

Εικόνα 1. Arduino Car Finished Project (Side)

Θέμα Πτυχιακής Εργασίας : Οπτικοποίηση
δεδομένων από συστήματα αυτοματισμού
με δημιουργία ενσωματωμένου
εξυπηρετητή.

Ονοματεπώνυμο Φοιτητή: Κωνσταντίνος Λιλής

Αριθμός Μητρώου : 2473098

Επιβλέπων καθηγητής : Δεληγιαννάκης Νεκτάριος

Περίληψη

Είναι γνωστό πως στην εποχή που ζούμε, οι νέες και ραγδαίες τεχνολογικές ανακαλύψεις μεταβάλλουν δραστικά τον τρόπο που ζει ο άνθρωπος στην σημερινή κοινωνία. Ειδικά, ο τρόπος που μετακινείται είτε στο κοντινό του περιβάλλον είτε σε άγνωστα περιβάλλοντα έχει αλλάξει ριζικά, όμως πολλές φορές οι υπάρχουσες τεχνολογίες δεν αρκούν για τη επίτευξη των στόχων του.

Θέμα, λοιπόν, της εργασίας μου είναι η οπτικοποίηση δεδομένων από συστήματα αυτοματισμού με δημιουργία ενσωματωμένου εξυπηρετητή, το οποίο έχω αναπτύξει σε μεγαλύτερο βαθμό έτσι ώστε να έρχεται στα πλαίσια ενός μη επανδρωμένου οχήματος. Στόχος της διερεύνησής μου είναι η δυνατότητα άντλησης δεδομένων από αισθητήρες ενός συστήματος αυτοματισμού και η δημιουργία μιας εσωτερικά ενσωματωμένης οντότητας εξυπηρετητή. Αυτός καλείται να λύνει το πρόβλημα της οπτικοποίησης δεδομένων από αισθητήρες που είναι τοποθετημένοι σε κάποιο σύστημα αυτοματισμού όπως το αυτοκινούμενο όχημα που έχω επιλέξει.

Η υλοποίηση έχει γίνει μέσω της χρήσης πλακέτας Arduino όπου δημιουργείται ένας ενσωματωμένος δικτυακός εξυπηρετητής (embedded web server) και δύναται κανείς να επικοινωνήσει με αυτόν μέσω Wi-Fi με την χρήση WebSocket από μια HTML γραφική διεπαφή με το χρήστη.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Άντληση δεδομένων αισθητήρων, Παρακολούθηση Δικτύων, Εγκαθίδρυση επικοινωνιών από συσκευές σε άκρα δικτύων (edge devices), οπτικοποίηση δεδομένων αισθητήρων, μηχανισμοί λήψης αποφάσεων σε δεδομένα αισθητήρων, ενσωματωμένα συστήματα.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αρντουϊνο, δεδομένα, αισθητήρες, δίκτυο, οπτικοποίηση, παρακολούθηση, άντληση, εξυπηρετητής, χρήστης, ενσωματωμένος, αυτοματισμός, μη επανδρωμένο, κινητήρες, τρισδιάστατη εκτύπωση, χαρτικοποίηση.

Abstract

It is well known that, in the era we live in, the new and rapid technological discoveries are drastically changing the way man lives today. Specifically, the way he moves around, whether in his immediate environment or in unknown environments, has changed radically, but many times the existing technologies are not enough to achieve his goals.

The subject of my work, then, is the visualization of data from automation systems by creating an embedded server, which I have developed to a greater extent, so that it fits within the framework of an unmanned vehicle. The goal of my research is the ability to extract data from sensors in an automation system and the creation of an internally integrated server entity. This entity is called upon to solve the problem of visualizing data from sensors that are placed in an automation system such as the self-driving vehicle that I have chosen.

The implementation has been done using an Arduino board where an embedded web server is created and one can communicate with it via Wi-Fi, using WebSocket Protocol from an HTML graphical user interface.

SUBJECT AREA: Sensor data mining, Network Monitoring, Establishing communications from devices to network edges (edge devices), sensor data visualization, decision-making mechanisms on sensor data, embedded systems.

KEYWORDS: Arduino, data, sensors, network, visualization, monitoring, pumping, hypervisor, user, embedded, automation, unmanned, motors, 3D printing, papermaking.

Ευχαριστίες

Για τη συγκεκριμένη εργασία θα ήθελα να ευχαριστήσω πρώτα από όλα τον επιβλέποντα της καθηγητή κ. Δεληγιαννάκη Νεκτάριο και φυσικά και τον υπεύθυνο του τμήματος Computer Science κ. Ιωάννη Μανωλόπουλο. Θερμές ευχαριστίες σε όλους τους προπτυχιακούς φοιτητές και καθηγητές που ήρθα σε επαφή για γνώμες, απόψεις και υποδείξεις που συνέβαλαν σημαντικά στην ορθή τεκμηρίωση της εργασίας αυτής.

Τέλος θα θέλω να ευχαριστήσω όλους εκείνους που είχα ως πρότυπά μου στην ακαδημαϊκή κοινότητα του κολλεγίου, που ενέπνεαν το σεβασμό αλλά και το πρότυπό τους μας έδινε ώθηση για δουλειά.

Θερμότερες ευχαριστίες.

Πίνακας περιεχομένων

Πίνακας περιεχομένων	5
Πίνακας Εικόνων	8
Πίνακας Αρκτικόλεξων – Ακρωνυμίων – Συντομογραφιών	10
1 Πρόλογος	13
2 Εισαγωγή	16
3 Θεωρητικό υπόβαθρο	17
3.1 Arduino	17
3.1.1 Γενικά	17
3.1.2 Ιστορία	18
3.1.3 Εσωτερικές διαμάχες	19
3.1.4 Υλικό - Hardware	20
3.1.5 Χρήση του Arduino στην εργασία	22
3.1.6 Arduino UNO R4 Wi-Fi	23
3.1.7 Χρήση βιβλιοθηκών	24
3.2 Ενσωματωμένος διαδικτυακός εξυπηρετητής	27
3.2.1 Γενικά	27
3.2.2 Χρήσεις	28
3.2.3 Χρήση WebSockets	28
3.3 Γραφική Διεπαφή HTML	29
3.3.1 Γενικά	29
3.3.2 Χρήσεις σήμερα	30
4 Προσέγγιση εργασίας	31
4.1 Διερεύνηση σκοπού	31
4.2 Software	32
4.3 Hardware	32
5 Ανάπτυξη έργου	34
5.1 Γενικά	34
5.2 Ξεκίνημα – Version 0.8	35
5.2.1 Γενικά	35
5.2.2 Setup()	35

5.2.3	Loop()	35
5.2.4	Ανασκόπηση	36
5.3	Ενσωμάτωση WebSocket - Version 1.3	36
5.3.1	Γενικά	36
5.3.2	Setup()	36
5.3.3	Loop()	36
5.3.4	HMTL Version 1.0	37
5.3.5	Ανασκόπηση	37
5.4	Ενσωμάτωση Αισθητήρων - Version 2.6.....	38
5.4.1	Γενικά	38
5.4.2	Αισθητήρες στην κίνηση.....	39
5.4.3	Λοιπές αλλαγές	39
5.4.4	Setup()	39
5.4.5	Loop()	40
5.4.6	Ανασκόπηση	40
5.5	Ολοκλήρωση σημαντικών αναβαθμίσεων – Αμφίδρομη επικοινωνία – Version 3.8.5.....	40
5.5.1	Γενικά	40
5.5.2	Αμφίδρομη επικοινωνία - web_me3	41
5.5.3	Servo motor συνδέσεις	42
5.5.4	Arduino UNO R4 Wi-Fi LED Matrix.....	42
5.5.5	Διαχείριση LED μέσω Shift Register	43
5.5.6	Ανανέωση συνάρτηση μετακίνησης οχήματος.....	45
5.5.7	Setup()	45
5.5.8	Loop()	46
5.5.9	HTML Version 2.0	46
5.6	Τελική έκδοση – Version 4.5.6.....	48
5.6.1	Γενικά	48
5.6.2	Προσθήκη δεύτερου Shift Register – Dedicated Library.....	49
5.6.3	Αλλαγές σε αισθητήρες	49
5.6.4	Step Counters.....	50
5.6.5	Λειτουργία μεταβλητής ταχύτητας κινητήρων	51

5.6.6	Αυτόματη λειτουργία	52
5.6.7	Αναβάθμιση στη λειτουργία κίνησης.....	53
5.6.8	Setup()	53
5.6.9	Loop()	54
5.6.10	HTML Version 3.0	54
6	Αποτελέσματα – Τελικές Σκέψεις.....	57
7	Περιθώριο για μελλοντικές αναβαθμίσεις.....	58
7.1	New Software	58
7.2	New Hardware	59
8	Βιβλιογραφία	60

Πίνακας Εικόνων

Εικόνα 1. Arduino Car Finished Project (Side)	1
Εικόνα 2. Πρώιμα μέσα μεταφορών	13
Εικόνα 3. Μοντέρνα μέσα μεταφορών.....	14
Εικόνα 4. Ενσωματωμένος διαδικτυακός εξυπηρετητής	16
Εικόνα 5. Arduino IDE	18
Εικόνα 6. Arduino evolution	19
Εικόνα 7. Arduino Hardware History	21
Εικόνα 8. Arduino Uno R4 Wi-Fi	23
Εικόνα 9. Arduino Libraries	24
Εικόνα 10. Web Server	27
Εικόνα 11. Embedded Web Server	27
Εικόνα 12. WebSocket Protocol.....	28
Εικόνα 13. HTML	29
Εικόνα 14 . Παράδειγμα κώδικα HTML.....	31
Εικόνα 15. Παράδειγμα κώδικα Arduino	32
Εικόνα 16. Πρώιμο στάδιο κατασκευής.....	34
Εικόνα 17. Πρώιμο HTML.....	35
Εικόνα 18. HTML Version 1.0	37
Εικόνα 19. Infrared Sensor connections	38
Εικόνα 20. Ultrasonic Sensor connections	38
Εικόνα 21. HTML Version 1.5	41
Εικόνα 22. Σύνδεση Servo Motor	42
Εικόνα 23. Arduino Uno R4 Led Matrix.....	43
Εικόνα 24. Shift Register 74HC595 Circuit.....	43
Εικόνα 25. 74HC959 Pinout	44
Εικόνα 26. 74HC595 Circuit	45
Εικόνα 27. HTML Version 2.0	46
Εικόνα 28. HTML Assisted Move initialized	47
Εικόνα 29. HTML Connected	47
Εικόνα 30. Lights On	48
Εικόνα 31. Auto Move Initialized(WIP)	48
Εικόνα 32. Two Shift Register Connection	49
Εικόνα 33. New Ultrasonic Setup.....	50
Εικόνα 34. Infrared Step Counter Sensors	50
Εικόνα 35. PWM Connection	52
Εικόνα 36. HTML Version 3.0	54
Εικόνα 37. Version 3.0 Initial Connection	55
Εικόνα 38. New Functions On HTML	55
Εικόνα 39. Map In Action.....	56



Εικόνα 40. Arduino Car Finished Project (Vehicle Plan).....57

Πίνακας Αρκτικόλεξων – Ακρωνυμίων – Συντομογραφιών

Ορολογίες	Σημασία
3D-Printed Parts	Εξαρτήματα φτιαγμένα με τρισδιάστατη εκτύπωση.
Adobe Flash Player	Λογισμικό για προβολή πολυμέσων σε παλιές ιστοσελίδες.
API	Διεπαφή για επικοινωνία μεταξύ λογισμικών.
Arduino IoT Cloud	Cloud πλατφόρμα της Arduino για IoT εφαρμογές.
Arduino shield	Πρόσθετη πλακέτα επέκτασης για Arduino.
Arduino πλακέτα	Κεντρική πλακέτα προγραμματισμού και αυτοματισμού.
arduino.cc	Επίσημος ιστότοπος της κοινότητας Arduino.
bit	Βασική μονάδα ψηφιακής πληροφορίας (0 ή 1).
BME280	Αισθητήρας θερμοκρασίας, πίεσης και υγρασίας.
boolean	Λογικός τύπος δεδομένων (αληθές ή ψευδές).
bootloader	Πρόγραμμα που εκκινεί μικροελεγκτή ή σύστημα.
bps	Bits ανά δευτερόλεπτο – μονάδα ταχύτητας μετάδοσης.
breadboard	Διάτρητη πλακέτα για πειραματικές συνδέσεις κυκλωμάτων.
buffer	Προσωρινή μνήμη αποθήκευσης δεδομένων.
bytes	Οκτώ bits – βασική μονάδα αποθήκευσης.
Cascading Style Sheets	Γλώσσα μορφοποίησης ιστοσελίδων (CSS).
Comet	Τεχνική για συνεχή ροή δεδομένων μέσω HTTP.
console logs	Μηνύματα εξόδου στον προγραμματιστικό κονσόλα.
CS-11M	Συγκεκριμένο μοντέλο αισθητήρα ή εξαρτήματος (ανάλογα χρήση).
DOM	Αντιπροσωπεία δομής HTML ως αντικείμενα.
double	Τύπος δεκαδικού αριθμού διπλής ακρίβειας.
embedded web Server	Ενσωματωμένος εξυπηρετητής σε συσκευή ή μικροελεγκτή.
ESP32-S3	Μικροελεγκτής με Wi-Fi και Bluetooth δυνατότητες.
flags in programming	Μεταβλητές για ένδειξη καταστάσεων ή επιλογών.
flip-flops	Ψηφιακό κύκλωμα αποθήκευσης bit.
float	Τύπος δεκαδικού αριθμού κινητής υποδιαστολής.
global variables	Μεταβλητές που είναι προσβάσιμες παντού στον κώδικα.
header της σελίδας	Το πάνω μέρος μιας ιστοσελίδας – συχνά με τίτλο/μενού.
HTML	Γλώσσα δομής ιστοσελίδων.
HTML ID	Μοναδικό αναγνωριστικό για HTML στοιχείο.
HTML Style	Στιλ εμφάνισης HTML στοιχείων.
HTTP	Πρωτόκολλο μεταφοράς ιστοσελίδων.
HTTP polling	Συχνή αποστολή αιτήσεων HTTP για έλεγχο αλλαγών.
HTTPS	Ασφαλής έκδοση του HTTP με κρυπτογράφηση.
Hz	Μονάδα συχνότητας (κύκλοι ανά δευτερόλεπτο).
IETF	Οργανισμός για τυποποίηση διαδικτυακών πρωτοκόλλων.
IndexedDB HTML	Βάση δεδομένων εντός του browser.
integer	Ακέραιος αριθμός.
IoT	Διαδίκτυο των Πραγμάτων – συνδεδεμένες συσκευές.
IoT Cloud	Πλατφόρμα αποθήκευσης/ανάλυσης IoT δεδομένων.

IP Address	Μοναδική διεύθυνση συσκευής στο δίκτυο.
ISP	Πάροχος υπηρεσιών διαδικτύου.
JavaScript	Γλώσσα προγραμματισμού για διαδραστικές ιστοσελίδες.
KiCad	Λογισμικό σχεδίασης κυκλωμάτων PCB.
LED	Δίοδος εκπομπής φωτός.
localStorage HTML	Μόνιμη αποθήκευση δεδομένων στον browser.
man-of-war	Πολεμικό ιστοφόρο πλοίο.
multithreading/νήματα	Ταυτόχρονη εκτέλεση πολλών εργασιών στον κώδικα.
PCB	Τυπωμένο κύκλωμα.
Processing IDE	Περιβάλλον για δημιουργικά διαδραστικά γραφικά.
rack υπολογιστών υψηλής ταχύτητας	Ειδική διάταξη για ισχυρούς server ή υπολογιστές.
Renesas RA4M1	Μικροελεγκτής της σειράς RA από τη Renesas.
responsive HTML	Ιστοσελίδα που προσαρμόζεται σε διαφορετικές οθόνες.
reverse flood fill	Αλγόριθμος χρωματισμού περιοχής σε εικόνες.
RFC6455	Προδιαγραφή για το WebSocket πρωτόκολλο.
Serial Monitor του Arduino IDE	Οθόνη επικοινωνίας σειριακής θύρας στον Arduino IDE.
slider bar HTML	Μπάρα επιλογής τιμών σε HTML.
SRAM	Τύπος γρήγορης αλλά πτητικής μνήμης.
string	Ακολουθία χαρακτήρων (κείμενο).
switch – case c++	Δομή ελέγχου πολλών επιλογών στην C++.
toggle buttons	Κουμπιά εναλλαγής κατάστασης.
Transmission Control Protocol	Αξιόπιστο πρωτόκολλο μεταφοράς δεδομένων (TCP).
us sound units	Μονάδες μέτρησης ήχου σε μικροδευτερόλεπτα.
vector	Δυναμικός πίνακας στην C++ (προγραμματισμός).
Visual Studio Code	Εκδότης κώδικα με υποστήριξη πολλών γλωσσών.
XML	Γλώσσα αναπαράστασης δομημένων δεδομένων.
αερόπλοια	Πτητικά μέσα που αιωρούνται με αέριο.
ακίδες μονής ή διπλής σειράς arduino	Επαφές για συνδέσεις εξαρτημάτων σε Arduino.
ανεμόπτερα	Ανεμοκίνητα αεροσκάφη χωρίς κινητήρα.
αντιστάσεις	Στοιχεία που περιορίζουν το ρεύμα σε κύκλωμα.
Ανώτερη Παλαιολιθική	Τελευταία φάση της Παλαιολιθικής εποχής.
αρματωσιά	Σύστημα σχοινιών και ιστίων πλοίου.
αρχεία manifest cache	Αρχεία για offline λειτουργία ιστοσελίδας.
αρχείο zip	Συμπίεσμένο αρχείο πολλαπλών δεδομένων.
ατμομηχανών υψηλής πίεσης	Κινητήρες ατμού με αυξημένη απόδοση.
ατμόπλοια	Πλοία που κινούνται με ατμό.
Βιομηχανικής Επανάστασης	Περίοδος μηχανοποίησης και τεχνολογικής ανάπτυξης.
βομβαρδιστικό μεγάλου βεληνεκούς	Αεροσκάφος για μακρινές αποστολές βομβαρδισμού.
γαλέρες	Πλοία με κουπιά και ιστία, κυρίως πολεμικά.
γείωση	Σύνδεση με τη γη για ασφάλεια στο κύκλωμα.
γλώσσες προγραμματισμού	Τρόποι γραφής εντολών προς υπολογιστές.
γλώσσες προγραμματισμού C και C++	Παραδοσιακές γλώσσες συστημάτων και ενσωματωμένων.
γραμμές εντολών	Διασύνδεση εισαγωγής εντολών με πληκτρολόγιο.

γραφική διεπαφή	Οπτική επικοινωνία χρήστη με λογισμικό.
διακομιστής ιστού	Server που φιλοξενεί ιστοσελίδες.
διεπαφή διαμόρφωσης	Περιβάλλον αλλαγής ρυθμίσεων συσκευής.
δρομολογητής	Συσκευή που κατευθύνει την κυκλοφορία στο δίκτυο.
εισόδου/εξόδου (I/O)	Επικοινωνία συσκευής με περιφερειακά.
ετικέτα HTML	Στοιχείο δομής σε HTML.
θηλυκές κεφαλίδες Arduino pin	Υποδοχές για σύνδεση εξαρτημάτων στην Arduino.
ιβηρικό караβελάκι	Μικρό πλοίο ισπανικής προέλευσης, εξερευνητικό.
ιστιοφόρα	Πλοία που κινούνται με ιστία και άνεμο.
ιστιοφόρα διπλού κύτους	Πλοία με δύο γάστρες για σταθερότητα.
ιστότοπος διαδικτύου υψηλής επισκεψιμότητας	Σελίδα με μεγάλο αριθμό επισκεπτών.
Λίθινη Εποχή	Προϊστορική εποχή χρήσης λίθινων εργαλείων.
μη επανδρωμένων οχημάτων	Ρομποτικά οχήματα χωρίς χειριστή.
μηχανοκίνητα αεροσκάφη	Αεροπλάνα με κινητήρα.
μικροελεγκτές	Μικροί επεξεργαστές για ελεγχόμενες λειτουργίες.
μικροελεγκτής BASIC Stamp	Επεξεργαστής για απλά ενσωματωμένα έργα.
μνήμης flash	Επαναπρογραμματιζόμενη, μη πτητική μνήμη.
μονάδες breakout	Πλακέτες με εύκολη πρόσβαση σε εξαρτήματα.
Νεολιθική	Εποχή γεωργίας και μόνιμων οικισμών.
οπτικοποίησης δεδομένων	Παρουσίαση δεδομένων σε γραφική μορφή.
παλμικό σήμα	Σήμα με μεταβαλλόμενες παλμικές τάσεις.
παλμογραφώ	Καταγραφή ή μέτρηση ηλεκτρικών παλμών.
προγραμματισμός AVR	Κώδικας για AVR μικροελεγκτές της Atmel.
πρόγραμμα ανίχνευσης ιστού	Εργαλείο που εξερευνά ιστοσελίδες (web crawler).
πρόγραμμα περιήγησης ιστού	Εφαρμογή πλοήγησης σε ιστοσελίδες (browser).
σειριακού διαύλου	Τύπος επικοινωνίας δεδομένων με σειρά bit.
τάσης	Ηλεκτρική διαφορά δυναμικού (Volt).
ταχύτερη ταχύτητα ρολογιού	Υψηλότερη συχνότητα λειτουργίας επεξεργαστή.
το κινεζικό πλοίο θησαυρού	Μεγάλο κινεζικό πλοίο του 15ου αιώνα.
υδρογόνο	Ελαφρύτερο στοιχείο – χρησιμοποιείται σε αερόπλοια.
υπέρυθρο φως	Ηλεκτρομαγνητική ακτινοβολία πέραν του ορατού.
ψηφιακών και αναλογικών ακίδων arduino	Επαφές Arduino για είσοδο/έξοδο σημάτων.

1 Πρόλογος

Από την αρχή της ιστορίας του, ο άνθρωπος ήταν ένα ον που μετακινούνταν συνεχώς. Ιστορικά, η πρώιμη ανθρώπινη μετανάστευση περιλαμβάνει τους ανθρώπους του κόσμου, δηλαδή τη μετανάστευση σε περιοχές του κόσμου όπου προηγουμένως δεν υπήρχε ανθρώπινη κατοίκηση, κατά την Ανώτερη Παλαιολιθική*. Από τη Νεολιθική*, οι περισσότερες μεταναστεύσεις ήταν κατά κύριο λόγο πολεμικές, αποτελούμενες από κατάκτηση ή από την πλευρά των επεκτεινόμενων πληθυσμών. [1]

Η ιστορία των μεταφορών είναι σε μεγάλο βαθμό ιστορία τεχνολογικής καινοτομίας. Η πρόοδος της τεχνολογίας έχει επιτρέψει στους ανθρώπους να ταξιδεύουν μακρύτερα, να εξερευνούν περισσότερες περιοχές και να επεκτείνουν την επιρροή τους σε όλο και μεγαλύτερες περιοχές. Ακόμη και στην



Εικόνα 2. Πρώιμα μέσα μεταφορών

αρχαιότητα, νέα εργαλεία όπως καλύμματα ποδιών, σκι και χιονοπέδιλα επιμήκυναν τις αποστάσεις που μπορούσαν να διανυθούν. Καθώς οι νέες εφευρέσεις και ανακαλύψεις εφαρμόστηκαν σε προβλήματα μεταφοράς, ο χρόνος ταξιδιού μειώθηκε ενώ η ικανότητα μετακίνησης περισσότερων και μεγαλύτερων φορτίων αυξήθηκε. [2]

Στη στεριά τα πρώτα ίχνη δημιουργήθηκαν από ανθρώπους που μετέφεραν αγαθά και συχνά ακολουθούσαν μονοπάτια. Οι διαδρομές δημιουργούνταν φυσικά σε σημεία υψηλής πυκνότητας κυκλοφορίας και καθώς τα ζώα εξημερώθηκαν, τα άλογα, τα βόδια και τα γαϊδούρια έγιναν ένα στοιχείο στη δημιουργία τροχοφόρων οχημάτων που έλκονταν από ζώα. Κατά τη διάρκεια της Βιομηχανικής Επανάστασης* σχεδιάστηκαν οι πρώτοι σύγχρονοι αυτοκινητόδρομοι [3] ενώ παράλληλα με ανάπτυξη των ατμομηχανών υψηλής πίεσης* ευημέρησαν οι σιδηρόδρομοι και τα τρένα. Οι σιδηρόδρομοι όχι μόνο αύξησαν την ταχύτητα των μεταφορών, αλλά μείωσαν δραματικά και το κόστος τους. Για παράδειγμα, ο πρώτος διηπειρωτικός σιδηρόδρομος στις Ηνωμένες Πολιτείες είχε ως αποτέλεσμα οι επιβάτες και τα εμπορεύματα να μπορούν να διασχίσουν τη χώρα

σε λίγες μέρες αντί για μήνες και στο ένα δέκατο του κόστους μεταφοράς με αμαξίδιο ή βαγόνι. [4]

Στις θαλάσσιες μεταφορές ξεκινάμε από την Λίθινη Εποχή*, που τα πρωτόγονα σκάφη αναπτύχθηκαν για να επιτρέπουν τη ναυσιπλοΐα στα ποτάμια και το ψάρεμα στα ποτάμια και στις ακτές. Έχει υποστηριχθεί ότι τα σκάφη κατάλληλα για μια σημαντική θαλάσσια διέλευση ήταν απαραίτητα για να φτάσουν οι άνθρωποι στην Αυστραλία πριν από περίπου 40.000-45.000 χρόνια. Με την ανάπτυξη του πολιτισμού, τα σκάφη εξελίχθηκαν για επέκταση και γενικά μεγάλωσαν σε μέγεθος για εμπόριο και πόλεμο. Στη Μεσόγειο, οι γαλέρες* αναπτύχθηκαν περίπου το 3000 π.Χ. ενώ πολυνησιακά ιστιοφόρα διπλού κύτους* με προηγμένη αρματωσιά* χρησιμοποιήθηκαν μεταξύ 1.300 π.Χ. και 900 π.Χ. από τους πολυνησιακούς απογόνους. Οι γαλέρες κατέστησαν τελικά απαρχαιωμένες από ιστιοφόρα* ωκεανού, όπως το ιβηρικό καραβελάκι* τον 14ο αιώνα, το κινεζικό πλοίο θησαυρού* στις αρχές του 15ου αιώνα και το μεσογειακό man-of-war* στα τέλη του 15ου αιώνα. Στη Βιομηχανική Επανάσταση αναπτύχθηκαν τα πρώτα ατμόπλοια*, εξειδικευμένα σκάφη για ποτάμια και κανάλια μεταφοράς και αργότερα πλοία που κινούνταν με ντίζελ. Τελικά τα υποβρύχια αναπτύχθηκαν κυρίως για στρατιωτικούς σκοπούς για το γενικό όφελος των ανθρώπων. [5]

Επίσης, η επιθυμία της ανθρωπότητας να πετάξει πιθανότατα χρονολογείται από την πρώτη φορά που ο άνθρωπος παρατήρησε πουλιά. Μεγάλο μέρος της εστίασης της πρώιμης έρευνας ήταν στη μίμηση πτηνών, αλλά μέσω δοκιμής και λάθους, εφευρέθηκαν μπαλόνια, αερόπλοια*, ανεμόπτερα* και τελικά



Εικόνα 3. Μοντέρνα μέσα μεταφορών

μηχανοκίνητα αεροσκάφη* και άλλοι τύποι ιπτάμενων μηχανών. Οι χαρταετοί ήταν η πρώτη μορφή ιπτάμενων αντικειμένων που κατασκευάστηκαν από τον άνθρωπο, ενώ το 17ο και 18ο αιώνα, οι επιστήμονες άρχισαν να αναλύουν την ατμόσφαιρα της Γης και ανακάλυψαν αέρια όπως το υδρογόνο*, που οδήγησαν στην εφεύρεση

των μπαλονιών υδρογόνου. Ο Δεύτερος Παγκόσμιος Πόλεμος σημείωσε μια δραστική αύξηση στον ρυθμό ανάπτυξης και παραγωγής αεροσκαφών. Όλες οι χώρες που συμμετείχαν στον πόλεμο ενίσχυσαν την ανάπτυξη και την παραγωγή αεροσκαφών και συστημάτων παράδοσης όπλων που βασίζονται σε πτήση, όπως

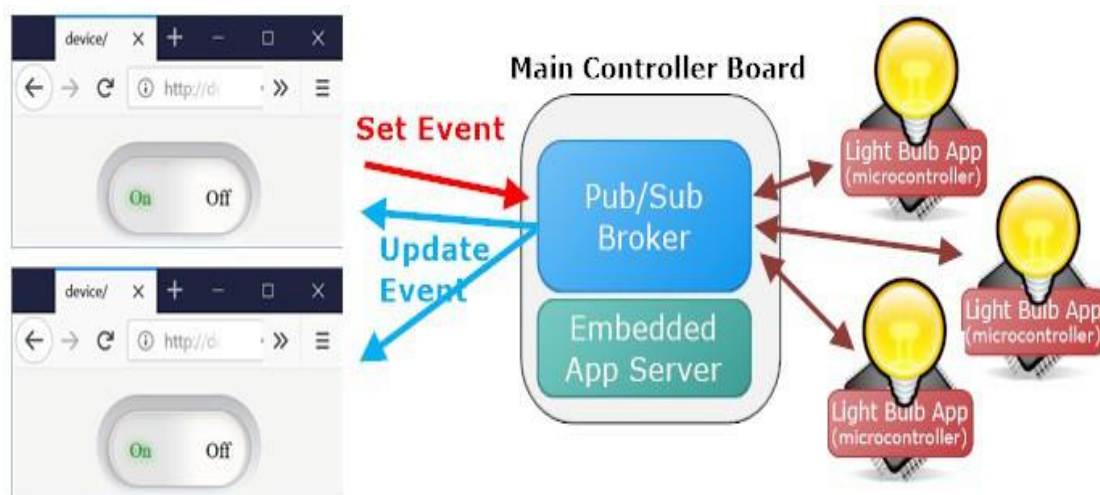
το πρώτο βομβαρδιστικό μεγάλου βεληνεκούς*. Μετά το τέλος του πολέμου, η εμπορική αεροπορία αναπτύχθηκε γρήγορα, χρησιμοποιώντας κυρίως πρώην στρατιωτικά αεροσκάφη για τη μεταφορά ανθρώπων και φορτίων. [6]

Οι σύγχρονες μεταφορές αναφέρονται στα προηγμένα συστήματα και τεχνολογίες που χρησιμοποιούνται για την αποτελεσματική μετακίνηση ανθρώπων και αγαθών σε διάφορες αποστάσεις. Περιλαμβάνει πολλούς βασικούς τρόπους, συμπεριλαμβανομένων των οδικών μεταφορών με αυτοκίνητα, λεωφορεία και φορτηγά. Οι σιδηροδρομικές μεταφορές, όπως τα τρένα υψηλής ταχύτητας και τα μετρό, προσφέρουν γρήγορα, αξιόπιστα και φιλικά προς το περιβάλλον ταξίδια, ειδικά σε αστικές περιοχές, ενώ οι αεροπορικές μεταφορές, μέσω αεροπλάνων και μη επανδρωμένων αεροσκαφών, επιτρέπουν τα γρήγορα ταξίδια και παράδοση μεγάλων αποστάσεων. Οι θαλάσσιες μεταφορές παραμένουν ζωτικής σημασίας για το παγκόσμιο εμπόριο, καθώς καινοτομίες όπως τα αυτοματοποιημένα πλοία και οι καθαρότερες πηγές ενέργειας βελτιώνουν την απόδοση. Επιπλέον, οι διαστημικές μεταφορές, αν και είναι ακόμα σε πρώιμο στάδιο και δεν μπορούν να προσφέρουν ουσιαστική μετακίνηση μαζικού πληθυσμού ή/και εμπορίου, αναπτύσσονται με ραγδαίους ρυθμούς. [7]

2 Εισαγωγή

Καταλαβαίνουμε, λοιπόν, ότι χρειάστηκαν πολλά χρόνια, πολύς κόπος και πόροι και ακόμα περισσότερη μελέτη από τον ίδιο τον άνθρωπο για να φτάσει σε αυτό το σημείο που είναι σήμερα. Όμως παρόλο αυτή την ανάπτυξη, ορισμένα μέρη του κόσμου μας παραμένουν δύσκολα ως προς την πρόσβαση και την εξερεύνηση τους. Λύση σε αυτό το πρόβλημα φέρνει η μοντέρνα τεχνολογία, με την ανάπτυξη **μη επανδρωμένων οχημάτων***. Σε συνδυασμό με την **αυτοματοποίηση** τους γίνεται δυνατή η πρόσβαση σε τόπους που δεν είναι εύκολο να μετακινηθεί ο ίδιος ο άνθρωπος, τόσο στην επιφάνεια της Γης, στο βυθό της θάλασσας, στον αέρα όσο και σε άλλους πλανήτες μακριά από εμάς.

Στην παρούσα εργασία το βασικό θέμα είναι η διερεύνηση της δυνατότητας **άντλησης δεδομένων από αισθητήρες ενός συστήματος αυτοματισμού** και δημιουργίας μιας εσωτερικά **ενσωματωμένης οντότητας εξυπηρετητή**. Στα πλαίσια αυτής της διατριβής, έχω υλοποιήσει έναν ενσωματωμένο εξυπηρετητή μικρής κλίμακας, δηλαδή ένα ενσωματωμένο δικτυακό εξυπηρετητή (**embedded web server***) ο οποίος «**τρέχει**» μέσα στο σύστημα και λύνει το πρόβλημα της οπτικοποίησης δεδομένων* από αισθητήρες που είναι τοποθετημένοι σε ένα αυτοκινούμενο όχημα. Η επικοινωνία με το χρήστη γίνεται μέσω **Wi-Fi** και με την χρήση **WebSockets**. Πέρα από τις βασικές λειτουργίες ενός ενσωματωμένου εξυπηρετητή, έχω αναπτύξει έξυπνα συστήματα για την **υποβοήθηση της κίνησης** του αμαξιδίου όταν το χειρίζεται ο χρήστης και επίσης όρισα ένα **αυτόματο σύστημα χαρτογράφησης και αναπαράστασης χώρου**.



Εικόνα 4. Ενσωματωμένος διαδικτυακός εξυπηρετητής

3 Θεωρητικό υπόβαθρο

Για την ανάπτυξη της εργασίας χρησιμοποίησα, όσο αφορά τον προγραμματισμό της πλακέτας και τον κώδικα του γραφικού περιβάλλοντος:

- Το **Arduino IDE**, ένα λογισμικό ανοιχτού κώδικα, το οποίο χρησιμοποιείται για τον προγραμματισμό των πλακετών Arduino*.
- Διάφορες βιβλιοθήκες του Arduino IDE, όπως η βιβλιοθήκη **WebSocket** για την επικοινωνία **server-client**.
- Ενσωματωμένο διαδικτυακό εξυπηρετητή για back end της **επικοινωνίας server-client**.
- **HTML κώδικα** που αναπτύχθηκε σε περιβάλλον **Visual Studio Code*** και χρησιμοποιείται για την γραφική διεπαφή* μεταξύ χρήστη και εξυπηρετητή.

3.1 Arduino

3.1.1 Γενικά

Το **Arduino** [8] είναι μια ιταλική εταιρεία υλικού και λογισμικού ανοιχτού κώδικα, έργο και κοινότητα χρηστών που σχεδιάζει και κατασκευάζει μικροελεγκτές* και κιτ μικροελεγκτών μιας πλακέτας για την κατασκευή ψηφιακών συσκευών. Τα προϊόντα υλικού της διαθέτουν άδεια χρήσης επιτρέποντας την κατασκευή πλακετών Arduino και τη διανομή λογισμικού από οποιονδήποτε. Οι πλακέτες Arduino διατίθενται στο εμπόριο από τον επίσημο ιστότοπο ή μέσω εξουσιοδοτημένων διανομέων. Τα σχέδια πλακέτας Arduino χρησιμοποιούν μια ποικιλία μικροεπεξεργαστών και ελεγκτών. Οι πλακέτες είναι εξοπλισμένες με σετ **ψηφιακών και αναλογικών ακίδων* εισόδου/εξόδου (I/O)*** που μπορούν να συνδεθούν με διάφορες πλακέτες επέκτασης («**ασπίδες***») ή **breadboards*** (για την κατασκευή πρωτοτύπων) και άλλα κυκλώματα. Οι πλακέτες διαθέτουν διασυνδέσεις σειριακής επικοινωνίας, συμπεριλαμβανομένου του Universal Serial Bus (**USB**) σε ορισμένα μοντέλα, οι οποίες χρησιμοποιούνται επίσης για τη φόρτωση προγραμμάτων.

Οι μικροελεγκτές μπορούν να προγραμματιστούν χρησιμοποιώντας τις **γλώσσες προγραμματισμού* C και C++ *** (Embedded C), χρησιμοποιώντας ένα τυπικό **API*** που είναι επίσης γνωστό ως Arduino Programming Language, εμπνευσμένο από τη γλώσσα επεξεργασίας και χρησιμοποιείται με μια τροποποιημένη έκδοση του **Processing IDE***. Εκτός από τη



Εικόνα 5. Arduino IDE

χρήση παραδοσιακών αλυσίδων εργαλείων μεταγλωττιστή, το έργο Arduino παρέχει ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) και ένα εργαλείο γραμμής εντολών*. Το έργο Arduino ξεκίνησε το 2005 ως εργαλείο για φοιτητές στο Interaction Design Institute Ivrea, Ιταλία, με στόχο να παρέχει έναν χαμηλού κόστους και εύκολο τρόπο για αρχάριους και επαγγελματίες να δημιουργήσουν συσκευές που αλληλεπιδρούν με το περιβάλλον τους χρησιμοποιώντας αισθητήρες και ενεργοποιητές. Συνήθη παραδείγματα τέτοιων συσκευών που προορίζονται για αρχάριους χομπίστες περιλαμβάνουν απλά ρομπότ, θερμοστάτες και ανιχνευτές κίνησης. [9]

3.1.2 Ιστορία

Το όνομα Arduino προέρχεται από ένα καφέ στην Ivrea της Ιταλίας, όπου συναντιόντουσαν ορισμένοι από τους ιδρυτές του έργου. Το μπαρ πήρε το όνομά του από τον Arduin of Ivrea, ο οποίος ήταν ο μαργάφος της Πορείας της Ivrea και βασιλιάς της Ιταλίας από το 1002 έως το 1014. Όταν ξεκίνησε το Arduino οι μαθητές χρησιμοποίησαν έναν μικροελεγκτή BASIC Stamp* με κόστος 50 \$. Το 2004, ο Hernando Barragán δημιούργησε την πλατφόρμα ανάπτυξης Wiring ως μεταπτυχιακό έργο στο IDII, υπό την επίβλεψη των Massimo Banzì και Casey Reas. Ο Casey Reas είναι γνωστός για τη συνδημιουργία, με τον Ben Fry, την πλατφόρμα ανάπτυξης Processing. Ο στόχος του έργου ήταν να δημιουργηθούν απλά, χαμηλού κόστους εργαλεία για τη δημιουργία ψηφιακών έργων από μη μηχανικούς. Η πλατφόρμα καλωδίωσης αποτελούνταν από μια πλακέτα τυπωμένου κυκλώματος (**PCB**)* με έναν μικροελεγκτή ATmega128, ένα IDE που βασίζεται σε λειτουργίες επεξεργασίας και βιβλιοθήκης για τον εύκολο προγραμματισμό του μικροελεγκτή. Το 2005, ο Massimo Banzì, με τον David Mellis, έναν άλλο φοιτητή IDII, και τον David Cuartielles, επέκτεινε το Wiring προσθέτοντας υποστήριξη για τον φθηνότερο μικροελεγκτή ATmega8. Το νέο

έργο, διχαλωτό από το Wiring, ονομάστηκε Arduino. Η αρχική βασική ομάδα του Arduino αποτελούνταν από τους Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino και David Mellis. Μετά την ολοκλήρωση της πλατφόρμας, διανεμήθηκαν ελαφρύτερες και λιγότερο ακριβές εκδόσεις στην κοινότητα ανοιχτού κώδικα. Υπολογίστηκε στα



Εικόνα 6. Arduino evolution

μέσα του 2011 ότι πάνω από 300.000 επίσημα Arduinos είχαν παραχθεί εμπορικά, και το 2013 ότι 700.000 επίσημες πλακέτες ήταν στα χέρια των χρηστών. [10]

3.1.3 Εσωτερικές διαμάχες

Στις αρχές του 2008, οι πέντε συνιδρυτές του έργου Arduino δημιούργησαν μια εταιρεία, την Arduino LLC, για να κατέχει τα εμπορικά σήματα που σχετίζονται με την Arduino. Η κατασκευή και η πώληση των πλακετών θα γίνονταν από εξωτερικές εταιρείες και η Arduino LLC θα λάμβανε δικαιώματα από αυτές. Το ιδρυτικό καταστατικό της Arduino LLC όριζε ότι κάθε ένας από τους πέντε ιδρυτές θα μεταβιβάσει την κυριότητα της μάρκας Arduino στη νεοσύστατη εταιρεία. Στα τέλη του 2008, η εταιρεία του Gianluca Martino, Smart Projects, κατοχύρωσε το εμπορικό σήμα Arduino στην Ιταλία και το κράτησε μυστικό από τους άλλους συνιδρυτές για περίπου δύο χρόνια. Αυτό αποκαλύφθηκε, όταν η εταιρεία Arduino προσπάθησε να κατοχυρώσει το εμπορικό σήμα σε άλλες περιοχές του κόσμου (αρχικά κατοχύρωσαν μόνο στις ΗΠΑ) και ανακάλυψε ότι ήταν ήδη κατοχυρωμένο στην Ιταλία. Οι διαπραγματεύσεις με τον Martino και την εταιρεία του για να θέσουν το εμπορικό σήμα υπό τον έλεγχο της αρχικής εταιρείας Arduino απέτυχαν. [11] Το 2014, η Smart Projects άρχισε να αρνείται να καταβάλει δικαιώματα. Στη συνέχεια, διόρισαν έναν νέο Διευθύνοντα Σύμβουλο, τον Federico Musto, ο οποίος μετονόμασε την εταιρεία Arduino SRL και δημιούργησε τον ιστότοπο arduino.org, αντιγράφοντας τα γραφικά και τη διάταξη του αρχικού **arduino.cc***. Αυτό οδήγησε σε ρήξη στην ομάδα ανάπτυξης του Arduino. Τον Ιανουάριο του 2015, η Arduino LLC κατέθεσε αγωγή κατά της Arduino SRL. Τον Μάιο του 2015, η Arduino LLC δημιούργησε το παγκόσμιο εμπορικό σήμα Genuino, το οποίο χρησιμοποιείται ως εμπορικό σήμα εκτός των Ηνωμένων Πολιτειών. Στην World Maker Faire στη Νέα Υόρκη την 1η Οκτωβρίου 2016, ο συνιδρυτής και διευθύνων σύμβουλος της Arduino LLC, Massimo Banzi, και ο διευθύνων σύμβουλος της Arduino SRL, Federico Musto, ανακοίνωσαν τη συγχώνευση των δύο εταιρειών, σχηματίζοντας

την Arduino AG. Την ίδια περίπου εποχή, ο Massimo Banzi ανακοίνωσε ότι εκτός από την εταιρεία θα λανσαριζόταν ένα νέο Ίδρυμα Arduino ως «μια νέα αρχή για την Arduino», αλλά αυτή η απόφαση αποσύρθηκε αργότερα. [12]

Τον Απρίλιο του 2017, το Wired ανέφερε ότι ο Musto είχε «κατασκευάσει το ακαδημαϊκό του ιστορικό. Στον ιστότοπο της εταιρείας του, στους προσωπικούς λογαριασμούς του στο LinkedIn, ακόμη και σε ιταλικά επιχειρηματικά έγγραφα, ο Musto, μέχρι πρόσφατα, αναφερόταν ως κάτοχος διδακτορικού από το Τεχνολογικό Ινστιτούτο της Μασαχουσέτης. Σε ορισμένες περιπτώσεις, το βιογραφικό του ανέφερε επίσης MBA από το Πανεπιστήμιο της Νέας Υόρκης.» Το Wired ανέφερε ότι κανένα από τα δύο πανεπιστήμια δεν είχε κανένα αρχείο για την παρακολούθηση του Musto, και ο Musto αργότερα παραδέχτηκε σε μια συνέντευξη στο Wired ότι δεν είχε ποτέ αποκτήσει αυτά τα πτυχία. [13] Η διαμάχη γύρω από τον Musto συνεχίστηκε όταν, τον Ιούλιο του 2017, φέρεται να απέσυρε πολλές άδειες χρήσης ανοιχτού κώδικα, σχηματικά και κώδικα από τον ιστότοπο του Arduino, προκαλώντας έλεγχο και κατακραυγή. Μέχρι το 2017, ο Arduino «AG» κατείχε πολλά εμπορικά σήματα Arduino. Τον Ιούλιο του 2017, η BCMI, που ιδρύθηκε από τους Massimo Banzi, David Cuartielles, David Mellis και Tom Igoe, απέκτησε την Arduino AG και όλα τα εμπορικά σήματα της Arduino. Ο Fabio Violante είναι ο νέος Διευθύνων Σύμβουλος αντικαθιστώντας τον Federico Musto, ο οποίος δεν εργάζεται πλέον για την Arduino AG. [14]

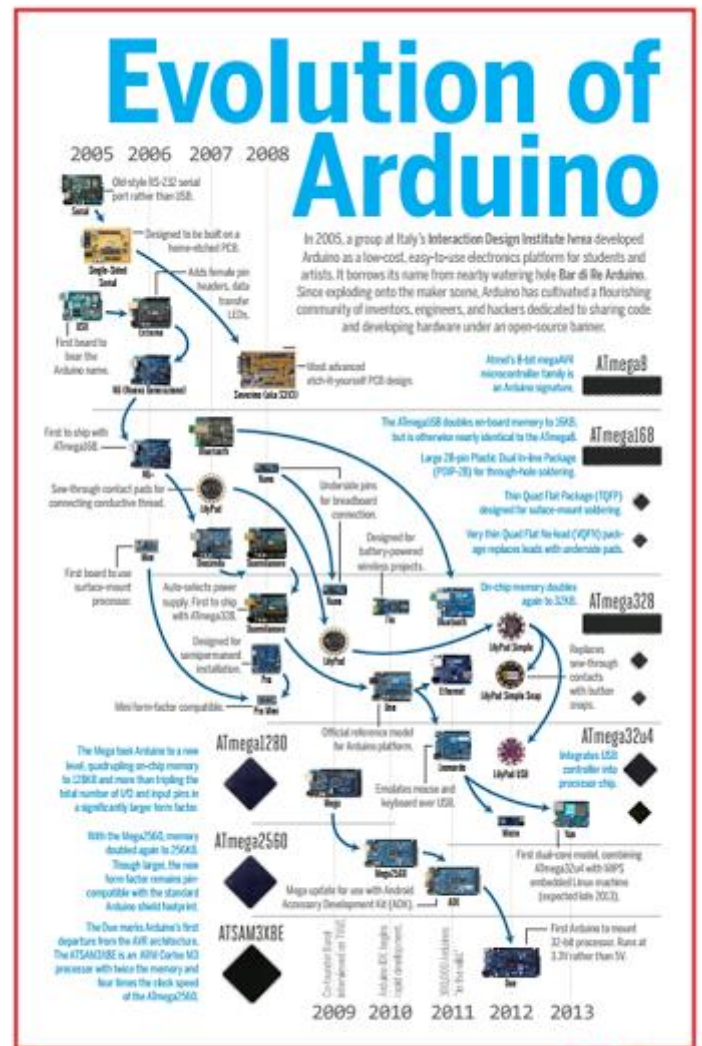
Τον Οκτώβριο του 2017, η Arduino ανακοίνωσε τη συνεργασία της με την Arm Holdings (ARM). Η ανακοίνωση ανέφερε, εν μέρει, ότι «η ARM αναγνώρισε την ανεξαρτησία ως βασική αξία του Arduino χωρίς καμία δέσμευση με την αρχιτεκτονική ARM». Η Arduino σκοπεύει να συνεχίσει να συνεργάζεται με όλους τους προμηθευτές τεχνολογίας και τις αρχιτεκτονικές. [15] Υπό την καθοδήγηση του Violante, η εταιρεία άρχισε να αναπτύσσεται ξανά και να κυκλοφορεί νέα σχέδια. Το εμπορικό σήμα Genuino απορρίφθηκε και όλα τα προϊόντα ονομάστηκαν ξανά Arduino. Τον Αύγουστο του 2018, η Arduino ανακοίνωσε το νέο εργαλείο γραμμής εντολών ανοιχτού κώδικα (arduino-cli), το οποίο μπορεί να χρησιμοποιηθεί ως αντικατάσταση του IDE για τον προγραμματισμό των πλακετών από ένα κέλυφος. Τον Φεβρουάριο του 2019, η Arduino ανακοίνωσε την υπηρεσία IoT* Cloud* ως επέκταση του περιβάλλοντος Create online. Από τον Φεβρουάριο του 2020, η κοινότητα Arduino περιλάμβανε περίπου 30 εκατομμύρια ενεργούς χρήστες με βάση τις λήψεις του IDE. [16]

3.1.4 Υλικό - Hardware

Το Arduino είναι υλικό ανοιχτού κώδικα. Τα σχέδια αναφοράς υλικού διανέμονται με άδεια Creative Commons Attribution Share-Alike 2.5 και είναι διαθέσιμα στον ιστότοπο του Arduino. Διατίθενται επίσης αρχεία διάταξης και παραγωγής για ορισμένες εκδόσεις του υλικού. Παρόλο που τα σχέδια υλικού και

λογισμικού είναι ελεύθερα διαθέσιμα με άδειες copyleft, οι προγραμματιστές έχουν ζητήσει το όνομα Arduino να είναι αποκλειστικό για το επίσημο προϊόν και να μην χρησιμοποιείται για παράγωγα έργα χωρίς άδεια. Το επίσημο έγγραφο πολιτικής σχετικά με τη χρήση του ονόματος Arduino τονίζει ότι το έργο είναι ανοιχτό στην ενσωμάτωση εργασίας άλλων στο επίσημο προϊόν. Αρκετά προϊόντα συμβατά με Arduino που έχουν κυκλοφορήσει εμπορικά έχουν αποφύγει το όνομα του έργου χρησιμοποιώντας διάφορα ονόματα που τελειώνουν σε -duino. [17]

Οι περισσότερες πλακέτες Arduino αποτελούνται από έναν μικροελεγκτή Atmel 8-bit AVR (ATmega8, ATmega168, ATmega328, ATmega1280 ή ATmega2560) με ποικίλες ποσότητες **μνήμης flash***, ακίδων και χαρακτηριστικών. Οι πλακέτες χρησιμοποιούν **ακίδες μονής ή διπλής σειράς*** ή **θηλυκές κεφαλίδες*** που διευκολύνουν τις συνδέσεις για προγραμματισμό και ενσωμάτωση σε άλλα κυκλώματα. Αυτά μπορούν να συνδεθούν με πρόσθετες μονάδες που ονομάζονται ασπίδες. Πολλαπλές και πιθανώς στοιβαγμένες ασπίδες μπορεί να είναι ξεχωριστά διευθυνσιοδοτούμενες μέσω ενός **σειριακού διαύλου***. Οι μικροελεγκτές Arduino είναι προγραμματισμένοι εκ των προτέρων με ένα bootloader που απλοποιεί την αποστολή προγραμμάτων στη μνήμη flash που είναι ενσωματωμένη στο τσιπ. Ο προεπιλεγμένος **bootloader*** του Arduino Uno είναι ο bootloader Optiboot. Οι πλακέτες φορτώνονται με κώδικα προγράμματος μέσω σειριακής



Make:

Εικόνα 7. Arduino Hardware History

σύνδεσης σε έναν άλλο υπολογιστή. [18] Οι τρέχουσες πλακέτες Arduino προγραμματίζονται μέσω Universal Serial Bus (USB), που υλοποιείται χρησιμοποιώντας τσιπ προσαρμογέα USB-σε-σειριακό όπως το FTDI FT232. Ορισμένες πλακέτες, όπως οι νεότερες πλακέτες Uno, αντικαθιστούν το τσιπ FTDI με ένα ξεχωριστό τσιπ AVR που περιέχει firmware USB-σε-σειριακό, το οποίο είναι επαναπρογραμματιζόμενο μέσω της δικής του κεφαλίδας ICSP. Άλλες

παραλλαγές, όπως το Arduino Mini και το ανεπίσημο Boarduino, χρησιμοποιούν μια αποσπώμενη πλακέτα ή καλώδιο προσαρμογέα USB-σε-σειριακό, Bluetooth ή άλλες μεθόδους. Όταν χρησιμοποιείται με παραδοσιακά εργαλεία μικροελεγκτών, αντί για το Arduino IDE, χρησιμοποιείται ο τυπικός **προγραμματισμός AVR*** εντός συστήματος (**ISP***). [19]

Υπάρχουν πολλές πλακέτες συμβατές με Arduino και προερχόμενες από Arduino. Μερικές είναι λειτουργικά ισοδύναμες με ένα Arduino και μπορούν να χρησιμοποιηθούν εναλλακτικά. Πολλές βελτιώνουν το βασικό Arduino προσθέτοντας προγράμματα οδήγησης εξόδου, συχνά για χρήση στην εκπαίδευση σχολικού επιπέδου, για να απλοποιήσουν την κατασκευή καρτσιών και μικρών ρομπότ. Άλλες είναι ηλεκτρικά ισοδύναμες, αλλά αλλάζουν τον παράγοντα μορφής, διατηρώντας μερικές φορές τη συμβατότητα με τις ασπίδες, μερικές φορές όχι. Ορισμένες παραλλαγές χρησιμοποιούν διαφορετικούς επεξεργαστές, ποικίλης συμβατότητας. Εκτός από τις παραλλαγές υλικού, έχουν αναπτυχθεί βιβλιοθήκες ανοιχτού κώδικα για την υποστήριξη υλικού Arduino σε εργαλεία EDA. Ένα τέτοιο έργο παρέχει σχηματικά σύμβολα (**KiCad***) και αποτυπώματα PCB για μονάδες Arduino, πλακέτες επέκτασης και συνδετήρες, διευκολύνοντας τους μηχανικούς να ενσωματώσουν το Arduino στα σχέδιά τους. [20]

3.1.5 Χρήση του Arduino στην εργασία

Όπως καταλαβαίνουμε οι Arduino πλακέτες χρησιμοποιούνται για την δημιουργία πρωτοτύπων μικροηλεκτρονικών συσκευών. Σκοπός μου είναι πάνω σε αυτή την πλακέτα να ολοκληρώσω συναρτήσεις/κώδικα για την δημιουργία:

- Ενσωματωμένου εξυπηρετητή.
- Συναρτήσεων για την επικοινωνία πελάτη – εξυπηρετητή.
- Συναρτήσεων για τον έλεγχο κίνησης/παροχής ενέργειας των κινητήρων (DC motors and Servo motors).
- Συναρτήσεων για την ενεργοποίηση, σύνδεση και άντληση δεδομένων από τους αισθητήρες.
- Συναρτήσεων για την χρήση των δεδομένων των αισθητήρων για την υποβοήθηση στην χειροκίνητη κίνηση αρχικά.
- Συναρτήσεων για την παροχή ενέργειας σε εξωτερικά LED.
- Συναρτήσεων για τον έλεγχο του LED Matrix πάνω στην πλακέτα.
- Συναρτήσεων για την αυτόματη κίνηση του οχήματος καθώς και την χαρτογράφηση του γύρω χώρου του.

3.1.6 Arduino UNO R4 Wi-Fi

Η πλακέτα που έχω επιλέξει να υλοποιήσω την εργασία μου είναι το Arduino UNO R4 Wi-Fi. Η επιλογή αυτή έγινε με βάση τα παρακάτω χαρακτηριστικά του:

1. Συμβατότητα υλικού με τον παράγοντα μορφής UNO γιατί διατηρεί τον ίδιο παράγοντα μορφής, pinout και τάση λειτουργίας 5 V με τον προκάτοχό του, το UNO R3 με του οποίου την χρήση έχω ήδη εξοικειωθεί καθώς ήταν μέσα στο κιτ του σεμιναρίου «**Ανάλυση και Προγραμματισμός Συστημάτων**



Εικόνα 8. Arduino Uno R4 Wi-Fi

2. **Αυτοματισμού**», εξασφαλίζοντας μια απρόσκοπτη μετάβαση για υπάρχουσες ασπίδες και έργα.
3. Επεκταμένη μνήμη και ταχύτερο ρολόι αφού διαθέτει αυξημένη μνήμη και **ταχύτερη ταχύτητα ρολογιού***, επιτρέποντας πιο ακριβείς υπολογισμούς και εύκολη διαχείριση σύνθετων έργων.
4. Εισάγει μια σειρά ενσωματωμένων περιφερειακών, συμπεριλαμβανομένου ενός 12-bit DAC, CAN BUS και OP AMP, παρέχοντας εκτεταμένες δυνατότητες και ευελιξία σχεδιασμού.
5. Εκτεταμένη ανοχή 24 V καθώς υποστηρίζει ένα ευρύτερο εύρος τάσης* εισόδου, επιτρέποντας την απρόσκοπτη ενσωμάτωση με κινητήρες, ταινίες LED και άλλους ενεργοποιητές χρησιμοποιώντας μία μόνο πηγή τροφοδοσίας. Αυτό είναι σημαντικό καθώς προστατεύει την πλακέτα μου από σφάλματα ή προβλήματα που μπορεί να εμφανίσουν οι κινητήρες ή/και αισθητήρες που έχω στο κύκλωμα μου.
6. Με ενσωματωμένη υποστήριξη HID, το UNO R4 Wi-Fi μπορεί να προσομοιώσει ένα ποντίκι ή πληκτρολόγιο όταν συνδέεται σε έναν υπολογιστή μέσω USB, διευκολύνοντας την αποστολή πληκτρολογήσεων και κινήσεων του ποντικιού.

6. Το UNO R4 Wi-Fi φιλοξενεί μια μονάδα ESP32-S3, επιτρέποντας στους κατασκευαστές να προσθέσουν ασύρματη συνδεσιμότητα **Wi-Fi®** και **Bluetooth®** στα έργα τους. Σε συνδυασμό με το **Arduino IoT Cloud***, μπορώ να παρακολουθήσω και να ελέγχω τα έργα μου από απόσταση. Πάνω σε αυτή την μονάδα θα στηθεί ο embedded web server μου.
7. Περιλαμβάνει επιπλέον ακίδες, συμπεριλαμβανομένης μιας ακίδας "OFF" για την απενεργοποίηση της πλακέτας και μιας ακίδας "VRTC" για να διατηρεί το εσωτερικό ρολόι πραγματικού χρόνου σε λειτουργία.
8. Ενσωματώνει έναν φωτεινό κόκκινο πίνακα LED 12x8, ιδανικό για δημιουργικά έργα με κινούμενα σχέδια ή σχεδίαση δεδομένων αισθητήρων, εξαλείφοντας την ανάγκη για πρόσθετο υλικό. Με τη χρήση αυτού του πίνακα είναι εύκολο να καταλαβαίνω ποιες συναρτήσεις μου εκτελούνται σε πραγματικό χρόνο χωρίς να υπάρχει ανάγκη για εξωτερική σύνδεση **LED*** ή άλλων ενδείξεων.
9. Τέλος, περιλαμβάνει έναν μηχανισμό ανίχνευσης σφαλμάτων που ανιχνεύει σφάλματα χρόνου εκτέλεσης και παρέχει λεπτομερείς εξηγήσεις και συμβουλές σχετικά με τη γραμμή κώδικα που προκαλεί το σφάλμα.

3.1.7 Χρήση βιβλιοθηκών

Οι **βιβλιοθήκες Arduino** είναι συνήθως μικρά πακέτα κώδικα που έγραψε κάποιος άλλος για έναν συγκεκριμένο σκοπό. Ένα παράδειγμα είναι μια βιβλιοθήκη που προσθέτει υποστήριξη για έναν συγκεκριμένο αισθητήρα, όπως το **BME280***. Η χρήση βιβλιοθηκών σημαίνει ότι δεν χρειάζεται να γνωρίζουμε έναν αισθητήρα ή μια μονάδα από την αρχή μέχρι το τέλος για να τη συνδέσουμε σε ένα Arduino και να τη δοκιμάσουμε. Επιπλέον, δεν χρειάζεται να φτιάξουμε νέες συναρτήσεις εάν υπάρχει μια βιβλιοθήκη που ήδη υλοποιεί μια συνήθως χρησιμοποιούμενη λειτουργία (π.χ., μαθηματικές συναρτήσεις όπως υπολογισμοί διανυσμάτων και πινάκων). Το Arduino IDE συνοδεύεται από μερικές ενσωματωμένες βιβλιοθήκες αμέσως. Εκτός από αυτές, συνήθως μπορούμε να βρούμε νέες βιβλιοθήκες που προσθέτουν υποστήριξη για **μονάδες breakout*** στον ιστότοπο του προμηθευτή ή του κατασκευαστή. Καθώς ο καθένας είναι ελεύθερος να γράψει τις δικές του βιβλιοθήκες Arduino, μπορούμε επίσης να κατεβάσουμε νέες βιβλιοθήκες σε πολλούς προσωπικούς ιστότοπους. Όταν κατεβάζουμε χειροκίνητα μια βιβλιοθήκη για το Arduino IDE, συνήθως καταλήγουμε σε ένα μόνο **αρχείο zip*** που περιέχει όλα τα αρχεία κώδικα. Υπάρχει, ωστόσο, και ένας ευκολότερος τρόπος εγκατάστασης βιβλιοθηκών Arduino με τον διαχειριστή βιβλιοθηκών. [21]



Εικόνα 9. Arduino Libraries

Οι βιβλιοθήκες που χρησιμοποίησα για την εργασία περιλαμβάνουν :

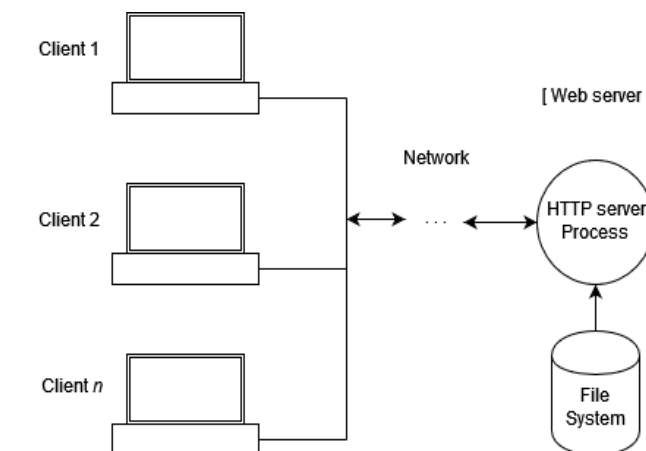
- Το «**Arduino.h**» περιέχει πολλούς ορισμούς και τυπικά στοιχεία. Εάν δεν συμπεριλάβω το **Arduino.h**, ενδέχεται να φορτωθεί από άλλα αρχεία κεφαλίδας. Το IDE του **Arduino** το εισάγει αυτό κατά τη στιγμή της μεταγλώττισης, εάν δεν πληκτρολογήσω τη γραμμή στον κώδικά μου. Άλλα IDE, όπως το **VS Code/Platformio** ή το **eclipse**, δεν το κάνουν, επομένως πρέπει να το κάνω μόνος.
- Το «**Arduino_LED_Matrix**» για το **UNO R4 Wi-Fi** λειτουργεί με βάση την αρχή της δημιουργίας ενός **frame** και στη συνέχεια της φόρτωσής του σε ένα **buffer** που εμφανίζει το **frame**. Ένα **frame** είναι αυτό που ονομάζουμε "εικόνα" που εμφανίζεται σε οποιαδήποτε δεδομένη στιγμή στο **matrix**. Εάν μια κινούμενη εικόνα είναι μια σειρά εικόνων, ένα **frame** είναι μία από αυτές τις εικόνες της σειράς. [22]
- Το «**Servo**» επιτρέπει σε μια πλακέτα **Arduino** να ελέγχει σερβοκινητήρες **RC**. Τα σερβοκινητήρες διαθέτουν ενσωματωμένα γρανάζια και έναν άξονα που μπορεί να ελεγχθεί με ακρίβεια. Τα τυπικά σερβοκινητήρες επιτρέπουν την τοποθέτηση του άξονα σε διάφορες γωνίες, συνήθως μεταξύ 0 και 180 μοιρών. Οι σερβοκινητήρες συνεχούς περιστροφής επιτρέπουν τη ρύθμιση της περιστροφής του άξονα σε διάφορες ταχύτητες. [23]
- Η βιβλιοθήκη **Arduino «ShiftRegister74HC595»** του καταχωρητή μετατόπισης (**shift register**) **74HC595** απλοποιεί τη χρήση των καταχωρητών μετατόπισης. Επιτρέπει να ορίσουμε μεμονωμένες ακίδες του καταχωρητή μετατόπισης είτε υψηλά είτε χαμηλά, όπως ακριβώς και οι κανονικές ακίδες **Arduino**. Με αυτόν τον τρόπο, εξαλείφει την επιβάρυνση των **bytes*** μετατόπισης που δημιουργήθηκαν με περίπλοκες λειτουργίες **bit-wise**. [24]
- Η βιβλιοθήκη «**ArduinSTL**» είναι μια υλοποίηση μιας τυπικής βιβλιοθήκης **C++** που έχει συσκευαστεί ως βιβλιοθήκη **Arduino**. Η βιβλιοθήκη υποστηρίζει τη διδασκαλία του μαθήματος **CS-11M*** προσθέτοντας βασικά χαρακτηριστικά **C++** στο περιβάλλον **Arduino**. Από αυτή τη βιβλιοθήκη χρησιμοποιούμε τους τύπους «**string**»* και «**vector**»*. [25]
- Η βιβλιοθήκη «**mWebSockets**» του **Dawid Kurek** είναι μια ελαφριά και εύχρηστη υλοποίηση του πρωτοκόλλου **WebSocket (RFC6455*)** που έχει σχεδιαστεί ειδικά για μικροελεγκτές. Επιτρέπει την αμφίδρομη επικοινωνία σε πραγματικό χρόνο μεταξύ συσκευών που βασίζονται σε **Arduino** και **web clients**, καθιστώντας την ιδανική για εφαρμογές του Διαδικτύου των Πραγμάτων (**IoT**) όπου η αλληλεπίδραση χαμηλής καθυστέρησης είναι απαραίτητη. Προσφέρει αμφίδρομη επικοινωνία σε πραγματικό χρόνο σύμφωνα με το πρωτόκολλο **WebSocket**, εξασφαλίζοντας αξιόπιστη και τυποποιημένη επικοινωνία, ενώ παράλληλα υποστηρίζει αποδοτικά διάφορες αρχιτεκτονικές **Arduino** χωρίς να χρησιμοποιεί πολλούς πόρους. [26]

- Η βιβλιοθήκη «**WiFiS3**» αναφέρεται στο **Spresense 3**, τον ενσωματωμένο συνεπεξεργαστή **ESP32-S3*** στο Arduino Uno R4 Wi-Fi. Αυτός ο ESP32 χειρίζεται την επικοινωνία Wi-Fi και η βιβλιοθήκη διευκολύνει την αλληλεπίδραση μεταξύ του κύριου μικροελεγκτή **Renesas RA4M1*** και του τσιπ ESP32-S3. [27]

3.2 Ενσωματωμένος διαδικτυακός εξυπηρετητής

3.2.1 Γενικά

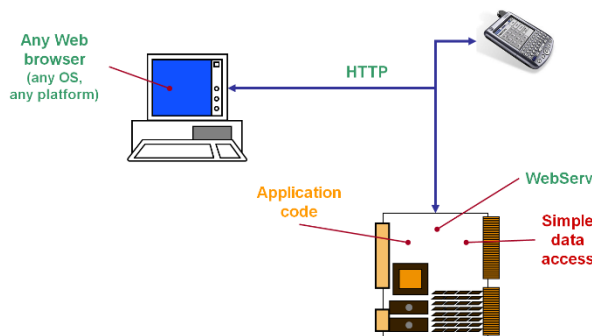
Ένας **διακομιστής ιστού*** είναι λογισμικό υπολογιστή και υποκείμενο υλικό που δέχεται αιτήματα μέσω **HTTP*** (το πρωτόκολλο δικτύου που δημιουργείται για τη διανομή περιεχομένου ιστού) ή της ασφαλούς παραλλαγής του **HTTPS***. Ένας παράγοντας χρήστη, συνήθως ένα **πρόγραμμα περιήγησης ιστού*** ή ένα **πρόγραμμα ανίχνευσης ιστού***, ξεκινά την επικοινωνία υποβάλλοντας ένα αίτημα για μια ιστοσελίδα ή άλλο πόρο



Εικόνα 10. Web Server

χρησιμοποιώντας HTTP και ο διακομιστής απαντά με το περιεχόμενο αυτού του πόρου ή ένα μήνυμα σφάλματος. Ένας διακομιστής ιστού μπορεί επίσης να δέχεται και να αποθηκεύει πόρους που αποστέλλονται από τον παράγοντα χρήστη, εάν έχει ρυθμιστεί να το κάνει αυτό. Το υλικό που χρησιμοποιείται για την εκτέλεση ενός διακομιστή ιστού μπορεί να ποικίλλει ανάλογα με τον όγκο των αιτημάτων που πρέπει να χειριστεί. Στο χαμηλότερο άκρο του εύρους βρίσκονται τα ενσωματωμένα συστήματα, όπως ένας **δρομολογητής*** που εκτελεί έναν μικρό διακομιστή ιστού ως **διεπαφή διαμόρφωσης***. Ένας **ιστότοπος διαδικτύου υψηλής επισκεψιμότητας*** μπορεί να χειρίζεται αιτήματα με εκατοντάδες διακομιστές που εκτελούνται σε **rack υπολογιστών υψηλής ταχύτητας***. [28]

Συγκεκριμένα ένας ενσωματωμένος διακομιστής είναι ένας διακομιστής που χρησιμοποιείται σε ένα ενσωματωμένο σύστημα και συνήθως υλοποιείται ως στοιχείο λογισμικού ενός συστήματος εφαρμογής (ενσωματωμένου) που ελέγχει ή/και παρακολουθεί ένα μηχάνημα με μηχανικά ή/και ηλεκτρικά μέρη. Ο διακομιστής υλοποιεί το πρωτόκολλο HTTP προκειμένου να επιτρέψει την επικοινωνία με έναν ή περισσότερους τοπικούς ή απομακρυσμένους χρήστες



Εικόνα 11. Embedded Web Server

χρησιμοποιώντας ένα πρόγραμμα περιήγησης. Στόχος είναι να επιτρέψει στους χρήστες να αλληλεπιδρούν με πληροφορίες που παρέχονται από το ενσωματωμένο σύστημα (διεπαφή χρήστη, παρακολούθηση δεδομένων, καταγραφή δεδομένων, διαμόρφωση

δεδομένων κ.λπ.) μέσω δικτύου, χωρίς να χρησιμοποιούν παραδοσιακά περιφερειακά που απαιτούνται για τοπικές διεπαφές χρήστη (οθόνη, πληκτρολόγιο κ.λπ.). Σε ορισμένες περιπτώσεις, οι λειτουργίες που παρέχονται μέσω του διακομιστή HTTP επιτρέπουν επίσης επικοινωνίες μεταξύ προγραμμάτων, π.χ. για την ανάκτηση δεδομένων που έχουν καταγραφεί σχετικά με το παρακολουθούμενο μηχάνημα κ.λπ. [29]

3.2.2 Χρήσεις

Μερικά παραδείγματα χρήσης είναι η παροχή μιας διεπαφής για μια παραδοσιακή εφαρμογή, εργαλείων ευρετηρίασης, αναφοράς και εντοπισμού σφαλμάτων κατά το στάδιο ανάπτυξης, εφαρμογή ενός πρωτοκόλλου για τη διανομή και την απόκτηση πληροφοριών που θα εμφανίζονται στην κανονική διεπαφή, πιθανώς μια υπηρεσία ιστού και πιθανώς χρησιμοποιώντας **XML*** ως μορφή δεδομένων, για την ανάπτυξη μιας εφαρμογής ιστού.

3.2.3 Χρήση WebSockets

Το WebSocket είναι ένα πρωτόκολλο επικοινωνίας υπολογιστή, το οποίο παρέχει ένα ταυτόχρονο κανάλι αμφίδρομης επικοινωνίας μέσω μίας μόνο σύνδεσης **Transmission Control Protocol*** (**TCP**). Το πρωτόκολλο WebSocket τυποποιήθηκε από το **IETF*** ως RFC 6455 το 2011. Η τρέχουσα προδιαγραφή που επιτρέπει σε εφαρμογές ιστού να χρησιμοποιούν αυτό το πρωτόκολλο είναι γνωστή ως WebSockets. Το WebSocket διαφέρει από το HTTP που χρησιμοποιείται για την εξυπηρέτηση των περισσότερων ιστοσελίδων. Αν και είναι διαφορετικές, το RFC 6455 δηλώνει ότι το WebSocket "έχει σχεδιαστεί για να λειτουργεί μέσω των θυρών HTTP 443 και 80, καθώς και για να υποστηρίξει διακομιστές μεσολάβησης και ενδιάμεσους HTTP", καθιστώντας το έτσι συμβατό με το HTTP. Για να επιτευχθεί συμβατότητα, η χειραψία WebSocket χρησιμοποιεί την κεφαλίδα αναβάθμισης HTTP για να αλλάξει από το πρωτόκολλο HTTP στο πρωτόκολλο WebSocket. [30]



Εικόνα 12. WebSocket Protocol

Το πρωτόκολλο WebSocket επιτρέπει την αμφίδρομη αλληλεπίδραση μεταξύ ενός προγράμματος περιήγησης ιστού (ή άλλης εφαρμογής-πελάτη) και ενός διακομιστή ιστού με χαμηλότερο κόστος από τις εναλλακτικές λύσεις ημιαμφίδρομης επικοινωνίας, όπως η **HTTP polling***, διευκολύνοντας τη μεταφορά δεδομένων σε πραγματικό χρόνο από και προς τον διακομιστή. Αυτό καθίσταται δυνατό παρέχοντας έναν τυποποιημένο τρόπο για τον διακομιστή να στέλνει περιεχόμενο στον πελάτη χωρίς να του ζητηθεί πρώτα από τον πελάτη και

επιτρέποντας την ανταλλαγή μηνυμάτων μεταξύ τους διατηρώντας παράλληλα τη σύνδεση ανοιχτή. Με αυτόν τον τρόπο, μπορεί να πραγματοποιηθεί μια αμφίδρομη συνεχής συνομιλία μεταξύ του πελάτη και του διακομιστή. Οι επικοινωνίες συνήθως πραγματοποιούνται μέσω της θύρας TCP με αριθμό 443 (ή 80 στην περίπτωση μη ασφαλών συνδέσεων), κάτι που είναι επωφελές για περιβάλλοντα που αποκλείουν τις συνδέσεις στο Διαδίκτυο εκτός ιστού χρησιμοποιώντας ένα τείχος προστασίας. Επιπλέον, το WebSocket επιτρέπει ροές μηνυμάτων πάνω από το TCP. Το TCP από μόνο του χειρίζεται ροές byte χωρίς εγγενή έννοια μηνύματος. Παρόμοιες αμφίδρομες επικοινωνίες προγράμματος περιήγησης-διακομιστή έχουν επιτευχθεί με μη τυποποιημένους τρόπους χρησιμοποιώντας τεχνολογίες προσωρινής αποθήκευσης όπως το **Comet*** ή το **Adobe Flash Player***. Τα περισσότερα προγράμματα περιήγησης υποστηρίζουν το πρωτόκολλο, συμπεριλαμβανομένων των Google Chrome, Firefox, Microsoft Edge, Internet Explorer, Safari και Opera. [31] Για αυτό το λόγο έγινε και η επιλογή του συγκεκριμένου πρωτοκόλλου για την αμφίδρομη επικοινωνία μεταξύ του ενσωματωμένου εξυπηρετητή και πελάτη.

3.3 Γραφική Διεπαφή HTML

3.3.1 Γενικά

Η **HTML*** (**Hypertext Markup Language**), στα ελληνικά γλώσσα σήμανσης υπερκειμένου,) είναι η τυπική γλώσσα σήμανσης για έγγραφα που έχουν σχεδιαστεί για εμφάνιση σε ένα πρόγραμμα περιήγησης ιστού. Ορίζει το περιεχόμενο και τη δομή του περιεχομένου ιστού. Συχνά υποστηρίζεται από τεχνολογίες όπως τα **Cascading Style Sheets*** (**CSS**) και γλώσσες δέσμης ενεργειών όπως η **JavaScript***, μια γλώσσα προγραμματισμού. Τα προγράμματα περιήγησης ιστού λαμβάνουν έγγραφα HTML από έναν διακομιστή ιστού ή από τοπικό χώρο αποθήκευσης και αποδίδουν τα έγγραφα σε ιστοσελίδες πολυμέσων. Η HTML περιγράφει σημασιολογικά τη δομή μιας ιστοσελίδας και αρχικά περιλάμβανε ενδείξεις για την εμφάνισή της. [32]

Τα στοιχεία HTML είναι τα δομικά στοιχεία των σελίδων HTML. Με τις δομές HTML, εικόνες και άλλα αντικείμενα όπως διαδραστικές φόρμες μπορούν να ενσωματωθούν στη σελίδα που αποδίδεται. Η HTML παρέχει ένα μέσο για τη δημιουργία δομημένων εγγράφων υποδηλώνοντας δομική σημασιολογία για κείμενο, όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους, εισαγωγικά και



Εικόνα 13. HTML

άλλα στοιχεία. Τα στοιχεία HTML οριοθετούνται από ετικέτες, γραμμένες χρησιμοποιώντας αγκύλες. Η HTML μπορεί να ενσωματώσει προγράμματα γραμμένα σε μια γλώσσα scripting όπως η JavaScript, η οποία επηρεάζει τη συμπεριφορά και το περιεχόμενο των ιστοσελίδων. Η συμπερίληψη του CSS καθορίζει την εμφάνιση και τη διάταξη του περιεχομένου. Η Κοινοπραξία του Παγκοσμίου Ιστού (**W3C***), πρώην συντηρητής της HTML και νυν συντηρητής των προτύπων CSS, ενθαρρύνει τη χρήση του CSS αντί της σαφούς HTML παρουσίασης από το 1997. [33]

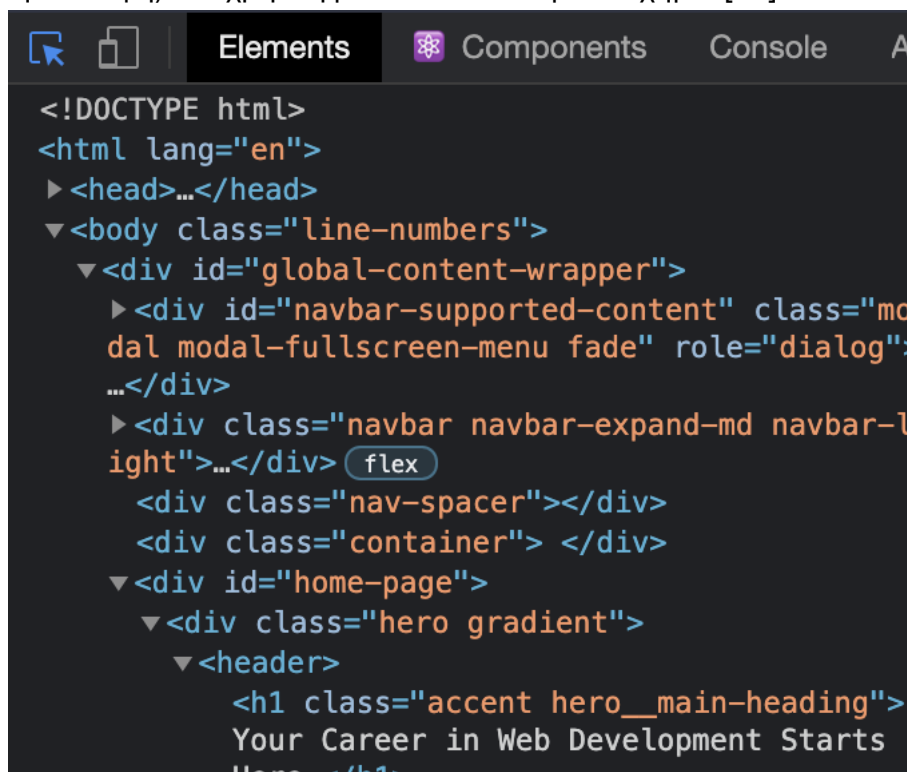
3.3.2 Χρήσεις σήμερα

Οι βασικές χρήσεις του HTML σήμερα περιλαμβάνουν:

1. **Ανάπτυξη Ιστοσελίδων:** Από το 1999, έχει γίνει πρότυπο και χρησιμοποιείται ευρέως για τη δημιουργία ιστοσελίδων. Ωστόσο, οι προγραμματιστές το χρησιμοποιούν με CSS για καλύτερη ποιότητα. Η ανάπτυξη ιστοσελίδων είναι ο πιο σημαντικός τομέας στον οποίο χρησιμοποιείται η HTML.
2. **Πλοήγηση στο Διαδίκτυο:** Η έννοια του υπερκειμένου στην HTML σημαίνει ότι ο χρήστης μπορεί να μεταβαίνει από τη μία ιστοσελίδα στην άλλη απρόσκοπτα.
3. **Δημιουργία Ιστοσελίδων:** Η HTML κυριαρχεί στο διαδίκτυο στη δημιουργία εγγράφων μέσω των βασικών εννοιών της, όπως το μοντέλο αντικειμένου εγγράφου μέσω **DOM*** και **ετικέτας***.
4. **Χαρακτηριστικό αιχμής:** Η HTML έχει χρησιμοποιηθεί σε πολλές τέτοιες πρόσφατες καινοτομίες, χρησιμοποιώντας τα πρότυπα και τα API της.
5. **Εισαγωγή Δεδομένων:** Κάθε ποσότητα δεδομένων και πληροφοριών μπορεί να κοινοποιηθεί σχεδόν αμέσως.
6. **Σχεδιασμός Εικόνας με Ανταπόκριση:** Ο σχεδιασμός με Ανταπόκριση(**responsive***) είναι αναπόσπαστο μέρος της ανάπτυξης ιστοσελίδων.
7. **Δυνατότητες εκτός σύνδεσης:** Η HTML επιτρέπει στους χρήστες και τους προγραμματιστές να έχουν πρόσβαση σε εφαρμογές και ιστότοπους ενώ είναι εκτός σύνδεσης. Χρησιμοποιεί **αρχεία manifest cache*** για να επιλέξει ποια δεδομένα θα αποθηκεύσει εκτός σύνδεσης.
8. **Ανάπτυξη παιχνιδιών:** Μπορεί να χρησιμοποιηθεί σε πολλές πλατφόρμες, συμπεριλαμβανομένων των παιχνιδιών. Οι προγραμματιστές παιχνιδιών μπορούν να ενσωματώσουν HTML με JavaScript για να αναπτύξουν διαδικτυακά παιχνίδια.
9. **Αποθήκευση:** Διαθέτει πολλές νέες δυνατότητες, όπως το **localStorage*** και το **IndexedDB***, που έχουν μεταμορφώσει τον τρόπο αποθήκευσης των δεδομένων χρήστη.
10. **Εγγενή API για ιστότοπους:** Έχει εισαγάγει πολλές νέες δυνατότητες και εργαλεία που ήταν αδιανόητα στο παρελθόν. Με νέα API για βοήθεια με

συστήματα αρχείων, μεταφορά και απόθεση, γεωγραφική τοποθεσία, αποθήκευση πελατών κ.λπ., μπορούμε να δημιουργήσουμε ιστοσελίδες με ευκολία.

Στην εργασία μου χρησιμοποιείται ιστοσελίδα HTML ως user interface (γραφική διεπαφή) του χρήστη με το αυτοκινούμενο όχημα. [34]



```

<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body class="line-numbers">
    <div id="global-content-wrapper">
      <div id="navbar-supported-content" class="modal modal-fullscreen-menu fade" role="dialog">...</div>
      <div class="navbar navbar-expand-md navbar-light">...</div> flex
        <div class="nav-spacer"></div>
        <div class="container"> </div>
      <div id="home-page">
        <div class="hero gradient">
          <header>
            <h1 class="accent hero__main-heading">
              Your Career in Web Development Starts
              Here </h1>
          </header>
        </div>
      </div>
    </div>
  </body>
</html>
  
```

Εικόνα 14 . Παράδειγμα κώδικα HTML

4 Προσέγγιση εργασίας

4.1 Διερεύνηση σκοπού

Το αρχικό θέμα της εργασίας ήταν η «**Οπτικοποίηση δεδομένων από συστήματα αυτοματισμού με δημιουργία ενσωματωμένου εξυπηρετητή**», όμως έχω διευρύνει το πλαίσιο της εργασίας έτσι ώστε να μελετήσω και να προσπαθήσω να δώσω λύση στην δημιουργία ενός οχήματος το οποίο θα ελέγχεται από τον ενσωματωμένο διαδικτυακό εξυπηρετητή. Ο έλεγχος αυτός θέλω να γίνεται χειροκίνητα από το χρήστη, με ή χωρίς την χρήση αισθητήρων για υποβοήθεια στην οδήγηση, και αυτόματα. Στόχος της αυτόματης κίνησης είναι η χαρτογράφηση χώρου και η παρουσίαση του στο χρήστη. Συνολικά, θέλω πάνω στην πλακέτα Arduino να δημιουργείται ο server και μόλις συνδεθεί ένας client μέσω browser, να του σερβίρεται μια ιστοσελίδα από την οποία μπορεί να ελέγχει την κίνηση του οχήματος.

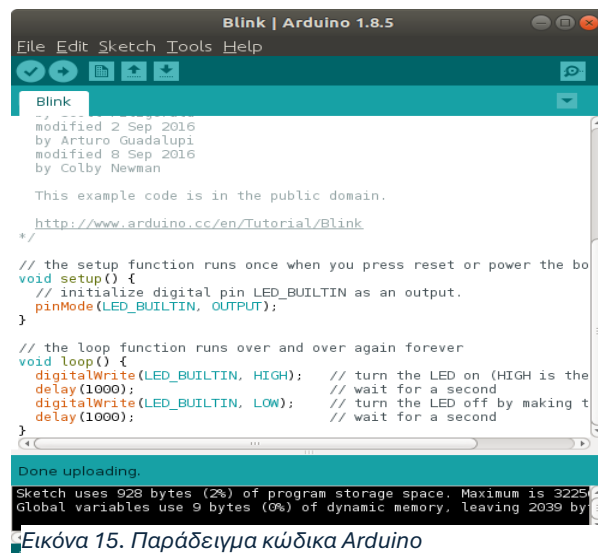
4.2 Software

Όπως έχω πολύ αναφερθεί, η ανάπτυξη του κώδικα που ανεβαίνει πάνω στην πλακέτα έχει γίνει σε περιβάλλον Arduino IDE και το graphical interface σε Visual Studio Code. Η επιλογή του υλικού τόσο της κύριας κατασκευής όσο και των επιπλέον στοιχείων προηγείται της ανάπτυξης του κώδικα καθώς αυτός μεταβάλλεται ανάλογα με τα προηγούμενα.

4.3 Hardware

Όσο αφορά το υλικό της κατασκευής αποτελείται από τρισδιάστατα εκτυπωμένα μέρη που προήλθαν από την συμμετοχή μου στο σεμινάριο «**Ανάλυση και Προγραμματισμός Συστημάτων Αυτοματισμού**» από το NOESIS HUB, συνεργασία του **Μητροπολιτικού Κολεγίου** με το τμήμα **HMTY** του **Πανεπιστημίου Πατρών**. [35] Στο υλικό αυτό έγιναν κάποιες αλλαγές ως προς την πλακέτα που τελικά χρησιμοποίησα και στην προσθήκη περισσότερων αισθητήρων και άλλων στοιχείων. Συγκεκριμένα, τα υλικά στοιχεία που χρησιμοποίησα είναι:

- 1) Τα **3D-Printed Parts*** και κάποια ακόμα πλαστικά στηρίγματα που πρόσθεσα αποτελούν τη βάση του σκελετού. Πάνω σε αυτό τοποθετώ όλα τα υπόλοιπα στοιχεία που θα χρειαστώ.
- 2) Η πλακέτα **Arduino Uno R4 Wi-Fi**, όπως έχω προαναφέρει.
- 3) **Power bank** για την ανεξάρτητη παροχή ενέργειας.
- 4) Το **Breadboard**, δηλαδή μία πλακέτα διασύνδεσης χωρίς κολλήσεις ή ράστερ ή πλακέτα δοκιμών ή πλακέτα γενικών συνδέσεων είναι μια βάση πάνω στην οποία προσαρμόζονται (συνήθως) προσωρινά ηλεκτρονικά εξαρτήματα ώστε να δοκιμαστεί ένα πρωτότυπο (μοντέλο) ηλεκτρονικό κύκλωμα. Η τοποθέτηση των εξαρτημάτων, δεν χρειάζεται κόλληση, είναι προσωρινή και έτσι τόσο τα εξαρτήματα όσο και το breadboard μπορούν να επαναχρησιμοποιηθούν. [36]
- 5) Κινητήρες συνεχούς ρεύματος ή αλλιώς ηλεκτρικοί κινητήρες που χρησιμοποιούν συνεχές ρεύμα (**DC**) για την παραγωγή μηχανικής δύναμης. Οι πιο συνηθισμένοι τύποι βασίζονται σε μαγνητικές δυνάμεις που παράγονται από ρεύματα στα πηνία. Σχεδόν όλοι οι τύποι κινητήρων συνεχούς ρεύματος διαθέτουν κάποιο εσωτερικό μηχανισμό, είτε ηλεκτρομηχανικό είτε ηλεκτρονικό, για να αλλάζουν περιοδικά την κατεύθυνση του ρεύματος σε ένα μέρος του κινητήρα. [37]



Εικόνα 15. Παράδειγμα κώδικα Arduino

- 6) Κινητήρα **servo** που είναι ένας περιστροφικός ή γραμμικός ενεργοποιητής που επιτρέπει τον ακριβή έλεγχο της γωνιακής ή γραμμικής θέσης, της ταχύτητας και της επιτάχυνσης σε ένα μηχανικό σύστημα. Αποτελεί μέρος ενός σερβομηχανισμού και αποτελείται από έναν κατάλληλο κινητήρα συνδεδεμένο με έναν αισθητήρα για ανάδραση θέσης και έναν ελεγκτή. [38]
- 7) Μονάδες οδήγησης κινητήρων 2 καναλιών **L9110S** που είναι μια συμπαγείς πλακέτες και τις χρησιμοποιώ για την λειτουργία των Dc motor. Η κάθε μονάδα διαθέτει δύο ανεξάρτητα τσιπ οδήγησης κινητήρα, τα οποία μπορούν να οδηγήσουν έως και **800mA** συνεχούς ρεύματος το καθένα. [39]
- 8) Μονάδες υπερηχητικής απόστασης **HC-SR04**, με τις οποίες μετρώ απόσταση 2cm - 400cm. [40]
- 9) Αισθητήρες αποφυγής εμποδίων υπέρυθρων IR **KY-032**. [41] Αυτούς τους αισθητήρες τους χρησιμοποιώ αρχικά για να μετρώ απόσταση δεξιά και αριστερά του αμαξιδίου, όμως στις τελευταίες version τους αντικαθιστώ με μονάδες HC-SR04.
- 10) Μονάδες αισθητήρα μέτρησης περιστροφής υπέρυθρων. Κάθε φορά που ένα αντικείμενο διακόπτει ή επαναφέρει την ροή υπέρυθρων από το ένα λαμπάκι στο άλλο μεταβάλλεται η έξοδος του. [42]
- 11) **Shift Registers** (καταχωρητές μετατόπισης) τύπου **74HC595** . Είναι ένας τύπος 8-βιτ ψηφιακού κυκλώματος που χρησιμοποιεί μια αλληλουχία από **flip-flops*** όπου η έξοδος ενός flip-flop συνδέεται με την είσοδο του επόμενου. Μοιράζονται ένα μόνο σήμα ρολογιού, το οποίο προκαλεί τη μετατόπιση των δεδομένων που είναι αποθηκευμένα στο σύστημα από τη μία θέση στην άλλη. [43]
- 12) Διάφορα αλλά στοιχεία όπως φωτάκια τύπου LED, **αντιστάσεις***, καλώδια για την σύνδεση των στοιχείων και μονωτική ταινία για την στήριξη τους.

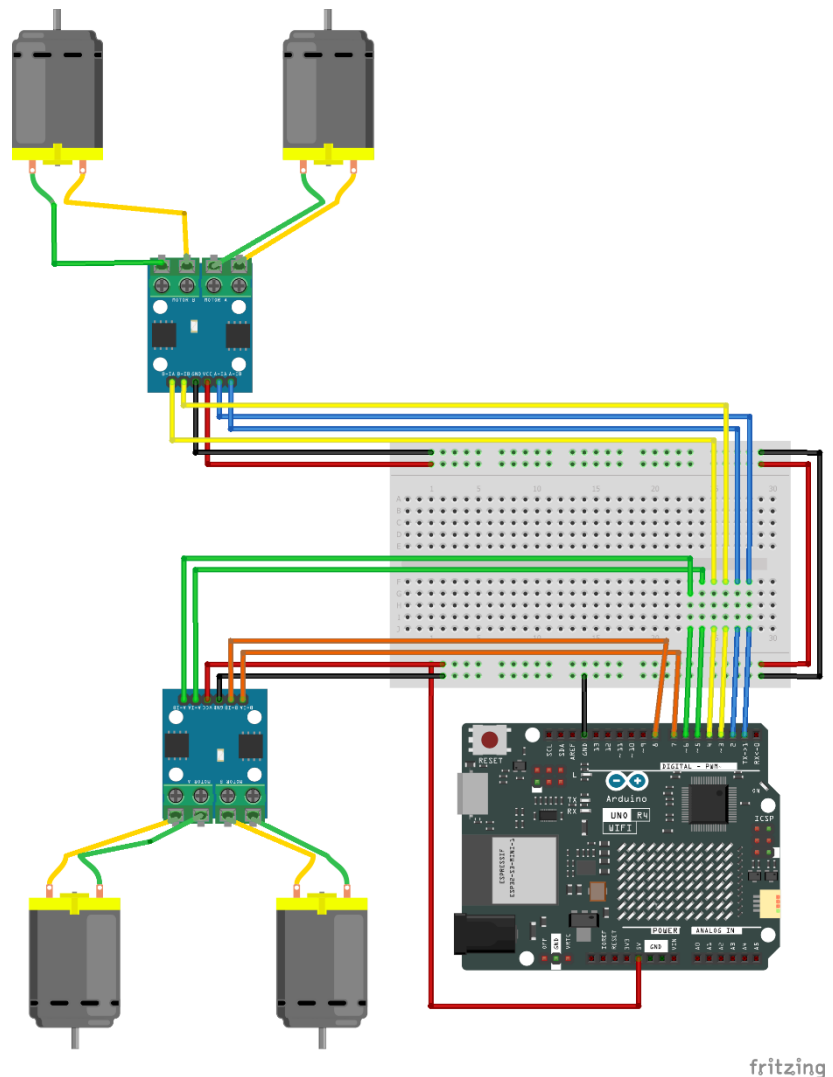
5 Ανάπτυξη έργου

5.1 Γενικά

Αρχικά, συνδέω τα **τρισδιάστατα εκτυπωμένα** μέρη της κατασκευής και το power bank, σχηματίζοντας τον σκελετό του οχήματος. Στη συνέχεια, τοποθετώ τους DC motors, τα L9110S H-Shields και την πλακέτα πάνω στην κατασκευή, με τις βίδες που έχουν παραχωρηθεί. Κατασκευάζω τα στηρίγματα των αισθητήρων και του servo motor και τέλος ασφαλίζω καλά το breadboard πάνω στην κατασκευή και συνδέω όλα τα παραπάνω στοιχεία μεταξύ τους.

Ως **βασική κατασκευή** οι συνδεσμολογίες γίνονται ως εξής:

- Συνδέω τις ράγες ισχύος του breadboard το ένα με το **Ground(γείωση)*** το άλλο με τα 5 Volt της πλακέτας δεξιά και αριστερά.
- Αντίστοιχα συνδέω το VCC και GRD των L9110S στην τάση και στη γείωση. Στο δεξί shield συνδέω τα AIB1, AIA1, BIB1, BIA1 με τα pin της πλακέτας 1-4 και στο αριστερό με τον ίδιο τρόπο τα AIB2, AIA2, BIB2, BIA2 με τα pin 5-8.
- Τέλος τις άκρες κάθε τροχού με τις εξωτερικές υποδοχές των L9110S.



fritzing

Εικόνα 16. Πρώμο στάδιο κατασκευής

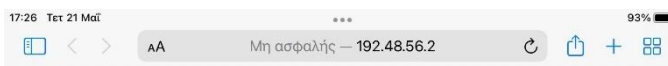
5.2 Ξεκίνημα – Version 0.8

5.2.1 Γενικά

Η εργασία μου ξεκίνησε σαν ένα απλό πρόγραμμα Arduino. Κάθε πρόγραμμα **Arduino sketch** (σχέδιο) περιλαμβάνει μια συνάρτηση **setup** που εκτελείται πάντα μια φορά στην αρχή και μια συνάρτηση **loop** που επαναλαμβάνεται συνεχώς όσο λειτουργεί το Arduino. Αρχικά με το **#include** συμπεριλαμβάνω ποιες έξτρα βιβλιοθήκες θέλω να φορτώσω στο πρόγραμμά μου, στη περίπτωση αυτή θέλω την **WiFiS3** για την δημιουργία **Wi-Fi Access point**. Ορίζω τις τιμές των εντολών μεταξύ client-server, τα **pin*** που αντιστοιχούν στις συνδέσεις με τα L9110S, το όνομα και τον κωδικό του access point καθώς και ποια θύρα του server χρησιμοποιώ για επικοινωνία με τον client.

5.2.2 Setup()

Στη συνάρτηση **setup** με την εντολή **«Serial.begin(9600);»** κατοχυρώνω επικοινωνία μεταξύ του Arduino και του υπολογιστή με ταχύτητα **9600 bps*** έτσι ώστε να μπορώ να δω τα αποτελέσματα που μου δίνει η πλακέτα, με την εντολή **«pinMode»** δηλώνω τα pin του Arduino ως εξόδους ώστε να παρέχουν ενέργεια στους κινητήρες, δημιουργώ το access point με μια σταθερή **IP* 192.48.56.2** και αφού έχω ελέγξει ότι έχει δημιουργηθεί σωστά το point και είναι εγκατεστημένο το τελευταίο firmware ξεκινάω τη λειτουργία του server. Για να συνδεθώ στην πλακέτα μου απλά χρειάζεται να συνδεθώ στο σωστό Wi-Fi δίκτυο και μετά σε ένα browser να πληκτρολογήσω την παραπάνω διεύθυνση IP.



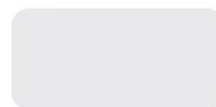
Click [here](#) to move forward

Click [here](#) to move backwards

Click [here](#) to stop

Click [here](#) to turn right

Click [here](#) to turn left



Εικόνα 17. Πρώιμο HTML

5.2.3 Loop()

Στη συνάρτηση **loop**, που επαναλαμβάνεται συνεχώς, εξετάζω αν υπάρχει χρήστης που να είναι συνδεδεμένος. Στην περίπτωση αυτή με την εντολή **«client.println();»** τυπώνω σειρά ανά σειρά την σελίδα HTML στο χρήστη μέχρι να τυπώσω όλες τις γραμμές. [44] Αυτή η σελίδα εμπεριέχει link που αλλάζουν το header της σελίδας και ανάλογα με τι καταλήγει εκτελείται συγκεκριμένη κίνηση

του οχήματος έως ότου να κλείσει η σύνδεση με το client. [45] Το πως θα κινείται ορίζεται μέσα από συναρτήσεις που έχω φτιάξει και είναι προσαρμοσμένες για το όχημα μου.

5.2.4 Ανασκόπηση

Έτσι, παρόλο που έχει ολοκληρωθεί η βασική ιδέα της εργασίας μου, υπάρχει περιθώριο για περεταίρω εξέλιξη. Ο τρόπος κίνησης του οχήματος πρέπει να αλλάξει καθώς αν διακοπεί απρόοπτα η επικοινωνία με το client ή δεν πατηθεί το stop, το αμάξι δεν σταματάει την κίνηση του, ενώ πολλές φορές προβλήματα του http στο **header της σελίδας*** δημιουργούν προβλήματα στην σύνδεση, που σημαίνει ότι η επικοινωνία με το server πρέπει να αλλάξει και να γίνεται με WebSocket, για τους λόγους που έχω ήδη αναφέρει στην θεωρία μου. Επίσης χρειάζεται να τοποθετήσω και να χρησιμοποιήσω αισθητήρες για την βοήθεια στην κίνηση και τη συλλογή δεδομένων μου , πρέπει να αναπτυχθεί η αυτόματη κίνηση του οχήματος και φυσικά το περιβάλλον διεπαφής του χρήστη να γίνει πιο φιλικό προς αυτόν.

5.3 Ενσωμάτωση WebSocket - Version 1.3

5.3.1 Γενικά

Σε αυτή τη version έχουμε ολοκληρώσει την ενσωμάτωση του πρωτοκόλλου WebSocket για την επικοινωνία client - server με την χρήση της βιβλιοθήκης «**mWebSockets**» . Κάνοντας «**#include <WebSocketServer.h>**» και με την εντολή «**WebSocketServer WebSocket(81);**» φτιάχνουμε ένα αντικείμενο τύπου **WebSocketServer** που ελέγχει το πρωτόκολλο αυτό και του δίνουμε ως θύρα επικοινωνίας την θύρα 81 του Esp. [46] Επίσης ορίζουμε και κωδικό σύνδεσης με το access point για περισσότερη ασφάλεια.

5.3.2 Setup()

Στην συνέχεια, στη setup διαμορφώνω πως θα λειτουργεί το WebSocket, δηλαδή, όταν θα δέχεται κάποια εντολή κατά την σύνδεση, θα ελέγχει με εντολές **switch – case*** αν τα δεδομένα έχουν ληφθεί σωστά και εφόσον έχει γίνει αυτό, ποια συνάρτηση θα εκτελεί και τι θα κάνει στο κλείσιμο της σύνδεσης. Τέλος ενεργοποιεί αυτή τη λειτουργία με την εντολή «**websocket.begin();**» και τυπώνει στο Serial Monitor την πορεία του setup δηλαδή τις εντολές που εκτελούνται.

5.3.3 Loop()

Στη συνάρτηση loop, με την εντολή «**websocket.listen();**» εκκινείται ο διακομιστής WebSocket, επιτρέποντάς του να «**ακούει**» και να χειρίζεται τις εισερχόμενες συνδέσεις WebSocket από τους πελάτες. Αυτό είναι ένα κρίσιμο βήμα στη ρύθμιση ενός διακομιστή WebSocket, καθώς ενεργοποιεί τις

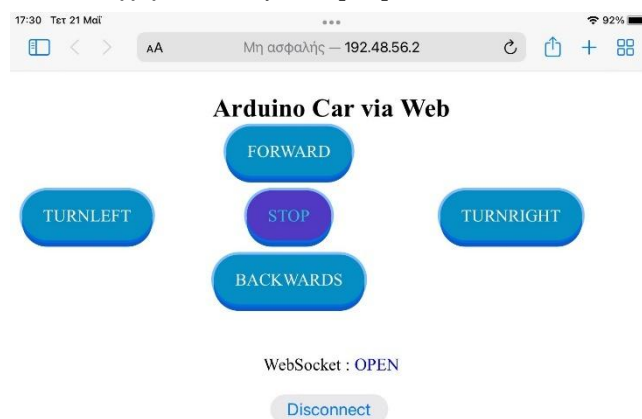
δυνατότητες ακρόασης του διακομιστή. Πλέον η σύνδεση με τον πελάτη αλλάζει. Ο ιστότοπος επικοινωνίας δεν τυπώνεται σειρά ανά σειρά, αλλά με την μια με την εντολή «**client.println(html);**» όπου το «html» είναι μια μεταβλητή τύπου string που περιέχει όλες τις γραμμές κώδικα HTML . [47]

5.3.4 HMTL Version 1.0

Η ανάπτυξη του κώδικα της σελίδας γίνεται με το **Studio Visual Code**. Πρώτα, σε ένα αρχείο .html φτιάχνω τον κώδικα της σελίδας. Περιέχει:

- **Κουμπιά** με τα οποία μπορώ να συνδεθώ στο server μέσω του πρωτοκόλλου WebSocket και να χειριστώ/οδηγήσω το αμάξι. [48]
- **JavaScript** κώδικα που ελέγχει την σύνδεση στο εξυπηρετητή αλλά και τον τρόπο λειτουργίας των κουμπιών. [49] [50]
- **Styles*** (στυλ) που αφορά την εμφάνιση των κουμπιών και γενικά της σελίδας.

Αυτό τον κώδικα τον αντιγράφω αυτούσιο σε ένα αρχείο «**index.h**» και αποτελεί τα δεδομένα της μεταβλητής με όνομα HTML_CONTENT που αποτελεί ένα πίνακα χαρακτήρων (**const char ***). Έπειτα, μετατρέπεται σε μια μεταβλητή τύπου **string** και εκτυπώνεται/εμφανίζεται στον χρήστη μέσα από την συνάρτηση loop.



Εικόνα 18. HMTL Version 1.0

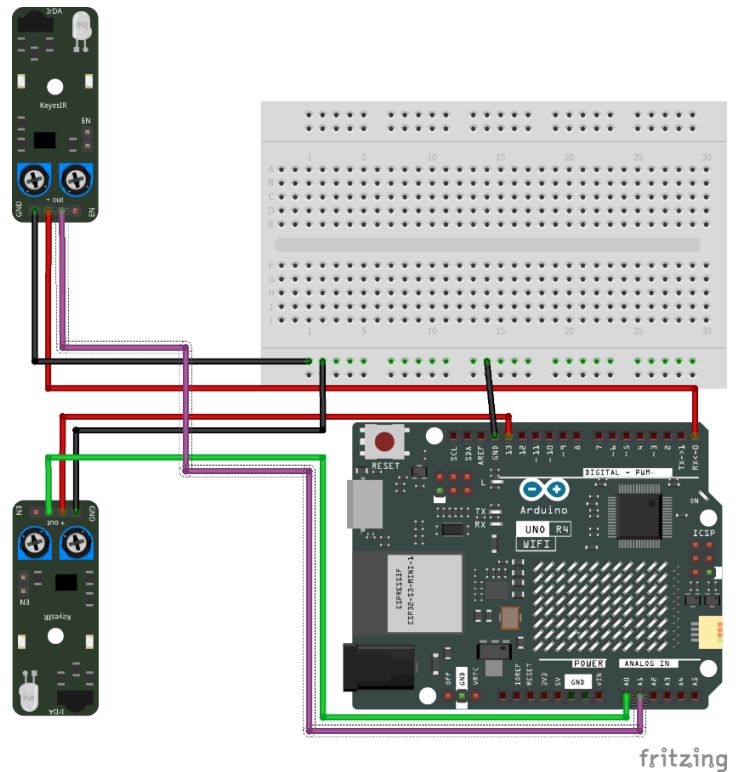
5.3.5 Ανασκόπηση

Με τη έκδοση αυτή το πρόγραμμα έχει ενσωματώσει τη λειτουργία των WebSocket και έχει οργανωθεί ένα λιτό και κατανοητό γραφικό περιβάλλον.

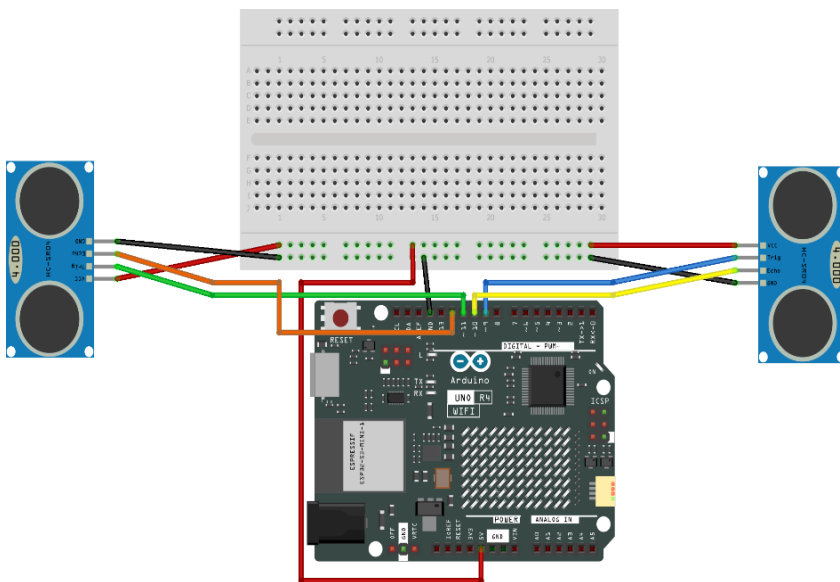
5.4 Ενσωμάτωση Αισθητήρων - Version 2.6

5.4.1 Γενικά

Οι πιο σημαντικές αλλαγές σε αυτή την έκδοση του προγράμματος μου είναι η διασύνδεση του Arduino με τους αισθητήρες υπερηχητικής απόστασης **HC-SR04** και τους αισθητήρες αποφυγής εμποδίων υπέρυθρων IR **KY-032**. Στους αισθητήρες υπέρυθρων KY-032 συνδέω πάλι τα VCC με τα 5 VOLT παροχή και το GRD με την γείωση και την αναλογική έξοδο τους με μια αναλογική είσοδο της πλακέτας αντίστοιχα. Οι συνδέσεις αυτές φαίνονται και στις εικόνες που έχω. Για τον κάθε αισθητήρα υπερήχων έχω συνδέσεις των trigger και echo pin ξεχωριστά το καθένα με ένα αντίστοιχο pin του Arduino και συνδέω τα VCC με τα 5 VOLT παροχή και το GRD με την γείωση.



Εικόνα 19. Infrared Sensor connections



Εικόνα 20. Ultrasonic Sensor connections

Αφού έχουν γίνει οι συνδέσεις πρέπει να φτιάξω τις συναρτήσεις με τις οποίες θα τσεκάρω την λειτουργία τους και θα μπορώ να λάβω δεδομένα από τους αισθητήρες. Με τα δεδομένα αυτά μπορώ να ελέγχω και να βοηθώ το χρήστη στην κίνηση. Με τη

συνάρτηση `checkDistanceUltrasonic(int trigP, int echoP)`, που δέχεται σαν ορίσματα τα trigger echo pin του αισθητήρα που λαμβάνω δεδομένα, υπολογίζω την απόσταση που έχει από την επιφάνεια απέναντι του. Η βασική αρχή της λειτουργίας είναι ότι χρησιμοποιώντας την ενεργοποίηση εισόδου/εξόδου (IO) για σήμα υψηλού επιπέδου τουλάχιστον 10us* στέλνει αυτόματα οκτώ 40 kHz* και ανιχνεύει εάν υπάρχει **παλμικό σήμα*** επιστροφής. Αν το σήμα επιστρέψει, μέσω

υψηλού επιπέδου, ο χρόνος διάρκειας εισόδου/εξόδου υψηλής εξόδου είναι ο χρόνος από την αποστολή υπερήχων έως την επιστροφή. [51]

Απόσταση δοκιμής = (χρόνος υψηλού επιπέδου * ταχύτητα ήχου (340M/S) / 2

Η τιμή που επιστρέφει η απόσταση σε εκατοστά (cm) με τύπο float, δηλαδή περιέχει και δεκαδικά. Για τους αισθητήρες υπέρυθρων KY-032 φτιάχνω την συνάρτηση **IR_output(int output)**, που παίρνει σαν όρισμα το pin του αισθητήρα και επιστρέφει αναλογικές τιμές από 0 μέχρι 1024. Εάν το εκπεμπόμενο **υπέρυθρο*** φως χτυπήσει ένα εμπόδιο, το φως θα ανακλαστεί και θα ανιχνευθεί από τη φωτοδίοδο, το 1024 αλλιώς όσο πιο μακριά είναι πλησιάζει το 0. Η εμβέλεια ανίχνευσης μπορεί να ρυθμιστεί από τους 2 ελεγκτές πάνω στην μονάδα. [52]
Πρακτικά για το σήμα του αισθητήρα :

Ψηφιακή κατάσταση -> Ενεργοποιημένο / Απενεργοποιημένο

5.4.2 Αισθητήρες στην κίνηση

Με την ανάπτυξη των λειτουργιών αυτών μπορώ να εξελίξω την κίνηση μου με νέες συναρτήσεις. Χρησιμοποιώντας τις βασικές συναρτήσεις κίνησης και ελέγχοντας την απόσταση αντικειμένων από το όχημα, μπορώ να το σταματήσω στην επιθυμητή απόσταση. Η κίνηση εμπρός και πίσω δουλεύει με τους ultrasonic που τσεκάρουν την απόσταση του οχήματος από τα αντικείμενα να είναι μεγαλύτερη από 20 εκατοστά. Αν η απόσταση είναι μικρότερη από 20 εκατοστά το τυπώνει στο Serial Monitor και δεν αφήνει το αμάξι να προχωρήσει. Με παρόμοιο τρόπο λειτουργεί η στροφή δεξιά και αριστερά αξιοποιώντας τους υπέρυθρους. Επειδή αυτοί δεν επιστρέφουν απόσταση, αλλά ένα ναι ή όχι, ρυθμίζω χειροκίνητα πάνω στην κάθε μονάδα να ενεργοποιούνται περίπου στα 20 εκατοστά. Έτσι έχω ολοκλήρωση την κίνηση με την βοήθεια αισθητήρων.

5.4.3 Λοιπές αλλαγές

Εκτός του προγράμματος μου έχω προσθέσει μερικά πράγματα ακόμα. Αυτά είναι Global variables δηλαδή καθολικές μεταβλητές, που περιλαμβάνουν την κατάσταση της κίνησης του οχήματος καθώς και μεταβλητές για την καλύτερη εκτύπωση στο Serial Monitor. Αξιοποιώντας αυτά συντάσσω μια νέα συνάρτηση την **move(int command)** που ελέγχει ποιες συναρτήσεις εκτελούνται όταν δέχεται η πλακέτα εντολή μέσω WebSocket.

5.4.4 Setup()

Με αυτές τις αλλαγές μεταβάλλεται ο τρόπος λειτουργίας της setup. Καταρχάς, είναι αναγκαίο να σετάρω τα pin της πλακέτας ως εισόδους/εξόδους έτσι ώστε να λειτουργούν οι αισθητήρες και να συλλέγουμε δεδομένα. Πλέον η συνάρτηση **ws.onMessage** της **websocket.onConnection** απλά δέχεται δεδομένα από το WebSocket μετά τη σύνδεση και δεν καθορίζει την κίνηση του

οχήματος καθώς, όπως προανέφερα, την ορίζει άλλη συνάρτηση. Μοναδική λειτουργία της είναι να λάβει δεδομένα και να ανανεώσει την καθολική μεταβλητή **command** τύπου **integer***, που χρησιμοποιεί το σύστημα μου για την κίνηση. Στην περίπτωση που το WebSocket δεν λάβει σωστά τα δεδομένα ή λάβει δεδομένα που δεν αντιστοιχούν σε ένα τύπο κίνησης από αυτούς που έχουμε ορίσει, τότε η καθολική μεταβλητή μου παίρνει την τιμή «0» που αντιστοιχεί στο stop.

5.4.5 Loop()

Πρωταρχική αλλαγή στη συνάρτηση loop είναι ότι με κάθε επανάληψη της «τρέχει» η συνάρτηση **move(int command)** που μεταβάλλει την κίνηση του οχήματος αναλόγως την τιμή της καθολικής μεταβλητής **command**. Επίσης έχουμε κάποιες βελτιώσεις στο κώδικα του arduino για την εκτύπωση της σελίδας HTML στο χρήστη, χωρίς να έχουμε κάνει αλλαγές στο κώδικα της σελίδας.

5.4.6 Ανασκόπηση

Συμπερασματικά, σε αυτή την έκδοση της εργασίας μου, έχει γίνει η ενσωμάτωση αισθητήρων στην κατασκευή καθώς και η ένταξη τους στην συλλογή δεδομένων για την μεταβολή του τρόπου κίνησης ου αμαξιδίου. Παρόλα αυτά, αυτό τώρα μετακινείται μόνο με την χρήση αισθητήρων και δεν έχει την δυνατότητα να μπορεί να παραβλέψει τα αποτελέσματα τους και να λειτουργήσει εντελώς χειροκίνητα. Αυτή την εναλλαγή θα την ενεργοποιήσω σε επόμενη version μαζί με άλλες βελτιώσεις.

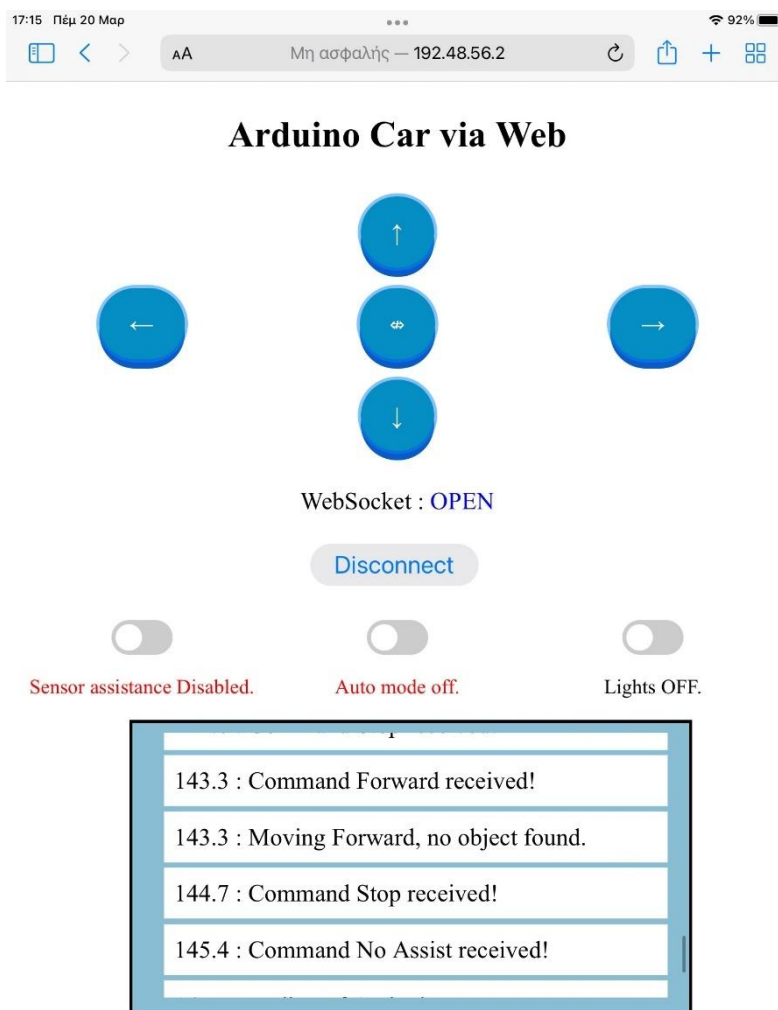
5.5 Ολοκλήρωση σημαντικών αναβαθμίσεων – Αμφίδρομη επικοινωνία – Version 3.8.5

5.5.1 Γενικά

Η version 3.8.5 έχει φέρει πολλές αλλαγές στην λειτουργία του συστήματος μου. Αρχικά, έχω συνδέσει αμφίδρομα το server μου με τη σελίδα HTML που σερβίρει στο χρήστη και έχω οριστικοποιήσει την σύνδεση του servo κινητήρα μου καθώς και τις εντολές και βιβλιοθήκες που χρειάζεται για την στρέψη του. Πλέον, μπορώ ως χρήστης να διαχωρίζω στην κίνηση που κάνει αν θα ενεργοποιεί και θα χρησιμοποιεί τους αισθητήρες και έχω θέσει τις βάσεις για την ανάπτυξη αυτόματης κίνησης σε επόμενη έκδοση. Είναι σημαντικό να αναφερθώ και στην προσθήκη LED και στην ενεργοποίηση του on board led matrix του arduino, που ανάβουν διαφορετικά αναλόγως το είδος κίνησης. Τέλος, έχω προσθέσει ακόμα **καθολικές μεταβλητές*** για την λειτουργία των παραπάνω και για την καλύτερη εκτύπωση της κατάστασης του arduino τόσο στην δικιά του κονσόλα όσο και στην κονσόλα της ιστοσελίδας.

5.5.2 Αμφίδρομη επικοινωνία - web_me3

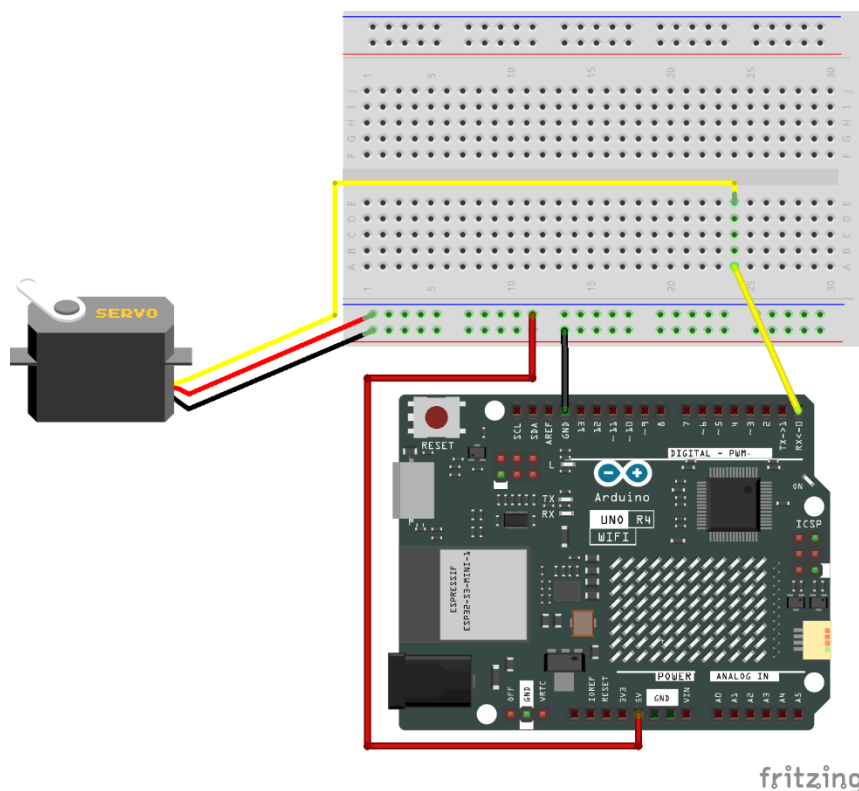
Στις προηγούμενες εκδόσεις της εφαρμογής μου ο ενσωματωμένος διαδικτυακός εξυπηρετητής δεχόταν από τον κάθε πελάτη εντολές τις οποίες εκτελούσε. Όμως με την ανάπτυξη της συνάρτησης **web_me3(const char *message)**, όπου 3 η τρίτη εκδοχή της web_me, μπορεί ο εξυπηρετητής να «μιλάει» στον πελάτη, δηλαδή να στέλνει και αυτός μήνυμα στο πελάτη, που θα εμφανίζονται σε ένα παράθυρο, παρόμοιο με αυτό του **Serial Monitor*** του Arduino IDE και φυσικά το εκτυπώνει και στο IDE προς διευκόλυνση μου. Την έφτιαξα για να μπορεί να βλέπω ως ο χρήστης, τι ακριβώς εντολές εκτελεί η πλακέτα μου όταν δεν είναι συνδεδεμένο στο IDE. Καθώς το όχημα μου προορίζεται για να κινείται σε μεγάλες αποστάσεις είτε χειροκίνητα είτε αυτόματα οι περιπτώσεις που δεν θα είναι συνδεδεμένο σε κάποιο υπολογιστή ξεπερνούν κατά χιλιάδες αυτές που θα είναι συνδεδεμένο. Λειτουργεί παίρνοντας σαν όρισμα ένα πίνακα χαρακτήρων (**const char *message**) τον οποίο μετατρέπει στην κατάλληλη μορφή και τύπο, το στέλνει μέσω της **websocket.broadcast(const WebSocket::DataType dataType, const char *message, uint16_t length)**; με WebSocket δηλαδή, και αυτό εμφανίζεται σε ένα πλαίσιο στη σελίδα. Στην περίπτωση κάποιου απλού μηνύματος, στέλνει το μήνυμα μαζί με το χρόνο που έχει περάσει από την εκκίνηση της πλακέτας μου, αξιοποιώντας την συνάρτηση **millis()**; που επιστρέφει αυτή την τιμή σε τύπο **double*** για να μην έχω πολλά δεκαδικά, ενώ στην περίπτωση που είναι κενό το μήνυμα, στέλνει αυτό το κενό. Αυτό το αξιοποιώ αρκετές φορές σε άλλες συναρτήσεις έτσι ώστε να στέλνονται με την σωστή σειρά τα μηνύματα και στην σωστή στιγμή. Πρέπει να αναφέρω ότι γίνεται χρήση της βιβλιοθήκης «**string**» στην μετατροπή των δεδομένων σε κατάλληλη μορφή ώστε να μπορώ να τα στείλω μέσω WebSocket.



Εικόνα 21. HTML Version 1.5

5.5.3 Servo motor συνδέσεις

Στον servo κινητήρα από τα τρία καλώδια που έχει συνδέω το κόκκινο στην τάση 5V, το μαύρο στην γείωση και το κίτρινο στο Pin 0 του arduino. Με τη χρήση της κατάλληλης βιβλιοθήκης και καθολικής μεταβλητής «**angle**» τύπου **integer** συνδέω την θύρα 0 ώστε να μπορώ να ελέγχω την γωνία που είναι στραμμένος ο κινητήρας. Ορίζω με το **#define** τιμές για τις νέες εντολές του servo και φτιάχνω τη συνάρτηση **turnservo(int n)**, που μεταβάλλοντας την τιμή της **angle** αλλάζει και την γωνία του καθώς και την **angle_turn()**



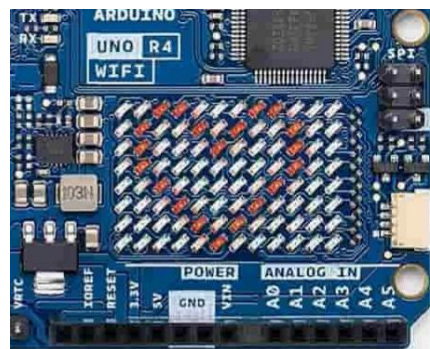
Εικόνα 22. Σύνδεση Servo Motor

που ευθύνεται για τον έλεγχο της αλλαγής γωνίας λαμβάνοντας υπόψη την τιμή της καθολικής μεταβλητής **command**. Με ένα **switch-case statement** μπορεί να γυρίσει τον κινητήρα κατευθείαν σε 0, 90 και 180 μοίρες καθώς και να το περιστρέψει λίγο λίγο από 0 σε 180 και αντίστροφα. Μετά από κάθε κίνηση στέλνει μέσω της **web_me3** ένα ειδικό μήνυμα που περιέχει την τιμή της γωνίας. Η **web_me3** καταλαβαίνει αυτό το μήνυμα και το στέλνει με διαφορετικό τρόπο από ότι στέλνει τα άλλα μηνύματα και παρόμοια ανανεώνεται το HTML σε συγκεκριμένο σημείο.

5.5.4 Arduino UNO R4 Wi-Fi LED Matrix

Το Arduino UNO R4 Wi-Fi διαθέτει ενσωματωμένο πίνακα LED 12x8, ο οποίος είναι διαθέσιμος για προγραμματισμό για την εμφάνιση γραφικών, κινούμενων σχεδίων, τη λειτουργία διεπαφής ή ακόμα και για την αναπαραγωγή παιχνιδιών. Τον χρησιμοποιώ να μπορώ να καταλάβω στο περίπου τα στάδια της λειτουργίας του Arduino σε περίπτωση που δεν υπάρχει κάποιος άλλος τρόπος. Για να χρησιμοποιήσω το LED Matrix, πρέπει να συμπεριλάβω τη βιβλιοθήκη στην κορυφή του σκίτσου «**Arduino_LED_Matrix.h**» και να δημιουργήσω ένα αντικείμενο LED Matrix στο σκίτσο με την εντολή «**ArduinoLEDMatrix matrix;**». Η βιβλιοθήκη LED Matrix για το UNO R4 Wi-Fi λειτουργεί με βάση την αρχή της δημιουργίας ενός **frame** και στη συνέχεια της φόρτωσής του σε ένα **buffer*** που

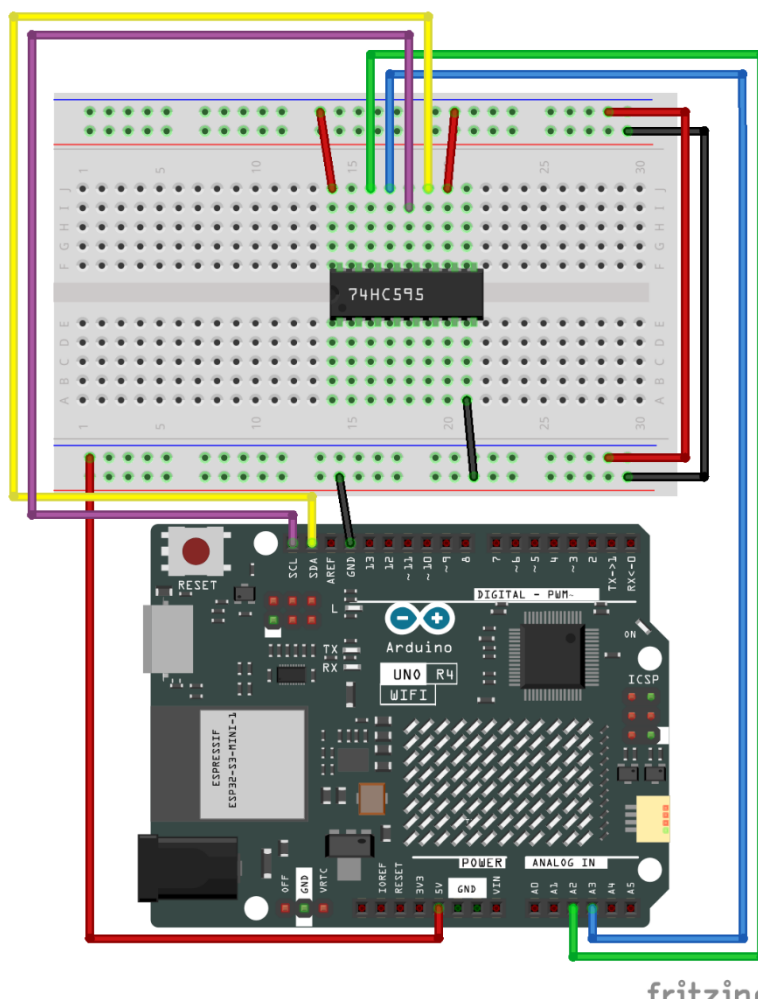
εμφανίζει το frame. Ένα **frame** είναι η "εικόνα" που εμφανίζεται σε οποιαδήποτε δεδομένη στιγμή στο matrix. Εάν μια κινούμενη εικόνα είναι μια σειρά εικόνων, ένα frame είναι μία από αυτές τις εικόνες της σειράς. Κατά την εκκίνηση του arduino παίζει ένα έτοιμο animation ενώ για τις άλλες λειτουργίες του χρησιμοποιώ **frames** που έχω φτιάξει. Έχω ένα κύριο **byte frame[8][12]** που με την συνάρτηση **frame_change(byte arr[8][12])** του δίνω τιμές των frame που έχω φτιάξει και την **matrix_move(byte arr[8][12])** τις αλλάζω αναλόγως τις ανάγκες μου. Σε κάθε custom frame το «0» συμβολίζει κλειστό λαμπάκι και το «1» ανοιχτό. [53]



Εικόνα 23. Arduino Uno R4 Led Matrix

5.5.5 Διαχείριση LED μέσω Shift Register

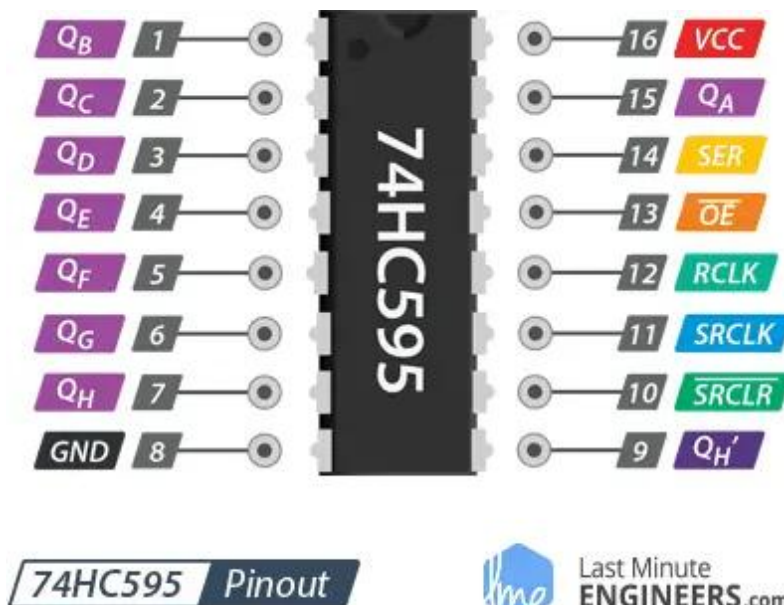
Για την καλύτερη παρουσίαση του οχήματος και να το φτάσω πιο κοντά στο αληθινό θέλω να του προσθέσω λαμπάκια τύπου Led, όμως παρατηρώ ότι οι θύρες που μένουν στο Arduino δεν φτάνουν για συνδεθούν όλα και να τα χειρίζομαι ξεχωριστά. Έτσι κάνω χρήση ενός στοιχείου Shift Register



Εικόνα 24. Shift Register 74HC595 Circuit

(καταχωρητές μετατόπισης) τύπου 74HC595 για να ελέγχω 8 εξόδους που σε κάθε μια είναι συνδεδεμένο ένα λαμπάκι. Ο τρόπος με τον οποίο λειτουργούν τέτοια στοιχεία είναι μέσω κάτι που ονομάζεται «σύγχρονη σειριακή επικοινωνία», δηλαδή παλμογραφώ* έναν ακροδέκτη πάνω και κάτω, μεταδίδοντας έτσι ένα byte δεδομένων στον καταχωρητή bit προς bit. Παλμογραφώντας τον δεύτερο ακροδέκτη, τον ακροδέκτη ρολογιού, οριοθετώ τα bit. Αυτό

έρχεται σε αντίθεση με τη χρήση της «ασύγχρονης σειριακής επικοινωνίας» της συνάρτησης **Serial.begin()**, η οποία βασίζεται στον ανεξάρτητο καθορισμένο ρυθμό δεδομένων μεταξύ του αποστολέα και του δέκτη. Μόλις μεταδοθεί ολόκληρο το byte στον καταχωρητή, τα μηνύματα HIGH ή LOW που περιέχονται σε κάθε bit κατανέμονται σε κάθε έναν από τους μεμονωμένους

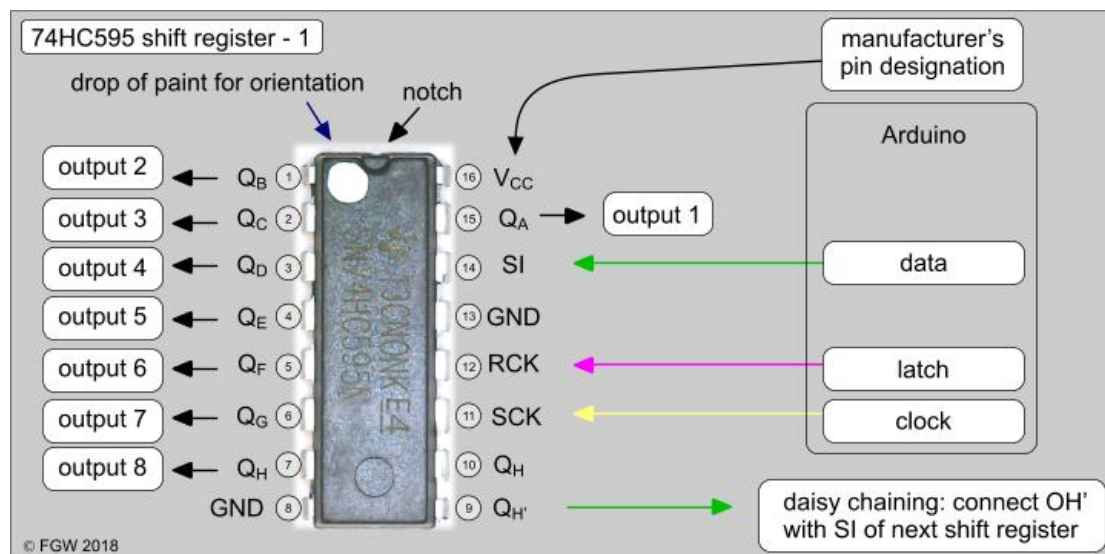


Εικόνα 25. 74HC595 Pinout

ακροδέκτες εξόδου. Αυτό είναι το μέρος της «παράλληλης εξόδου», όπου όλοι οι ακροδέκτες κάνουν αυτό που θέλετε να κάνουν ταυτόχρονα. Το μέρος της «σειριακής εξόδου» αυτού του στοιχείου προέρχεται από τον επιπλέον ακροδέκτη του, ο οποίος μπορεί να περάσει τις σειριακές πληροφορίες που λαμβάνονται από τον μικροελεγκτή χωρίς αλλαγές. Αυτό σημαίνει ότι μπορείτε να μεταδώσω 16 **bit*** στη σειρά (2 byte) και τα πρώτα 8 θα ρέουν μέσω του πρώτου καταχωρητή στον δεύτερο καταχωρητή και θα εκφραστούν εκεί. Αυτό είναι σημαντικό και θα το χρησιμοποιήσω σε επόμενη version. Οι "3 καταστάσεις" αναφέρονται στο γεγονός ότι μπορώ να ορίσω τις ακίδες εξόδου είτε ως υψηλή, χαμηλή είτε ως "υψηλή σύνθετη αντίσταση". Σε αντίθεση με τις καταστάσεις ΥΨΗΛΗ και ΧΑΜΗΛΗ, δεν μπορώ να ορίσω τις ακίδες στην κατάσταση υψηλής σύνθετης αντίστασης ξεχωριστά. [54]

Συγκεκριμένα, συνδέω τα «**dataPin**» με το «**A2**», το «**enable**» με το «**A3**» το «**latchPin**» με το «**SDA(A4)**» και το «**clockPin**» με το «**SCL(A5)**». Ο λογικός πίνακας, που βρίσκεται στο datasheet που έχω ήδη αναφέρει, είναι αυτός που λέει ότι ουσιαστικά όλα τα σημαντικά συμβαίνουν σε μια ανοδική τάση. Όταν το clockPin μεταβαίνει από χαμηλό σε υψηλό, ο καταχωρητής ολίσθησης διαβάζει την κατάσταση του pin δεδομένων. Καθώς τα δεδομένα μετατοπίζονται, αποθηκεύονται σε έναν εσωτερικό καταχωρητή μνήμης. Όταν το latchPin μεταβαίνει από χαμηλό σε υψηλό, τα αποσταλμένα δεδομένα μετακινούνται από τον προαναφερθέντα καταχωρητή μνήμης στους pin εξόδου, ανάβοντας τα LED. Τέλος, με το enable μπορώ να ανοιγοκλείνω το στοιχείο αυτό σημαίνοντας ότι όταν το enable είναι «HIGH» το στοιχείο κλείνει ενώ όταν είναι «LOW» το στοιχείο λειτουργεί. [55] Όλα τα παραπάνω τα έχω οργανώσει στη συνάρτηση **shift_send(int numberToDisplay)** με την οποία ενεργοποιώ και απενεργοποιώ τα

λαμπάκια αναλόγως με τον αριθμό που βάζω. Οι αριθμοί είναι στην μορφή «**B10101011**» που είναι δυαδική αναπαράσταση δεκαδικού αριθμού και κάθε μια από τις 8 θέσεις μπορεί να είναι 1 ή 0, δηλαδή ανοιχτό/κλειστό. Για παράδειγμα η εντολή **shift_send(B11111111);** ενεργοποιεί όλα τα φώτα.



Εικόνα 26. 74HC595 Circuit

Με την συνάρτηση λοιπόν **lights()**, μέσω ενός switch-case, χρησιμοποιώ τα παραπάνω τόσο για να αλλάξω την κατάσταση στα Led όσο και για να ενημερώσω τον χρήστη για τη λειτουργία του συστήματος. Αξιοποιώντας τις καθολικές μεταβλητές ενεργοποιώ και απενεργοποιώ το Shift Register και τυπώνω οργανωμένα στα log windows, δηλαδή έχω κάποιες καθολικές μεταβλητές τύπου «**Boolean**»* που λειτουργούν ως «**flags**»* για να μην τυπώνονται κάθε φορά που τρέχει το loop.

5.5.6 Ανανέωση συνάρτηση μετακίνησης οχήματος

Πλέον η function **move()** δίνει την δυνατότητα να μετακινηθεί το όχημα με ή χωρίς την βοήθεια των αισθητήρων. Έχω δημιουργήσει ξεχωριστές συναρτήσεις για το κάθε είδος κίνησης, για παράδειγμα **forward_assist()** και **forward_no_assist()** τις οποίες εμπεριέχουν άλλες συναρτήσεις αναλόγως το είδος τους. Με ένα if-else statement γίνεται η επιλογή ανάμεσα στο αν θα τους χρησιμοποιήσω ή όχι και θα τρέξει είτε η **move_no_assist()**, χωρίς βοήθεια, είτε η **move_assist()**, με βοήθεια. Κάθε μια από αυτές λειτουργεί με switch-case και με global variables που αντιστοιχούν σε «σημαίες» τυπώνει οργανωμένα τα αποτελέσματα και ελέγχει την παροχή ενέργειας στους sensors.

5.5.7 Setup()

Κατά το setup σετάρω σωστά τα επιπλέον pin που χρησιμοποιώ για την λειτουργία του shift register, την οργάνωση του πίνακα με φωτάκια της πλακέτας, εκτυπώνω με πιο γρήγορο τρόπο τα Serial.print(); και με την web_me3

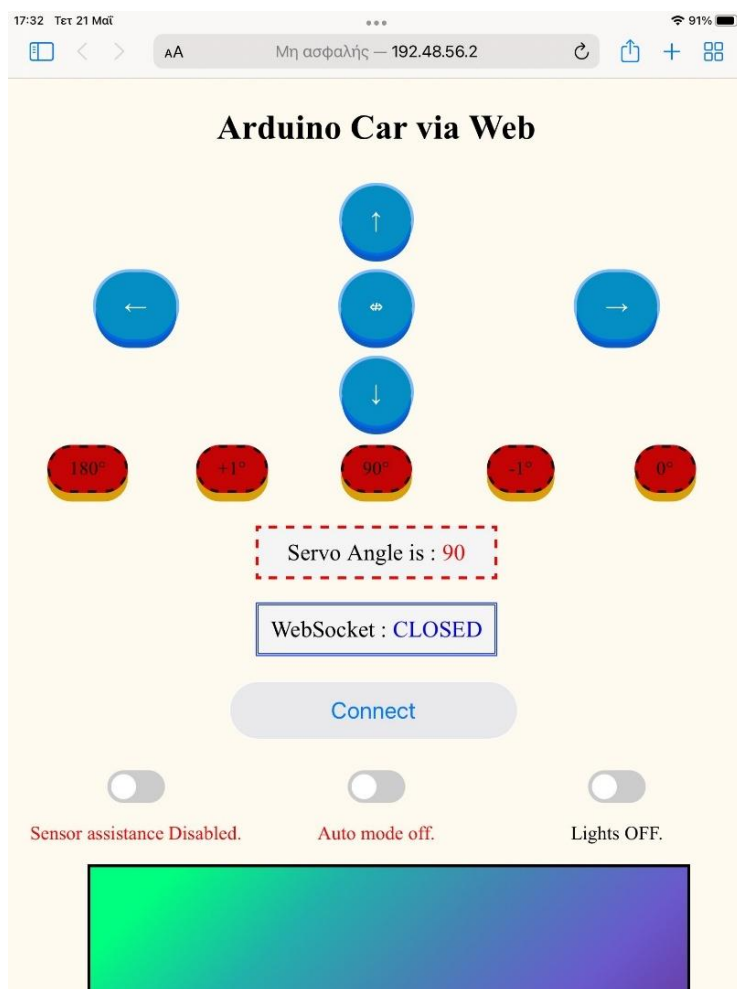
ενημερώνω για την σύνδεση του το χρήστη. Στην λειτουργία **ws.onMessage(WebSocket &ws, const WebSocket::DataType dataType, const char *message, uint16_t length)** προσθέτω τις επιπλέον εντολές που έφτιαξα και επιλέγοντας την web_me3 του γνωστοποιώ την επιλογή εντολής. Τελικά, όταν εκτελείται, κατά το κλείσιμο της σύνδεσης, η **ws.onClose();** επαναφέρω τις τιμές όλων των μεταβλητών.

5.5.8 Loop()

Στο loop οι βασικές αλλαγές είναι στην εκτέλεση της συνάρτησης ,που συνεχίζει να ελέγχει την μετακίνηση του αμαξιδίου, της συνάρτησης , που ελέγχει τα φώτα και της που ορίζει την λειτουργία του servo κινητήρα. Σημαντική αναβάθμιση έχουμε και στην παράδοση/εκτύπωση της σελίδας στο client. Από τύπο **const char *** μετατρέπεται σε τύπο **const char * PROGMEM** που σημαίνει ότι τώρα η μεταβλητή που περιέχει τον HTML κώδικα διατηρείται σταθερά στη μνήμη flash (προγράμματος), αντί να αντιγράφεται σε **SRAM*** κατά την εκκίνηση του προγράμματος. Η λέξη-κλειδί **PROGMEM** είναι ένας τροποποιητής μεταβλητών, και χρησιμοποιείται μόνο με τους τύπους δεδομένων που ορίζονται στο «**pgmspace.h**». [56] Επιπλέον, στην εντολή **client.println(F(HTML_CONTENT));** που εκτυπώνει το HTML στο πελάτη, η εντολή **F(...)** είναι μια **μακροεντολή Arduino*** που χρησιμοποιείται για την αποθήκευση συμβολοσειρών σε μνήμη flash (PROGMEM) αντί για RAM, κάτι που περιορίζεται στους μικροελεγκτές. Χωρίς **F()**, η **HTML_CONTENT** θα αντιγραφόταν στη RAM πριν εκτυπωθεί, ενώ με την **F()**, παραμένει στη μνήμη flash, εξοικονομώντας πολύτιμη μνήμη RAM. [57]

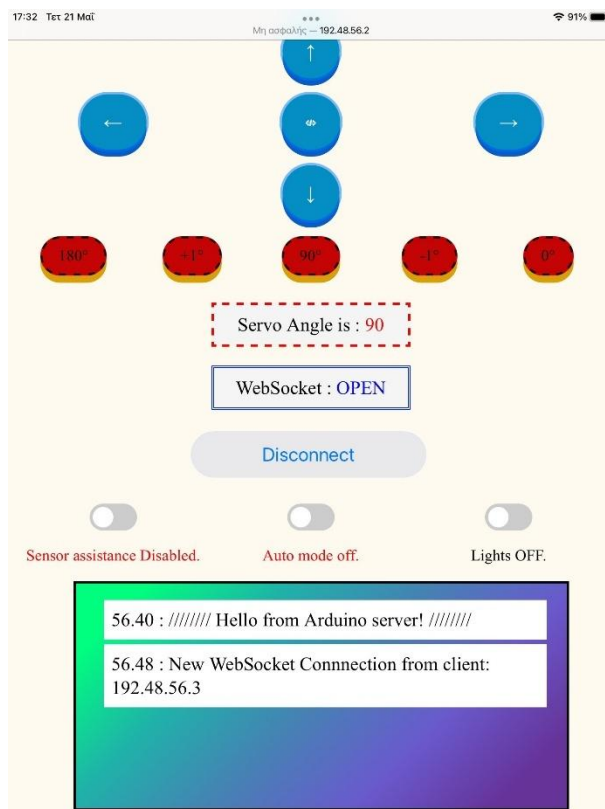
5.5.9 HTML Version 2.0

Υπάρχει νέα σελίδα που ενσωματώνει το πρόγραμμα μου. Πρώτον, η σελίδα είναι πιο καλά οργανωμένη και συμπεριφέρεται πιο responsive σε κινητές συσκευές. Έχω προσθέσει καινούργια κουμπιά που ελέγχουν το servo κινητήρα καθώς και κουτάκι που τυπώνει τις μοίρες που είναι στραμμένο. Έφτιαξα κάποια **toggle buttons*** τα οποία ορίζουν αν η κίνηση θα γίνεται ή όχι

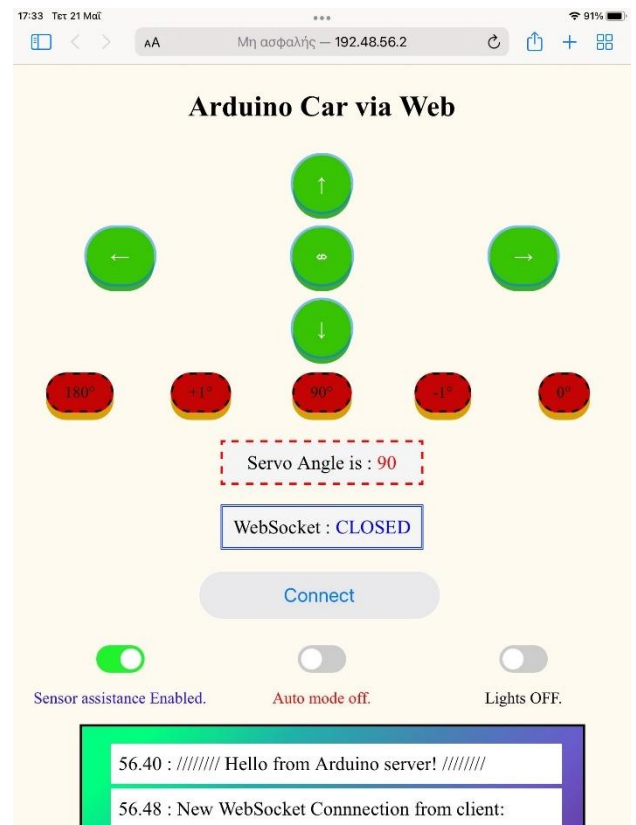


Εικόνα 27. HTML Version 2.0

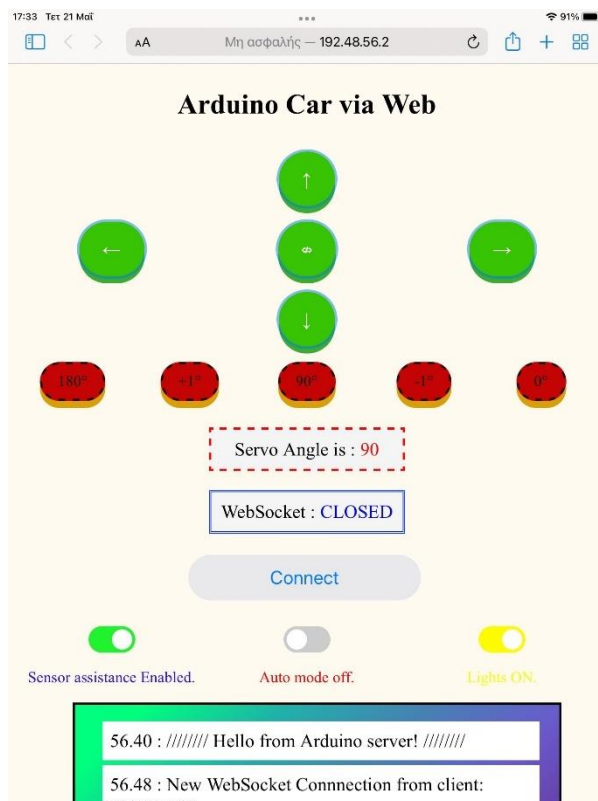
με χρήση αισθητήρων, αν θα ενεργοποιείται η αυτόματη μετακίνηση(not implemented yet - wip) και αν θα έχω τα φωτάκια αναμμένα. Στο κάτω μέρος της σελίδας υπάρχει ένα παράθυρο κύλισης που φαίνονται τα **console logs*** (αρχεία καταγραφής κονσόλας) που έρχονται από το arduino. Μέσω JavaScript διαφοροποιώ αν τα μηνύματα έχουν έρθει σωστά, αν είναι κενά, αν προορίζονται για αυτό το παράθυρο ή αν αφορούν μεταβολές στις μοίρες του σερβοκινητήρα. Τέλος, με τα διάφορα styles του HTML βελτιώνω την λειτουργία των κουμπιών, των κουμπιών εναλλαγής, του console παραθύρου και γενικά την εμφάνιση της σελίδας.



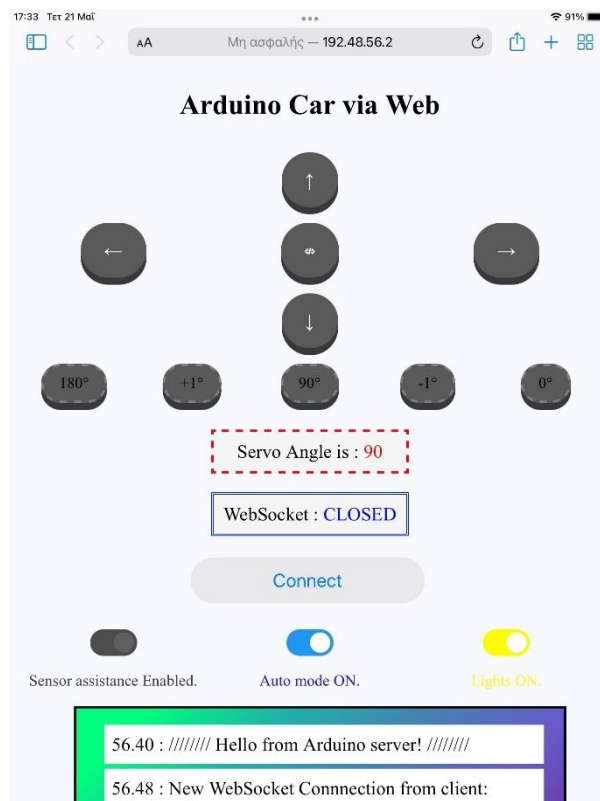
Εικόνα 29. HTML Connected



Εικόνα 28. HTML Assisted Move initialized



Εικόνα 30. Lights On



Εικόνα 31. Auto Move Initialized(WIP)

5.6 Τελική έκδοση – Version 4.5.6

5.6.1 Γενικά

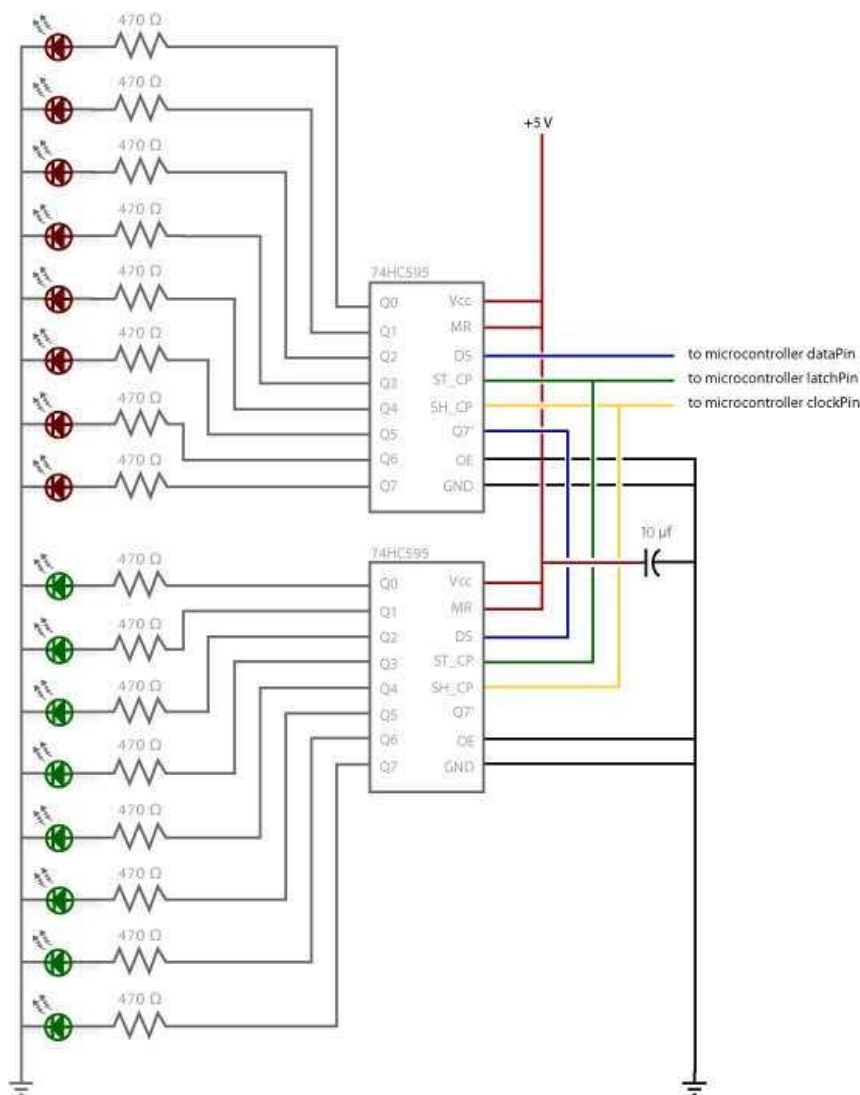
Στην έκδοση αυτή, έχω κάνει πολλές αλλαγές στο υλικό κομμάτι και ολοκλήρωσα ορισμένες λειτουργίες που πιστεύω έπρεπε να γίνουν. Καταρχάς, έγινε αντικατάσταση των αισθητήρων αποφυγής εμποδίων υπέρυθρων **IR KY-032** δεξιά και αριστερά με τους αισθητήρες υπερηχητικής απόστασης **HC-SR04**, προστέθηκαν οι αισθητήρες υπέρυθρων μέτρησης περιστροφής τύπου «**HR0172**» για μέτρηση στροφών των DC κινητήρων και με αυτό τον τρόπο, των βημάτων που κάνει το όχημα και ενσωματώθηκε η χρήση shift register για την παροχή ενέργειας στους νέους αισθητήρες αλλά και στα ήδη υπάρχον φωτάκια. Οριστικοποιήθηκε νέα δομή των συναρτήσεων κίνησης καθώς και νέα λειτουργία για τον έλεγχο της ταχύτητας της κίνησης. Μεγάλο μέρος της αναβάθμισης του κώδικα αυτή τη φορά είναι ένα βασικό στάδιο της αυτοματοποιημένης κίνησης και η αποθήκευση των δεδομένων αυτών σε δυναμικούς πίνακες τύπου «**vector**». Τα προηγούμενα αποτελέσματα όλων των λειτουργιών στέλνονται κατάλληλα στην σελίδα μου, όπου μπορώ να δω και να αλλάζω την ταχύτητα, τον μέσο όρο των βημάτων ενώ σε ένα χάρτη παρατηρώ την κίνηση του οχήματος.

5.6.2 Προσθήκη δεύτερου Shift Register – Dedicated Library

Επειδή οι θύρες σύνδεσης που έχει η πλακέτα μου είναι περιορισμένες και έχω ανάγκη από περισσότερες, χρησιμοποιώ καταχωρητή μετατόπισης. Ήδη η χρήση του προσθέτει 8 ακόμα θύρες για την λειτουργία των led. Όμως με την προσθήκη επιπλέον υπερηχητικών αισθητήρων και αισθητήρων καταμέτρησης περιστροφής τροχών θέλω να αυξήσω τις θύρες μου. Βασική, λοιπόν, είναι η σύνδεση δεύτερου καταχωρητή και η χρήση της βιβλιοθήκης

«**ShiftRegister74HC595**», καθώς με αυτή ο έλεγχος κατάστασης των εξόδων τους γίνεται μακράν πιο κατανοητή και εύκολη στη χρήση. Πρέπει πρώτα να οριστεί ο αριθμός καταχωρητών «**numberOfShiftRegisters**», τα «**dataPin**», «**latchPin**» και «**clockPin**» και τέλος να δημιουργήσω ένα αντικείμενο αυτής της κλάσης

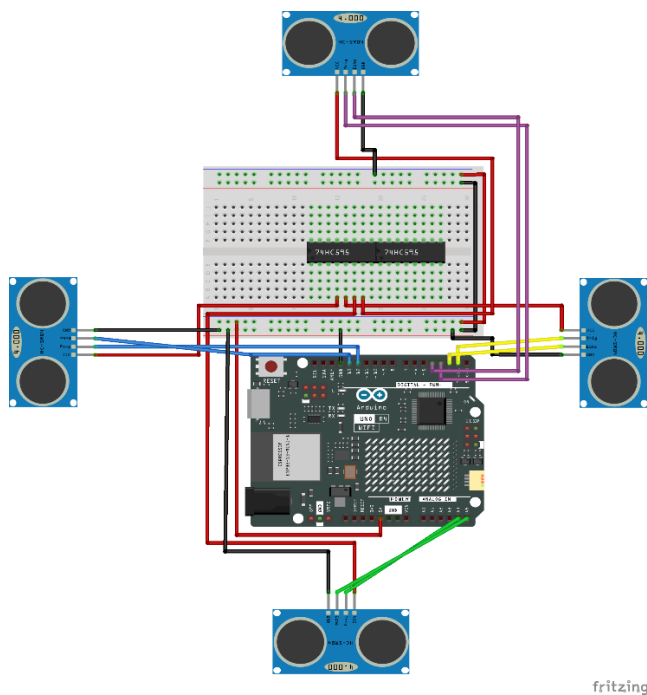
«**ShiftRegister74HC595<numberOfShiftRegisters> sr(dataPin, clockPin, latchPin);**» από το οποίο γίνεται ο χειρισμός των καταστάσεων των output pin των καταχωρητών. Πλέον, μπορώ λιτά και κατανοητά να ελέγχω τα φώτα και να ενεργοποιώ/απενεργοποιώ τους διάφορους αισθητήρες.



Εικόνα 32. Two Shift Register Connection

5.6.3 Αλλαγές σε αισθητήρες

Η χρήση υπερηχητικών αισθητήρων για την στροφή δεξιά και αριστερά ήρθε αφού παρατήρησα κάποιες δυσκολίες και αδυναμίες των προηγούμενων. Οι αισθητήρες υπέρυθρων δεν επιστρέφουν την απόσταση αλλά ένα σήμα ενεργοποίησης όταν αντανακλάται το υπέρυθρο φως από το ένα λαμπάκι στο άλλο, δηλαδή όταν εντοπίζει κάποιο αντικείμενο σε συγκεκριμένη απόσταση. Αυτή η απόσταση δεν είναι ακριβώς ορισμένη αλλά την ορίζει ο κάθε χρήστης περιστρέφοντας κάποιες βίδες που έχει πάνω ο αισθητήρας. Αυτό λοιπόν με

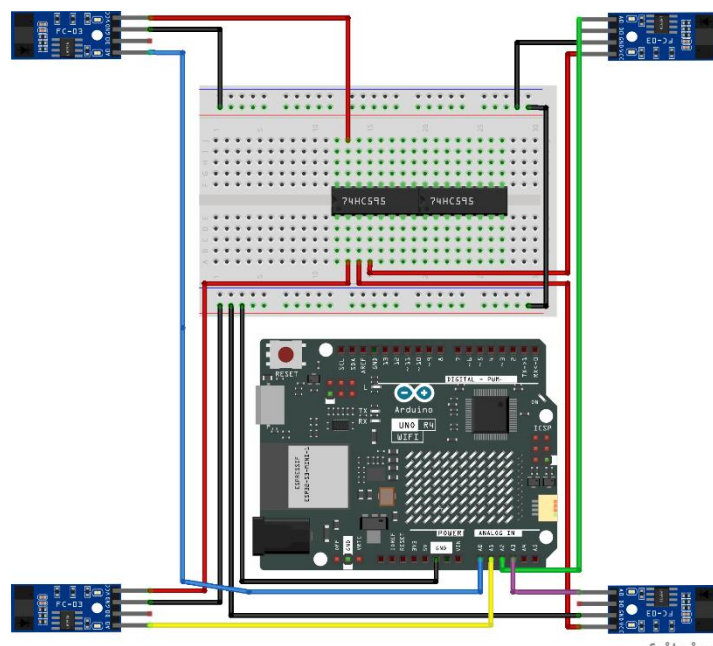


περιορίζει, επειδή δεν μπορώ να θέσω εύκολα να επιστρέφει το σήμα σε οποία απόσταση θέλω, με ακρίβεια και δεν είναι πάντα εφικτό να ρυθμίσω και τους δυο αισθητήρες να λειτουργούν ακριβώς το ίδιο. Οπότε αξιοποιώ σε όλες τις κατευθύνσεις τους αισθητήρες υπερήχων, κανονικοποιώντας τους ως στάνταρ λειτουργίες του προγράμματος μου και η ενεργοποίηση ή απενεργοποίηση τους γίνεται από τους shift registers.

Εικόνα 33. New Ultrasonic Setup

5.6.4 Step Counters

Προδιαγραφή του οχήματος μου, είναι και η χρήση αισθητήρων που μετράνε την περιστροφή των DC Motors. Η Μονάδα Αισθητήρα Ταχύτητας Υπέρυθρων είναι ένας μετρητής υπέρυθρων που διαθέτει 1 φωτοκύτταρο τύπου **H2010**, το οποίο αποτελείται από ένα φωτοτρανζίστορ, δέκτη υπέρυθρων, και έναν πομπό υπέρυθρου φωτός, συσκευασμένο σε ένα μαύρο πλαστικό περίβλημα πλάτους 10 cm . Εάν τοποθετηθεί οποιοδήποτε εμπόδιο μεταξύ αυτών των αισθητήρων, αποστέλλεται ένα σήμα στον μικροελεγκτή. Η μονάδα χρησιμοποιείται σε συνδυασμό με έναν τροχό, που έχει ακτίνες και κενά, και τοποθετείται στην βάση περιστροφής του κάθε κινητήρα. Για την σύνδεση με το Arduino θα ενώσω το VCC με την είσοδο τροφοδοσίας που είναι έξοδος του shift register, το GND με τη γείωσης και το κάθε OUT(Ψηφιακή έξοδος) με τα pin A0 έως και A3 της πλακέτας. Όταν ο αισθητήρας ταχύτητας



Εικόνα 34. Infrared Step Counter Sensors

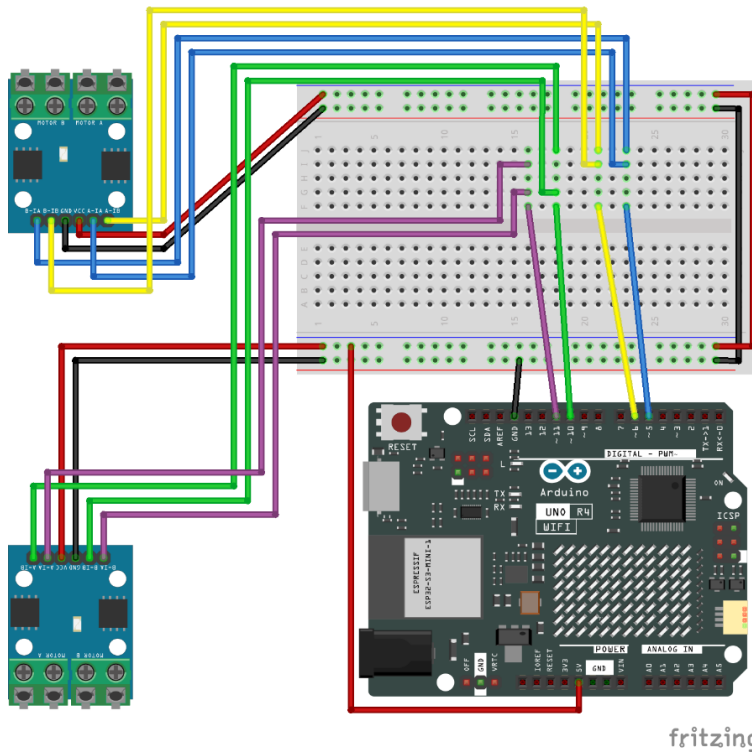
εμποδίζεται, εξάγει υψηλό επίπεδο(λογικό ναι), ενώ όταν δεν εμποδίζεται, εξάγει χαμηλό επίπεδο(λογικό όχι). Γενικά η μονάδα αισθητήρα ταχύτητας χρησιμοποιείται κυρίως για την ανίχνευση αλλαγών στην ταχύτητα περιστροφής ή την ταχύτητα. Όταν ένα αντικείμενο περνάει από τον αισθητήρα **H2010**, παράγει ένα παλμικό σήμα. Ο ενσωματωμένος συγκριτής **LM393** μέσα στη μονάδα συγκρίνει αυτό το παλμικό σήμα με ένα προκαθορισμένο όριο, παράγοντας ένα σταθερό σήμα εξόδου υψηλού επιπέδου.

Για τον κάθε αισθητήρα έχω μια καθολική μεταβλητή που «κρατάει» την προηγούμενη κατάσταση της εξόδου και την συγκρίνει με το τωρινό αποτέλεσμα της εξόδου του. Αν ανίχνευση αλλαγή, ενημερώνει αυτή την μεταβλητή και προσθέτει μια περιστροφή σε μια άλλη καθολική μεταβλητή. Τέλος, «τρέχω» συγκεντρωτικά τις συναρτήσεις της κάθε ρόδας μέσω της συνάρτησης **steps()**.

5.6.5 Λειτουργία μεταβλητής ταχύτητας κινητήρων

Μαζί με την ολοκλήρωση των προηγούμενων λειτουργιών στην συνδεσμολογία, άλλαξα και την σύνδεση των τροχών για να έχουν μεταβλητή ταχύτητα. Συνδέω τις μονάδες οδήγησης κινητήρων 2 καναλιών **L9110S** με τις θύρες του Arduino 5,6,10 και 11 με τέτοιο τρόπο έτσι ώστε κάθε μεριά τροχών να εκτελεί την ίδια κίνηση ταυτόχρονα. Η επιλογή θυρών δεν ήταν τυχαία αλλά έγινε γιατί αυτές υποστηρίζουν την λειτουργία **PWM**(Pulse With Modulation). [58] Η Διαμόρφωση Πλάτους Παλμού, ή PWM, είναι μια τεχνική για τη λήψη αναλογικών αποτελεσμάτων με ψηφιακά μέσα. Ο ψηφιακός έλεγχος χρησιμοποιείται για τη δημιουργία ενός τετραγωνικού κύματος, ενός σήματος που εναλλάσσεται μεταξύ ενεργοποίησης και απενεργοποίησης. Αυτό το μοτίβο ενεργοποίησης-απενεργοποίησης μπορεί να προσομοιώσει τάσεις μεταξύ του πλήρους Vcc της πλακέτας (π.χ., 5 V σε UNO, 3,3 V σε μια πλακέτα MKR) και απενεργοποίησης (0 Volts) αλλάζοντας το μέρος του χρόνου που το σήμα παραμένει ενεργό έναντι του χρόνου που το σήμα παραμένει απενεργοποιημένο. Η διάρκεια του «χρόνου ενεργοποίησης» ονομάζεται πλάτος παλμού. Για να λάβω μεταβαλλόμενες αναλογικές τιμές, διαμορφώνω κατάλληλα αυτό το πλάτος παλμού. Εάν επαναλάβω αυτό το μοτίβο ενεργοποίησης-απενεργοποίησης αρκετά γρήγορα με ένα LED για παράδειγμα, το αποτέλεσμα είναι σαν το σήμα να είναι μια σταθερή τάση μεταξύ 0 και Vcc που ελέγχει τη φωτεινότητα του LED. Η συχνότητα PWM του Arduino είναι περίπου στα 500Hz. Σε ορισμένους μικροελεγκτές, το PWM είναι διαθέσιμο μόνο σε επιλεγμένες ακίδες και συμβολίζονται με ένα σύμβολο περισπωμένης (~).

Μια κλήση στην `analogWrite()` είναι σε μια κλίμακα από 0 - 255, έτσι ώστε η `analogWrite(255)` να ζητά κύκλο λειτουργίας 100% (πάντα ενεργοποιημένη) και η `analogWrite(127)` να είναι κύκλος λειτουργίας 50% (ενεργοποιημένη τη μισή φορά) για παράδειγμα. Συγκεκριμένα, στο πρόγραμμα μου καλώ αυτή την συνάρτηση με



Εικόνα 35. PWM Connection

την εντολή `analogWrite(leftF, speed);`, όπου η καθολική μεταβλητή «**speed**» ορίζει την ταχύτητα. Η τιμή της μεταβλητής αλλάζει από το χρήστη σε 6 επίπεδα, δηλαδή ξεκινάει από 0 και οι υπόλοιπες τιμές υπολογίζονται από την συνάρτηση «**speed_set(int x)**». Σαν όρισμα παίρνει το επίπεδο τιμής από 1 έως και 5 και παρόμοια με την συνάρτηση «**map**» [59] δίνει τιμές της ταχύτητας, από 125(πολύ αργή ταχύτητα) μέχρι και 255 (μέγιστη ταχύτητα). Φυσικά, όρισα και καινούργιες εντολές για να γίνεται αυτή η αλλαγή.

5.6.6 Αυτόματη λειτουργία

Όπως ανέφερα, σε αυτή την φάση της αναβάθμισής μου, ολοκληρώθηκε και το πρώτο στάδιο της αυτόματης λειτουργίας του αυτοκίνητου. Ανέπτυξα την συνάρτηση «**auto_run()**» που εκτελεί μια βασική αυτόματη κίνηση, δηλαδή εκτελεί βήματα και στροφές με βάση το χρόνο. Οι χρόνοι για τις κινήσεις έχουν υπολογιστεί πειραματικά, με βάση τις απλές λειτουργίες της κίνησής του. Παράλληλα με την κίνηση εκτελεί και συλλογή δεδομένων υπολογισμού απόστασης εμποδίων μέσα στο χώρο, τα αποθηκεύει και μετά ενημερώνει και την σελίδα, επομένως και το χρήστη. Η αποθήκευση γίνεται σε ένα 10x10 «χάρτη», δηλαδή σε ένα στοιχείο `vector<std::vector<int>> vec_map` τύπου «vector». Στη γλώσσα προγραμματισμού C++, vector είναι ένας δυναμικός πίνακας που αποθηκεύει μια συλλογή στοιχείων του ίδιου τύπου σε συνεχή μνήμη. Έχει τη δυνατότητα να αλλάζει αυτόματα το μέγεθός του όταν εισάγεται ή διαγράφεται ένα στοιχείο. [60] Πριν τα ενσωματώσω στο πρόγραμμα μου χρησιμοποίησα την εφαρμογή Code::Blocks [61] για να επαληθεύσω ότι λειτουργούν σωστά και με τον τρόπο που χρειάζεται η εφαρμογή μου.

Στο πρόγραμμα μου ο αρχικός πίνακας περιέχει τιμές τύπου integer και είναι γεμάτος με συγκεκριμένες τιμές που καθορίζουν την έλλειψη δεδομένων. Μόλις, ξεκινήσει η αυτόματη κίνηση αρχίζει και περνάει από όλες τις θέσεις του «χάρτη»

και συλλέγει τιμές που δείχνουν αν υπάρχει η όχι εμπόδιο. Κάθε φορά παίρνει δεδομένα από τον μπροστινό αισθητήρα και από έναν από τους πλαϊνούς, αναλόγως την πορεία του μέσα στο χώρο. Χρησιμοποιεί καθολικές μεταβλητές για να κρατήσει την τρέχουσα θέση του, να ενημερώσει με δεδομένα τον πίνακα τιμών δεδομένων και στέλνει τα αποτελέσματα αυτά με μια ειδική συνάρτηση, την «**web_box(int p_x, int p_y, const char *d)**» που αξιοποιεί την συνάρτηση **web_me3**, για να στείλει τα αποτελέσματα στην ιστοσελίδα. Ελέγχει κάθε φορά αν έχει φτάσει την τελευταία θέση του πίνακα και ενημερώνει μια **global variable**. Σε περίπτωση που ισχύει αυτό, η μεταβλητή παίρνει την τιμή «**true**», τυπώνει τα αποτελέσματα του πίνακα, και με ένα **reverse flood fill*** [62] γεμίζει εσωτερικά κενά μέσα στις θέσεις που έχει βρεθεί αντικείμενο. Τέλος, με την συνάρτηση «**average_step()**» υπολογίζει το μέσω όρο περιστροφών και το στέλνει στην σελίδα με ειδικό τρόπο.

5.6.7 Αναβάθμιση στη λειτουργία κίνησης

Πλέον, η συνάρτηση **move()** ευθύνεται για τον έλεγχο της κίνησης, της λειτουργίας των **led** και της γωνίας του **servo** κινητήρα. Με καθολικές μεταβλητές αναγνωρίζει αν το αμάξι κινείται αυτόματα ή χειροκίνητα, αξιοποιώντας ή όχι τους αισθητήρες για την υποβοήθηση στην κίνηση, με την χρήση βέβαια ξεχωριστών συναρτήσεων. Η συνάρτηση που λειτουργεί και τις τρεις φορές είναι αυτή που ελέγχει αν θα είναι αναμμένα ή όχι τα φώτα.

5.6.8 Setup()

Στη συνάρτηση **setup()**, οργανώνω με διαφορετικό τρόπο την λειτουργία των **pin** προσαρμόζοντας την στις αλλαγές υλικού που έκανα, όπως είναι η χρήση επιπλέον υπερηχητικών αισθητήρων, η ένωση επιπλέον **shift register** και η ενεργοποίηση της λειτουργίας μέτρησης βημάτων του αυτοκίνητου. Ριζική αλλαγή έγινε στην συνάρτηση «**ws.onMessage**» αφού μπορεί να ξεχωρίζει αν έχει σταλεί από το χρήστη μια θέση στο χάρτη κίνησης. Στην περίπτωση που ο χάρτης έχει γεμίσει με δεδομένα, το πρόγραμμα βρίσκει σε ποια θέση στο χάρτη είναι το αμάξι και σε ποια θέση θέλει να το στείλει ο χρήστης, μετατρέπει τις θέσεις αυτές σε μια δομή, την «**struct Point { int x, y; }**», δηλαδή ένα ζευγάρι ακαίρων που χρησιμοποιούνται ως συντεταγμένες του κάθε σημείου στο χάρτη. Η μεταβλητή «**go_pos**» παίρνει τις συντεταγμένες του σημείου στόχου και η μεταβλητή «**start**» τις συντεταγμένες του οχήματος πάνω στο χάρτη. Ύστερα εκτελείται η συνάρτηση «**shortestPathWithTrace(const std::vector<std::vector<int>> &grid, Point start, Point end)**» και οδηγεί το όχημα στο σημείο που θέτει ο πελάτης, από την πιο σύντομη διαδρομή που βρίσκει μέσα στο χάρτη.

Αν το μήνυμα δεν είναι συντεταγμένες στο χάρτη, τότε με ένα **switch-case** δέχεται εντολές. Εφόσον, η καθολική μεταβλητή για την αυτόματη κίνηση είναι **false** τότε τρέχει η συνάρτηση «**auto_off_switch()**», διαφορετικά λειτουργεί η

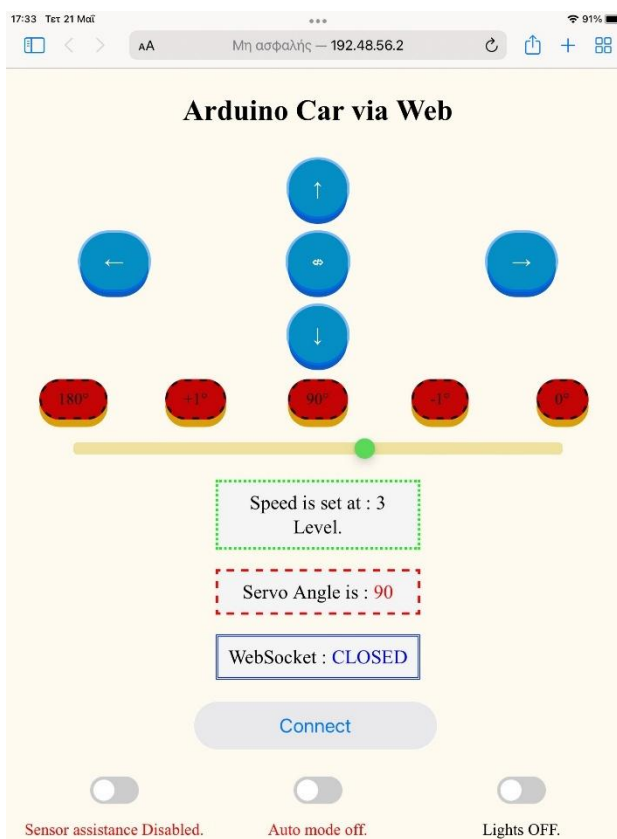
«**auto_on_switch()**». Κατά την εκτέλεση της πρώτης, το πρόγραμμα «ακούει» όλες τις διαθέσιμες εντολές και σε περίπτωση που έχει εντολή για αυτόματη κίνηση θέτει την ταχύτητα στο μέγιστο και ενεργοποιεί όλους τους αισθητήρες. Έπειτα, εκτελείται η δεύτερη, που αφήνει να περάσουν στο server εντολές μόνο για τα φώτα, την ταχύτητα και το κλείσιμο της αυτόματης κίνησης, έτσι ώστε να μην μπορεί αυτή την περίοδο ο χρήστης να δώσει χειροκίνητα εντολές για τη μετακίνηση του.

5.6.9 Loop()

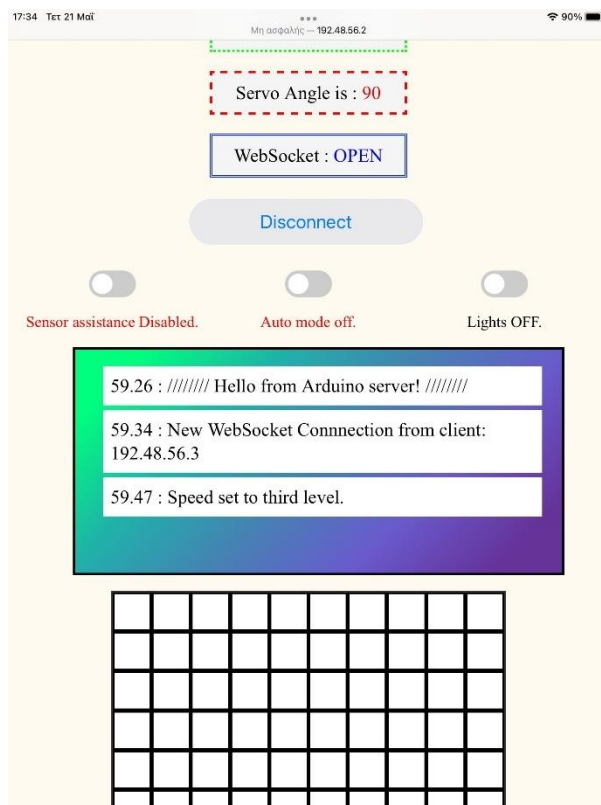
Οι βελτιώσεις που έγιναν στο loop αφορούν τις βελτιώσεις στην συνάρτηση μετακίνησης move() και στη συνάρτηση αποθήκευσης του μέσου όρου στροφών των κινητήρων. Επίσης έχω κάποιες βελτιώσεις στο τρόπο και χρόνο εκτύπωσης της σελίδας στο user.

5.6.10 HTML Version 3.0

Σε αυτή την έκδοση της ιστοσελίδας, προσάρμοσα και πρόσθεσα νέα στοιχεία για την χρήση των παραπάνω λειτουργιών, που έφτιαξα στο arduino, και ανανέωσα τα στυλ των στοιχείων της σελίδας μου για να διευκολύνει το πελάτη κατά την επίσκεψή του σε αυτή. Υπάρχει **μπάρα*** για την μεταβολή ταχύτητας από 0 που είναι η μηδενική ταχύτητα μέχρι και 5 που είναι η μέγιστη, ειδικό πλαίσιο που δείχνει την τρέχουσα ταχύτητα και με έξυπνο κώδικα προσαρμόζω την ταχύτητα αν έγινε κάποια αλλαγή όσο δεν ήταν ανοιχτό το WebSocket. Φτιάχνω ένα «**10x10**» χάρτη από τετραγωνάκια με έξυπνο κώδικα

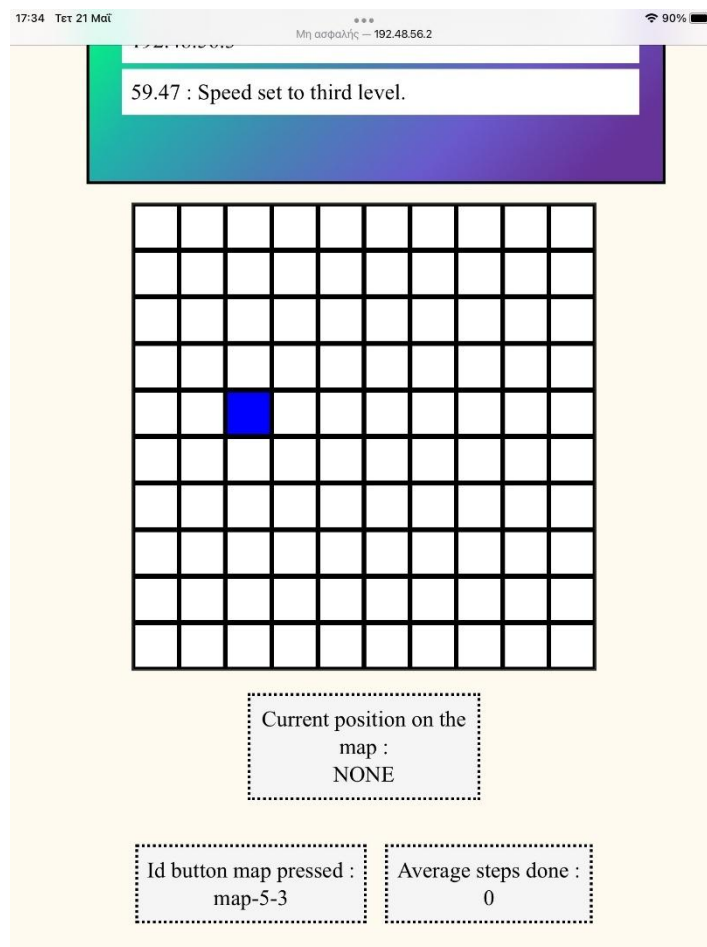


Εικόνα 36. HTML Version 3.0

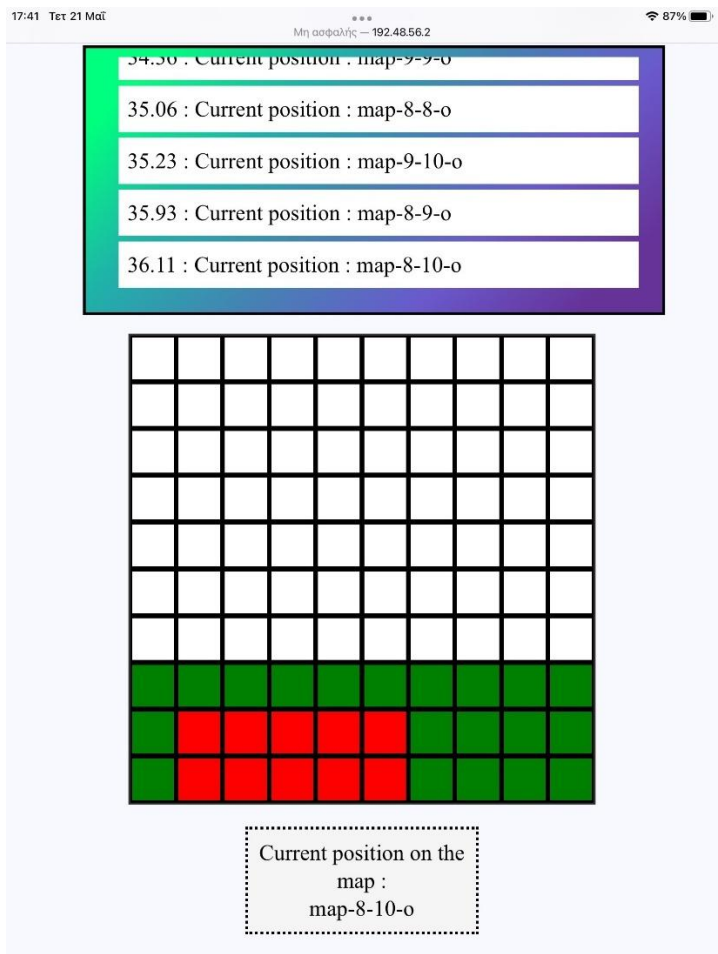


Εικόνα 37. Version 3.0 Initial Connection

JavaScript και τους δίνω ξεχωριστό «ID»*. Κάθε φορά που επιλέγω κάποιο από αυτά γίνεται μόνο αυτό μπλε και εμφανίζεται το ID του σε ξεχωριστό πλαίσιο. Όταν ξεκινάει την αυτόματη κίνηση η τρέχων θέση του είναι στις συντεταγμένες 10-10 και από εκεί ξεκινάει την πορεία του. Κάθε φορά που προχωράει μια θέση, ο server του στέλνει τα αποτελέσματα της θέσης σε μορφή «**map-N-N-o**», αν δεν βρεθεί εμπόδιο και σε μορφή «**map-N-N-x**», αν βρεθεί, όπου N ένας αριθμός συντεταγμένων από 1 έως και το 10, τα διαβάζει και με ειδικό τρόπο τα εκτυπώνει στο console window και ανανεώνει την τρέχουσα θέση του.



Εικόνα 38. New Functions On HTML



Εικόνα 39. Map In Action

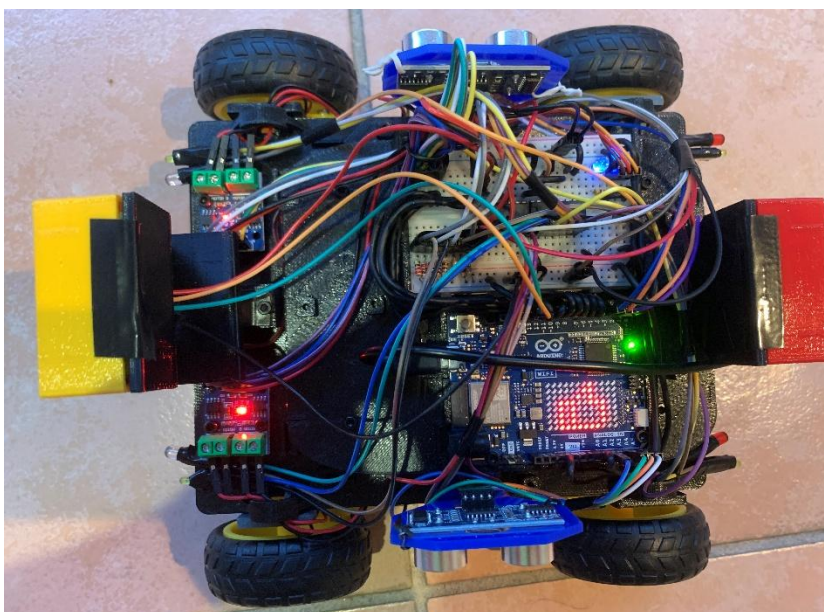
Επίσης στο χάρτη αλλάζει το χρώμα του μπροστινού και πλαϊνού(δεξιού ή αριστερού) κουτιού σε πράσινο αν δεν έχει κάποιο εμπόδιο αλλιώς σε κόκκινο. Τέλος, υπάρχει πλαίσιο που εμφανίζει το μέσο όρο των βημάτων που έχει διανύσει.

6 Αποτελέσματα – Τελικές Σκέψεις

Εν κατακλείδι, μπορώ να αναφέρω ότι έχω επεκτείνει σημαντικά τον βασικό θέμα της εργασίας. Αρχικά, αυτό ήταν η διερεύνηση της δυνατότητας άντλησης δεδομένων από αισθητήρες ενός συστήματος αυτοματισμού και δημιουργίας μιας εσωτερικά ενσωματωμένης οντότητας εξυπηρετητή που λύνει το πρόβλημα της οπτικοποίησης δεδομένων από αισθητήρες που είναι τοποθετημένοι σε κάποιο σύστημα αυτοματισμού. Επιλογή του συστήματος αυτού ήταν ένα αμαξίδιο που προήλθε από το σεμινάριο «Ανάλυση και Προγραμματισμός Συστημάτων Αυτοματισμού» από το NOESIS HUB, συνεργασία του Μητροπολιτικού Κολεγίου με το τμήμα HMTY του Πανεπιστημίου Πατρών.

Για να ολοκληρωθεί η τρέχουσα πρόοδος του του προγράμματος μου, χρειάστηκα σταδιακά να ανανεώνω τόσο το κομμάτι του κώδικα όσο και τη υλική κατασκευή με την προσθήκη και αλλαγή στοιχείων. Μπορώ να πω ότι το υλικό και ψηφιακό μέρος της εργασίας μου είναι τόσο πολύ αλληλένδετα και δεν αφήνουν μεγάλα περιθώρια για αλλαγές, μόνο με την μεταβολή της μορφής του ενός μέρους. Ξεκίνησα από το μηδέν και βήμα-βήμα πέρασα από μια λιτή και απλή εφαρμογή σε κάτι πολύ πιο σύνθετο. Κάθε νέα ενημέρωση έφερε νέες λειτουργίες στο κώδικα μου και αύξησε την αποδοτικότητα των προηγούμενων με αποτέλεσμα να φτάσω το πρόγραμμα μου μέχρι εδώ.

Συνοπτικά, στην τελευταία έκδοση της κατασκευής μου, το αμαξίδιο δημιουργεί ενσωματωμένο εξυπηρετητή, που συνδέεται με πελάτες, μέσω του πρωτοκόλλου WebSocket, και τους σερβίρει ταχύτατα μια HTML σελίδα για την κατεύθυνση του οχήματος. Πάνω στο όχημα έχω αισθητήρες για την συλλογή δεδομένων αλλά και πολλαπλά στοιχεία εξόδου τόσο στο αμάξι όσο και στην



Εικόνα 40. Arduino Car Finished Project (Vehicle Plan)

ιστοσελίδα, που ενεργοποιούνται και επιστρέφουν αποτελέσματα ανάλογα με την είσοδο των δεδομένων. Φυσικά, έχω δημιουργήσει και το πρώτο στάδιο για την αυτόματη κίνηση του οχήματος κατά την οποία συλλέγει δεδομένα, τα επιστρέφει και αποθηκεύει σε μια μορφή χάρτη και είναι δυνατόν να τον δω στην σελίδα μου.

7 Περιθώριο για μελλοντικές αναβαθμίσεις

Παρόλο που έχω προχωρήσει πολύ το έργο μου, βλέπω πως υπάρχει περιθώριο για αρκετές αναβαθμίσεις και στο κομμάτι του κώδικα αλλά και στο κομμάτι της κατασκευής.

7.1 New Software

Καταρχάς, θα μπορούσα να αλλάξω ριζικά τον τρόπο που εκτελούνται οι συναρτήσεις μου. Σε αυτή τη φάση του έργου, ο κώδικας τρέχει σειριακά, δηλαδή η μια συνάρτηση μετά την άλλη. Με την χρήση όμως της βιβλιοθήκης «FreeRTOS» θα μπορούσα να τις τρέχω παράλληλα. Το FreeRTOS, που σημαίνει Ελεύθερο Λειτουργικό Σύστημα Πραγματικού Χρόνου (Free Real-Time Operating System), είναι ένα ελαφρύ λειτουργικό σύστημα σχεδιασμένο ειδικά για ενσωματωμένα συστήματα. Όταν χρησιμοποιείται με το Arduino, η βιβλιοθήκη FreeRTOS επιτρέπει ένα πιο προηγμένο στυλ προγραμματισμού, επιτρέποντας την ταυτόχρονη εκτέλεση πολλαπλών εργασιών ή **νημάτων***. Αυτή η δυνατότητα πολλαπλών εργασιών είναι χρήσιμη σε έργα που απαιτούν να συμβαίνουν πολλές λειτουργίες ταυτόχρονα, όπως η ανάγνωση αισθητήρων, η ενημέρωση οθονών και ο ανεξάρτητος χειρισμός επικοινωνιών. Σε αντίθεση με το παραδοσιακό μοντέλο προγραμματισμού Arduino, το οποίο βασίζεται σε έναν μόνο βρόχο για τη διαχείριση όλων των λειτουργιών, το FreeRTOS παρέχει έναν χρονοπρογραμματιστή εργασιών που διαχειρίζεται την εκτέλεση πολλαπλών εργασιών με βάση τις προτεραιότητες και το χρονοδιάγραμμα. Προσφέρει επίσης εργαλεία όπως σηματοφόρους, ουρές και χρονοδιακόπτες για να διευκολύνει την επικοινωνία και τον συγχρονισμό μεταξύ εργασιών. Ενώ το FreeRTOS μπορεί να προστεθεί ως βιβλιοθήκη σε πολλές πλακέτες Arduino, είναι σημαντικό να σημειωθεί ότι ορισμένες πλακέτες, όπως η ESP32, έχουν ενσωματωμένο το FreeRTOS στο βασικό τους λογισμικό. Η χρήση του FreeRTOS μπορεί να βελτιώσει σημαντικά την οργάνωση και την ανταπόκριση σύνθετων εφαρμογών Arduino, ειδικά εκείνων που πρέπει να διαχειρίζονται συμβάντα σε πραγματικό χρόνο ή να λειτουργούν πολλά συστήματα παράλληλα. [63]

Επίσης θα μπορούσα να συνεχίσω την ανάπτυξη της αυτόματης κίνησης, ώστε να μπορεί να προηγηθεί ακόμη και στα πιο περίπλοκα εδάφη, αποφεύγοντας πιο αποτελεσματικά εμπόδια. Μεγάλη βοήθεια θα ήταν η δημιουργία κεντρικού Server για τον ταυτόχρονο έλεγχο πολλών συσκευών αλλά και για την δημιουργία πάνω σε αυτόν μιας βάσης δεδομένων που έχουν συλλέξει οι αισθητήρες.

Συνδυάζοντας, λοιπόν, όλα τα παραπάνω θα μπορούσα να φτιάξω έναν μεγαλύτερο και πολύ πιο έγκυρο χάρτη όχι μόνο ενός σπιτιού αλλά μιας ολόκληρης περιοχής, ανάλογα με τον αριθμό των οχημάτων στην διάθεση μου.

7.2 New Hardware

Κατά την άποψη μου, πρωταρχική αλλαγή στο υλικό της κατασκευής μου θα ήταν η αναβάθμιση των τεχνολογιών των αισθητήρων με άλλους αισθητήρες που είναι πιο έγκυροι και λεπτομερείς στις λήψεις των δεδομένων τους και η προσθήκη και προσαρμογή περισσότερων λειτουργιών όπως κάμερα πάνω σε ένα servo, αισθητήρες που μετρούν την σύσταση του εδάφους, το υψόμετρο και πολλούς άλλους.

Το αμαξίδιο αυτή την στιγμή μπορεί να προηγηθεί μόνο πάνω σε σταθερό έδαφος. Όμως θα μπορούσα, πιθανόν με εξωτερική βοήθεια, να εξελίξω το υλικό για καλύτερη και αποτελεσματικότερη κίνηση πάνω στο έδαφος αλλά και να δημιουργήσω νέες κατασκευές για την κίνηση σε νερό και αέρα. Η πορεία επέκτασης των φυσικών δυνατοτήτων του συστήματος μου εξελίσσεται σε κάθε περίπτωση αναλόγως με τις ανάγκες του ανθρώπου.

Η σχεδίαση του πίνακα ορολογιών έγινε με την βοήθεια της σελίδας Text Fixer [64] και η σχεδίαση κυκλωμάτων έγινε με το λογισμικό Fritzing [65].

8 Βιβλιογραφία

- [1] Wikipedia Contributors (2019). *History of human migration*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/History_of_human_migration.
- [2] Internet Archive. (2025). ReOrient [electronic resource] : global economy in the Asian Age : Frank, Andre Gunder, 1929-2005 : Free Download, Borrow, and Streaming : Internet Archive. [online] Available at: <https://archive.org/details/reorient00andr>.
- [3] Wikipedia Contributors (2019). *History of road transport*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/History_of_road_transport.
- [4] Wikipedia Contributors (2019). *Rail transport*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Rail_transport.
- [5] Wikipedia Contributors (2019). *Maritime history*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Maritime_history.
- [6] Wikipedia Contributors (2019). *History of aviation*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/History_of_aviation.
- [7] Wikipedia. (2020). *History of spaceflight*. [online] Available at: https://en.wikipedia.org/wiki/History_of_spaceflight.
- [8] Arduino (2018). *What Is Arduino?* [online] www.arduino.cc. Available at: <https://www.arduino.cc/en/Guide/Introduction/>.
- [9] Kushner, D. (2011). *The Making of Arduino*. [online] IEEE Spectrum. Available at: <https://spectrum.ieee.org/the-making-of-arduino>.
- [10] web.archive.org. (2017). *Arduino FAQ*. [online] Available at: <https://web.archive.org/web/20170906182556/http://medea.mah.se/2013/04/arduino-faq/>.
- [11] Allan, A. (2015). *Arduino Wars: Group Splits, Competing Products Revealed? - Make*: [online] Make: DIY Projects and Ideas for Makers. Available at: <https://makezine.com/article/technology/arduino/arduino-vs-arduino/> [Accessed 25 May 2025].
- [12] audioXpress (2017). *A New Era for Arduino: Original Arduino Founders Finally Get 100% Control*. [online] audioXpress. Available at:

<https://audioxpress.com/news/a-new-era-for-arduino-original-arduino-founders-finally-get-100-control> [Accessed 25 May 2025].

- [13] Stockton, N. (2017). *Arduino's New CEO, Federico Musto, May Have Fabricated His Academic Record*. [online] WIRED. Available at: <https://www.wired.com/2017/04/arduinios-new-ceo-federico-musto-may-fabricated-academic-record/> [Accessed 25 May 2025].
- [14] Davis, T. (2017). *BCMI Acquires Arduino AG and Makers Breathe a Sigh of Relief | TechWombat*. [online] TechWombat. Available at: <https://techwombat.com/bcmi-acquires-arduino-ag-makers-breathe-sigh-relief> [Accessed 25 May 2025].
- [15] Wilson, R. (2017). *Arduino reborn partners with ARM | Electronics Weekly*. [online] Electronics Weekly. Available at: <https://www.electronicsworld.com/blogs/distribution-world/arduino-reborn-partners-arm-2017-10/> [Accessed 25 May 2025].
- [16] Emilio, M.D.P. (2020). *Open-source HW in the Modern Era: Interview of Arduino's CEO Fabio Violante*. [online] EE Times Europe. Available at: <https://www.eetimes.eu/open-source-hardware-in-the-modern-era-interview-of-arduinios-ceo-fabio-violante/>.
- [17] Archive.org. (2025). *Freeduino Open Source Hardware*. [online] Available at: https://web.archive.org/web/20080410220309/http://www.freeduino.org/freeduino_open_designs.html [Accessed 25 May 2025].
- [18] www.arduino.cc. (n.d.). *Arduino Hardware*. [online] Available at: <https://www.arduino.cc/en/hardware/>.
- [19] Adafruit Industries (2025). *DC Boarduino (Arduino compatible) Kit (w/ATmega328)*. [online] Adafruit.com. Available at: <https://www.adafruit.com/product/72> [Accessed 25 May 2025].
- [20] Stefano Di Tore, Todino, M.D. and Sibilio, M. (2019). Disuffo: Design, prototyping and development of an open-source educational robot. *DOAJ (DOAJ: Directory of Open Access Journals)*. [online] doi:<https://doi.org/10.13128/formare-24446>.
- [21] Arduino.cc. (2024). Available at: <https://docs.arduino.cc/libraries/>.
- [22] Arduino.cc. (2024). Available at: <https://docs.arduino.cc/tutorials/uno-r4-wifi/led-matrix/>.

- [23] Arduino.cc. (2025). Available at: https://docs.arduino.cc/libraries/servo/?_gl=1 [Accessed 25 May 2025].
- [24] Denk, T. (2019). *Shift Register 74HC595 Arduino Library - Timo Denk's Blog*. [online] Timodenk.com. Available at: <https://blog.timodenk.com/shift-register-arduino-library/> [Accessed 25 May 2025].
- [25] mike-matera (2022). *GitHub - mike-matera/ArduinoSTL: An STL and iostream implementation based on uClibc++ that supports my CS-11M class*. [online] GitHub. Available at: <https://github.com/mike-matera/ArduinoSTL> [Accessed 25 May 2025].
- [26] skaarj1989 (2024). *GitHub - skaarj1989/mWebSockets: WebSockets for microcontrollers*. [online] GitHub. Available at: <https://github.com/skaarj1989/mWebSockets> [Accessed 25 May 2025].
- [27] Arduino.cc. (2024). Available at: <https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples/>.
- [28] Google Books. (2025). *Web Server Technology*. [online] Available at: https://books.google.gr/books?id=0jExRH3_-hQC&q=%22Web+server%22+-wikipedia&pg=PA14&redir_esc=y#v=snippet&q=%22Web%20server%22%20-wikipedia&f=false [Accessed 25 May 2025].
- [29] Johny, A. and Madhusoodanan, K.N. (2021). Edge Computing Using Embedded Webserver with Mobile Device for Diagnosis and Prediction of Metastasis in Histopathological Images. *International Journal of Computational Intelligence Systems*, 14(1). doi:<https://doi.org/10.1007/s44196-021-00040-x>.
- [30] websockets.spec.whatwg.org. (n.d.). *WebSockets Standard*. [online] Available at: <https://websockets.spec.whatwg.org/>.
- [31] Mozilla (2019). *The WebSocket API (WebSockets)*. [online] MDN Web Docs. Available at: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API.
- [32] WHATWG (2019). *HTML Standard*. [online] Whatwg.org. Available at: <https://html.spec.whatwg.org/multipage/>.
- [33] Berners-Lee, T. (1989). *The original proposal of the WWW, HTMLized*. [online] W3.org. Available at: <https://www.w3.org/History/1989/proposal.html>.
- [34] Hire Remote Developers | Top 5% Freelance Developers. (2025). *Optimize - Hire Remote Developers | Top 5% Freelance Developers*. [online] Available at:

<https://optymize.io/blog/top-10-excellent-uses-of-html-in-the-real-world/>
[Accessed 25 May 2025].

- [35] Noesishub.com. (2023). *edgeLAB – noesishub.com*. [online] Available at: <https://noesishub.com/el/edgelab/> [Accessed 25 May 2025].
- [36] Γνωρίζω τον κόσμο των Ρομπότ. (2025). *Χρησιμοποιώντας Breadboard*. [online] Available at: <https://4dimkal-robot.weebly.com/chirhoetasigmaiotamuomicronpiomicroniota974nutaualpha-sigmaf-breadboard.html> [Accessed 25 May 2025].
- [37] Google Books. (2025). *Industrial Motor Control*. [online] Available at: https://books.google.gr/books?id=E9nP3VBRO6AC&dq=shunt+series+compou+nd+dc+motor+connections&pg=PA251&redir_esc=y [Accessed 25 May 2025].
- [38] Last Minute Engineers (2019). *How Servo Motor Works & Interface It With Arduino*. [online] Last Minute Engineers. Available at: <https://lastminuteengineers.com/servo-motor-arduino-tutorial/>.
- [39] L9110 2-CHANNEL MOTOR DRIVER. (n.d.). Available at: https://www.laskakit.cz/user/related_files/l9110_2_channel_motor_driver.pdf.
- [40] ElecFreaks (2011). *Ultrasonic Ranging Module HC -SR04*. [online] Available at: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>.
- [41] Hellas Digital. (2025). *IR Infrared Obstacle Avoidance Sensor*. [online] Available at: <https://www.hellasdigital.gr/electronics/sensors/infrared-sensors/ir-infrared-obstacle-avoidance-sensor/> [Accessed 25 May 2025].
- [42] Hellas Digital. (2025). *Beam Photoelectric Sensor - Infrared Rotation Count Sensor Module for Arduino*. [online] Available at: <https://www.hellasdigital.gr/electronics/sensors/infrared-sensors/beam-photoelectric-sensor-infrared-radiation-count-sensor-module-for-arduino/> [Accessed 25 May 2025].
- [43] Wikipedia Contributors (2019). *Shift register*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Shift_register.
- [44] Arduino Forum. (2013). *calling a function from HTML button*. [online] Available at: <https://forum.arduino.cc/t/calling-a-function-from-html-button/161360/7> [Accessed 25 May 2025].
- [45] Newbiely.com (2025). *Arduino UNO R4 - Web Server | Arduino UNO R4 Tutorial*. [online] Tutorials for Newbies. Available at:

<https://newbiely.com/tutorials/arduino-uno-r4/arduino-uno-r4-web-server>
[Accessed 25 May 2025].

- [46] Newbiely.com (2025). *Arduino UNO R4 - Controls Car via Web | Arduino UNO R4 Tutorial*. [online] Tutorials for Newbies. Available at: <https://newbiely.com/tutorials/arduino-uno-r4/arduino-uno-r4-controls-car-via-web> [Accessed 25 May 2025].
- [47] Instructables (2018). *WiFi WebSocket Remote Robot*. [online] Instructables. Available at: <https://www.instructables.com/WiFi-WebSocket-Remote-Robot/> [Accessed 25 May 2025].
- [48] MDN Web Docs. (n.d.). *Writing WebSocket client applications*. [online] Available at: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_client_applications.
- [49] Hpe.com. (2021). *Example: Browser-based WebSocket connection*. [online] Available at: https://arubanetworking.hpe.com/techdocs/AOS-CX/10.08/HTML/rest_v1/Content/Chp_notifi/ex-bro-bas-not-sub6.htm [Accessed 25 May 2025].
- [50] AminM (2013). *JavaScript code does not work in HEAD tag*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/15675745>.
- [51] Isaac100 (2017). *Getting Started with the HC-SR04 Ultrasonic sensor*. [online] projecthub.arduino.cc. Available at: <https://projecthub.arduino.cc/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-7cabe1>.
- [52] Wokwi.com. (2019). *IRSensor - Wokwi ESP32, STM32, Arduino Simulator*. [online] Available at: <https://wokwi.com/projects/377298899531999233> [Accessed 25 May 2025].
- [53] Arduino.cc. (2024). Available at: <https://docs.arduino.cc/tutorials/uno-r4-wifi/led-matrix/>.
- [54] Arduino.cc. (2024). Available at: <https://docs.arduino.cc/tutorials/communication/guide-to-shift-out/>.
- [55] Lastminuteengineers.com. (2021). *How 74HC595 Shift Register Works & Interface it with Arduino*. [online] Available at: <https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/>.

- [56] Arduino.cc. (2025). Available at: <https://docs.arduino.cc/language-reference/en/variables/utilities/PROGMEM/> [Accessed 25 May 2025].
- [57] Arduino Forum. (2012). *What does the F() do exactly?* [online] Available at: <https://forum.arduino.cc/t/what-does-the-f-do-exactly/89384/3> [Accessed 25 May 2025].
- [58] Hirzel, T. (2022). *Basics of PWM (Pulse Width Modulation)*. [online] Arduino.cc. Available at: <https://docs.arduino.cc/learn/microcontrollers/analog-output/>.
- [59] Arduino.cc. (2024). *map()*. [online] Available at: <https://docs.arduino.cc/language-reference/en/functions/math/map/>.
- [60] GeeksforGeeks (2015). *Vector in C++ STL*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/vector-in-cpp-stl/>.
- [61] www.codeblocks.org. (n.d.). *Code::Blocks*. [online] Available at: <https://www.codeblocks.org/>.
- [62] GeeksforGeeks. (2020). *Flood Fill Algorithm*. [online] Available at: <https://www.geeksforgeeks.org/flood-fill-algorithm/>.
- [63] Arduino.cc. (2024). Available at: <https://docs.arduino.cc/libraries/freertos/>.
- [64] TextFixer. (2024). *Sort Lines Alphabetically Online*. [online] Available at: <https://www.textfixer.com/tools/sort-lines-alphabetically-online.php> [Accessed 25 May 2025].
- [65] Fritzing (n.d.). *Fritzing*. [online] fritzing.org. Available at: <https://fritzing.org/>.