπλισμένα με συστήματα αυτόματου ποτίσματος που εντοπίζουν τις περιοχές που χρειάζονται περισσότερο νερό και τις ποτίζουν ανάλογα.

#### 3. Εφαρμογή Λιπασμάτων και Φυτοφαρμάκων:

Με τη χρήση προγραμματιζόμενων συστημάτων, τα ρομποτικά οχήματα μπορούν να διανέμουν λιπάσματα και φυτοφάρμακα με ακρίβεια και αποφεύγοντας την υπερβολική χρήση ουσιών.

#### 4. Εξάλειψη Άγριων Φυτών:

Μέσω ειδικών αισθητήρων και προγραμματιζόμενων αλγορίθμων, τα ρομποτικά οχήματα μπορούν να ανιχνεύουν και να εξαλείφουν άγρια φυτά από τον κήπο.

#### 5. Προγραμματιζόμενες Εργασίες:

Τα ρομποτικά οχήματα κήπου είναι συχνά εξοπλισμένα με λογισμικό που επιτρέπει στους χρήστες να προγραμματίσουν διάφορες εργασίες, όπως το πότισμα, η κοπή γκαζόν, η εφαρμογή λιπασμάτων και φυτοφαρμάκων, σε συγκεκριμένες χρονικές στιγμές και περιοχές του κήπου τους.

#### 6. Ενσωματωμένη Τεχνολογία IoT (Διαδίκτυο των Πραγμάτων):

Ορισμένα ρομποτικά οχήματα κήπου είναι συνδεδεμένα στο διαδίκτυο και μπορούν να επικοινωνούν με τους χρήστες τους μέσω εφαρμογών κινητών τηλεφώνων. Αυτή η ενσωματωμένη τεχνολογία επιτρέπει στους χρήστες να ελέγχουν και να παρακολουθούν τις εργασίες του κήπου τους ακόμη και από μακρινή απόσταση.

### **Προκλήσεις και Μελλοντικές Εξελίξεις**

Παρά τα πολλά πλεονεκτήματα των ρομποτικών οχημάτων κήπου, υπάρχουν και μερικές προκλήσεις που καλούνται να αντιμετωπίσουν. Μερικές από αυτές τις προκλήσεις παρουσιάζονται παρακάτω.

#### 1. Κόστος:

Παρόλο που η τεχνολογία αυτή βελτιώνεται συνεχώς, τα ρομποτικά οχήματα κήπου μπορεί να είναι ακόμα αρκετά ακριβά για πολλούς καταναλωτές.

#### 2. Αναγνώριση Και Αντιμετώπιση Εμποδίων:

Ακόμα και με τη χρήση προηγμένων αισθητήρων, τα ρομποτικά οχήματα κήπου μπορεί να αντιμετωπίζουν προκλήσεις στο να αναγνωρίζουν και να αντιμετωπίζουν εμπόδια, όπως πέτρες ή άλλα αντικείμενα που ενδέχεται να είναι στο δρόμο τους.

#### 3. Απόδοση σε Ακραίες Συνθήκες:

Σε περιοχές με ακραίες καιρικές συνθήκες, όπως έντονες βροχοπτώσεις ή υψηλές θερμοκρασίες, η απόδοση των ρομποτικών οχημάτων κήπου μπορεί να μειωθεί.

Μελλοντικές εξελίξεις στον τομέα των ρομποτικών οχημάτων κήπου αποτελούν η ανάπτυξη πιο εξελιγμένων αισθητήρων και αλγορίθμων επεξεργασίας δεδομένων, καθώς και η ενσωμάτωση τεχνητής νοημοσύνης και μηχανικής μάθησης για βελτιωμένη απόδοση και λειτουργία.

### **Συμπεράσματα**

Τα ρομποτικά οχήματα κήπων αντιπροσωπεύουν μια εξέλιξη στον κόσμο της γεωργίας και του κήπου. Με τη συνδυασμένη χρήση τεχνολογίας και βιώσιμων πρακτικών, μπορούν να συμβάλουν στην διατήρηση του περιβάλλοντος και στην βελτίωση της ποιότητας ζωής.



### 3. Λογισμικά που χρησιμοποιήθηκαν

#### 3.1. Solidworks.

Το SolidWorks είναι ένα από τα πιο δημοφιλή λογισμικά σχεδίασης στην κατηγορία της τρισδιάστατης μηχανικής σχεδίασης (CAD), αλλά χρησιμοποιείται επίσης για την προσομοίωση, τη μηχανική ανάλυση, και τη δημιουργία τεχνικής τεκμηρίωσης. Η ιστορία του SolidWorks είναι ενδιαφέρουσα και σχετίζεται με την εξέλιξη της τεχνολογίας CAD.

#### Η Ιστορία του SolidWorks:

Το SolidWorks δημιουργήθηκε από την εταιρεία Solid Works Corporation, η οποία ιδρύθηκε το 1993 από τους Jon Hirschtick, Scott Harris, Dave Corcoran, και John Mc Elenev. Ο στόχος της εταιρείας ήταν να δημιουργήσει ένα CAD λογισμικό που θα ήταν εύκολο στη χρήση, προσιτό και θα επέτρεπε στους μη μηχανικούς να κάνουν τρισδιάστατη σχεδίαση με ευκολία.

Το πρώτο λειτουργικό πρόγραμμα SolidWorks κυκλοφόρησε το 1995 και ήταν ένας πρωτοποριακός συνδυασμός γραφικής διεπαφής με βάση το παράδειγμα του γραφικού περιβάλλοντος των Windows. Αυτό έκανε το SolidWorks ευκολόχρηστο και προσιτό σε νέους χρήστες.

Από τότε, το SolidWorks έχει υποστεί πολλές αναβαθμίσεις και επεκτάσεις, προσθέτοντας σημαντικές λειτουργίες όπως η προσομοίωση, η μηχανική ανάλυση, η εξαγωγή δεδομένων για παραγωγή και πολλά άλλα.

#### Τι είναι το SolidWorks και ποιος το χρησιμοποιεί:

Το SolidWorks είναι ένα πρόγραμμα CAD που χρησιμοποιείται για τη σχεδίαση τρισδιάστατων μοντέλων. Είναι δημοφιλές μεταξύ μηχανικών, αρχιτεκτόνων, σχεδιαστών προϊόντων και άλλων επαγγελματιών που ασχολούνται με τη σχεδίαση.

Το SolidWorks διαθέτει μια ευέλικτη και εύχρηστη διεπαφή που επιτρέπει στους χρήστες να δημιουργούν πολύπλοκα τρισδιάστατα μοντέλα με ευκολία. Οι χρήστες μπορούν να δημιουργήσουν αντικείμενα από το μηδέν ή να εισαγάγουν δεδομένα από άλλα συστήματα CAD για περαιτέρω επεξεργασία.

Οι βασικές λειτουργίες του SolidWorks περιλαμβάνουν τη δημιουργία τρισδιάστατων μοντέλων, τη δημιουργία τεχνικών σχεδίων και εγγράφων, την προσομοίωση και ανάλυση μηχανικών χαρακτηριστικών, την αποθήκευση και τη διαχείριση των σχεδίων και πολλά άλλα.

Συνοψίζοντας, το SolidWorks είναι ένα ισχυρό εργαλείο CAD που χρησιμοποιείται για τη δημιουργία και την ανάπτυξη τρισδιάστατων μοντέλων σε πολλούς τομείς, από τη μηχανολογία μέχρι την αρχιτεκτονική και τον σχεδιασμό προϊόντων.

Το SolidWorks χρησιμοποιεί μια σειρά τεχνολογιών για τη δημιουργία και τη διαχείριση των τρισδιάστατων μοντέλων και των σχετικών δεδομένων. Αυτές οι τεχνολογίες περιλαμβάνουν:

#### 1. Πυρήνας Γεωμετρίας (Geometry Kernel):

Ο πυρήνας γεωμετρίας είναι ο κεντρικός μηχανισμός που χρησιμοποιείται για τη δημιουργία και τη διαχείριση των γεωμετρικών σχημάτων στο SolidWorks. Αυτός ο πυρήνας υποστηρίζει διάφορες γεωμετρικές λειτουργίες, όπως επιφανειακή και στερεομετρική μοντελοποίηση, επεξεργασία και αλληλεπίδραση με τα στοιχεία του μοντέλου.

#### 2. Προσομοίωση και Ανάλυση:

Το SolidWorks ενσωματώνει τεχνολογίες προσομοίωσης και ανάλυσης που επιτρέπουν στους χρήστες να εκτελούν μηχανικές αναλύσεις για την αξιολόγηση των σχεδίων τους. Αυτές οι τεχνολογίες περιλαμβάνουν προσομοιώσεις αντοχής, ροής υγρών, θερμοκρασίας και άλλων μηχανικών χαρακτηριστικών.

#### 3. Επεξεργασία Κορυφαίων (Topological Editing):

Αυτή η τεχνολογία επιτρέπει στους χρήστες να επεξεργάζονται τη γεωμετρία του μοντέλου τους χωρίς να χρειάζεται να αντιμετωπίζουν περιορισμούς από την αρχική δομή του μοντέλου.

#### 4. Διαχείριση Δεδομένων (Data Management):

Το SolidWorks προσφέρει λειτουργίες διαχείρισης δεδομένων που επιτρέπουν στους χρήστες να οργανώνουν, να αναζητούν και να διαμοιράζονται τα σχέδιά τους με άλλους μέσω ενός κεντρικού συστήματος αποθήκευσης.

Συνολικά, οι προηγμένες τεχνολογίες που χρησιμοποιεί το SolidWorks επιτρέπουν στους χρήστες να δημιουργούν, να επεξεργάζονται και να αξιολογούν σύνθετα τρισδιάστατα μοντέλα και σχέδια.

Το SolidWorks έχει πολλά πλεονεκτήματα που το καθιστούν ένα ισχυρό εργαλείο για την τρισδιάστατη μηχανική σχεδίαση, αλλά και μερικά μειονεκτήματα που πρέπει να ληφθούν υπόψη.

## **Πλεονεκτήματα:**

### **1. Ευκολία Χρήσης**

Διαθέτει μια φιλική προς τον χρήστη διεπαφή που καθιστά εύκολη τη δημιουργία τρισδιάστατων μοντέλων ακόμη και για αρχάριους χρήστες.

### **2. Εκτεταμένες Λειτουργίες:**

Προσφέρει εκτεταμένες λειτουργίες για την προσομοίωση, την ανάλυση, τη δημιουργία τεχνικής τεκμηρίωσης και πολλά άλλα, καθιστώντας το κατάλληλο για μια ευρεία γκάμα εφαρμογών.

### **3. Ολοκληρωμένη Διαχείριση Δεδομένων:**

Παρέχει εργαλεία διαχείρισης δεδομένων που επιτρέπουν την οργάνωση και τη διαμοίραση των σχεδίων μεταξύ των μελών της ομάδας.

### **4. Ευελιξία:**

Είναι ευέλικτο και επιτρέπει την εύκολη προσαρμογή των μοντέλων και των σχεδίων ανάλογα με τις απαιτήσεις του έργου.

## **Μειονεκτήματα:**

### **1. Κόστος:**

Είναι ένα εμπορικό λογισμικό και μπορεί να είναι ακριβό για ατομικούς χρήστες ή μικρές επιχειρήσεις.

### **2. Απαιτήσεις Υπολογιστικού Εξοπλισμού:**

Η εκτέλεση του SolidWorks μπορεί να απαιτεί αρκετούς πόρους υπολογιστή, όπως ισχυρό επεξεργαστή και μνήμη RAM.

### **3. Εκμάθηση:**

Λόγω της πολυπλοκότητάς του, η εκμάθηση του SolidWorks μπορεί να απαιτεί χρόνο και προσήλωση.

Συνολικά, το SolidWorks προσφέρει μια πληθώρα πλεονεκτημάτων για την τρισδιάστατη μηχανική σχεδίαση, αλλά πρέπει να ληφθούν υπόψη τα μειονεκτήματα του κατά την επιλογή και τη χρήση του.



Σχήμα 1: Λογότυπο του Solidworks

### **3.2. JavaScript**

Η **JavaScript (JS)** είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-sidescripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JS είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Επίσης χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων. Τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specificbrowsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktopwidgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JS (όπως το Node.js) έχουν επίσης κάνει τη JS πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται **ECMAScript**.

## Ιστορία της JavaScript

Η γλώσσα προγραμματισμού JS δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, η Mocha μετονομάστηκε σε **LiveScript**, και τελικά σε **JavaScript**, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. Το LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator έκδοση 2.0 τον Σεπτέμβριο του 1995. Η LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape 2.0B3. Η **JScript** απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφάρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. Η JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java. Η JScript περιλήφθηκε στο πρόγραμμα InternetExplorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο **EcmaInternational** (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript. Η JS έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (μαζί με άλλους λόγους). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side).

## Javascript και Java

Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης "Java" στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστρεφείς γλώσσες. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι "**JavaScript**" και όχι 'Java script' σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.



Σχήμα 2: Λογότυπο της Javascript

### **3.3. HTML**

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός περιηγητή ιστού (Webbrowser) είναι να διαβάζει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το

περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα.

Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JS, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML και από στατικές τις κάνουν διαδραστικές.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Η Κοινοπραξία του Παγκοσμίου Ιστού, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

### **Ιστορία της HTML**

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN. Στις προσωπικές του σημειώσεις από το 1990, ο Μπέρνερς Λι αριθμεί «μερικές από τις πολλές χρήσεις του υπερκειμένου», όπως την γενική παρουσίαση πληροφοριών (π.χ. μια εγκυκλοπαίδεια), στοχευμένη δημοσίευση (κείμενα βοήθειας, τεκμηρίωσης, εκπαίδευσης, κλπ) μέχρι ακόμα και για καταγραφή προσωπικών σημειώσεων.

### **Πρώτες προδιαγραφές**

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGML guid, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML 4.

Το ίδιο το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τυπογράφων, αλλά εκτός από απλή μίμηση της τυπογραφίας προσθέτει γενικευμένη σήμανση βασισμένη σε στοιχεία, τα οποία μπορούν να εμφωλεύονται το ένα μέσα στο άλλο και να φέρουν ιδιότητες. Ακόμα, το SGML διαχωρίζει τη δομή από το περιεχόμενο, κατεύθυνση προς την οποία αργότερα κινήθηκε και η HTML, με τα CSS. Πολλά από τα στοιχεία κειμένου προέρχονται από την τεχνική αναφορά ISO 9537, Techniques for using SGML (τεχνικές χρήσης της SGML), η οποία με τη σειρά της καλύπτει τα χαρακτηριστικά των πρώιμων γλωσσών μορφοποίησης κειμένου που χρησιμοποιούνταν από τα TYPSET και RUNOFF, και είχαν αναπτυχθεί στις αρχές της δεκαετίας του 1960 για το λειτουργικό σύστημα CTSS.

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993, η οποία περιλάμβανε και έναν ορισμό τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το ανταγωνιστικό πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.

Μετά την λήξη των προχειρών HTML και HTML+, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασίζονταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Η αριθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν.

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης απόψεων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με την σύμφωνη γνώμη των δημιουργών του λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).



## Σήμανση

- Η σήμανση HTML αποτελείται από μερικά βασικά συστατικά, συμπεριλαμβανομένων των στοιχείων (και των ιδιοτήτων τους), βασισμένους σε χαρακτήρες τύπου δεδομένων, τις αναφορές χαρακτήρων και τις αναφορές οντοτήτων. Ένα ξεχωριστό σημαντικό συστατικό είναι η δήλωση τύπου εγγράφου (document type declaration), η οποία ορίζει στο πρόγραμμα περιήγησης (browser) τον τρόπο εμφάνισης της σελίδας.
- Στην HTML, το πρόγραμμα HelloWorld, ένα συνηθισμένο πρόγραμμα υπολογιστή που χρησιμεύει για τη σύγκριση γλωσσών προγραμματισμού, γλωσσών σεναρίων και γλωσσών σήμανσης, φτιάχνεται με 9 γραμμές κώδικα, παρότι οι νέες γραμμές είναι προαιρετικές στην HTML.

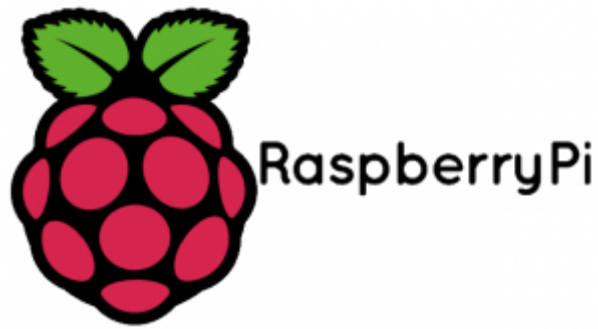
### **3.4. Περιβάλλον και λογισμικό smart συσκευής**

#### Τι είναι το Raspberry Pi



Σχήμα 3: Σκίτσο Rpi.

Το Raspberry Pi είναι ένας πλήρης υπολογιστής με μέγεθος πιστωτικής κάρτας. Αν και μικρό σε όγκο, το RaspberryPi διαθέτει τετραπύρρηνο επεξεργαστή 1200MHz, διπύρρηνη κάρτα γραφικών, 1GBRAM, τέσσερις θύρες USB, έξοδο HDMI, τροφοδοτείται μέσω Micro USB, και 40 pins γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά.



Σχήμα 4: Λογότυπο Raspberry Pi

### **Πώς ξεκίνησε το Raspberry Pi**

Το 2006, στο πανεπιστήμιο του Cambridge, μια ομάδα εργαζομένων στο τμήμα Computer Laboratory ανησυχούσε για το μειωμένο ενδιαφέρον των φοιτητών στην πληροφορική, καθώς και τις περιορισμένες γνώσεις τους.

Η ομάδα αυτή, που περιλάμβανε μεταξύ άλλων τους EbenUpton, RobMullins, JackLang, και AlanMycroft, σκέφτηκε πως η λύση θα ήταν ένας εξαιρετικά μικρός και προσιτός υπολογιστής.

Με ένα τέτοιο οικονομικό σύστημα, θα ήταν δυνατόν να διδάσκεται πρακτικά η πληροφορική στα σχολεία, και να έχει μεγαλύτερο ενδιαφέρον για τους μαθητές.

Ενώ η ομάδα σχεδίασε αρκετά αρχικά πρωτότυπα του RaspberryPi, πρέπει να αναλογιστούμε πως το 2006 υπήρχαν σημαντικοί περιορισμοί, λόγω του υψηλού κόστους και χαμηλής ισχύος των επεξεργαστών για mobile συσκευές.

Σταδιακά, όμως, με την κυκλοφορία του πρώτου iPhone το 2007 και την μετέπειτα επέλαση των smartphones, το κόστος της τεχνολογίας άρχισε να μειώνεται αρκετά για να γίνει βιώσιμη η υλοποίηση του RaspberryPi.

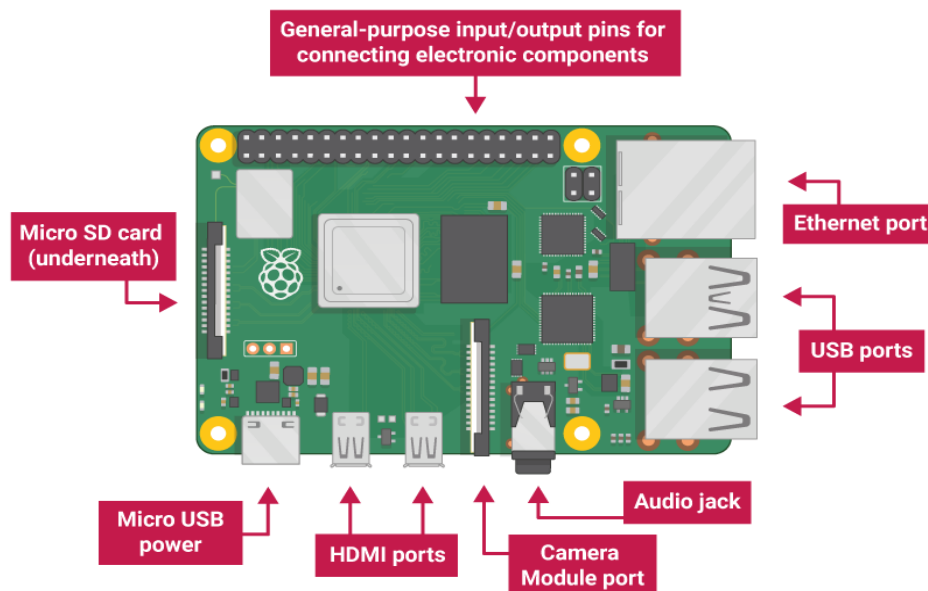
Το 2008, με τη συνεργασία των PeteLomas και DavidBraben δημιουργήθηκε το φιλανθρωπικό ίδρυμα RaspberryPiFoundation. Τρία χρόνια αργότερα κυκλοφόρησε το πρώτο RaspberryPi, με τα ModelA, ModelA+ και Model B.



Σχήμα 5: Πλακέτα Raspberry Pi.

Τα μοντέλα αυτά διέθεταν επεξεργαστή ARMv6k στα 700 MHz, 256MB RAM, κάρταγραφικών Broadcom Video Core IV, και κατανάλωση από 1 έως 3.5 watt, ενώ η αποθήκευση των δεδομένων γινόταν σε κάρτες SD, SD HC και Micro SD.

Μόνο το Raspberry Pi Model B, μέσα σε δύο χρόνια από όταν βγήκε στη μαζική παραγωγή, πούλησε πάνω από δύο εκατομμύρια κομμάτια.



Σχήμα 6: Raspberry Pi 3 – Επεξήγηση εξαρτημάτων

Στη συνέχεια κυκλοφόρησαν οι εκδόσεις Model B rev 2 και Model B+ με 512MB RAM. Τον Οκτώβριο του 2014, οι συνολικές πωλήσεις προσέγγιζαν τα 4 εκατομμύρια.

Το Φεβρουάριο του 2015 κυκλοφόρησε το RaspberryPiGeneration 2 Model B, ανέβασε τη RAM στο 1GB και εξαπλασίασε την ταχύτητα του επεξεργαστή, με τον τετραπύρηνο Cortex-A7 (ARMv7), ενώ έχει διπύρηνη κάρτα γραφικών Broad com Video Core IV.

Το Νοέμβριο του 2015 κυκλοφόρησε το RaspberryPiZero, μία έκδοση με το μισό μέγεθος από το RaspberryPi, 512MBRAM, και επεξεργαστή ARM1176JZF-S στα 1000MHz.



Σχήμα 7: Raspberry Pi Zero

Το Raspberry Pi Generation 3 Model B κυκλοφόρησε τον Φεβρουάριο του 2016. Έρχεται εξοπλισμένο με ακόμα ταχύτερο επεξεργαστή ARMCortex-A53 στα 1200MHz, 1GBRAM και κάρτα γραφικών BroadcomVideoCoreIV χρονισμένη στα 250MHz, συχνότητα υψηλότερη από κάθε προηγούμενη γενιά.

### **Τι μπορεί να κάνει κάποιος με το RaspberryPi**

Ενώ είναι εντυπωσιακό πώς αυτή η υπολογιστική ισχύς συγκεντρώνεται σε τόσο λίγο χώρο και με τόσο χαμηλό κόστος – σημαντικά χαμηλότερο από ενός smartphone – το ερώτημα είναι το τι μπορεί να κάνει κάποιος με το RaspberryPi.

Όπως αποδεικνύεται, με μεράκι και φαντασία οι εφαρμογές του RaspberryPi είναι πρακτικά απεριόριστες.

Το RaspberryPi μπορεί να χρησιμοποιήσει διάφορα λειτουργικά συστήματα, ανάλογα με τις ανάγκες και τις προτιμήσεις του χρήστη. Τα δημοφιλέστερα λειτουργικά συστήματα που χρησιμοποιούνται στο RaspberryPi είναι τα εξής:

### 1. Raspberry Pi OS (προηγουμένως Raspbian):

Το Raspberry Pi OS είναι ένα ειδικά διαμορφωμένο λειτουργικό σύστημα βασισμένο στο Debian, σχεδιασμένο για το RaspberryPi. Είναι προεγκατεστημένο με πολλά προγράμματα και εργαλεία για αρχάριους και προχωρημένους χρήστες.

### 2. Ubuntu:

Η έκδοση του Ubuntu για το RaspberryPi προσφέρει μια γνώριμη εμπειρία χρήσης Linux, καθώς και πρόσβαση σε μια ευρεία γκάμα πακέτων λογισμικού που υποστηρίζονται από την κοινότητα του Ubuntu.

3. ArchLinux: Το ArchLinux είναι ένα ελαφρύ και ευέλικτο λειτουργικό σύστημα που στοχεύει σε προχωρημένους χρήστες. Είναι γνωστό για την απλότητά του και τη συνεχή ενημέρωση του λογισμικού.

Επιπλέον, το RaspberryPi μπορεί να εκτελέσει και άλλα λειτουργικά συστήματα όπως το Fedora, το Gentoo, το Open SUSE και πολλά άλλα, ανάλογα με τις ανάγκες του χρήστη και την υποστήριξη της κοινότητας.

## **Χαρακτηριστικά και Λειτουργίες του Rpi.**

### 1. Προεγκατεστημένα Προγράμματα:

Το RaspberryPiOS έρχεται με προεγκατεστημένα προγράμματα όπως περιηγητής ιστού, εφαρμογές γραφείου, προγραμματιστικά εργαλεία και άλλα, προσφέροντας μια πλήρη εμπειρία του χρήστη από την αρχή.

### 2. Περιβάλλον Εργασίας:

Το RaspberryPiOS χρησιμοποιεί τον γραφικό περιβάλλον εργασίας LXDE (Light weight X11 Desktop Environment) που είναι ελαφρύ και αποδοτικό, κατάλληλο για τους περιορισμένους πόρους του RaspberryPi.

### 3. Εργαλεία Προγραμματισμού:

Το λειτουργικό σύστημα περιλαμβάνει πολλά εργαλεία προγραμματισμού και ανάπτυξης λογισμικού, καθιστώντας το ιδανικό για αρχάριους και προχωρημένους προγραμματιστές.

#### 4.Υποστήριξη Κοινότητας:

Το Raspberry Pi OS έχει ένα ενεργό κοινοτικό οικοσύστημα που παρέχει υποστήριξη, αναβαθμίσεις και εφαρμογές από την κοινότητα των χρηστών και προγραμματιστών.

#### **Χρήσεις**

Το RaspberryPiOS είναι ιδανικό για διάφορες χρήσεις, συμπεριλαμβανομένων:

- Εκπαιδευτικά προγράμματα και πειράματα.
- Δημιουργία προσωπικού υπολογιστή ή mediacenter.
- Ανάπτυξη έργων IoT (InternetofThings) και έξυπνων συσκευών.
- Αυτοματισμός σπιτιού και ελέγχους.

Το Raspberry Pi OS παρέχει μια πλήρη και ευέλικτη πλατφόρμα για να εξερευνήσετε τις δυνατότητες του RaspberryPi και να δημιουργήσετε διάφορα έργα και εφαρμογές.

*\*Το Raspberry Pi που έχει χρησιμοποιηθεί στην κατασκευή είναι το Rpi Model B 4GB με το λογισμικό **Bookworm 64bit (Full)**.*

### **3.5. Creality Slicer**

Το CrealitySlicer είναι ένα λογισμικό για τον υπολογιστή που χρησιμοποιείται για την προετοιμασία των αρχείων για εκτύπωση 3D με εκτυπωτές της εταιρείας Creality. Η Creality είναι γνωστή για την παραγωγή ευρέως χρησιμοποιούμενων εκτυπωτών 3D, και το Creality Slicer είναι ένα λογισμικό που παρέχεται για την ευκολία των χρηστών της στην προετοιμασία των αρχείων εκτύπωσης.

Το Creality Slicer προσφέρει λειτουργίες όπως το στήσιμο (slicing) των 3D μοντέλων σε στρώσεις για εκτύπωση, την προσαρμογή των ρυθμίσεων εκτύπωσης (όπως οι ταχύτητες, οι θερμοκρασίες και άλλα), καθώς και τη διαχείριση των αρχείων και των εκτυπωτικών προφίλ. Το λογισμικό συνήθως προσφέρει μια ευρεία γκάμα επιλογών για την προσαρμογή των ρυθμίσεων εκτύπωσης προσαρμοσμένα στις ανάγκες του χρήστη, ενώ παράλληλα παρέχει εύκολο στη χρήση περιβάλλον.

Συνολικά, το Creality Slicer είναι ένα εργαλείο που βοηθάει τους χρήστες των εκτυπωτών 3D της Creality να προετοιμάσουν και να διαχειριστούν τα αρχεία εκτύπωσης τους με ευκολία και αποτελεσματικότητα. Ο χρήστης περνάει το 3D σχέδια που έχει ετοιμάσει με την μορφή STI, και αφού προετοιμάσει την διαδικασία εκτύπωσης (να ορίσει τις συνθήκες εκτύπωσης του δοκιμίου) παράγει το G-Code και τον μεταφέρει στον 3D εκτυπωτή.

#### 4. Σκοπός εργασίας

Στο πλαίσιο της παρούσας εργασίας θα σχεδιαστεί και θα αναπτυχθεί σύστημα τηλεχειρισμού οχήματος με χρήση smartσυσκευής. Το σύστημα τηλεχειρισμού θα είναι στη μορφή FPV των τηλεχειριζόμενων drones, ενώ η κίνηση του οχήματος θα πρέπει να είναι εφικτή σε διάφορα εδάφη με κατάλληλη ανάρτηση και με χειρισμό τεσσάρων κατευθύνσεων.

Το σύστημα θα μπορεί να καταγράφει ιστορία χειρισμών, θέσεις καταγραφής εικόνας ή video, και θα μπορεί να την επαναλάβει αυτόματα με κλήση της σε μορφή μακροεντολής.

Η σχεδίαση του μηχανικού μέρους θα γίνει με κριτήριο τη δυνατότητα κίνησης σε εσωτερικούς περιορισμένων διαστάσεων χώρους, σε διάφορα εδάφη, υπό μικρή ορισμένη κλίση, θα φέρει ηλεκτροκινητήρες κατάλληλης ροπής, με ωφέλιμο φορτίο 1kg. Το συνολικό μηχανικό σύστημα θα σχεδιαστεί σε κατάλληλο λογισμικό 3D σχεδίασης, με σκοπό τη μελλοντική του εξέλιξη του.

## 5.Σχεδιασμός Hardware και Software οχήματος

### Σχεδιασμός Hardware και Software οχήματος.

Όπως θα έχει γίνει αντιληπτό από αυτά που ειπώθηκαν στα προηγούμενα κεφάλαια, έχει γίνει αναφορά σε ένα project όπου έχει δημιουργηθεί από το μηδέν. Σε αυτό το κεφάλαιο θα αναλυθεί για το πώς ακριβώς σχεδιάστηκε το hardware αλλά και το software της κατασκευής.

#### 5.1. Σχεδιασμός Hardware χειρισμού και επικοινωνίας

Στην επόμενη σελίδα στον Πίνακα 1 & 2, απεικονίζεται μια λίστα με όλα τα απαιτούμενα υλικά για την υλοποίηση του Hardware.

Το πείραμα ξεκίνησε παίρνοντας ένα RaspberryPi 4 ModelB. Έγινε η πρώτη εκκίνηση με εγκατάσταση του λογισμικού και ρύθμιση της συσκευής.

Έπειτα έγινε η σύνδεση της πλακέτας L298N με το Rpi. Στην πλακέτα L298N επάνω συνδέθηκαν και τα 2 μοτέρ.

Στην πλακέτα L298N έχει δωθεί τροφοδοσία από τις μπαταρίες λιθίου της Samsung 18650. Για να λειτουργήσει η πλακέτα L298N χρειάζεται τάση τροφοδοσίας στα 7V. Επίσης κάτι άλλο πολύ σημαντικό που αξίζει να δωθεί ιδιαίτερη προσοχή είναι πως το Rpi και η L298N χρειάζονται κοινή γείωση.


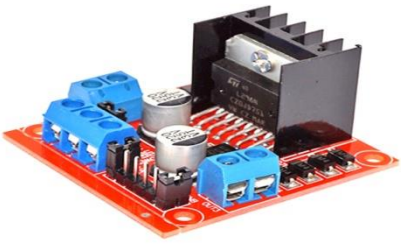


Σε αυτό το σημείο δημιουργήθηκε ένα απλό πρόγραμμα pyhton για απλό χειρισμό – έλεγχο των μοτέρ.

Το RaspberryPi καθ' όλο το κατασκευαστικό στάδιο τροφοδοτείται από το τροφοδοτικό του.

Αφότου ολοκληρώθηκαν όλες οι δοκιμές περί του προγραμματισμού, έπρεπε να βρεθεί ένας τρόπος ώστε να υπάρξει με τον ίδιο εξοπλισμό, επιπλέον μια παροχή στα 5V για την τροφοδοσία του Raspberry Pi ώστε να μην είναι εξαρτώμενο από το τροφοδοτικό του. Έτσι λοιπόν τοποθετήθηκε ένας αντάπτορας ο οποίος συνδέεται με τις μπαταρίες και μετατρέπει την τάση από 7.4 voltσε 5V.

Στο σχήμα 8 φαίνεται το ηλεκτρικό διάγραμμα με τις συνδέσεις.

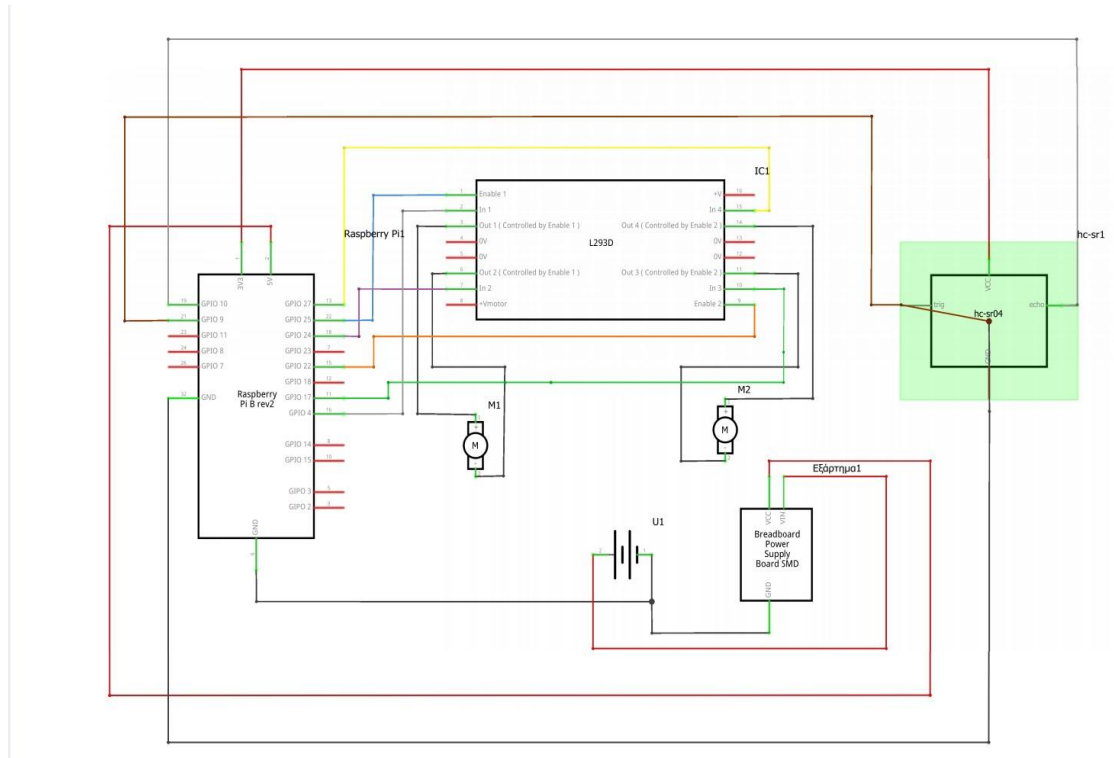


| N/A | Περιγραφή                 | Εικόνα                                                                               |
|-----|---------------------------|--------------------------------------------------------------------------------------|
| 1   | Raspberry Pi 4<br>Model B |    |
| 2   | Motor Driver L298N        |    |
| 3   | Motor 3-6V                |  |
| 4   | Motor 3-6V                |  |

Πίνακας 1: Λίστα χρησιμοποιούμενου εξοπλισμού

| N/A | Περιγραφή                | Εικόνα                                                                                |
|-----|--------------------------|---------------------------------------------------------------------------------------|
| 5   | Adaptor 24/12V to 5V     |    |
| 6   | Camera                   |    |
| 7   | HC-SR04                  |     |
| 8   | Samsung Battery<br>18650 |  |
| 9   | Socket 18650             |  |
| 10  | Cables                   |   |

Πίνακας 2: Λίστα χρησιμοποιούμενου εξοπλισμού



**Σχήμα 8\*:** Απεικόνιση ηλεκτρικού διαγράμματος - Συνδέσεις

## 5.2. Σχεδιασμός software επικοινωνίας και χειρισμού

Το Project ξεκίνησε με το σκεπτικό να βασιστεί επάνω σε ένα ήδη υπάρχον project που είχα διεκπεραιώσει και το οποίο απευθυνόταν σε ένα δίτροχο ρομποτικό όχημα χρησιμοποιώντας Arduino και Rpi με τηλεχειρισμό μέσω ασύρματου πληκτρολογίου USB.

Κατά την ανάπτυξη και διεκπεραίωση αυτού του Project υπήρξε μια μετατροπή. Πραγματοποιήθηκε αφαίρεση του Arduino για ευνότερους λόγους (πχ. βάρος, έξτρα επιβάρυνση του Rpi, μείωση κατανάλωσης ενέργειας κ.α.) και συνδέθηκε απευθείας με όλον τον υπόλοιπο εξοπλισμό επάνω στο Rpi.

Έχοντας ως στόχο τον σκοπό της εργασίας όπου είναι η ανάπτυξη του τηλεχειρισμού το project ξεκίνησε με το πολύ απλό σκεπτικό:

1) Δημιουργία ενός προγράμματος Python του οποίου η δουλειά θα είναι να ελέγχει κατά κύριο λόγο τα μοτέρ.

2) Δημιουργία μιας Ip με την βοήθεια ενός HTML αρχείου το οποίο θα είναι ουσιαστικά ο χειρισμός του ρομπότ. Ο λόγος που εφαρμόστηκε η χρήση της HTML ήταν επειδή θεωρήθηκε πολύ καλύτερο το γεγονός ο χρήστης να μην μπαίνει στον κόπο να κατεβάζει μια εφαρμογή από το playstore ή το Appstore. Αυτό φέρει μια σειρά από πλεονεκτήματα (με τις κατάλληλες αυτοματοποιημένες ρυθμίσεις, είναι λειτουργικό από οποιαδήποτε ηλεκτρονική συσκευή είτε αυτή είναι android, apple ή windows pc

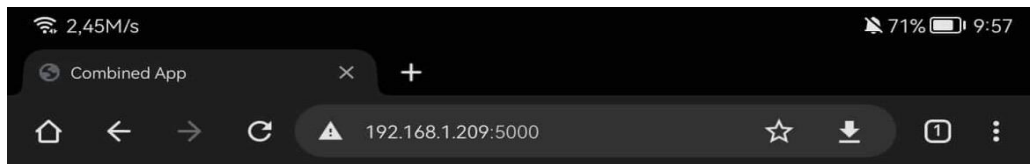
πληκτρολογώντας απλά και μόνο στον browser μια Ip. Η Ip εξαρτάται από το δίκτυο wifi που θα είναι συνδεδεμένο το ρομπότ ή καλύτερα το Rpi.

(πχ. xxx.xxx.x.xxx:5000).

Στην αρχή η δημιουργία του HTML αρχείου ήταν πολύ απλή. Πρόσφερε στην ουσία 6-7 κουμπιά ελέγχου του ρομπότ. (Σχήμα 9)

Στην πορεία όμως κατά την διάρκεια του προγραμματισμού προέκυψε η ανάγκη για πλήρη έλεγχο του ρομπότ όσο το δυνατόν με μέγιστη ακρίβεια. Αυτό προέκυψε επειδή πατώντας το οποιοδήποτε κουμπί δεν υπήρχε η δυνατότητα της μετακίνησης του ρομπότ να πάει με αργό ρυθμό ή αλλιώς να μετακινηθεί με ακρίβεια. Το κουμπί έδινε την εντολή πχ. να κουνηθούν τα μοτέρ στην ταχύτητα που ήταν δηλωμένη μέσα στην python.

Ψάχνοντας στο διαδίκτυο ανακαλύφθηκε μια ιστοσελίδα όπου είχε τον μηχανισμό ενός joystick. Έτσι χρησιμοποιήθηκε αυτός ο κώδικας ως πρότυπο που βρέθηκε στο διαδίκτυο ο οποίος είναι γραμμένος και σε Javascript. Σε αυτό το σημείο αξίζει να σημειωθεί πως παρ' όλο που βρέθηκε στο διαδίκτυο ένας κώδικας, υπέστει ριζική τροποποίηση καθώς δεν κάλυπτε τις ανάγκες κίνησης. Με τις κατάλληλες τροποποιήσεις λοιπόν ο JavaScript καλείτε από το HTML αρχείο το οποίο είναι απαραίτητο για να μας δημιουργήσει την IP. Το HTML αρχείο στην ουσία καλείται όταν τρέχει το python πρόγραμμα ή αλλιώς το Flask.



## Combined App

### Video Feed



### Motor Control



Σχήμα 9: Δοκιμαστικός ελεγκτής με την βοήθεια της HTML

Όπως φαίνεται στο σχήμα 10 παρατηρεί κανείς πως βλέπει μια webapp. Όλο αυτό το webapp με το joystick έχει υποστεί μορφοποίηση μέσω του main.css (Χρωματισμός κουμπιών, μέγεθος, φόντο κλπ).



Σχήμα 10: Webapp

\*Όλα τα προγράμματα (software) αναλύονται στο κεφάλαιο 7.

### **5.3. Συναρμολόγηση και έλεγχος κατασκευής**

Για την συναρμολόγηση της τελικής κατασκευής χρειάστηκε να τροποποιηθεί το σασί για την τοποθέτηση νέου διαφορικού συστήματος. Έπειτα πραγματοποιήθηκε ο σχεδιασμός και εκτύπωση των βασικών δοκιμών για την στήριξη του εξοπλισμού. Η κατασκευή συνεχίζει με την τοποθέτηση του εξοπλισμού και γενικά της συναρμολόγησης της, η οποία αναλύεται στις παρακάτω ενότητες. Κατασκευαστικά έχει προσφέρει ένα πολύ αξιόλογο αποτέλεσμα καθώς έχει πλέον κατάλληλη ροπή καταπολεμώντας αποτελεσματικά τις οποιεσδήποτε δυσκολίες του εδάφους μπορεί να αντιμετωπίσει.

Τέλος πραγματοποιείται η σωστή παραμετροποίηση (ρυθμίζοντας δηλαδή το ρυθμό πρόγραμμα ανάλογα με τις απαιτήσεις μας) ώστε ο χρήστης να έχει τον καλύτερο δυνατό έλεγχο.

Τα στάδια της κατασκευής – συναρμολόγησης φαίνονται παρακάτω από το σχήμα 11 έως το σχήμα 45.

## 6. Σχεδιασμός διάταξης

Σε αυτό το κεφάλαιο θα αναλυθεί το σκεπτικό εκλογής, σχεδίασης και τροποποίησης του οχήματος.

Αρχικά το σκεπτικό ήταν η κατασκευή να βασιστεί επάνω σε ένα ήδη υπάρχον project και πιο συγκεκριμένα σε ένα δίτροχο Robot. Το δίτροχο Robot χρησιμοποιούσε 2 μοτέρ τροχών από 3-5V, DriverMotor, Arduino, και Rpi.

Το DriverMotor ήταν ο ελεγκτής των μοτέρ, και το Arduino περιείχε το βασικό πρόγραμμα οδήγησης και ελέγχου των μοτέρ μέσω της L298N. Όλο αυτό το σύστημα ήταν συνδεδεμένο με το Rpi όπου μέσω ενός προγράμματος Python γινόταν ο τηλεχειρισμός από ένα πληκτρολόγιο.

Ο σκοπός ήταν ουσιαστικά να αλλάξει ο χειρισμός και να δημιουργηθεί κατά τον ίδιο τρόπο ένα πρόγραμμα τα οποίο θα επέτρεπε τον τηλεχειρισμό από οποιαδήποτε smart συσκευή.

Κατά την σκέψη και δημιουργία αυτού το οχήματος υπήρξαν πολλά ερωτηματικά. Μερικά από αυτά ίσως και τα πιο σημαντικά εμφανίζονται παρακάτω.

- 1) Χρειάζεται να δημιουργηθεί κάποιο app για τον τηλεχειρισμό ;
- 2) Τι μπορούμε να αφαιρέσουμε από τον εξοπλισμό ώστε να ελαφρύνουμε την κατασκευή ;
- 3) Ποιος ο τελικός εξοπλισμός Hardware και πως αυτός θα τοποθετηθεί επάνω στην ήδη υπάρχουσα κατασκευή;

Στα επόμενα 3 κεφάλαια θα εξεταστεί η εκλογή του οχήματος, το hardware που χρησιμοποιήθηκε αλλά και το software που αναπτύχθηκε για την επίτευξη του επιθυμητού αποτελέσματος.

### **6.1. Σχεδιασμός γεωμετρίας και εκλογή οχήματος εφαρμογής.**

Όπως αναφέρθηκε και παραπάνω, το όχημα πρέπει να είναι ειδικά σχεδιασμένο για να κινείται μέσα σε κλειστούς χώρους, σε διάφορα εδάφη υπό μικρή ορισμένη κλίση όπως καθώς θα φέρει και ηλεκτροκινητήρες κατάλληλης ροπής. Για αυτόν τον λόγο επιλέχθηκε ένα υπερυψωμένο όχημα με τρακτεροτά ελαστικά (Σχήμα 11.)





Σχήμα 11: Αρχική μορφή οχήματο που εκλέχθηκε για το Project

Πραγματοποιήθηκε λοιπόν η αγορά και εκλογή οχήματος (Σχήμα 13.) από γνωστή εμπορική ιστοσελίδα και ξεκίνησε αμέσως η διαδικασία αφαίρεσης όλων των περιττών εξαρτημάτων.

## **6.2. Μετατροπές οχήματος**

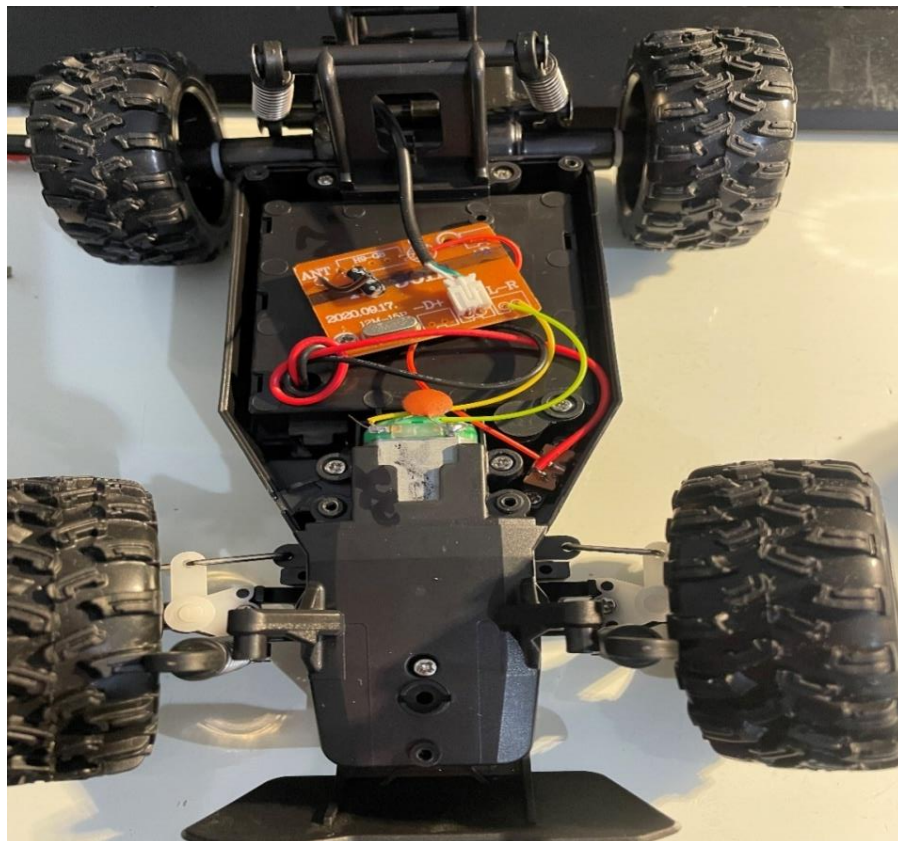
Αρχικά αφαιρέθηκε η 'καμπίνα' και το επάνω καπάκι. (Σχήμα 12 & 13)

Στην αρχή τοποθετήθηκε ο ήδη υπάρχον εξοπλισμός (RaspberryPi, L298N, PowerBank, 2xLi-oh 18650 Samsung, 1 case 186650 1 Microsoft camera) πρόχειρα επάνω στο όχημα συνδέοντας το μοτέρ κινήσεως και διευθύνσεως του οχήματος καθώς ήταν τα μόνα που είχαν μείνει επάνω και ξεκίνησαν τα πειράματα. Σε αυτό το σημείο δημιουργήθηκαν 2 πολύ σοβαρά προβλήματα.





Σχήμα 12: Αφαίρεση καμπίνας του οχήματος



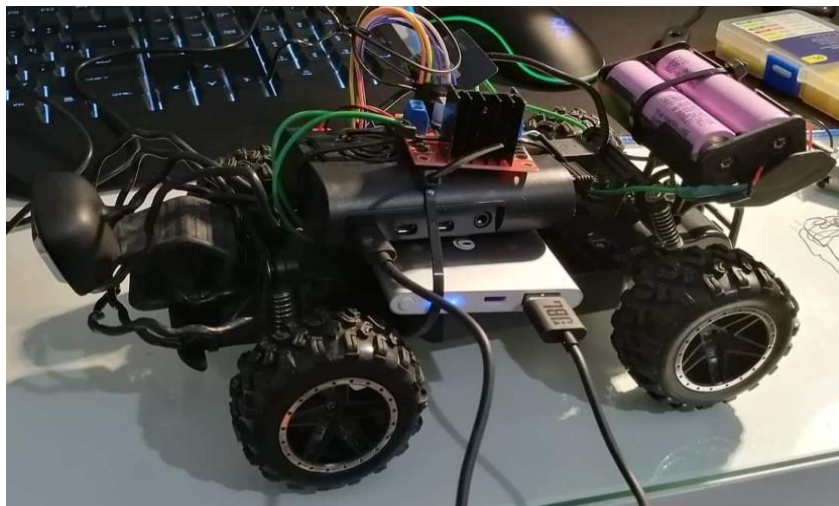
Σχήμα 13: Αφαίρεση πλαστικού καλύμματος για πρόσβαση στο ηλεκτρονικό κομμάτι

Το 1<sup>ο</sup> πρόβλημα ήταν πως είχε αυξηθεί πλέον το βάρος κατά πολύ, με αποτέλεσμα το robot εν κινήσει να μην έχει την δύναμη της ροπής, δηλαδή την δύναμη να ανέβει την οποιαδήποτε ανωμαλία του εδάφους θα μπορούσε να βρει εκείνη την στιγμή.

Το 2<sup>ο</sup> πρόβλημα είχε να κάνει πάλι με το υπερβολικό βάρος καθώς η ανάρτηση όλου του μπροστινού συστήματος είχε τερματίσει με αποτέλεσμα να δυσκολεύει το robot να κινηθεί. (Σχήμα 14, 15 )



Σχήμα 14: Μπροστινή όψη οχήματος με προσωρινά τοποθετημένο του ήδη υπάρχοντος εξοπλισμού



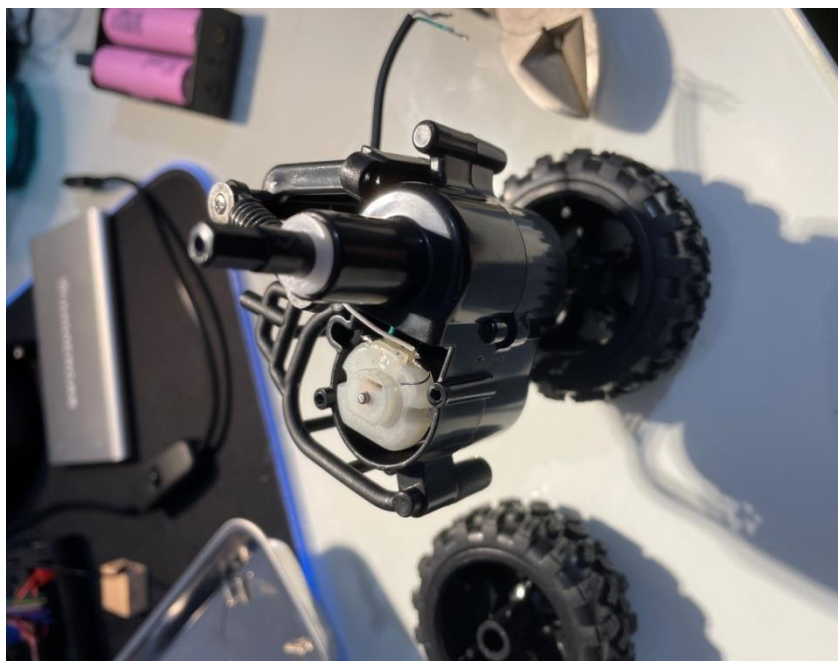
Σχήμα 15: Αριστερή πλάγια όψη οχήματος με προσωρινά τοποθετημένο του ήδη υπάρχοντος εξοπλισμού

Για την εξάλειψη των παραπάνω προβλημάτων έπρεπε να πραγματοποιηθεί μετατροπή του οχήματος.

Η μετατροπή ήταν να αφαιρεθεί πλήρως το διαφορικό «μαμά» που διέθετε το όχημα και να προσαρμοστεί επάνω ένα διαφορικό πολύ πιο δυνατό από άλλο όχημα. (Σχήμα 16 έως 23.)

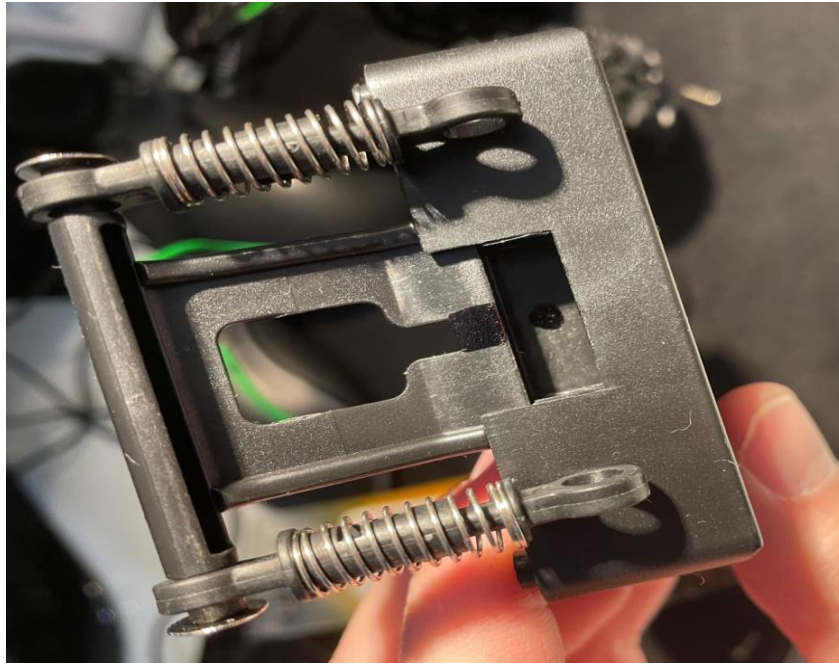


Σχήμα 16: Αφαίρεση διαφορικού οχήματος

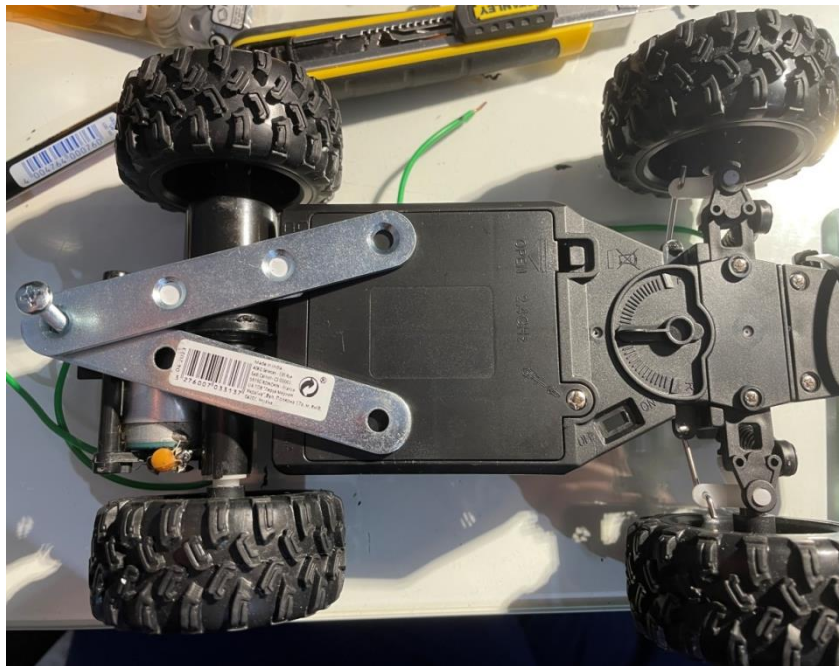


Σχήμα 17: Διαφορικό οχήματος

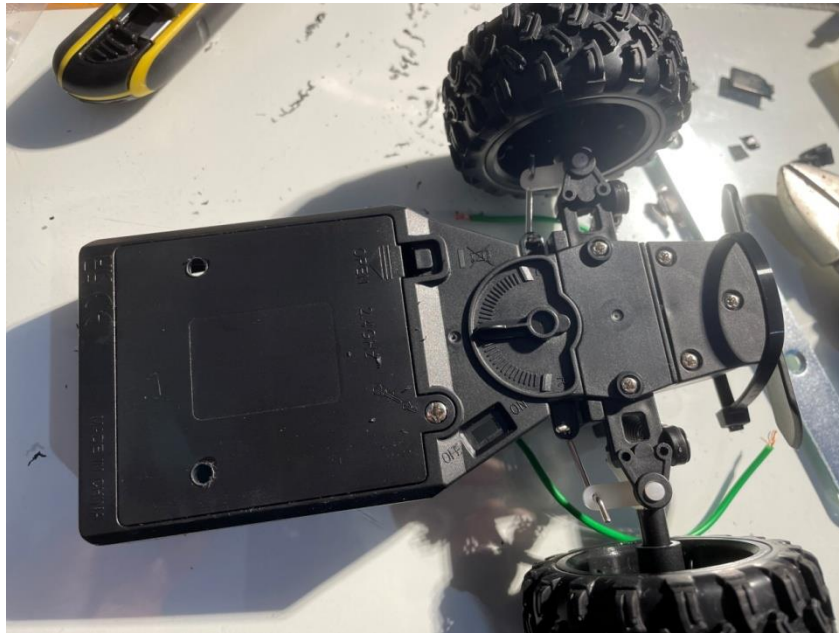




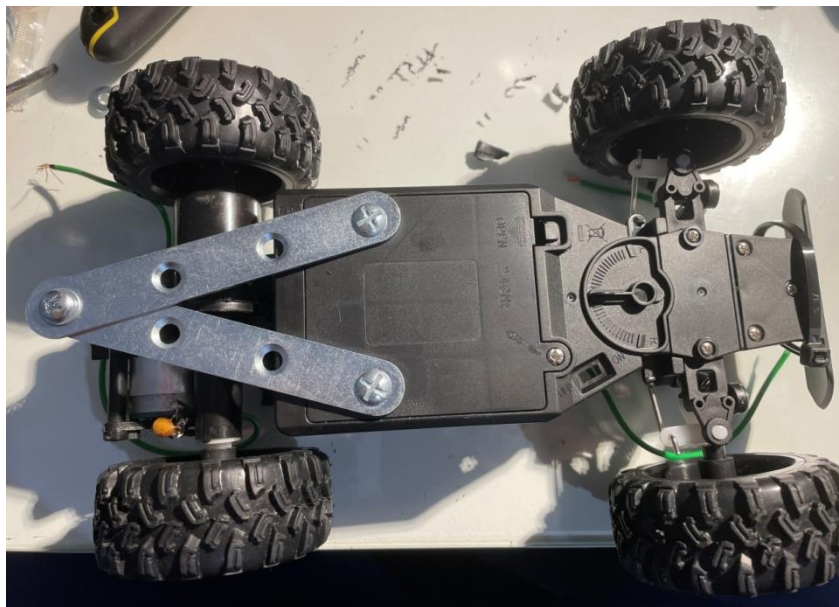
Σχήμα 18: Πίσω μέρος οχήματος χωρίς διαφορικό



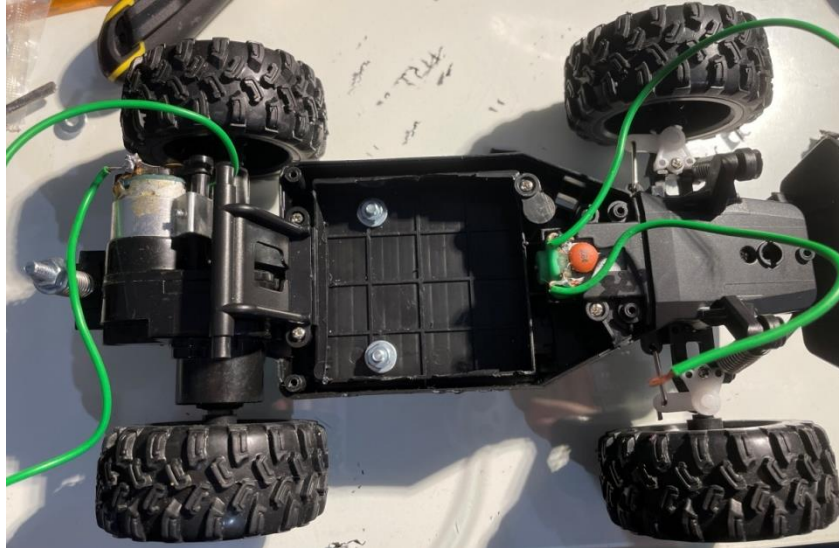
Σχήμα 19: Αναπαράσταση τοποθέτησης καινούριου διαφορικού στο όχημα



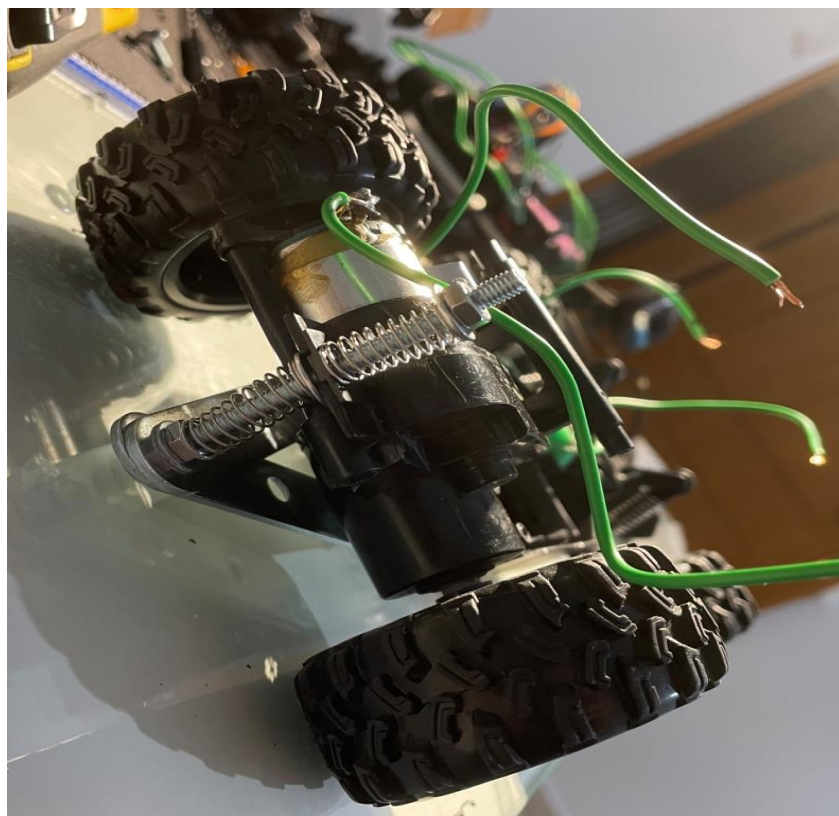
Σχήμα 20: Διαδικασία προσαρμοσης καινούριου διαφορικού



Σχήμα 21: Τοποθέτηση ειδικών λαμών για την στήριξη διαφορικού



Σχήμα 22: Επάνω όψη οχήματος – σκέτο σασί με διαφορικό και σύστημα διεύθυνσεως



Σχήμα 23: Πίσω όψη οχήματος με τοποθετημένο το καινούριο διαφορικό

Όπως φαίνεται στις παραπάνω εικόνες έχει γίνει η προσαρμογή του νέου τύπου διαφορικού επάνω στο όχημα και έχει αφαιρεθεί ένα κομμάτι πλαστικό από το κύριο σασί ώστε να κερδηθεί χώρος για την εγκατάσταση του εξοπλισμού.

Όπως διακρίνεται στο σχήμα 25, η στήριξη του διαφορικού γίνεται με την βοήθεια μιας βίδας- ντίζας και με 2 ελατήρια τα οποία βοηθάνε στην απόσβεση των κραδασμών. Εκτός αυτού βοηθάνε και το όχημα κατά την κίνηση του ώστε να περνάει τις διάφορες

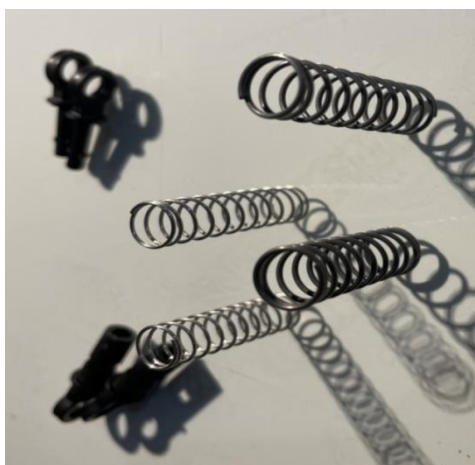


ανωμαλίες του εδάφους πιο εύκολα. Το πλεονέκτημα αυτής της πατέντας είναι ότι σφίγγοντας τα παξιμαδιαμπορεί να ρυθμιστεί η ανάρτηση (μαλακή – σκληρή).

Εκτός από την προσαρμογή του νέου τύπου διαφορικού έγινε και αντικατάσταση των μπροστινών ελατηρίων με πιο σκληρά ώστε να μπορεί να σηκώσει το βάρος ο μπροστινός άξονας. (Σχήμα 24 & 25)

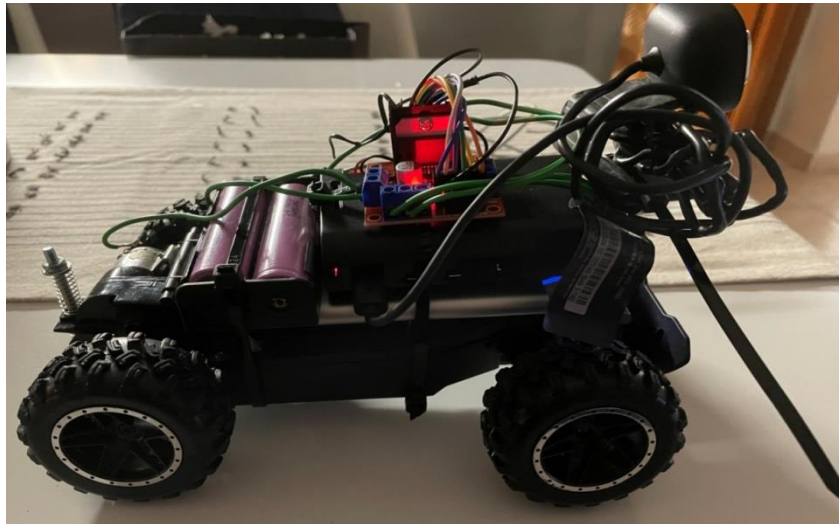


Σχήμα 24: Μπροστινή ανάρτηση οχήματος

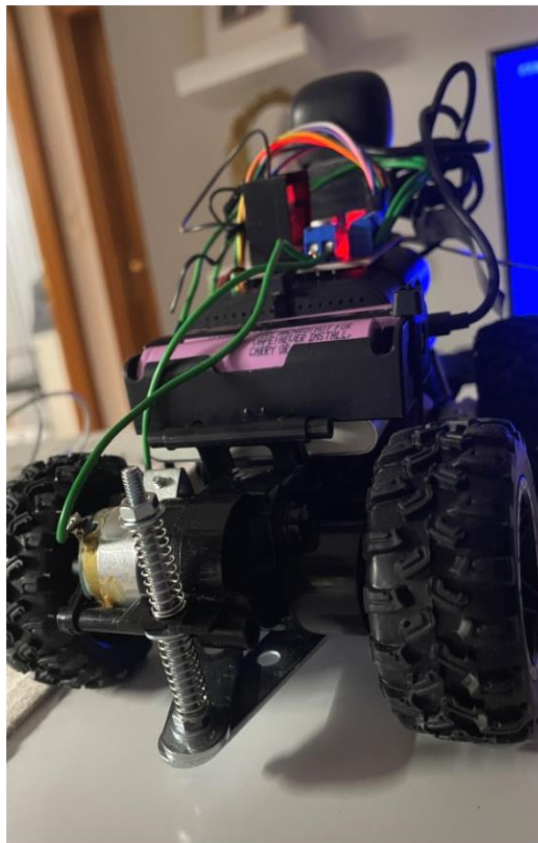


Σχήμα 25: Αριστερά της εικόνας: Γνήσια ελατήρια οχήματος. Δεξιά της εικόνας: ελατήρια μετατροπής.

Σε αυτό το σημείο επανατοποθετείται προσωρινά ο εξοπλισμός επάνω στο όχημα για δοκιμή και διαπιστώνεται πως από δύναμη είναι αυτό που ακριβώς ψάχνουμε στην συγκεκριμένη περίπτωση. Η κατασκευή πλέον μοιάζει με αυτή των σχημάτων 26, 27 & 28.



Σχήμα 26: Πλάγια όψη οχήματος μετατροπής με τοποθετημένο προσωρινά τον εξοπλισμό



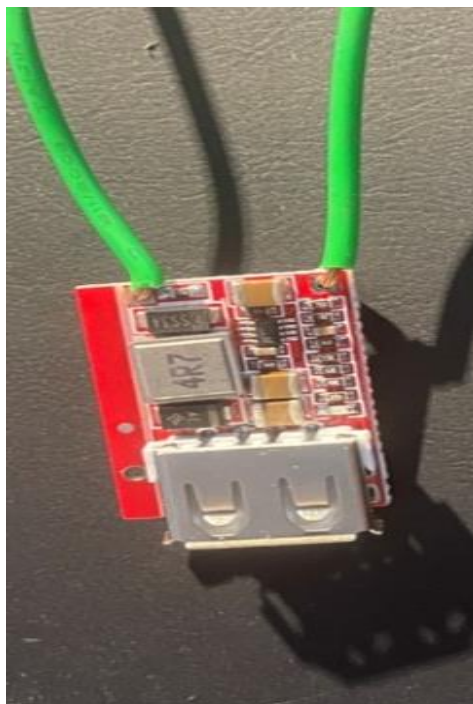
Σχήμα 27: Πίσω όψη οχήματος μετατροπής με τοποθετημένο προσωρινά τον εξοπλισμό





Σχήμα 28: Εμπρός όψη οχήματος μετατροπής με τοποθετημένο προσωρινά τον εξοπλισμό

Στην προσπάθεια μείωσης βάρους, αγοράστηκε από το εμπόριο ένας μετατροπέας ο οποίος μετατρέπει την τάση μέχρι και 24v σε 5V όπου είναι η απαιτούμενη τροφοδοσία για το RaspberryPi. (Σχήμα 29)



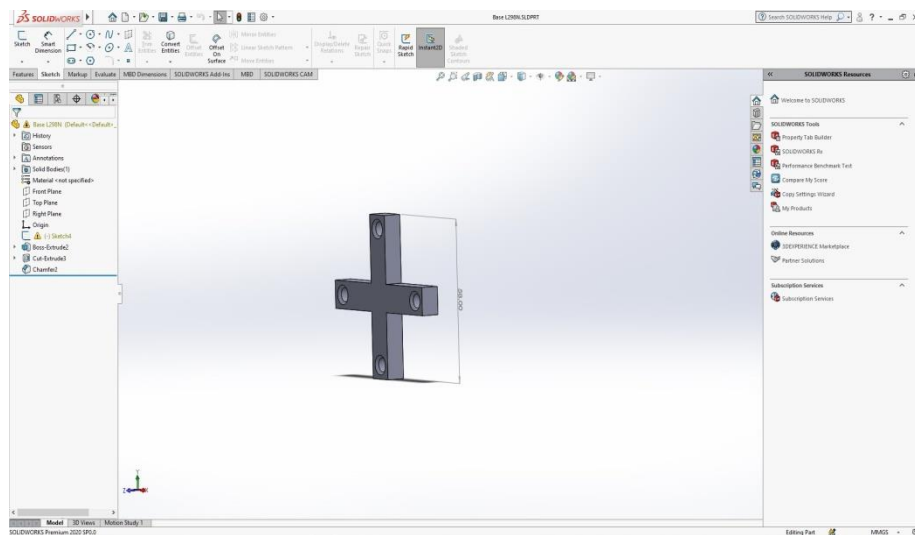
Σχήμα 29: Μετατροπέας τάσης

### 6.3. Σχεδίαση εξαρτημάτων με 3DPrinting

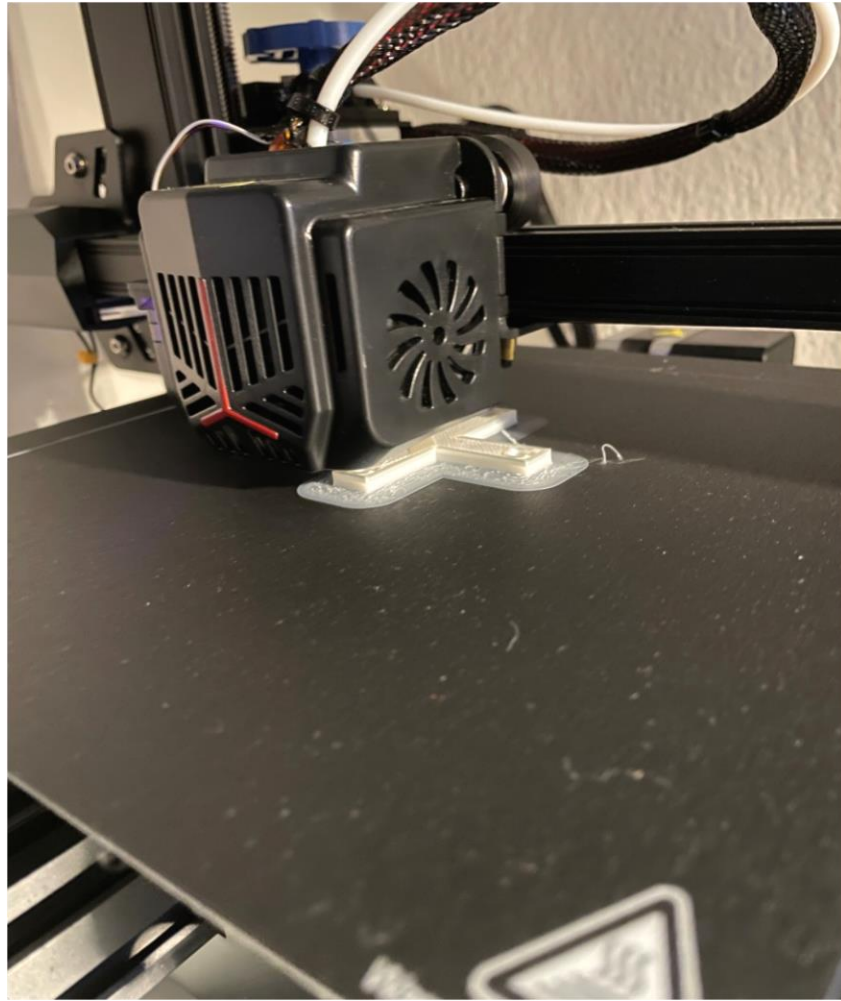
Έχοντας φτάσει την κατασκευή σε ένα επιθυμητό σημείο όσον αφορά τη δύναμη και το βάρος έχει ξεκινήσει η διαδικασία σχεδίασης και εκτύπωσης των εξαρτημάτων για την σωστή τοποθέτηση του εξοπλισμού επάνω στο όχημα.

Αρχικά σχεδιάστηκε και εκτυπώθηκε μία βάση σε σχήμα σταυρού η οποία τοποθετήθηκε στο χαμηλότερο σημείο του σασί έχοντας 2 ρόλους. (Σχήμα 30, 31, 32)

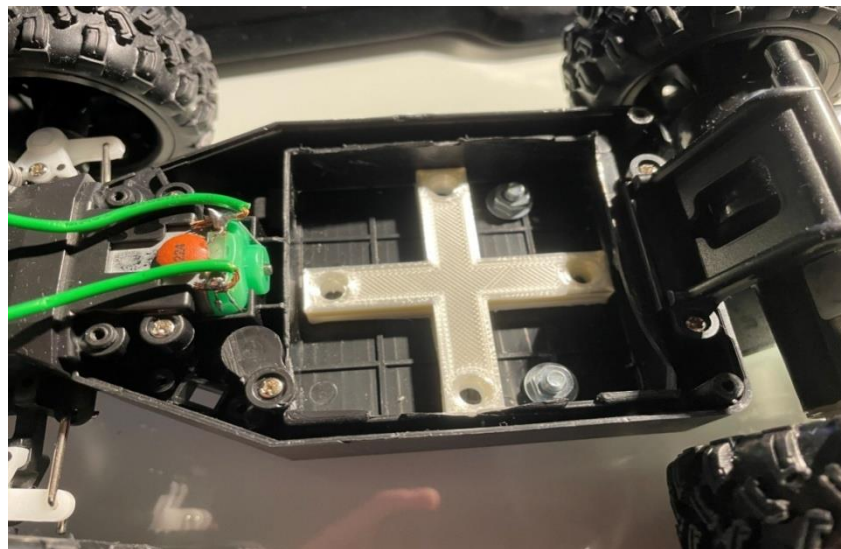
- 1) Κρατάει κόντρα το πλαστικό ώστε να αποφευχθούν οι παραμορφώσεις από πιθανές προσκρούσεις στο έδαφος
- 2) Επάνω εκεί τοποθετήθηκε ο αντάπτορας-μετατροπέας της τάσης.



Σχήμα 30: 3D σχέδιο βάσης



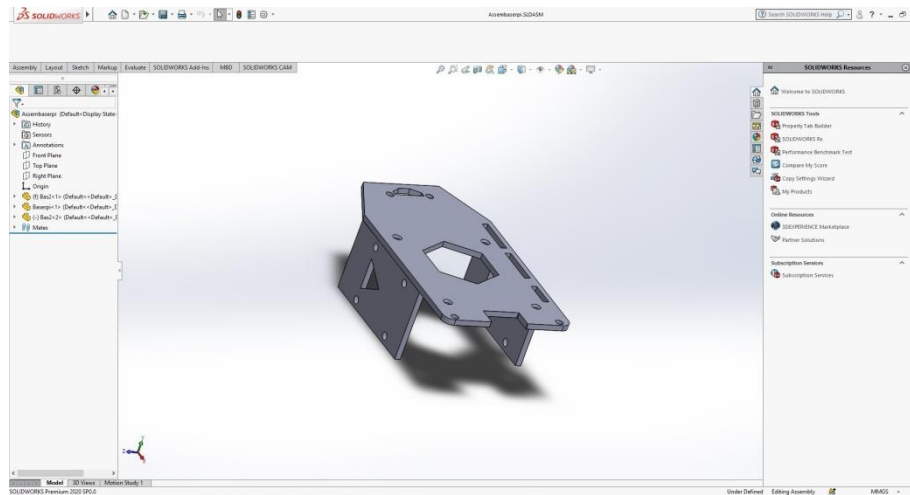
Σχήμα 31: Κατά την διάρκεια εκτύπωσης του εξαρτήματος



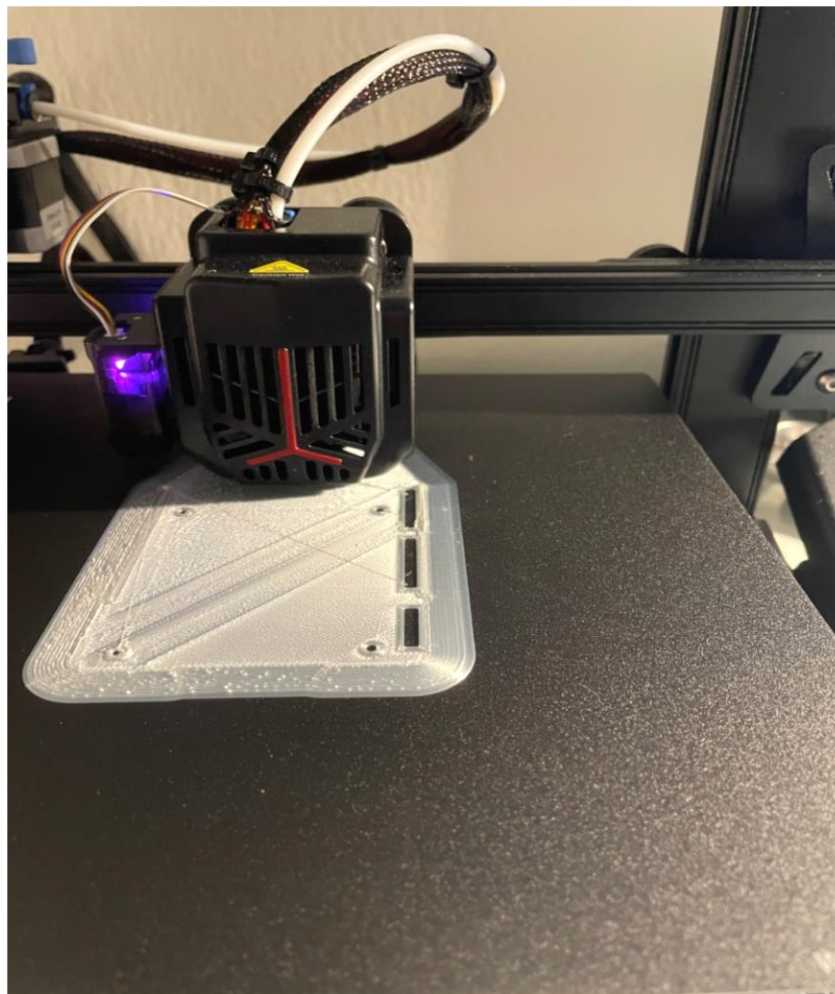
Σχήμα 32: Το εκτυπωμένο εξάρτημα τοποθετημένο στο σασί

Έπειτα σχεδιάστηκε και εκτυπώθηκε η κύρια βάση του οχήματος όπου εκεί επάνω τοποθετήθηκε από την μέσα και κάτω πλευρά η πλακέτα οδήγησης των μοτέρ L298N και από την επάνω πλευρά τοποθετήθηκε το RaspberryPi.

(Σχήμα 33, 34, 35 & 36.)

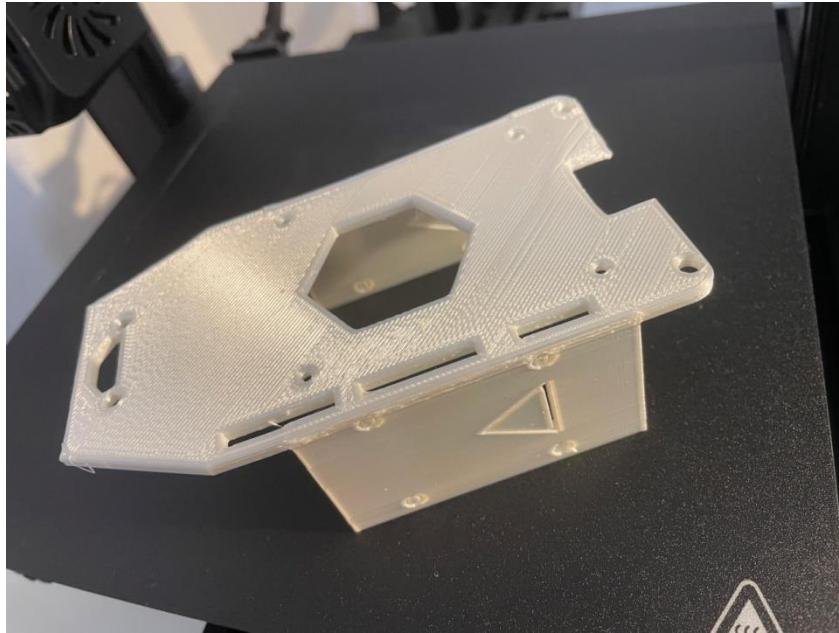


**Σχήμα 33:** 3Δ Σχεδίο κύριας βάσης οχήματος (Βάση Rpi)

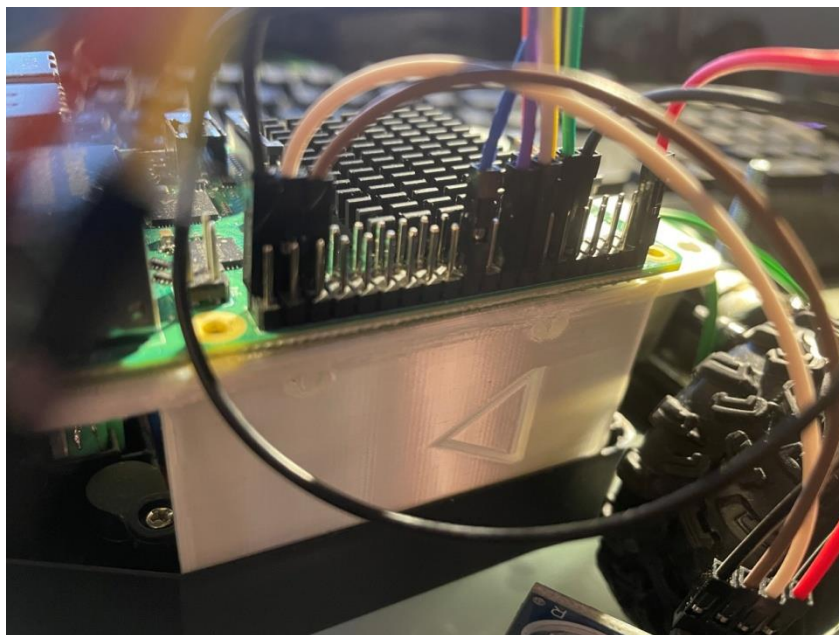


**Σχήμα 34:** Εκτύπωση βάσης Rpi



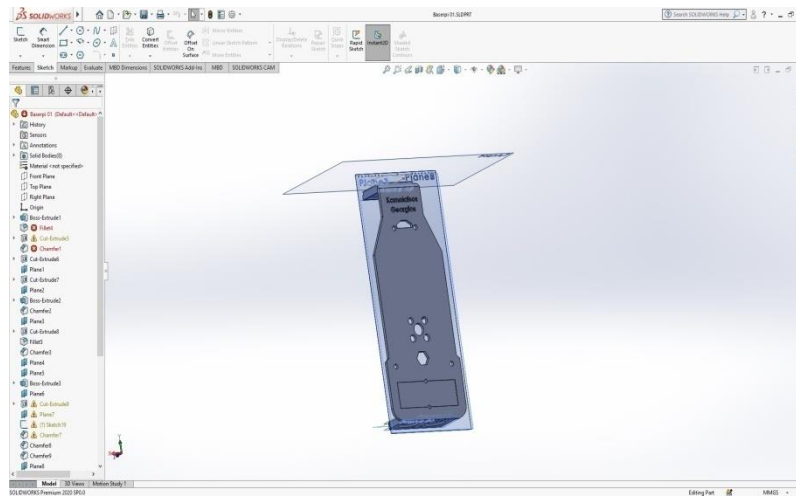


Σχήμα 35: Εκτυπωμένη βάση Rpi (κύριο σώμα οχήματος)

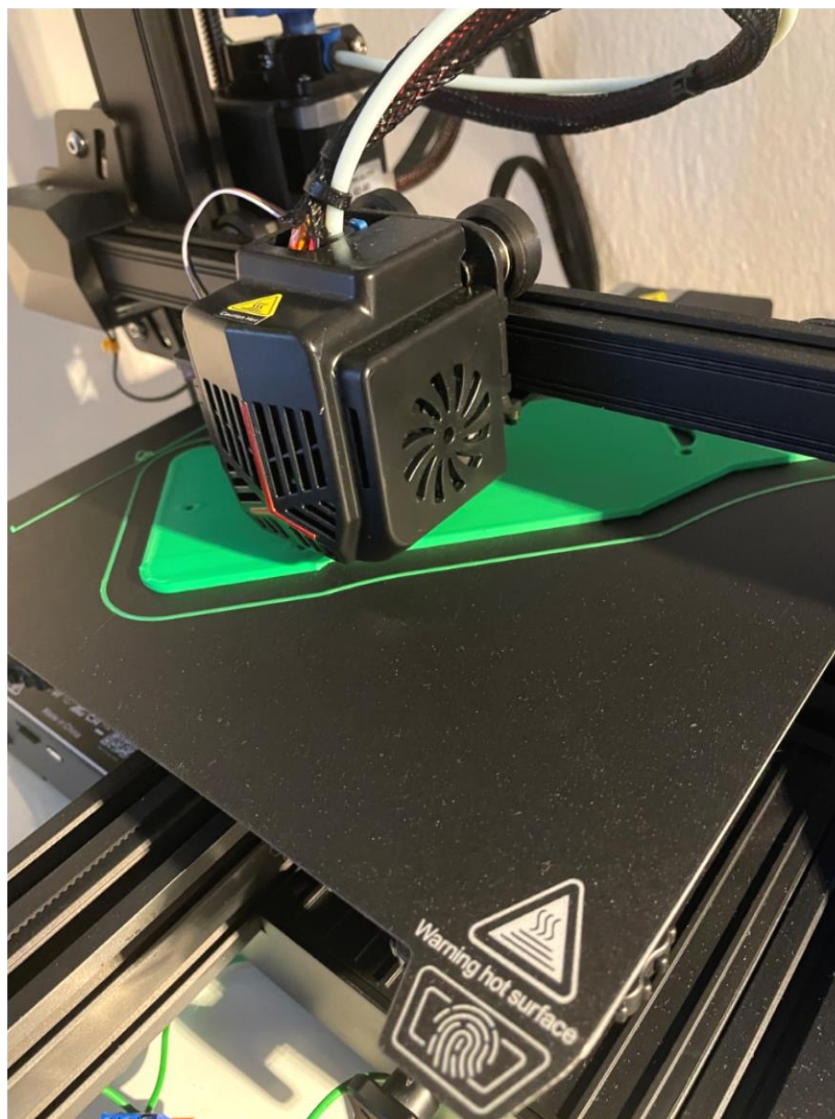


Σχήμα 36: Τοποθετημένο το κύριο σώμα στο όχημα με την Rpi.

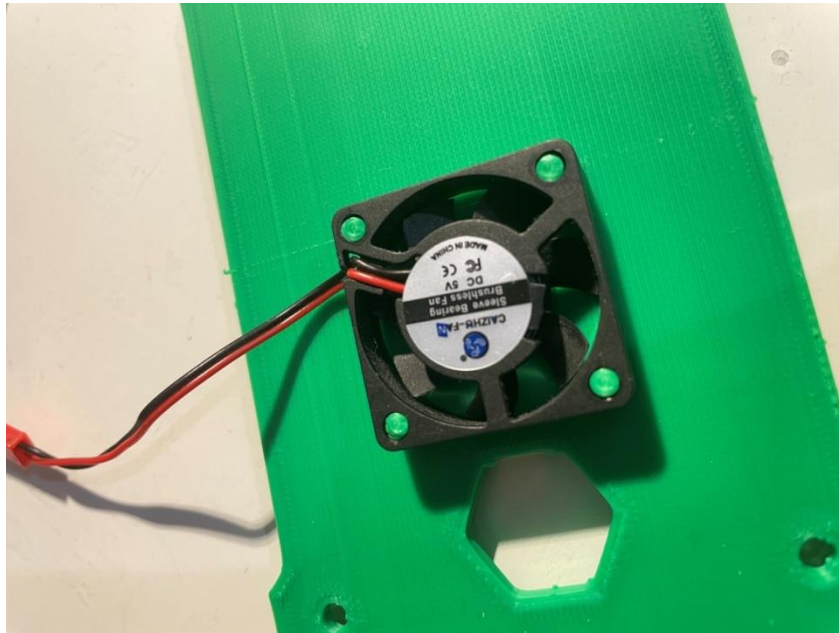
Τέλος σχεδιάστηκε και εκτυπώθηκε το τελικό επίπεδο της κατασκευής όπου από την κάτω πλευρά περιέχει τον ανεμιστήρα που αποτελεί και την ενεργητική ψύξη του Raspberry Pi, την κάμερα στο μπροστινό μέρος και τους αισθητήρες απόστασης στο πίσω μέρος. Από την επάνω πλευρά έχει τοποθετηθεί το socket υποδοχής των μπαταριών 18650. (Σχήμα 37, 38, 39 & 40)



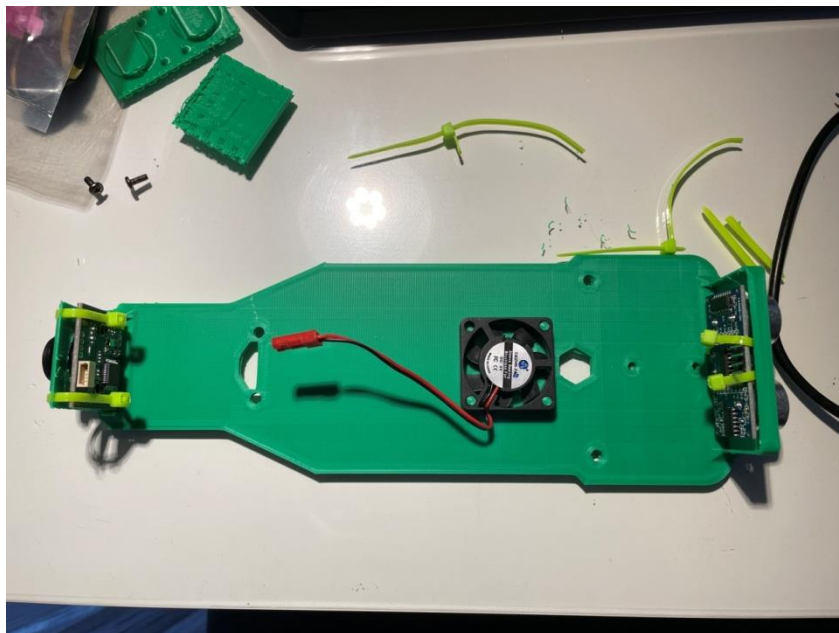
Σχήμα 37: 3Δ Σχέδιο τελικού επιπέδου κατασκευής.



Σχήμα 38: Κατά την διάρκεια εκτύπωσης του τελικού επιπέδου του οχήματος



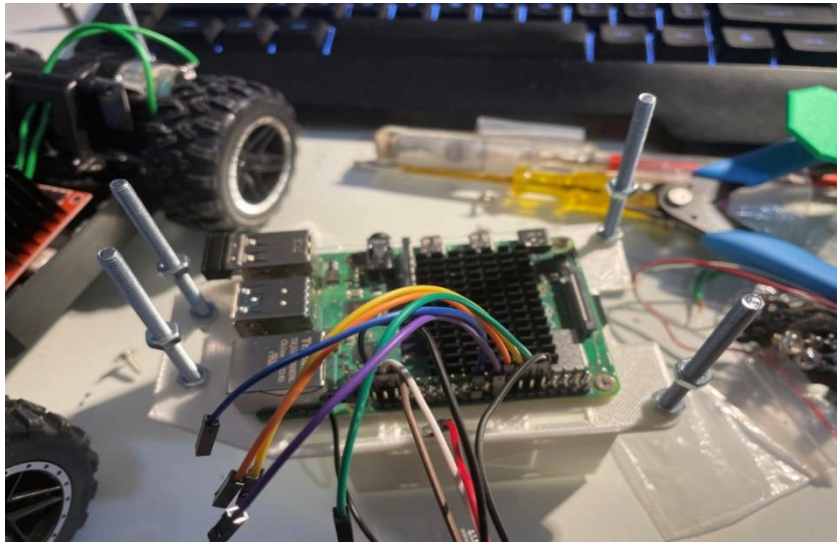
Σχήμα 39: Τοποθέτηση ενεργητικής ψύξης του Rpi.



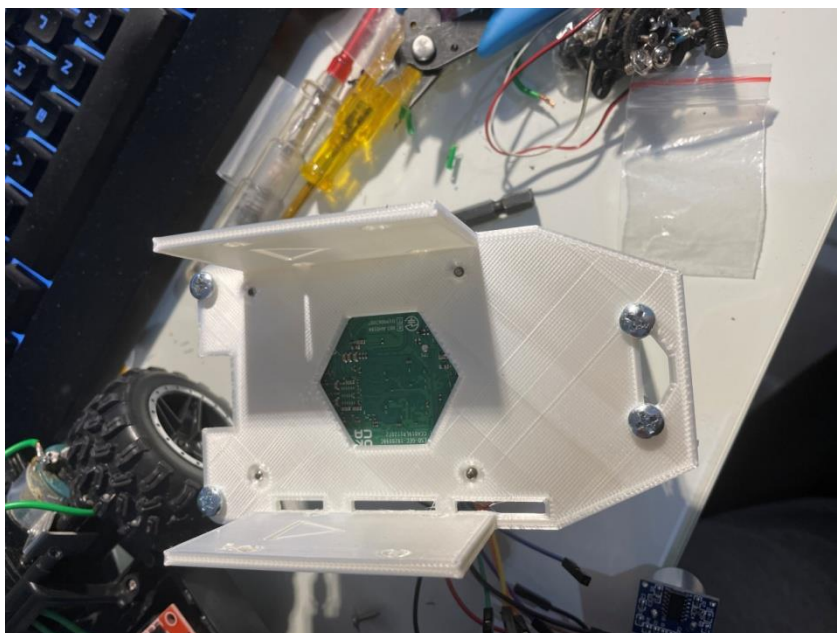
Σχήμα 40: Κάτω πλευρά του τελικού επιπέδου του οχήματος με τοποθετημένα τους αισθητήρες HC-SR04, την κάμερα και την ενεργητική ψύξη του Rpi.

Οι παρακάτω εικόνες δείχνουν μερικά στάδια κατά την συναρμολόγηση του οχήματος με τα εκτυπωμένα εξαρτήματα. (Σχήμα 41, 42, 43.)



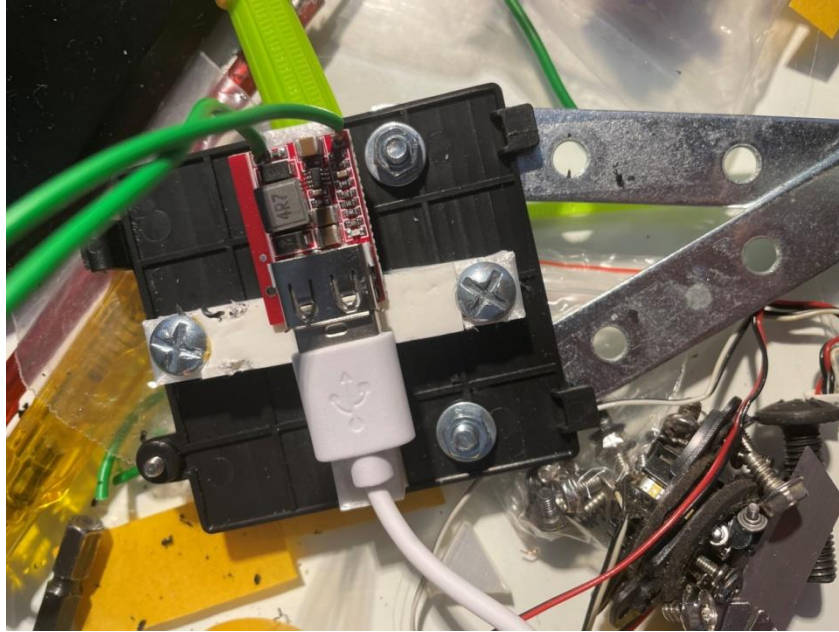


Σχήμα 41: Διαδικασία τοποθέτησης Rριεπάνω στην εκτυπωμένη βάση.



Σχήμα 42: Κάτω όψη βάσης Rρι – κύριου σώματος





Σχήμα 43: Τοποθέτηση βάσης κόντρας και αντάπτορα

Η κατασκευή πλέον έχει την ακόλουθη μορφή. (Σχήμα 44, 45)



Σχήμα 44: Τελική μορφή οχήματος (εμπρός όψη)



Σχήμα 45: Τελική μορφή οχήματος (πίσω όψη)

## 7. Ανάπτυξη κώδικα

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, το software αποτελείται στην ουσία από 3 κώδικες όπου συνδέονται μεταξύ τους.

- 1) Κώδικας Python – Flask πρόγραμμα. Αποτελεί τον κώδικα ελέγχου του ρομπότ.
- 2) Κώδικας HTML. Αποτελεί τον κώδικα επικοινωνίας του ρομπότ με την οποιαδήποτε συσκευή.
- 3) Κώδικας JavaScript. Είναι ο κώδικας χειρισμού και αυτοματισμών.

### 7.1. Κώδικας επικοινωνίας

Σε αυτήν την ενότητα θα αναλυθεί ο κώδικας επικοινωνίας HTML.

Το HTML αρχείο είναι το template αρχείο που σερβίρει η Python με το flask.

Ξεκινάει φορτώνοντας βιβλιοθήκες.

Η πρώτη βιβλιοθήκη είναι το createJs. Είναι μια βιβλιοθήκη όπου μέσω αυτής σχεδιάζονται πραγματάκια με την βοήθεια της JavaScript. Αυτή η βιβλιοθήκη ήταν μέσα στο joystick που βρέθηκε στο διαδίκτυο. Σε αυτό το σημείο πρέπει να γίνει κατανοητό πως το κομμάτι του κώδικα που εντοπίστηκε στο διαδίκτυο απλά «ζωγράφισε» το joystick χωρίς αυτό να το καθιστά έτοιμο προς χρήση. Στο Createjs.com εξηγείται πως μπορεί κάποιος να δημιουργήσει διάφορα σχήματα. Αυτή η βιβλιοθήκη είναι online.

Πέραν του σχηματισμού του joystick όμως, πρέπει να υπάρχει και λειτουργικότητα, δηλαδή μετακινώντας το, να δίνει την αντίστοιχη κίνηση στο ρομπότ. Για να επιτευχθεί αυτό το αποτέλεσμα χρειάστηκαν ριζικές αλλαγές στον κώδικα, ο οποίος φυσικά διασυνδέθηκε με το HTML αρχείο.

Έπειτα χρησιμοποιείται η βιβλιοθήκη easytimer. Το easytimer χρησιμοποιείται για το χρονόμετρο του βίντεο.

Επίσης χρησιμοποιείται η βιβλιοθήκη hamer όπου είναι η βιβλιοθήκη για τις χειρονομίες, πχ η κίνηση του joystick με το ποντίκι ή το joystick.

Η Js-file-download είναι μια μικρή βιβλιοθήκη όπου χρησιμοποιείται για να γίνει download η εικόνα ή το βίντεο στον browser. (πχ. αν δεν κατεβαίνει κάποιο αρχείο μπορεί να ευθύνεται αυτή η βιβλιοθήκη).

Αφού γίνουν αυτές οι βιβλιοθήκες import υπάρχουν τα αρχεία main όπου το ένα είναι js και το άλλο css(είναι το styling) (σημαίνει cascadingstylesheets όπου πάει πακέτο με

την html ώστε να δημιουργηθεί μια webeφαρμογή) όπου τοποθετείται μέσα ο κώδικας που έχει δημιουργηθεί.

Η html δεν δίνει την δυνατότητα μορφοποιήσεων του κειμένου (μέγεθος, χρώμα, θέση) συνεπώς για αυτό χρησιμοποιείται ηcss.

Συνεπώς στο main.css ορίζονται οι μορφοποιήσεις αρεσκείας και έπειτα καλούντε στο html.

Στο τέλος του HTMLαρχείου όταν δεν έχει path και είναι έτσι :

```
106     <script>
107         initialize();
108     </script>
109 </body>
110
111 </html>
```

στην ουσία ζητείται να φορτώσει όλη την HTML και όταν το κάνει να καλέσει την initialize. Η initialize φορτώνει το joystick και προετοιμάζει και το χρονόμετρο. Αν αυτό γινόταν παραπάνω στα imports θα υπήρχε το εξής πρόβλημα: πχ Κάποιος μπορεί να καλέσει κάτι στην js μέσα στο αρχείο κάπου ψιλά και να μην υπάρχει το joystick ακόμα. Ως αποτέλεσμα δεν θα μπορεί να ζωγραφιστεί εκεί πάνω το οτιδήποτε. Έτσι μπαίνοντας τέρμα κάτω η initialize περιμένει να φορτώσει όλο το gui (το κομμάτι της HTML) και ξεκινάει να φορτώνει τα βασικά πράγματα. (πχ τα κυκλάκια).

## 7.2. Κώδικας ελέγχου οχήματος

Ο κώδικας αυτός είναι γραμμένος σε γλώσσα python και αποτελεί τον κύριο κώδικα του ρομπότ. Είναι σχεδιασμένος για να ελέγχει ένα ρομποτικό σύστημα που περιλαμβάνει τους κινητήρες και μια κάμερα. Επιπλέον παρέχει την δυνατότητα να αναπτυχθεί ανάλογα με τις ανάγκες που μπορεί να προκύψουν στην κατασκευή.

Ο κώδικας ξεκινάει με την εισαγωγή των απαραίτητων βιβλιοθηκών (Flask, OpenCV, RPi.GPIO, threading, time, κλπ). Έπειτα γίνεται καθορισμός των pins του RPi που ελέγχουν τους κινητήρες και οτιδήποτε άλλο χρειαστεί, καθώς γίνεται και η αρχικοποίηση τους. Χρησιμοποιώντας την OpenCV ελέγχει την κάμερα.

Δημιουργεί μια λίστα για την αποθήκευση των frames της κάμερας κατά την εγγραφή.

Ορίζει τις συναρτήσεις για την κίνηση του ρομπότ, τον έλεγχο της κάμερας και άλλες λειτουργίες. Ορίζει τις διαδρομές (routes) της εφαρμογής Flask. Πιο συγκεκριμένα υπάρχει μια διαδρομή που επιστρέφει το κυρίως HTML αρχείο της διεπαφής, και άλλες διαδρομές οι οποίες εξυπηρετούν την ροή των βίντεο από την κάμερα, τον έλεγχο της

κίνησης του ρομπότ, την εκκίνηση και την παύση της εγγραφής βίντεο, τη λήψη φωτογραφίας από την κάμερα, την εκτέλεση μακροεντολών, στοπ της κίνησης και καθαρισμό των pins.

Αυτός ο κώδικας επιτρέπει στον χρήστη να ελέγχει το ρομπότ, να παρακολουθεί την εικόνα από την κάμερα, να εγγράφει βίντεο και να εκτελεί μακροεντολές μέσω μιας διεπαφής Flask.

Όταν τρέξει το ρυθον πρόγραμμα δημιουργείται μια ip (ανάλογα με το δίκτυο όπου είναι συνδεδεμένο το Rpi στην Port 5000 και μπαίνοντας εκεί έχουμε τον controller του ρομπότ.

### **7.3. Κώδικας χειρισμού και αυτοματισμών**

Είναι το αρχείο Main.js

Ο κώδικας ξεκινάει με τη φόρτωση του joystick. Είναι βασισμένο στο παράδειγμα που έχει βρεθεί στο παρακάτω link και έχει υποστεί τροποποιήσεις καθώς δεν κάλυπτες απαιτούμενες ανάγκες για τον χειρισμό του ρομπότ.

<https://codepen.io/jiffy/pen/zrqwON> [1]

Αυτός ο κώδικας είναι ένα σύνολο JavaScript που λειτουργεί σε συνδυασμό με τον ρυθον κώδικα που περιγράφηκε προηγουμένως. Αυτός ο κώδικας αναλαμβάνει τη διεπαφή με τον χρήστη, την επικοινωνία με το Flask server και τον έλεγχο των διάφορων λειτουργιών του ρομπότ.

Ιδιαίτερη προσοχή στον κώδικα αυτό αξίζει να δώσει κανείς στις συναρτήσεις του κώδικα js που περιγράφονται παρακάτω.

1.initialize: Αρχικοποιεί τα βασικά στοιχεία, όπως του joystick και του timer για την εγγραφή βίντεο.

2.initializeJoyStick: Δημιουργεί ένα joystick που επιτρέπει στον χρήστη να ελέγχει την κίνηση του ρομπότ. Αυτό υλοποιείται με χρήση του Hammer.js και της βιβλιοθήκης CreateJS για το rendering του joystick.

3.initializeVideoRecordingTimer: Δημιουργεί έναν timer για την εγγραφή βίντεο. Εμφανίζει δηλαδή τον χρόνο εγγραφής στο πεδίο του χρήστη.

4.calculateCoords: Υπολογίζει τις συντεταγμένες (x, y) για το joystick βάσει της γωνίας και της απόστασης. Βάση αυτής της συνάρτησης το ρομπότ καταλαβαίνει ακριβώς τι πρέπει να κάνει διότι δίνεται ακρίβεια στο πάτημα του joystick.

5.getDirectionByAngle: Καθορίζει την κατεύθυνση της κίνησης του ρομπότ με βάση τη γωνία του joystick.

6.onMovementChange: Επικοινωνεί με τον server Flask για να εκτελέσει την κίνηση του ρομπότ με βάση την κατεύθυνση και την ταχύτητα που καθορίζεται από τη θέση του joystick.

7.capturePhoto: Αποστέλλει αίτηση στον server Flask για λήψη φωτογραφίας από την κάμερα.

8.toggleVideoRecording: Ξεκινά ή σταματά την εγγραφή βίντεο ανάλογα με το αν είναι ήδη σε εξέλιξη η εγγραφή.

9.toggleStreaming: Ενεργοποιεί ή απενεργοποιεί την ροή της κάμερας από τον server Flask.

10.startStreaming και stopStreaming: Ξεκινά και σταματά την ροή της κάμερας αντίστοιχα.

Ο κώδικας αυτός είναι υπεύθυνος για την αλληλεπίδραση του χρήστη με το ρομπότ και την εκτέλεση των κατάλληλων ενεργειών στον server Flask για τον έλεγχό του.

## 8. Ανάπτυξη παραδειγμάτων χρήσης

Τα ρομποτικά οχήματα έχουν εφαρμογές σε πολλούς τομείς και μπορούν να χρησιμοποιηθούν για πολλούς σκοπούς. Μερικοί από αυτούς τους σκοπούς αναφέρονται παρακάτω.

### 1. Εξερεύνηση Δύσβατων Περιοχών:

Ρομποτικά οχήματα μπορούν να χρησιμοποιηθούν για την εξερεύνηση δύσβατων περιοχών, όπως απομονωμένα νησιά ή απόκεντρες περιοχές, όπου οι άνθρωποι δυσκολεύονται να φτάσουν. Για παράδειγμα, μπορούν να χρησιμοποιηθούν για την εξερεύνηση ανεξερεύνητων θαλάσσιων χώρων ή απομονωμένων περιοχών σε αναρίθμητα μέρη του πλανήτη.

### 2. Αγροτική Χρήση:

Στη γεωργία, ρομποτικά οχήματα μπορούν να χρησιμοποιηθούν για την αυτοματοποίηση διαδικασιών όπως η σπορά, ο θερισμός και ο έλεγχος της καλλιέργειας. Μπορούν να εφαρμοστούν επίσης για την παρακολούθηση της ενυδάτωσης του εδάφους και τη διανομή λιπασμάτων, βοηθώντας έτσι στη βελτίωση της παραγωγικότητας των καλλιεργειών.

### 3. Διανομή Εμπορευμάτων:

Στον τομέα της λογιστικής και της διανομής, ρομποτικά οχήματα μπορούν να χρησιμοποιηθούν για την αυτοματοποίηση της διανομής εμπορευμάτων σε μεγάλες αποθήκες ή σε περιοχές που δυσκολεύεται η πρόσβαση των ανθρώπων.

### 4. Εξορυκτική Βιομηχανία:

Σε εξορυκτικές εφαρμογές, ρομποτικά οχήματα μπορούν να χρησιμοποιηθούν για την ανίχνευση και εξόρυξη ορυκτών πόρων σε επικίνδυνα ή δύσκολα προσβάσιμα μέρη.

### 5. Ιατρική:

Στον τομέα της ιατρικής, ρομποτικά οχήματα μπορούν να χρησιμοποιηθούν για την παράδοση φαρμάκων ή ιατρικού εξοπλισμού σε ασθενείς σε απομακρυσμένες περιοχές ή για την εκτέλεση αποστειρωμένων δειγμάτων σε εργαστήρια.

Αυτά είναι μερικά παραδείγματα των πολλών τρόπων με τους οποίους τα ρομποτικά οχήματα μπορούν να χρησιμοποιηθούν για τη βελτίωση της αποτελεσματικότητας, της ασφάλειας και της ευελιξίας σε ποικίλους τομείς.

## 9. Αξιολόγηση αποτελεσμάτων – Συμπεράσματα

Κατά την διεκπεραίωση αυτού του project διεύρυνα τις γνώσεις μου επάνω στον κλάδο των ρομποτικών οχημάτων τόσο ως προς την χρήση τους αλλά και ειδικά ως προς την κατασκευή αυτών. Κατά την διάρκεια της κατασκευής τόσο του οχήματος αυτού όσο και του λογισμικού του (software) αλλά και του εξοπλισμού (hardware) προέκυψαν πολλά προβλήματα τόσο κατασκευαστικά όσο και λειτουργικά. Το γεγονός ότι αγοράστηκε έτοιμο το όχημα δεν το καθιστά έτοιμο ως προς χρήση με την αλλαγή απλά του εξοπλισμού.

Αλλάζοντας τον εξοπλισμό αλλάχθηκε πλήρως η όλη γεωμετρία του οχήματος αλλά και το βάρος. Αυτό είχε ως αποτέλεσμα μια αναζήτηση λύσεων αρχικά στο προγραμματιστικό κομμάτι και μετέπειτα στο κατασκευαστικό.

Έπειτα αξίζει να σημειωθεί πως η δημιουργία του webapp δεν ήταν καθόλου εύκολη. Προέκυψαν πάρα πολλά προβλήματα καθώς απαιτήθηκαν ιδιαίτερες γνώσεις προγραμματισμού και γνώσεις των γλωσσών που εφαρμόστηκαν κυρίως αυτής της HTML αλλά και της Javascript.

Τέλος παρ' όλες αυτές τις προκλήσεις που δημιουργήθηκαν κατά την πορεία αυτού του project, έπειτα από πολλή υπομονή, αρκετό ψάξιμο κυρίως στο διαδίκτυο και έπειτα από πολλά Tutorial το αποτέλεσμα θα έλεγε κανείς πως είναι εκπληκτικό, λαμβάνοντας υπόψη του ότι όλα έχουν ξεκινήσει από το μηδέν. Χειρίζοντας κανείς αυτό το ρομπότ του δίνεται τελικά η δυνατότητα τηλεχειρισμού μέσω οποιασδήποτε συσκευής με την χρήση του livestreaming (εφόσον το επιθυμεί) και κατασυνέπεια, λήψη φωτογραφίας και βίντεο. Έπειτα κατασκευαστικά το όχημα είναι ικανό να μεταφέρει αντικείμενα βάρους έως 1.5kg.

Σίγουρα έχει πολλές ευκαιρίες περεταίρω εξέλιξης. Αναβαθμίζοντας τον βασικό εξοπλισμό (RaspberryPi) μπορεί παρέχεται η δυνατότητα τοποθέτησης servo για περιστροφική κάμερα ως «κεφάλι» οδηγού.

Επιπλέον με την προσθήκη κατάλληλου εξοπλισμού επάνω στο Rpi δίνεται η δυνατότητα περισσότερων αναμονών σύνδεσης για την ένταξη επιπλέον λειτουργιών μιας και το η συγκεκριμένη κατασκευή θα αποτελέσει το πείραμα όσο αναφορά τον τηλεχειρισμό ο οποίος μελλοντικά θα εφαρμοστεί σε τρακτεράκι για διάφορες ενέργειες.



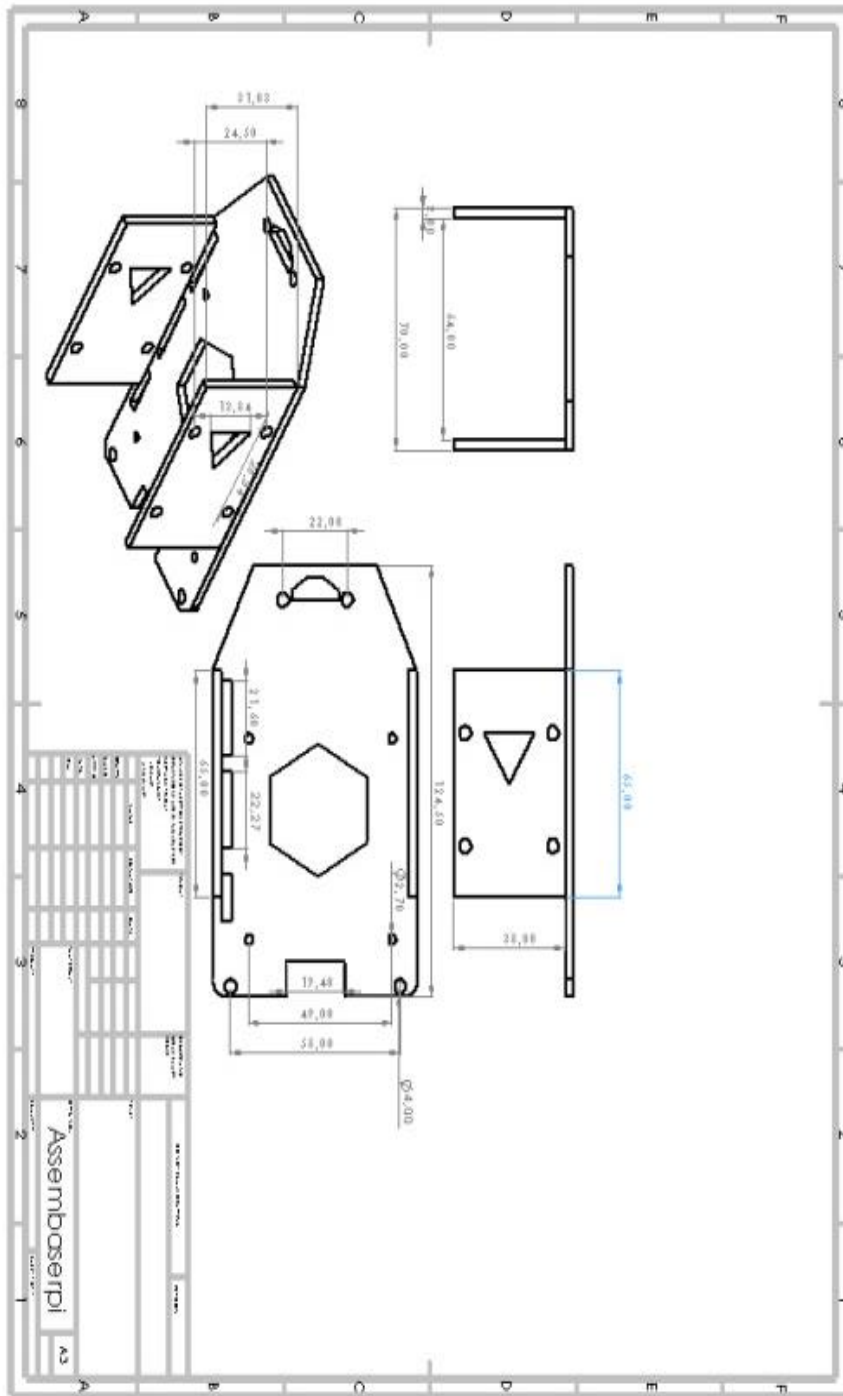
## 10. Βιβλιογραφία

- 1) Ιωάννης Καλομοίρος, Ιωάννης Βουρβουλάκης, Σπυρίδων Καζαρλής“Ενσωματωμένα συστήματα”Σημειώσεις διδασκόντων καθηγητών του ΠΜΣ Ρομποτικής του ΔΙ.ΠΑ.Ε.
- 2) Σπυρίδων Καζαρλής, “Σημειώσεις μαθήματος Αυτόνομα Ρομποτικά Συστήματα”.
- 3) Robots".Roland SIEGWART, Illah R. NOURBAKHSI, Davide SCARAMUZZA  
"Introduction to Autonomous Mobile
- 4) ΜΑΘΙΟΥΣ ΕΡΙΚ “Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΡΥΘΜΟΝ ΜΕΣΑ ΑΠΟ ΠΑΡΑΔΕΙΓΜΑΤΑ ΚΑΙ ΑΣΚΗΣΕΙΣ” Εκδόσεις Δίσιγμα
- 5) JohnCraig“Εισαγωγή στη Ρομποτική”, (μετάφραση), 978-960-418-160-5, ΕΔΟΣΕΙΣ ΤΖΙΟΛΑ ΑΕ.
- 6) “Ρομποτική”, Δ. Εμίρης, Δ. Κουλουριώτης, Εκδόσεις ΣΕΛΚΑ – 4Μ ΕΠΕ.
- 7) WilliamL. Whittaker"Ρομποτική: Ιστορία και Προοπτικές"
- 8) UweZimmer"Αυτόνομα Οχήματα: Πρόοδος στην Επιστήμη και την Τεχνολογία"
- 9) Smith, P., 2018. Robotic Lawn Mowers: A Review. Journal of Gardening Technology
- 10)Johnson, A., 2019. Advances in Autonomous Garden Maintenance. Robotics Today.
- 11)Greenfield, L., 2020. Sustainable Practices in Robotic Lawn Care. Environmental Robotics
- 12)[https://www.robotique.tech/robotics/controlling-a-car-using-a-smartphone-and-microbit/?fbclid=IwAR2kv1kq0\\_ZBmZyU0fbjMvD6KKw\\_AaXepOm0owJbluXB2OKdxKqJaJc-gh8](https://www.robotique.tech/robotics/controlling-a-car-using-a-smartphone-and-microbit/?fbclid=IwAR2kv1kq0_ZBmZyU0fbjMvD6KKw_AaXepOm0owJbluXB2OKdxKqJaJc-gh8)
- 13)<https://www.mobileye.com/blog/history-autonomous-vehicles-renaissance-to-reality/>
- 14)<https://www.instructables.com/FPV-RC-Car/>
- 15)[www.Createjs.com](http://www.Createjs.com)
- 16)<https://codepen.io/jiffy/pen/zrqwON> [1]
- 17)<http://users.sch.gr/tsibinos/raspberry-pi/>
- 18)[https://www.youtube.com/watch?v=aE\\_J7B-O4VQ](https://www.youtube.com/watch?v=aE_J7B-O4VQ)
- 19)[https://www.youtube.com/watch?v=aE\\_J7B-O4VQ](https://www.youtube.com/watch?v=aE_J7B-O4VQ)
- 20)[https://www.youtube.com/watch?v=-Jn\\_WhLMq9U](https://www.youtube.com/watch?v=-Jn_WhLMq9U)
- 21)<https://www.youtube.com/watch?v=tyY7AN132Xs>
- 22)<https://www.youtube.com/watch?v=XIk7Osr9bNQ>

- 23) <https://www.youtube.com/watch?v=HfQ7IhhgDOK>
- 24) [https://www.youtube.com/watch?v=Fx8-Mnk4sTo&list=RDCMUC6IQJHk\\_CKlb8NKffGIrtAQ&index=1](https://www.youtube.com/watch?v=Fx8-Mnk4sTo&list=RDCMUC6IQJHk_CKlb8NKffGIrtAQ&index=1)
- 25) <https://www.youtube.com/watch?v=hO1fFNvt2CM>
  
- 26) <https://www.youtube.com/watch?v=l2RNq90l2nc>
- 27) <https://www.youtube.com/watch?v=Pqs-3GgWW3s>
- 28) <https://www.youtube.com/watch?v=GkXH8ZeeIKY>
- 29) <https://www.nasa.gov/>
- 30) <https://spectrum.ieee.org/>
- 31) <https://gocoding.org/el/>
- 32) <https://www.youngwonks.com/>
- 33) <https://linuxhint.com/>
- 34) <https://roboticsbackend.com/>
- 35) <https://www.youtube.com/>
- 36) <https://www.wikipedia.org/>
- 37) <https://www.arduino.cc/>
- 38) <https://www.raspberrypi.com/>

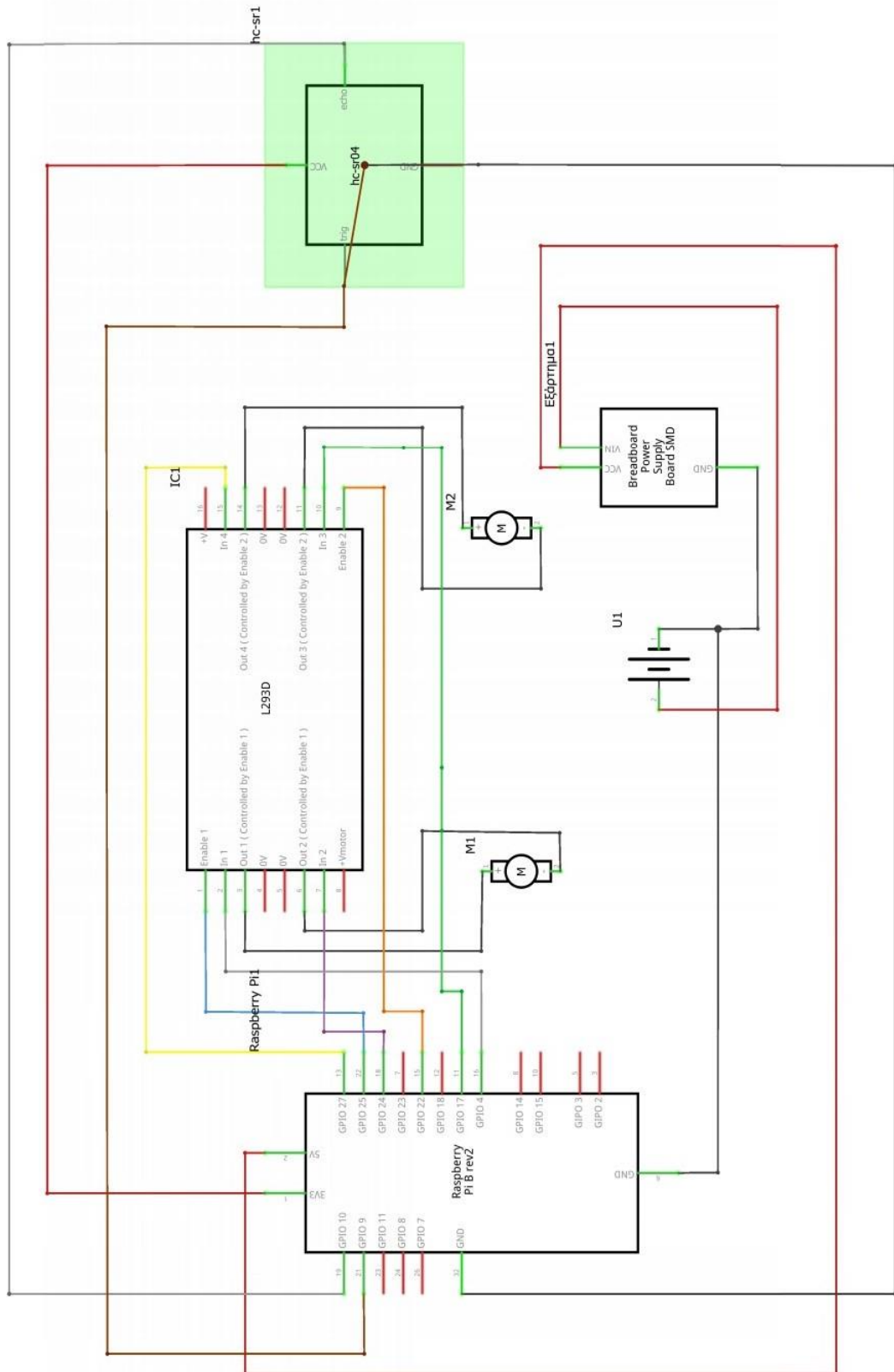






Σχήμα 48: Σχέδιο κύριου σώματος οχήματος και βάσης Rpri

## 11. ΠΑΡΑΡΤΗΜΑ - Β (Ηλεκτρολογικό Διάγραμμα)



\*Σχήμα 49: Απεικόνιση ηλεκτρικού διαγράμματος

## 11. ΠΑΡΑΡΤΗΜΑ - Γ (Κώδικας)

### Κώδικας Python

```
from flask import Flask, render_template, request, Response, make_response, send_file
from io import BytesIO
import cv2
import RPi.GPIO as GPIO
import threading
import time
import sched

app = Flask(__name__)

Motor1A = 16
Motor1B = 18
Motor1E = 22

Motor2A = 11
Motor2B = 13
Motor2E = 15

#TRIG = 37
#ECHO = 35

CAMERA_SHRINK_RATIO = 0.25
CAMERA_FPS = 8
CAMERA_FRAME_RATE = 8
CAMERA_WIDTH = 640
CAMERA_HEIGHT = 480

GPIO.setmode(GPIO.BOARD)

GPIO.setup(Motor1A, GPIO.OUT)
GPIO.setup(Motor1B, GPIO.OUT)

GPIO.setup(Motor1E, GPIO.OUT)

GPIO.setup(Motor2A, GPIO.OUT)
GPIO.setup(Motor2B, GPIO.OUT)
GPIO.setup(Motor2E, GPIO.OUT)

#GPIO.setup(TRIG, GPIO.OUT)
#GPIO.setup(ECHO, GPIO.IN)

pwm1 = GPIO.PWM(Motor1E, 5000)
pwm2 = GPIO.PWM(Motor2E, 1000)

camera = cv2.VideoCapture(0)
camera.set(cv2.CAP_PROP_FRAME_WIDTH, CAMERA_WIDTH)
camera.set(cv2.CAP_PROP_FRAME_HEIGHT, CAMERA_HEIGHT)
#camera.set(cv2.CAP_PROP_FPS, CAMERA_FPS)
#camera.set(cv2.CAP_PROP_FRAME_COUNT, CAMERA_FRAME_RATE)

currentFrameBytes = None
availableFrames = []
isRecording = False
isStreaming = False

scheduler = sched.scheduler(time.time, time.sleep)

#def get_distance(trigger_pin, echo_pin):
#    #GPIO.output(trigger_pin, False)
```

```

#time.sleep(0.5)

#GPIO.output(trigger_pin, True)
#time.sleep(0.00001)
#GPIO.output(trigger_pin, False)

#timeout = time.time() + 1
#while GPIO.input(echo_pin) == 0 and time.time() < timeout:
    #pulse_start_time = time.time()

#timeout = time.time() + 1
#while GPIO.input(echo_pin) == 1 and time.time() < timeout:
    #pulse_end_time = time.time()

#pulse_duration = pulse_end_time - pulse_start_time
#distance = pulse_duration * 17150
#distance = round(distance, 2)

#return distance

def gen(camera):
    global currentFrameBytes, availableFrames, isStreaming

    while isStreaming == True:
        _, frame = camera.read()
    #    resizedFrame = cv2.resize(frame, None, CAMERA_SHRINK_RATIO, CAMERA_SHRINK_RATIO)
        _, jpeg = cv2.imencode('.jpg', frame)
    currentFrameBytes = jpeg.tobytes()
        if isRecording:
    availableFrames.append(frame)
            yield (b'--frame\n'
    b'Content-Type: image/jpeg\n\n' + currentFrameBytes + b'\n\n')

def move_forwards():
    pwm1.start(100)
    GPIO.output(Motor1A, GPIO.HIGH)
    GPIO.output(Motor1B, GPIO.LOW)
    GPIO.output(Motor1E, GPIO.HIGH)

def move_backwards():
    pwm1.start(100)
    #distance = get_distance(TRIG, ECHO)
    #if distance > 10:
    GPIO.output(Motor1A, GPIO.LOW)
    GPIO.output(Motor1B, GPIO.HIGH)
    GPIO.output(Motor1E, GPIO.HIGH)
    #else:
        #GPIO.output(Motor1A, GPIO.LOW)
        #GPIO.output(Motor1B, GPIO.LOW)
        #GPIO.output(Motor1E, GPIO.LOW)

def move_left():
    pwm2.start(90)
    GPIO.output(Motor2A, GPIO.HIGH)
    GPIO.output(Motor2B, GPIO.LOW)
    GPIO.output(Motor2E, GPIO.HIGH)

def move_right():
    pwm2.start(90)
    GPIO.output(Motor2A, GPIO.LOW)
    GPIO.output(Motor2B, GPIO.HIGH)
    GPIO.output(Motor2E, GPIO.HIGH)

def move_forwardsleft():

```



```

    pwm1.start(100)
    pwm2.start(100)
GPIO.output(Motor1A, GPIO.HIGH)
GPIO.output(Motor1B, GPIO.LOW)
GPIO.output(Motor1E, GPIO.HIGH)
GPIO.output(Motor2A, GPIO.HIGH)
GPIO.output(Motor2B, GPIO.LOW)
GPIO.output(Motor2E, GPIO.HIGH)

def move_forwardsright():
    pwm1.start(100)
    pwm2.start(100)
GPIO.output(Motor1A, GPIO.HIGH)
GPIO.output(Motor1B, GPIO.LOW)
GPIO.output(Motor1E, GPIO.HIGH)
GPIO.output(Motor2A, GPIO.LOW)
GPIO.output(Motor2B, GPIO.HIGH)
GPIO.output(Motor2E, GPIO.HIGH)

def move_backwardsleft():
    pwm1.start(100)
    pwm2.start(100)
GPIO.output(Motor1A, GPIO.LOW)
GPIO.output(Motor1B, GPIO.HIGH)
GPIO.output(Motor1E, GPIO.HIGH)
GPIO.output(Motor2A, GPIO.HIGH)
GPIO.output(Motor2B, GPIO.LOW)
GPIO.output(Motor2E, GPIO.HIGH)

def move_backwardsright():
    pwm1.start(100)
    pwm2.start(100)
GPIO.output(Motor1A, GPIO.LOW)
GPIO.output(Motor1B, GPIO.HIGH)
GPIO.output(Motor1E, GPIO.HIGH)
GPIO.output(Motor2A, GPIO.LOW)
GPIO.output(Motor2B, GPIO.HIGH)
GPIO.output(Motor2E, GPIO.HIGH)

def move_stop():
    pwm1.stop()
    pwm2.stop()
GPIO.output(Motor1A, GPIO.LOW)
GPIO.output(Motor1B, GPIO.LOW)
GPIO.output(Motor1E, GPIO.LOW)
GPIO.output(Motor2A, GPIO.LOW)
GPIO.output(Motor2B, GPIO.LOW)
GPIO.output(Motor2E, GPIO.LOW)

def cleanup():
GPIO.cleanup()

@app.route("/")
def index():
    return render_template("index_combined.html")

@app.route('/video_feed')
def video_feed():
    return Response(gen(camera), mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/start-streaming', methods=["POST"])
def startStreaming():
    global isStreaming
    isStreaming = True

```

```

    return "Streaming started"

@app.route('/stop-streaming', methods=["POST"])
def stopStreaming():
    global currentFrameBytes, availableFrames, isRecording, isStreaming
    isRecording = False
    isStreaming = False
    currentFrameBytes = None
    availableFrames = []
    return "Streaming stopped"

@app.route('/start-recording', methods=["POST"])
def startRecording():
    global isRecording

    if isRecording == True:
        return "Already recording"

    isRecording = True
    return "Recording started"

@app.route('/stop-recording', methods=["POST"])
def stopRecording():
    global availableFrames, isRecording

    if isRecording == False:
        return "Camera is not recording"

    # Define the codec and create a VideoWriter object
    width = int(camera.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(camera.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fourcc = cv2.VideoWriter_fourcc(*'mp4v') # You can use other codecs like 'XVID', 'MJPG', etc.

    video_writer = cv2.VideoWriter('video.mp4', fourcc, 20, (width, height))

    for frame in availableFrames:
        video_writer.write(frame)

    video_writer.release()

    availableFrames = []
    isRecording = False

    return send_file('video.mp4')

@app.route("/get-photo", methods=["POST"])
def getPhoto():
    if currentFrameBytes != None:
        response = make_response(currentFrameBytes)
        response.headers.set('Content-Type', 'image/jpeg')
        return response

    return make_response("Camera is not currently operating", 403)

@app.route("/move", methods=["POST"])
def move_route():
    move_stop()
    direction = request.form["direction"]

    if direction == "forwards":
        move_forwards()
    elif direction == "backwards":
        move_backwards()
    elif direction == "left":

```

```

move_left()
elif direction == "right":
move_right()
elif direction == "forwardsleft":
move_forwardsleft()
elif direction == "forwardsright":
move_forwardsright()
elif direction == "backwardsleft":
move_backwardsleft()
elif direction == "backwardsright":
move_backwardsright()

    return "Moving " + direction

@app.route("/macro-commands", methods=["POST"])
def executeMacrocommands():
move_stop()

    command = request.form["command"]
    duration = request.form["duration"]
    print(duration)

    if not duration.isnumeric():
        return make_response("Invalid duration", 400)

    duration = int(duration)

    if command == "forwards":
move_forwards()
elif command == "backwards":
move_backwards()
elif command == "stop":
move_stop()
elif command == "left":
move_left()
elif command == "right":
move_right()
elif command == "forwardsleft":
move_forwardsleft()
elif command == "forwardsright":
move_forwardsright()
elif command == "backwardsleft":
move_backwardsleft()
elif command == "backwardsright":
move_backwardsright()
        if duration > 0:
scheduler.enter(duration, 1, move_stop, ())
scheduler.run()
        else:
move_stop()
        return "Macro commands received"

@app.route("/stop", methods=["POST"])
def stop_route():
move_stop()
    return "Stop"

@app.route("/cleanup", methods=["POST"])
def cleanup_route():
cleanup()
    return "Clean up"

if __name__ == "__main__":
app.run(host="0.0.0.0", port=5000)

```

## Κώδικας HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<script src="https://code.createjs.com/1.0.0/createjs.min.js"></script>
<script src="/static/js/easytimer.min.js"></script>
<script src="/static/js/hammer.min.js"></script>
<script src="/static/js/js-file-download.js"></script>
<script src="/static/js/main.js"></script>
<link rel="stylesheet" href="/static/css/main.css">
<title>Robot Control</title>
</head>

<body>
<h1>Robot Control</h1>
<div class="container">
<div class="joystick-wrapper">
<canvas id="joystick" height="300" width="300"></canvas>
</div>
<div class="stream-content">

<span id="stream-timer" class="timer"></span>
<div class="camera-controls">
<label class="switch">
<input type="checkbox" onclick="toggleStreaming(this, event)">
<span class="slider round"></span>
</label>
<div class="camera-buttons">
<button type="button" class="photo-btn" disabled onclick="capturePhoto(event)">
<imgsrc="/static/images/camera.png" alt="camera">
</button>
<button type="button" class="video-btn" disabled onclick="toggleVideoRecording(event)">

</button>
</div>
</div>
</div>
<div class="macro-command-container">
<table class="command-table">
<thead>
<tr>
<td>
<span style="font-weight: bold;">Commands</span>
</td>
</tr>
</thead>
<tbody>
<tr>
<td>
forwards
</td>
</tr>
<tr>
<td>
backwards
</td>
</tr>
<tr>
<td>
stop
</td>
</tr>
</tbody>
</table>
</div>
```

```

</tr>
<tr>
<td>
left
</td>
</tr>
<tr>
<td>
right
</td>
</tr>
<tr>
<td>
forwardsleft
</td>
</tr>
<tr>
<td>
forwardsright
</td>
</tr>
<tr>
<td>
backwardsleft
</td>
</tr>
<tr>
<td>
backwardsright
</td>
</tr>
</tbody>
</table>
<form onsubmit="sendMacroCommands(this, event)">
<label class="label" for="macro-commands">Usage: command,duration</label>
<textarea id="macro-commands" name="macro-commands" rows="20" required></textarea>
<br>
<button id="macro-command-submit" class="macro-command-submit-btn" type="submit">Send
commands</button>
</form>
</div>
</div>

<script>
initialize();
</script>
</body>

</html>

```

## Κώδικας JavaScript

```
const BASE_URL = "";

// See https://hammerjs.github.io/api/
const DIRECTION = {
  NONE: 1,
  LEFT: 2,
  RIGHT: 4,
  UP: 8,
  DOWN: 16,
  HORIZONTAL: 6,
  VERTICAL: 24,
  ALL: 30,
  // Sums
  UP_LEFT: 10,
  UP_RIGHT: 12,
  DOWN_LEFT: 18,
  DOWN_RIGHT: 20
};

var currentJoyStickDirection = DIRECTION.NONE;
var isStreaming = false;
var isRecording = false;
var streamTimer;

function initialize() {
  initializeJoyStick();
  initializeVideoRecordingTimer();
}

// See: https://codepen.io/jiffy/pen/zrqwON
function initializeJoyStick() {
  // easal stuff goes hur
  var xCenter = 150;
  var yCenter = 150;
  var stage = new createjs.Stage('joystick');

  var psp = new createjs.Shape();
  psp.graphics.beginFill("#333333").drawCircle(xCenter, yCenter, 50);

  psp.alpha = 0.25;

  var vertical = new createjs.Shape();
  var horizontal = new createjs.Shape();

  stage.addChild(psp);
  stage.addChild(vertical);
  stage.addChild(horizontal);
  createjs.Ticker.framerate = 60;
  createjs.Ticker.addEventListener('tick', stage);
  stage.update();

  var joystickElement = document.getElementById("joystick");

  // create a simple instance
  // by default, it only adds horizontal recognizers
  var mc = new Hammer(joystickElement);
  mc.get("pan").set({ direction: Hammer.DIRECTION_ALL });

  mc.on("panstart", function (event) {
    var { angle } = event;
    var direction = getDirectionByAngle(angle);

    xCenter = joystickElement.offsetWidth - joystickElement.clientWidth;
```

```

yCenter = joystickElement.offsetHeight - joystickElement.clientHeight;

psp.alpha = 0.5;
stage.update();

    if (currentJoyStickDirection !== direction) {
currentJoyStickDirection = direction;
onDirectionChange(direction);
    }
});

mc.on("panmove", function (event) {
    var { angle, distance } = event;
    var coords = calculateCoords(angle, distance);
    var direction = getDirectionByAngle(angle);

psp.x = coords.x;
psp.y = coords.y;
psp.alpha = 0.5;

stage.update();

    if (currentJoyStickDirection !== direction) {
currentJoyStickDirection = direction;
onDirectionChange(direction);
    }
});

mc.on("panend", function (event) {
psp.alpha = 0.25;
createjs.Tween.get(psp).to({ x: xCenter, y: yCenter }, 750, createjs.Ease.elasticOut);

    if (currentJoyStickDirection !== DIRECTION.NONE) {
currentJoyStickDirection = DIRECTION.NONE;
onDirectionChange(DIRECTION.NONE);
    }
});
}

function initializeVideoRecordingTimer() {
    var streamTimerEl = document.getElementById("stream-timer");

streamTimer = new easytimer.Timer();

    if (streamTimerEl) {
streamTimer.addEventListener('started', function (e) {
streamTimerEl.innerHTML = streamTimer.getTimeValues().toString();
    });

streamTimer.addEventListener('stopped', function (e) {
streamTimerEl.innerHTML = "";
    });

streamTimer.addEventListener('secondsUpdated', function (e) {
streamTimerEl.innerHTML = streamTimer.getTimeValues().toString();
    });
}

function calculateCoords(angle, distance) {
    var coords = {};
    distance = Math.min(distance, 100);
    var rads = (angle * Math.PI) / 180.0;

```

```

coords.x = distance * Math.cos(rads);
coords.y = distance * Math.sin(rads);

return coords;
}

function getDirectionByAngle(angle) {
    var direction;

    if (angle < 112.5 && angle > 67.5) {
        direction = DIRECTION.DOWN;
    } else if (angle <= -112.5 && angle >= -157.5) {
        direction = DIRECTION.UP_LEFT;
    } else if (angle >= -67.5 && angle <= -22.5) {
        direction = DIRECTION.UP_RIGHT;
    } else if (angle < -157.5 && angle >= -179.9 || angle <= 180 && angle > 157.5) {
        direction = DIRECTION.LEFT;
    } else if (angle > -22.5 && angle < 0 || angle >= 0 && angle < 22.5) {
        direction = DIRECTION.RIGHT;
    } else if (angle <= 157.5 && angle >= 112.5) {
        direction = DIRECTION.DOWN_LEFT;
    } else if (angle >= 22.5 && angle <= 67.5) {
        direction = DIRECTION.DOWN_RIGHT;
    } else {
        direction = DIRECTION.UP;
    }

    return direction;
}

async function onDirectionChange(direction) {
    if (direction === DIRECTION.NONE) {
        let response = await fetch(BASE_URL + "/stop", {
            method: "POST"
        });
        let text = await response.text();
        console.log(text);
    } else {
        let piDirection;
        switch (direction) {
            case DIRECTION.DOWN:
                piDirection = "backwards";
                break;
            case DIRECTION.LEFT:
                piDirection = "left";
                break;
            case DIRECTION.RIGHT:
                piDirection = "right";
                break;
            case DIRECTION.UP_LEFT:
                piDirection = "forwardsleft";
                break;
            case DIRECTION.UP_RIGHT:
                piDirection = "forwardsright";
                break;
            case DIRECTION.DOWN_LEFT:
                piDirection = "backwardsleft";
                break;
            case DIRECTION.DOWN_RIGHT:
                piDirection = "backwardsright";
                break;
            default:
                piDirection = "forwards";
                break;
        }
    }
}

```



```

    }

    let response = await fetch(BASE_URL + "/move", {
      method: "POST",
      headers: {
        "Content-Type": "application/x-www-form-urlencoded",
      },
      body: "direction=" + piDirection,
    });
    let text = await response.text();
    console.log(text);
  }
}

async function capturePhoto(event) {
  let response = await fetch(BASE_URL + "/get-photo", {
    method: "POST",
  });

  if (response.status === 200) {
    let blob = await response.blob();
    download(blob, Date.now() + ".png");
  } else {
    let text = await response.text();
    alert(text);
  }
}

async function toggleVideoRecording(event) {
  var streamTimerEl = document.getElementById("stream-timer");

  if (isRecording) {
    if (streamTimerEl) {
      streamTimerEl.innerText = "stopping...";
    }

    let response = await fetch(BASE_URL + "/stop-recording", {
      method: "POST",
    });

    let blob = await response.blob();
    download(blob, Date.now() + ".mp4");

    let videoButtonImage = document.getElementById("video-recording-img");
    if (videoButtonImage) {
      videoButtonImage.src = "/static/images/video-start.png";
    }

    streamTimer.reset();
    streamTimer.stop();
  } else {
    if (streamTimerEl) {
      streamTimerEl.innerText = "starting...";
    }

    await fetch(BASE_URL + "/start-recording", {
      method: "POST",
    });

    let videoButtonImage = document.getElementById("video-recording-img");
    if (videoButtonImage) {
      videoButtonImage.src = "/static/images/video-stop.png";
    }
  }
}

```

```

streamTimer.start();
}

isRecording= !isRecording;
}

function toggleStreaming(element, event) {
  if (element.checked) {
    startStreaming();
  } else {
    stopStreaming();

    if (isRecording) {
      let videoButtonImage = document.getElementById("video-recording-img");
      if (videoButtonImage) {
        videoButtonImage.src = "/static/images/video-start.png";
      }
    }

    streamTimer.reset();
    streamTimer.stop();
  }
}

async function startStreaming() {
  var element = document.getElementById("stream-view");
  if (element) {
    await fetch(BASE_URL + "/start-streaming", {
      method: "POST",
    });

    element.onload = async event => {
      let buttons = document.querySelectorAll(".camera-controls .camera-buttons button");
      for (let button of buttons) {
        button.disabled = false;
      }

      isStreaming = true;
      element.onload = null;
    };

    element.src = BASE_URL + "/video_feed";
  }
}

function stopStreaming() {
  var element = document.getElementById("stream-view");
  if (element) {
    element.onload = async event => {
      await fetch(BASE_URL + "/stop-streaming", {
        method: "POST",
      });

      let buttons = document.querySelectorAll(".camera-controls .camera-buttons button");
      for (let button of buttons) {
        button.disabled = true;
      }

      isStreaming = false;
      element.onload = null;
    };
    element.src = "/static/images/no-camera.png";

  }
}

```

```

}

async function sendMacroCommands(form, event) {
event.preventDefault();

var textArea = form.elements["macro-commands"];
var button = form.elements["macro-command-submit"];
if (textArea) {
let content = textArea.value;
let commandData = content.split("\n");

textArea.disabled = true;
button.disabled = true;

for (let data of commandData) {
let values = data.split(",");
if (values.length !== 2) {
alert("Please type command and duration separated by comma");
return;
}

let commandName = values[0].trim();
let duration = values[1].trim();

let response = await fetch(BASE_URL + "/macro-commands", {
method: "POST",
headers: {
"Content-Type": "application/x-www-form-urlencoded",
},
body: `command=${commandName}&duration=${duration}`,
});
let text = await response.text();
console.log(text);
}

textArea.value = "";
textArea.disabled = false;
button.disabled = false;

alert("Commands executed successfully!");
}
}

```

## Σύντομο Βιογραφικό Σημείωμα

|                              |                                                                                               |
|------------------------------|-----------------------------------------------------------------------------------------------|
| Προσωπικά στοιχεία:          | Καμνιάτσος Γεώργιος                                                                           |
| Βασική εκπαίδευση:           | 10 <sup>ο</sup> ΓΕΛ Θεσσαλονίκης                                                              |
| Σπουδές:                     | Τμήμα Μηχανολόγων Μηχανικών (Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κεντρικής Μακεδονίας και Θράκης) |
| Επαγγελματική δραστηριότητα: | Ιδιωτικός υπάλληλος ως Μηχανόλογος<br>Μηχανικός σε Μεγάλη Ανώνυμη Εμπορική<br>Εταιρεία        |



ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΣΕΡΡΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ,  
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ