



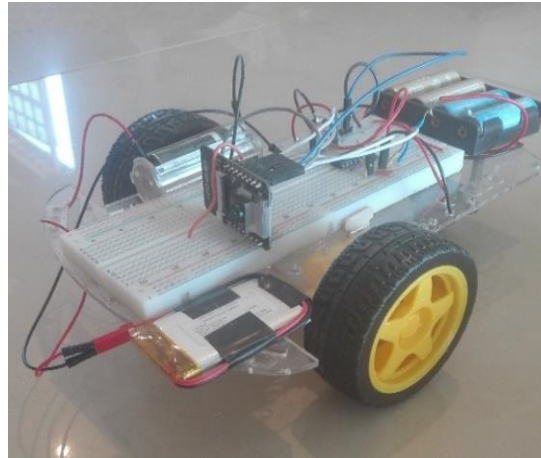
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ  
ΠΑΡΑΓΩΓΗΣ

ΘΕΜΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

ΑΝΑΠΤΥΞΗ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΜΕ ΑΣΥΡΜΑΤΟ, ΑΠΟΜΑΚΡΥΣΜΕΝΟ  
ΕΛΕΓΧΟ ΚΑΙ ΔΥΝΑΤΟΤΗΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕΣΩ ΔΙΑΔΙΚΤΥΟΥ



ΟΝΟΜΑ ΦΟΙΤΗΤΗ:

ΚΟΡΟΜΗΛΑΣ ΓΕΩΡΓΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΑΒΡΑΑΜ ΧΑΤΖΟΠΟΥΛΟΣ

ΑΙΓΑΛΕΩ, ΙΟΥΛΙΟΣ 2021

Η παρούσα διπλωματική εργασία εγκρίθηκε ομόφωνα από την τριμελή εξεταστική επιτροπή, η οποία ορίστηκε από την Γ.Σ. του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, σύμφωνα με το νόμο και τον εγκεκριμένο Οδηγό Σπουδών του τμήματος.

**Επιβλέπων:** Χατζόπουλος Αβραάμ  
Λέκτορας Εφαρμογών

**Επιτροπή Αξιολόγησης:**

.....  
Χατζόπουλος Αβραάμ  
Λέκτορας Εφαρμογών

.....  
Σκλαβούνου Ελένη Ορσαλία  
Λέκτορας Εφαρμογών

.....  
Δρόσος Χρήστος  
ΕΔΙΠ

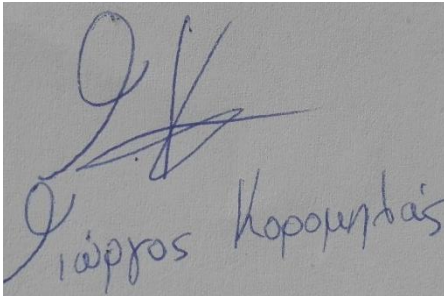
## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Κορομηλάς Γεώργιος του Νικολάου, με αριθμό μητρώου 71444292 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Γεώργιος Κορομηλάς

Θα ήθελα να ευχαριστήσω ιδιαίτερα το εκπαιδευτικό προσωπικό του τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής για την εκπαιδευτική κατάρτιση που μου παρείχε, την οικογένεια μου για την πολυετή υποστήριξη που βοήθησε να ολοκληρώσω τις σπουδές μου, και τους πιστούς μου φίλους Γιώργο Μακέδο και Άγγελο Συκιανάκη, που συμμετείχαν στις λειτουργικές δοκιμές της διπλωματικής εργασίας.

## **ΠΕΡΙΛΗΨΗ**

Η διπλωματική εργασία που ακολουθεί, πραγματεύεται την δυνατότητα υλοποίησης ενός ρομποτικού οχήματος που μπορεί να ελεγχθεί και να προγραμματιστεί απομακρυσμένα, χρησιμοποιώντας το όλο και πιο ταχύρρυθμο διαδίκτυο. Η εφαρμογή έχει σχεδιαστεί για την εύκολη και οικονομική κατασκευή της, χωρίς να απαιτείται εξειδικευμένη γνώση και εργαλεία. Το ρομποτικό όχημα εδραιώνει αμφίδρομη επικοινωνία με τον χρήστη μέσω μια ιστοσελίδας που προωθείται στο διαδίκτυο, η οποία παρέχει την πραγματική εικόνα της κίνησης αλλά και του χώρου στον οποίο βρίσκεται η κατασκευή χρησιμοποιώντας κάμερες. Επιπρόσθετα στο ίδιο περιβάλλον έχει δημιουργηθεί μια γραμμή κατάστασης όπου ο χρήστης μπορεί να συντάξει με την χρήση εικονιδίων, κώδικα που αποσκοπεί στην κίνηση και να προγραμματίσει το ρομποτικό όχημα. Τέλος, με την βοήθεια λογισμικού ο χρήστης διαθέτει την ευχέρεια, απεικόνισης της κίνησης του ρομποτικού οχήματος, σε πραγματικό χρόνο μέσω δυσδιάστατης γραφικής αναπαράστασης (2D mapping).

## **ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**

Ρομποτικό όχημα, Μικροελεγκτής ESP32-CAM, Απομακρυσμένος Έλεγχος

## **ABSTRACT**

The dissertation that follows, deals with the possibility of creating a robotic vehicle that can be controlled and programmed remotely, using the ever-fast internet. The application is designed for easy and economical construction, without requiring specialized knowledge and tools. The robotic vehicle establishes two-way communication with the user through a web page that is promoted on the internet, which provides the real image of the movement and the space where the construction is located using cameras. Additionally in the same environment a status bar has been created where the user can write with the use of icons, code intended to move and program the robotic vehicle. Finally, with the help of software the user has the ability to visualize the movement of the robotic vehicle, in real time through two-dimensional graphic representation (2D mapping).

## **KEYWORDS**

Robotic vehicle, Microcontroller ESP32-CAM, Remote control

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ.....	iii
ΠΕΡΙΛΗΨΗ.....	v
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ.....	v
ABSTRACT .....	vi
KEYWORDS .....	vi
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ .....	vii
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	ix
ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ .....	x
ΕΙΣΑΓΩΓΗ.....	1
ΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΩΝ .....	2
ΜΕΘΟΔΟΛΟΓΙΑ.....	3
ΣΧΕΔΙΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΥΛΙΚΟΥ ΜΕΡΟΥΣ (HARDWARE).....	5
Λειτουργικά Μέρη Κατασκευής .....	5
Μηχανολογικό σχέδιο .....	12
Ηλεκτρονικό κύκλωμα .....	14
Λίστα υλικών.....	15
ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΓΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ (SOFTWARE) .....	16
Διάγραμμα Ροής Ιστοσελίδας.....	16
Κώδικας Προγράμματος Ιστοσελίδας .....	17
Αρχεία HTML .....	17
Αρχείο CSS.....	33
Αρχεία Javascript.....	39
Οργάνωση αρχείων.....	64
Διάγραμμα Ροής Ρομποτικού Οχήματος.....	64
Κώδικας Προγράμματος ESP32-CAM .....	65
Έλεγχος ρομποτικού οχήματος.....	66
Έλεγχος περιφερικών καμερών .....	81
ΡΥΘΜΙΣΕΙΣ ΤΟΠΙΚΟΥ ΔΙΚΤΥΟΥ ΚΑΙ ΕΦΑΡΜΟΓΗ ΧΑΜΡΡ .....	85
Ρυθμίσεις Router/Modem.....	85
Ρυθμίσεις εφαρμογής ΧΑΜΡΡ .....	88
ΔΙΕΚΠΕΡΑΙΩΣΗ .....	91

ΑΠΟΤΕΛΕΣΜΑΤΑ .....	92
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	93
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	94



## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1 - Τεχνικά στοιχεία ESP32-CAM.....	6
Πίνακας 2 - Απόλυτες μέγιστες προδιαγραφές L293D .....	8
Πίνακας 3 - Συστημένες συνθήκες λειτουργίας L293D.....	8
Πίνακας 4 - Κατανομή ακροδεκτών του L293D.....	9
Πίνακας 5 - Χαρακτηριστικά ηλεκτροκινητήρων .....	10
Πίνακας 6 - Λίστα απαιτούμενων υλικών .....	15

## ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Εικόνα 1 - Αρχιτεκτονική δικτύου .....	3
Εικόνα 2 - Λίστα με διαθέσιμους ακροδέκτες [4] .....	6
Εικόνα 3 - Ηλεκτρικό διάγραμμα συστήματος ESP32-CAM [4].....	7
Εικόνα 4 - Πρόσωση και πίσω όψη ESP32-CAM [4] .....	7
Εικόνα 5 - Πλάγια όψη ESP32-CAM με OV2640 .....	7
Εικόνα 6 - Κάτοψη L293D [6] .....	8
Εικόνα 7 - Πίνακας οδήγησης εξόδων [6].....	9
Εικόνα 8 - Ηλεκτρικό σχέδιο L293D [6].....	9
Εικόνα 9 - Όψη κινητήρα DC [8].....	10
Εικόνα 10 - Όψη επαναφορτιζόμενης μπαταρίας 1.2V [9] .....	11
Εικόνα 11 - Όψη επαναφορτιζόμενης μπαταρίας 3.7V [10] .....	11
Εικόνα 12 - Όψη ενσύρματου τροφοδοτικού 5V [11].....	11
Εικόνα 13 - Όψη USB σε Serial μετατροπέα [12] .....	12
Εικόνα 14 - Μηχανολογικό σχέδιο(Πρόωση) με FreeCad [13].....	12
Εικόνα 15 - Μηχανολογικό σχέδιο(Εμπρόσθια όψη) με FreeCad [13].....	12
Εικόνα 16 - Μηχανολογικό σχέδιο(Ανω όψη) με FreeCad [13] .....	13
Εικόνα 17 - Μηχανολογικό σχέδιο(Κάτω όψη) με FreeCad [13] .....	13
Εικόνα 18 - Μηχανολογικό σχέδιο(Πλάγια όψη) με FreeCad [13].....	13
Εικόνα 19 - Εμπρόσθια όψη ρομποτικού οχήματος.....	13
Εικόνα 20 - Κάτω όψη ρομποτικού οχήματος .....	14
Εικόνα 21 - Πλάγια όψη ρομποτικού οχήματος.....	14
Εικόνα 22 - Ηλεκτρικό κύκλωμα ρομποτικού οχήματος [14].....	15
Εικόνα 23 - Ηλεκτρικό κύκλωμα περιφερικών καμερών [14] .....	15
Εικόνα 24 - Διάγραμμα ροής Ιστοσελίδας .....	16
Εικόνα 25 - Σελίδα index.html .....	20
Εικόνα 26 - Σελίδα ελεύθερου ελέγχου (easy-gr.html) .....	25
Εικόνα 27- Βοήθεια (easy-gr.html) .....	25
Εικόνα 28 - Σελίδα προγραμματισμένου ελέγχου (medium-gr.html) .....	32
Εικόνα 29 - Βοήθεια 2 (medium-gr.html) .....	33
Εικόνα 30 - Οργάνωση αρχείων ιστοσελίδας.....	64
Εικόνα 31 - Διάγραμμα ροής ρομποτικού οχήματος.....	65
Εικόνα 32 - Είσοδος στο router/modem.....	85
Εικόνα 33 - Στατικές διευθύνσεις ESP32-CAM .....	86
Εικόνα 34 - Port Forward ρομποτικού οχήματος.....	87
Εικόνα 35 - Port Forward ESP-32 CAMERA 1 .....	87
Εικόνα 36 - Port Forward ESP-32 CAMERA 2 .....	88
Εικόνα 37 - Port Forward Εφαρμογής.....	88
Εικόνα 38 - Τροποποίηση httpd.conf αρχείου .....	89
Εικόνα 39 - Τροποποίηση httpd-ssl.conf αρχείου .....	89
Εικόνα 40 - Ρυθμίσεις XAMPP.....	90
Εικόνα 41 - Άνοιγμα ιστοσελίδας στο διαδίκτυο .....	90

## ΕΙΣΑΓΩΓΗ

Η δημιουργία της εφεύρεσης που είναι γνωστή ως τηλέφωνο, αποτέλεσε σημαντικό στοιχείο στην ανάπτυξη της κοινωνίας αλλά και παραμένει αναπόσπαστο κομμάτι του ανθρωπίνου πολιτισμού. Η έως τότε αφάνταστη για πολλούς καινοτομία, ώθησε την επιστημονική κοινότητα στην αναζήτηση νέων εφαρμογών που έχουν παρόμοιο σκοπό, δηλαδή την αναμετάδοση πληροφορίας από απόσταση. Έχοντας φτάσει στον 21<sup>ο</sup> αιώνα, ο ρυθμός ανάπτυξης της τεχνολογία έχει εκτοξευτεί, με συνέπεια η ζωή μας να είναι άμεσα συνδεδεμένη με τις νέες καινοτομίες που εντάσσονται. Η πιο πρόσφατη και σημαντικότερη είναι το Διαδίκτυο [1], όπου αποτελεί έναν παγκόσμιο συνδετικό ιστό και επιτρέπει στους χρήστες του την ανταλλαγή πληροφοριών σε διάφορες μορφές όπως, μηνύματα, βίντεο, φωτογραφίες και αρχεία

Παράλληλα, εκτός από την επιρροή του διαδικτύου, η τεχνολογία διανύει την 4<sup>η</sup> Βιομηχανική επανάσταση [2], όπου συμπεριλαμβάνει πολλές νέες τεχνολογίες όπως τα κυβερνοφυσικά συστήματα, το διαδίκτυο των αντικειμένων και την γνωστική υπολογιστική. Η ανάγκη της βιομηχανίας για την εισαγωγή και εξέλιξη των ανωτέρω τεχνολογιών, έχει ανεβάσει τους ρυθμούς ανταγωνιστικότητας μεταξύ αυτών που τις κατέχουν και αυτών που δεν τις έχουν εισάγει ακόμα. Ο λόγος και παράλληλα το κύριο όφελος που προσφέρει η ένταξη τους, είναι η ολοένα υψηλότερη σε βαθμό αυτοματοποίηση της εργασίας, όπου μειώνει σημαντικά τις αποκλίσεις και τα σφάλματα που προέρχονται από τον ανθρώπινο παράγοντα. Εν συνέπεια όσο περνάει ο χρόνος το χάσμα αυξάνεται με τις πιο εξελιγμένες και αυτοματοποιημένες βιομηχανίες να γίνονται πιο ισχυρές και κερδοφόρες, ενώ οι βιομηχανίες που είτε δεν είχαν το κεφάλαιο για επένδυση είτε θεωρήσαν ότι δεν απαιτούνται οι νέες τεχνολογίες, να παραμένουν στάσιμες και τελικά να παύουν να λειτουργούν.

Έχοντας λάβει υπόψη τα παραπάνω, ο σκοπός της παρούσας διπλωματικής εργασίας, είναι η εύρεση, απόκτηση και αξιοποίηση των κατάλληλων γνώσεων και τεχνικών που απαιτούνται για την δημιουργία ενός χαμηλούς κόστους ρομποτικού οχήματος, το οποίο φέρει την δυνατότητα να ελέγχεται και να προγραμματίζεται εξ αποστάσεως και απομακρυσμένα μέσω της χρήσης του Διαδικτύου ως κανάλι επικοινωνίας του χρήστη με την εφαρμογή. Επιπρόσθετα πάνω στην δομή του αλλά και στο χώρο που χρησιμοποιείται θα έχει εγκαταστημένες κάμερες ώστε ο χρήστης να έχει άμεση οπτική επαφή με την τροχία που διανύει το ρομποτικό όχημα «σε πρώτο πρόσωπο», μαζί με τον χώρο που περιβάλλεται η εφαρμογή «σε τρίτο πρόσωπο». Εκτός από τις ψηφιακές κάμερες που απαιτούνται για την καθοδήγηση, θα υπάρχει και ψηφιακή αναπαράσταση της κίνησης σε πραγματικό χρόνο μέσω λογισμικού, ώστε ο χρήστης να ξέρει ακριβώς πως θα δράσει το ρομποτικό όχημα.

Η διπλωματική εργασία αυτή κατέχει εξαιρετικά μεγάλη σημασία διότι μπορεί να χρησιμοποιηθεί ως μια εισαγωγή για τα άτομα που ενδιαφέρονται ή ως ένα είδος πλατφόρμας για την δημιουργία εκπαιδευτικών, βιομηχανικών, γεωργικών, αμυντικών κ.α. εφαρμογών.

## ΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΩΝ

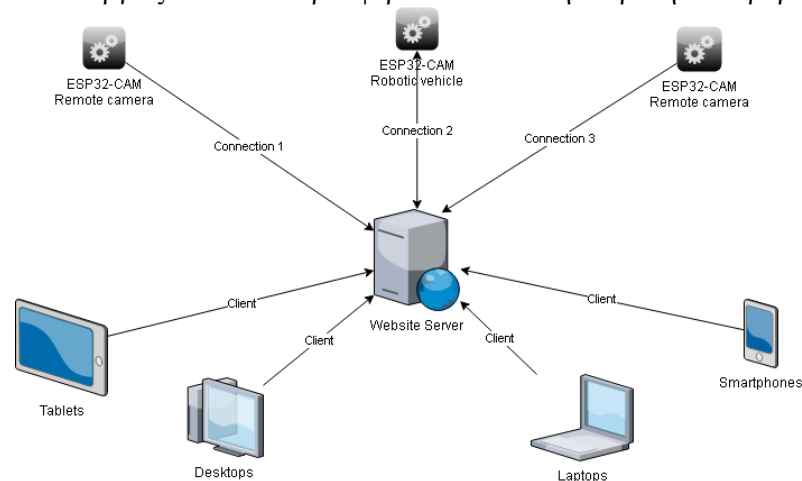
Για την επιτυχημένη υλοποίηση της διπλωματικής εργασίας θα πρέπει ο χρήστης που χρησιμοποιεί την ιστοσελίδα να ανταλλάσσει πληροφορίες με το ρομποτικό όχημα και να λαμβάνει την οπτική εικόνα των καμερών. Για να επιτευχθεί αυτό θα πρέπει να λυθούν τα εξής προβλήματα :

- 1) Σχεδίαση του δικτύου με γνώμονα την ελαχιστοποίηση των λειτουργικών σφαλμάτων.
- 2) Εύρεση του κατάλληλου πρωτοκόλλου επικοινωνίας χρήστη και συσκευής.
- 3) Μετατροπή των μηνυμάτων μέσω επεξεργασίας σε αντιληπτά δεδομένα για την αξιοποίηση τους είτε στην κίνηση είτε στην δημιουργία γραφικής αναπαράστασης.

# ΜΕΘΟΔΟΛΟΓΙΑ

Η εύρεση των παρακάτω λύσεων αποτέλεσε σημαντικό δομικό στοιχείο στην υλοποίηση της διπλωματικής εργασίας, χωρίς όμως να υφίστανται μοναδικές ή βέλτιστες.

- 1) Κατά τον σχεδιασμό του δικτύου, λήφθηκε σημαντικά υπόψη η ιεραρχία της δομής της επικοινωνίας. Ως κεντρικό κομβικό στοιχείο ορίστηκε μια ιστοσελίδα, για να ελέγχει τις υπόλοιπες συσκευές που βρίσκονται στο δίκτυο και είναι ανεξάρτητες η μια από την άλλη. Αυτό έγινε σκόπιμα για να μπορεί ο χρήστης να χειρίζεται ξεχωριστά τις λειτουργικές συσκευές και ταυτόχρονα να μην υπάρχει γενικό λειτουργικό σφάλμα σε περίπτωση κάποιας δυσλειτουργίας. Επιπλέον προσφέρει ευκολία στην εύρεση του προβλήματος.



Εικόνα 1 - Αρχιτεκτονική δικτύου

- 2) Η εδραίωση επικοινωνίας μεταξύ χρήστη και κατασκευής αποτελεί το σημαντικότερο κομμάτι της εργασίας. Η δυνατότητα αποστολής και αποδοχής μηνυμάτων στα πλαίσια του διαδικτύου γίνεται από πολλά πρωτόκολλα επικοινωνίας, με γνώμονα το είδος της λειτουργίας. Στην παρούσα περίπτωση, η ιστοσελίδα παρέχεται από ένα Server [3] και οι χρήστες αποτελούν τους client [4], συνεπώς η επικοινωνία γίνεται μεταξύ client και ρομποτικού οχήματος. Λόγο του είδους της επικοινωνίας, οι θύρες που χρησιμοποιούνται είναι δυναμικές και δεν είναι εύκολο να δημιουργηθεί σύνδεση εφόσον κάθε φορά που ένας client στέλνει μήνυμα εναλλάσσεται και η θύρα αποστολής. Για να λυθεί αυτό το πρόβλημα, χρησιμοποιήθηκε το πρωτόκολλο web socket [5], το οποίο επιτρέπει την αλληλεπίδραση μεταξύ ενός προγράμματος περιήγησης ιστού (browser [6]) και ενός διακομιστή ιστού (Server), παρέχοντας έναν τυποποιημένο τρόπο αποστολής περιεχόμενου στον χρήστη, επιτρέποντας τη μετάδοση μηνυμάτων εμπρός και πίσω, παράλληλα διατηρώντας τη σύνδεση ανοιχτή.
- 3) Τέλος, έχοντας καταφέρει την αμφίδρομη επικοινωνία με την χρήση του παραπάνω πρωτόκολλου θα αναλυθεί το σκεπτικό της ανάλυσης των δεδομένων. Ο τύπος των μηνυμάτων που ανταλλάσσονται είναι συμβολοσειρά (string) και αυτό επιτρέπει διάφορους τρόπους επεξεργασίας, Η μέθοδος που επιλέχθηκε, είναι η αποστολή μια «συμπυκνωμένης» συμβολοσειράς, όπου το λογισμικό:

A) Αρχικά ψάχνει μια συγκεκριμένη φράση για την επιλογή της κατάστασης λειτουργίας του ρομποτικού οχήματος. Η φράση μπορεί να είναι είτε «b=» είτε «program=». Εφόσον ανιχνευθεί η φράση, αφαιρείται από την συμβολοσειρά.

B) Μετέπειτα διαβάζει τους τέσσερις πρώτους χαρακτήρες και ανιχνεύει το είδος της κίνησης. Οι χαρακτήρες μπορεί να είναι «Forw», «Back», «Left», «Righ» και υποδεικνύουν τις κινήσεις εμπρός, πίσω, αριστερά και δεξιά. Ακολουθεί μετά η αφαίρεση των απαραίτητων χαρακτήρων από την συμβολοσειρά.

Γ) Συνεχίζει με το διάβασμα ενός χαρακτήρα όπου αποτελεί έναν δείκτη του πλήθους των χαρακτήρων που συντάσσουν το επόμενο στοιχείο. Ο χαρακτήρας αφαιρείται και συνεχίζει το λογισμικό.

Δ) Διαβάζοντας το πλήθος των επομένων χαρακτήρων το λογισμικό συλλεγεί την πληροφορία της διάρκειας της κίνησης. Οι χαρακτήρες επομένως αφαιρούνται.

Ε) Τέλος, το λογισμικό έχει αποθηκεύσει το είδος και την διάρκεια της επιθυμητής κίνησης όπου και την εκτελεί. Μετά το πέρας της εκτέλεσης, το λογισμικό αρχίζει ξανά από την διαδικασία B) για να εκτελέσει την επόμενη επιθυμητή κίνηση, μέχρις ότου να παραμείνει μόνο ο χαρακτήρας «e » που σηματοδοτεί το τέλος του προγράμματος.

Παρακάτω δίνεται παράδειγμα για την κατανόηση της επεξεργασίας:

Έστω μήνυμα με συμβολοσειρά «program=Forward3700Left41000Backward3999e»

Βήμα 1:Εύρεση του «program=», αρά το λογισμικό απευθύνεται στο προγραμματιστικό περιβάλλον. Η συμβολοσειρά γίνεται «Forward3700Left41000Backward3999e».

Βήμα 2:Διάβασμα τεσσάρων πρώτων χαρακτήρων βγάζει «Forw», άρα απευθύνεται σε εμπρόσθια κίνηση. Η συμβολοσειρά γίνεται «3700Left41000Backward3999e».

Βήμα 3: Διάβασμα του δείκτη. Ο δείκτης αποτελείται από τον αριθμό τρία, συνεπώς οι επόμενοι τρεις χαρακτήρες πρέπει να διαβαστούν. Η συμβολοσειρά μετατρέπεται σε «700Left41000Backward3999e».

Βήμα 4: Διάβασμα των τριών χαρακτήρων αποδίδει την πληροφορία 700. Έχοντας λάβει την διάρκεια το λογισμικό δίνει εντολή εμπρόσθιας κίνησης για 700ms. Επίσης η συμβολοσειρά μετατρέπεται σε «Left41000Backward3999e».

Βήμα 5: Εφόσον η διάρκεια της κίνησης ολοκληρώθηκε, το λογισμικό ξαναδιαβάζει τα τέσσερα πρώτα στοιχεία και συνεχίζει την ίδια διαδικασία έως η συμβολοσειρά να αποτελείται μόνο από το «e ».

Με την ίδια λογική γίνεται και η δυσδιάστατη σχεδίαση στην ιστοσελίδα, με την διαφορά ότι η συμβολοσειρά που εισέρχεται ως μήνυμα ξεκινάει με «DRAW=».

## ΣΧΕΔΙΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΥΛΙΚΟΥ ΜΕΡΟΥΣ (HARDWARE)

Αρχικά πρέπει να αναφερθεί ότι για την κατασκευή και ανάπτυξη του ρομποτικού οχήματος έχουν τεθεί κάποιοι περιορισμοί οι οποίοι συμβάλουν σημαντικά στον σχεδιασμό αλλά και στις δυνατότητες που μπορεί να προσφέρει η κατασκευή. Αυτοί είναι οι εξής :

- 1) Το συνολικό κόστος της κατασκευής να περιοριστεί έως τα 50 ευρώ.
- 2) Να μπορεί να κατασκευαστεί από οποιονδήποτε ανεξαρτήτως τις γνώσεις που κατέχει.
- 3) Ο μικροεπεξεργαστής να έχει την δυνατότητα να επικοινωνεί με το διαδίκτυο.
- 4) Να αποτελείται από επιμέρους κομμάτια που είναι ευκολά να βρεθούν και να αντικατασταθούν σε περίπτωση βλάβης.
- 5) Να υπάρχει ξεχωριστή διάρκεια λειτουργίας του ρομποτικού οχήματος δηλαδή να αποτελείται από δυο τμήματα τροφοδοσίας (κάμερα /κίνηση) που είναι ανεξάρτητα το ένα από το άλλο.
- 6) Να είναι ευκολά τροποποιήσιμο .

### Λειτουργικά Μέρη Κατασκευής

Αρχικά θα παρουσιαστεί το σημαντικότερο λειτουργικό μέρος της κατασκευής που είναι η κεντρική μονάδα επεξεργασίας του ρομποτικού οχήματος. Γίνετε χρήση της έκδοση ESP32-CAM που αναπτύσσεται από την εταιρία Ai-Thinker και χρησιμοποιεί το μοντέλο επεξεργαστή ESP32-S [7]. Το ESP32-CAM εκτός από την επεξεργαστική ισχύς που απαιτείται και την δυνατότητα δημιουργίας επικοινωνίας με το διαδίκτυο, φέρει υποδοχή για σύνδεση κάμερας . Η κάμερα που παρέχεται με την αγορά είναι η OV2640 όπου οι προδιάγραφες της καλύπτουν τις ανάγκες της εργασίας. Συνεπώς καταφέρθηκε η υλοποίηση του ελέγχου, της επικοινωνίας και της οπτικής επαφής της εφαρμογής με μόνο μια αγορά.

Παρακάτω παρουσιάζονται αναλυτικά τα τεχνικά στοιχεία του μικροεπεξεργαστή [8]:

Μοντέλο	ESP32-CAM
Ενθυλάκωση	DIP-16
Μέγεθος	27 * 40,5 * 4,5 (± 0,2) χιλ
SPI Flash	Προεπιλεγμένο στα 32Mbit
Μνήμη RAM	Εσωτερική 520KB + εξωτερική 8MB PSRAM
Bluetooth	Πρότυπα Bluetooth 4.2 BR / EDR και BLE
Wi-Fi	802.11 b / g / n / e / i
Διεπαφή υποστήριξης	UART, SPI, I2C, PWM
Υποστήριξη κάρτας TF	Μέγιστη υποστήριξη 4G
Θύρες Input/Output	9 I/O
Ταχύτητα σειριακής θύρας	Προεπιλεγμένο στα 115200 bps
Μορφή εξόδου εικόνας	JPEG (υποστηρίζεται μόνο από OV2640), BMP, GRAYSCALE
Εύρος φάσματος	2400 ~ 2483,5MHz

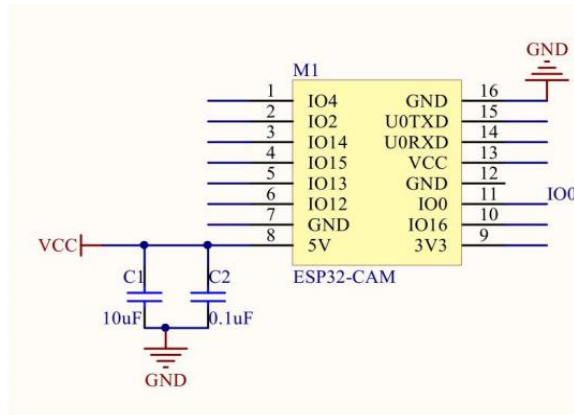
Κεραία	Ενσωματωμένη κεραία PCB, Απολαβή 2dBi
Μετάδοση ισχύος	802.11b: 17 ± 2 dBm (@ 11Mbps) 802.11g: 14 ± 2 dBm (@ 54Mbps) 802.11n: 13 ± 2 dBm (@ MCS7
Λήψη ευαισθησίας	CCK, 1 Mbps: -90dBm CCK, 11 Mbps: -85dBm 6 Mbps (1/2 BPSK): -88dBm 54 Mbps (3/4 64-QAM): -70dBm MCS7 (65 Mbps, 72,2 Mbps): -67dBm
Κατανάλωση ενέργειας	Απενεργοποιημένο φλας: 180mA @ 5V Ενεργοποιημένο φλας και ρυθμισμένη η φωτεινότητα στο μέγιστο: 310mA @ 5V Deep-Sleep κατάσταση: Η χαμηλότερη κατανάλωση ενέργειας μπορεί να φτάσει τα 6mA @ 5V Modern-sleep κατάσταση: : Η χαμηλότερη κατανάλωση ενέργειας μπορεί να φτάσει τα 20mA @ 5V Light-Sleepκατάσταση: : Η χαμηλότερη κατανάλωση ενέργειας μπορεί να φτάσει τα 6,7mA@5V
Ασφάλεια δικτύου	WPA/WPA2/WPA2-Enterprise/WPS
Εύρος τροφοδοσίας	4.75-5.25V
Θερμοκρασία λειτουργίας	-20 °C ~ 70 °C
Θερμοκρασία περιβάλλοντος	-40 °C ~ 125 °C, <90% RH
Βάρος	10 γρ

Πίνακας 1 - Τεχνικά στοιχεία ESP32-CAM

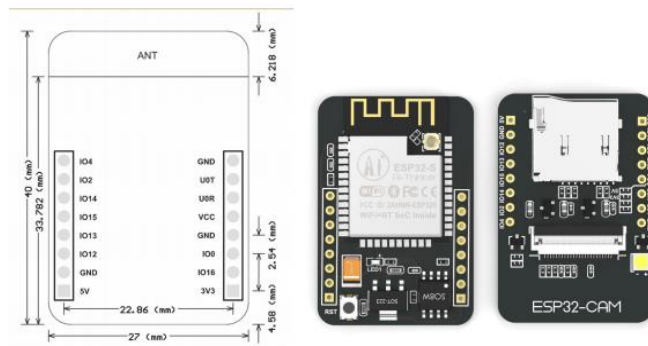
CAM	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1/闪光灯	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		

Εικόνα 2 - Λίστα με διαθέσιμους ακροδέκτες [8]

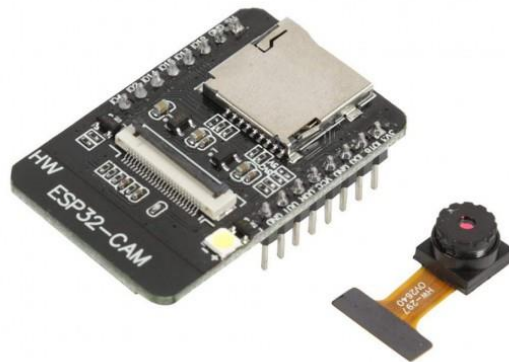




Εικόνα 3 - Ηλεκτρικό διάγραμμα συστήματος ESP32-CAM [8]



Εικόνα 4 - Πρόσωση και πίσω όψη ESP32-CAM [8]



Εικόνα 5 - Πλάγια όψη ESP32-CAM με OV2640

Επόμενο λειτουργικό κομμάτι της κατασκευής αποτελεί το ολοκληρωμένο σύστημα L293D το οποίο αποτελείται από «γέφυρες» τύπου H με σκοπό την εναλλαγή της πολικότητας μιας τάσης που εφαρμόζεται σε ένα φορτίο. Με αυτή την διαδικασία πραγματοποιείται ο έλεγχος διάφορων κινητήρων DC ως προς την κατεύθυνση που κινούνται [9]. Το συγκεκριμένο ολοκληρωμένο κύκλωμα παρασκευάζεται από την εταιρία Texas Instruments και διαθέτει τις εξής απόλυτα μέγιστες προδιαγραφές [10]:

	Ελάχιστο	Μέγιστο	Μονάδα
Τάση τροφοδοσίας VCC1	-	36	V
Τάση τροφοδοσίας VCC2	-	36	V
Τάση εισόδου «λογικής» πύλης	-	7	V
Τάση εξόδου «λογικής» πύλης	-3	VCC2+3	V
Μέγιστο ρεύμα εξόδου, «λογικής» πύλης (μη επαναλαμβανόμενο, $t \leq 100 \mu s$ ):	-1.2	1.2	A
Συνεχές ρεύμα εξόδου «λογικής» πύλης	-600	600	mA
Μέγιστη θερμοκρασία σύνδεσης	-	150	°C
Θερμοκρασία αποθήκευσης	-65	150	°C

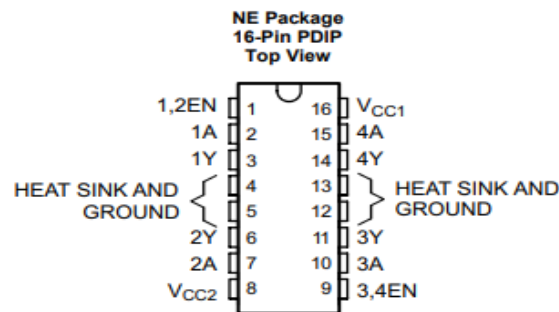
Πίνακας 2 - Απόλυτες μέγιστες προδιαγραφές L293D

Εφόσον η εφαρμογή που θα υλοποιηθεί δεν απαιτεί τις απόλυτα μέγιστες ή υψηλότερες τιμές, και υπάρχει ενδιαφέρον μόνο για την ομαλή λειτουργία του κυκλώματος, θα γίνει αναφορά στις συστημένες συνθήκες λειτουργίας, όπου το κύκλωμα λειτουργεί κανονικά χωρίς επιπτώσεις στην απόδοση του ή την πιθανή δημιουργία μόνιμης βλάβης [10] :

	Ελάχιστο	Μέγιστο	Μονάδα
Τάση τροφοδοσίας VCC1	4.5	7	V
Τάση τροφοδοσίας VCC2	VCC1	36	V
Τάση εισόδου «λογικής» πύλης σε υψηλό επίπεδο για VCC1 $\leq 7 V$	2.3	VCC1	V
Τάση εισόδου «λογικής» πύλης σε υψηλό επίπεδο για VCC1 $\geq 7 V$	2.3	7	V
Τάση εξόδου «λογικής» πύλης σε χαμηλό επίπεδο	-0.3	1.5	V
Θερμοκρασία λειτουργίας σε ανοιχτό περιβάλλον	0	70	°C

Πίνακας 3 - Συστημένες συνθήκες λειτουργίας L293D

Παρακάτω παρουσιάζεται η κάτοψη και η λίστα με τους ακροδέκτες του ολοκληρωμένου [10] :



Εικόνα 6 - Κάτοψη L293D [10]

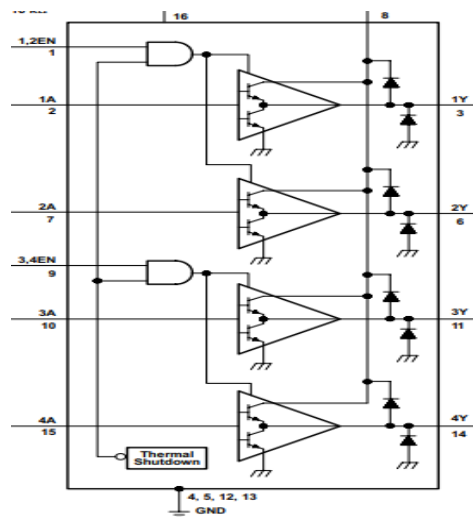
ΟΝΟΜΑ	ΑΡΙΘΜΟΣ ΑΚΡΟΔΕΚΤΗ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
1,2EN	1	Είσοδος	Ενεργοποιεί τα κανάλια 1 και 2
<1:4>A	2, 7, 10, 15	Είσοδος	Είσοδοι οδήγησης
<1:4>Y	3, 6, 11, 14	Έξοδος	Έξοδοι οδήγησης
3,4EN	9	Είσοδος	Ενεργοποιεί τα κανάλια 3 και 4
GROUND	4, 5, 12, 13	-	Αποτελούν γείωση και ψήκτρα του κυκλώματος
VCC1	16	-	Τροφοδοσία για την «λογική» του κυκλώματος
VCC2	8	-	Τροφοδοσία για τις εξόδους οδήγησης

Πίνακας 4 - Κατανομή ακροδεκτών του L293D

INPUTS <sup>(2)</sup>		OUTPUT (Y)
A	EN	
H	H	H
L	H	L
X	L	Z

- (1) H = high level, L = low level, X = irrelevant, Z = high impedance (off)  
(2) In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

Εικόνα 7 - Πίνακας οδήγησης εξόδων [10]



Εικόνα 8 - Ηλεκτρικό σχέδιο L293D [10]

Εφόσον αναλυθήκαν τα λειτουργικά μέρη της κατασκευής που σχετίζονται με την εντολοδότηση και τον έλεγχο, θα γίνει αναφορά στους κινητήρες που θα χρησιμοποιηθούν ώστε να πραγματοποιείτε η κίνηση της εφαρμογής. Οι κινητήρες είναι οι απλοί DC motors που μπορούν να βρεθούν στο κατάστημα GROBOTRONICS [11] ή άλλα παρεμφερή καταστήματα ηλεκτρονικών. Για την κατασκευή απαιτούνται δυο κινητήρες μόνο.

Τα χαρακτηριστικά των κινητήρων είναι [12] :

Συμβατό για χρήση στο breadboard	ΝΑΙ
Τάση τροφοδοσίας (Εύρος)	3 έως 6Volt
Ρεύμα τροφοδοσίας	200 mA
Ταχύτητα	130 RPM
Είδος κιβωτίου	HOBBY
Υλικό γραναζιών	Πλαστικό
Μετάδοση γραναζιών	48:1
Κωδικοποιητής	OXI
Διαστάσεις	70x22x18mm
Βάρος	29g
Μέγεθος άξονα	Ø5.4x8mm

Πίνακας 5 - Χαρακτηριστικά ηλεκτροκινητήρων



Εικόνα 9 - Όψη κινητήρα DC [12]

Όσο αφορά την τροφοδότηση του συστήματος, το κύκλωμα έχει χωριστεί σε δυο διαφορετικές παροχές . Αυτό έγινε σκόπιμα διότι προσφέρει ασφάλεια σε περίπτωση βλάβης αν υπήρχε μόνο μια κοινή παροχή και μας επιτρέπει την αποκλειστική χρήση της κάμερας χωρίς να επιβαρύνεται το κύκλωμα της μετακίνησης της εφαρμογής . Έχοντας υπόψη τα παραπάνω, παραθέτονται οι επιλογές που έγιναν για την τροφοδοσία.

- 1) Κύκλωμα εντολοδότησης και ελέγχου (ESP32-CAM )  
 Το κύκλωμα τροφοδοτείται από δυο επαναφορτιζόμενες μπαταρίες εν σειρά. Μια της τάξης των 1.2 Volt [13] και μια των 3.7 Volt [14] ώστε να παρέχεται 4.9 Volt τροφοδοσία που είναι εντός των προβλεπόμενων ορίων. Οι μπαταρίες έχουν ισάξια χωρητικότητα των 1800 mAh.
- 2) Κύκλωμα οδήγησης κινητήρων DC (L293D)  
 Στο συγκεκριμένο κύκλωμα ή τροφοδότηση γίνεται από τέσσερις επαναφορτιζόμενες μπαταρίες εν σειρά. Όλες έχουν τα κοινά χαρακτηριστικά των 1.2 Volt και 1800 mAh [13].



Εικόνα 10 - Όψη επαναφορτιζόμενης μπαταρίας 1.2V [13]



Εικόνα 11 - Όψη επαναφορτιζόμενης μπαταρίας 3.7V [14]

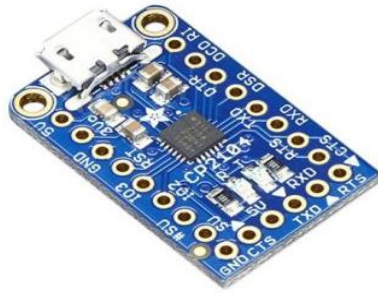
Έχοντας ολοκληρώσει τα λειτουργικά μέρη του απευθύνονται στην κατασκευή του ρομποτικού οχήματος, θα γίνει αναφορά στην τροφοδότηση των ESP32-CAM που λειτουργούν ως σταθερές περιφερικές κάμερες για την αναμετάδοση της εικόνας του χώρου που λειτουργεί το ρομποτικό όχημα. Στην παρούσα περίπτωση απαιτείται ένα απλό ενσύρματο τροφοδοτικό της τάξεως των 5V που μπορεί να βρεθεί από παλιές συσκευές ή κατόπιν αγοράς από ένα κατάστημα ηλεκτρονικών. Εφόσον ολοκληρώθηκε η κατοχή δυο τροφοδοτικών, ξεγυμνώθηκαν οι ακροδέκτες της τροφοδοσίας για να μπορούν να συνδεθούν στα ESP32-CAM.



Εικόνα 12 - Όψη ενσύρματου τροφοδοτικού 5V [15]

Τέλος, λόγω του περιορισμού που διακατέχει ο μικροεπεξεργαστής ESP32-CAM στην μη ύπαρξη διαθέσιμης θύρα USB για τον προγραμματισμό του, θα πρέπει να γίνει χρήση ενός

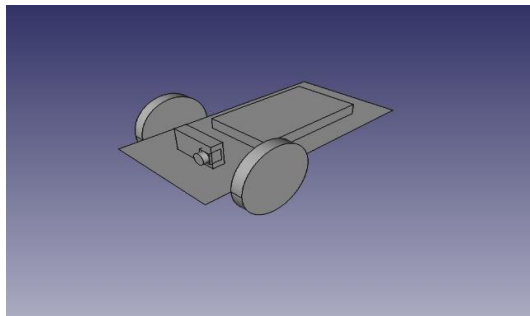
μεταγωγτιστή από USB σε Serial. Για την λειτουργία αυτή χρησιμοποιήθηκε ο Adafruit CP2104 Friend - USB to Serial Converter [16].



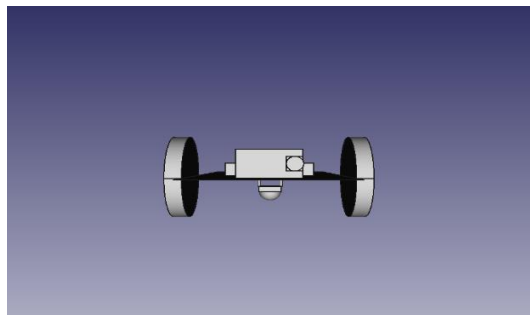
*Εικόνα 13 - Όψη USB σε Serial μετατροπέα [16]*

## Μηχανολογικό σχέδιο

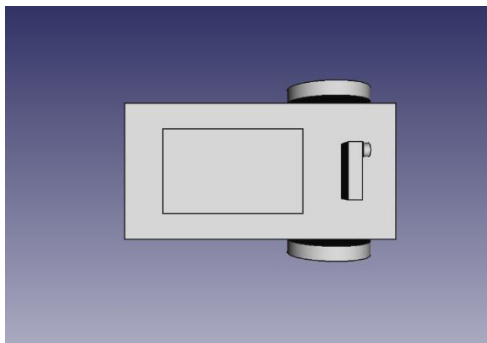
Ο σχεδιασμός για την ανάπτυξη και κατασκευή του ρομποτικού οχήματος έγινε χρησιμοποιώντας το πρόγραμμα FreeCad [17]. Παρακάτω παραθέτονται αρχικά οι φωτογραφίες από το εικονικό μηχανολόγο σχέδιο που χρησιμοποιήθηκε ως βάση για την επίτευξη της εργασίας και μετέπειτα συμπληρώνονται με τις πραγματικές εφόσον είχε ολοκληρωθεί η κατασκευή.



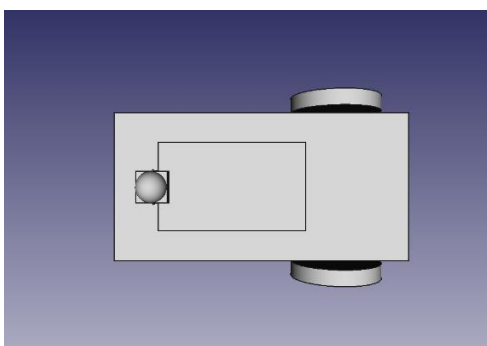
*Εικόνα 14 - Μηχανολογικό σχέδιο(Πρόψη) με FreeCad [17]*



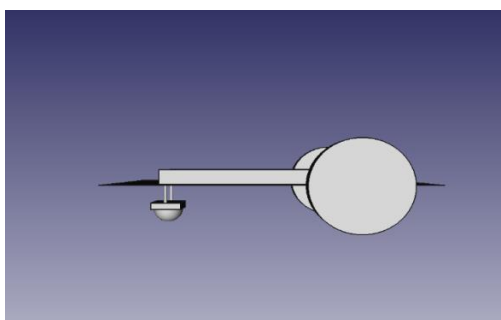
*Εικόνα 15 - Μηχανολογικό σχέδιο(Εμπρόσθια όψη) με FreeCad [17]*



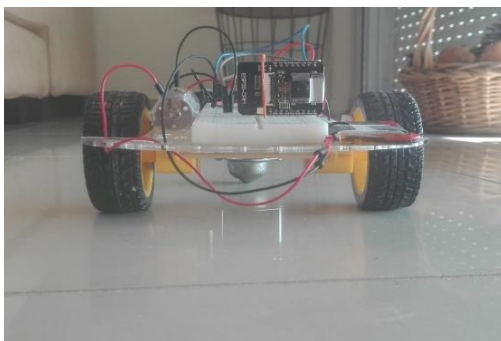
Εικόνα 16 - Μηχανολογικό σχέδιο(Ανω όψη) με FreeCad [17]



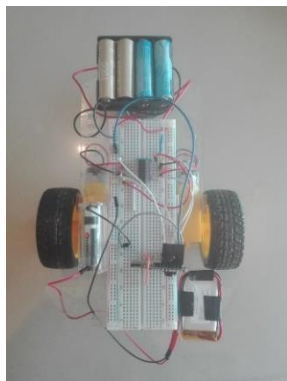
Εικόνα 17 - Μηχανολογικό σχέδιο(Κάτω όψη) με FreeCad [17]



Εικόνα 18 - Μηχανολογικό σχέδιο(Πλάγια όψη) με FreeCad [17]



Εικόνα 19 - Εμπρόσθια όψη ρομποτικού οχήματος



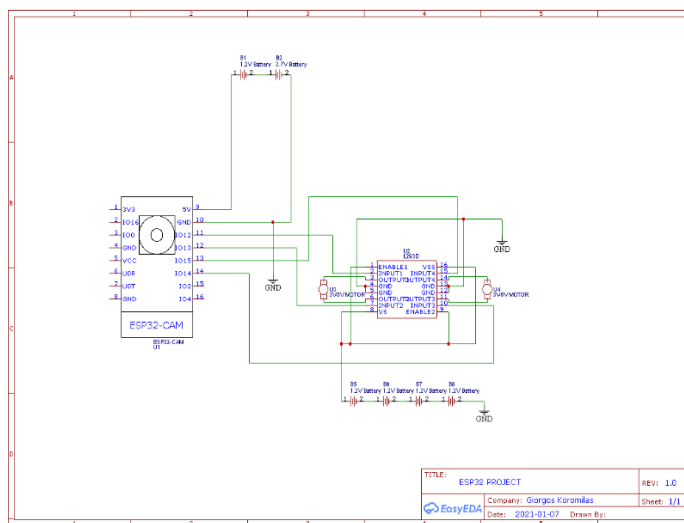
Εικόνα 20 - Κάτω όψη ρομποτικού οχήματος



Εικόνα 21 - Πλάγια όψη ρομποτικού οχήματος

## Ηλεκτρονικό κύκλωμα

Το ηλεκτρονικό κύκλωμα του ρομποτικού οχήματος που θα υλοποιηθεί απεικονίζεται με την χρήση του προγράμματος EasyEDA [18] και αποτελεί βασικό κομμάτι για την σωστή και ομαλή λειτουργία του συστήματος .







# ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΓΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ (SOFTWARE)

Στο κεφάλαιο αυτό θα γίνει αναλυτική παρουσία της δομής του κώδικα συμπεριλαμβάνοντας τα διαγράμματα ροής που απαιτούνται για την διεκπεραίωση της εργασίας.

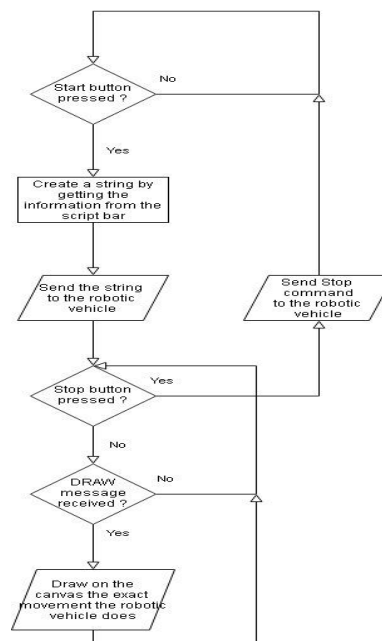
Αρχικά, η εργασία έχει την εξής δομή :

1. Ένας υπολογιστής που θα χαρακτηριστεί ως Server, προβάλει μια ιστοσελίδα στο διαδίκτυο, όπου στην οποία παρέχεται ο έλεγχος, προγραμματισμός και η οπτική επικοινωνία του χρήστη με την απομακρυσμένη κατασκευή.
2. Η ιστοσελίδα εδραιώνει αμφίδρομη επικοινωνία μεταξύ του χρήστη που αποτελεί πλέον client του server με το ρομποτικό όχημα και τις κάμερες που είναι εγκατεστημένες στον χώρο λειτουργίας του, χρησιμοποιώντας κάποια πρωτόκολλα επικοινωνίας.
3. Τέλος, το ρομποτικό όχημα εφόσον του δοθεί εντολή να πραγματοποιήσει, στέλνει στον client την ακριβής κίνηση που πραγματοποιεί για όλη την διάρκεια που εκτελούνται οι εντολές, σε μορφή γραφικής αναπαράστασης της πορείας .

Έχοντας λάβει υπόψη τα παραπάνω, η ανάλυση του λογισμικού θα χωριστεί σε δυο ενότητες, το λογισμικό της ιστοσελίδας και το λογισμικό της κατασκευής.

## Διάγραμμα Ροής Ιστοσελίδας

Το διάγραμμα που ακολουθεί, αναλύει συνοπτικά την δομή του προγράμματος που χρησιμοποιεί η ιστοσελίδα ώστε να επικοινωνεί με το ρομποτικό όχημα και ταυτόχρονα να παρέχει μια γραφική αναπαράσταση στον χρήστη.



Εικόνα 24 - Διάγραμμα ροής Ιστοσελίδας

## Κώδικας Προγράμματος Ιστοσελίδας

Ο κώδικας του προγράμματος που εκτελεί η ιστοσελίδα, έχει συνταχθεί χρησιμοποιώντας γλώσσες προγραμματισμού HTML [19], CSS [20] και JavaScript [21]. Η ιστοσελίδα δομείται από 59 αρχεία εκ των οποίων, τα οχτώ αποτελούν τις λειτουργίες που εκτελούνται σε πηγαίο κώδικα JavaScript, δυο απευθύνονται στα γραφικά χαρακτηριστικά σε μορφή CSS, τρεις είναι οι σελίδες που έχει πρόσβαση ο χρήστης και είναι γραμμένες σε HTML αρχεία και τέλος σαράντα - οχτώ φωτογραφίες και εικονίδια σε μορφή PNG [22].

Επιπρόσθετα, η ιστοσελίδα που θα χρησιμοποιηθεί αποτελεί κομμάτι πνευματικού έργου με ονομασία «Innovative Robot for Educational Robotics and STEM» [23] και κατέχει στοιχεία που έχουν παρθεί από τις ιστοσελίδες W3Schools [24] και GitHub [25], τα οποία δεν αποτελούν προϊόν πνευματικής ιδιοκτησίας του συγγραφέα της εργασίας. Ο κώδικας που παραθέτεται, έχει τροποποιηθεί κατόπιν ανάλυσης και έχει γίνει προσθήκη πηγαίου κώδικα που καλύπτει τις απαιτούμενες ανάγκες της εργασίας.

### Αρχεία HTML

#### 1) Αρχική σελίδα (index.html)

Η αρχική σελίδα αποτελεί τη πρώτη σελίδα που έχει πρόσβαση ο χρήστης, όπου δίνετε η επιλογή για ποιο θα είναι το επιθυμητό επιπέδου ελέγχου του ρομποτικού οχήματος. Ο πηγαίος κώδικας κατέχει κυρίως γραφικά στοιχεία διεύθυνσης.

```
<!DOCTYPE html>      <!-- The declaration that this is an HTML file-->

<html lang="el">      <!-- Website declares the browser/search engine that the page
is declared in Greek-->

<head>

    <title>SmartBot v1</title>      <!-- Website title in the top right of the
screen (Tab)-->

    <meta charset="utf-8">          <!-- Website characters are encoded in UTF-
8-->

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Website is being responsive to the screen of the device-->

    <link rel="stylesheet" type="text/css" href="lib/dragula.min.css">
<!-- Inserting css library to the website -->

    <link rel="stylesheet" type="text/css" href="css/template.css?">
<!-- Inserting css library to the website-->

    <script src="js/socket.js"></script>
<!-- Inserting javascript library to the website-->

</head>

<body>
```

```

<div class="page-index">    <!-- Creating a bar at the top of the page-->
    <div class="top">
        <div class="left">
            </div>
        <div class="center">    <!-- Creating a logo at the center -->
            <p style="text-align: center">SmartBot</p>
        </div>
        <div class="right" style="text-align:right;">
<!-- Creating a button which provides help information-->
            <a id="myBtn" class="tooltip" style="display: inline-
block;">
                <span class="tooltiptext bottom_left">Πάτησε
εδώ για βοήθεια</span>
            </a>
        </div>
    </div>
    <div id="myModal" class="modal">
<!-- Creating the popup help message that is displayed when the button is
pressed-->
        <div class="modal-content">
            <div class="modal-header">
<!-- Contains the headline of the popup message-->
                <span class="close">&times;</span>
                <h2>Βοήθεια & Οδηγίες Χρήσης </h2>
            </div>
            <div class="modal-body">
<!-- Contains the information we want to share with the user-->
                <p>Αυτό είναι το αρχείο με τη βοήθεια. Αυτό
είναι το αρχείο με τη βοήθεια. Αυτό είναι το αρχείο με τη βοήθεια. Αυτό είναι το
αρχείο με τη βοήθεια.<br>Αυτό είναι το αρχείο με τη βοήθεια.</p>
                <p>Αυτό είναι το αρχείο με τη βοήθεια. Αυτό
είναι το αρχείο με τη βοήθεια. Αυτό είναι το αρχείο με τη βοήθεια. Αυτό είναι το
αρχείο με τη βοήθεια.<br>Αυτό είναι το αρχείο με τη βοήθεια.</p>
            </div>
        </div>
    </div>

```

```

        <div class="modal-footer">
<!-- Contains the footer of the popup message-->

            <h6>Γ.Κορομηλάς 2021</h6>

        </div>

    </div>

</div>

<script>

    var modal = document.getElementById("myModal");
    // Get the modal

    var btn = document.getElementById("myBtn");
    // Get the button that opens the modal

    var span = document.getElementsByClassName("close")[0];
    // Get the <span> element that closes the modal

    btn.onclick = function() { // When the user clicks on
the button, open the modal

        modal.style.display = "block"; }

    span.onclick = function() { // When the user clicks on
<span> (x), close the modal

        modal.style.display = "none"; }

    window.onclick = function(event) { // When the user clicks
anywhere outside of the modal, close it

        if (event.target == modal) {

            modal.style.display = "none";

        }

    }

</script>

    <div class="main"> <!-- Creating an area in the middle of the
page that provides the options the user has-->

        <div class="main-intro">

            <p class="top-msg">ΔΙΑΛΕΞΕΙΣ ΕΠΙΠΕΔΟ ΛΕΙΤΟΥΡΓΙΑΣ</p>

            <div class="onethird"> <!-- Redirects the user to
the easy-gr.html-->

                <a href="easy-gr.html" class="level"></a>

```

```

        <p>ΕΛΕΥΘΕΡΟΣ ΕΛΕΓΧΟΣ (ΕΥΚΟΛΟ)</p>
    </div>

    <div class="onethird">    <!-- Redirects the user to
the medium-gr.html-->

        <a href="medium-gr.html" class="level"></a>

        <p>ΠΡΟΓΡΑΜΜΑΤΙΣΜΕΝΟΣ ΕΛΕΓΧΟΣ (ΜΕΤΡΙΟ)</p>

    </div>

    <div class="clr">&nbsp;</div>

</div>

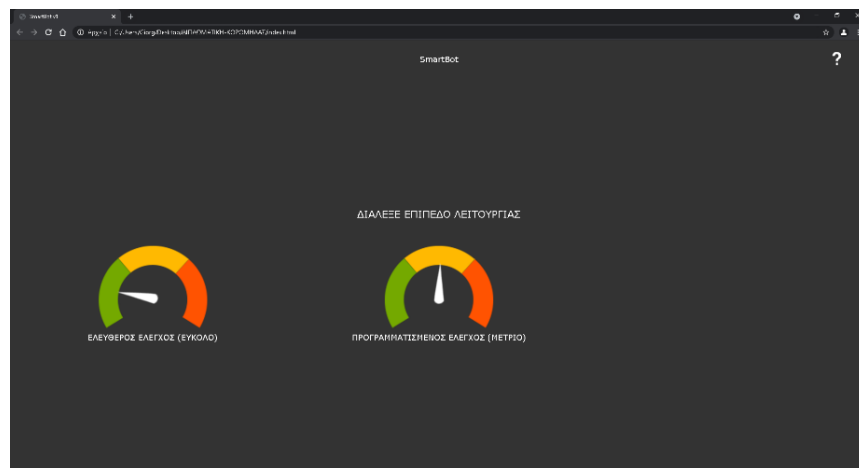
</div>

</div>

</body>

</html>    <!-- The declaration that this HTML file ends here-->

```



*Εικόνα 25 - Σελίδα index.html*

## 2) Σελίδα ελεύθερου ελέγχου (easy-gr.html)

Η σελίδα ελεύθερου ελέγχου απευθύνεται στην ελεύθερή λειτουργία του ρομποτικού οχήματος με το απλό πάτημα των κουμπιών κίνησης και стоп. Ο πηγαίος κώδικας επιτρέπει στον χρήστη να στέλνει εντολές μέσω απλών γραφικών στοιχείων.

```

<!DOCTYPE html>    <!-- The declaration that this is an HTML file-->

<html lang="el">    <!-- Website declares the browser/search engine that the
page is declared in Greek-->

```

```

<head>

    <title>SmartBot v1</title> <!-- Website title in the top right of the
screen (Tab)-->

    <meta charset="utf-8"> <!-- Website characters are encoded in UTF-8-
->

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Website is being responsive to the screen of the device-->

    <link href="css/template.css?" rel="stylesheet" type="text/css">
<!-- Inserting css library to the website -->

    <script src="js/socket.js" type="text/javascript"></script>
<!-- Inserting javascript library to the website-->

</head>

<body>

    <div class="page-easy">
<!-- Creating a bar at the top of the page-->

        <div class="top">

            <div class="left">

                <div class="buttons">
<!-- Creating a button that redirects user back to the index page -->

                    <a class="tooltip" href="index.html">

                        <span class="tooltiptext
bottom_right">Πάτησε εδώ για επιστροφή στην αρχική οθόνη</span>

                    </a>

                </div>

            </div>

            <div class="center"><p style="text-align:
center">SmartBot</p></div> <!-- Creating a logo at the center -->

            <div class="right" style="text-align:right;">
<!-- Creating a button which provides help information-->

                <a id="myBtn" class="tooltip">

                    <span class="tooltiptext bottom_left">Πάτησε
εδώ για βοήθεια</span>

                </a>

            </div>

        </div>

```

```

        </div>

        <div id="myModal" class="modal">
<!-- Creating the popup help message that is displayed when the button is
pressed-->

            <div class="modal-content">

                <div class="modal-header">
<!-- Contains the headline of the popup message-->

                    <span class="close">&times;</span>

                    <h2>Βοήθεια & Οδηγίες Χρήσης </h2>

                </div>

                <div class="modal-body">
<!-- Contains the information we want to share with the user-->

                    <p>Πάτησε τα πλήκτρα για να κινηθεί το ρομπότ.

</p>

                </div>

                <div class="modal-footer">
<!-- Contains the footer of the popup message-->

                    <h6> Γ.Κορομηλάς 2021</h6>

                </div>

            </div>

        </div>

</div>

<script>

    var modal = document.getElementById("myModal");
    // Get the modal

    var btn = document.getElementById("myBtn");
    // Get the button that opens the modal

    var span = document.getElementsByClassName("close")[0];
    // Get the <span> element that closes the modal

    btn.onclick = function() {
    // When the user clicks on the button, open the modal

        modal.style.display = "block"; }

    span.onclick = function() {
    // When the user clicks on <span> (x), close the modal

        modal.style.display = "none"; }

```





```

        </tr>

        <tr>

                <td></td> <!-- Creating an
icon for Left which when clicked sends a message to the robotic vehicle-->

                <td></td> <!-- Creating an
icon for Stop which when clicked sends a message to the robotic vehicle-->

                <td></td> <!-- Creating an
icon for Right which when clicked sends a message to the robotic vehicle-->

        </tr>

        <tr>

                <td>&nbsp;</td>

                <td></td> <!-- Creating
an icon for Backward which when clicked sends a message to the robotic vehicle--
>

                <td>&nbsp;</td>

        </tr>

        <tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td></tr>

        </table>

        </div>

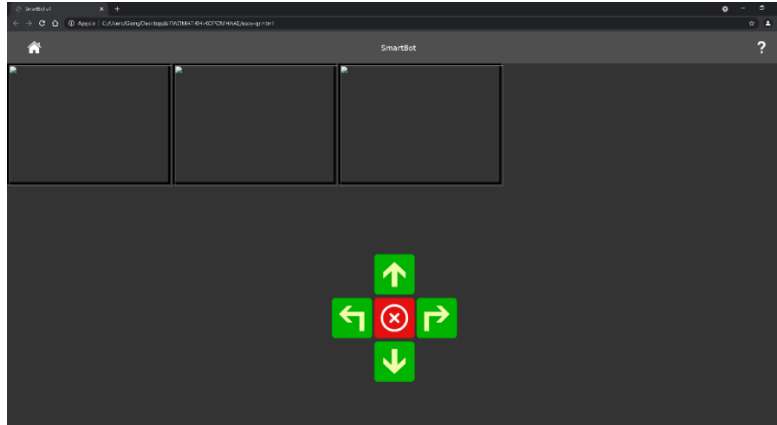
</div>

</div>

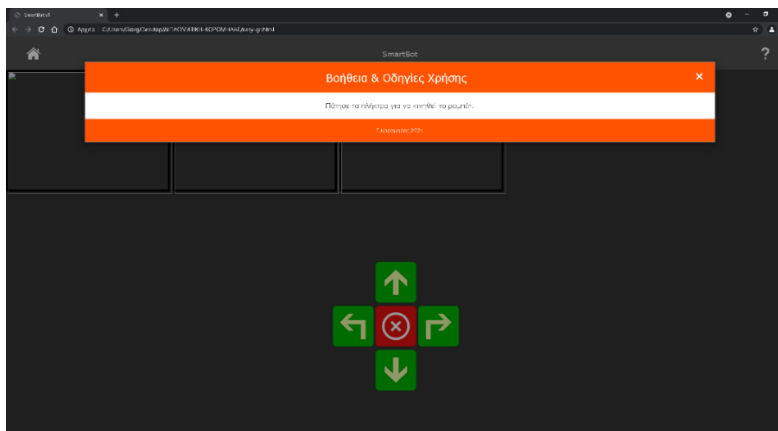
</body>

</html> <!-- The declaration that this HTML file ends here-->

```



Εικόνα 26 - Σελίδα ελεύθερου ελέγχου (easy-gr.html)



Εικόνα 27- Βοήθεια (easy-gr.html)

### 3) Σελίδα προγραμματισμένου ελέγχου (medium-gr.html)

Η σελίδα προγραμματισμένου ελέγχου απευθύνεται στον προγραμματισμό του ρομποτικού οχήματος. Ο πηγαίος κώδικας επιτρέπει στον χρήστη να συντάξει ,αποθηκεύσει και να ανεβάσει κώδικα στο ρομποτικό όχημα χρησιμοποιώντας γραφικά στοιχεία μέσα σε μια γραμμή κατάστασης . Επιπρόσθετα παρέχει στον χρήστη την γραφική αναπαράσταση του προγράμματος σε πραγματικό χρόνο.

```
<!DOCTYPE html>      <!-- The declaration that this is an HTML file-->

<html lang="el">      <!-- Website declares the browser/search engine that the
page is declared in Greek-->

<head>

    <title>SmartBot v1</title>      <!-- Website title in the top right of the
screen (Tab)-->

    <meta charset="utf-8">      <!-- Website characters are encoded in UTF-
8-->
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Website is being responsive to the screen of the device-->

<link rel="stylesheet" type="text/css" href="lib/dragula.min.css">
<!-- Inserting css library to the website -->

<link rel="stylesheet" type="text/css" href="css/template.css?">
<!-- Inserting css library to the website -->

<script src="js/socket.js"></script>
<!-- Inserting javascript library to the website-->

<!-- Custom style for some elements of the page-->

<style>

    .slider {

        -webkit-appearance: none;

        width: 495px;

        height: 25px;

        background: #d3d3d3;

        outline: solid;

        opacity: 0.7;

        -webkit-transition: .2s;

        transition: opacity .2s;

    }

    .slider:hover {

        opacity: 1;

    }

    .slider::-webkit-slider-thumb {

        -webkit-appearance: none;

        appearance: none;

        width: 25px;

        height: 25px;

        background: #4CAF50;

        cursor: pointer;

    }

}
```

```

        .slider::-moz-range-thumb {
            width: 25px;
            height: 25px;
            background: #4CAF50;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="page-medium">
<!-- Creating a bar at the top of the page-->
        <div class="top">
            <div class="left">
                <div class="buttons">
<!-- Creating a button that redirects user back to the index page -->
                    <a class="tooltip" href="index.html">
                        <span class="tooltiptext
bottom_right">Πάτησε εδώ για επιστροφή στην αρχική οθόνη</span>
                    </a>
                </div>
            </div>
            <div class="center">
<!-- Creating an area at the center of the bar-->
                <a class="clear-action tooltip"
href="javascript:void(0)"> <!-- Creating a button that deletes the program the
user has compiled -->
                    
                    <span class="tooltiptext bottom_right">Πάτησε
εδώ για διαγραφή του προγράμματος</span>
                </a>
            </div>
        </div>
    </div>

```

```

        <a class="save-action tooltip"
href="javascript:void(0)"> <!-- Creating a button that enables the user to
save the program in his personal computer (JSON extension) -->

        <span class="tooltiptext bottom_right">Πάτησε
εδώ για να αποθηκεύσεις το πρόγραμμα στη συσκευή σου</span>

        </a>

        <a type="file" class="load-action tooltip"
href="javascript:void(0)"> <!-- Creating a button that enables the user to
upload a program (JSON extension) from his personal computer to the page -->

        <span class="tooltiptext bottom_right">Πάτησε
εδώ για να φορτώσεις ένα πρόγραμμα από τη συσκευή σου</span>

        </a>

    </div>

    <div class="right" style="text-align:right;">
<!-- Creating a button which provides help information-->

        <a id="myBtn" class="tooltip">

        <span class="tooltiptext bottom_left">Πάτησε
εδώ για βοήθεια</span>

        </a>

    </div>

</div>

<div id="myModal" class="modal">
<!-- Creating the popup help message that is displayed when the button is
pressed-->

    <div class="modal-content">

        <div class="modal-header">
<!-- Contains the headline of the popup message-->

            <span class="close">&times;</span>

            <h2>Βοήθεια & Οδηγίες Χρήσης </h2>

        </div>

        <div class="modal-body">
<!-- Contains the information we want to share with the user-->

```

<p>Χρησιμοποιείσαι τα εικονίδια που βρίσκονται στο κάτω μέρος της σελίδας, ώστε να συντάξεις το δικό σου πρόγραμμα οδήγησης του ρομποτικού οχήματος. <br>Οι τιμές που βρίσκονται κάτω από τα εικονίδια αναλογούν σε milliseconds και ορίζουν τον χρόνο εκτέλεσης του εικονιδίου στο οποίο ανήκουν.<br>Εφόσον ολοκληρώσεις το επιθυμητό πρόγραμμα πάτα το εικονίδιο έναρξη ώστε να κλειδώσει η γραμμή εκτέλεσης του προγράμματος.<br>

Σε περίπτωση που θες να λήξεις το πρόγραμμα πρόωρα ή θες να συντάξεις νέο θα πρέπει να ξεκλειδώσεις την γραμμή εκτέλεσης πατώντας το κόκκινο κουμπί.</p>

</div>

<div class="modal-footer">

<!-- Contains the footer of the popup message-->

<h6> Γ.Κορομηλάς 2021</h6>

</div>

</div>

</div>

<script>

```
var modal = document.getElementById("myModal");  
// Get the modal
```

```
var btn = document.getElementById("myBtn");  
// Get the button that opens the modal
```

```
var span = document.getElementsByClassName("close")[0];  
// Get the <span> element that closes the modal
```

```
btn.onclick = function() {  
// When the user clicks on the button, open the modal
```

```
modal.style.display = "block"; }
```

```
span.onclick = function() {  
// When the user clicks on <span> (x), close the modal
```

```
modal.style.display = "none"; }
```

```
window.onclick = function(event) {  
// When the user clicks anywhere outside of the modal, close it
```

```
if (event.target == modal) {
```

```
modal.style.display = "none";
```

```
}
```

```
}
```

</script>

```

        <div style="position: absolute;">                                <!--
Creating an area at the top of the page that provides the user with three camera
feeds-->

                                        <!--
Main camera the robotic vehicle is equipped (Gets camera feed from port-
forwarded address)-->

                                        <!--
Camera in the vicinity of the robotic vehicle (Gets camera feed from port-
forwarded address)-->

                                        <!--
Camera in the vicinity of the robotic vehicle (Gets camera feed from port-
forwarded address)-->

        <div class="canvas-column" id="map" style="display: inline-
block; vertical-align:top; ">

                <div class="canvas-placeholder">                                <!--
Creating a canvas which tracks the movement the robotic vehicle is doing (real
time)-->

                        <canvas id="canv" style="border: 1px solid;"
class="canvas"></canvas>

                </div>

                <input type="range" min="1" max="100" value="1"
class="slider" id="myRange"> <!-- Creating a slider that enables the user to
magnify the canvas-->

                <div>

                        <button style="margin-left: 35%;"
id="recn">ΑΡΧΙΚΟΠΟΙΗΣΗ CANVAS</button> <!-- Creating a button which redirects
the canvas current position to the center -->

                </div>

        </div>

</div>

<script>

        var slider = document.getElementById("myRange");
//Gets desired element based on id

```



```

        var zoom = slider.value;
//Variable which stores the zoom level magnification) of the canvas

        slider.oninput = function () {
// Changes zoom value whenever the bar slider is moved

            zoom = slider.value;

        }

        function showhide() {
// Function which hides or shows the canvas

            var x = document.getElementById("map");
// Gets desired element based on id

            if (x.style.visibility === "hidden") {

                x.style.visibility = "visible";

            } else {

                x.style.visibility = "hidden";

            }

        }

    </script>

    <div class="main">                                <!-- Creating an area at
the middle of the page that provides the place to store the program (script
bar)-->

        <div class="script-column">

            <button
onclick="showhide()">ΕΝΕΡΓΟΠΟΙΗΣΗ/ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ CANVAS</button> <!-- Creating
the button which hides/shows the canvas -->

                <div class="script" id="zz">

                     <!-- Creating the button which unlocks the
script bar and ends the program regardless if it has completed or not -->

                     <!-- Creating the button which locks the
script bar and starts the program -->

                </div>

            </div>

        </div>

    </div>

```

```

        <div class="bottom">                                <!-- Creating an area at the
bottom of the page that provides the usable icons to make a program (drag and
drop to the script bar) -->

            <div class="menu-column">

                <div class="menu">

                    </div>

                </div>

            </div>

        </div>                                <!-- End of page -->

        <script src="lib/dragula.min.js"></script>        <!-- Inserting
javascript library to the website-->

        <script src="js/util.js"></script>              <!-- Inserting
javascript library to the website-->

        <script src="js/file.js"></script>              <!-- Inserting
javascript library to the website-->

        <script src="js/blocks-medium.js"></script>    <!-- Inserting
javascript library to the website-->

        <script src="js/menu-medium.js"></script>     <!-- Inserting
javascript library to the website-->

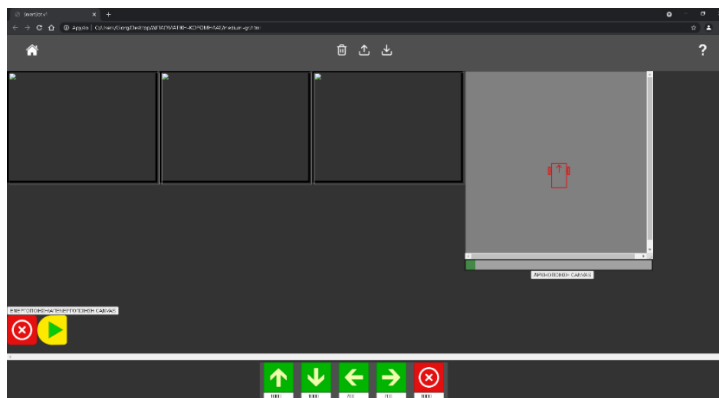
        <script src="js/drag.js"></script>              <!-- Inserting
javascript library to the website-->

        <script src="js/turtle-medium.js"></script>   <!-- Inserting
javascript library to the website-->

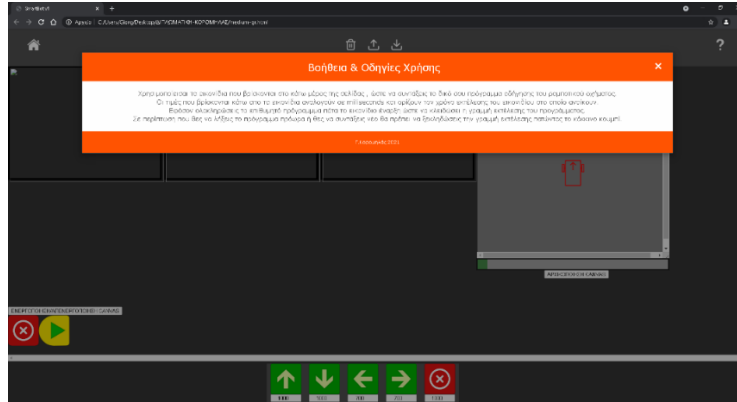
    </body>

</html>                                <!-- The declaration that this HTML file ends here-->

```



Εικόνα 28 - Σελίδα προγραμματισμένου ελέγχου (medium-gr.html)



Εικόνα 29 - Βοήθεια 2 (medium-gr.html)

## Αρχείο CSS

### Γραφικά στοιχεία (template.css)

Ο συγκεκριμένος πηγαίος κώδικας αποτελεί όλες τις ρυθμίσεις που κατέχουν τα γραφικά στοιχεία όπως χρώμα, μέγεθος, τοποθέτηση κτλ.

```
.clearfix:after { content: "."; display: block; clear: both; visibility:
hidden; line-height: 0; height: 0;}

.clearfix { display: inline-block;}

html[xmlns] .clearfix { display: block;}

* html .clearfix { height: 1%;}

/* ----- All Pages ----- */

html, body{ width:100%; height: 100%; padding: 0; border: 0; margin: 0;}

body {background-color:#333; font-family: Verdana, Arial, Helvetica, sans-serif;
}

h1, h2, h3, h4, h5, h6, p {margin:0; padding: 5px; color:#fff; font-weight:
normal; text-align: center; margin: 0em; }

h1 {font-size: 30px;} h2 {font-size: 26px;}

h3 {font-size: 22px;} h6 {font-size: 12px;}

p {font-size: 18px;}

* { box-sizing: border-box;}

/* ----- All Pages Icons ----- */

img.el, img.en {max-width: 40px; height: auto;}

img.el {margin-right:20px;}
```

```

img.help, img.icon { max-height: 30px; width: auto; cursor:pointer; }

img.help { margin-right: 10px; }

img.icon { margin-right: 20px; }

/* ----- All Pages Animations ----- */

img.anim:hover { animation-name: scale-up-center; animation-duration: 1.3s;
animation-timing-function: ease;

animation-delay: 0s; animation-iteration-count: 1; animation-direction:
normal; animation-fill-mode: none;}

/* ----- Intro page -----*/

.page-index { width:100%; height:100%; display:table;}

.page-index .top { margin: 0; padding: 0; display: table-row; vertical-align:
top; height: 80px;}

.page-index .top .left {float:left; width:20%; padding: 25px 0 0 50px;}

.page-index .top .left .flags {padding-left:10px;}

.page-index .top .right {float:left; width:20%; padding: 25px 50px 0 0;}

.page-index .top .center {float:left; width:60%; text-align: center; padding:
25px 0 0 0;}

.page-index .main {display: table; width: 100%; height: 100%;}

.page-index .main-intro {display: table-cell; vertical-align: middle;}

.page-index .main .onethird {float:left; width:33.3333%; text-align: center;}

.page-index p.top-msg {margin: 0 auto 50px auto; text-align: center; font-size:
22px;}

.page-index img.easy, .page-index img.medium, .page-index img.hard {max-width:
250px; margin:0 auto; width: 100%; height: 100%;}

.page-index a.level {padding:15px; text-align:center; margin: 0 auto;}

/* ----- Easy page ----- */

.page-easy { width:100%; height:100%; display:table; background-color: #333;}

.page-easy .top { margin: 0; padding: 0; display: table-row; vertical-align:
top; height: 80px; background-color: #4f4f4f;}

.page-easy .top .left {float:left; width:20%; padding: 25px 0 0 50px;}

.page-easy .top .right {float:left; width:20%; padding: 25px 50px 0 0;}

.page-easy .top .center {float:left; width:60%; text-align: center; padding:
25px 0 0 0;}

```

```

.page-easy .main {display: table; width: 100%; height: 100%; text-align: center;
}

.page-easy .main-intro {display: table-cell; vertical-align: middle;}

.page-easy .main .main-intro table {border: 1px solid #4f4f4f; background-
color:#4f4f4f; padding: 30px 25px 20px 25px;

border-radius: 10px; margin:20px auto 0 auto;}

.page-easy .main .main-intro td {padding: 5px;}

.page-easy .main .main-intro p#speed_value {background-color:#fff; color:#333;
font-size: 20px; text-align: center; padding: 2px 5px; max-width: 100px; border-
radius: 5px; height: 2.5em; line-height: 2.4em; margin: 5px 0 0 0;}

/* ---- Easy page Speed's Slider ---- */

.page-easy input[type=range] { -webkit-appearance: none; outline: none; width:
100%; }

.page-easy input[type=range]::-webkit-slider-thumb { -webkit-appearance: none;
border: 0px; height: 20px; width: 20px; margin-top: -8px;

border-radius: 10px; } /*
slider's knob*/

.page-easy input.enabled[type=range]::-webkit-slider-thumb { background: red;
cursor: pointer; box-shadow: 1px 1px 2px #777, 0px 0px 1px #777; }

.page-easy input[type=range]::-webkit-slider-runnable-track { width: inherit;
height: 5px; background: #fde900; border-radius: 1px; }

.page-easy input.enabled[type=range]::-webkit-slider-runnable-track { cursor:
pointer; }

/* ---- Easy page Blocks ---- */

.page-easy img.block-icon { max-width: 100px; width: auto; cursor:pointer;
border-radius: 10px; transition: 0.3s; display: inline-block;}

.page-easy img.block-icon:hover {opacity: 0.5;}

/* ----- Medium page -----
*/

.page-medium {width:100%; height:100%; display:table; background-color: #333;}

.page-medium .top { margin: 0; padding: 0; display: table-row; vertical-align:
top; height: 10%; background-color: #4f4f4f;}

.page-medium .top .left {float:left; width:20%; padding: 25px 0 0 50px;}

.page-medium .top .right {float:left; width:20%; padding: 25px 50px 0 0;}

.page-medium .top .center {float:left; width:60%; text-align: center; padding:
25px 0 0 0;}

```

```

.page-medium .top p { text-align: center; display: inline-block; margin-
right:100px; font-size:26px; }

.page-medium .top a { display: inline-block; }

.page-medium .top .examples {display:inline-block; padding:0; vertical-align:
top; }

.page-medium .top select.choose-example { height: 37px; border-radius: 7px;
margin: 0; padding: 0 10px; }

.page-medium .main {position: absolute;

    bottom: 80px;width: 100%; height:20%; display: table; vertical-align: middle;
margin: 0; padding:10; }

.page-medium .main .script-column {display: table-cell; vertical-align:
middle;}

.page-medium .main .script{ width: 100%; height: 120px; white-space: nowrap;
overflow-x: scroll; /* border: 1px dotted red; */ }

.page-medium .main .script.over{ background-color: CadetBlue; }

.page-medium .bottom { position: absolute;

    bottom: 0px; margin-left:35%;display: table ; vertical-align: middle;
background-color: #4f4f4f;}

.page-medium .bottom .menu-column {width: 500px; height:100px; white-space:
nowrap; }

.page-medium .bottom .menu{ border-radius: 70px; text-align: center ;height:0%;
}

.page-medium .bottom .menu.over{ background-color: BurlyWood; }

/* ---- Medium page Blocks - Script area Blocks ---- */

.page-medium .main .script .block { /* float:left; */ display: inline-block;
margin: 0 1px 0 0; height: 80px; padding: 0;}

.page-medium .main .script .block img {width: 80px; height:80px; border-
radius:5px;}

.page-medium .main .script img.block-icon { /* float: left; */ margin: 0 1px 0
0; max-width: 80px; width: auto; cursor: pointer;

border-radius: 10px; transition: 0.3s; display: inline-block ;}

.page-medium .main .script img.block-icon.run-action { vertical-align: bottom;
margin: 0; padding: 0; float:left;}

.page-medium .main .script img.block-icon.run-action:hover {opacity: 0.5;}

.page-medium .main .script img.block-icon.stop-action { vertical-align: bottom;
margin: 0; padding: 0; float:left;}

```

```

.page-medium .main .script img.block-icon.stop-action:hover {opacity: 0.5;}

/* ---- Medium page Blocks - Menu area Blocks ---- */

.page-medium .bottom .menu .block { /* float:left; */ display: inline-block;
margin: 0px 10px 0px 10px; height: 10px; padding: 0; }

.page-medium .bottom .menu .block img {width: 80px; height:80px; border-
radius:5px;}

.page-medium .block{ cursor: move; }

.page-medium .block.running{ outline: 5px solid red; }
    /* Do I need this? */

.page-medium .block.next{ outline: 5px solid green; }          /* Do
I need this? */

.page-medium .block.dragging{ opacity: 0.4; }

.page-medium .block input{ width: 5em; margin: -3px 0 0 0; display: block;
width: 80px; text-align: center; padding: 0;}

/* ----- Dethe Code Block. css -----
container */

.container{ margin: 0 -6px 0 20px; padding-bottom: 10px; border: 1px inset
#021429; border-radius: 7px; border-right: 0; background-color:
LightSteelBlue; border-radius: 7px 0 0 7px;}

.over .container{ background-color: CadetBlue; }

/*----- Dethe Code Block. css ----- */

.canvas-column{display: inline-block; }

.canvas-placeholder{ width: 500px; height: 500px; resize: both;

    overflow: auto; }

.canvas{background-color: white; }

/* ----- Animations -----*/
@keyframes scale-up-center {

    0%{ -webkit-transform: scale(1); transform: scale(1);    }

    50%{ -webkit-transform: scale(1.2); transform: scale(1.2); }

    100%{ -webkit-transform: scale(1); transform: scale(1); }

}

/* ----- Tooltips container ----- */

.tooltip { position: relative; display: inline-block;}

```

```

.tooltip .tooltiptext { visibility: hidden; width: 120px; background-color:
black; color: #fff; text-align: center; padding: 8px 12px; border-radius:
6px; font-size: 15px; position: absolute; z-index: 1; }

.tooltiptext.bottom_left {width: 120px; top: 30px; right: 30px; }

.tooltiptext.bottom_right {width: 180px; top: 30px; left: 30px; }

.tooltiptext.forbutton {width: 180px; bottom: 100px; left: -45px; }

/* Show the tooltip text when you mouse over the tooltip container */

.tooltip:hover .tooltiptext { visibility: visible;}

/* ----- Help Modal ----- */

.modal {

    display: none; position: fixed; /* Stay in place */

    z-index: 1; padding-top: 60px; left: 0; top: 0; width: 100%; height: 100%;

    overflow: auto; /* Enable scroll if needed */

    background-color: rgb(0,0,0); /* Fallback color */

    background-color: rgba(0,0,0,0.4); /* Black w/ opacity */

.modal-content {

    position: relative; background-color: #fefefe; margin: auto; padding: 0;

    border: 1px solid #888; width: 80%; box-shadow: 0 4px 8px 0
rgba(0,0,0,0.2),0 6px 20px 0 rgba(0,0,0,0.19);

    -webkit-animation-name: animatetop; -webkit-animation-duration: 0.4s;
animation-name: animatetop; animation-duration: 0.4s}

@-webkit-keyframes animatetop { from {top:-300px; opacity:0} to {top:0;
opacity:1} }

@keyframes animatetop { from {top:-300px; opacity:0} to {top:0; opacity:1} }

.close { color: white; float: right; font-size: 28px; font-weight: bold; }

.close:hover, .close:focus { color: #000; text-decoration: none; cursor:
pointer;}

.modal-header { padding: 16px 26px; background-color: #ff5301; color: white; }

.modal-body {padding: 16px 26px;}

.modal-footer { padding: 16px 26px; background-color: #ff5301; color: white; }

.modal-body p {color:#3e3e3e; line-height: 1.4em; font-size: 16px;}

```



```

/* ----- Media Queries ----- */
@media only screen and (max-width:400px) {
  /* For mobile phones: */
  .left, .center, .right{ width:100%; }
}

```

## Αρχεία Javascript

### 1) Δημιουργία εικονιδίων (blocks-medium.js)

Πηγαίος κώδικας που αποτελείται από πολλές συναρτήσεις που αφορούν την δημιουργία και την απόδοση πληροφοριών σε γραφικά στοιχεία που χειρίζεται ο χρήστης.

```

(function (global) {          // Create a global function

  "use strict";

  function createBlock(name, value, contents) {          // Function which
creates the blocks in the bottom area of the medium-gr page

    var classname ="block " + name;          //Variable that stores the name

    var item = elem(          // Variable that stores the element
<div> with all the custom settings below

      "div",

      { class: classname, draggable: true, "data-name": name },

    );

    if (value !== undefined && value !== null) {          //To proceed the value
given in the function has to be something

      var imgfile = "icons/" + name + ".png";          //Variable that stores
the path which corresponds to an image file in icons folder

      item.appendChild(elem('img',{ 'classname': name, src : imgfile }));
//Gives the item variable a class and image setting

      item.appendChild(elem("input", { type: "number", value: value }));
//Gives the item variable an input (numeric) setting

    }

    if (Array.isArray(contents)) {

      item.appendChild(

        elem(

          "div",

```

```

        { class: "container" },
        contents.map(function (block) {
            return createBlock.apply(null, block);
        })
    )
);
} else if (typeof contents === "string") {
    item.appendChild(document.createTextNode(" " + contents));
}
return item;
}

function blockContents(block) {
    var container = block.querySelector(".container");
    return container ? Array.from(container.children) : null;
}

function blockValue(block) {
    var input = block.querySelector("input");
    return input ? Number(input.value) : null;
}

function blockUnits(block) {
    if (
        block.children.length > 1 &&
        block.lastChild.nodeType === Node.TEXT_NODE &&
        block.lastChild.textContent
    ) {
        return block.lastChild.textContent.slice(1);
    }
}

function blockScript(block) {
    var script = [block.dataset.name];

```

```

var value = blockValue(block);

if (value !== null) {
    script.push(blockValue(block));
}

var contents = blockContents(block);

var units = blockUnits(block);

if (contents) {
    script.push(contents.map(blockScript));
}

if (units) {
    script.push(units);
}

return script.filter(function (notNull) {
    return notNull !== null;
});
}

function runBlocks(blocks) { // Function that gives the "run" status
on the blocks

    blocks.forEach(function (block) {
        trigger("run", block);
    });
}

global.Block = { //Give the functions a global name to call
(call these functions in different javascript files)

    create: createBlock,

    value: blockValue,

    contents: blockContents,

    script: blockScript,

    run: runBlocks,
};

```

```

    window.addEventListener("unload", file.saveLocal, false);    //Event linked
function

    window.addEventListener("load", file.restoreLocal, false);  //Event linked
function

}) (window);

```

## 2) Λειτουργία γραμμής προγραμματισμού (menu-medium.js)

Πηγαίος κώδικας που αποτελείται από πολλές συναρτήσεις που αφορούν την λειτουργικότητα της γραμμής κατάστασης, όπου ο χρήστης μπορεί να συντάξει κώδικα σε μορφή εικονιδίων και να προγραμματίσει το ρομποτικό όχημα.

```

(function (global) {          // Create a global function

    "use strict";

    var menu = document.querySelector(".menu");

    var script = document.querySelector(".script");

    var scriptRegistry = {};

    var scriptDirty = false;

    function runSoon() {

        scriptDirty = true;

    }

    function menuItem(name, fn, value, units) {

        var item = Block.create(name, value, units);

        scriptRegistry[name] = fn;

        menu.appendChild(item);

        return item;

    }

    function run() {

        if (scriptDirty) {

            scriptDirty = false;

            trigger("beforeRun", script);

            var blocks = Array.from(document.querySelectorAll(".script > .block"));

            Block.run(blocks);

        }

    }

} (global));

```

```

        trigger("afterRun", script);
    } else {
        trigger("everyFrame", script);
    }
    requestAnimationFrame(run);
}
requestAnimationFrame(run);
function runEach(evt) {
    var elem = evt.target;
    if (!elem.matches(".script .block")) return;
    if (elem.dataset.name === "Define block") return;
    elem.classList.add("running");
    scriptRegistry[elem.dataset.name](elem);
    elem.classList.remove("running");
}
function repeat(block) {
    var count = Block.value(block);
    var children = Block.contents(block);
    for (var i = 0; i < count; i++) {
        Block.run(children);
    }
}
global.Menu = { //Give the functions a global name to
call (call these functions in different javascript files)
    runSoon: runSoon,
    item: menuItem,
};
document.addEventListener("scriptChanged",
function(){if(window.WorKing==false){ //Event driven function which
draws on the canvas the robotic vehicles destination based if the program is
running or not
    runSoon();
}
}

```

```

    } }, false);

    document.getElementById('myRange')

    .addEventListener("input", function(){if(window.WorKing==false){
//Event driven function which draws on the canvas the robotic vehicles
destination based if the canvas magnification has changed

    runSoon();

} }, false);

document

//Event driven function which unlocks the script bar, the start program button
and the upload / delete program buttons

.querySelector('.stop-action')

.addEventListener('click', function(){

    runSoon();

    for (var i = 0; i < window.RUNBLK.length; i++) {

        window.RUNBLK[i].style.pointerEvents="auto";

    }

    var x=document.getElementsByClassName("run-action");

    x[0].style.pointerEvents="auto";

    x=document.getElementsByClassName("clear-action");

    x[0].style.pointerEvents="auto";

    x=document.getElementsByClassName("load-action");

    x[0].style.pointerEvents="auto";

}, false);

script.addEventListener("run", runEach, false);
//Event driven function when the "run" status is triggered

script.addEventListener("change",function(){if(window.WorKing==false){
//Event driven function which draws on the canvas the robotic vehicles
destination based if the program has changed

    runSoon();

} }, false);

script.addEventListener("keyup",function(){if(window.WorKing==false){
//Event driven function which draws on the canvas the robotic vehicles
destination based if the "keyup" status has triggered

```

```

runSoon();

} }, false);

})(window);

```

### 3) Σχεδίαση εικονιδίων και canvas (turtle-medium.js)

Πηγαίος κώδικας που αποτελείται από πολλές συναρτήσεις που αφορούν την σχεδίαση γραφικών στοιχείων, συμπεριλαμβάνοντας και τις λειτουργίες που απαιτούνται για την γραφική αναπαράσταση του ρομποτικού οχήματος σε πραγματικό χρόνο.

```

(function (global) {
    // Create a global function

    'use strict';

    window.CanvInd = { x: 0, y: 0 }; // Create a global variable that
    stores the canvas pane position

    window.RUNBLK; // Create a global variable that
    stores all the blocks in the script bar (Array type)

    var PIXEL_RATIO = window.devicePixelRatio || 1; //Variable that
    stores pixel ratio

    var canvasPlaceholder = document.querySelector('.canvas-placeholder');
    //Variable that gets all the elements

    var canvas = document.querySelector('.canvas');

    var script = document.querySelector('.script');

    var ctx = canvas.getContext('2d'); //Variable that stores the
    canvas drawing context

    var cos = Math.cos, sin = Math.sin, sqrt = Math.sqrt, PI = Math.PI;
    //Variables that have mathematical uses

    var DEGREE = PI / 180;

    var WIDTH, HEIGHT, position, direction, visible, pen, color;

    function onResize(evt) { //Function for the canvas dimensions

        WIDTH = canvasPlaceholder.getBoundingClientRect().width *
PIXEL_RATIO;

        HEIGHT = canvasPlaceholder.getBoundingClientRect().height *
PIXEL_RATIO;

        canvas.setAttribute('width', WIDTH);

        canvas.setAttribute('height', HEIGHT);

        canvas.style.top = canvasPlaceholder.getBoundingClientRect().top +
"px";
    }
}

```

```

+ "px";
    canvas.style.left = canvasPlaceholder.getBoundingClientRect().left

    canvas.style.width = (WIDTH / PIXEL_RATIO) + "px"
    canvas.style.height = (HEIGHT / PIXEL_RATIO) + "px"

    if (evt) {
        if (window.WorKing == true) {
            CurrentPos();
        }
        else {
            Menu.runSoon();
        }
    }
}

function reset() { //Function which resets the canvas pen
    recenter();
    direction = deg2rad(90); // facing "up"
    visible = true;
    pen = true; // when pen is true we draw, otherwise we
move without drawing
    color = 'black';
}

function deg2rad(degrees) { return DEGREE * degrees; }

function drawTurtle() { //Function which draws the robotic vehicle
on the canvas
    var userPen = pen; // save pen state
    if (visible) {
        penUp(); _moveForward(5, zoom, 'red'); penDown();
        _turn(-90); _moveForward(20, zoom, 'red');
        _turn(-90); _moveForward(20, zoom, 'red');
        _turn(90); _moveForward(3, zoom, 'red');
    }
}

```



```

    _turn(90); _moveForward(10, zoom, 'red');
    _turn(-90); _moveForward(5, zoom, 'red');
    _turn(-90); _moveForward(20, zoom, 'red');
    _turn(-90); _moveForward(5, zoom, 'red');
    _turn(-90); _moveForward(10, zoom, 'red');
    penUp(); _turn(90); _moveForward(3, zoom, 'red');
    penDown(); _turn(90); _moveForward(45, zoom, 'red');
    _turn(-90); _moveForward(40, zoom, 'red');
    _turn(-90); _moveForward(45, zoom, 'red');
    _turn(90); _moveForward(3, zoom, 'red');
    _turn(90); _moveForward(10, zoom, 'red');
    _turn(-90); _moveForward(5, zoom, 'red');
    _turn(-90); _moveForward(20, zoom, 'red');
    _turn(-90); _moveForward(5, zoom, 'red');
    _turn(-90); _moveForward(10, zoom, 'red');
    penUp(); _turn(90); _moveForward(3, zoom, 'red');
    penDown(); _turn(90); _moveForward(20, zoom, 'red');
    _turn(-90); _moveForward(20, zoom, 'red');
    penUp(); _turn(-90); _moveForward(5, zoom, 'red');
    penDown(); _turn(-45); _moveForward(10, zoom, 'red');
    penUp(); _turn(-180); _moveForward(10, zoom, 'red');
    penDown(); _turn(-90); _moveForward(10, zoom, 'red');
    penUp(); _turn(-180); _moveForward(10, zoom, 'red');
    penDown(); _turn(135); _moveForward(20, zoom, 'red');
    if (userPen) {
        penDown(); // restore pen state
    }
}
}

```

```

function drawCircle(radius) {           //Function which draws circles

    var userPen = pen; // save pen statec

    if (visible) {

        penUp(); _moveForward(-radius, zoom, 'blue'); penDown();

        _turn(-90);

        var steps = Math.min(Math.max(6, Math.floor(radius / 2)),
360);

        var theta = 360 / steps;

        var side = radius * 2 * Math.sin(Math.PI / steps);

        _moveForward(side / 2, zoom, 'blue');

        for (var i = 1; i < steps; i++) {

            _turn(theta); _moveForward(side, zoom, 'blue');

        }

        _turn(theta); _moveForward(side / 2, zoom, 'blue');

        _turn(90);

        penUp(); _moveForward(radius, zoom, 'blue'); penDown();

        if (userPen) {

            penDown(); // restore pen state

        }

    }

}

function drawSquare() {           //Function which draws squares

    var userPen = pen; // save pen statec

    if (visible) {

        penUp(); _moveForward(20, zoom, 'gold' ); penDown();

        _turn(90); _moveForward(45, zoom, 'gold' ); _turn(90);

        _moveForward(90, zoom, 'gold' ); _turn(90);

        _moveForward(90, zoom, 'gold' ); _turn(90);

        _moveForward(90, zoom, 'gold' ); _turn(90);

        _moveForward(45, zoom, 'gold' ); penUp(); _turn(90);

```

```

        _moveForward(20, zoom, 'gold'); _turn(180); penDown();

        if (userPen) {

            penDown(); // restore pen state

        }

    }

    function StartPoint() { //Function which draws the starting area in
circle

        reset();

        drawCircle(100);

    }

    function EndPoint() { //Function which draws the ending area in
circle

        drawCircle(100);

    }

    global.Turtle = { //Making the functions global

        CurrentPos: CurrentPos,

    };

    function CurrentPos() { //Function which draws the current position
of the robotic vehicle

        clear();

        Block.run(window.RUNBLK);

        EndPoint();

        StartPoint();

        for (var i = 0; i < (window.POSindex + 1); i++) {

            if (window.POS1[i] == "Forw") {

                _moveForward(window.POS2[i], zoom, 'black');

            }

            else if (window.POS1[i] == "Back") {

                _moveForward(-window.POS2[i], zoom, 'black');

            }

        }

    }

```

```

        else if (window.POS1[i] == "Left") { // 128.6 are the
degrees of a one second turn in either direction

            _turn((window.POS2[i] / 1000) * 128.6);

        }

        else if (window.POS1[i] == "Righ") {

            _turn((-window.POS2[i] / 1000) * 128.6);

        }

    }

    drawTurtle();

}

function _moveForward(distance, zoom, color) { //Function which moves
the canvas pen to draw lines

    var start = position;

    position = {

        x: cos(direction) * (distance / zoom) * PIXEL_RATIO +
start.x,

        y: -sin(direction) * (distance / zoom) * PIXEL_RATIO +
start.y

    };

    if (distance != 0) {

        if (pen) {

            if (!color) {

                ctx.strokeStyle = 'green';

            }

            else {

                ctx.strokeStyle = color;

            }

            ctx.beginPath();

            ctx.moveTo(start.x, start.y);

            ctx.lineTo(position.x, position.y);

            ctx.lineWidth = 2;

            ctx.stroke();

        }

    }

}

```

```

        }
    }

    return position;
}

function penUp() { pen = false; } //Various functions
function penDown() { pen = true; }
function hideTurtle() { visible = false; }
function showTurtle() { visible = true; }
function forward(block) {
    var x = _moveForward(Block.value(block), zoom,);
    return x;
}

function back(block) { _moveForward(-Block.value(block), zoom,); }
function circle(block) { drawCircle(Block.value(block)); }
function _turn(degrees) { direction += deg2rad(degrees); }
function left(block) { _turn((Block.value(block) / 1000) * 128.6); }
function right(block) { _turn((-Block.value(block) / 1000) * 128.6); }
function recenter() {
    position = { x: WIDTH / 2, y: HEIGHT / 2 };
}

function stop() { drawSquare(); }

function clear() { //Function which clears the canvas
    ctx.save();
    ctx.fillStyle = 'gray';
    if ((window.CanvInd.x >= 0) && (window.CanvInd.y >= 0)) {
        ctx.fillRect(-window.CanvInd.x, - window.CanvInd.y, 500 +
window.CanvInd.x, 500 + window.CanvInd.y);
    } else if ((window.CanvInd.x < 0) && (window.CanvInd.y >= 0)) {
        ctx.fillRect(-window.CanvInd.x, - window.CanvInd.y, 500 -
window.CanvInd.x, 500 + window.CanvInd.y);
    } else if ((window.CanvInd.x >= 0) && (window.CanvInd.y < 0)) {

```

```

        ctx.fillRect(-window.CanvInd.x, -window.CanvInd.y, 500 +
window.CanvInd.x, 500 - window.CanvInd.y);

        } else if ((window.CanvInd.x < 0) && (window.CanvInd.y < 0)) {
            ctx.fillRect(-window.CanvInd.x, -window.CanvInd.y, 500 -
window.CanvInd.x, 500 - window.CanvInd.y);

        }

        ctx.restore();

        reset();

        ctx.moveTo(position.x, position.y);

    }

    onResize();

    Menu.item('Forward', forward, 1000);    //Creates the Forward block with
initial value of 1000 ms

    Menu.item('Backward', back, 1000);    //Creates the Backward block with
initial value of 1000 ms

    Menu.item('Left', left, 700);    //Creates the Left block with
initial value of 700 ms

    Menu.item('Right', right, 700);    //Creates the Right block with
initial value of 700 ms

    Menu.item('Stop', stop,1000);    //Creates the Stop block with
initial value of 1000 ms

    function getCursorPosition(canv, event) {    // Function which tracks
the cursor positon on the canvas so it can move the viewing angle

        const rect = canv.getBoundingClientRect();

        const x3 = event.clientX - rect.left;

        const y3 = event.clientY - rect.top;

        console.log("x: " + x3 + " y: " + y3);

        position = { x: x3, y: y3 };

        if (x3 < 250) {

            if (y3 < 250) {

                ctx.translate(75, 75);

                window.CanvInd.x = window.CanvInd.x + 75;

                window.CanvInd.y = window.CanvInd.y + 75;

                console.log(window.CanvInd.x);

```

```

        console.log(window.CanvInd.y);
    }
    else if (y3 < 500) {
        ctx.translate(75, -75);
        window.CanvInd.x = window.CanvInd.x + 75;
        window.CanvInd.y = window.CanvInd.y - 75;
        console.log(window.CanvInd.x);
        console.log(window.CanvInd.y);
    }
}
else if (x3 < 500) {
    if (y3 < 250) {
        ctx.translate(-75, 75);
        window.CanvInd.x = window.CanvInd.x - 75;
        window.CanvInd.y = window.CanvInd.y + 75;
        console.log(window.CanvInd.x);
        console.log(window.CanvInd.y);
    }
    else if (y3 < 500) {
        ctx.translate(-75, -75);
        window.CanvInd.x = window.CanvInd.x - 75;
        window.CanvInd.y = window.CanvInd.y - 75;
        console.log(window.CanvInd.x);
        console.log(window.CanvInd.y);
    }
}
clear();
if (window.WorKing == true) {
    CurrentPos();
}

```

```

        else {
            Menu.runSoon();
        }
    }

    const canV = document.querySelector('canvas')

    canV.addEventListener('mousedown', function (e) { //Event driven function
for the cursor position

        getCursorPosition(canV, e)

    })

    script.addEventListener('beforeRun', clear, false); // always clear
canvas first

    script.addEventListener('afterRun', drawTurtle, false); // show turtle if
visible

    document

        .querySelector('.run-action')

        .addEventListener('click', function () { //Function which locks
the script bar, the upload and clear button

            window.RUNBLK =
Array.from(document.querySelectorAll(".block"));

            CurrentPos();

            for (var i = 0; i < window.RUNBLK.length; i++) {

                window.RUNBLK[i].style.pointerEvents = "none";

            }

            var x = document.getElementsByClassName("run-action");

            x[0].style.pointerEvents = "none";

            x = document.getElementsByClassName("clear-action");

            x[0].style.pointerEvents = "none";

            x = document.getElementsByClassName("load-action");

            x[0].style.pointerEvents = "none";

        }, false);

    document

```



```

        .getElementById('myRange')

        .addEventListener("input", function () { //Function which draws
the current position of the robotic vehicle on canvas when the magnification
changes

            if (window.WorKing == true) {

                CurrentPos();

            }

        }, false);

document

        .getElementById('recn')

        .addEventListener("click", function () { //Function which
recenters the viewing angle of the canvas

            window.CanvInd = { x: 0, y: 0 };

            onResize("x");

        }, false);

        window.addEventListener('resize', onResize, false); //Event driven
function when the window is resized

    })(window);

```

#### 4) Σύνδεση και επικοινωνία (socket.js)

Πηγαίος κώδικας που αφορά την δημιουργία σύνδεσης μεταξύ ιστοσελίδας και ρομποτικού οχήματος. Επιπλέον κατέχει λειτουργίες αποθήκευσης των εισερχομένων δεδομένων.

```

window.POS1 = {}; //Global variable that stores the DRAW commands
( Forward,Backward,Left,Right,Stop)

window.POS2 = {}; //Global variable that stores the DRAW command
values (numeric up to 9 digits)

window.POSindex = 0; //Global variable that stores the index of the
above arrays

var connection = new WebSocket('ws://2.84.204.83:2210/ws', []); //Creating a
connection to the robotic vehicle using websocket protocol

connection.onopen = function () { //Message to appear when connection
is established

        console.log('Connect ' + new Date());

};

```

```

connection.onerror = function (error) {          //Message to appear when error is
found
    console.log('WebSocket Error ', error);
};

connection.onclose = function () {              //Message to appear when
connection is closed
    console.log(event.code);
    console.log('WebSocket connection closed');
};

connection.onmessage = function (e) {          //Function which fills the
global variables with the information the robotic vehicle sends
    console.log('Server: ', e.data);
    if (window.WorKing == true) {
        if (e.data.includes("DRAW") == true) {
            var command = e.data.slice(4, e.data.length);
            if (command.includes(window.POS1[window.POSindex])) {
                window.POS2[window.POSindex] =
(parseInt(window.POS2[window.POSindex]) + parseInt(command.slice(4,
command.length))).toString();
            }
            else {
                if (window.POS1[window.POSindex] == null) {
                    window.POS1[window.POSindex] =
command.slice(0, 4);
                    window.POS2[window.POSindex] =
command.slice(4, command.length);
                }
                else {
                    window.POSindex = window.POSindex + 1;
                    window.POS1[window.POSindex] =
command.slice(0, 4);
                    window.POS2[window.POSindex] =
command.slice(4, command.length);
                }
            }
        }
    }
};

```



```

}

function jsonToScript(json) {           //Function which translates the JSON
string to script bar program

    clearScript();

    JSON.parse(json).forEach(function (block) {

        scriptElem.appendChild(Block.create.apply(null, block));

    });

    Menu.runSoon();

}

function restoreLocal() {           //Function which restores the program in
local memory

    jsonToScript(localStorage[title] || "[]");

}

function clearScript() {           //Function deletes the current program in the
script bar

    Array.from(document.querySelectorAll(".script > .block")).forEach(function (

        block

    ) {

        block.parentElement.removeChild(block);

    });

    Menu.runSoon();

}

function saveFile(evt) {           //Function which lets the user download the
current program to their computer

    var title = prompt("Save file as: ");

    if (!title) {

        return;

    }

    var file = new Blob([scriptToJson()], { type: "application/json" });

    var reader = new FileReader();

    var a = document.createElement("a");

    reader.onloadend= function () {

```

```

    var a = elem("a", { href: reader.result, download: title + ".json" });
    a.click();
};

reader.readAsDataURL(file);
console.log(reader.result);
    console.log(reader.error);
}

function readFile(file) {    //Function which reads an uploaded file
    var fileName = file.name;
    if (fileName.indexOf(".json", fileName.length - 5) === -1) {
        return alert("Not a JSON file");
    }
    var reader = new FileReader();
    reader.readAsText(file);
    reader.onload = function (evt) {
        jsonToScript(evt.target.result);
    };
}

function loadFile() {    //Function which lets the user upload a JSON file
as a program
    var input = elem("input", { type: "file", accept: "application/json" });
    if (!input) {
        return;
    }
    input.addEventListener("change", function (evt) {
        readFile(input.files[0]);
    });
    input.click();
}

```

```

function loadExample(evt) { //Function which loads examples ( Disabled)

    var exampleName = evt.target.value;

    if (exampleName === "") {

        return;

    }

    clearScript();

    file.examples[exampleName].forEach(function (block) {

        scriptElem.appendChild(Block.create.apply(null, block));

    });

    Menu.runSoon();

}

function runRobot(){ // Function which sends the program data as
one string to the robotic vehicle

    var nam = zz.querySelectorAll('input[type="number"]');

    var ono = zz.querySelectorAll('.block');

    var times = [], onomata = [];

    nam.forEach((item) => {

        times.push(item.value);

    });

    ono.forEach((item) => {

        onomata.push(item.dataset.name);

    });

    var mun = "program=";

    for (var i = 0; i < onomata.length; i++) {

        mun = mun + String(onomata[i]) + String(times[i].length) +
String(times[i]);

    }

    mun = mun + "e";

        var datatosend = mun;

        console.log("**** datatosend = " + datatosend);

```

```

        connection.send(datatosend);

    window.WorKing=true;

}

function stopRobot(){ // Function which sends message to the robotic
vehicle (either the program has ended or emergency stop )

    console.log("**** datatosend = program=Stop10e");

    connection.send("program=Stop10e");

    window.POS1={};

    window.POS2={};

    window.POSindex=0;

    window.WorKing=false;

}

global.file = { //Making global functions

    saveLocal: saveLocal,

    restoreLocal: restoreLocal,

    runRobot: runRobot,

    stopRobot:stopRobot,

};

document //Event driven function for Clear button

    .querySelector(".clear-action")

    .addEventListener("click", clearScript, false);

document //Event driven function for Save button

    .querySelector(".save-action")

    .addEventListener("click", saveFile, false);

document //Event driven function for Load button

    .querySelector(".load-action")

    .addEventListener("click", loadFile, false);

document

    .querySelector('.run-action') //Event driven function for Start button

    .addEventListener('click', runRobot, false);

```

```

document

.querySelector('.stop-action') //Event driven function for Stop button

.addEventListener('click', stopRobot, false);

})(window);

```

## 6) Λειτουργικό αρχείο (util.js)

Πηγαίος κώδικας που αποτελείται από διάφορες γενικές λειτουργίες.

```

(function(global){

  'use strict';

  global.elem = function elem(name, attrs, children){

    attrs = attrs || {};

    children = children || [];

    var e = document.createElement(name);

    Object.keys(attrs).forEach(function(key){

      e.setAttribute(key, attrs[key]);

    });

    children.forEach(function(child){

      if (typeof child === 'string'){

        child = document.createTextNode(child);

      }

      e.appendChild(child);

    });

    return e;

  };

  global.trigger = function trigger(name, target){

    target.dispatchEvent(new CustomEvent(name, {bubbles: true, cancelable:

false}));

  };

})(window);

```

## 7) Ρυθμίσεις Drag and Drop (drag.js)

Πηγαίος κώδικας που αποτελείται από τις ρυθμίσεις της βιβλιοθήκης dragula που έχουμε διαλέξει.



```

let drake = dragula({
  isContainer: function (el) {
    return el.matches(".menu, .script, .script .container");
  },
  moves: function (el, source, handle, sibling) {
    console.log("moves: %s", el.matches(".block"));
    return el.matches(".block");
  },
  accepts: function (el, target, source, sibling) {
    return !target.matches(".menu"); // elements can be dropped in any container
    by default
  },
  invalid: function (el, handle) {
    return false; // don't prevent any drags from initiating by default
  },
  copy: function (el, source) {
    if (source.matches(".menu")) {
      return true;
    }
    return false;
  },
  copySortSource: false, // elements in copy-source containers can be reordered
  revertOnSpill: false, // spilling will put the element back where it was
  dragged from, if this is true
  removeOnSpill: true, // spilling will `.remove` the element, if this is true
});
drake.on("drop", () => trigger("scriptChanged", document.body));
drake.on("remove", () => trigger("scriptChanged", document.body));

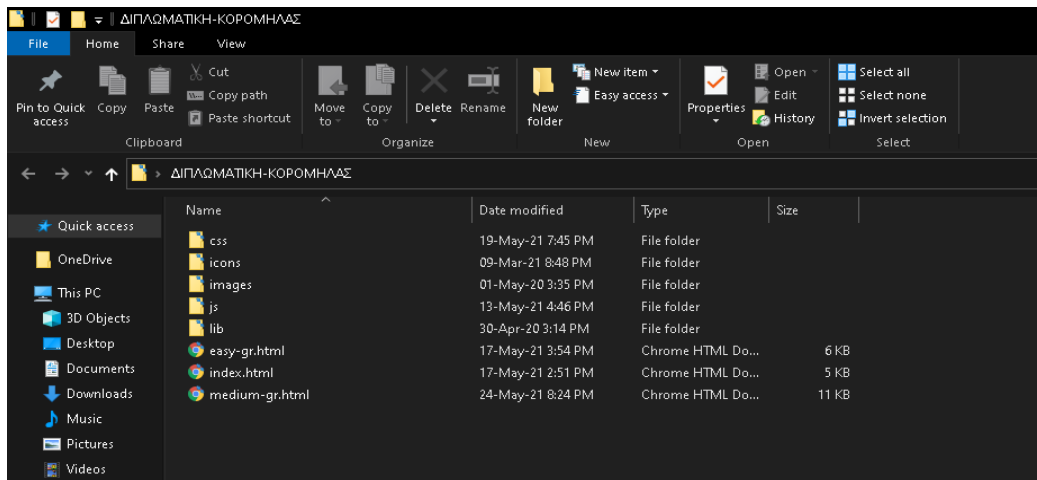
```

## Οργάνωση αρχείων

Για την σωστή λειτουργία της ιστοσελίδας, θα πρέπει να γίνει η οργάνωση των αρχείων κατά ομάδες. Οπότε πρέπει να δημιουργηθεί ένας φάκελος ο οποίος περιέχει:

- 1) Ένα φάκελο με ονομασία “CSS” που περιέχει το `template.css`.
- 2) Ένα φάκελο με ονομασία “icons” που περιέχει τα εικονίδια επιλογής μας.
- 3) Ένα φάκελο με ονομασία “images” που περιέχει τις φωτογραφίες.
- 4) Ένα φάκελο με ονομασία “js” που περιέχει τα `blocks-medium.js`, `menu-medium.js`, `turtle-medium.js`, `file.js`, `drag.js`, `util.js` και `socket.js`.
- 5) Ένα φάκελο με ονομασία “lib” που περιέχει την βιβλιοθήκη `drag.min.css` και `drag.min.js`.
- 6) Τέλος τα τρία αρχεία `index.html`, `easy-gr.html` και `medium-gr.html`.

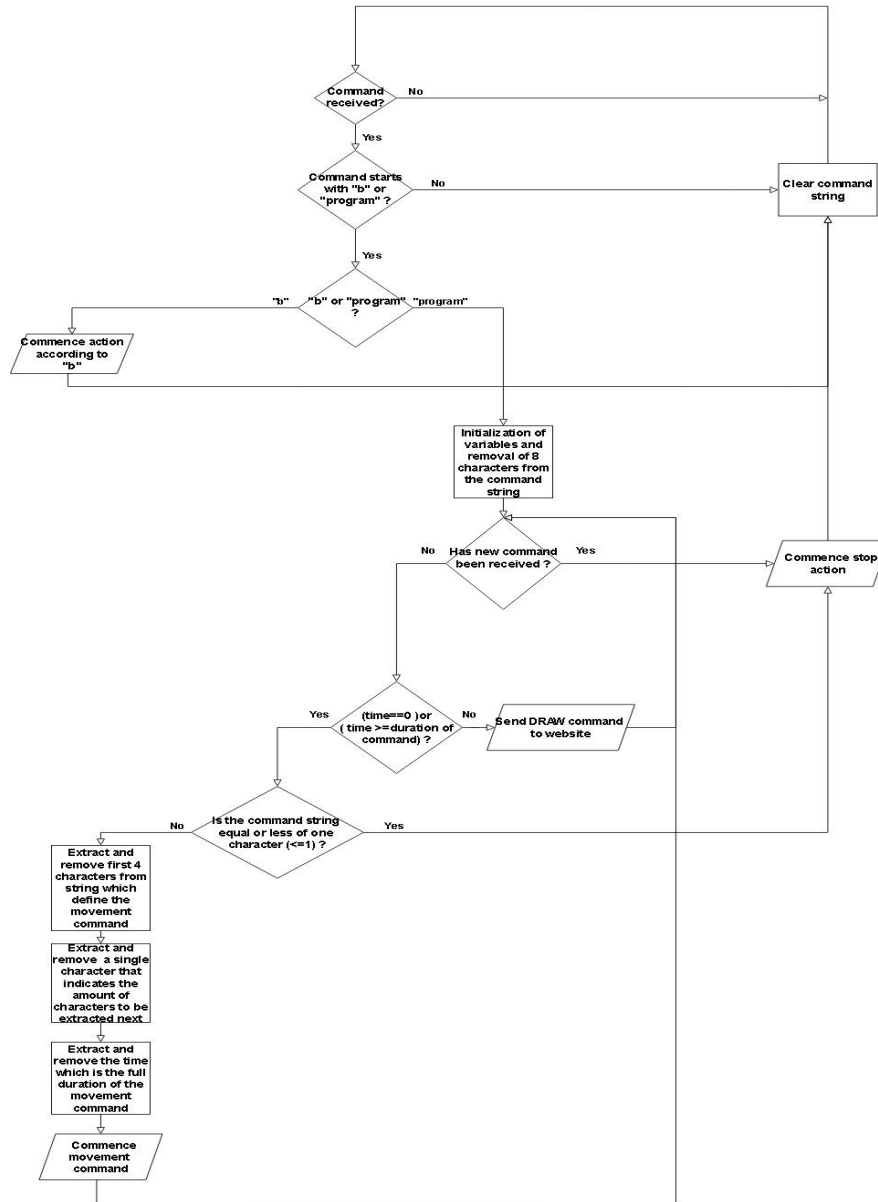
Παραθέεται φωτογραφία ως σημείο αναφοράς :



Εικόνα 30 - Οργάνωση αρχείων ιστοσελίδας

## Διάγραμμα Ροής Ρομποτικού Οχήματος

Το διάγραμμα που ακολουθεί, αναλύει συνοπτικά την δομή του προγράμματος που χρησιμοποιεί το ρομποτικό όχημα ώστε όταν του δοθεί η εντολή, να εκτελεί τις κατάλληλες κινήσεις και να ενημερώνει την ιστοσελίδα ανά τακτά χρονικά διαστήματα για την θέση που βρίσκεται.



Εικόνα 31 - Διάγραμμα ροής ρομποτικού οχήματος

## Κώδικας Προγράμματος ESP32-CAM

Ο κώδικας που θα περαστεί στους μικροεπεξεργαστές ESP32-CAM και αποτελεί την λογική της λειτουργίας τους, είναι διαφορετικός και χωρίζεται στα κομμάτια του ελέγχου του ρομποτικού οχήματος και στον έλεγχο των περιφερικών καμερών. Αρχικά για την σύνταξη του απαιτούνται γνώσεις C++ [26] και χρήση του προγραμματιστικού περιβάλλοντος Arduino IDE [27].

Για την υλοποίηση της εργασίας έχουν χρησιμοποιηθεί ως «κορμός» οι βιβλιοθήκες που, ελέγχουν τις κάμερες με ονομασία esp\_camera.h [28], δημιουργούν ασύγχρονο δίκτυο όπως το

ESPAsyncWebserver.h [29] και το συμπληρωματικό για την εφαρμογή του δικτύου AsyncTCP.h [30], τα οποία δεν αποτελούν προϊόν πνευματικής ιδιοκτησίας του συγγραφέα της εργασίας.

## Έλεγχος ρομποτικού οχήματος

Ο παρακάτω πηγαίος κώδικας αναπτύχθηκε για την χρήση του στο ρομποτικό όχημα και αποτελείται από τις λειτουργίες κίνησης και αμφίδρομης επικοινωνίας. Διαβάζοντας το εισερχόμενο μήνυμα, το πρόγραμμα επιλεγεί ποια θα είναι η επιθυμητή κίνηση βάσει αναζήτησης συγκεκριμένων στοιχείων. Μετέπειτα εκτελείται η αντίστοιχη ενέργεια και ταυτόχρονα αποστέλλεται ανά ταχτά χρονικά διαστήματα των 99 ms μια αναφορά της κίνησης στην ιστοσελίδα.

```

/***** ESP32 Project
    Program and control a robotic vehicle via any device with browser

    Purpose:
    Create a robotic vehicle which can be controlled and programmed wirelessly
    via the internet.
    The mC (ESP32-CAM) initially forms a connection with the website by using
    websocket protocols,
    then waits for the message that contains certain characters. The message can
    either start with "b" or "program"
    and depending on the message it acts differently.

    Examples:
    String "b=Forward"
    String "program=Forward3700Backward42000Stop43000Right3700e"
    The number that comes after the movement command indicates how many
    characters need to be collected to get the
    full value of the duration. The program message always ends with the
    character "e".

    Hardware:
    The mC (ESP32-CAM) has 4 pins connected to a L293D motor driver
    pins(12,13,14,15)
    which are necessary for the mC to control the movement of the vehicle.

    Software:
    Arduino standard library calls like millis(), digitalWrite(),etc.
    WiFi library calls for internet purposes.
    ESP_camera.h library calls for camera feedback.
    ESPAsyncWebServer.h and AsyncTCP.h calls to create an asynchronous websocket
    connection.

    Reference
    Giwrgos Koromilas, June. 2021.
    *****/

#include "esp_camera.h"           //Include camera library
#include <WiFi.h>                 //Include the Wi-Fi library
#include "esp_timer.h"
#include "ESPAsyncWebServer.h"   //Include Async Webserver library
#include <AsyncTCP.h>            //Include AsyncTCP library(MUST HAVE for
ESPAsyncWebserver)
```

```

const char* ssid = "net43";           //Replace with your network
credentials
const char* password = "q123123123"; //Replace with your network
credentials

#define PWDN_GPIO_NUM    32           //Defining internal pins of the
ESP32-CAM
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM      5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

/***** function Forward()

    Purpose:
    Gives the pins that connect to L293D the corresponding states for the robotic
    vehicle to move forward.

    Arguments:
    None.

    Results:
    None.

    Hardware:
    The mC (ESP32-CAM) has 4 pins connected to a L293D motor driver
pins(12,13,14,15)
    which are necessary for the mC to control the movement of the vehicle.

    Software:
    Arduino standard library calls like millis(), digitalWrite(),etc.

    Reference
    Giwrgos Koromilas, June. 2021.
*****/
void Forward() {
    Serial.write("FORWARD");
    digitalWrite(14, LOW);
    digitalWrite(15, LOW);
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    delay(100);
    digitalWrite(15, HIGH);
    digitalWrite(14, LOW);
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
}

/***** function Backward()

```

Purpose:

Gives the pins that connect to L293D the corresponding states for the robotic vehicle to move backward.

Arguments:  
None.

Results:  
None.

Hardware:  
The mC (ESP32-CAM) has 4 pins connected to a L293D motor driver pins(12,13,14,15) which are necessary for the mC to control the movement of the vehicle.

Software:  
Arduino standard library calls like millis(), digitalWrite(),etc.

#### Reference

Giwrgos Koromilas, June. 2021.

```
*****/  
void Backward() {  
  Serial.write("BACKWARD");  
  digitalWrite(14, LOW);  
  digitalWrite(15, LOW);  
  digitalWrite(13, LOW);  
  digitalWrite(12, LOW);  
  delay(100);  
  digitalWrite(15, LOW);  
  digitalWrite(14, HIGH);  
  digitalWrite(12, LOW);  
  digitalWrite(13, HIGH);  
}
```

```
/***** function Left()
```

Purpose:  
Gives the pins that connect to L293D the corresponding states for the robotic vehicle to move left.

Arguments:  
None.

Results:  
None.

Hardware:  
The mC (ESP32-CAM) has 4 pins connected to a L293D motor driver pins(12,13,14,15) which are necessary for the mC to control the movement of the vehicle.

Software:  
Arduino standard library calls like millis(), digitalWrite(),etc.

#### Reference

Giwrgos Koromilas, June. 2021.

```
*****/  
void Left() {  
  Serial.write("LEFT");  
  digitalWrite(14, LOW);  
  digitalWrite(15, LOW);  
  digitalWrite(13, LOW);  
}
```

```

digitalWrite(12, LOW);
delay(100);
digitalWrite(15, HIGH);
digitalWrite(14, LOW);
}

/***** function Right()

    Purpose:
    Gives the pins that connect to L293D the corresponding states for the robotic
    vehicle to move
    right.

    Arguments:
    None.

    Results:
    None.

    Hardware:
    The mC (ESP32-CAM) has 4 pins connected to a L293D motor driver
pins(12,13,14,15)
    which are necessary for the mC to control the movement of the vehicle.

    Software:
    Arduino standard library calls like millis(), digitalWrite(),etc.

    Reference
    Giwrgos Koromilas, June. 2021.
*****/
void Right() {
    Serial.write("RIGHT");
    digitalWrite(14, LOW);
    digitalWrite(15, LOW);
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    delay(100);
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
}

/***** function Stop()

    Purpose:
    Gives the pins that connect to L293D the corresponding states for the robotic
    vehicle to stop
    moving.

    Arguments:
    None.

    Results:
    None.

    Hardware:
    The mC (ESP32-CAM) has 4 pins connected to a L293D motor driver
pins(12,13,14,15)
    which are necessary for the mC to control the movement of the vehicle.

    Software:
    Arduino standard library calls like millis(), digitalWrite(),etc.

    Reference

```

```

    Giwrgos Koromilas, June. 2021.
*****/
void Stop() {
  Serial.write("STOP");
  digitalWrite(14, LOW);
  digitalWrite(15, LOW);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
}

/***** function FFlashOn()

Purpose:
Gives the pin 4 of mC(ESP32-CAM) HIGH state to be used as flashlight.

Arguments:
x    Current state of the pin (Bool type variable)

Results:
x    Returns the variable in true state

Hardware:
Internal pin of mC (ESP32-CAM)

Software:
Arduino standard library calls like millis(), digitalWrite(),etc.

Reference
Giwrgos Koromilas, June. 2021.
*****/
bool FlashOn(bool x) {
  Serial.write("FLASH ON");
  digitalWrite(4, HIGH);
  x = true;
  return x;
}

/***** function FFlashOff()

Purpose:
Gives the pin 4 of mC(ESP32-CAM) LOW state ( close flashlight).

Arguments:
x    Current state of the pin (Bool type variable)

Results:
x    Returns the variable in false state

Hardware:
Internal pin of mC (ESP32-CAM)

Software:
Arduino standard library calls like millis(), digitalWrite(),etc.

Reference
Giwrgos Koromilas, June. 2021.
*****/
bool FlashOff(bool x) {
  Serial.write("FLASH OFF");
  digitalWrite(4, LOW);
  x = false;
  return x;
}

```



```
/***** function ExtractTime()
```

Purpose:

Extract from the incoming string the time value, based on the amount of digits we want to remove.

Arguments:

header The string we are currently supplying the function with.  
y The amount of characters to be removed.

Results:

ms Returns the variable that contains the time value of the string.

Hardware:

None.

Software:

Arduino standard library calls like millis(), digitalWrite(),etc.

Reference

Giwrgos Koromilas, June. 2021.

```
*****/
```

```
unsigned long ExtractTime(String header, int y) {  
    int digits; //Stores the number of digits  
    unsigned long ms ; //Stores the time  
    String sub; // Stores the substring  
    header.remove(0, y); //Remove the wanted characters  
    sub = header.substring(0, 1); //Create substring of one character  
    digits = sub.toInt(); //Translate string to integer  
    header.remove(0, 1); //Remove one character  
    sub = header.substring(0, digits); //Create substring  
    ms = sub.toInt(); //Translate string to integer  
    return ms;  
}
```

```
/***** function ExtractString()
```

Purpose:

Extract from the incoming string the information we want, based on the amount of digits we want to remove.

Arguments:

header The string we are currently supplying the function with.  
y The amount of characters to be removed.

Results:

header Returns the variable that contains the information of the string.

Hardware:

None.

Software:

Arduino standard library calls like millis(), digitalWrite(),etc.

Reference

Giwrgos Koromilas, June. 2021.

```
*****/
```

```
String ExtractString(String header, int y) {  
    int digits; //Stores the number of digits  
    String sub; //Stores the substring  
    header.remove(0, y); //Remove the wanted characters  
    sub = header.substring(0, 1); //Create substring of one character  
    digits = sub.toInt(); //Translate string to integer  
    header.remove(0, 1 + digits); //Remove the wanted characters
```

```

    return header;
}

/***** function CalcInterval()

    Purpose:
    Calculates how how many intervals of 99ms fit in given value

    Arguments:
        ms      Given time in milliseconds
    Results:
        Interval Returns the variable that contains the integer number of intervals

    Hardware:
    None.

    Software:
    None.

    Reference
    Giwrgos Koromilas, June. 2021.
*****/
int CalcInterval(unsigned long ms) {
    int Interval = (ms / 99);
    return Interval;
}

/***** function CalcRemainder()

    Purpose:
    Calculates the remainder of the division between given time and 99ms

    Arguments:
        ms      Given time in milliseconds
    Results:
        Remainder Returns the variable that contains the remainder value

    Hardware:
    None.

    Software:
    None.

    Reference
    Giwrgos Koromilas, June. 2021.
*****/
int CalcRemainder(unsigned long ms) {
    int Remainder = (ms % 99);
    return Remainder;
}

/***** function FillArray()

    Purpose:
    Creates an array containing time values

    Arguments:
        Array      Given empty Array
        Interval   The variable that contains the integer number of intervals
        Remainder  The variable that contains the remainder value
    Results:
        Array      The array has been filled with the corresponding values

```

```

Hardware:
None.

Software:
None.

Reference
Giwrgos Koromilas, June. 2021.
*****/
void FillArray(int * Array, int Interval, int Remainder) {
    if (Remainder == 0) { //Fills the Array with the value
99
        for ( int i = 0; i < Interval; i++)
        {
            Array[i] = 99;
        }
    }
    else { //Fills the Array with the value
99 until the last index where */
        for ( int i = 0; i < Interval; i++ ) { /* it stores 99 + Remainder */
            if (i == (Interval - 1)) {
                Array[i] = 99 + Remainder;
            }
            else {
                Array[i] = 99;
            }
        }
    }
}

// ----- Start Async Webserver -----
AsyncWebServer server(2210); //Start webserver in port 2210
AsyncWebSocket ws("/ws"); //Webserver uses websocket protocol
AsyncWebSocketClient * Myclient = NULL; //Create a variable with AsyncWeb
class
String Command = ""; //Stores the information it gets
from the website
void onEvent(AsyncWebSocket * server, AsyncWebSocketClient * client,
AwsEventType type, void * arg, uint8_t *data, size_t len) {
    if (type == WS_EVT_CONNECT) {
        //client connected
        Serial.printf("ws[%s][%u] connect\n", server->url(), client->id());
        client->printf("Hello Client %u :)", client->id());
        client->ping();
        Myclient = client;
    } else if (type == WS_EVT_DISCONNECT) {
        //client disconnected
        Serial.printf("ws[%s][%u] disconnect: %u\n", server->url(), client->id());
    } else if (type == WS_EVT_ERROR) {
        //error was received from the other end
        Serial.printf("ws[%s][%u] error(%u): %s\n", server->url(), client->id(),
*(uint16_t*)arg, (char*)data);
    } else if (type == WS_EVT_DATA) {
        //data packet
        AwsFrameInfo * info = (AwsFrameInfo*)arg;
        if (info->final && info->index == 0 && info->len == len) {
            //the whole message is in a single frame and we got all of it's data
            // Serial.printf("ws[%s][%u] %s-message[%llu]: ", server->url(),
client->id(), (info->opcode == WS_TEXT)?"text":"binary", info->len);
            if (info->opcode == WS_TEXT) {
                data[len] = 0;
                Serial.printf("%s\n", (char*)data);

                Command = (char *)data; //Store the information

```

```

        } else {
            for (size_t i = 0; i < info->len; i++) {
                Serial.printf("%02x ", data[i]);
            }
            Serial.printf("\n ");
        }
    }
}
}
// ----- Start Camera Buffer -----

typedef struct {
    camera_fb_t * fb;
    size_t index;
} camera_frame_t;

#define PART_BOUNDARY "12345678900000000000000987654321"
static const char* STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* STREAM_PART = "Content-Type: %s\r\nContent-
Length: %u\r\n\r\n";

static const char * JPG_CONTENT_TYPE = "image/jpeg";

class AsyncBufferResponse: public AsyncAbstractResponse {
private:
    uint8_t * _buf;
    size_t _len;
    size_t _index;
public:
    AsyncBufferResponse(uint8_t * buf, size_t len, const char * contentType) {
        _buf = buf;
        _len = len;
        _callback = nullptr;
        _code = 200;
        _contentLength = _len;
        _contentType = contentType;
        _index = 0;
    }
    ~AsyncBufferResponse() {
        if (_buf != nullptr) {
            free(_buf);
        }
    }
    bool _sourceValid() const {
        return _buf != nullptr;
    }
    virtual size_t _fillBuffer(uint8_t *buf, size_t maxLen) override {
        size_t ret = _content(buf, maxLen, _index);
        if (ret != RESPONSE_TRY_AGAIN) {
            _index += ret;
        }
        return ret;
    }
    size_t _content(uint8_t *buffer, size_t maxLen, size_t index) {
        memcpy(buffer, _buf + index, maxLen);
        if ((index + maxLen) == _len) {
            free(_buf);
            _buf = nullptr;
        }
        return maxLen;
    }
};

```

```

    }
};

class AsyncFrameResponse: public AsyncAbstractResponse {
private:
    camera_fb_t * fb;
    size_t _index;
public:
    AsyncFrameResponse(camera_fb_t * frame, const char * contentType) {
        _callback = nullptr;
        _code = 200;
        _contentLength = frame->len;
        _contentType = contentType;
        _index = 0;
        fb = frame;
    }
    ~AsyncFrameResponse() {
        if (fb != nullptr) {
            esp_camera_fb_return(fb);
        }
    }
    bool _sourceValid() const {
        return fb != nullptr;
    }
    virtual size_t _fillBuffer(uint8_t *buf, size_t maxLen) override {
        size_t ret = _content(buf, maxLen, _index);
        if (ret != RESPONSE_TRY_AGAIN) {
            _index += ret;
        }
        return ret;
    }
    size_t _content(uint8_t *buffer, size_t maxLen, size_t index) {
        memcpy(buffer, fb->buf + index, maxLen);
        if ((index + maxLen) == fb->len) {
            esp_camera_fb_return(fb);
            fb = nullptr;
        }
        return maxLen;
    }
};

class AsyncJpegStreamResponse: public AsyncAbstractResponse {
private:
    camera_frame_t _frame;
    size_t _index;
    size_t _jpg_buf_len;
    uint8_t * _jpg_buf;
    uint64_t lastAsyncRequest;
public:
    AsyncJpegStreamResponse() {
        _callback = nullptr;
        _code = 200;
        _contentLength = 0;
        _contentType = STREAM_CONTENT_TYPE;
        _sendContentLength = false;
        _chunked = true;
        _index = 0;
        _jpg_buf_len = 0;
        _jpg_buf = NULL;
        lastAsyncRequest = 0;
        memset(&_frame, 0, sizeof(camera_frame_t));
    }
    ~AsyncJpegStreamResponse() {

```

```

    if (_frame.fb) {
        if (_frame.fb->format != PIXFORMAT_JPEG) {
            free(_jpg_buf);
        }
        esp_camera_fb_return(_frame.fb);
    }
}
bool _sourceValid() const {
    return true;
}
virtual size_t _fillBuffer(uint8_t *buf, size_t maxLen) override {
    size_t ret = _content(buf, maxLen, _index);
    if (ret != RESPONSE_TRY_AGAIN) {
        _index += ret;
    }
    return ret;
}
size_t _content(uint8_t *buffer, size_t maxLen, size_t index) {
    if (!_frame.fb || _frame.index == _jpg_buf_len) {
        if (index && _frame.fb) {
            uint64_t end = (uint64_t)micros();
            int fp = (end - lastAsyncRequest) / 1000;
            log_printf("Size: %uKB, Time: %ums (%.1ffps)\n", _jpg_buf_len / 1024,
fp);
            lastAsyncRequest = end;
            if (_frame.fb->format != PIXFORMAT_JPEG) {
                free(_jpg_buf);
            }
            esp_camera_fb_return(_frame.fb);
            _frame.fb = NULL;
            _jpg_buf_len = 0;
            _jpg_buf = NULL;
        }
        if (maxLen < (strlen(STREAM_BOUNDARY) + strlen(STREAM_PART) +
strlen(JPG_CONTENT_TYPE) + 8)) {
            //log_w("Not enough space for headers");
            return RESPONSE_TRY_AGAIN;
        }
        //get frame
        _frame.index = 0;

        _frame.fb = esp_camera_fb_get();
        if (_frame.fb == NULL) {
            log_e("Camera frame failed");
            return 0;
        }

        if (_frame.fb->format != PIXFORMAT_JPEG) {
            unsigned long st = millis();
            bool jpeg_converted = frame2jpg(_frame.fb, 80, &_jpg_buf,
&_jpg_buf_len);
            if (!jpeg_converted) {
                log_e("JPEG compression failed");
                esp_camera_fb_return(_frame.fb);
                _frame.fb = NULL;
                _jpg_buf_len = 0;
                _jpg_buf = NULL;
                return 0;
            }
            log_i("JPEG: %lums, %uB", millis() - st, _jpg_buf_len);
        } else {
            _jpg_buf_len = _frame.fb->len;
            _jpg_buf = _frame.fb->buf;

```

```

    }

    //send boundary
    size_t blen = 0;
    if (index) {
        blen = strlen(STREAM_BOUNDARY);
        memcpy(buffer, STREAM_BOUNDARY, blen);
        buffer += blen;
    }
    //send header
    size_t hlen = sprintf((char *)buffer, STREAM_PART, JPG_CONTENT_TYPE,
_jpg_buf_len);
    buffer += hlen;
    //send frame
    hlen = maxLen - hlen - blen;
    if (hlen > _jpg_buf_len) {
        maxLen -= hlen - _jpg_buf_len;
        hlen = _jpg_buf_len;
    }
    memcpy(buffer, _jpg_buf, hlen);
    _frame.index += hlen;
    return maxLen;
}

size_t available = _jpg_buf_len - _frame.index;
if (maxLen > available) {
    maxLen = available;
}
memcpy(buffer, _jpg_buf + _frame.index, maxLen);
_frame.index += maxLen;

return maxLen;
}
};

void streamJpg(AsyncWebServerRequest *request) {
    AsyncJpegStreamResponse *response = new AsyncJpegStreamResponse();
    if (!response) {
        request->send(501);
        return;
    }
    response->addHeader("Access-Control-Allow-Origin", "*");
    request->send(response);
}
AsyncWebServer server2(2211); //Start second webserver in port 2211
void setup() {
    pinMode(4, OUTPUT);
    pinMode(14, OUTPUT);
    pinMode(15, OUTPUT);
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);

    // Set outputs to LOW
    digitalWrite(4, LOW);
    digitalWrite(14, LOW);
    digitalWrite(15, LOW);
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    Serial.begin(115200);
    Serial.setDebugOutput(false);
    camera_config_t config; //Config contains all the below
    configurations of the camera module
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;

```

```

config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_CIF;
config.jpeg_quality = 10;
config.fb_count = 1;
esp_err_t err = esp_camera_init(&config);    // Camera initialization
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

WiFi.begin(ssid, password);                // Wi-Fi connection
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.print("Go to: http://");
Serial.println(WiFi.localIP());
ws.onEvent(onEvent);                      // Start streaming web server
server.addHandler(&ws);
server2.on("/", HTTP_GET, streamJpg);
server.begin();                            // Run server (communications)
server2.begin();                           // Run server 2 (camera feed)
}
bool flashlight = false;                   //Stores flashlight state
void loop() {
    if (Command != "")
    {
        if (Command.indexOf("b=") != -1) {
            if (Command.indexOf("b=flash") != -1)
            {
                if (flashlight == false)
                {
                    flashlight = FlashOn(flashlight);
                }
                else
                {
                    flashlight = FlashOff(flashlight);
                }
            }
            else if (Command.indexOf("b=forward") != -1)
            {
                Forward();
            }
            else if (Command.indexOf("b=backward") != -1)

```



```

    {
        Backward();
    }
    else if (Command.indexOf("b=right") != -1)
    {
        Right();
    }
    else if (Command.indexOf("b=left") != -1)
    {
        Left();
    }
    else if (Command.indexOf("b=stop") != -1)
    {
        Stop();
    }
}
else if (Command.indexOf("program=") != -1) {
    bool finished = false;           //Stores the state of the program
    int digits;                       //Stores the number of digits
    unsigned long ms = 0;             //Stores time
    unsigned long StartTime = 0;      //Stores time relative to the start
of command
    String sub;                       //Stores substring
    String Command2 = Command;        //Stores the information from the
website
    String Command3 = Command;        //Stores the information from the
website
    int Interval = 0;                 //Stores the integer interval count
    int Remainder = 0;                //Stores the remainder
    int ArrayIndex = 0;               //Stores the index of the array
    int* InterArray = 0;              //Stores time values (the time it
gets to complete each command)
    Command3.remove(0, (Command.indexOf("program=") + 8)); //Removes the
"program=" characters from the string
    Serial.println(Command3);
    do {
        if (Command2 == Command) {           //The webserver is
asynchronous so the command variable is changed if a new message comes from the
website
            if (ms != 0) {
                if (millis() >= (StartTime + ((ArrayIndex + 1) * 99 ))) { //Checks
if its time to proceed to the next movement of the command string
                    String str = "DRAW" + sub + String(InterArray[ArrayIndex]);
//Sends to the website information to draw on the canvas
                    Serial.println(str);
                    Myclient->text(str);
                    ArrayIndex = ArrayIndex + 1;
                }
            }
            if ((ArrayIndex == 0) || (ArrayIndex == Interval)) { //Proceeds only
if the array is empty or has been used
                if ((ms == 0) || (millis() >= ms))
                {
                    if (Command3.length() <= 1)
                    {
                        goto telos;           //Jumps to line
951 when the string consists with one character (e for end)
                    }
                    sub = Command3.substring(0, 4); //Gets the first 4
characters (example: Forward41000 => reads Forw)
                    //Serial.println(sub);

                    if (sub == "flas")

```

```

{
  if (flashlight == false)
  {
    flashlight = FlashOn(flashlight);
    Command3.remove(0, 4);
  }
  else
  {
    flashlight = FlashOff(flashlight);
    Command3.remove(0, 4);
  }
}
else if (sub == "Forw")
{
  ms = ExtractTime(Command3, 7);           //Stores the time (we
remove 7 char because forward has 7 char)
  Command3 = ExtractString(Command3, 7);   //Stores the wanted
string
  Interval = CalcInterval(ms);             //Calculates intervals
  Remainder = CalcRemainder(ms);           //Calculates remainder
  delete[] InterArray;                     //The Array is dynamic
so to resize it we need to delete it first
  InterArray = new int[Interval];          //Recreating the array
  ArrayIndex = 0;
  FillArray(InterArray, Interval, Remainder); //Fill the array
with the time variables
  Serial.write("FOR");
  Serial.print(ms);
  Forward();                               //Robotic vehicle
goes forward
  StartTime = millis();                    //Measure the exact
time on the command
  ms = ms + StartTime;                     //Measure the end
time of the command
}
else if (sub == "Back")
{
  ms = ExtractTime(Command3, 8);
  Command3 = ExtractString(Command3, 8);
  Interval = CalcInterval(ms);
  Remainder = CalcRemainder(ms);
  delete[] InterArray;
  InterArray = new int[Interval];
  ArrayIndex = 0;
  FillArray(InterArray, Interval, Remainder);
  Serial.write("FOR");
  Serial.print(ms);
  Backward();
  StartTime = millis();
  ms = ms + millis();
}
else if (sub == "Righ")
{
  ms = ExtractTime(Command3, 5);
  Command3 = ExtractString(Command3, 5);
  Interval = CalcInterval(ms);
  Remainder = CalcRemainder(ms);
  delete[] InterArray;
  InterArray = new int[Interval];
  ArrayIndex = 0;
  FillArray(InterArray, Interval, Remainder);
  Serial.write("FOR");
  Serial.print(ms);
}
}

```

```

        Right();
        StartTime = millis();
        ms = ms + millis();
    }
    else if (sub == "Left")
    {
        ms = ExtractTime(Command3, 4);
        Command3 = ExtractString(Command3, 4);
        Interval = CalcInterval(ms);
        Remainder = CalcRemainder(ms);
        delete[] InterArray;
        InterArray = new int[Interval];
        ArrayIndex = 0;
        FillArray(InterArray, Interval, Remainder);
        Serial.write("FOR");
        Serial.print(ms);
        Left();
        StartTime = millis();
        ms = ms + millis();
    }
    else if (sub == "Stop")
    {
        ms = ExtractTime(Command3, 4);
        Command3 = ExtractString(Command3, 4);
        Interval = CalcInterval(ms);
        Remainder = CalcRemainder(ms);
        delete[] InterArray;
        InterArray = new int[Interval];
        ArrayIndex = 0;
        FillArray(InterArray, Interval, Remainder);
        Serial.write("FOR");
        Serial.print(ms);
        Stop();
        StartTime = millis();
        ms = ms + millis();
    }
    }
}
}
else {
telos: //Jump to save time
      //flashlight=FlashOff(flashlight); //Closes flashlight if left open
(disabled)
      Stop(); //When the program ends a
permanent stop is issued until next command
      finished = true; //Stores that the program has
ended
    }
} while (finished == false);
}
Command = ""; //Clears the command variable
}
}

```

## Έλεγχος περιφερικών καμερών

Ο παρακάτω πηγαίος κώδικας αναπτύχθηκε για την χρήση του στα δυο ESP32-CAM που λειτουργούν ως περιφερικές κάμερες και αποτελείται μόνο από μια λειτουργία μονόδρομης επικοινωνίας. Οι μικροεπεξεργαστές από την στιγμή της λειτουργίας τους, αναμεταδίνουν συνεχώς την οπτική εικόνα του χώρου σε μια διεύθυνση στο διαδίκτυο. Η συγκεκριμένη διεύθυνση

μετέπειτα πρέπει να εισαχθεί στην ιστοσελίδα διότι για λόγους ασφάλειας δεν είναι στατική , συνεπώς θα πρέπει να εισάγεται η νέα διεύθυνση κάθε φορά που αλλάζει.

```
/***** ESP32 Project
ESP32-CAM stream
```

**Purpose:**

Create a program which redirects the visual feedback from the camera module to a computer port.

Furthermore we can port forward the selected port by adjusting the router options and be able to access it globally and remotely.

**Hardware:**

None.

**Software:**

Arduino standard library calls like millis(), digitalWrite(),etc.

WiFi library calls for internet purposes.

ESP\_camera.h library calls for camera feedback.

Esp\_http\_server.h calls to create a streaming webserver.

**Reference**

Giwrgos Koromilas, June. 2021.

```
*****/
```

```
#include "esp_camera.h" //Include camera library
#include <WiFi.h> //Include the Wi-Fi library
#include "esp_timer.h"
#include "esp_http_server.h" //Include Webserver library

const char* ssid = "net43"; //Replace with your network credentials
const char* password = "q123123123"; //Replace with your network credentials

// ----- Start Camera Buffer -----
#define PART_BOUNDARY "1234567890000000000000987654321"
#define PWDN_GPIO_NUM 32 //Defining internal pins of the ESP32-
CAM
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-
Length: %u\r\n\r\n";
httpd_handle_t stream_httpd = NULL;
```

```

static esp_err_t stream_handler(httpd_req_t *req) {
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if (res != ESP_OK) {
        return res;
    }

    while (true) {
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println("Camera capture failed");
            res = ESP_FAIL;
        } else {

            if (fb->format != PIXFORMAT_JPEG) {
                bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
                esp_camera_fb_return(fb);
                fb = NULL;
                if (!jpeg_converted) {
                    Serial.println("JPEG compression failed");
                    res = ESP_FAIL;
                }
            } else {
                _jpg_buf_len = fb->len;
                _jpg_buf = fb->buf;
            }

        }

        if (res == ESP_OK) {
            size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
            res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
        }
        if (res == ESP_OK) {
            res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
        }
        if (res == ESP_OK) {
            res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
        }
        if (fb) {
            esp_camera_fb_return(fb);
            fb = NULL;
            _jpg_buf = NULL;
        } else if (_jpg_buf) {
            free(_jpg_buf);
            _jpg_buf = NULL;
        }
        if (res != ESP_OK) {
            break;
        }
        //Serial.printf("MJPG: %uB\n", (uint32_t) (_jpg_buf_len));
    }
    return res;
}

// ----- Start Camera server -----
void startCameraServer() {
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

```

```

    config.server_port = 4090;                //Define in which port the webstream
will be
    httpd_uri_t index_uri = {
        .uri      = "/",
        .method   = HTTP_GET,
        .handler  = stream_handler,
        .user_ctx = NULL
    };
    //Serial.printf("Starting web server on port: '%d'\n", config.server_port);
    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(stream_httpd, &index_uri);
    }
}
void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(false);
    camera_config_t config;                //Config contains all the below
configurations of the camera module
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;
    config.frame_size = FRAMESIZE_CIF;    //We change frame size (QVGA,VGA,CIF,
etc.)
    config.jpeg_quality = 10;            //We can change the quality to 1-60
value (less the better)
    config.fb_count = 1;
    esp_err_t err = esp_camera_init(&config); // Camera initialization
    if (err != ESP_OK) {
        Serial.printf("Camera init failed with error 0x%x", err);
        return;
    }
    WiFi.begin(ssid, password);        // Wi-Fi connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.print("Camera Stream Ready! Go to: http://");
    Serial.print(WiFi.localIP());
    startCameraServer();                // Start streaming web server
}
void loop() {
    // put your main code here, to run repeatedly (empty)
}

```

# ΡΥΘΜΙΣΕΙΣ ΤΟΠΙΚΟΥ ΔΙΚΤΥΟΥ ΚΑΙ ΕΦΑΡΜΟΓΗ ΧΑΜΡΡ

Η λειτουργικότητα της διπλωματικής εργασίας αποδίδεται στην δυνατότητα να χρησιμοποιείται από οποιονδήποτε χρήστη του διαδικτύου, με την μονή απαίτηση να γίνεται η χρήση browser [6]. Για να γίνει εφικτό αυτό, θα πρέπει να γίνουν κάποιες ρυθμίσεις στο τοπικό δίκτυο και ειδικότερα να τροποποιηθούν οι επιλογές που είναι αρχικοποιημένες στο modem/router [31] του δικτύου που στεγάζεται η εφαρμογή. Εφόσον γίνουν σωστά οι τροποποιήσεις θα πρέπει να χρησιμοποιηθεί ένα μέσο για την παροχή (self-host) της ιστοσελίδας στους πιθανούς χρήστες του διαδικτύου που θα εξυπηρετούνται ως clients.

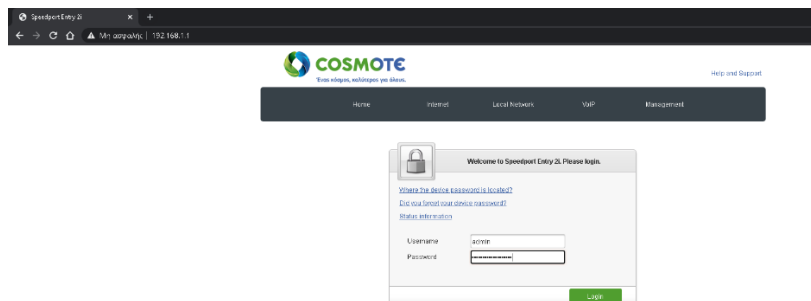
Η ανάλυση αυτής της διαδικασίας ακολουθεί στις παρακάτω δυο ενότητες.

## Ρυθμίσεις Router/Modem

Για την ευκολία της ανάλυσης θα χρησιμοποιηθεί ως παράδειγμα το router/modem που παρέχεται από τον ΟΤΕ, το μοντέλο Speedport Entry 2i [32].

Το πρώτο βήμα είναι, η είσοδος ως διαχειριστής της συσκευής.

- 1) Γίνεται άνοιγμα ενός browser και πληκτρολογείται η διεύθυνση 192.168.1.1
- 2) Μετέπειτα γίνεται εισαγωγή των στοιχείων Router username και Router password που βρίσκονται πάνω στο router.

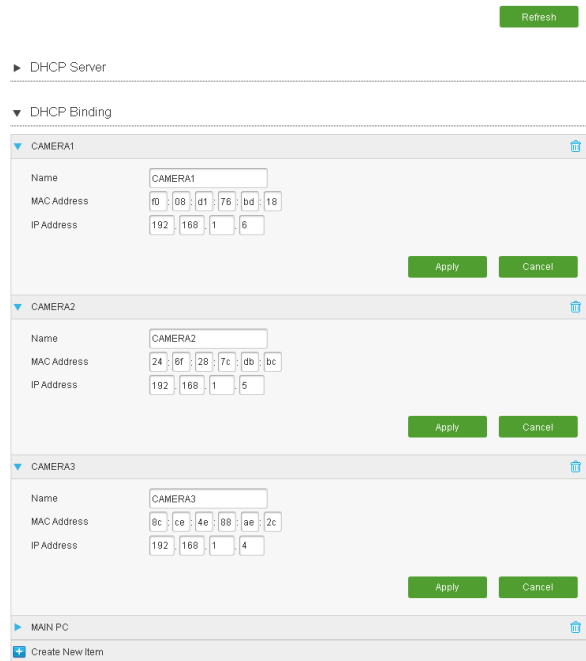


Εικόνα 32 - Είσοδος στο router/modem

Επόμενο βήμα είναι η τροποποίηση των ρυθμίσεων του τοπικού δικτύου

- 1) Γίνεται προσπέλαση στην καρτέλα Local Network που βρίσκεται ψηλά στο κέντρο.
- 2) Επόμενη προσπέλαση στην καρτέλα LAN (Local Area Network) ψηλά και αριστερά.
- 3) Εύρεση στην καρτέλα IPv4, της λίστας Allocated Address (DHCP) με όλες τις συσκευές που είναι συνδεδεμένες στο τοπικό δίκτυο.

- 4) Έχοντας βρει τις συσκευές που σχετίζονται με την διπλωματική εργασία γίνετε καταγραφή των MAC address που κατέχουν.
- 5) Στην ίδια καρτέλα βρίσκεται η λίστα DHCP Binding όπου και θα δημιουργηθούν στατικές διευθύνσεις για την κάθε μια συσκευή.



Εικόνα 33 - Στατικές διευθύνσεις ESP32-CAM

Τέλος, μένει να συνδεθούν οι παραπάνω διευθύνσεις που ρυθμίστηκαν, με συγκεκριμένες θύρες για την προώθηση τους στο διαδίκτυο (port forwarding) [33].

- 1) Αρχική προσπέλαση στην καρτέλα Internet που βρίσκεται ψηλά στο κέντρο.
- 2) Μετέπειτα άνοιγμα της καρτέλας Security ψηλά και αριστερά.
- 3) Πάνω στην εμφανιζόμενη κυλιόμενη μπάρα γίνεται αναζήτηση της καρτέλας Port Forwarding.

Έχοντας φτάσει στην καρτέλα για την δημιουργία των κατάλληλων συνδέσεων πρέπει να ληφθεί υπόψη ποιες θύρες μπορούν ελεύθερα να χρησιμοποιηθούν και ποιες είναι κλειδωμένες για την ασφάλεια του δικτύου. Συνεπώς πρέπει να αντληθούν πληροφορίες από τον πάροχο του εκάστοτε δικτύου. Στην περίπτωση της παρούσας ανάλυσης είναι ο ΟΤΕ και με βάση την ιστοσελίδα [34] δεν μπορεί να γίνει χρήση θυρών με TCP πρωτόκολλο : 1 μέχρι και 1023, 1433, 3389 και 5000, και θυρών με UDP πρωτόκολλο :161, 135 μέχρι και 139, και 1434 χωρίς να υπάρξει συνεννόηση με τον πάροχο για την απασφάλιση τους.

Από την στιγμή που δεν απαιτείται κάποια συγκεκριμένη θύρα, η επιλογή έγινε με γνώμονα την ιστοσελίδα για τις θύρες TCP/UDP [35], ώστε να μην δημιουργηθεί κανένα πρόβλημα με ήδη υπάρχοντες εφαρμογές. Οι θύρες που επιλέχθηκαν για port forward είναι :

- 1) Θύρες 2210-2211 : Για χρήση στο ESP32-CAM πάνω στο ρομποτικό όχημα

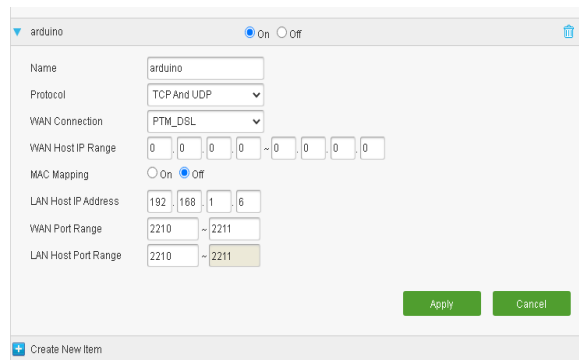


- 2) Θύρες 4089-4090 : Για χρήση στα ESP32-CAM ως περιφερικές κάμερες ( μια το καθένα)
- 3) Θύρες 6444-6445 : Για χρήση στο μέσο που θα παρέχει την ιστοσελίδα στο διαδίκτυο.

Οι παραπάνω θύρες μπορούν να αλλαχθούν κατά προτίμηση αλλά ταυτόχρονα θα πρέπει να τροποποιηθεί ο κώδικας στα ανάλογα προγράμματα.

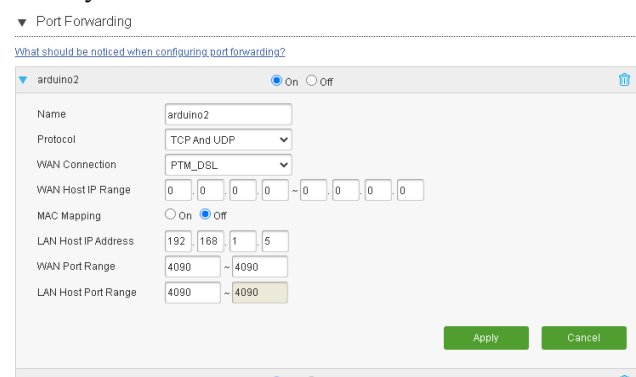
Το μόνο που παραμένει πλέον είναι η δημιουργία των συνδέσεων, όπου θα συμπληρωθούν για κάθε μια συσκευή/εφαρμογή τα απαραίτητα στοιχεία.

- 1) Δημιουργία της κύριας σύνδεσης του ρομποτικού οχήματος.  
 Συμπληρώθηκε στην είσοδο LAN Host IP Address η στατική διεύθυνση που ορίστηκε, στην παρούσα περίπτωση η συσκευή έχει διεύθυνση 192.168.1.6 . Μετέπειτα στο WAN Port Range συμπληρώθηκε με την θύρα που χρησιμοποιεί η συσκευή στον πηγαίο κώδικα . Για λόγο ευκολίας είχαν ορισθεί ακριβώς οι ίδιες θύρες που θα είναι και οι έξοδοι, αρά συμπληρώθηκε με 2210 έως 2211. Τέλος συμπληρώθηκε στο LAN Host Port Range οι θύρες εξόδου 2210 έως 2211.



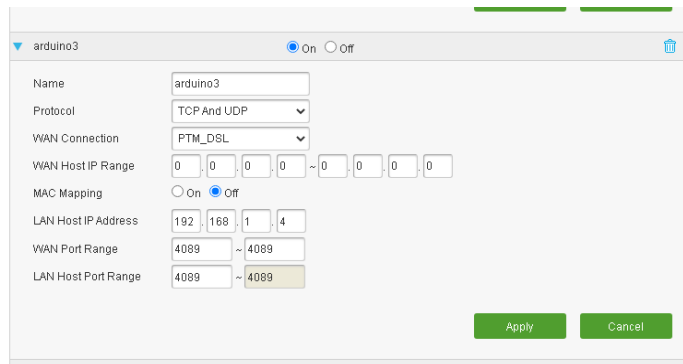
Εικόνα 34 - Port Forward ρομποτικού οχήματος

- 2) Δημιουργία συνδέσεων των περιφερικών καμερών.  
 Α. Συμπληρώθηκε στην είσοδο LAN Host IP Address η στατική διεύθυνση που ορίστηκε, στην παρούσα περίπτωση η συσκευή έχει διεύθυνση 192.168.1.5 . Μετέπειτα στο WAN Port Range συμπληρώθηκε με την θύρα που χρησιμοποιεί η συσκευή στον πηγαίο κώδικα . Για λόγο ευκολίας είχε ορισθεί ακριβώς η ίδια θύρα που θα είναι ως έξοδος, αρά συμπληρώθηκε με 4090 έως 4090. Τέλος συμπληρώθηκε στο LAN Host Port Range οι θύρες εξόδου 4090 έως 4090.



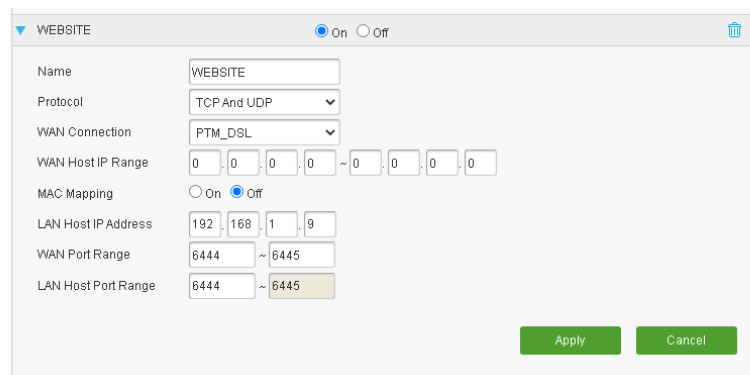
Εικόνα 35 - Port Forward ESP-32 CAMERA 1

Β. Συμπληρώθηκε στην είσοδο LAN Host IP Address η στατική διεύθυνση που ορίστηκε, στην παρούσα περίπτωση η συσκευή έχει διεύθυνση 192.168.1.4 . Μετέπειτα στο WAN Port Range συμπληρώθηκε με την θύρα που χρησιμοποιεί η συσκευή στον πηγαίο κώδικα . Για λόγο ευκολίας είχε ορισθεί ακριβώς η ίδια θύρα που θα είναι ως έξοδος, αρά συμπληρώθηκε με 4089 έως 4089. Τέλος συμπληρώθηκε στο LAN Host Port Range οι θύρες εξόδου 4089 έως 4089.



Εικόνα 36 - Port Forward ESP-32 CAMERA 2

- 3) Δημιουργία σύνδεσης για την προώθηση της ιστοσελίδας  
Συμπληρώθηκε στην είσοδο LAN Host IP Address η διεύθυνση που έχει ο κεντρικός υπολογιστής του δικτύου, διότι θα χρησιμοποιηθεί μια εφαρμογή σε σχέση με πριν που ήταν εξωτερικές συσκευές. Η διεύθυνση του κεντρικού υπολογιστή στην παρούσα περίπτωση είναι 192.168.1.9 . Μετέπειτα στο WAN Port Range συμπληρώθηκαν οι θύρες που χρησιμοποιεί η εφαρμογή, και μπορούν να βρεθούν με την χρήση του Resource Monitor στην καρτέλα του Task Manager . Οι θύρες που χρησιμοποιεί η εφαρμογή είναι 6444 έως 6445. Τέλος, συμπληρώθηκαν στο LAN Host Port Range οι θύρες εξόδου 6444 έως 6445.



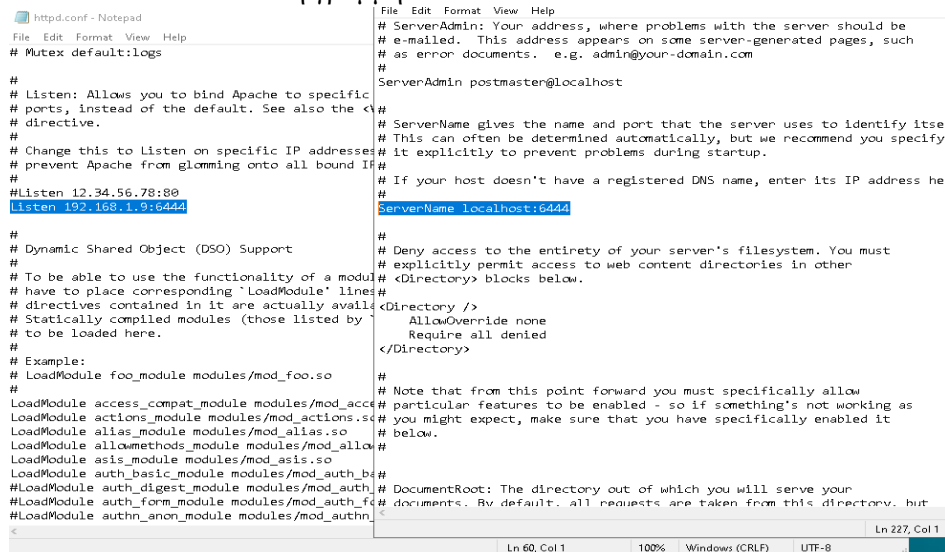
Εικόνα 37 - Port Forward Εφαρμογής

## Ρυθμίσεις εφαρμογής XAMPP

Η εφαρμογή που επιλέχθηκε ως μέσο αποθήκευσης, ελέγχου και προώθησης την ιστοσελίδας ονομάζεται XAMPP [36], και αποτελεί ένα πλήρες πακέτο που περιέχει όλα τα εργαλεία που μπορεί να απαιτηθούν για την διαχείριση της .

Για την σωστή ρύθμιση της εφαρμογής θα πρέπει να γίνουν τα εξής :

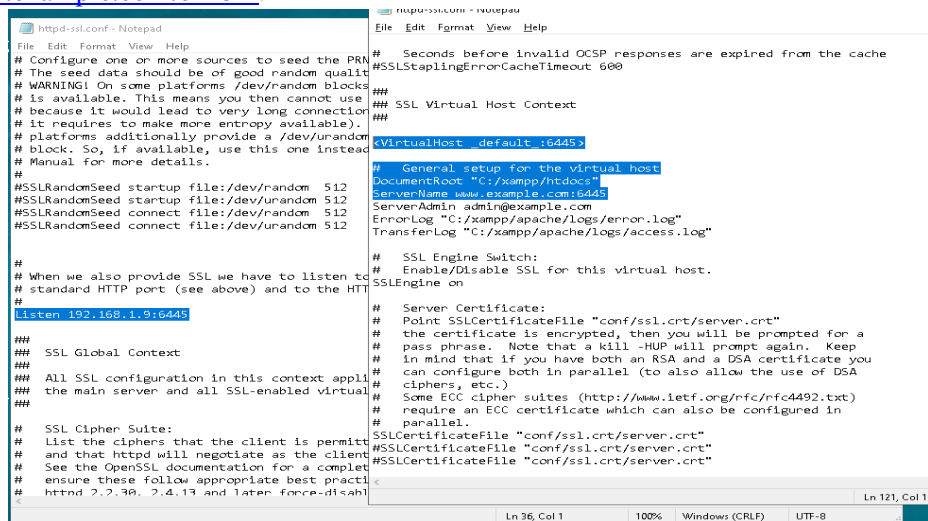
- 1) Αρχικά θα πρέπει να κατεβεί και να εγκατασταθεί το πρόγραμμα στον υπολογιστή που θα παρέχει την ιστοσελίδα.
- 2) Με το πέρας της εγκατάστασης θα ζητηθεί τι είδος δικτύων μπορεί να χρησιμοποιήσει η εφαρμογή. Θα γίνει επιλογή των ιδιωτικών δικτύων αλλά και των δημόσιων.
- 3) Εφόσον ολοκληρώθηκε η παραπάνω διαδικασία, θα γίνει προσπέλαση στο φάκελο που έχουν αποθηκευτεί τα αρχεία με ονομασία xampp . Μετέπειτα θα προσπελαστούν οι φάκελοι apache και conf, ώσπου να βρεθεί το αρχείο httpd.conf .
- 4) Μέσα στο httpd.conf θα πρέπει να τροποποιηθεί η γραμμή 60 σε «Listen 192.168.1.9:6444» αλλά και η γραμμή 227 σε «ServerName localhost:6444».



```
File Edit Format View Help
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin postmaster@localhost
#
# ServerName gives the name and port that the server uses to identify itself
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here
#
ServerName localhost:6444
#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a
# DSO you have to place corresponding 'LoadModule' lines at this location
# so the functions defined in the respective DSO modules are available
# to the compiled server. It is sufficient to load the appropriate shared
# objects only when needed in a specific context, and all the lines with
# external shared libraries must be placed at the same time to load the
# shared object and its dependencies. Uncommmented DSO support examples
# are as follows.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
# LoadModule access_compat_module modules/mod_access_compat.so
# LoadModule actions_module modules/mod_actions.so
# LoadModule alias_module modules/mod_alias.so
# LoadModule allowmethods_module modules/mod_allowmethods.so
# LoadModule asis_module modules/mod_asis.so
# LoadModule auth_basic_module modules/mod_auth_basic.so
# LoadModule auth_digest_module modules/mod_auth_digest.so
# LoadModule auth_form_module modules/mod_auth_form.so
# LoadModule authn_anon_module modules/mod_authn_anon.so
```

Εικόνα 38 - Τροποποίηση httpd.conf αρχείου

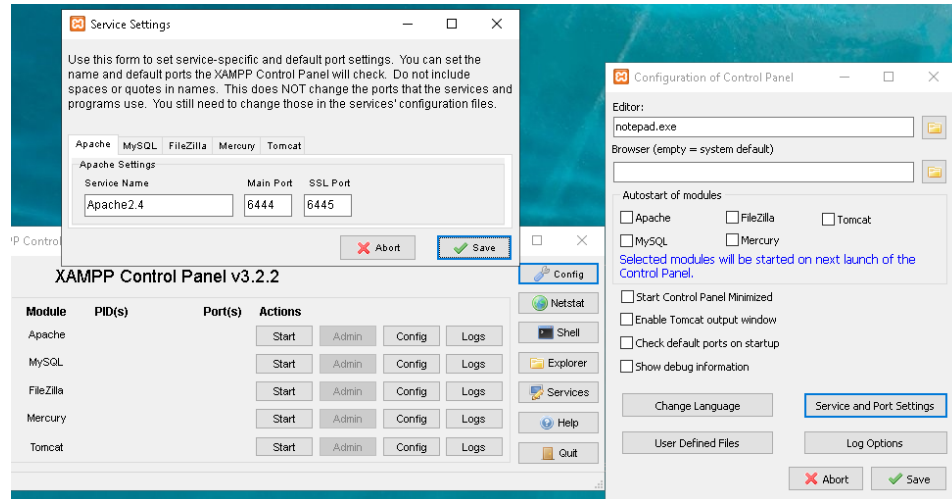
- 5) Στον ίδιο φάκελο conf , γίνεται εισαγωγή στον φάκελο extra και άνοιγμα του httpd-ssl.conf αρχείου .Μετέπειτα θα τροποποιηθεί η γραμμή 36 σε «Listen 192.168.1.9:6445», η γραμμή 121 σε «<VirtualHost \_default\_:6445>» και η γραμμή 125 σε «ServerName www.example.com:6445».



```
File Edit Format View Help
# Seconds before invalid OCSP responses are expired from the cache
#SSLStaplingErrorCacheTimeout 600
###
### SSL Virtual Host Context
###
<VirtualHost _default_:6445>
#
# General setup for the virtual host
DocumentRoot "C:/xampp/htdocs"
#Example: http://example.com:6445
ServerAdmin admin@example.com
ErrorLog "C:/xampp/apache/logs/error.log"
TransferLog "C:/xampp/apache/logs/access.log"
#
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on
#
# Server Certificate:
# Point SSLCertificateFile "conf/ssl.crt/server.crt"
# to the certificate file. If the certificate is encrypted, then you will be prompted for a
# pass phrase. Note that a kill -HUP will prompt again. Keep
# in mind that if you have both an RSA and a DSA certificate you
# can configure both in parallel (to also allow the use of DSA
# ciphers, etc.)
# Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
# require an ECC certificate which can also be configured in
# parallel.
SSLCertificateFile "conf/ssl.crt/server.crt"
SSLCertificateKeyFile "conf/ssl.crt/server.key"
#SSLCertificateFile "conf/ssl.crt/server.crt"
#SSLCertificateFile "conf/ssl.crt/server.crt"
###
### SSL Cipher Suite:
# List the ciphers that the client is permitted to negotiate as the
# client. See the OpenSSL documentation for a complete list.
# ensure these follow appropriate best practices
#
# httnd 2.2.36, 2.4.13 and later force-disable
```

Εικόνα 39 - Τροποποίηση httpd-ssl.conf αρχείου

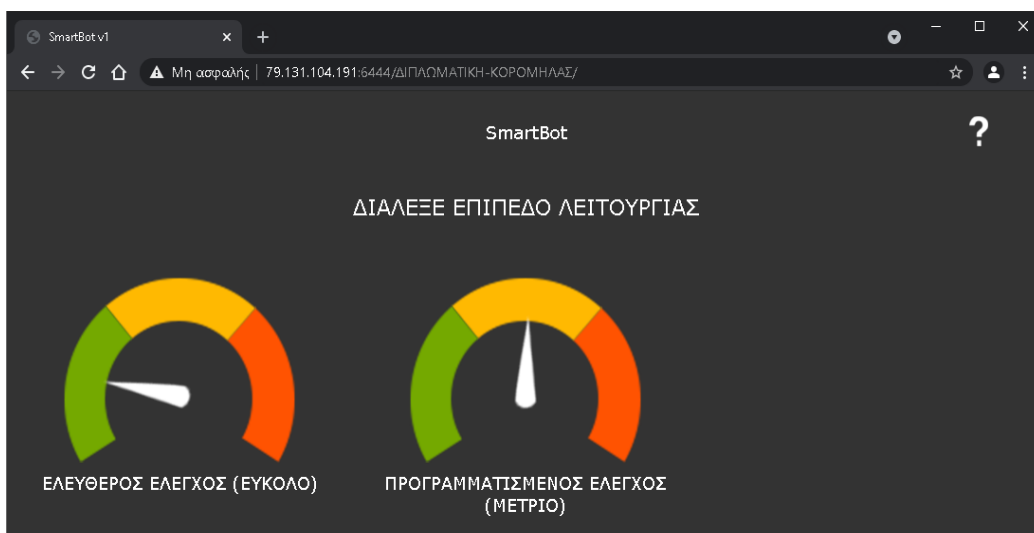
- 6) Τέλος, θα ανοιχθεί η εφαρμογή XAMPP στον υπολογιστή. Στο αναδυόμενο παράθυρο της εφαρμογής πατιέται πάνω δεξιά το κουμπί Config, και από εκεί γίνεται εισαγωγή στο Service and Port Settings. Η μονή καρτέλα που πρέπει να τροποποιηθεί είναι η Apache, όπου και γίνεται ρύθμιση του Main Port στην θύρα :6444 και του SSL Port στην θύρα :6445



Εικόνα 40 - Ρυθμίσεις XAMPP

Έχοντας τελειώσει όλες τις τροποποιήσεις, πρέπει να γίνει αποθήκευση του φάκελου που περιέχει όλα τα αρχεία που αποτελείται η ιστοσελίδα, στον φάκελο htdocs που βρίσκεται μέσα στον κεντρικό φάκελο xampp. Επόμενο βήμα είναι το πάτημα του κουμπιού Start στην εφαρμογή Apache και όλοι οι χρήστες του διαδικτύου μπορούν να έχουν πρόσβαση στην ιστοσελίδα με την προϋπόθεση ότι γνωρίζουν την διεύθυνση του κεντρικού υπολογιστή.

Πληκτρολογώντας την δημόσια διεύθυνση του δικτύου σε οποιαδήποτε συσκευή που υποστηρίζει browser, με την μορφή :XXX:XXX:XXX:XXX:6444/(Ονομασία φακέλου ιστοσελίδας)/ θα παρουσιάσει την ιστοσελίδα της εφαρμογής όπως φαίνεται παρακάτω.



Εικόνα 41 - Άνοιγμα ιστοσελίδας στο διαδίκτυο

## ΔΙΕΚΠΕΡΑΙΩΣΗ

Αρχικά, η παρούσα διπλωματική εργασία αναρτήθηκε στην ιστοσελίδα της σχολής τον Μάρτιο 2021 και είχε ως σκοπό την δημιουργία ενός ρομποτικού οχήματος που μπορούσε να χειρισθεί και να προγραμματισθεί από έναν χρήστη μέσω του διαδικτύου. Κατόπιν συνεννόησης με τον επιβλέπων διδάσκοντα αναλήφθηκε εντός της ιδίου μήνα και επιτόπου ξεκίνησε η ετοιμασία των απαραίτητων μέτρων για την εκπόνηση της.

Το πρώτο βήμα ήταν η ερευνά αγοράς των κατάλληλων ηλεκτρονικών. Μεγάλη βάση δόθηκε στην εύρεση του κατάλληλου μικροεπεξεργαστή ώστε να έχει την δυνατότητα επικοινωνίας με το διαδίκτυο. Η εύρεση του μικροεπεξεργαστή ESP32-CAM αποδείχθηκε ιδανική διότι κάλυπτε τις ανάγκες του διαδικτύου και επέτρεπε την ενσωμάτωση κάμερας , χωρίς να παραλείπεται το ελκυστικό χαμηλό κόστος. Έχοντας βρει το βασικό στοιχείο, η κατασκευή σχεδιάστηκε γύρω από αυτό.

Εφόσον η ερευνά ολοκληρώθηκε, ξεκίνησε η υλοποίηση του κατασκευαστικού κομματιού της εργασίας . Η κατασκευή ολοκληρώθηκε εντός τριών ημερών και αποτελούταν από όλα όσα αναγράφονται στο κεφάλαιο «Σχεδίαση και κατασκευή υλικού μέρους». Επόμενο βήμα ήταν η σύνταξη του λογισμικού που είχε ως σκοπό την αμφίδρομη επικοινωνία του χρήστη με την ρομποτική κατασκευή. Το κομμάτι του λογισμικού χωρίστηκε στην ιστοσελίδα και στο έλεγχο των ESP32-CAM. Ένα από τα ζητούμενα της εργασίας ήταν η ευκολία που πρέπει να παρέχεται στον χρήστη καθ' όλη την χρήση είτε στον απλό έλεγχο είτε στον προγραμματισμό, συνεπώς χρησιμοποιήθηκε ως βάση για την ιστοσελίδα ένα ήδη ολοκληρωμένο περιβάλλον που απευθυνόταν σε εκπαιδευτική ρομποτική όπως αναφέρεται στην ενότητα «Κώδικας Προγράμματος Ιστοσελίδας», όπου προστέθηκε και τροποποιήθηκε ο πηγαίος κώδικας για να καλύπτει τις ανάγκες της εργασίας. Παράλληλα γινόταν και η ανάπτυξη του λογισμικού που προοριζόταν για τους μικροεπεξεργαστές, όπου η μονή απαραίτητη προσθήκη ήταν οι βιβλιοθήκες για τον έλεγχο των καμερών και την δημιουργία ασύγχρονου δικτύου.

Τελικά, μετά από διάστημα δυο μηνών ολοκληρώθηκε η σύνταξη του λογισμικού και αναμενόταν η δοκιμαστική λειτουργία του για την επικύρωση της αποτελεσματικότητας της εργασίας. Για την χρήση του λογισμικού έπρεπε να γίνει τροποποίηση των ρυθμίσεων του modem/router ώστε να δοθούν στατικές διευθύνσεις στις συσκευές και μετέπειτά port forward για να γίνουν προσβάσιμες από το διαδίκτυο. Επιπρόσθετα χρησιμοποιήθηκε η πλατφόρμα XAMPP ως μέσο παροχής της ιστοσελίδας . Όλα τα παραπάνω αναφέρονται αναλυτικά στο κεφάλαιο «Ρυθμίσεις τοπικού δικτύου».

Έχοντας πλέον καταφέρει την σύνδεση της ιστοσελίδας με το παγκόσμιο δίκτυο, «ανέβηκε » ο κώδικας που απευθυνόταν στο ρομποτικό όχημα και στις περιφερικές κάμερες χρησιμοποιώντας έναν USB σε Serial μετατροπέα, λόγω του περιορισμού θυρών που υπήρχε στον μικροεπεξεργαστή ESP32-CAM. Το στάδιο που ακολουθήσε ήταν η επαλήθευση της σωστής λειτουργίας της εργασίας, όπου επικυρώθηκε με την βοήθεια φίλων που κατάφεραν να ελέγχουν και να προγραμματίζουν το ρομποτικό όχημα από την περιοχή της κατοικίας τους. Συνεπώς ο υπόλοιπος χρόνος αφιερώθηκε στην εύρεση και επιδιόρθωση τυχόν λαθών στο κομμάτι του λογισμικού είτε ήταν της ιστοσελίδας είτε της ρομποτικής κατασκευής .

## ΑΠΟΤΕΛΕΣΜΑΤΑ

Σύμφωνα με την δομή της εργασίας, ανακαλύφθηκε ότι :

- 1) Με κόστος κατασκευής γύρω στα πενήντα ευρώ υπάρχει η δυνατότητα υλοποίησης εφαρμογών που μπορούν να χρησιμοποιούν το διαδίκτυο για διάφορες χρήσεις.
- 2) Επιπρόσθετα, η ανάγκη μιας κεντρικής ιστοσελίδας για την διαχείριση των συνδεδεμένων συσκευών αποτέλεσε σημαντικό παράγοντα στη εύρεση της αδυναμίας της επικοινωνίας μεταξύ Server και client χωρίς την χρήση συγκεκριμένου τύπου πρωτοκόλλου όπως το web socket, διότι ο κεντρικός υπολογιστής (Server) αλλάζει θύρα επικοινωνίας κάθε φορά που στέλνει μήνυμα.
- 3) Τέλος, από την πρώτη στιγμή της δοκιμής της εφαρμογής, φάνηκε ότι δεν μπορεί παραπάνω από ένας χρήστης να έχει πρόσβαση στις κάμερες χωρίς να μην έχει σημαντική απώλεια στην ευκρίνεια της οπτικής εικόνας.

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Έχοντας φέρει εις πέρας την ολοκλήρωση της διπλωματικής εργασίας, βγήκε συμπέρασμα ότι οι εφαρμογές που απευθύνονται στην οδήγηση και έλεγχο μηχανολογικού εξοπλισμού με μέσο επικοινωνίας το διαδίκτυο, θα αρχίσουν να εμφανίζονται σε διάφορα επαγγέλματα και τομείς της κοινωνίας. Η ανάγκη τηλεχειρισμού με μεγάλη ακρίβεια ελέγχου, ήδη έχει αρχίσει να εντάσσεται στη καθημερινότητα είτε είναι με την μορφή των μη επανδρωμένων αεροσκαφών γνωστά ως drones ή UAV [37], είτε με την μορφή της εργασίας από απόσταση λόγω της μεταβατικής περιόδου που δημιούργησε η πανδημία του 2020 [38].

Επιπρόσθετα ή ταχύρρυθμη εξέλιξη της τεχνολογίας έχει ήδη αυτοματοποιήσει πολλές επαναλαμβανόμε διεργασίες σε διάφορους τομείς της βιομηχανίας, χωρίς όμως να μπορεί να ανταπεξέλθει με πλήρη αυτονομία σε κάθε πρόβλημα, φερειπείν ενέργειες επιδιόρθωσης ή αδυναμία διαχείρισης απρόσμενων συνθηκών. Σε ένα βαθμό πάντα θα υπάρχει ο ανθρώπινος παράγοντας για την αναγκαία διευθέτηση, συνεπώς ο τηλεχειρισμός θα αποτελέσει ένα μεγάλο βήμα στην διευκόλυνση της εργασίας, χωρίς να απαιτείται η φυσική παρουσία.

Η παρούσα διπλωματική εργασία είχε σκοπό την επίτευξη μιας προσπάθειας για την προσέγγιση των παραπάνω, δηλαδή τον έλεγχο και τον προγραμματισμό μιας μηχανολογικής κατασκευής εξ αποστάσεως με το δυνατότερο χαμηλό κόστος. Μολονότι η λειτουργικότητα ήταν η επιθυμητή, υπήρχαν πολλοί παράγοντες που δεν λήφθηκαν υπόψιν κατά την διάρκεια της σχεδίασης. Το ρομποτικό όχημα σχεδιάστηκε για να χρησιμοποιείται μόνο σε επίπεδη επιφάνεια χωρίς να υπάρχουν εμπόδια στον χώρο λειτουργίας του. Συνεπώς αν υπάρχει θέληση να λειτουργεί σε περιβάλλον διαφορετικό από αυτό που αναφέρθηκε, θα πρέπει να γίνει εισαγωγή κατάλληλων αισθητήρων όπως αισθητήρες απόστασης ή ηλεκτρονικό γυροσκόπιο, ώστε η προγραμματισμένη κίνηση που έχει συντάξει ο χρήστης να έχει επιτυχημένη εκτέλεση χωρίς αποκλίσεις ή απρόσμενα σφάλματα.

Τέλος, εφόσον διανύουμε την 4<sup>η</sup> βιομηχανική επανάσταση και πολλές καινοτομίες έχουν αρχίσει και θα συνεχίσουν να εμφανίζονται γύρω από την χρήση των μερικός ή πλήρης αυτοματοποιημένων drone, οποιοσδήποτε που ενδιαφέρεται να κάνει μια εισαγωγή στον τρόπο σχεδίασης και λειτουργίας μιας τηλεχειριζόμενης κατασκευής ή ψάχνει να χρησιμοποιήσει μια «πλατφόρμα» για την προσθήκη και εξέλιξη διάφορων εφαρμογών ανεξαρτήτως επιστημονικού πεδίου, μπορεί να αναφερθεί στην διπλωματική εργασία

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] WIKIPEDIA, «Internet,» 2021, 28 5. [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Internet>. [Πρόσβαση 4 6 2021].
- [2] WIKIPEDIA, «Fourth Industrial Revolution,» 4 6 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Fourth\\_Industrial\\_Revolution](https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution). [Πρόσβαση 12 6 2021].
- [3] WIKIPEDIA, «Server,» 1 6 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Server\\_\(computing\)](https://en.wikipedia.org/wiki/Server_(computing)). [Πρόσβαση 10 6 2021].
- [4] WIKIPEDIA, «Client,» 5 6 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Client\\_\(computing\)](https://en.wikipedia.org/wiki/Client_(computing)). [Πρόσβαση 10 6 2021].
- [5] WIKIPEDIA, «Websocket,» 24 4 2021. [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/WebSocket>. [Πρόσβαση 10 6 2021].
- [6] WIKIPEDIA, «Browser,» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser). [Πρόσβαση 14 5 2021].
- [7] Wikipedia, «ESP 32,» 21 4 2021. [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/ESP32>. [Πρόσβαση 24 4 2021].
- [8] Ai-Thinker, «ESP32-CAM,» Ai-Thinker, 2017. [Ηλεκτρονικό]. Available: [https://docs.ai-thinker.com/\\_media/esp32/docs/esp32-cam\\_product\\_specification\\_zh.pdf](https://docs.ai-thinker.com/_media/esp32/docs/esp32-cam_product_specification_zh.pdf). [Πρόσβαση 24 4 2021].
- [9] Wikipedia, «H-bridge,» 20 4 2021. [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/H-bridge>. [Πρόσβαση 25 4 2021].
- [10] Texas Instruments, «L293 Quadruple half-H drivers,» Texas Instruments, 2015. [Ηλεκτρονικό]. Available: <https://www.ti.com/document-viewer/L293/datasheet/abstract#SLRS0086522>. [Πρόσβαση 25 4 2021].
- [11] GROBOTRONICS, «GROBOTRONICS,» [Ηλεκτρονικό]. Available: <https://grobotronics.com/>. [Πρόσβαση 10 3 2021].
- [12] GROBOTRONICS, «DC Gear Motor TT - 130 RPM (With Wire),» [Ηλεκτρονικό]. Available: <https://grobotronics.com/dc-gear-motor-tt-130-rpm-with-wire.html>. [Πρόσβαση 30 4 2021].



- [13] GROBOTRONICS, «Μπαταρία Επαναφορτιζόμενη AA 1.2V 1800mAh,» [Ηλεκτρονικό]. Available: <https://grobotronics.com/rechargeable-battery-aa-1.2v-1800mah.html>. [Πρόσβαση 25 4 2021].
- [14] GROBOTRONICS, «Μπαταρία Li-Po - 3.7v 1800mAh,» Cellevia Power, [Ηλεκτρονικό]. Available: <https://grobotronics.com/polymer-lithium-ion-battery-3.7v-1800mah-el.html>. [Πρόσβαση 25 4 2021].
- [15] GROBOTRONICS, «Τροφοδοτικό 5V 3A - Output 5.5x2.5,» OEM, [Ηλεκτρονικό]. Available: <https://grobotronics.com/power-supply-5v-3a-output-5.5x2.5.html>. [Πρόσβαση 15 3 2021].
- [16] GROBOTRONICS, «Adafruit CP2104 Friend - USB to Serial Converter,» USA, [Ηλεκτρονικό]. Available: <https://grobotronics.com/adafruit-cp2104-friend-usb-to-serial-converter.html>. [Πρόσβαση 10 4 2021].
- [17] FreeCad, «FreeCad,» [Ηλεκτρονικό]. Available: <https://www.freecadweb.org/index.php>. [Πρόσβαση 10 4 2021].
- [18] EasyEDA, «EasyEDA,» [Ηλεκτρονικό]. Available: <https://easyeda.com/>. [Πρόσβαση 1 5 2021].
- [19] WIKIPEDIA, «HTML,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/HTML>. [Πρόσβαση 14 5 2021].
- [20] WIKIPEDIA, «CSS,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/CSS>. [Πρόσβαση 14 5 2021].
- [21] WIKIPEDIA, «JavaScript,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/JavaScript>. [Πρόσβαση 14 5 2021].
- [22] WIKIPEDIA, «Portable Network Graphics,» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Portable\\_Network\\_Graphics](https://en.wikipedia.org/wiki/Portable_Network_Graphics). [Πρόσβαση 14 5 2021].
- [23] M. P. K. P. Avraam Chatzopoulos, «Innovative Robot for Educational Robotics and STEM,» p. 10, 2020.
- [24] W3Schools, «W3Schools,» [Ηλεκτρονικό]. Available: <https://www.w3schools.com/>. [Πρόσβαση 20 4 2021].
- [25] GitHub-bevacqua. [Ηλεκτρονικό]. Available: <https://github.com/bevacqua/dragula>. [Πρόσβαση 20 3 2021].
- [26] WIKIPEDIA, «C++,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/C%2B%2B>. [Πρόσβαση 14 5 2021].

- [27] Arduino, «Arduino IDE,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/software>. [Πρόσβαση 20 3 2021].
- [28] GitHub - espressif, «ESP32 CAMERA,» 2021. [Ηλεκτρονικό]. Available: <https://github.com/espressif/esp32-camera>. [Πρόσβαση 20 3 2021].
- [29] GitHub - me-no-dev, «ESPAsyncWebServer,» 2021. [Ηλεκτρονικό]. Available: <https://github.com/me-no-dev/ESPAsyncWebServer>. [Πρόσβαση 20 3 2021].
- [30] GitHub - me-no-dev, «AsyncTCP,» 2019. [Ηλεκτρονικό]. Available: <https://github.com/me-no-dev/AsyncTCP>. [Πρόσβαση 20 4 2021].
- [31] WIKIPEDIA, «Router,» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Router\\_\(computing\)](https://en.wikipedia.org/wiki/Router_(computing)). [Πρόσβαση 14 5 2021].
- [32] COSMOTE, «Speedport Entry 2i,» OTE, [Ηλεκτρονικό]. Available: [https://help.cosmote.gr/system/templates/selfservice/gnosisgr/Files2/SPEEDPORT\\_ENTRY\\_2i\\_%CE%9F%CE%94%CE%97%CE%93%CE%99%CE%95%CE%A3\\_%CE%95%CE%93%CE%9A%CE%91%CE%A4%CE%91%CE%A3%CE%A4%CE%91%CE%A3%CE%97%CE%A3\\_GR.pdf](https://help.cosmote.gr/system/templates/selfservice/gnosisgr/Files2/SPEEDPORT_ENTRY_2i_%CE%9F%CE%94%CE%97%CE%93%CE%99%CE%95%CE%A3_%CE%95%CE%93%CE%9A%CE%91%CE%A4%CE%91%CE%A3%CE%A4%CE%91%CE%A3%CE%97%CE%A3_GR.pdf). [Πρόσβαση 15 5 2021].
- [33] WIKIPEDIA, «Port Forward,» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Port\\_forwarding](https://en.wikipedia.org/wiki/Port_forwarding). [Πρόσβαση 20 4 2021].
- [34] COSMOTE, «Speedport Entry 2i - Port Forwarding,» OTE, [Ηλεκτρονικό]. Available: [https://www.cosmote.gr/fixed/tech\\_speedport/-/asset\\_publisher/1X9NxR5xYkfu/content/speedport-entry-2i-port-forwarding](https://www.cosmote.gr/fixed/tech_speedport/-/asset_publisher/1X9NxR5xYkfu/content/speedport-entry-2i-port-forwarding). [Πρόσβαση 10 5 2021].
- [35] WIKIPEDIA, «List of TCP and UDP port numbers,» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers). [Πρόσβαση 20 4 2021].
- [36] VMware, «XAMPP,» VMware. [Ηλεκτρονικό]. [Πρόσβαση 23 4 2021].
- [37] WIKIPEDIA, «Unmanned aerial vehicle,» 9 6 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Unmanned\\_aerial\\_vehicle](https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle). [Πρόσβαση 9 6 2021].
- [38] WIKIPEDIA, «COVID-19 pandemic,» 10 6 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/COVID-19\\_pandemic](https://en.wikipedia.org/wiki/COVID-19_pandemic). [Πρόσβαση 12 6 2021].