



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και υλοποίηση έξυπνου οχήματος-ακόλουθου με
δυνατότητα αποφυγής εμποδίων

Ιωάννης Στολτίδης

Επιβλέπων Καθηγητής: Δρ. Μηνάς Δασυγένης

Εργαστήριο Ψηφιακών Συστημάτων & Αρχιτεκτονικής Υπολογιστών

Κοζάνη, Ιούλιος 2017



UNIVERSITY OF WESTERN MACEDONIA

FACULTY OF ENGINEERING

**DEPARTMENT OF INFORMATICS &
TELECOMMUNICATIONS ENGINEERING**

THESIS

Design and implementation of a smart car/follower with avoid obstacles function

Ioannis Stoltidis

Supervisor: Dr. Minas Dasygenis

Laboratory of Digital Systems & Computer Architecture

Kozani, July 2017

Κατάλογος Περιεχομένων

Κατάλογος Εικόνων	- 9 -
Κατάλογος πινάκων	- 13 -
Περίληψη	- 15 -
Abstract	- 17 -
Δήλωση πνευματικών δικαιωμάτων	- 19 -
Ευχαριστίες	- 21 -
Κεφάλαιο 1 – Εισαγωγή.....	- 23 -
1.1 Σημερινή Κατάσταση.....	- 23 -
1.2 Κίνητρα για την εκπόνηση της Διπλωματικής Εργασίας.....	- 24 -
1.3 Αντικείμενο της Διπλωματικής Εργασίας.....	- 25 -
1.4 Οφέλη της Διπλωματικής Εργασίας.....	- 25 -
1.5 Σύνοψη της Διπλωματικής Εργασίας.....	- 26 -
1.6 Σύνοψη κεφαλαίου	- 27 -
Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο.....	- 29 -
2.1 Πρόγραμμα χειρισμού του οχήματος.....	- 29 -
2.1.1 Ασφαλής φλοιός SSH.....	- 29 -
2.1.2 OpenCV.....	- 31 -
2.1.3 DirectX input και Xinput	- 31 -
2.1.4 Visual C++ και πλαίσιο .NET.....	- 32 -
	- 5 -

2.2 Η πλακέτα Raspberry Pi.....	- 33 -
2.3 Python	- 34 -
2.4 Η Πλακέτα Arduino και το υλικό του οχήματος.....	- 35 -
2.5 Προγράμματα ανάπτυξης.....	- 38 -
2.5.1 Microsoft Visual Studio	- 38 -
2.5.2 Arduino IDE.....	- 38 -
2.5.3 Notepad++.....	- 39 -
2.5.4 SketchUp.....	- 39 -
2.5.5 Win32DiskImager	- 39 -
2.6 Σύνοψη κεφαλαίου.....	- 39 -
Κεφάλαιο 3 – Ανάλυση υλικού μέρους	- 41 -
3.1 Υλικό μέρος οχήματος	- 41 -
3.1.1 Περί τροφοδοσίας του συστήματος.....	- 42 -
3.1.2 Αισθητήρες υπέρηχων και υπέρυθρης ακτινοβολίας	- 43 -
3.1.3 Διπλή Γέφυρα Η L298N.....	- 46 -
3.1.4 Σερβοκινητήρες και κύκλωμα θέσης τιμονιού.....	- 48 -
3.1.5 Συνολικό σύστημα του οχήματος.....	- 49 -
3.2 Υλικό μέρος πομπού	- 56 -
3.3 Κόστος Υλικών Κατασκευής	- 60 -
3.4 Σύνοψη Κεφαλαίου	- 61 -
Κεφάλαιο 4 – Ανάλυση λογισμικού μέρους	- 63 -
4.1 Λογισμικό στα Arduino	- 63 -

4.1.1 Arduino πομπού	- 63 -
4.1.2 Arduino δέκτη IR	- 64 -
4.1.3 Arduino Ελέγχου οχήματος.....	- 65 -
4.2 Ρύθμιση συστήματος του Raspberry Pi.....	- 67 -
4.2.1 Ενεργοποίηση SSH	- 68 -
4.2.2 Εγκατάσταση Wiring GPIO	- 69 -
4.2.3 Εγκατάσταση και ρύθμιση Motion	- 71 -
4.2.4 Ρύθμιση προσωρινών αρχείων και εφαρμογών εκκίνησης	- 73 -
4.3 Λογισμικό στο Raspberry Pi	- 75 -
4.3.1 Διεργασία ανάγνωσης απόστασης εμποδίων	- 77 -
4.3.2 Διεργασία ανάγνωσης Bluetooth.....	- 78 -
4.3.3 Λειτουργία «Follow me»	- 79 -
4.3.4 Λειτουργία αποφυγής εμποδίων.....	- 82 -
4.3.5 Λειτουργία ελέγχου οχήματος.....	- 84 -
4.4 Λογισμικό στα Windows – εφαρμογή χρήστη.....	- 86 -
4.4.1 Εγκατάσταση του CarControl	- 86 -
4.4.2 Χρήση & λειτουργία του CarControl.....	- 89 -
4.4.3 Μεταγλώττιση της εφαρμογής	- 94 -
4.4.4 Σύνδεση SSH.....	- 98 -
4.4.5 Ανάγνωση χειριστηρίου	- 99 -
4.4.6 Ανάγνωση τιμών αισθητήρων.....	- 101 -
4.5 Σύνοψη Κεφαλαίου	- 102 -

Κεφάλαιο 5 – Σύνοψη συστήματος και συμπεράσματα	- 103 -
5.1 Γενικά χαρακτηριστικά του συστήματος	- 103 -
5.1.1 Μετρήσεις συστήματος	- 104 -
5.1.2 Ασφάλεια του συστήματος.....	- 105 -
5.1.3 Ανάλυση SWOT.....	- 107 -
5.2 Γενική χρήση και έλεγχοι του συστήματος.....	- 108 -
5.2.1 Χρήση του συστήματος.....	- 108 -
5.2.2 Έλεγχοι ορθής λειτουργίας – αποσφαλμάτωση	- 109 -
5.3 Συμπεράσματα της προσέγγισης.....	- 112 -
5.4 Προβλήματα που προέκυψαν κατά την υλοποίηση	- 113 -
5.5 Προτάσεις για επέκταση της υλοποίησης	- 114 -
5.6 Σύνοψη κεφαλαίου	- 115 -
Βιβλιογραφία.....	- 117 -

Κατάλογος Εικόνων

Εικόνα 1: Αυθεντικοποίηση SSH.....	- 30 -
Εικόνα 2: Βασική χρήση SSH.....	- 31 -
Εικόνα 3: Στοιχεία του πλαισίου .NET	- 33 -
Εικόνα 4: Raspberry Pi Model B.....	- 33 -
Εικόνα 5: Raspberry Pi 3 Model B.....	- 34 -
Εικόνα 6: Arduino Uno	- 37 -
Εικόνα 7: Dual H bridge L298N	- 37 -
Εικόνα 8: LM2596s DC-DC.....	- 37 -
Εικόνα 9: HC-SR04.....	- 37 -
Εικόνα 10: Bluetooth BC417	- 38 -
Εικόνα 11: Τροφοδοσία συστήματος οχήματος.....	- 42 -
Εικόνα 12: Ακροδέκτες αισθητήρα υπέρηχων	- 44 -
Εικόνα 13: Περιορισμοί εντοπισμού εμποδίου	- 44 -
Εικόνα 14: Περιπτώσεις μη ανίχνευσης εμποδίων.....	- 45 -
Εικόνα 15: Τοπολογία Αισθητήρων IR.....	- 45 -
Εικόνα 16: Διάταξη ακροδεκτών αισθητήρα IR	- 46 -
Εικόνα 17: Λειτουργία της γέφυρας H.....	- 47 -
Εικόνα 18: Σύστημα θέσης τιμονιού	- 48 -
Εικόνα 19: Κύκλωμα συστήματος θέσης τιμονιού	- 49 -

Εικόνα 20: Pullup και Pulldown διάταξη	- 50 -
Εικόνα 21: Κύκλωμα του οχήματος με χρήση Raspberry Pi 3	- 52 -
Εικόνα 22: Κύκλωμα του οχήματος με χρήση Raspberry Pi	- 53 -
Εικόνα 23: Ολοκληρωμένη εγκατάσταση συστήματος.....	- 54 -
Εικόνα 24: Αισθητήρες απόστασης μπροστινού μέρους και κάμερας.....	- 54 -
Εικόνα 25: Εξωτερική εμφάνιση κατασκευής οχήματος	- 55 -
Εικόνα 26: Σχηματικό κυκλώματος πομπού	- 56 -
Εικόνα 27: προσανατολισμός 0°.....	- 57 -
Εικόνα 28: προσανατολισμός 90°.....	- 57 -
Εικόνα 29: προσανατολισμός 180°.....	- 57 -
Εικόνα 30: προσανατολισμός 270°.....	- 57 -
Εικόνα 31: Κάτω μέρος Θήκης	- 58 -
Εικόνα 32: Πάνω μέρος Θήκης	- 58 -
Εικόνα 33: Πάνω μεριά μεσαίου μέρους θήκης.....	- 59 -
Εικόνα 34: Κάτω μεριά μεσαίου μέρους θήκης	- 59 -
Εικόνα 35: Εγκατάσταση λειτουργικού στο Raspberry Pi.....	- 68 -
Εικόνα 36: Ενεργοποίηση SSH στο Raspberry Pi.....	- 69 -
Εικόνα 37: Εγκατάσταση git-core	- 69 -
Εικόνα 38: Εγκατάσταση GPIO Wiring.....	- 70 -
Εικόνα 39: Αρίθμηση ακροδεκτών GPIO	- 71 -

Εικόνα 40: Εγκατάσταση Motion.....	- 71 -
Εικόνα 41: Ορθή λειτουργία Motion.....	- 72 -
Εικόνα 42: Προσάρτηση προσωρινού αρχείου	- 73 -
Εικόνα 43: Ρύθμιση Crontab	- 74 -
Εικόνα 44: Λειτουργία Συστήματος.....	- 76 -
Εικόνα 45: Χρήση αισθητήρων συστήματος	- 79 -
Εικόνα 46: Λειτουργία της Follow me.....	- 81 -
Εικόνα 47: Ελάχιστες αποστάσεις για την αποφυγή εμποδίων.....	- 82 -
Εικόνα 48: Αλγόριθμος αποφυγής εμποδίων	- 84 -
Εικόνα 49: Λειτουργία της «fwdcom»	- 85 -
Εικόνα 50: Αρχική οθόνη εγκαταστάτη του CarControl.....	- 86 -
Εικόνα 51: Οθόνη του εγκαταστάτη συμφωνίας άδειας	- 87 -
Εικόνα 52: Οθόνη τοποθεσίας εγκατάστασης.....	- 87 -
Εικόνα 53: Οθόνη επιβεβαίωσης εγκατάστασης.....	- 88 -
Εικόνα 54: Εγκατάσταση Visual C++ πακέτων	- 88 -
Εικόνα 55: Οθόνη επιτυχίας εγκατάστασης	- 89 -
Εικόνα 56: Αρχική οθόνη του CarControl	- 89 -
Εικόνα 57: Μήνυμα λάθους στο CarControl.....	- 90 -
Εικόνα 58: CarControl - επιτυχής σύνδεση.....	- 91 -
Εικόνα 59: CarControl - ενεργοποίηση DX	- 91 -

Εικόνα 60: Διάταξη χειριστηρίου	- 92 -
Εικόνα 61: Συνολική λειτουργία ελέγχου οχήματος	- 93 -
Εικόνα 62: Ροή δεδομένων στην CarControl	- 93 -
Εικόνα 63: Βαθμολογία Codacy.....	- 105 -
Εικόνα 64: Απευθείας βραχυκύκλωση κινητήρα κίνησης	- 112 -

Κατάλογος πινάκων

Πίνακας 1: Χαρακτηριστικά μοντέλων Raspberry Pi	- 34 -
Πίνακας 2: Χαρακτηριστικά υλικών που χρησιμοποιήθηκαν	- 38 -
Πίνακας 3: Τιμές ακροδεκτών για τον χειρισμό των DC κινητήρων	- 48 -
Πίνακας 4: Μέτρηση έντασης με διαφορετικούς προσανατολισμούς στον πομπό. -	57 -
Πίνακας 5: Κόστος υλοποίησης	- 60 -
Πίνακας 6: Παραδείγματα δομής της IR μεταβλητής	- 64 -
Πίνακας 7: Ενέργειες ελέγχου βάση της μεταβλητής εισόδου του Arduino.....	- 66 -
Πίνακας 8: Ρυθμίσεις Motion.....	- 72 -
Πίνακας 9: Αναμενόμενες καταστάσεις των LEDs.....	- 77 -
Πίνακας 10: Μεταβλητή sonar_sens	- 78 -
Πίνακας 11: Τιμές & ενέργειες για την «sensors»	- 80 -
Πίνακας 12: Ενέργειες κουμπιών χειριστηρίου.....	- 92 -

Περίληψη

Στην παρούσα διπλωματική εργασία μετατράπηκε ένα ηλεκτροκίνητο αυτοκίνητο για παιδιά σε ένα έξυπνο όχημα με δυνατότητα χειρισμού μέσω διαδικτύου και συνοδείας ατόμου αποφεύγοντας εμπόδια.

Η τροποποίηση έγινε με την πλήρη αντικατάσταση του εργοστασιακού κυκλώματος (εκτός από τη μπαταρία του και τους δύο κινητήρες που είναι υπεύθυνοι για την κίνηση του οχήματος) με ένα άλλο, σχεδιασμένο για την κάλυψη των αναγκών της εργασίας. Η καρδιά αυτού του κυκλώματος είναι ένας μικρός υπολογιστής μονής πλακέτας Raspberry Pi στον οποίον είναι συνδεδεμένη μια κάμερα υπολογιστή, καθώς και δύο μικροελεγκτές Arduino Uno. Το υπόλοιπο κύκλωμα αποτελείται από κουμπιά-μπουτόν, οθόνη υγρών κρυστάλλων, αισθητήρες υπέρηχων μέτρησης απόστασης, αισθητήρες υπέρυθρης ακτινοβολίας, σερβοκινητήρες και κινητήρες συνεχούς ρεύματος. Επιπρόσθετα τοποθετήθηκε μια ακόμα μπαταρία με τα επιμέρους απαραίτητα εξαρτήματα, όπως ασφάλεια, διακόπτες και κυκλώματα μεταβολής τάσης, για την τροφοδοσία του παραπάνω κυκλώματος. Τέλος κατασκευάστηκε με χρήση ενός Arduino Uno ένας πομπός υπέρυθρης ακτινοβολίας εξοπλισμένος με πομπό Bluetooth.

Ο τομέας του λογισμικού αποτελείται από δύο βασικά μέρη. Το πρώτο περιλαμβάνει την υλοποίηση εφαρμογής που τρέχει σε Windows και είναι υπεύθυνη για τον έλεγχο του οχήματος μέσω διαδικτύου με χρήση ενός χειριστηρίου. Το δεύτερο αφορά τις υπηρεσίες που τρέχουν τόσο στο Raspberry Pi όσο και στο Arduino. Οι υπηρεσίες αυτές είναι υπεύθυνες για την εκτέλεση των εντολών ελέγχου, τη διαχείριση αισθητήρων, την αποτροπή σύγκρουσης και την αυτόματη κίνηση του οχήματος για την συνοδεία στόχου και αποφυγής εμποδίων.

Λέξεις κλειδιά: C++, C, Python, Raspberry Pi, Ηλεκτροκινητήρας, Arduino, Αισθητήρας Υπέρηχων, Αισθητήρας Υπέρυθρων, Χειριστήριο, SSH, Οθόνη Υγρών Κρυστάλλων, Bluetooth

Abstract

In this thesis an electrically driven vehicle for kids was converted into a smart car which can be remotely controlled via internet connection, and follow a person avoiding obstacles on its way.

This modification was implemented by replacing the factory controller circuit, except for the battery and the two DC motors which are responsible for the vehicle's movement, with another circuit especially designed for the purposes of this thesis. The heart of this circuit is a small single-board computer Raspberry Pi on which a web camera and two Arduino Uno microcontrollers are connected. The rest of the circuit involves some push buttons, LCD, ultrasound distance sensors, infrared receivers, servo motors and DC motors. A second battery, with some more components such as switches, DC regulators and electrical safety, were added for the power supply of the above circuit. Finally an accessory with infrared and Bluetooth transmitter was created using an Arduino Uno.

The software is composed of two parts. The first includes the application which runs on Windows and is responsible for the control of the vehicle via internet connection with the use of a gamepad. The second one includes the services running on both Raspberry Pi and Arduino. Those services are used for the execution of control commands, sensor management, crash avoidance and automated movement of the vehicle while following the target and avoiding the obstacles.

Keywords: C++, C, Python, Raspberry Pi, Electric Motor, Arduino, Ultrasound distance sensor, IR sensor, Gamepad, SSH, LCD, Bluetooth.

Δήλωση πνευματικών δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο «Σχεδιασμός και υλοποίηση έξυπνου οχήματος-ακόλουθου με δυνατότητα αποφυγής εμποδίων» καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μηνά Δασυγένη αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Ιωάννης Στολτίδης – Μηνάς Δασυγένης, 2017, Κοζάνη

Ευχαριστίες

Πρωτίστως θα ήθελα να ευχαριστήσω την οικογένεια μου, η οποία με στήριξε με κάθε τρόπο. Επίσης είμαι ευγνώμων στον καθηγητή μου Δρ. Μηνά Δασυγένη για την εμπιστοσύνη, την συνεργασία και τη βοήθεια του κατά την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας, αλλά και συνολικά στα έτη φοίτησής μου. Τέλος ευχαριστώ τους συμφοιτητές και καθηγητές μου για τη συνεργασία τους καθ' όλη την διάρκεια των σπουδών μου.

Κεφάλαιο 1 – Εισαγωγή

Ο όρος μη επανδρωμένα οχήματα εδάφους (unmanned ground vehicles) αναφέρεται σε οχήματα που λειτουργούν χωρίς την άμεση παρουσία ανθρώπου [1]. Χρήση τέτοιων οχημάτων γίνεται κυρίως σε συνθήκες κατά τις οποίες η παρουσία ανθρώπου καθίσταται επικίνδυνη ή αδύνατη. Τα οχήματα αυτά γενικότερα αποτελούνται από ένα σύνολο αισθητηρίων που χρησιμοποιούνται για την κατανόηση του περιβάλλοντος και είτε χειρίζονται εξ αποστάσεως από κάποιον χειριστή ή έχουν τη δυνατότητα να είναι πλήρως αυτόνομα και να παίρνουν τις δικές τους αποφάσεις. Ειδικά η δεύτερη κατηγορία περιλαμβάνει τεχνολογίες όπως τεχνητή νοημοσύνη και σύστημα πολλαπλών πρακτόρων, χαρακτηρίζοντας τα οχήματα αυτά ως «έξυπνα» (smart or intelligent cars) [2].

Επιπλέον λόγω της απότομης ανάπτυξης και διάδοσης τόσο, των ευρυζωνικών συνδέσεων, όσο και των μικροεπεξεργαστών, οδηγηθήκαμε στην ύπαρξη δικτύων από φυσικά αντικείμενα, οχήματα, κτήρια κ.α. Τα δίκτυα αυτά επιτρέπουν τις συνδεδεμένες οντότητες να ανταλλάσσουν πληροφορίες τόσο μεταξύ τους, όσο και με άλλα φυσικά πρόσωπα. Τα δίκτυα αυτά χαρακτηρίζονται ως Διαδίκτυο των Αντικειμένων (Internet of Things) [3].

Το πάντρεμα των δύο παραπάνω τεχνολογιών, δηλαδή, των έξυπνων μη επανδρωμένων οχημάτων, και του Δικτύου των Αντικειμένων, προσφέρει μεγάλη ευελιξία και πληθώρα εφαρμογών. Σε συνδυασμό μάλιστα με το χαμηλό κόστος κατασκευής και τη μεγάλη ταχύτητα πρόσβασης στο διαδίκτυο, βρίσκουμε εφαρμογές σε πολλούς τομείς όπως ταχυμεταφορές, στρατός, επιβολή νόμου, διάσωση και έρευνα.

1.1 Σημερινή Κατάσταση

Στην εποχή μας τα μη επανδρωμένα οχήματα χρησιμοποιούνται κυρίως από επαγγελματίες και χομπίστες. Μάλιστα αυτά που έχουν ευρέως διαδοθεί είναι τα τηλεκατευθυνόμενα οχήματα που χρησιμοποιούν άμεση επικοινωνία με τον χειριστή τους. Έτσι η εμβέλεια του συστήματος περιορίζεται από την τεχνολογία του σήματος που χρησιμοποιεί. Επακόλουθο αυτού είναι ο περιορισμένος αριθμός των εφαρμογών τέτοιων συστημάτων. Επιπλέον το μεγάλο κόστος αποτελεί επιβραδυντικό παράγοντα στην εξάπλωση τους. Τέλος στον μέσο καταναλωτή και στις μεσαίες επιχειρήσεις δεν εμφανίζονται αυτόνομα

οχήματα με την δυνατότητα να παίρνουν αποφάσεις, με εξαίρεση κάποια πολυτελή αυτοκίνητα τελευταίας τεχνολογίας που προσφέρουν κάποιες διευκολύνσεις.

Η απουσία μη επανδρωμένων οχημάτων (αυτόνομων και μη) σε εφαρμογές με στόχο την χρήση τους από το μέσο καταναλωτή και επιχείρηση είναι ένα κενό στην αγορά που προσπαθούν πολλοί να καλύψουν. Βρισκόμαστε σε ένα μεταβατικό στάδιο στο οποίο γίνεται προσπάθεια για την ανάπτυξη και την εισαγωγή αυτών των μηχανημάτων στην καθημερινότητα. Σημαντικός παράγοντας που πρέπει να ληφθεί υπ' όψη, κατά την υλοποίηση τέτοιων εφαρμογών, είναι ο τομέας της ασφάλειας τόσο με την έννοια του αγγλικού όρου «safety» όσο και με αυτήν του αγγλικού όρου «security».

Ένα παράδειγμα αποτελεί η υλοποίηση ενός μη επανδρωμένου ιπτάμενου οχήματος με σκοπό την αυτόματη μεταφορά φυσικών αντικειμένων από το ένα σημείο στο άλλο. Προφανώς ένα τέτοιο προϊόν πρέπει να είναι εξοπλισμένο με δικλίδες ασφαλείας έτσι ώστε, ούτε να προξενήσει κάποιο ατύχημα, αλλά ούτε να επιτρέψει την παραχώρηση πληροφοριών και δικαιωμάτων σε τρίτους που δεν έχουν την εξουσιοδότηση. Όπως για παράδειγμα να πάρει τον έλεγχο κάποιος ξένος που δεν έχει καμία σχέση με την υπηρεσία και δεν είναι πουθενά καταγεγραμμένη η ταυτότητά του, και να ρίξει το όχημα πάνω σε έναν στόχο με σκοπό κάποιον τραυματισμό.

Δεύτερο παράδειγμα αποτελεί η κατασκευή ενός έξυπνου οχήματος με τη δυνατότητα μεταφοράς συγκεκριμένου βάρους και όγκου, και παράλληλα μπορεί να ακολουθεί αυτόματα ένα άτομο [4]. Η συγκεκριμένη κατασκευή χρησιμοποιεί πολύ ακριβή τεχνολογία για τον εντοπισμό του στόχου, αλλά και για τη χαρτογράφηση του χώρου με αποτέλεσμα η τιμή του να είναι απαγορευτική τόσο για τον μέσο καταναλωτή όσο και για την μέση επιχείρηση. Επομένως είναι αδήριτη ανάγκη η εύρεση διαλλακτικών τεχνολογιών και καινοτόμων ιδεών με σκοπό την ασφαλή ένταξη των εφαρμογών που προαναφέρθηκαν με χαμηλότερο κόστος.

1.2 Κίνητρα για την εκπόνηση της Διπλωματικής Εργασίας

Η προαναφερθείσα ανάγκη για έρευνα μιας διαλλακτικής πρότασης σε συνδυασμό με την περιέργεια μου για το μυστήριο και την πολυπλοκότητα που κρύβονται πίσω από τέτοιες τεχνολογίες με οδήγησε να αναλάβω την υλοποίηση της παρούσας διπλωματικής εργασίας. Βασικό κίνητρο υπήρξε η ανάγκη χειρισμού ενός οχήματος μέσω σύνδεσης στο διαδίκτυο με

μεγάλο βαθμό ασφάλειας. Οι περισσότερες υπάρχουσες υλοποιήσεις αποφεύγουν τη χρήση διαδικτύου, αλλά και αυτές ακόμα που κάνουν χρήση πάσχουν από έλλειψη στοιχειώδους ασφάλειας. Ένα άλλο σημαντικό κίνητρο αποτέλεσε η ανάγκη ύπαρξης δικλίδων ασφαλείας ώστε να αποφευχθεί όσο το δυνατόν περισσότερο η σύγκρουση του οχήματος. Τέλος σοβαρό κίνητρο υπήρξε η μείωση του χρόνου απόκρισης του συστήματος και του κόστους υλοποίησης τέτοιων εφαρμογών, τα οποία αναλύονται στην επόμενη ενότητα.

1.3 Αντικείμενο της Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία επηρεασμένη από τα πολλά οφέλη του συνδυασμού των έξυπνων μη επανδρωμένων οχημάτων και του Δικτύου των Αντικειμένων που αναφέρθηκε προηγουμένως· αφορά την μετατροπή ενός απλού ηλεκτροκίνητου οχήματος για παιδιά, σε ένα έξυπνο όχημα με δυνατότητα χειρισμού μέσω σύνδεσης στο Διαδίκτυο, συνοδείας ατόμου και αποφυγής εμποδίων. Η υλοποίηση βασίζεται στην κάλυψη των αναγκών της ασφάλειας, της ευχρηστίας, της άμεσης απόκρισης και της οικονομίας. Η συγκεκριμένη τροποποίηση αποτελεί θεμελιώδη βάση για την προσθήκη επιπλέον μελλοντικών λειτουργιών και δυνατοτήτων.

1.4 Οφέλη της Διπλωματικής Εργασίας

Το αντικείμενο της παρούσας διπλωματικής αποτελεί μια διαλλακτική πρόταση στην ασφαλή και οικονομικότερη υλοποίηση μη επανδρωμένων, έξυπνων οχημάτων αλλά και γενικότερα έξυπνων συσκευών. Προσφέρεται ως αντικείμενο για την έρευνα περισσότερων μεθόδων αυτοματοποίησης π.χ. εύρεση καλύτερου αλγόριθμου κίνησης του οχήματος στον χώρο. Επιπλέον αποτελεί εργαλείο για την ανάπτυξη άλλων τεχνολογιών από μελλοντικούς συναδέλφους, όπως η χαρτογράφηση χώρου. Επιπρόσθετα αποτελεί βάση για προσθήκη επιπλέον δυνατοτήτων στον συγκεκριμένο (ή σε άλλο) όχημα προσφέροντας έτσι την δυνατότητα ακόμα και χρήσης του για τις καθημερινές ανάγκες του Πανεπιστημίου Δυτικής Μακεδονίας. Τέλος καθίσταται βάση για μελλοντικές μελέτες και εργασίες προσφέροντας εύρος γνωστικού πεδίου για έρευνα και ανάπτυξης τεχνολογίας από το Πανεπιστήμιο μας.

1.5 Σύνοψη της Διπλωματικής Εργασίας

Στο παρόν έγγραφο αναλύεται επιμελώς τόσο ο σχεδιασμός όσο και η κατασκευή του λογισμικού και υλικού μέρους της διπλωματικής εργασίας. Περιγράφονται, οι τεχνολογίες στις οποίες βασίστηκε η ανάπτυξη του λειτουργικού και υλικού μέρους, οι αδυναμίες του συστήματος, τα προβλήματα που προέκυψαν και οι λύσεις που εφαρμόστηκαν. Ταυτόχρονα αναφέρονται οι λειτουργίες, καθώς και οι τρόποι με τους οποίους μπορεί κάποιος να εκμεταλλευτεί πλήρως τις δυνατότητες που προσφέρει η παρούσα υλοποίηση.

Στο τρέχον κεφάλαιο βρίσκεται η περίληψη της διπλωματικής εργασίας, οι λόγοι που οδήγησαν στην εκπόνηση της, οι ανάγκες που έπρεπε να καλύψει και τέλος τα οφέλη που προσφέρει η υλοποίηση της.

Στο δεύτερο κεφάλαιο παρουσιάζονται αναλυτικά οι τεχνολογίες στις οποίες βασίστηκε και αναπτύχθηκε η διπλωματική εργασία καθώς και οι λόγοι που οδήγησαν στην επιλογή των συγκεκριμένων. Παρουσιάζονται οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν καθώς και τα πλεονεκτήματα της πλακέτας Raspberry Pi και Arduino.

Στο τρίτο κεφάλαιο παρουσιάζονται τα υλικά μέρη του συστήματος, η μεταξύ τους συνδεσμολογία, ο τρόπος λειτουργίας και επικοινωνίας καθώς και οι λόγοι επιλογής των συγκεκριμένων υλικών. Παράλληλα δίνονται τα σχεδιαγράμματα και τα σχηματικά των κυκλωμάτων.

Στο τέταρτο κεφάλαιο αναλύεται το λογισμικό μέρος του συστήματος, οι παρεχόμενες λειτουργίες και οι απαιτήσεις του. Ταυτόχρονα αναφέρονται σενάρια λειτουργίας, παραδείγματα χρήσης, καθώς και οι συμβάσεις που έγιναν πριν και κατά την διάρκεια της υλοποίησης. Τέλος παρουσιάζονται μετρήσεις εμβέλειας, αξιοπιστίας και απόκρισης του συστήματος.

Στο πέμπτο κεφάλαιο αναφέρονται προβλήματα που προέκυψαν κατά την υλοποίηση. Στη συνέχεια επισημαίνονται οι λύσεις που δόθηκαν για την αντιμετώπιση των προαναφερθέντων προβλημάτων, καθώς και οι βελτιστοποιήσεις που έγιναν για την καλύτερη απόκριση του συστήματος. Τέλος παρουσιάζονται τα συμπεράσματα από την υλοποίηση, μετρικές του κώδικα καθώς και μελλοντικές επεκτάσεις του συστήματος.

1.6 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο παρουσιάστηκαν οι προβληματισμοί που οδήγησαν στην εκπόνηση της συγκεκριμένης διπλωματικής, οι προϋποθέσεις που είναι αναγκαίο να καλύπτονται, τα οφέλη που παρέχονται καθώς και η δομή του συγκεκριμένου κειμένου. Ακολουθεί το επόμενο κεφάλαιο στο οποίο αναλύεται το θεωρητικό υπόβαθρο των τεχνολογιών στις οποίες στηρίζεται η υλοποίηση της διπλωματικής εργασίας. Ακολουθούν τα εργαλεία προγραμματισμού με την χρήση των οποίων αναπτύχθηκε το λογισμικό και τα προγράμματα σχεδιασμού.

Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο

Στο παρόν κεφάλαιο παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας διπλωματικής εργασίας. Εξηγούνται τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη των συγκεκριμένων λογισμικών, τόσο στο μέρος του πελάτη (client), όσο και σε αυτό του εξυπηρετή (server). Παράλληλα αναφέρονται τα υλικά που χρησιμοποιήθηκαν, τα χαρακτηριστικά τους και ο σκοπός που εξυπηρετεί το καθένα. Το συγκεκριμένο κεφάλαιο είναι απαραίτητο για την κατανόηση εννοιών που χρησιμοποιούνται στα επόμενα κεφάλαια.

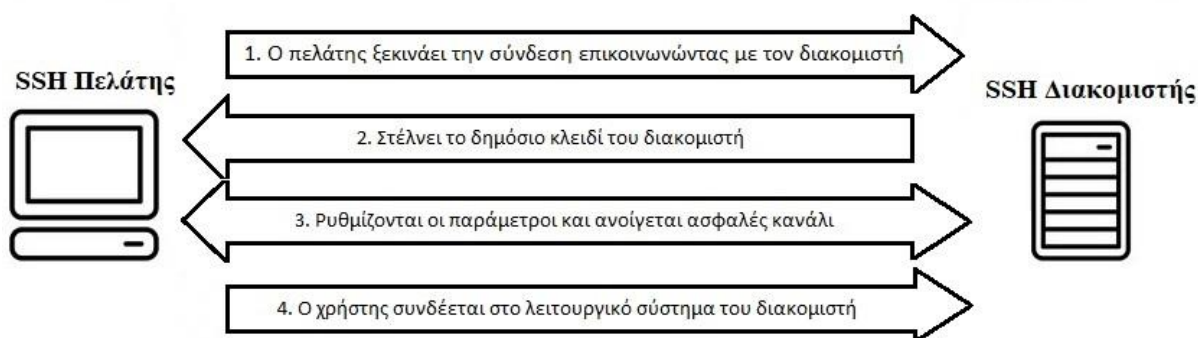
2.1 Πρόγραμμα χειρισμού του οχήματος

Η επικοινωνία του οχήματος με τον υπολογιστή του χειριστή γίνεται μέσω της σύνδεσης στο διαδίκτυο, ενώ για την μεταξύ τους επικοινωνία χρησιμοποιείται το πρωτόκολλο ασφαλούς φλοιού (Secure Shell, SSH). Για την υλοποίηση της εφαρμογής που εκτελείται σε Windows και είναι υπεύθυνη για τον έλεγχο του οχήματος, χρησιμοποιήθηκε η γλώσσα προγραμματισμού C++. Παράλληλα η χρήση της C++ συνδυάστηκε με άλλες τεχνολογίες και επεκτάσεις όπως πλαίσιο .NET, Visual C++, DX input, Xinput και OpenCV που αναλύονται στις παρακάτω ενότητες.

2.1.1 Ασφαλής φλοιός SSH

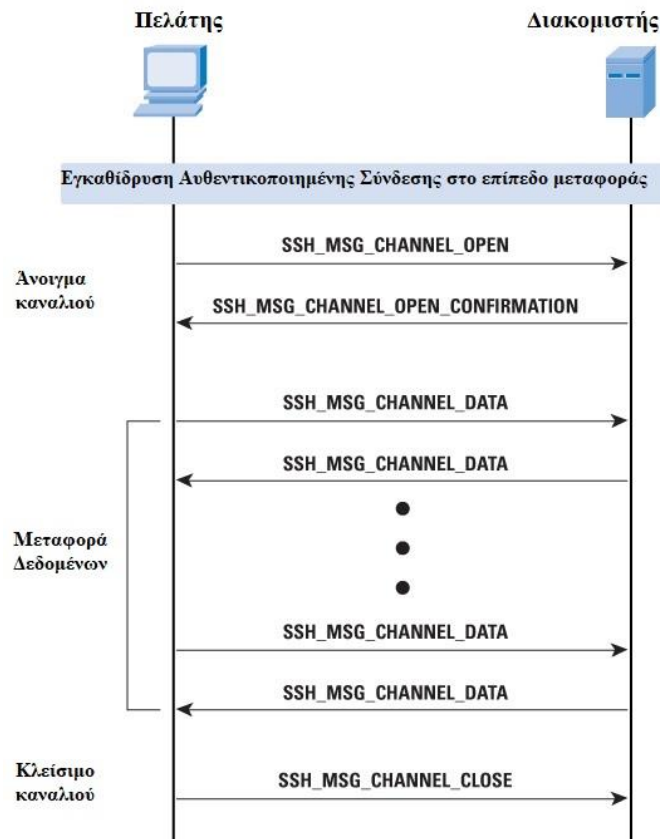
Ο ασφαλής φλοιός (Secure Shell, SSH) είναι ένα δικτυακό πρωτόκολλο που επιτρέπει την καθίδρυση κρυπτογραφημένης και ασφαλούς επικοινωνίας μεταξύ δύο κόμβων σε ένα μη ασφαλές δίκτυο, όπως είναι το διαδίκτυο [5]. Παρέχει ένα ασφαλές κανάλι βασισμένο στην αρχιτεκτονική πελάτη-εξυπηρετή, συνδέοντας με αυτόν τον τρόπο έναν SSH πελάτη με έναν SSH εξυπηρετή. Χαρακτηριστικό παράδειγμα αποτελεί η σύνδεση (με την έννοια του αγγλικού όρου log in) ενός χρήστη σε έναν απομακρυσμένο υπολογιστή. Παράλληλα επιτρέπει την απομακρυσμένη εκτέλεση εντολών καθώς και την ασφαλή μεταφορά αρχείων από μία μηχανή σε άλλη.

Συγκεκριμένα η διαδικασία σύνδεσης (**Εικόνα 1**), ξεκινάει με τον πελάτη να αποστέλλει αίτηση αυθεντικοποίησης στον εξυπηρετή. Ο εξυπηρετής με την σειρά του επιστρέφει στον πελάτη το δημόσιο κλειδί του. Στην συνέχεια εγκαθιδρύονται οι παράμετροι κρυπτογράφησης επιλέγοντας ένα τυχαίο αριθμό 256 bits, που αποτελεί το κλειδί συνόδου, και έναν αλγόριθμο κρυπτογράφησης. Μόλις ολοκληρωθεί αυτή η ανταλλαγή πληροφοριών αποστέλλεται ένα μήνυμα επιβεβαίωσης από τον εξυπηρετή στον πελάτη. Ο πελάτης στη συνέχεια ξεκινάει την διαδικασία σύνδεσης (slogin) του χρήστη στο λειτουργικό σύστημα του εξυπηρετή.



Εικόνα 1: Αυθεντικοποίηση SSH

Από αυτό το σημείο και έπειτα η επικοινωνία μεταξύ πελάτη και εξυπηρετή γίνεται κρυπτογραφημένα μέσω του καναλιού που δημιουργήθηκε από την παραπάνω διαδικασία. Η ανταλλαγή κρυπτογραφημένων μηνυμάτων συνεχίζεται μέχρι ο πελάτης να κλείσει το κανάλι. Στην **Εικόνα 2** βλέπουμε ένα παράδειγμα κρυπτογραφημένης επικοινωνίας με χρήση SSH. Η χρήση του SSH πρωτοκόλλου στην εφαρμογή γίνεται με την συμπερίληψη της βιβλιοθήκης ανοιχτού κώδικα libssh 0.7.1.



Εικόνα 2: Βασική χρήση SSH

2.1.2 OpenCV

Η OpenCV (Open Source Computer Vision, Όραση Μηχανής Ανοιχτού Κώδικα) είναι μία βιβλιοθήκη προγραμματισμού ανοιχτού κώδικα, η οποία στοχεύει στην πραγματικού χρόνου παρακολούθηση βίντεο από πλήθος συσκευών εισόδου [6]. Περιέχει διεπαφές σε αρκετές γνωστές γλώσσες προγραμματισμού όπως C, C++, Python και Java, ενώ υποστηρίζει τα περισσότερα διαδεδομένα λειτουργικά συστήματα όπως Windows, Linux, Mac OS, iOS και Android. Επιπρόσθετα παρέχει τη δυνατότητα επεξεργασίας πολλαπλών πυρήνων, σε περίπτωση βελτιστοποιημένης εγγραφής κώδικα με C ή C++. Η OpenCV χρησιμοποιήθηκε για την ζωντανή παρακολούθηση από την κάμερα που έχει τοποθετηθεί στο όχημα. Γι' αυτή την υλοποίηση ενσωματώθηκε η βιβλιοθήκη προγραμματισμού OpenCV v3.1.0.

2.1.3 DirectX input και Xinput

Για την είσοδο πληροφοριών από το χειριστήριο χρησιμοποιήθηκαν δύο τεχνολογίες, η DirectX input και η Xinput. Η DirectInput αποτελεί μέρος της προγραμματιστικής διεπαφής

εφαρμογών (Application programming interface, API) DirectX της Microsoft [7]. Σκοπός της είναι η επεξεργασία δεδομένων που προέρχονται από εξωτερικές συσκευές όπως το πληκτρολόγιο, το ποντίκι και διάφορα χειριστήρια παιχνιδιών [8]. Η Xinput αποτελεί επίσης προϊόν της Microsoft και χαρακτηρίζεται ως ξεχωριστή προγραμματιστική διεπαφή εφαρμογών [9]. Επιτρέπει στις εφαρμογές που την χρησιμοποιούν, την λήψη εισόδου από χειριστήρια της οικογένειας Xbox 360, και γενικότερα από χειριστήρια που χαρακτηρίζονται ως «επόμενης γενιάς». Εφόσον η DirectX Input υποστηρίζει μερικά χειριστήρια, ενώ η Xinput κάποια άλλα, για λόγους υποστήριξης μεγάλης ποικιλίας χειριστηρίων, αποφασίστηκε η χρήση και των δύο. Για την χρήση της Xinput χρησιμοποιήθηκε η βιβλιοθήκη προγραμματισμού ανοιχτού κώδικα της Nvidia NvGamepad v1.0 [10].

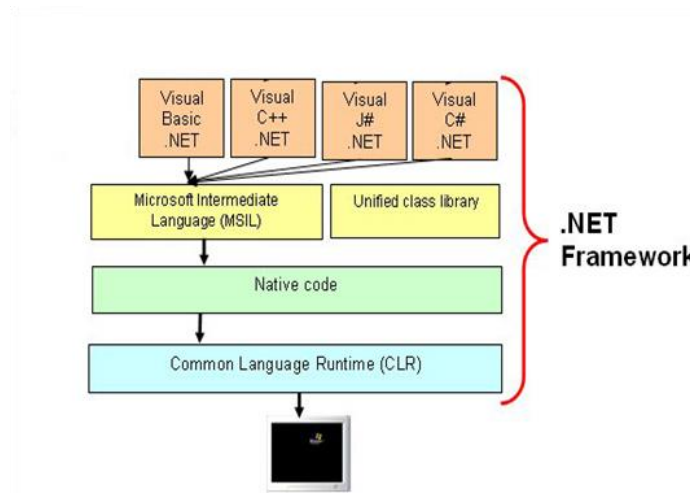
2.1.4 Visual C++ και πλαίσιο .NET

Η Microsoft Visual C++ (MSVC) είναι γλώσσα προγραμματισμού της Microsoft που βασίζεται στις γλώσσες C και C++ [11]. Αρχικά αποτελούσε αυτόνομο προϊόν αλλά πλέον έχει ενσωματωθεί στον αποσφαλματωτή Visual Studio, προσφέροντας έτσι μια πιο ολοκληρωμένη σουίτα λογισμικού [12]. Εμπεριέχει εργαλεία ανάπτυξης και αποσφαλμάτωσης κώδικα γραμμένο σε C++, ενώ η εκτέλεση τέτοιων παραγόμενων εφαρμογών απαιτεί από τον χρήστη την εγκατάσταση ειδικών αναδιανεμητέων Visual C++ πακέτων. Η Visual C++ είναι μία από τις τρεις κύριες γλώσσες προγραμματισμού που υποστηρίζει το πλαίσιο .NET της Microsoft, ενώ οι υπόλοιπες δύο είναι η Visual Basic και η Visual C# [12].

Το .NET είναι πλαίσιο λογισμικού, το οποίο παρέχει στις εκτελέσιμες εφαρμογές ένα πλήθος από υπηρεσίες [13]. Η χρήση του πλαισίου .NET επιτρέπει στους προγραμματιστές να γράφουν μέρη του λογισμικού σε όποια γλώσσα προγραμματισμού, είναι πιο εξοικειωμένοι, από τις γλώσσες που προσφέρει (Microsoft Intermediate Language, MSIL) [11]. Επιπρόσθετα παρέχει την λειτουργία κοινής γλώσσας χρόνου εκτέλεσης (Common Language Runtime, CLR), προσφέροντας με αυτόν τον τρόπο μια μηχανή που χειρίζεται τις εκτελέσιμες εφαρμογές. Η αλληλεπίδραση του πλαισίου .NET με τις γλώσσες προγραμματισμού που υποστηρίζει (MSIL), την λειτουργία κοινής γλώσσας χρόνου εκτέλεσης (CLR) και του κύριου κώδικα φαίνεται στην **Εικόνα 3**.

Στην υλοποίηση της εφαρμογής ελέγχου του οχήματος χρησιμοποιήθηκε το πλαίσιο .NET v4.5.2 με σκοπό την διευκόλυνση των συναδέρφων που θα επιλέξουν να συνεχίσουν το


παρόν έργο. Επιπλέον για την δημιουργία του γραφικού περιβάλλοντος επιλέχθηκε η Visual C++ λόγω της εξοικείωσης μου με την C++.




Εικόνα 3: Στοιχεία του πλαισίου .NET

2.2 Η πλακέτα Raspberry Pi

Η «καρδιά» του υλικού μέρους του συστήματος αποτελείται από έναν υπολογιστή πλακέτας Raspberry Pi. Ουσιαστικά είναι ένας ολοκληρωμένος υπολογιστής με χαμηλές απαιτήσεις ρεύματος, ενώ το λειτουργικό του σύστημα είναι αποθηκευμένο σε μία κάρτα μνήμης τύπου SD [14]. Είναι χαμηλού κόστους, με μέτρια (για την εποχή μας) υπολογιστική ισχύ, ενώ παράλληλα προσφέρει την δυνατότητα διαχείρισης των ακροδεκτών γενικού σκοπού εισόδου-εξόδου (General Purpose Input Output, GPIO) που διαθέτει. Βασικά χαρακτηριστικά αποτελούν ο επεξεργαστής τύπου ARM και η μνήμη RAM που διαθέτει, ενώ προσφέρει πλήθος διεπαφών εισόδου-εξόδου όπως φαίνεται στον **Πίνακας 1**.

Μοντέλα Raspberry Pi	Χαρακτηριστικά
 <p>Εικόνα 4: Raspberry Pi Model B</p>	<p>CPU/GPU: ARM11 @700MHz ARMv6 / 250MHz VideoCore IV Μνήμη αποθήκευσης/RAM: Micro-SD / 512 MB SDRAM Ethernet: 10/100 Mbps Είσοδοι/Εξοδοι: 2x USB, HDMI /Composite, Mini-Jack 3.5mm, 26 GPIO, CSI σύνδεση κάμερας, DSI σύνδεση οθόνης</p>

 <p>Εικόνα 5: Raspberry Pi 3 Model B</p>	<p>CPU/GPU: Quad Cortex A53 @1.2GHz ARMv8-A / 400MHz VideoCore IV Μνήμη αποθήκευσης/RAM: Micro-SD / 1 GB SDRAM Ethernet: 10/100 Mbps Ασύρματη Επικοινωνία: 802.11 n ,Bluetooth 4.1 Είσοδοι/Εξοδοι: 4x USB, HDMI /Composite, Mini-Jack 3.5mm, 40 GPIO, CSI σύνδεση κάμερας, DSI σύνδεση οθόνης</p>
---	---

Πίνακας 1: Χαρακτηριστικά μοντέλων Raspberry Pi

Επιπλέον όλα τα μοντέλα διαθέτουν μια επιπλέον θύρα micro USB που είναι υπεύθυνη για τη τροφοδοσία της πλακέτας. Στην υλοποίηση της παρούσας διπλωματικής χρησιμοποιήθηκε το Raspberry Pi 3 Model B με λειτουργικό σύστημα Raspbian Jessie Lite 4.4.34-v7+. Το Raspbian είναι το επίσημο λειτουργικό σύστημα που υποστηρίζεται από την εταιρία Raspberry Pi για της πλακέτες της και είναι βασισμένο στην Debian διανομή λειτουργικού Linux. Παρόλο που η υλοποίηση έγινε σε Raspberry Pi 3 Model B το όλο σύστημα δοκιμάστηκε και είναι πλήρως λειτουργικό και στην πλακέτα Raspberry Pi Model B που έχει αρκετά μικρότερη επεξεργαστική ισχύ, χρησιμοποιώντας το ίδιο λειτουργικό σύστημα. Τέλος προτιμήθηκε η συγκεκριμένη πλακέτα από κάποια άλλη, όπως η BeagleBoard-Xm rev C, λόγω της πολύ καλής υποστήριξης της πρώτης σε σύγκριση με την δεύτερη. Το λογισμικό που χρησιμοποιήθηκε στο Raspberry Pi είναι υλοποιημένο σε Python όπως αναλύεται στην επόμενη ενότητα. Τέλος χρησιμοποιήθηκε και ένα λογισμικό από τρίτους, με την ονομασία Motion, το οποίο είναι υπεύθυνο για την ζωντανή μετάδοση εικόνας από την κάμερα που είναι συνδεδεμένη στο Raspberry Pi.

2.3 Python

Η Python είναι μια υψηλού επιπέδου, γενικού σκοπού γλώσσα προγραμματισμού [15]. Επιπλέον χαρακτηρίζεται ως αντικειμενοστραφής με δυνατότητα χρήσης διερμηνέα [16]. Στην συγκεκριμένη υλοποίηση χρησιμοποιήθηκε η ιδιότητα αυτή, με σκοπό την εκμετάλλευση της ευελιξίας που προσφέρει. Ουσιαστικά το λογισμικό που δημιουργήθηκε με την Python χρησιμοποιεί τον διερμηνέα της συγκεκριμένης γλώσσας για να εκτελεστεί στο λειτουργικό που το φιλοξενεί [17]. Αποτέλεσμα αυτού είναι η εύκολη προσθήκη επιπλέον λειτουργιών, η γρήγορη αποσφαλμάτωση και η άμεση διευκόλυνση στην κατανόηση της λειτουργίας του συστήματος από τρίτους.

Η χρήση διερμηνέα για ένα σύστημα έχει ως κύριο μειονέκτημα την χαμηλότερη ταχύτητα εκτέλεσης σε σύγκριση με την χρήση ενός εκτελέσιμου που έχει παραχθεί από μεταγλωττιστή. Αυτή η διαφορά στην ταχύτητα εκτέλεσης στο συγκεκριμένο σύστημα που υλοποιήθηκε είναι αμελητέα και δεν επηρεάζει το τελικό αποτέλεσμα. Ένα δεύτερο σημαντικό μειονέκτημα της χρήσης διερμηνέα, αποτελεί η έλλειψη στατικού ελέγχου του τύπου των μεταβλητών, αντικειμένων, συναρτήσεων και κλάσεων, με αποτέλεσμα να είναι απαραίτητη η ακόμα μεγαλύτερη προσοχή κατά την συγγραφή του κώδικα.

Η Python περιέχει μεγάλη ποικιλία βιβλιοθηκών και παρέχει υψηλή υποστήριξη της πλακέτας Raspberry Pi. Χρησιμοποιήθηκε η Python v2.7.9 για την υλοποίηση του λογισμικού που εκτελείται στην πλακέτα, ενώ παράλληλα συμπεριλήφθηκαν βιβλιοθήκες για την εκμετάλλευση των δυνατοτήτων της πλακέτας όπως η χρήση GPIO και Bluetooth.

2.4 Η Πλακέτα Arduino και το υλικό του οχήματος

Εκτός από την πλακέτα Raspberry Pi το υλικό μέρος του συστήματος περιέχει και τρεις πλακέτες Arduino Uno R3 για την διαχείριση των υπόλοιπων συσκευών και αισθητήρων. Η πλακέτα Arduino σε αντίθεση με το Raspberry Pi δεν αποτελεί ολοκληρωμένο υπολογιστή αλλά πλακέτα μικροελεγκτή [18]. Ο κύριος μικροελεγκτής με τον οποίον είναι εξοπλισμένη η πλακέτα είναι αρχιτεκτονικής AVR της Atmel, ενώ για την εκμετάλλευση των λειτουργιών του περιέχει ακροδέκτες GPIO.

Μερικοί ακροδέκτες GPIO επιτρέπουν τη διαχείριση σήματος διαμόρφωσης εύρους παλμού (Pulse Width Modulation, PWM), κάτι που δεν υποστηρίζει η πλακέτα Raspberry Pi. Από την άλλη στο Arduino δεν υπάρχει η δυνατότητα ύπαρξης νημάτων για υποστήριξη παραλληλίας όπως συμβαίνει με το Raspberry Pi [19]. Επιπλέον η πλατφόρμα Arduino είναι ανοιχτού κώδικα δίνοντας την δυνατότητα ύπαρξης οικονομικότερων επιλογών στην αγορά [20].





Στην παρούσα υλοποίηση χρησιμοποιήθηκαν δύο κλώνοι του Arduino Uno R3 και ένα γνήσιο Arduino Uno R3, μειώνοντας έτσι κατά πολύ το κόστος. Επιλέχθηκε η χρήση δύο Arduino Uno R3 στην κατασκευή του οχήματος αντί για κάποιο μεγαλύτερο (πχ. Arduino Mega) με σκοπό να εξασφαλιστεί η καλύτερη απόκριση του συστήματος προσφέροντας έτσι ένα είδους παραλληλίας, ενώ το τρίτο αποτελεί παρελκόμενο για τον χρήστη.


Στις πλακέτες Arduino υλοποιήθηκαν λογισμικά για τον έλεγχο του οχήματος, την υποστήριξη της λειτουργίας συνοδείας ατόμου, καθώς και μικροεφαρμογές άμεσης απόκρισης που λειτουργούν ως δικλίδες ασφαλείας. Για την υλοποίηση αυτή χρησιμοποιήθηκε η γλώσσα προγραμματισμού Wiring που είναι βασισμένη στη C++ προσφέροντας δυνατότητες της C και αποτελεί την βασική γλώσσα προγραμματισμού της πλακέτας Arduino [20]. Παράλληλα ενσωματώθηκαν βιβλιοθήκες από τρίτους που επιτρέπουν την ευκολότερη και αποδοτικότερη εκμετάλλευση αισθητηρίων, όπως η Arduino-IRremote για την εκμετάλλευση πολλαπλών αισθητήρων υπέρυθρης ακτινοβολίας και αντίστοιχα η NewPing για αισθητήρες υπέρηχων [21] [22].

Το υπόλοιπο υλικό μέρος αποτελείται από στοιχεία που αλληλοεπιδρούν είτε με το Raspberry Pi είτε με κάποιο Arduino. Άξια αναφοράς είναι η πλακέτα ελέγχου κινητήρα διπλής γέφυρας H L298N LC (Dual H Bridge L298N LC Motor Controller), η οποία βασίζεται στο ολοκληρωμένο L298N και αποτελεί την οικονομικότερη και πλήρως λειτουργική επιλογή για την διαχείριση των δύο DC κινητήρων του συστήματος από το Arduino [23]. Επιπλέον για την περιστροφή της κάμερας χρησιμοποιήθηκαν δύο σερβοκινητήρες SG90 οι οποίοι είναι οικονομικοί και χειρίζονται εύκολα από το Arduino [24]. Για τον εντοπισμό εμποδίων χρησιμοποιήθηκαν συνολικά πέντε αισθητήρες υπέρηχων δύο ειδών, οι HC-SR04 και οι US-020 με παρόμοια χαρακτηριστικά, ενώ επιλέχθηκαν επειδή παρέχουν την επιθυμητή εμβέλεια (2 μέτρα) σε προσιτή τιμή [25].

Για την λήψη σήματος υπέρυθρης ακτινοβολίας (infrared, IR) χρησιμοποιήθηκαν τέσσερις δέκτες IR VS1838b 38KHz, οι οποίοι καλύπτουν τις ανάγκες της διπλωματικής και αποτελούν οικονομική λύση [26]. Επιπλέον ως οικονομική λύση επιλέχθηκε η οθόνη υγρών κρυστάλλων LCD 16x2 HD44780 για προβολή πληροφοριών από το Raspberry Pi. Για την τροφοδοσία του συστήματος χρησιμοποιήθηκε η μπαταρία MW MWS 7.2-12 των 12V και 7.2Ah λόγω της μεγάλης αυτονομίας που προσφέρει [27]. Παράλληλα για τον μετασχηματισμό της τάσης από 12V σε 5V χρησιμοποιήθηκε ο μετατροπέας τάσης LM2596s και ένας USB αντάπτορας φόρτισης για αυτοκίνητα [28]. Ο πρώτος επιλέχθηκε λόγω της αντοχής του σε ρεύμα μέχρι 3A, ενώ ο δεύτερος λόγω της άμεσης διαθεσιμότητας του στο εργαστήριο ψηφιακών συστημάτων και αρχιτεκτονικής υπολογιστών. Για την κίνηση του οχήματος χρησιμοποιήθηκε η εργοστασιακή του μπαταρία 3FM4.5 των 6V και 4.5Ah λόγω σίγουρης συμβατότητάς της με τους κινητήρες του οχήματος για τους οποίους δεν μπόρεσαν να βρεθούν πληροφορίες [29]. Τέλος χρησιμοποιήθηκε για τον πομπό, που έχει ο χρήστης, το Bluetooth

BC417 το οποίο είναι συμβατό με το Arduino και παρέχει την επιθυμητή εμβέλεια [30]. Τα τεχνικά χαρακτηριστικά των σημαντικότερων υλικών παρουσιάζονται στον **Πίνακα 2**.

Υλικό	Τεχνικά Χαρακτηριστικά
 <p>Εικόνα 6: Arduino Uno</p>	<p>Atmel ATmega328 & Atmel ATmega16U2 για programmer Τάση λειτουργίας: 5V Τάση εισόδου (όρια/προτεινόμενη): 7-12V/6-20V Είσοδοι/έξοδοι: 14 ψηφιακές (6 με υποστήριξη PWM), 6 αναλογικές εισοδοι Μνήμη Flash/SRAM/EEPROM: 32KB/2KB/1KB Συχνότητα ρολογιού: 16MHz</p>
 <p>Εικόνα 7: Dual H bridge L298N</p>	<p>Τάση σήματος: 5V Ρεύμα σήματος: 0 mA-36 mA Τάση οδήγησης: 5V-35V Ρεύμα οδήγησης μέχρι: 2A Μέγιστη Ισχύς: 25W</p>
 <p>Εικόνα 8: LM2596s DC-DC</p>	<p>Τάση εισόδου: 3-40V Τάση εξόδου: 1.5-35V Μέγιστο ρεύμα εξόδου: 3A Θερμοκρασίες λειτουργίας: -40 με 85° C Αποδοτικότητα μετατροπής: 92%</p>
 <p>Εικόνα 9: HC-SR04</p>	<p>Τάση λειτουργίας: 5V Ρεύμα λειτουργίας: 3mA Θερμοκρασίες λειτουργίας: 0 με 70° C Γωνία πρόσπτωσης: λιγότερο από: 15° Απόκλιση: 0.3cm + 1% Εμβέλεια εντοπισμού: 2cm-2m</p>

 <p data-bbox="355 394 668 421">Εικόνα 10: Bluetooth BC417</p>	<p data-bbox="890 203 1278 230">Μέγιστη τάση λειτουργίας: 5.6V</p> <p data-bbox="890 232 1310 259">Προτεινόμενη τάση λειτουργίας: 3.6V</p> <p data-bbox="959 262 1241 288">Ρεύμα λειτουργίας: 35mA</p> <p data-bbox="943 291 1257 318">Πρωτόκολλο Bluetooth: v2.0</p> <p data-bbox="943 320 1257 347">Ευαισθησία λήψης: -85 dBm</p> <p data-bbox="850 349 1350 376">Εύρος συχνότητας κύματος: 2.4GHz - 2.8GHz</p>
---	---

Πίνακας 2: Χαρακτηριστικά υλικών που χρησιμοποιήθηκαν

2.5 Προγράμματα ανάπτυξης

Για την ανάπτυξη του λογισμικού μέρους αυτής της διπλωματικής εργασίας χρησιμοποιήθηκε πλήθος εργαλείων. Στις παρακάτω ενότητες αναφέρονται αναλυτικά τα εργαλεία αυτά, καθώς και ο σκοπός του καθένα στην εργασία.

2.5.1 Microsoft Visual Studio

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο αναπτυξιακό περιβάλλον, το οποίο χρησιμοποιείται για την ανάπτυξη εφαρμογών σε Windows [31]. Ενσωματώνει κειμενογράφο, αποσφαλματωτή και εργαλείο σχεδιασμού. Για το σχεδιασμό του γραφικού περιβάλλοντος χρησιμοποιήθηκε ο ενσωματωμένος σχεδιαστής για C++ (Visual C++). Λόγω χρήσης της Visual C++ απαραίτητη προϋπόθεση για την σωστή λειτουργία της εφαρμογής, είναι η ύπαρξη προ εγκαταστημένων αναδιανεμητέων Visual C++ πακέτων. Για τον σκοπό αυτό υλοποιήθηκε ένας εγκαταστάτης της εφαρμογής, με την χρήση της επέκτασης InstallShield Limited Edition Project. Η έκδοση που χρησιμοποιήθηκε είναι η Visual Studio 2015 ενώ η αναλυτική χρήση του περιγράφεται σε παρακάτω κεφάλαιο.

2.5.2 Arduino IDE

Το Arduino IDE είναι το επίσημο ολοκληρωμένο αναπτυξιακό περιβάλλον της Arduino για τις πλακέτες της [32]. Επιτρέπει την συγγραφή κώδικα μέσω του κειμενογράφου που προσφέρει, ενώ παρέχει δυνατότητες μεταγλώττισης και φόρτωσης της υλοποίησης στην πλακέτα. Επιπρόσθετα υπάρχει δυνατότητα εισαγωγής βιβλιοθήκης από τρίτους ενώ υπάρχουν διανομές του προγράμματος για Windows, Linux και Mac OS X. Στην συγκεκριμένη υλοποίηση χρησιμοποιήθηκε το Arduino IDE 1.6.13.

2.5.3 Notepad++

Για την συγγραφή κώδικα σε Python χρησιμοποιήθηκε ο κειμενογράφος Notepad++ v7.3.3. Αποτελεί μια πολύ καλή αντικατάσταση του ενσωματωμένου σημειωματάριου που παρέχουν τα Windows επειδή αναγνωρίζει πολλές γλώσσες προγραμματισμού [33]. Επιπλέον είναι ελεύθερης διανομής και ανοιχτού κώδικα αποτέλεσμα του οποίου είναι η ύπαρξη πολλών επεκτάσεων. Τέλος προσφέρει αυτόματη στοίχιση και χρωματισμό στα διάφορα στοιχεία της γλώσσας κάνοντας έτσι την συγγραφή και ανάγνωση του κώδικα ευκολότερη [20].

2.5.4 SketchUp

Το SketchUp αποτελεί μία εύκολη λύση για σχεδιασμό τρισδιάστατου μοντέλου [34]. Επι της ουσίας είναι ένα λογισμικό που παρέχει πλήθος εργαλείων για τον σχεδιασμό αντικειμένων σε τρισδιάστατο χώρο. Παράλληλα προσφέρει ένα πλήθος επιπρόσθετων εφαρμογών, παρέχοντας έτσι περισσότερες δυνατότητες. Τέλος δίνει την δυνατότητα εξαγωγής αρχείου σε πολλές μορφές και με μεγάλη ποικιλία κωδικοποιήσεων, έτσι ώστε να υπάρχει συμβατότητα με πολλά μοντέλα 3D εκτυπωτών. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε η δοκιμαστική έκδοση του SketchUp 2016 για τον σχεδιασμό της θήκης Arduino του πομπού.

2.5.5 Win32DiskImager

Για την εγκατάσταση λειτουργικού στην κάρτα μνήμης SD που διαθέτει το Raspberry Pi, χρησιμοποιήθηκε το Win32DiskImager. Το συγκεκριμένο εργαλείο είναι ένα εκτελέσιμο πρόγραμμα σε Windows, το οποίο δίνει την δυνατότητα εγκατάστασης λειτουργικού σε κάποιο εξωτερικό μέσο αποθήκευσης όπως είναι η κάρτα SD [35]. Επιπλέον διαμορφώνει κατάλληλα το αποθηκευτικό μέσο πριν την εγκατάσταση, ενώ παρέχει την δυνατότητα δημιουργίας αντιγράφου ασφαλείας από το μέσο αποθήκευσης (κάρτα SD) στον τρέχον H/Y.

2.6 Σύνοψη κεφαλαίου

Στο συγκεκριμένο κεφάλαιο παρουσιάστηκαν οι τεχνολογίες που χρησιμοποιήθηκαν στο υλικό και λογισμικό μέρος της υλοποίησης της παρούσας διπλωματικής εργασίας. Επιπλέον αναφέρθηκαν τα εργαλεία που χρησιμοποιήθηκαν σε αυτήν την υλοποίηση. Ακολουθεί το επόμενο κεφάλαιο στο οποίο παρουσιάζονται τα υλικά μέρη του συστήματος, η μεταξύ τους συνδεσμολογία και το τρόπος λειτουργίας του υλικού μέρους.

Κεφάλαιο 3 – Ανάλυση υλικού μέρους

Στο παρόν κεφάλαιο παρουσιάζεται διεξοδικά το υλικό μέρος του συστήματος. Αναλύεται η λειτουργία του κάθε υλικού και λόγος επιλογής του συγκεκριμένου. Το κεφάλαιο αυτό χωρίζεται σε δύο κύριες ενότητες, εκ των οποίων η πρώτη αφορά το υλικό του οχήματος ενώ η δεύτερη την κατασκευή του πομπού. Ακολουθούν άλλες δύο μικρότερες ενότητες που αφορούν το κόστος κατασκευής και συνοψίζουν τις πληροφορίες που αναφέρθηκαν σε αυτό το κεφάλαιο. Το συγκεκριμένο μέρος του κειμένου παρέχει όλες της απαραίτητες πληροφορίες για την κατανόηση της δομής του υλικού μέρους και του τρόπου λειτουργίας του.

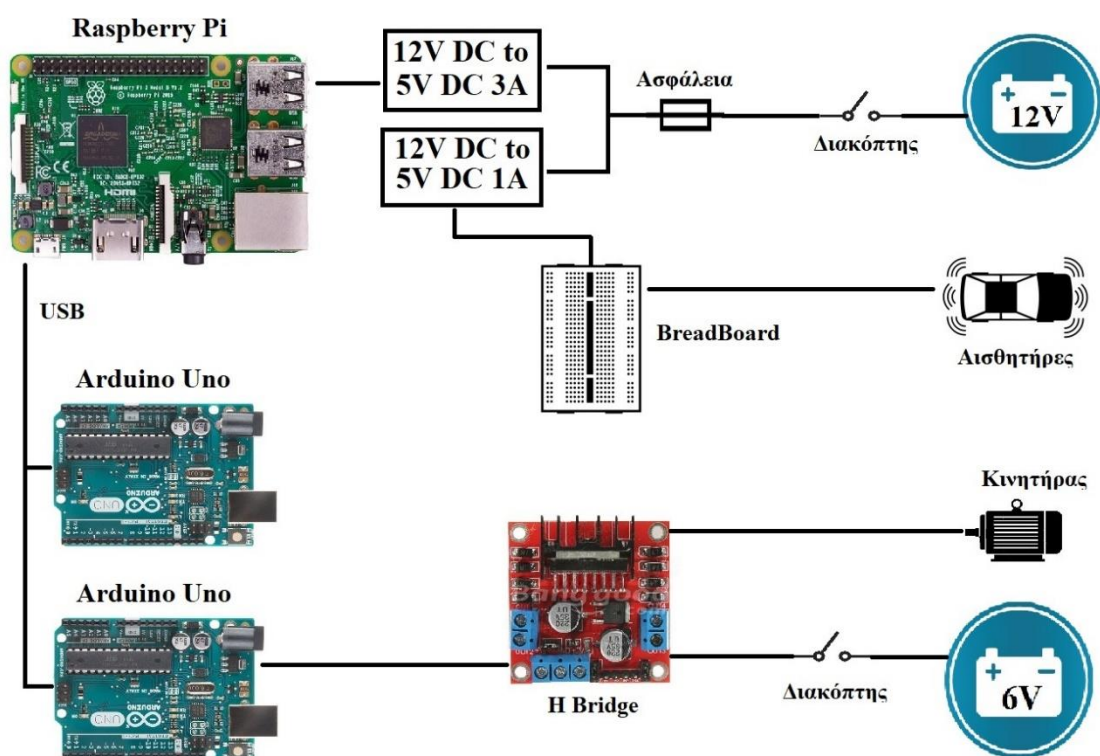
3.1 Υλικό μέρος οχήματος

Στην συγκεκριμένη ενότητα θα μας απασχολήσει το σύστημα που έχει υλοποιηθεί στο όχημα. Είναι το μεγαλύτερο από τα δύο συστήματα που έχουν υλοποιηθεί, τόσο από άποψη πολυπλοκότητας, όσο και από πλήθος υλικών. Ο πυρήνας του κύριου συστήματος είναι ένα Raspberry Pi 3 Model B το οποίο επικοινωνεί άμεσα και έμμεσα με τα ηλεκτρονικά υλικά της κατασκευής. Άμεσα επικοινωνεί μόνο με τις διόδους εκπομπής φωτός (light-emitting diode, LED) και με την οθόνη υγρών κρυστάλλων (LCD), ενώ έμμεσα με όλα τα υπόλοιπα υλικά μέσω των δύο Arduino Uno R3. Επιπλέον τα Arduino υλοποιούν και άλλα μικρότερα υποσυστήματα προσφέροντας με αυτόν τον τρόπο καλύτερη απόκριση του συστήματος. Συγκεκριμένα το σύστημα του οχήματος περιλαμβάνει τα εξής υλικά στοιχεία:

- Γέφυρα H L298N
- Οθόνη υγρών κρυστάλλων LCD 16x2 HD44780
- Διακόπτης – μπουτόν
- Δίοδοι Εκπομπής Φωτός (LED)
- Αισθητήρας Υπέρυθρης ακτινοβολίας VS1838b 38KHz
- Αισθητήρας Υπέρηχων HC-SR04/US-020
- Μετατροπέας 12V σε 5V LM2596s
- Σερβοκινητήρας SG90
- Ηλεκτροκινητήρας συνεχούς ρεύματος (ο εργοστασιακός του οχήματος)
- Διαδικτυακή Κάμερα Logilink Ua0067
- Μπαταρίες τροφοδοσίας MW MWS 7.2-12/3FM4.5

3.1.1 Περί τροφοδοσίας του συστήματος

Ένα απ' τα βασικά μέρη ενός συστήματος είναι η τροφοδοσία του με ηλεκτρικό ρεύμα με σκοπό την λειτουργία του. Για την κίνηση του οχήματος χρησιμοποιήθηκε η εργοστασιακή του μπαταρία 3FM4.5, η οποία προσφέρει μία ώρα και τριάντα λεπτά περίπου αυτονομία κίνησης, ενώ απαιτεί οχτώ με δεκαπέντε (8-15) ώρες φόρτισης σύμφωνα με τις οδηγίες του κατασκευαστή. Παράλληλα προστέθηκε και μια δεύτερη μπαταρία MW MWS 7.2-12, η οποία με την χρήση μετατροπέων τάσης τροφοδοτεί το υπόλοιπο σύστημα. Με αυτόν τον τρόπο δεν μειώνεται η προαναφερθείσα αυτονομία κίνησης, ενώ παρέχεται επιπλέον αυτονομία συστήματος 10 ωρών για παρακολούθηση μέσω της κάμερας, χειρισμό προσανατολισμού αυτής, και ανάμμα φώτων. Η συγκεκριμένη μπαταρία φορτίζεται μέσω ενός UPS που είναι ανεξάρτητο από το σύστημα και απαιτεί περίπου τέσσερις με πέντε ώρες φόρτισης.



Εικόνα 11: Τροφοδοσία συστήματος οχήματος

Όπως παρουσιάζεται στην **Εικόνα 11** μετά τον διακόπτη που είναι τοποθετημένος στην μπαταρία MW MWS 7.2-12 των 12V έχει συνδεθεί μια ασφάλεια, έτσι ώστε αν προκύψει

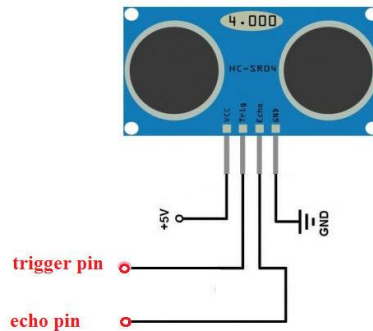
κάποιο βραχυκύκλωμα να προστατευθεί, τόσο το κύκλωμα, όσο και ο κατασκευαστής σε περίπτωση που έχει άμεση επαφή με αυτό. Στην συνέχεια το ρεύμα κατευθύνεται σε δύο μετατροπείς συνεχούς τάσης από 12V σε 5V. Ο ένας είναι ο LM2596s ο οποίος αντέχει ρεύμα έως 3A και τροφοδοτεί, μέσω micro USB, αποκλειστικά το Raspberry Pi. Το Raspberry Pi είναι μια συσκευή η οποία απαιτεί περίπου 2A ρεύματος, ενώ λειτουργεί και με λιγότερα με χαμηλότερες επιδόσεις. Θέλοντας να αξιοποιήσουμε στο έπακρο τις δυνατότητες που προσφέρει η συγκεκριμένη πλακέτα, επιλέξαμε τον συγκεκριμένο μετατροπέα τάσης.

Ο δεύτερος μετατροπέας είναι ένας USB αντάπτορας αυτοκινήτου ο οποίος αντέχει μέχρι 1A και τροφοδοτεί με 5V μια προτυποποιημένη πλακέτα (breadboard). Στο συγκεκριμένο breadboard είναι συνδεδεμένα όλα τα υπόλοιπα υλικά που χρειάζονται τροφοδοσία τάσης 5V. Τα υλικά αυτά είναι οι αισθητήρες υπέρυθρης ακτινοβολίας, οι αισθητήρες υπέρηχων, οι σερβοκινητήρες, η οθόνη υγρών κρυστάλλων και κουμπιά-μπουτόν. Επειδή όλα αυτά τα υλικά είναι χαμηλής κατανάλωσης ρεύματος δεν υπήρχε ανάγκη για εύρεση συγκεκριμένου μετατροπέα. Έτσι χρησιμοποιήθηκε ο αντάπτορας που αναφέρθηκε παραπάνω και ήταν άμεσα διαθέσιμος στο εργαστήριο. Τα δύο Arduino του συστήματος είναι συνδεδεμένα, μέσω USB, με το Raspberry Pi. Αυτός ο τρόπος σύνδεσης επιτρέπει, τόσο την ανταλλαγή πληροφοριών μεταξύ των Arduino και του Raspberry Pi, όσο και την τροφοδοσία των πρώτων από το δεύτερο. Τέλος η γέφυρα H συνδέθηκε με την μπαταρία 6V 3FM4.5 με σκοπό την καθοδήγηση αυτής της τάσης στους ηλεκτροκινητήρες του οχήματος, ενώ τις εντολές χειρισμού τις δέχεται από τους ακροδέκτες που είναι συνδεδεμένοι στο Arduino. Η λειτουργία της γέφυρας H αναλύεται σε επόμενη ενότητα.

3.1.2 Αισθητήρες υπέρηχων και υπέρυθρης ακτινοβολίας

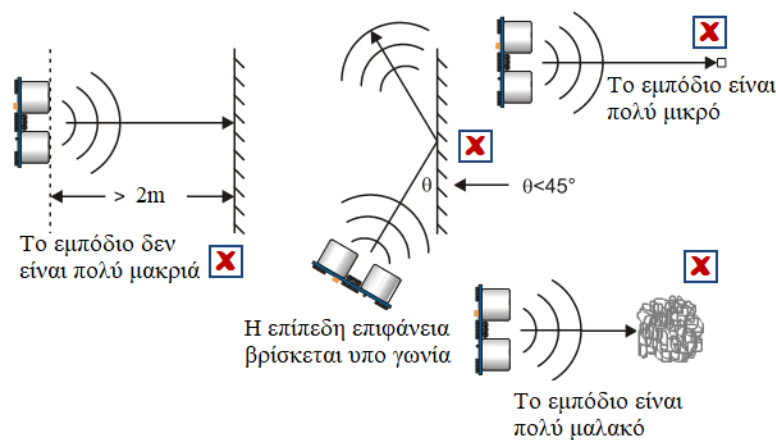
Οι αισθητήρες υπέρηχων χρησιμοποιήθηκαν για τον εντοπισμό εμποδίων στον χώρο. Χρησιμοποιήθηκαν δύο μοντέλα τα HC-SR04 και οι US-020 τα οποία είναι παρόμοιων χαρακτηριστικών. Συγκεκριμένα η υλοποίηση ξεκίνησε με τα US-020, όμως λόγω ανάγκης προσθήκης περισσότερων αισθητήρων και παράλληλα έλλειψης αποθεμάτων στο εργαστήριο, επιλέχθηκαν τα αισθητήρια HC-SR04. Και τα δύο μοντέλα έχουν τον ίδιο τρόπο λειτουργίας και την ίδια διάταξη ακροδεκτών (**Εικόνα 12**). Η διάταξη αυτή αποτελείται από τέσσερις ακροδέκτες. Από αριστερά προς τα δεξιά έχουμε τον ακροδέκτη τροφοδοσίας 5V, trigger, echo, και γείωσης. Για την χρήση του στέλνεται ένα σήμα ενεργοποίησης στον ακροδέκτη trigger. Στην συνέχεια γίνεται η αποστολή υπέρηχου από τον αισθητήρα ο οποίος περιμένει κάποιο μικρό χρονικό διάστημα μέχρι να επιστρέψει ο ανακλασμένος υπέρηχος. Αμέσως μετά μέσω

του ακροδέκτη echo, ο αισθητήρας επιστρέφει έναν αριθμό σε μορφή PWM. Ο αριθμός αυτός αντιστοιχεί στον χρόνο που έκανε ο υπέρηχος να επιστρέψει, ενώ αν δεν επέστρεψε μέσα στο χρονικό διάστημα που αναφέραμε προηγουμένως, τότε η τιμή αυτή είναι μηδέν. Τέλος με βάση τον χρόνο αυτόν και την ταχύτητα διάδοσης του υπέρηχου υπολογίζεται η απόσταση του εμποδίου [22] [25].



Εικόνα 12: Ακροδέκτες αισθητήρα υπέρηχων

Συνολικά τοποθετήθηκαν πέντε αισθητήρες με την εξής διάταξη: δυο στο μπροστινό μέρος του οχήματος, δύο στο πίσω και ένα πάνω στην κάμερα. Ειδικά ο τελευταίος τοποθετήθηκε στην συγκεκριμένη θέση με σκοπό την εκμετάλλευση του μηχανισμού περιστροφής της κάμερας, για έλεγχο εμποδίων στις δύο μεριές εκατέρωθεν του οχήματος. Οι συγκεκριμένοι αισθητήρες είναι ικανοί να εντοπίσουν κάποιο εμπόδιο από απόσταση 2cm μέχρι 2m, εφόσον τηρούνται οι περιορισμοί που αναφέρονται στην **Εικόνα 13**.



Εικόνα 13: Περιορισμοί εντοπισμού εμποδίου

Παράλληλα λόγω της θέσης που έχουν τοποθετηθεί οι αισθητήρες αλλά και λόγω του αριθμού τους, ο εντοπισμός εμποδίου περιορίζεται με επιπλέον κανόνες. Ο εντοπισμός υφίσταται μόνο στο μπροστινό και πίσω μέρος του οχήματος για αποφυγή σύγκρουσης. Παρόλα αυτά δεν είναι εφικτός αν το εμπόδιο βρίσκεται στις γωνίες του οχήματος, ή στα ενδιάμεσα σημεία μεταξύ των αισθητήρων. Αναλυτική περιγραφή των θέσεων των εμποδίων που δεν μπορεί να εντοπίσει η παρούσα υλοποίηση δίνεται στην **Εικόνα 14**.



Εικόνα 14: Περιπτώσεις μη ανίχνευσης εμποδίων

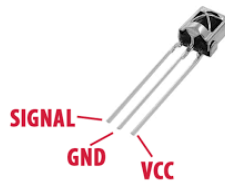
Οι αισθητήρες υπέρυθρης ακτινοβολίας (αισθητήρες IR, infrared) χρησιμοποιούνται για την λήψη σήματος από τον πομπό του χρήστη με σκοπό τη δυνατότητα συνοδείας του από το όχημα. Χρησιμοποιήθηκαν τέσσερις τέτοιοι αισθητήρες τοποθετημένοι σε παράταξη τόξου στο μπροστινό μέρος του οχήματος (**Εικόνα 15**) προσφέροντας με αυτόν τον τρόπο μεγάλη γωνία εμβέλειας.



Εικόνα 15: Τοπολογία Αισθητήρων IR

Η λειτουργία τους είναι απλή μιας και αποτελούνται από τρεις ακροδέκτες (Εικόνα 16). Από αριστερά προς τα δεξιά έχουμε τον ακροδέκτη σήματος, γείωσης, και τροφοδοσίας (5V). Μόλις ανιχνεύσει παλμό στην συγκεκριμένη ακτινοβολία, μεταδίδει τον παλμό αυτό από τον ακροδέκτη σήματος, μετατρέποντας με αυτόν τον τρόπο έναν οπτικό σήμα σε ηλεκτρικό.

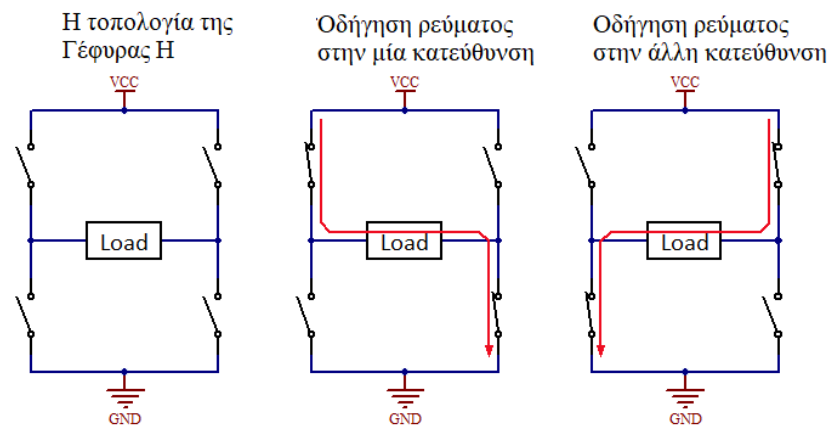
Η χρήση PWM από τους αισθητήρες απόστασης μας οδήγησε στην χρήση του Arduino αντί για το Raspberry Pi, μιας και το δεύτερο δεν έχει αυτή την δυνατότητα (με εξαίρεση ένας μόνο ακροδέκτης του, που προορίζεται για σήμα ήχου) [36]. Για λόγους παραλληλίας, αλλά και λόγο καλύτερης υποστήριξης των αισθητήρων IR συγκριτικά με το Raspberry Pi, χρησιμοποιήθηκε το δεύτερο Arduino του συστήματος.



Εικόνα 16: Διάταξη ακροδεκτών αισθητήρα IR

3.1.3 Διπλή Γέφυρα H L298N

Η γέφυρα H όπως έχει αναφερθεί και στο προηγούμενο κεφάλαιο είναι ένα κύκλωμα που επιτρέπει, τον έλεγχο ενός συνεχούς ρεύματος (DC) ηλεκτροκινητήρα, και την εναλλαγή μεταξύ αριστερόστροφης και δεξιόστροφης περιστροφής (Εικόνα 17) [37].



Εικόνα 17: Λειτουργία της γέφυρας H

Ένα τέτοιο κύκλωμα ήταν εφικτό να κατασκευαστεί στο εργαστήριο ηλεκτρονικής ή στο εργαστήριο ψηφιακών συστημάτων & αρχιτεκτονικής υπολογιστών του Πανεπιστημίου μας. Παρ' όλα αυτά βέλτιστη λύση αποτέλεσε η αγορά έτοιμου κυκλώματος, καθώς το κόστος ήταν χαμηλότερο και έγινε εξοικονόμηση χρόνου που αξιοποιήθηκε σε λύση άλλων προβλημάτων. Μάλιστα το κύκλωμα που αγοράστηκε είναι διπλής γέφυρας επιτρέποντας έτσι τον χειρισμό δύο DC κινητήρων από το ίδιο κύκλωμα. Επιπλέον επιτρέπει τον χειρισμό ενός βηματικού κινητήρα (stepper motor) αντί για δύο DC. Στην παρούσα υλοποίηση γίνεται ο χειρισμός δύο ίδιων DC κινητήρων με Arduino μέσω γέφυρας H. Το Arduino είναι υπεύθυνο για την αποστολή του κατάλληλου σήματος, ενώ η γέφυρα H το ενισχύει χρησιμοποιώντας στην μπαταρία 6V που αναφέραμε σε προηγούμενη ενότητα.

Συγκεκριμένα οι ακροδέκτες του Arduino που είναι υπεύθυνοι για τον χειρισμό των κινητήρων είναι οι 2,3 και 4,7 οι οποίοι είναι συνδεδεμένοι στους ακροδέκτες της γέφυρας in1,in2 και in3,in4 αντίστοιχα. Το πρώτο ζευγάρι, ακροδέκτες 2,3 είναι υπεύθυνο για το χειρισμό του κινητήρα κίνησης, ενώ το δεύτερο ζευγάρι 4,7 χειρίζεται τον κινητήρα περιστροφής του τιμονιού. Ο τρόπος λειτουργίας περιγράφεται στον παρακάτω πίνακα (Πίνακας 3).

Τιμές Ακροδεκτών 2,3	Συμπεριφορά	Τιμές Ακροδεκτών 4,7	Συμπεριφορά
0 0	Δεν υπάρχει κίνηση	0 0	Δεν υπάρχει περιστροφή
0 1	Κίνηση προς τα πίσω	0 1	Περιστροφή στα δεξιά

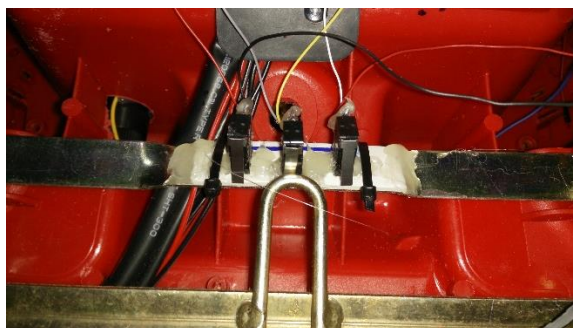
1 0	Κίνηση προς τα εμπρός	1 0	Περιστροφή στα αριστερά
1 1	Δεν υπάρχει κίνηση	1 1	Δεν υπάρχει περιστροφή

Πίνακας 3: Τιμές ακροδεκτών για τον χειρισμό των DC κινητήρων

Επιπλέον το Arduino επειδή έχει άμεση επικοινωνία με τους αισθητήρες απόστασης, υλοποιεί υποσύστημα αποτροπής σύγκρουσης, οπότε ήταν αναγκαίος ο χειρισμός των κινητήρων αυτών από το Arduino και όχι από το Raspberry Pi. Αναλυτικότερα η λειτουργία της αποτροπής σύγκρουσης περιγράφεται στο επόμενο κεφάλαιο.

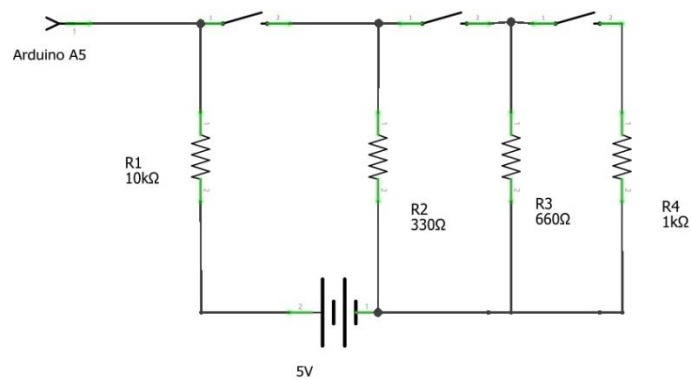
3.1.4 Σερβοκινητήρες και κύκλωμα θέσης τιμονιού

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν δύο σερβοκινητήρες SG90 για την περιστροφή της κάμερας (πάνω/κάτω, δεξιά/αριστερά). Οι σερβοκινητήρες αυτοί είναι ικανοί για περιστροφή μεταξύ 0°-180°, ενώ ο χειρισμός τους γίνεται με PWM. Λόγω της χρήσης PWM οι σερβοκινητήρες είναι συνδεδεμένοι με Arduino και όχι με το Raspberry Pi. Ιδανικό θα ήταν να γινόταν και ο χειρισμός του τιμονιού του οχήματος με σερβοκινητήρα, όμως δεν υπήρχε η κατασκευαστική δυνατότητα αντικατάστασης του παρεχόμενου DC κινητήρα, με έναν σερβοκινητήρα. Αντί αυτού κατασκευάστηκε ένα σύστημα, το οποίο είναι σε θέση να εντοπίζει την τρέχουσα θέση του τιμονιού (**Εικόνα 18**). Ένα τέτοιο σύστημα θεωρήθηκε απαραίτητο καθώς παρέχει σημαντική πληροφορία για την υλοποίηση ορθότερων και πολυπλοκότερων αλγορίθμων.



Εικόνα 18: Σύστημα θέσης τιμονιού

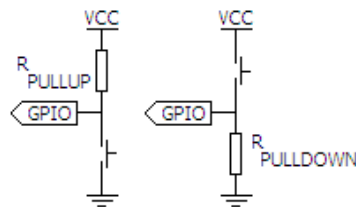
Η κατασκευή ουσιαστικά αποτελείται από τρία κουμπιά-μπουτόν τοποθετημένα με συγκεκριμένο τρόπο, έτσι ώστε σε κάθε θέση του τιμονιού να είναι κάποιο από αυτά πατημένο. Κάθε κουμπί αντιστοιχεί σε συγκεκριμένη αντίσταση στο κύκλωμα (**Εικόνα 19**). Έτσι ανάλογα ποιο κουμπί είναι πατημένο, βραχυκυκλώνεται η ανάλογη αντίσταση αλλάζοντας με αυτόν τον τρόπο την τάση εισόδου που δέχεται ο αναλογικός ακροδέκτης του Arduino. Ανάλογα με την τιμή εισόδου (0-1023, 0V-5V) γνωρίζει το σύστημα σε ποια θέση βρίσκεται το τιμόνι [38]. Επιπλέον η όλη κατασκευή έχει υλοποιηθεί με τρόπο έτσι ώστε πάντα κάποιο κουμπί να είναι πατημένο. Για λόγους ασφαλείας έχει τοποθετηθεί στην αρχή του κυκλώματος μία αντίσταση 10kΩ έτσι ώστε να προστατευτεί το κύκλωμα σε περίπτωση που κανένα κουμπί δεν έχει πατηθεί.



Εικόνα 19: Κύκλωμα συστήματος θέσης τιμονιού

3.1.5 Συνολικό σύστημα του οχήματος

Εφόσον είδαμε όλα τα απαραίτητα επιμέρους συστήματα και υλικά του οχήματος, ήρθε η στιγμή να μελετήσουμε την ολική κατασκευή και υλοποίηση. Στο Raspberry Pi 3 έχουμε συνδέσει με USB μια κάμερα και δύο Arduino Uno. Επιπλέον στο Raspberry Pi 3 έχουμε συνδέσει δύο κουμπιά-μπουτόν, τρία LEDs (πράσινο, κόκκινο, μπλε), την οθόνη υγρών κρυστάλλων (LCD), και τα φώτα του οχήματος. Τα κουμπιά είναι συνδεδεμένα σε διάταξη PULLDOWN (**Εικόνα 20**) ενώ τροφοδοτούνται με τάση 3.3V απευθείας από το Raspberry Pi.



Εικόνα 20: Pullup και Pulldown διάταξη

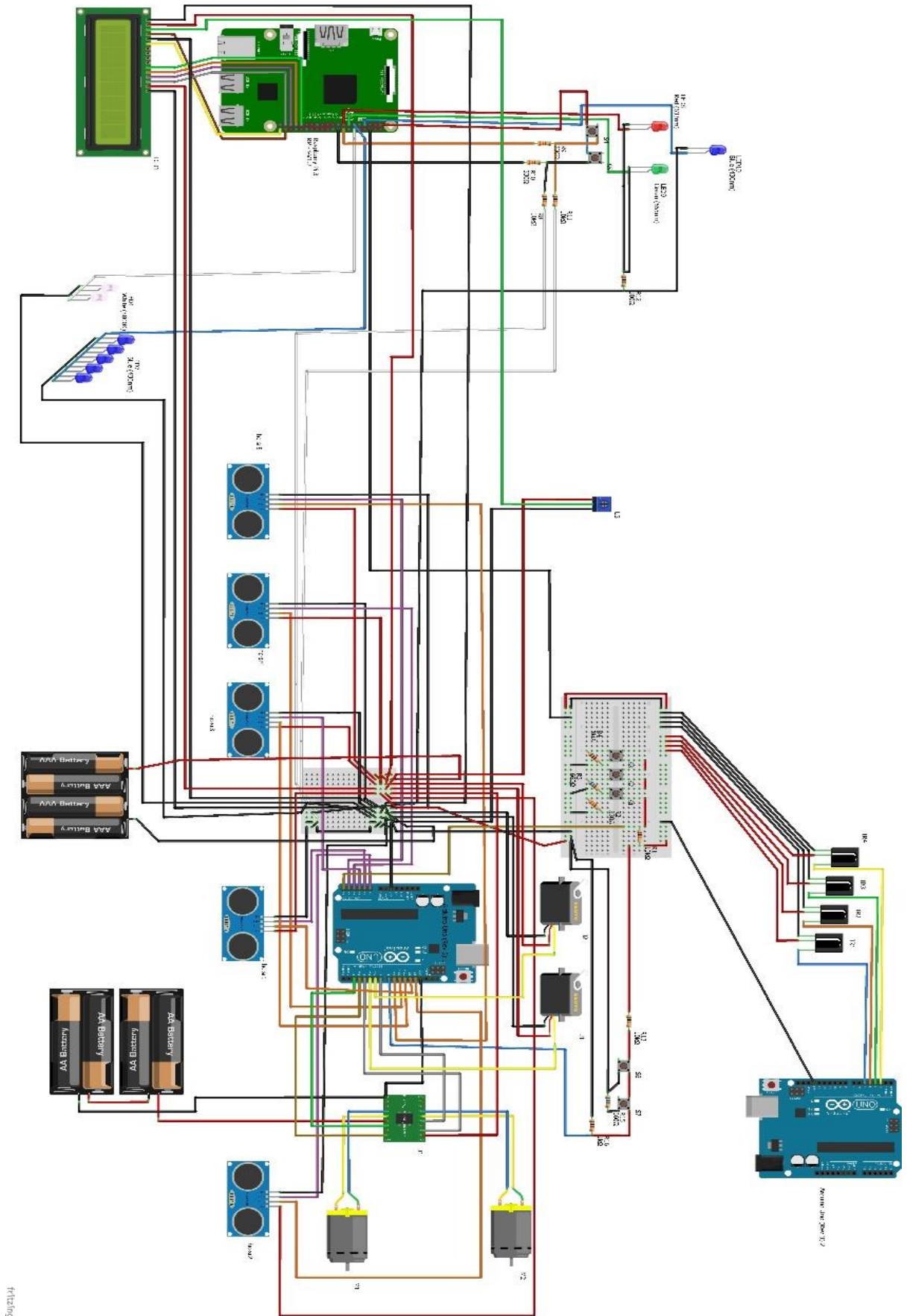
Επιπρόσθετα έχουν τοποθετηθεί αντιστάσεις των 330Ω πριν τους ακροδέκτες του Raspberry Pi 3 με σκοπό να τους προστατέψουμε μιας και $3.3V$ είναι η μέγιστη τάση που μπορούν να δεχτούν [36]. Ταυτόχρονα έχουν τοποθετηθεί και αντιστάσεις των $10k\Omega$ προς την γείωση με σκοπό την αποφυγή βραχυκυκλώματος. Τα φώτα του οχήματος είναι LEDs χωρισμένα σε δύο χρωματικές ομάδες (άσπρα, μπλε). Τα LEDs της κάθε ομάδας είναι συνδεδεμένα μεταξύ τους παράλληλα.

Για την ρύθμιση της αντίθεσης της οθόνης έχει τοποθετηθεί ένα ποτενσιόμετρο που τροφοδοτεί τον αντίστοιχο ακροδέκτη της οθόνης. Το όλο κύκλωμα το οχήματος φαίνεται στο σχηματικό που ακολουθεί (**Εικόνα 21**). Από τα δύο Arduino του κυκλώματος παρατηρούμε ότι στο ένα έχουν τοποθετηθεί μόνο οι αισθητήρες IR, ενώ στο δεύτερο όλα τα υπόλοιπα εξαρτήματα. Αυτό έγινε με σκοπό την επίτευξη παραλληλοποίησης του συστήματος, μιας και ο μικροελεγκτής του Arduino λειτουργεί σειριακά. Έτσι έχουμε την λειτουργία λήψης, ελέγχου, και αποκωδικοποίησης του σήματος IR να εκτελείται παράλληλα και ανεξάρτητα από το υπόλοιπο σύστημα, προσφέροντας άμεση απόκριση σε πραγματικό χρόνο. Επιπρόσθετα όπως αναφέρθηκε και σε προηγούμενη ενότητα, οι αισθητήρες απόστασης και οι κινητήρες κίνησης πρέπει να χειρίζονται από το ίδιο Arduino έτσι ώστε να μπορεί να υλοποιηθεί το υποσύστημα αποφυγής εμποδίων.

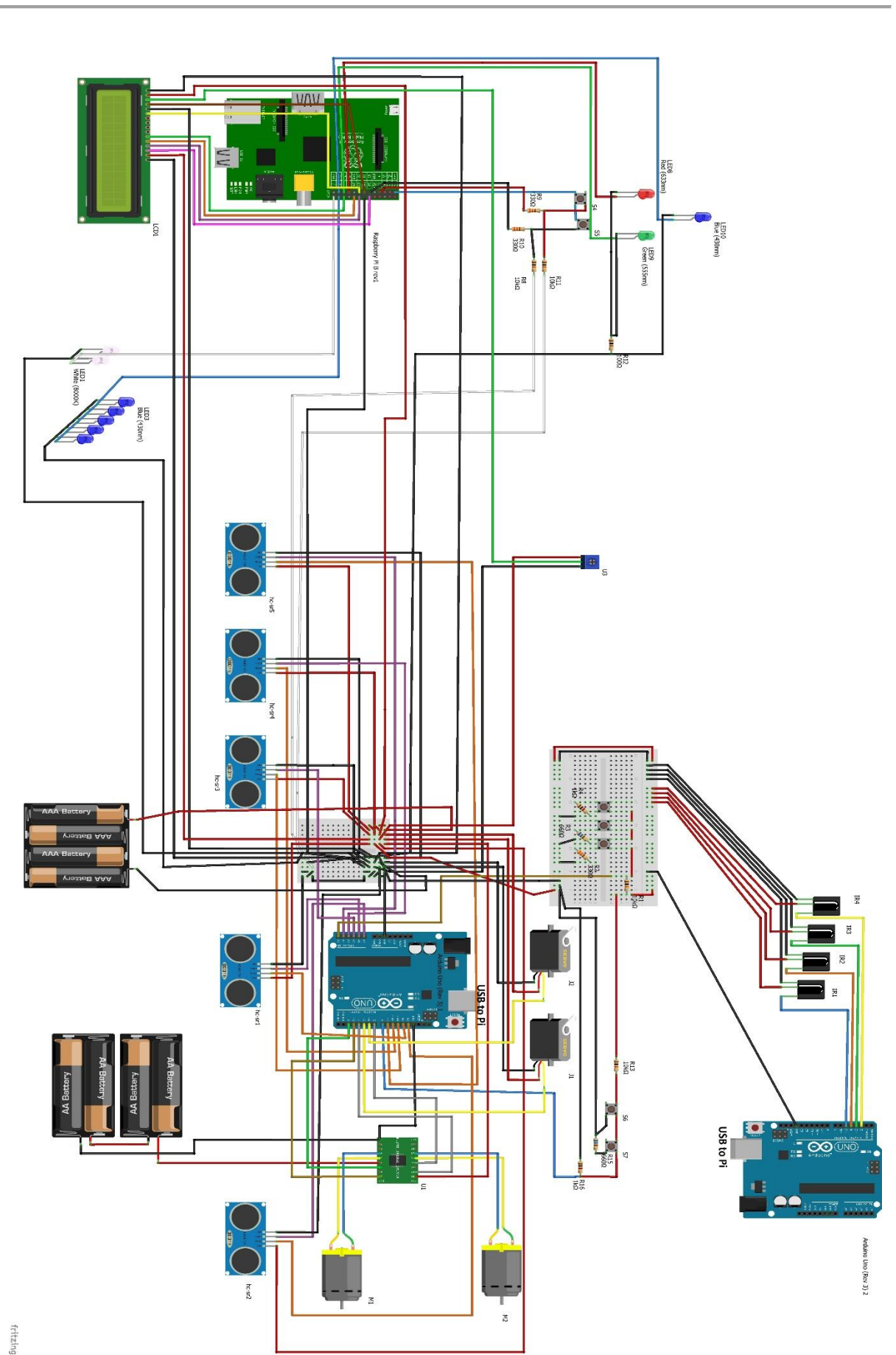
Τα κουμπιά S1,S2,S3 αντιστοιχούν στο σύστημα αναγνώρισης της θέσης του τιμονιού που αναλύσαμε στην προηγούμενη ενότητα. Επιπλέον τα κουμπιά S6 και S7 είναι συνδεδεμένα με διάταξη PULLUP (**Εικόνα 20**) και αποτελούν μέρος του «έξυπνου» προφυλακτήρα του οχήματος. Με το πάτημα του ενός ή και των δυο κουμπιών γειώνεται το κύκλωμα με αποτέλεσμα η είσοδος στο Arduino να είναι λογικό μηδέν, δηλαδή έλλειψη τάσης. Μόλις συμβεί κάτι τέτοιο το Arduino ακινητοποιεί το όχημα και δεν επιτρέπει την προς τα εμπρός κίνησή του. Η υλοποίηση ενός τέτοιου συστήματος αποτελεί μια δικλίδα ασφαλείας έτσι ώστε

στην περίπτωση σύγκρουσης να μην συνεχίσει το όχημα να κινείται και προκαλέσει επιπλέον ζημιά.

Παράλληλα υλοποιήθηκε το ίδιο σύστημα και με Raspberry Pi Model B, το οποίο είναι οικονομικότερο, για λόγους έρευνας της συμβατότητας του συστήματος. Το αποτέλεσμα είναι το σύστημα να είναι πλήρως λειτουργικό χωρίς να χρειαστεί καμία διαφορά στο λογισμικό. Στο υλικό εκτός από την αντικατάσταση του Raspberry Pi, προστέθηκαν ένας δέκτης Wi-Fi (Edimax EW-7811Un), ένας δέκτης Bluetooth (Canyon CNR-BTU3), και ένας USB διανομέας, λόγω έλλειψης αυτών των στοιχείων από την πλακέτα. Εκτός από την έλλειψη ενσωματωμένου Bluetooth και κεραίας Wi-Fi, το Raspberry Pi Model B παρέχει και μικρότερο αριθμό ακροδεκτών GPIO συγκριτικά με το Raspberry Pi 3 Model B. Λόγω αυτού του περιορισμού χρησιμοποιήθηκαν άλλες συνδέσεις οι οποίες παρουσιάζονται στην **Εικόνα 22**.

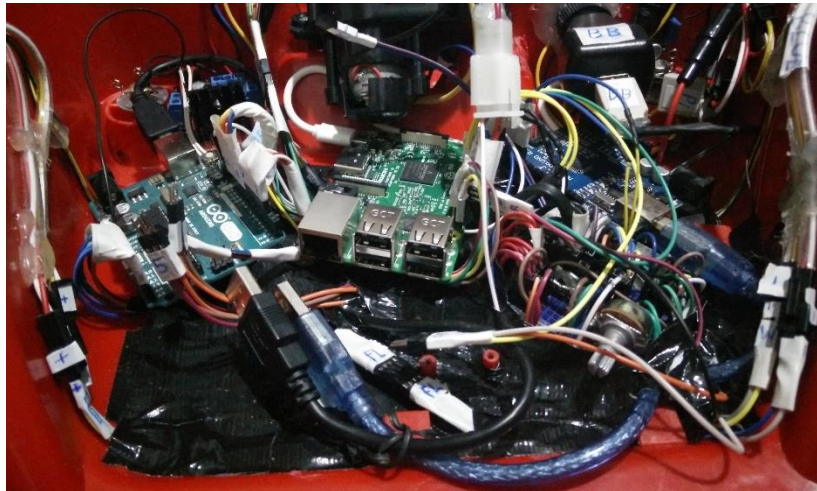


Εικόνα 21: Κύκλωμα του οχήματος με χρήση Raspberry Pi 3



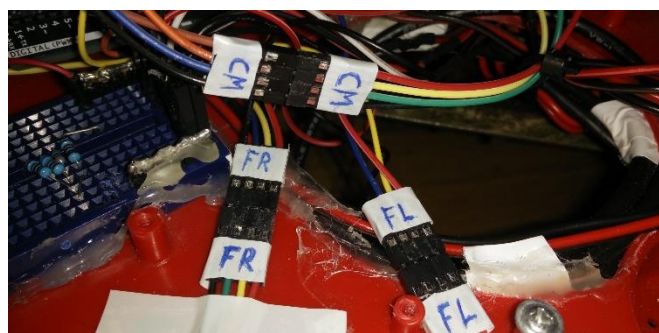
Εικόνα 22: Κύκλωμα του οχήματος με χρήση Raspberry Pi

Κατασκευαστικά έπρεπε να εκμεταλλευτεί στο έπακρο ο χώρος στο καπό του οχήματος για την εγκατάσταση όλου του εξοπλισμού (**Εικόνα 23**). Αναγκαία ήταν όχι μόνο η τοποθέτηση στην κατάλληλη θέση του κάθε υλικού αλλά και ο σωστός προσανατολισμός του, που πολλές στιγμές αποτέλεσε πρόκληση.



Εικόνα 23: Ολοκληρωμένη εγκατάσταση συστήματος

Παράλληλα αναπτύχθηκαν πρότυπα σύνδεσης μεταξύ των εξαρτημάτων με ύπαρξη κατάλληλων ετικετών (**Εικόνα 24**). Έτσι η διαχείριση των υλικών γίνεται με μεγαλύτερη ευκολία, ενώ μειώνεται σημαντικά η πιθανότητα λάθους κατά την σύνδεση τους.



Εικόνα 24: Αισθητήρες απόστασης μπροστινού μέρους και κάμερας

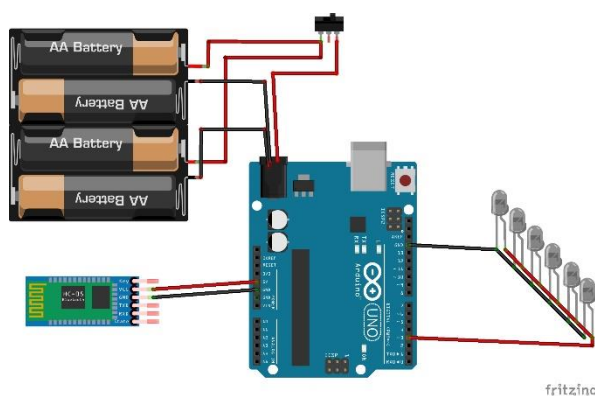
Το τελικό αποτέλεσμα είναι ένα όχημα το οποίο εκθέτει σε μικρό βαθμό το ηλεκτρονικό του εξοπλισμό προφυλάγοντάς το από εξωτερικούς κινδύνους (**Εικόνα 25**). Παράλληλα αποτελεί σταθερή κατασκευή προσφέροντας αξιοπιστία κατά την χρήση του.



Εικόνα 25: Εξωτερική εμφάνιση κατασκευής οχήματος

3.2 Υλικό μέρος πομπού

Το υλικό μέρος απ' το οποίο αποτελείται ο πομπός είναι απλούστερο συγκριτικά με το προηγούμενο. Η όλη κατασκευή αποτελείται από έναν κλώνο Arduino Uno στον οποίο είναι συνδεδεμένα έξι IR LEDs και ένας πομπός Bluetooth BC417 (**Εικόνα 26**). Η τροφοδοσία του συστήματος γίνεται με χρήση τεσσάρων μπαταριών τύπου AA των 1.5V για λόγους φορητότητας. Επιπλέον το Bluetooth είναι συνδεδεμένο μόνο για παροχή ρεύματος στο Arduino καθώς μας ενδιαφέρει μόνο η εκπομπή σήματος και όχι η ανταλλαγή μηνυμάτων.

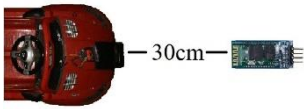
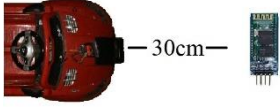
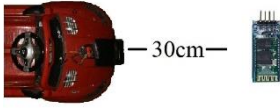


Εικόνα 26: Σχηματικό κυκλώματος πομπού

Άξιο αναφοράς είναι το γεγονός ότι δεν χρησιμοποιήθηκε καμία αντίσταση στην σύνδεση των LEDs στο Arduino. Εσκεμμένα δεν τοποθετήθηκε καμία αντίσταση έτσι ώστε να επιτευχθεί η μέγιστη δυνατή φωτεινότητα μιας και αποτελεί σημαντικό παράγοντα στην εμβέλεια του συστήματος κατά την λειτουργία συνοδείας. Το κάθε LED λειτουργεί με μέγιστη τάση 1.4V και μέγιστο ρεύμα 0.1A, δηλαδή με μέγιστη ισχύ 0.14W [39]. Το Arduino στην κάθε έξοδό του βγάζει 5V και 0.02A [40]. Επομένως σε κάθε LED παρέχεται τάση 5V και ρεύμα 0.0033A, που αντιστοιχεί σε ισχύ 0.016W. Από τα παραπάνω συμπεραίνουμε ότι με αυτή τη διάταξη δεν υπάρχει κίνδυνος καταστροφής του κυκλώματος και των υλικών του.

Όσο αφορά το Bluetooth όπως αναφέρθηκε και προηγουμένως τροφοδοτείται από το Arduino με αποτέλεσμα μόνο να εκπέμπει το όνομα της συσκευής και την MAC διεύθυνση της. Ο σκοπός του είναι η προσέγγιση της απόστασης μεταξύ οχήματος και χρήστη, έτσι ώστε όταν ελαττωθεί αρκετά να σταματήσει το όχημα. Η λειτουργία αυτή υλοποιήθηκε μετρώντας

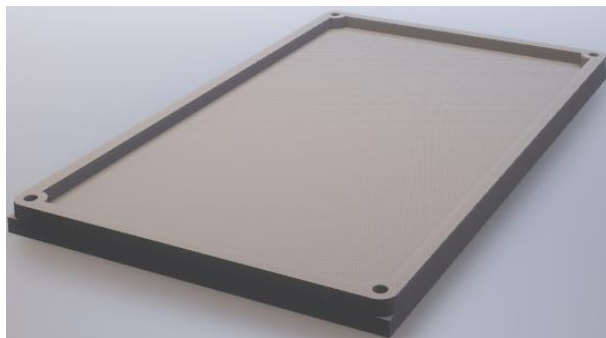
από το Raspberry Pi 3 την ένταση του σήματος της συγκεκριμένης MAC διεύθυνσης του πομπού. Με αυτόν τον τρόπο εξασφαλίζεται η ασφάλεια, (με την έννοια του αγγλικού όρου security) στον υπολογισμό της απόστασης που αναφέρθηκε. Παρόλα αυτά η μέτρηση της έντασης του σήματος έχει μεγάλη απόκλιση μέχρι και 13 dB, με τον συγκεκριμένο εξοπλισμό, εξαιτίας του διαφορετικού προσανατολισμού του πομπού. Στον **Πίνακα 4** παρουσιάζονται τα αποτελέσματα από 600 μετρήσεις της έντασης του σήματος Bluetooth από το όχημα. Τόσο το όχημα όσο και ο πομπός βρίσκονται σε πλήρη ακινησία, ενώ η μεταξύ τους απόσταση είναι 30cm.

	Ελάχιστη ένταση (dB)	Μέγιστη ένταση (dB)	Μέσος όρος έντασης (dB)
 Εικόνα 27: προσανατολισμός 0°	-11	-1	-3
 Εικόνα 28: προσανατολισμός 90°	Δεν συνδέθηκε	Δεν συνδέθηκε	Δεν συνδέθηκε
 Εικόνα 29: προσανατολισμός 180°	-14	-1	-6
 Εικόνα 30: προσανατολισμός 270°	-6	-2	-4

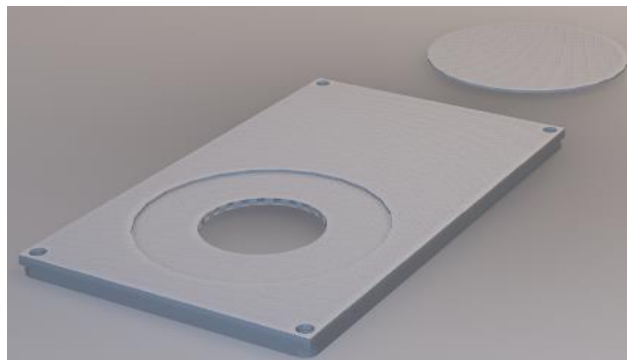
Πίνακας 4: Μέτρηση έντασης με διαφορετικούς προσανατολισμούς στον πομπό

Σχετικά με τα LEDs έχουν τοποθετηθεί έξι, ενώ έχουν χωριστεί σε τρία ζευγάρια. Κάθε ζευγάρι έχει παραταχθεί με διαφορετική κλίση έτσι ώστε να λειτουργεί σωστά ο πομπός ανεξάρτητα από το ύψος του χρήστη. Παράλληλα σχεδιάστηκε, με την χρήση της δοκιμαστικής

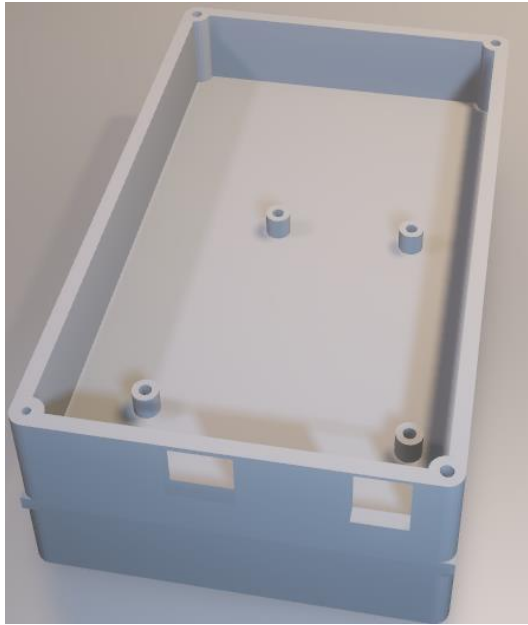
έκδοσης του εργαλείου SketchUp 2016, ένα τρισδιάστατο μοντέλο ως θήκη της συγκεκριμένης κατασκευής. Το σχέδιο αυτό μπορεί να τυπωθεί σε έναν 3D εκτυπωτή και να χρησιμοποιηθεί για να φιλοξενήσει το υλικό του πομπού. Η σχεδιασμένη θήκη αποτελείται από τρία μέρη. Το κάτω μέρος χρειάζεται για την απομόνωση των μπαταριών από το εξωτερικό περιβάλλον (**Εικόνα 31**). Το μεσαίο μέρος από την μία μεριά φιλοξενεί τις μπαταρίες (**Εικόνα 34**), ενώ από την άλλη το Arduino Uno R3 (**Εικόνα 33**). Τέλος στο πάνω μέρος μπορεί να τοποθετηθεί το Bluetooth, ενώ στον κυκλικό τομέα του εγκαθίστανται τα LEDs (**Εικόνα 32**). Με το τελευταίο μέρος κλείνει και από την πάνω μεριά η θήκη.



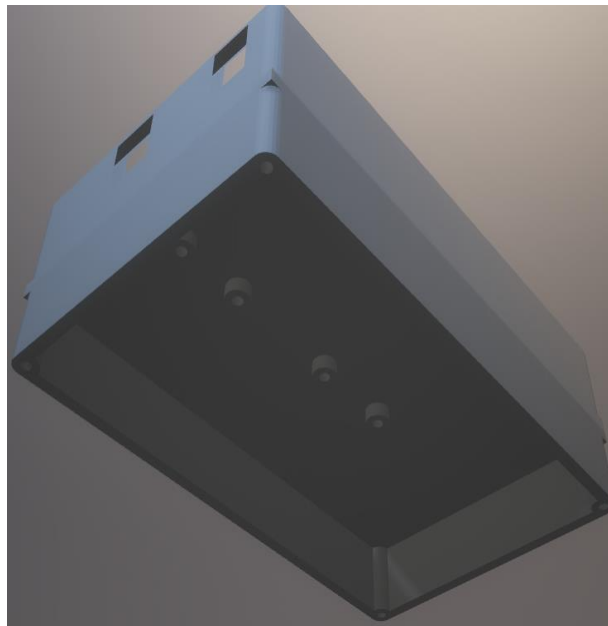
Εικόνα 31: Κάτω μέρος Θήκης



Εικόνα 32: Πάνω μέρος Θήκης



Εικόνα 33: Πάνω μεριά μεσαίου μέρους θήκης



Εικόνα 34: Κάτω μεριά μεσαίου μέρους θήκης

3.3 Κόστος Υλικών Κατασκευής

Ο προσανατολισμός της υλοποίησης του συστήματος ήταν η εκμετάλλευση των πόρων που ήδη υπήρχαν, στοχεύοντας έτσι τόσο στο χαμηλό κόστος κατασκευής, όσο και στην αξιοπιστία του προϊόντος. Το αναλυτικό κόστος κατασκευής περιγράφεται στον παρακάτω πίνακα (Πίνακας 5).

Υλικό	Τιμή Μονάδας (€)	Ποσό (€)
1x όχημα για παιδιά	100	100
1x Μπαταρία MW MWS 7.2-12	15.8	15.8
1x Raspberry Pi 3 Model B	35	35
3x κλώνοι Arduino Uno R3	1.80	5.4
1x Bluetooth BC417	5.16	5.16
1x Γέφυρα H	1.30	1.30
1x Web Camera	15.26	15.26
1x Μετατροπέας LM2596s	1.02	1.02
1x LCD 16x2 HD44780	0.88	0.88
2x Σερβοκινητήρες SG90	0.87	1.74
2x Αισθητήρες US-020	1.2	2.4
3x Αισθητήρες HC-SR04	0.85	1.7
4x Αισθητήρες IR VS1838b	0.05	0.2
9x IR LEDs	0.01	0.09
7x Διακόπτης-μπουτόν	0.01	0.07
11x Αντιστάσεις	0.01	0.11
1x Ποτενσιόμετρο	0.01	0.01
1x Breadboard mini	0.4	0.4
1x Μετατροπέας USB αυτοκινήτου	0.7	0.7
ΣΥΝΟΛΟ		187,24 €

Πίνακας 5: Κόστος υλοποίησης

3.4 Σύνοψη Κεφαλαίου

Στο παρόν κεφάλαιο παρουσιάστηκε το υλικό μέρος του οχήματος και του πομπού. Αναλύθηκε ο τρόπος λειτουργίας και ο σκοπός του κάθε υλικού. Παράλληλα αναφέρθηκε η όλη συνδεσμολογία του συστήματος καθώς και του εναλλακτικού συστήματος που υλοποιήθηκε. Τέλος παραδόθηκε μία λίστα με όλα τα υλικά που χρησιμοποιήθηκαν και το κόστος τους. Ακολουθεί το τέταρτο κεφάλαιο στο οποίο παρουσιάζεται το λογισμικό που υλοποιήθηκε, ποιες λειτουργίες περιλαμβάνει και τι απαιτήσεις καλύπτει. Παράλληλα περιγράφεται η λειτουργία του και ο τρόπος επικοινωνίας μεταξύ των συσκευών. Τέλος αναφέρονται οι τρόποι χρήσης του συστήματος, οι περιορισμοί του και οι συμβάσεις που έγιναν.

Κεφάλαιο 4 – Ανάλυση λογισμικού μέρους

Αφού καλύφθηκαν οι απαραίτητες γνώσεις για το υλικό μέρος του συστήματος, προχωράμε στην ανάλυση του λογισμικού. Σε αυτό το κεφάλαιο θα παρουσιαστούν τα λογισμικά που υλοποιήθηκαν σε όλες τις επιμέρους συσκευές, και η μεταξύ τους επικοινωνία. Η ανάλυση ξεκινάει από τα μικρότερα και απλούστερα υποσυστήματα, καταλήγοντας στα μεγαλύτερα και πολυπλοκότερα. Παράλληλα αναφέρονται τρόποι λειτουργίας, θέματα ασφάλειας, ποιοτικά χαρακτηριστικά του συστήματος καθώς και συμβάσεις που έγιναν κατά την υλοποίηση της παρούσας διπλωματικής εργασίας.

4.1 Λογισμικό στα Arduino

Στο παρόν σύστημα γίνεται η χρήση τριών Arduino Uno για λόγους που αναφέρθηκαν σε προηγούμενα κεφάλαια. Το καθένα από αυτά είναι προγραμματισμένο να εκτελεί συγκεκριμένες λειτουργίες ανάλογα με τον σκοπό που εξυπηρετεί. Σε αυτήν την ενότητα παρουσιάζονται οι λειτουργίες που εκτελεί το κάθε Arduino.

4.1.1 Arduino πομπού

Η πιο απλή από τις τρεις λειτουργίες που θα αναφερθούν για τα Arduino είναι αυτή του πομπού. Σκοπός του πομπού είναι να στέλνει περιοδικά σήμα σε κάποιον ακροδέκτη του, αναβοσβήνοντας έτσι ένα σύμπλεγμα από IR LEDs. Με αυτόν τον τρόπο επιτυγχάνεται η μετάδοση IR σήματος στον χώρο. Για τον σκοπό αυτό χρησιμοποιήθηκε η έτοιμη βιβλιοθήκη IRremote.h, η οποία επιτρέπει την εύκολη διαχείριση και εκπομπή IR σήματος [41]. Η βιβλιοθήκη αυτή χρησιμοποιεί συγκεκριμένους ακροδέκτες του Arduino για την μετάδοση σήματος. Συγκεκριμένα για το Arduino Uno χρησιμοποιεί τους ακροδέκτες 3 και 9, στην παρούσα υλοποίηση επιλέχθηκε ο ακροδέκτης 3. Επιπλέον μέσω της βιβλιοθήκης παρέχεται η δυνατότητα εκπομπής αλφαριθμητικού με την μορφή IR σήματος. Έτσι εξασφαλίζεται η ασφάλεια του συστήματος, καθώς κάποιος κακόβουλος χρήστης θα πρέπει να ξέρει το κωδικοποιημένο αλφαριθμητικό που χρησιμοποιείται. Επιπλέον με τη χρήση του αλφαριθμητικού, το σύστημα παραλείπει ανεπιθύμητα σήματα IR, που μπορεί να υπάρχουν στον χώρο και να οδηγήσουν το όχημα σε διαφορετική κατεύθυνση. Με αυτόν τον τρόπο το

όχημα «υπακούει» μόνο στο σήμα που δέχεται από τον πομπό. Στο συγκεκριμένο σύστημα για αλφαριθμητικό χρησιμοποιήθηκε το όνομα του χρήστη και κατασκευαστή «John», ενώ η αλλαγή του είναι εφικτή αντικαθιστώντας το με το επιθυμητό, στο αρχείο «ir_trans.ino», σειρά 14 και στο αρχείο «ir_rcv.ino», σειρά 15.

4.1.2 Arduino δέκτη IR

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο το όχημα είναι εξοπλισμένο με δύο Arduino Uno, εκ των οποίων το ένα είναι υπεύθυνο για την αποκωδικοποίηση του εισερχόμενου IR σήματος. Στο συγκεκριμένο Arduino είναι συνδεδεμένοι οι τέσσερις IR αισθητήρες με την τοπολογία που αναφέρθηκε στο κεφάλαιο **3.1.5 Συνολικό σύστημα του οχήματος (Εικόνα 21, Εικόνα 22)**. Το Arduino περιοδικά ελέγχει τους αισθητήρες για εισερχόμενο σήμα. Εάν υπάρχει σήμα το αποκωδικοποιεί και το αποτέλεσμα το αποθηκεύει σε έναν πίνακα. Στην συνέχεια αναζητεί στον πίνακα το «σωστό» αλφαριθμητικό που έχει στείλει ο πομπός. Η κάθε γραμμή του πίνακα αντιστοιχεί σε έναν IR αισθητήρα. Με αυτόν τον τρόπο το σύστημα μπορεί να καταλάβει ποιος/ποιοι αισθητήρες δέχτηκαν το σήμα του πομπού. Το αποτέλεσμα αποθηκεύεται στη μεταβλητή 4Bits «command_hist» (ένα Bit για κάθε αισθητήρα, **Πίνακας 6**). Σε αυτό το σημείο πρέπει να αναφερθεί ότι επειδή το Arduino λειτουργεί σειριακά, δεν είναι εφικτός ο ταυτόχρονος έλεγχος όλων των αισθητήρων. Αυτό έχει ως αποτέλεσμα την παράληψη κάποιου χαρακτήρα ή την μετατόπιση ολόκληρου του αλφαριθμητικού κατά μία θέση. Γι' αυτόν τον λόγο χρησιμοποιήθηκε ένας πίνακας εννέα θέσεων, έτσι ώστε να μην χαθεί πολύτιμη πληροφορία.

Αριστερός αισθητήρας	Δεξιός αισθητήρας	Μπροστά-Αριστερός αισθητήρας	Μπροστά-Δεξιός αισθητήρας	Μεταβλητή
1	0	0	0	1000
0	1	0	0	0100
0	0	1	1	0011

Πίνακας 6: Παραδείγματα δομής της IR μεταβλητής

Επιπλέον το Arduino, σε διαφορετική περίοδο από την προαναφερθείσα, τυπώνει στην σειριακή έξοδο (USB) την μεταβλητή των αισθητήρων οι οποίοι δέχτηκαν επιτυχώς το σήμα. Με αυτόν τον τρόπο ενημερώνεται το Raspberry Pi και αξιοποιεί αντιστοίχως την συγκεκριμένη πληροφορία. Για την περιοδική εκτέλεση συναρτήσεων χρησιμοποιήθηκε ξεχωριστή βιβλιοθήκη, η SimpleTimer.h καθώς προσφέρει μεγαλύτερη ακρίβεια χρόνου [42]. Αντίθετα η χρήση της βασικής εντολής «delay» όχι μόνο δεν είναι ακριβής, αλλά επειδή κάνει παύση το ρολόι του μικροελεγκτή προκαλεί δυσλειτουργία στην χρήση των ακροδεκτών, και κατ' επέκταση προβλήματα στην αποκωδικοποίηση του σήματος. Τέλος για τον έλεγχο των IR αισθητήρων χρησιμοποιήθηκε μια παραλλαγή της IRremote.h μιας και η αρχική επιτρέπει τον έλεγχο μόνο σε έναν αισθητήρα και όχι σε πολλαπλούς [43].

4.1.3 Arduino Ελέγχου οχήματος

Το δεύτερο Arduino του οχήματος χρησιμοποιείται για τον έλεγχό του, την υποστήριξη των αισθητήρων απόστασης και για την υλοποίηση συστήματος αποτροπής πρόσκρουσης. Συγκεκριμένα εκτελούνται περιοδικά δύο συναρτήσεις, με διαφορετική περίοδο η κάθε μία. Η πρώτη έχει το όνομα read_sonars και, όπως είναι προφανές, είναι υπεύθυνη για τον έλεγχο των αισθητήρων απόστασης. Με την ολοκλήρωση του ελέγχου καλείται η συνάρτηση allowmoving, η οποία εξετάζει αν τηρούνται οι προϋποθέσεις για ασφαλή κίνηση, ενώ σε αντίθετη περίπτωση εφαρμόζεται ακινητοποίηση του οχήματος. Έπειτα τυπώνεται στην σειριακή έξοδο (USB) η απόσταση του κάθε αισθητήρα για αξιοποίηση της πληροφορίας από το Raspberry Pi. Τέλος γίνεται ο έλεγχος της θέσης του τιμονιού και δίνεται η δυνατότητα χρήσης αυτής της πληροφορίας. Στην παρούσα υλοποίηση δεν ήταν αναγκαία η χρήση αυτής της πληροφορίας, αλλά αποτελεί βάση για μεταγενέστερες υλοποιήσεις. Υπενθυμίζεται ότι ο έλεγχος της θέσης του τιμονιού γίνεται με την μέτρηση της τάσης στον αναλογικό ακροδέκτη εισόδου του Arduino όπως αναφέρεται στο κεφάλαιο **3.1.4 Σερβοκινητήρες και κύκλωμα θέσης τιμονιού**. Οι τιμές που χρησιμοποιούνται είναι: 35-50mV για δεξιά, 65-80mV για κέντρο, και 95-110mV για αριστερά.

Η δεύτερη συνάρτηση, που εκτελείται περιοδικά, έχει το όνομα move_me και αφορά τον έλεγχο του οχήματος. Αρχικά ελέγχει τις εισόδους που αντιστοιχούν στον «έξυπνο» προφυλακτήρα και αν θεωρηθεί αναγκαίο καλεί την allowmoving για ενημέρωση των μέτρων περί κίνησης του οχήματος. Συγκεκριμένα θα κληθεί η allowmoving σε περίπτωση που το

όχημα, ενώ βρισκόταν σε κίνηση, τράκαρε, οπότε θα πρέπει να ακινητοποιηθεί και να μην επιτραπεί άλλο η προς τα εμπρός κίνηση. Η δεύτερη περίπτωση, στην οποία γίνεται κλήση της `allowmoving`, είναι όταν το όχημα μετά από σύγκρουση (ενεργοποίηση του «έξυπνου» προφυλακτήρα) έχει φύγει από το εμπόδιο και δεν πιέζεται ο προφυλακτήρας του. Σε αυτήν την περίπτωση πρέπει να επιτραπεί ξανά η προς τα εμπρός κίνηση. Στην συνέχεια γίνεται μια σειρά ελέγχων της μεταβλητής `stringComplete` η οποία αποτελεί την είσοδο που δέχεται σειριακά το Arduino, μέσω USB από το Raspberry Pi. Η μεταβλητή αυτή ενημερώνεται κάθε φορά που γίνεται κάποιο γεγονός στην σειριακή θύρα, με την χρήση της συνάρτησης `serialEvent`. Με την ανάγνωση της μεταβλητής `stringComplete` τυπώνεται στην σειριακή έξοδο ένα μήνυμα επιβεβαίωσης, το οποίο εκμεταλλεύεται στην συνέχεια το Raspberry Pi όπως αναλύεται σε επόμενη ενότητα. Ακολουθεί ο έλεγχος της μεταβλητής και η εκτέλεση ανάλογων ενεργειών (**Πίνακας 7**).

stringComplete	Σημασία	Ενέργεια
cu	Camera up / κάμερα πάνω	Αύξηση της γωνίας του σερβοκινητήρα 1 για κλίση προς τα επάνω
cd	Camera down / κάμερα κάτω	Μείωση της γωνίας του σερβοκινητήρα 1 για κλίση προς τα κάτω
cl	Camera left / κάμερα αριστερά	Αύξηση της γωνίας του σερβοκινητήρα 2 για κλίση προς τα αριστερά
cr	Camera right / κάμερα δεξιά	Μείωση της γωνίας του σερβοκινητήρα 2 για κλίση προς τα δεξιά
cc	Camera center / κάμερα στο κέντρο	Επαναφορά των δύο σερβοκινητήρων στις 90° - κέντρο
cR	Camera Right / κάμερα Δεξιά	Περιστροφή του σερβοκινητήρα 2 στις 20°- 60° δεξιά από το κέντρο
cL	Camera Left / κάμερα Αριστερά	Περιστροφή του σερβοκινητήρα 2 στις 180°- 90° αριστερά από το κέντρο
mf	Move forward / κίνηση προς τα εμπρός	Προς τα εμπρός περιστροφή του κινητήρα 1, ακροδέκτες 2,3 (1,0)
mb	Move backwards / κίνηση προς τα πίσω	Προς τα πίσω περιστροφή του κινητήρα 1, ακροδέκτες 2,3 (0,1)
ml	Move left / κίνηση προς τα αριστερά	Προς τα εμπρός περιστροφή του κινητήρα 2 στο τιμόνι, περιστροφή τιμονιού στα αριστερά, ακροδέκτες 4,7 (1,0)
mr	Move right / κίνηση προς τα δεξιά	Προς τα πίσω περιστροφή του κινητήρα 2 στο τιμόνι, περιστροφή τιμονιού στα δεξιά, ακροδέκτες 4,7 (0,1)
ms	Move stop / σταμάτημα κίνησης	Σταμάτημα κίνησης, ακροδέκτες 2,3 (0,0)
ss	Steering stop / σταμάτημα στριψίματος	Σταμάτημα περιστροφής τιμονιού, ακροδέκτες 4,7 (0,0)

Πίνακας 7: Ενέργειες ελέγχου βάσει της μεταβλητής εισόδου του Arduino

Τέλος η allowmoving, όπως αναφέρθηκε προηγουμένως, αποφασίζει για το αν θα επιτραπεί ή όχι η κίνηση του οχήματος σε κάθε διεύθυνση. Αναλυτικότερα ελέγχει την απόσταση στους δύο μπροστά αισθητήρες απόστασης και την κατάσταση του «έξυπνου» προφυλακτήρα. Αν δεν ικανοποιούνται στα κριτήρια ασφάλειας τότε ακινητοποιεί το όχημα και δεν επιτρέπει την προς τα εμπρός κίνηση μέχρι την επόμενη ανανέωση. Τα κριτήρια ασφαλείας είναι:

- α) Η απόσταση σε έναν ή και στους δύο αισθητήρες να είναι μικρότερη από 40cm
- β) Ο «έξυπνος» προφυλακτήρας να μην είναι πατημένος

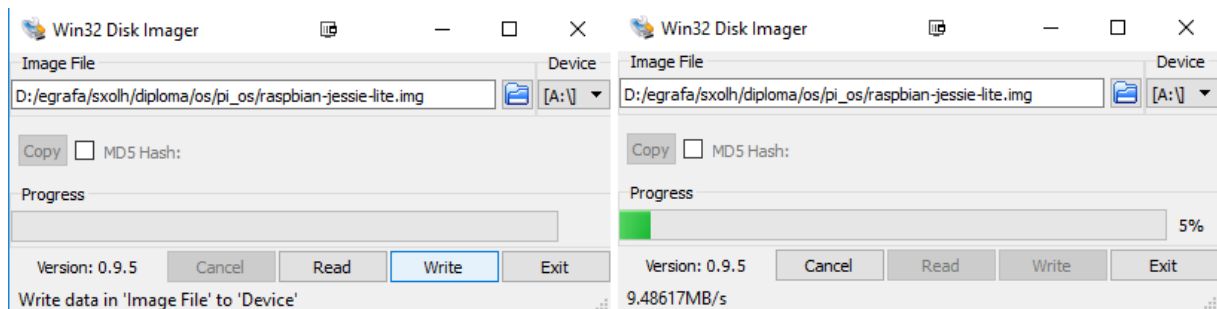
Στην δεύτερη περίπτωση το σύστημα δίνει στους δύο μπροστινούς αισθητήρες απόστασης την τιμή «1». Με αυτόν τον τρόπο το Raspberry Pi μπορεί να «καταλάβει» αν έχει πατηθεί ο «έξυπνος» προφυλακτήρας. Την πληροφορία αυτή εκμεταλλεύεται η λειτουργία αποφυγής εμποδίων (**4.3.4 Λειτουργία αποφυγής εμποδίων**) με σκοπό την προσπάθεια αποφυγής μικρότερων εμποδίων, που το όχημα δεν μπορεί να αντιληφθεί μέσω των αισθητήρων.

Ομοίως για την πίσω κίνηση εφαρμόζονται οι ίδιοι έλεγχοι και ενέργειες, με εξαίρεση την χρήση του «έξυπνου» προφυλακτήρα. Η κίνηση σε κάθε κατεύθυνση διαχειρίζεται ξεχωριστά από το σύστημα για δύο λόγους. Πρώτον η κίνηση προς τα εμπρός επηρεάζεται από διαφορετικές συνθήκες, απ' ότι η προς τα πίσω. Δεύτερον πρέπει να υπάρχει η δυνατότητα να απαγορευτεί μόνο η μία από τις δύο και όχι και οι δύο ταυτόχρονα. Για παράδειγμα, αν συμβεί σύγκρουση στο μπροστινό του οχήματος, σταματάει και έχει την δυνατότητα να κινηθεί προς τα πίσω για να αποφύγει το εμπόδιο. Επιπλέον υπάρχει και η δυνατότητα αναίρεσης κάποιας απαγόρευσης. Συνεχίζοντας το προηγούμενο παράδειγμα, μόλις το όχημα απομακρυνθεί από το εμπόδιο, μπορεί να κινηθεί ξανά προς τα εμπρός. Τέλος η απόσταση ασφαλείας που αναφέρθηκε (40cm) είναι αποτέλεσμα πειραματικών δοκιμών που έγιναν. Εξαρτάται από το βάρος και την ταχύτητα του οχήματος, καθώς και από άλλους παράγοντες όπως η κλίση εδάφους.

4.2 Ρύθμιση συστήματος του Raspberry Pi

Μετά την ανάλυση της λειτουργίας των Arduino, επόμενο στην σειρά είναι το Raspberry Pi, το οποίο είναι η «καρδιά» του συστήματος. Στην παρούσα υλοποίηση

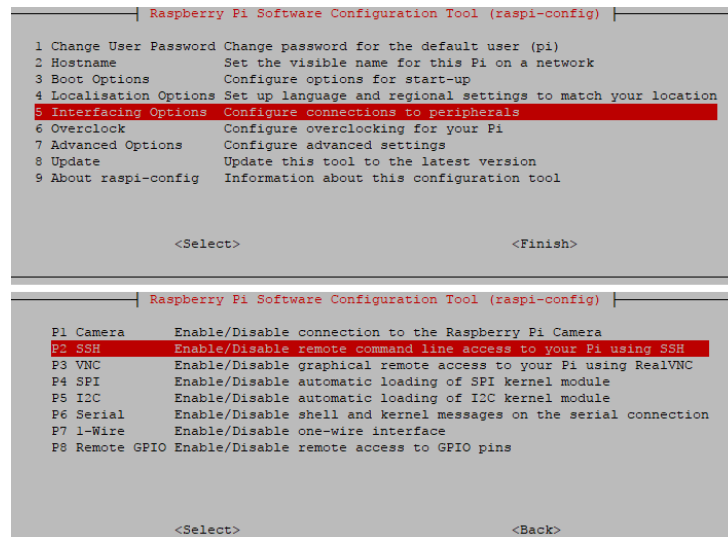
χρησιμοποιήθηκε για το λειτουργικό του συστήματος το εικονικό αρχείο Raspbian Jessie Lite 4.4.34.img. Το συγκεκριμένο αρχείο είναι το επίσημο λειτουργικό που παρέχει η Raspberry για της πλακέτες της και βγαίνει σε δύο εκδόσεις. Η μία έκδοση είναι η Lite, ενώ η δεύτερη η WITH PIXEL, η οποία παρέχει και γραφικό περιβάλλον. Και οι δύο εκδόσεις είναι συμβατές για όλες τις πλακέτες της Raspberry. Στο παρόν σύστημα χρησιμοποιήθηκε η Lite έκδοση για εξοικονόμηση πόρων του συστήματος. Η εγκατάσταση του αρχείου έγινε με την χρήση του προγράμματος Win32DiskImager σε Windows. Συγκεκριμένα η SD κάρτα μνήμης συνδέθηκε σε έναν H/Y μέσω μιας συσκευής ανάγνωσης καρτών (card reader). Έπειτα με την χρήση του Win32DiskImager επιλέχθηκε το αρχείο Raspbian Jessie Lite 4.4.34.img (**Εικόνα 35**). Τέλος με την επιλογή εγγραφή (write), η κάρτα διαμορφώθηκε καταλλήλως και το λειτουργικό εγκαταστάθηκε με επιτυχία. Με την εισαγωγή της κάρτας στο Raspberry Pi το λειτουργικό σύστημα είναι έτοιμο για χρήση.



Εικόνα 35: Εγκατάσταση λειτουργικού στο Raspberry Pi

4.2.1 Ενεργοποίηση SSH

Η πρώτη κίνηση ήταν η ενεργοποίηση του SSH εξυπηρετή με σκοπό την απομακρυσμένη χρήση του Raspberry Pi. Με δικαιώματα διαχειριστή (sudo su ή sudo και εντολή) δόθηκε η εντολή raspi-config, η οποία οδηγεί σε ένα μενού ρυθμίσεων (**Εικόνα 36**) [44] [45]. Από αυτό το μενού μεταβήκαμε στην πέμπτη επιλογή (Επιλογές διασύνδεσης , Interfacing Options). Στην συνέχεια επιλέξαμε την δεύτερη επιλογή (P2 SSH) από το νέο μενού. Με αυτόν τον τρόπο ενεργοποιήθηκε η υπηρεσία SSH εξυπηρετή στο Raspberry Pi.



Εικόνα 36: Ενεργοποίηση SSH στο Raspberry Pi

4.2.2 Εγκατάσταση Wiring GPIO

Επόμενο βήμα ήταν η εγκατάσταση της εντολής gpio (wiring_gpio), η οποία παρέχει την δυνατότητα ελέγχου των ακροδεκτών GPIO του Raspberry Pi από την γραμμή εντολών [46]. Αρχικά με την χρήση του APT βοηθητικού προγράμματος διαχείρισης πακέτων έγινε ενημέρωση της λίστας των πακέτων (apt-get update) [45]. Στην συνέχεια εγκαταστάθηκε με παρόμοιο τρόπο (apt-get install) το πακέτο git-core, το οποίο επιτρέπει την διαχείριση αποθετηρίου Git (Εικόνα 37).

```
root@raspberrypi:/home/pi# apt-get install git-core
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  git git-man liberror-perl rsync
Suggested packages:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch git-cvs git-mediawiki
  git-svn
The following NEW packages will be installed:
  git git-core git-man liberror-perl rsync
0 upgraded, 5 newly installed, 0 to remove and 52 not upgraded.
Need to get 3,995 kB/4,390 kB of archives.
After this operation, 21.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Processing triggers for systemd (215-17+deb8u6) ...
root@raspberrypi:/home/pi#
```

Εικόνα 37: Εγκατάσταση git-core

Με χρήση του git-core κατέβηκε η τελευταία έκδοση του εργαλείου wiring_gpio. Τέλος με την εντολή δόμησης (build), έγινε μεταγλώττιση και εγκατάσταση του εργαλείου wiring_gpio (Εικόνα 38).

```
root@raspberrypi:/home/pi# git clone git://git.drogon.net/wiringPi
Cloning into 'wiringPi'...
remote: Counting objects: 1151, done.
remote: Compressing objects: 100% (957/957), done.
remote: Total 1151 (delta 804), reused 212 (delta 142)
Receiving objects: 100% (1151/1151), 380.39 KiB | 254.00 KiB/s, done.
Resolving deltas: 100% (804/804), done.
Checking connectivity... done.
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi# cd wiringPi/
root@raspberrypi:/home/pi/wiringPi# git pull origin
Already up-to-date.
root@raspberrypi:/home/pi/wiringPi#
root@raspberrypi:/home/pi/wiringPi# ./build
wiringPi Build script
=====

WiringPi Library
[UnInstall]
[Compile] wiringPi.c
[Compile] wiringSerial.c
[Compile] piHiPri.c
[Compile] wiringShift.c

All Done.

NOTE: To compile programs with wiringPi, you need to add:
      -lwiringPi
      to your compile line(s) To use the Gertboard, MaxDetect, etc.
      code (the devLib), you need to also add:
      -lwiringPiDev
      to your compile line(s).

root@raspberrypi:/home/pi/wiringPi# gpio -v
gpio version: 2.44
Copyright (c) 2012-2017 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Model B, Revision: 02, Memory: 512MB, Maker: Sony
  * Device tree is enabled.
  *--> Raspberry Pi Model B Rev 2
  * This Raspberry Pi supports user-level GPIO access.
root@raspberrypi:/home/pi/wiringPi#
```

Εικόνα 38: Εγκατάσταση GPIO Wiring

Το εργαλείο wiring_gpio χρησιμοποιείται για την διαχείριση των ακροδεκτών GPIO ως έξοδος, με σκοπό το άναμμα/σβήσιμο των φώτων του οχήματος. Για την σωστή χρήση πρέπει πρώτα να οριστεί ο κατάλληλος ακροδέκτης ως ακροδέκτης εξόδου. Στην συνέχεια μπορούμε να ορίσουμε αν θα είναι «ανοιχτός» ή όχι (0/1). Στο παρακάτω παράδειγμα γίνεται χειρισμός των λευκών φώτων του οχήματος που είναι συνδεδεμένα στον ακροδέκτη 12 (GPIO 18) του Raspberry Pi 3 (Εικόνα 21). Για τον ορισμό του ακροδέκτη ως έξοδο χρησιμοποιείται η εντολή *gpio export 18 out*. Για το άναμμα και σβήσιμο δίνονται αντίστοιχα οι εντολές *gpio write 1 1* και *gpio write 1 0*. Η αρίθμηση των ακροδεκτών περιγράφεται στην παρακάτω εικόνα (Εικόνα 39) ενώ μπορεί να εμφανιστεί και με την χρήση της εντολής *gpio readall*.

GPIO Numbers

Raspberry Pi B Rev 1 P1 GPIO Header				Raspberry Pi A/B Rev 2 P1 GPIO Header				Raspberry Pi B+ B+ J8 GPIO Header			
Pin No.				Pin No.				Pin No.			
3.3V	1	2	5V	3.3V	1	2	5V	3.3V	1	2	5V
GPIO0	3	4	5V	GPIO2	3	4	5V	GPIO2	3	4	5V
GPIO1	5	6	GND	GPIO3	5	6	GND	GPIO3	5	6	GND
GPIO4	7	8	GPIO14	GPIO4	7	8	GPIO14	GPIO4	7	8	GPIO14
GND	9	10	GPIO15	GND	9	10	GPIO15	GND	9	10	GPIO15
GPIO17	11	12	GPIO18	GPIO17	11	12	GPIO18	GPIO17	11	12	GPIO18
GPIO21	13	14	GND	GPIO27	13	14	GND	GPIO27	13	14	GND
GPIO22	15	16	GPIO23	GPIO22	15	16	GPIO23	GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24	3.3V	17	18	GPIO24	3.3V	17	18	GPIO24
GPIO10	19	20	GND	GPIO10	19	20	GND	GPIO10	19	20	GND
GPIO9	21	22	GPIO25	GPIO9	21	22	GPIO25	GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8	GPIO11	23	24	GPIO8	GPIO11	23	24	GPIO8
GND	25	26	GPIO7	GND	25	26	GPIO7	GND	25	26	GPIO7
								DNC	27	28	DNC
								GPIO5	29	30	GND
								GPIO6	31	32	GPIO12
								GPIO13	33	34	GND
								GPIO19	35	36	GPIO16
								GPIO26	37	38	GPIO20
								GND	39	40	GPIO21

Key		
Power +	UART	
GND	SPI	
IC	GPIO	

Εικόνα 39: Αρίθμηση ακροδεκτών GPIO

4.2.3 Εγκατάσταση και ρύθμιση Motion

Η εγκατάσταση του Motion κρίθηκε απαραίτητη για την υλοποίηση της παρακολούθησης μέσω της κάμερας σε πραγματικό χρόνο [47]. Το Motion μαζί με την δυνατότητα αυτή, προσφέρει πλήθος ρυθμίσεων. Η εγκατάστασή του γίνεται με χρήση της εντολής *apt-get install* (Εικόνα 40).

```
root@raspberrypi:/home/pi# apt-get install motion
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libavcodec56 libavformat56 libavresample2 libavutil54 libdrm-amdgpu libdrm-nouveau2 libdrm-radeon1 libelf1
  libgsm1 libjpeg8 liblvm3.9 libmp3lame0 libmysqlclient18 libogg0 libopenjpeg5 libopus0 liborc-0.4-0
  libschroedinger-1.0-0 libspeex1 libtheora0 libval libvorbis0a libvorbisenc2 libvpx1 libx11-xcb1 libx264-142
  libxcb-dri2-0 libxcb-dri3-0 libxcb-present0 libxcb-sync1 libxshmfence1 libxvidcore4 mesa-va-drivers mysql-common
  va-driver-all
Suggested packages:
  opus-tools speex mysql-client postgresql-client
Recommended packages:
  ffmpeg
The following NEW packages will be installed:
  libavcodec56 libavformat56 libavresample2 libavutil54 libdrm-amdgpu libdrm-nouveau2 libdrm-radeon1 libelf1
  libgsm1 libjpeg8 liblvm3.9 libmp3lame0 libmysqlclient18 libogg0 libopenjpeg5 libopus0 liborc-0.4-0
  libschroedinger-1.0-0 libspeex1 libtheora0 libval libvorbis0a libvorbisenc2 libvpx1 libx11-xcb1 libx264-142
  libxcb-dri2-0 libxcb-dri3-0 libxcb-present0 libxcb-sync1 libxshmfence1 libxvidcore4 mesa-va-drivers motion
  mysql-common va-driver-all
0 upgraded, 36 newly installed, 0 to remove and 52 not upgraded.
Need to get 21.3 MB of archives.
After this operation, 76.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Done.
Setting up va-driver-all:armhf (1.7.3-2) ...
Processing triggers for libc-bin (2.19-18+deb8u7) ...
Processing triggers for systemd (215-17+deb8u6) ...
root@raspberrypi:/home/pi#
```

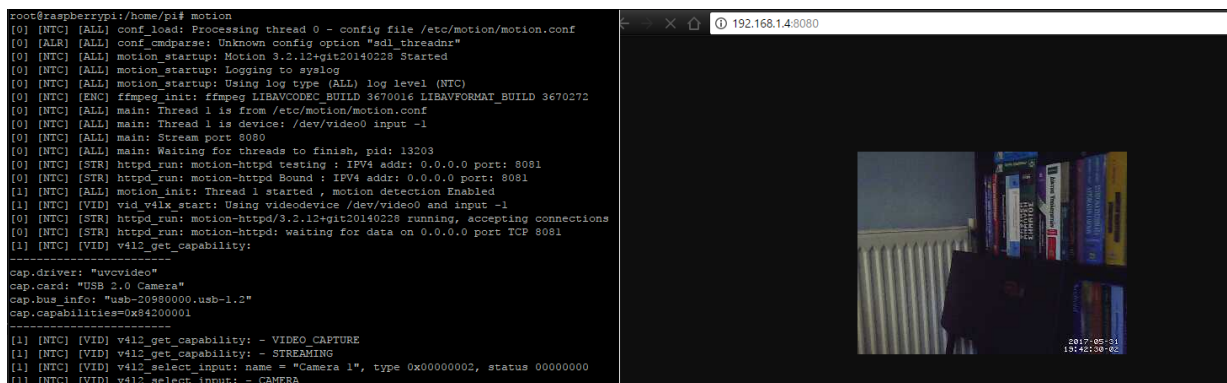
Εικόνα 40: Εγκατάσταση Motion

Στην παρούσα υλοποίηση η έκδοση του Motion που χρησιμοποιήθηκε είναι η v3.2.12. Για την λειτουργία του Motion είναι απαραίτητο να γίνουν μερικές ρυθμίσεις. Οι ρυθμίσεις γίνονται μέσω της επεξεργασίας του αρχείου `motion.conf` που βρίσκεται στην τοποθεσία `/etc/motion`, με την χρήση της εντολής `nano /etc/motion/motion.conf`. Οι απαραίτητες ρυθμίσεις δίνονται στον παρακάτω πίνακα (**Πίνακας 8**).

Παράμετρος	Τιμή	Εξήγηση
<code>v4l2_palette</code>	15	ανάλογα την κάμερα που χρησιμοποιούμε
<code>framerate</code>	30	αρκετά καρέ για ομαλή κίνηση
<code>output_pictures</code>	off	δεν πρέπει να αποθηκεύονται εικόνες γιατί καταναλώνονται άσκοπα πόροι του συστήματος
<code>ffmpeg_output_movies</code>	off	δεν πρέπει να αποθηκεύονται βίντεο γιατί καταναλώνονται άσκοπα πόροι του συστήματος
<code>stream_port</code>	8080	η προεπιλεγμένη πόρτα για ροή δεδομένων βίντεο
<code>stream_maxrate</code>	30	ίδια τιμή με το <code>framerate</code>
<code>stream_localhost</code>	off	επιτρέπει την πρόσβαση στην ροή δεδομένων σε άλλα μηχανήματα
<code>webcontrol_port</code>	8081	πρέπει να είναι διαφορετική από την πόρτα ροής δεδομένων βίντεο

Πίνακας 8: Ρυθμίσεις Motion

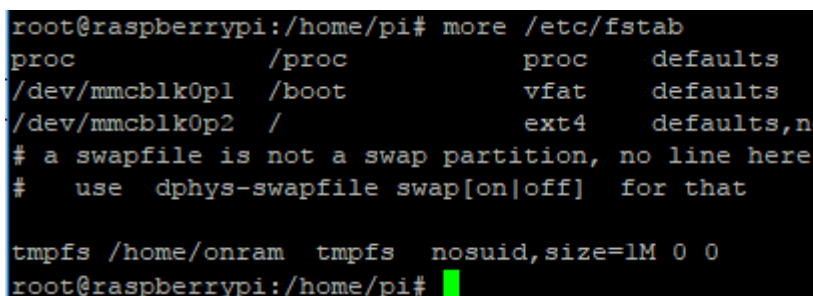
Για την εκτέλεση του εργαλείου αυτού αρκεί να δώσουμε την εντολή `motion`. Για επαλήθευση της σωστής λειτουργίας μπορούμε να συνδεθούμε, από κάποιο μηχάνημα εντός του τοπικού δικτύου, μέσω ενός φυλλομετρητή, στην IP του Raspberry Pi, στην πόρτα 8080. Με αυτόν τον τρόπο εμφανίζεται η εικόνα που λαμβάνει η κάμερα (**Εικόνα 41**). Τέλος για την αυτόματη εκτέλεση του Motion με την εκκίνηση του λειτουργικού συστήματος, δίνουμε την τιμή `yes` στην παράμετρο `start_motion_daemon` (με χρήση της εντολής `nano`) που βρίσκεται στη διαδρομή `/etc/default/motion`.



Εικόνα 41: Ορθή λειτουργία Motion

4.2.4 Ρύθμιση προσωρινών αρχείων και εφαρμογών εκκίνησης

Το παρόν σύστημα χρησιμοποιεί τρία προσωρινά αρχεία για την ανταλλαγή πληροφοριών (camera.txt, movement.txt, sonar.txt). Τα αρχεία αυτά βρίσκονται στον φάκελο /home/onram, και όπως υπονοεί το όνομά του προσαρτάται στην RAM του συστήματος. Η προσάρτηση αυτή γίνεται με την χρήση του προσωρινού συστήματος αρχείων tmpfs, ενώ προσφέρει πολύ γρήγορη απόκριση του συστήματος και μακροζωία στην κάρτα SD [48]. Η διαδικασία προσάρτησης γίνεται με την προσθήκη της γραμμής *tmpfs /home/onram tmpfs, nosuid, size=1M 0 0* στο αρχείο */etc/fstab* (Εικόνα 42). Επι της ουσίας ορίζουμε τον φάκελο onram ως tmpfs με μέγιστο μέγεθος στην RAM 1MB (δεν χρειάζεται παραπάνω). Επιπλέον με το 0 0 θέτουμε το σύστημα να μην κάνει αντίγραφο ασφαλείας γι' αυτόν τον φάκελο και να μην τον ελέγχει για λάθη κατά την εκκίνηση του συστήματος, με σκοπό την αποφυγή περιττής κατανάλωσης πόρων [49]. Τέλος με την nosuid δεν επιτρέπουμε τα bits ορισμού αναγνωριστικού χρήστη (set-user-identifier) και ορισμού αναγνωριστικού ομάδας (set-group-identifier) να τεθούν σε ισχύ για λόγους ασφαλείας [50].



```
root@raspberrypi:/home/pi# more /etc/fstab
proc                /proc              proc               defaults           0                 0
/dev/mmcblk0p1      /boot              vfat               defaults           0                 2
/dev/mmcblk0p2      /                  ext4               defaults,noatime   0                 1
# a swapfile is not a swap partition, no line here
#   use dphys-swapfile swap[on|off] for that

tmpfs /home/onram tmpfs nosuid,size=1M 0 0
root@raspberrypi:/home/pi#
```

Εικόνα 42: Προσάρτηση προσωρινού αρχείου

Με την ολοκλήρωση της ρύθμισης για την προσάρτηση των προσωρινών αρχείων το μόνο που έμεινε είναι η προσθήκη των αρχείων του λογισμικού που υλοποιήθηκε. Τα αρχεία αυτά είναι τα, *arduino_com_full.py* και *ip_lcd.py*, τα οποία είναι γραμμένα σε Python όπως αναφέρθηκε σε προηγούμενο κεφάλαιο. Το πρώτο είναι υπεύθυνο για την επικοινωνία μεταξύ Raspberry Pi και Arduino, ενώ το δεύτερο τυπώνει σε τακτά χρονικά διαστήματα την τοπική IP του συστήματος στην εγκατεστημένη LCD οθόνη. Η αναλυτική λειτουργία τους περιγράφεται σε επόμενες ενότητες. Για την αυτόματη εκκίνηση των δύο αυτών λογισμικών υλοποιήθηκαν δύο αρχεία σεναρίου (script, .sh), τα *launcher.sh* και *launcher1.sh*. Το πρώτο εκτελεί το αρχείο *ip_lcd.py*. Το δεύτερο δημιουργεί τα τρία προσωρινά αρχεία που

αναφέρθηκαν προηγουμένως. Επιπλέον δίνει δικαιώματα ανάγνωσης, εκτέλεσης και εγγραφής στον απλό χρήστη για τα προσωρινά αρχεία και για τις δύο συσκευές USB (ttyACM0, ttyUSB0), που αντιστοιχούν στα δύο Arduino του συστήματος. Παράλληλα ορίζει τους ακροδέκτες (GPIO 17, GPIO 18) του Raspberry Pi ως έξοδοι για τα φώτα του οχήματος. Τέλος εκτελεί το αρχείο `arduino_com_full.py`.

Η αυτόματη εκκίνηση των δύο αρχείων σεναρίου γίνεται με την χρήση του `crontab`, το οποίο είναι ένα εργαλείο που επιτρέπει την αυτόματη και περιοδική εκτέλεση εφαρμογών στο παρασκήνιο [51]. Το `crontab` είναι προ-εγκατεστημένο στο λειτουργικό που χρησιμοποιήθηκε, ενώ η χρήση του γίνεται με την εντολή `crontab -e`, με δικαιώματα διαχειριστή. Αφού επιλέξουμε τον επιθυμητό κειμενογράφο προσθέτουμε τις γραμμές `@reboot sh /home/pi/scripts/launcher.sh` και `@reboot sh /home/pi/scripts/launcher1.sh` (Εικόνα 43), όπου θέτουμε την εκτέλεση σεναρίου κελύφους του αρχείου `launcher.sh` και αντίστοιχα του `launcher1.sh` κατά την επανεκκίνηση του συστήματος. Πρέπει να αναφερθεί ότι το `crontab` δεν επιτρέπει άμεση εκτέλεση κώδικα `python`. Επομένως υλοποιήθηκε το script `launcher.sh`, το οποίο τον εκτελεί έμμεσα. Τέλος και στα δύο λογισμικά, που είναι γραμμένα σε `python` (`arduino_com_full.py`, `ip_lcd.py`), χρησιμοποιούνται ατέρμονοι βρόγχοι, γεγονός που δεν επιτρέπει την εκτέλεση και των δύο εφαρμογών από το ίδιο script. Αυτό συμβαίνει επειδή το script «περιμένει» να τερματίσει η πρώτη εφαρμογή και στην συνέχεια να εκτελέσει τη δεύτερη. Γι' αυτόν τον λόγο χρησιμοποιήθηκαν δύο scripts και όχι ένα.

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
@reboot sh /home/pi/scripts/launcher.sh
@reboot sh /home/pi/scripts/launcher1.sh
```

Εικόνα 43: Ρύθμιση Crontab

4.3 Λογισμικό στο Raspberry Pi

Με την ολοκλήρωση των ρυθμίσεων του συστήματος, γνωρίζουμε ότι τα λογισμικά που υλοποιήθηκαν στα αρχεία `ip_lcd.py` και `arduino_com_full.py`, εκτελούνται με την εκκίνηση του συστήματος στο παρασκήνιο. Οπότε σε αυτό το σημείο θα αναλυθεί ο σκοπός και ο τρόπος λειτουργίας τους.

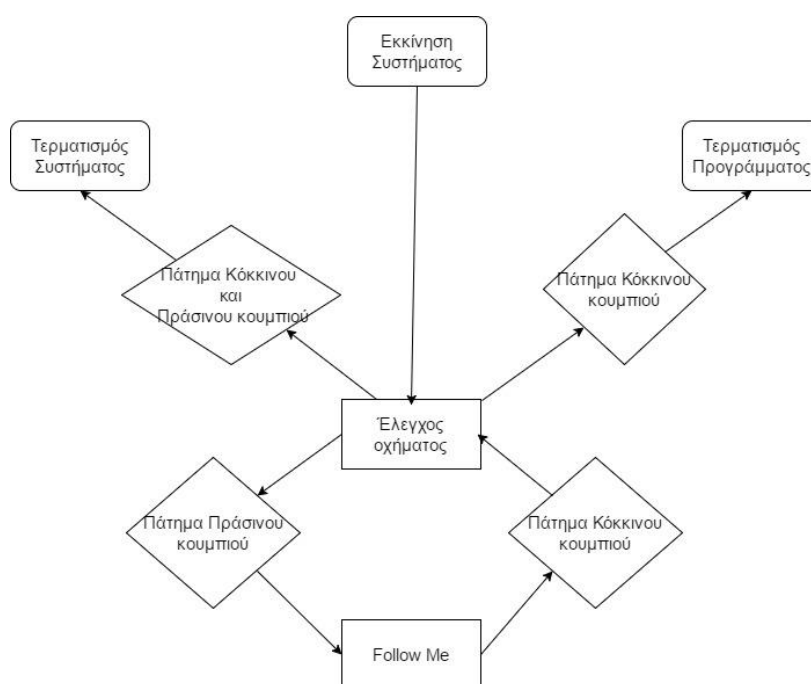
Αρχίζοντας από το `ip_lcd.py`, που είναι το απλούστερο από τα δύο, συμπεραίνουμε από το όνομά του ότι εμφανίζει την IP του συστήματος στην LCD οθόνη. Συγκεκριμένα έχει εισαχθεί η βιβλιοθήκη `Adafruit_CharLCD` η οποία ήταν προεγκαταστημένη στο λειτουργικό του Raspberry Pi, και είναι υπεύθυνη για την διαχείριση της οθόνης. Στο παρόν αρχείο δίνεται τόσο η διάταξη των ακροδεκτών του Raspberry Pi 3 Model B, όσο και του Raspberry Pi Model B, που χρησιμοποιήθηκαν για την LCD. Μετά την αρχικοποίηση της οθόνης ορίζουμε δύο εντολές, που εκτελούνται στη γραμμή εντολών, και επιστρέφουν την IP της ασύρματης και ενσύρματης σύνδεσης δικτύου. Σε έναν ατέρμονο βρόγχο ελέγχεται η IP της ενσύρματης σύνδεσης. Σε περίπτωση που δεν υπάρχει η προαναφερθείσα, ελέγχεται αυτή της ασύρματης, ενώ στην περίπτωση που δεν υπάρχει ούτε η δεύτερη τυπώνεται ανάλογο μήνυμα.

Το `arduino_com_full.py` είναι αρκετά πολυπλοκότερο, μιας και καλύπτει όλες τις υπόλοιπες ανάγκες λειτουργίας. Αρχικά πρέπει να αναφερθεί ότι για την διαχείριση του Bluetooth χρησιμοποιήθηκε η εξωτερική βιβλιοθήκη «`bt_proximity`» η οποία καλύπτει της ανάγκες της παρούσας διπλωματικής εργασίας [52]. Η βιβλιοθήκη «`bt_proximity`» απαιτεί την εγκατάσταση των πακέτων «`bluetooth`» και «`python-bluez`», τα οποία, όμως, ήταν προεγκατεστημένα. Στην συνέχεια για την εγκατάσταση της «`bt_proximity`», αφού έχει αποθηκευτεί στο μηχάνημα, δίνεται η εντολή `python setup.py install`.

Στην συνέχεια ορίζουμε τις δύο συσκευές Arduino για την σειριακή τους επικοινωνία, καθώς και την ταχύτητα αυτής. Συγκεκριμένα στο Arduino που ελέγχει μόνο τους αισθητήρες IR χρησιμοποιήθηκε η προκαθορισμένη ταχύτητα 9600. Αντίθετα στο δεύτερο Arduino, για γρηγορότερη απόκριση του συστήματος και υποστήριξη του πρωτοκόλλου επιβεβαίωσης που θα δούμε παρακάτω, χρησιμοποιήθηκε μεγαλύτερη ταχύτητα (115200). Έπειτα ορίζονται τα προσωρινά αρχεία που αναφέρθηκαν σε προηγούμενη ενότητα, και ο τρόπος που θα χρησιμοποιηθεί το καθένα. Τέλος γίνεται αρχικοποίηση κοινόχρηστων μεταβλητών, και τον ακροδεκτών GPIO που χρησιμοποιούνται. Παρέχεται έτοιμη διάταξη των GPIO τόσο σε Raspberry Pi 3 Model B όσο και σε Raspberry Pi Model B.

Το παρόν σύστημα αποτελείται από τρία LEDs (πράσινο, κόκκινο, μπλε) και δύο κουμπιά (πράσινο, κόκκινο). Με την εκκίνηση του συστήματος ξεκινάν τρεις διεργασίες (το κύριο πρόγραμμα, και δύο διεργασίες παιδιά) ενώ αρχικά είναι αναμμένο μόνο το κόκκινο LED. Το «arduino_com_full.py» είναι υπεύθυνο, τόσο για τον χειρισμό του οχήματος, με προώθηση των κατάλληλων μηνυμάτων στο κατάλληλο Arduino, όσο και για την λειτουργία συνοδείας ατόμου (follow me). Οι διεργασίες - παιδιά εκτελούνται παράλληλα με το κύριο πρόγραμμα, ενώ χρησιμοποιούνται για την ενημέρωση των μεταβλητών απόστασης εμποδίου και έντασης σήματος Bluetooth αντίστοιχα.

Το κύριο πρόγραμμα κατά την εκκίνηση, εκτελεί την λειτουργία χειρισμού του οχήματος. Με το πάτημα του πράσινου κουμπιού η λειτουργία χειρισμού διακόπτεται, και εκτελείται η λειτουργία συνοδείας «follow me». Από αυτό το σημείο, αν πατηθεί το κόκκινο κουμπί θα διακοπεί η λειτουργία «follow me» και θα συνεχίσει η λειτουργία χειρισμού. Κατά την διάρκεια εκτέλεσης της λειτουργίας χειρισμού αν πατηθεί το κόκκινο κουμπί τερματίζει το κύριο πρόγραμμα και τα παιδιά διεργασίες του. Αν πατηθεί το κόκκινο και πράσινο κουμπί μέχρι να ανάψει το πράσινο LED, τότε, μετά από ένα δευτερόλεπτο το πράσινο LED σβήνει και ακολουθεί πλήρης τερματισμός του συστήματος (shutdown). Μόλις σβήσει και το κόκκινο LED, το σύστημα έχει τερματιστεί με επιτυχία και μπορούμε να διακόψουμε με ασφάλεια την παροχή ρεύματος (τοποθέτηση στην θέση OFF του διακόπτη τροφοδοσίας συστήματος). Η συνολική λειτουργία του συστήματος παρουσιάζεται στην **Εικόνα 44**.



Εικόνα 44: Λειτουργία Συστήματος

Αντίστοιχα οι πιθανοί συνδυασμοί της κατάστασης των LEDs, και η σημασία του καθένα παρουσιάζονται στον παρακάτω πίνακα (**Πίνακας 9**). Οι συνδυασμοί που απουσιάζουν δεν έχουν κάποια συγκεκριμένη σημασία ή δεν είναι σχεδιασμένη η ύπαρξή τους στο σύστημα. Σε περίπτωση που εμφανιστεί κάποιος τέτοιος συνδυασμός (εκτός πίνακα), πρέπει να γίνει έρευνα σφάλματος στο λογισμικό και υλικό μέρος του συστήματος.

Κόκκινο LED	Πράσινο LED	Μπλε LED	Σημασία
Αναμμένο	Σβηστό	Σβηστό	Εκτελείται η λειτουργία χειρισμού
Σβηστό	Αναμμένο	Σβηστό	Εκτελείται η λειτουργία «Follow me»
Αναμμένο	Αναμμένο	Σβηστό	Κατά την εκτέλεση της «Follow me» οι δύο ακριανοί αισθητήρες IR (δεξιός και αριστερός) δέχτηκαν το ίδιο σήμα, το όχημα ακινητοποιείται
Σβηστό	Αναμμένο	Αναμμένο	Κατά την εκτέλεση της «Follow me» το όχημα πλησίασε πολύ κοντά στον χρήστη (μέσω bluetooth), το όχημα ακινητοποιείται
Αναμμένο	Αναμμένο	Αναμμένο	Ισχύουν οι δύο παραπάνω συνθήκες ταυτόχρονα
Αναμμένο	Αναμμένο για 1s	Σβηστό	Εκκίνηση διαδικασίας τερματισμού
Σβηστό	Σβηστό	Σβηστό	Η εφαρμογή ή όλο το σύστημα έχουν τερματίσει

Πίνακας 9: Αναμενόμενες καταστάσεις των LEDs

Τέλος πρέπει να αναφερθεί ότι για την πρόσβαση σε κοινόχρηστες μεταβλητές, είτε από το κύριο πρόγραμμα, είτε από τις διεργασίες - παιδιά του, έγινε χρήση κλειδαριών (locks). Η χρήση αυτού του τρόπου προσπέλασης είναι απαραίτητη για την αντιμετώπιση συνθηκών ανταγωνισμού μεταξύ των διεργασιών. Ακολουθεί η διεξοδική ανάλυση καθενός από τις αναφερόμενες διεργασίες.

4.3.1 Διεργασία ανάγνωσης απόστασης εμποδίων

Η διεργασία ανάγνωσης απόστασης εμποδίων «read_sonar», εκτελείται σε ατέρμονο βρόγχο μέχρι να δοθεί από την διεργασία - πατέρα σήμα τερματισμού. Κατά την διάρκεια της εκτέλεσης γίνεται σειριακή ανάγνωση από τη θήρα USB, στην οποία είναι συνδεδεμένο το Arduino με τους αισθητήρες απόστασης. Η ανάγνωση τιμής εκτελείται μέχρι η εισερχόμενη τιμή να είναι τιμή αποστάσεων, πληρώνοντας συγκεκριμένη δομή (τιμές χωρισμένες με « - »).

Μόλις επιτευχθεί η εισαγωγή τέτοιας τιμής, ελέγχεται αν συμβαδίζει με τις προϋποθέσεις (πχ να έχει 5 μετρήσεις) και γίνονται οι απαραίτητες διορθώσεις, όπου αυτό είναι αναγκαίο (πχ σε περίπτωση κενού τοποθετείται η τιμή «0»). Στην συνέχεια οι τιμές αυτές αυτούσιες εγγράφονται στην πρώτη θέση (πάντα) του προσωρινού αρχείου «camera.txt». Τέλος όλες οι τιμές, από αλφαριθμητικό που είναι, μετατρέπονται σε αριθμούς και προστίθενται σε μια μεταβλητή μονοδιάστατου πίνακα την «sonar_sens». Η διάταξη της μεταβλητής αυτής, από το πρώτο στοιχείο στο τελευταίο, περιέχει τις αποστάσεις από τους αισθητήρες: μπροστά δεξιά (FR), μπροστά αριστερά (FL), πίσω δεξιά (BR), πίσω αριστερά (BL), κάμερα(CM) (**Πίνακας 10**).

0	1	2	3	4
FR	FL	BR	BL	CM

Πίνακας 10: Μεταβλητή sonar_sens

4.3.2 Διεργασία ανάγνωσης Bluetooth

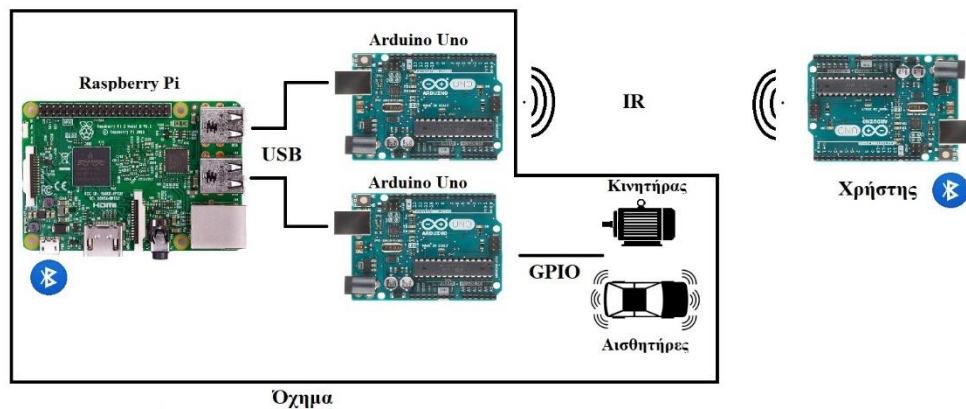
Η δεύτερη διεργασία - παιδί του συστήματος είναι η «read_bluetooth», η οποία χρησιμοποιείται για τον υπολογισμό της απόστασης μεταξύ οχήματος και χρήστη. Επι της ουσίας μετράει την ισχύ του σήματος και αν αυτή είναι πολύ μεγάλη, τότε υποτίθεται πως ο χρήστης (ο πομπός του τουλάχιστον) είναι κοντά. Για την μέτρηση της έντασης, όπως αναφέρθηκε και προηγουμένως χρησιμοποιήθηκε η βιβλιοθήκη «bt_proximity». Μέσω αυτής δημιουργείται ένα αντικείμενο bluetooth με αναγνωριστικό την MAC διεύθυνση που ορίσαμε. Προφανώς η MAC διεύθυνση αντιστοιχεί σε αυτή που κατέχει η συσκευή Bluetooth του πομπού.

Έπειτα σε έναν ατέρμονο βρόγχο γίνεται σύνδεση (pairing) με το Bluetooth του πομπού, και μετρίεται η ισχύς του σήματος. Σε περίπτωση αποτυχίας σύνδεσης η τιμή της ισχύος είναι 0, ενώ οι προσπάθειες σύνδεσης συνεχίζονται μέχρι να επιτευχθεί η σύνδεση, ή να τερματιστεί η διεργασία. Προφανώς προσπάθεια σύνδεσης γίνεται και στην περίπτωση αποσύνδεσης της συσκευής (πχ βγήκε εκτός εμβέλειας). Μάλιστα η παρούσα βιβλιοθήκη

καθυστερούσε στην αναγνώριση αποσύνδεσης. Στο παρόν σύστημα βελτιώθηκε αυτή η διαδικασία μετρώντας τις συνεχόμενες ίδιες τιμές της ισχύος του σήματος. Η ισχύς δεν μπορεί να έχει την ίδια ακριβώς τιμή για πολλές συνεχόμενες φορές. Έτσι ορίστηκε ένα κατώφλι το οποίο περιγράφει τον αποδεκτό αριθμό συνεχόμενων ίδιων τιμών της ισχύος. Τέλος η τιμή αυτή αποθηκεύεται σε μια κοινόχρηστη μεταβλητή, την «rssi», έτσι ώστε να μπορεί να χρησιμοποιηθεί και από άλλες διεργασίες. Η διεργασία αυτή όπως και η «read_sonar» τερματίζει μόλις δεχθεί συγκεκριμένο σήμα τερματισμού από τη διεργασία - πατέρα.

4.3.3 Λειτουργία «Follow me»

Η μία από τις δύο λειτουργίες της κύριας διεργασίας είναι αυτή της «Follow me». Για την επίτευξη της λειτουργίας αυτής, αξιοποιούνται τα δεδομένα από τους αισθητήρες απόστασης και της έντασης ισχύος του Bluetooth, μέσω της διεργασίας «read_sonar» και της «read_bluetooth» αντίστοιχα. Επιπλέον η παρούσα διεργασία ελέγχει τις τιμές της μεταβλητής «command_hist» που τυπώνει το Arduino δέκτη IR (4.1.2 Arduino δέκτη IR). Η γενικότερη χρήση των αισθητήρων του συστήματος παρουσιάζεται στην **Εικόνα 45**.



Εικόνα 45: Χρήση αισθητήρων συστήματος

Το περιεχόμενο που επιστρέφει η σειριακή ανάγνωση του Arduino αποθηκεύεται (μετά από επεξεργασία) στη μεταβλητή «sensors». Η μεταβλητή αυτή είναι ένα αλφαριθμητικό τεσσάρων χαρακτήρων, όπου κάθε ένας μπορεί να είναι, είτε 0, είτε 1. Ουσιαστικά θα μπορούσε να γίνει μετατροπή αυτής, σε έναν δυαδικό μονοδιάστατο πίνακα τεσσάρων θέσεων, αλλά τέτοια αλλαγή δεν θεωρήθηκε αναγκαία. Από αριστερά προς τα δεξιά ο κάθε χαρακτήρας

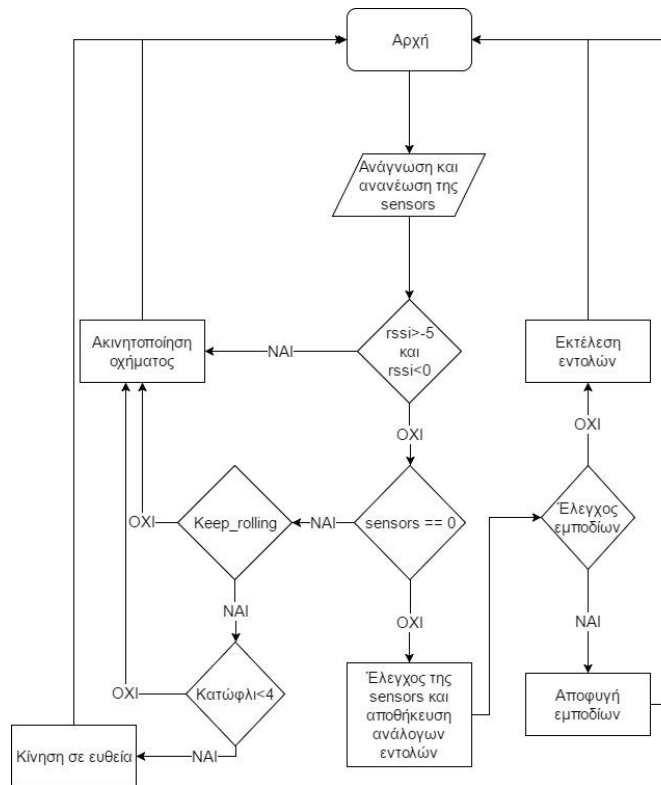
συμβολίζει: αριστερός IR (L IR), δεξιός IR (R IR), μπροστά αριστερός IR (FL IR), μπροστά δεξιός IR (FR IR). Ανάλογα με την τιμή της «sensors», το σύστημα αποφασίζει την ενέργεια που θα εκτελέσει (**Πίνακας 11**). . Όλες οι εντολές χειρισμού του οχήματος (κίνηση, σταμάτημα, στροφή, κλπ) προωθούνται στο Arduino χειρισμού (**4.1.3 Arduino Ελέγχου οχήματος, Πίνακας 7**) μέσω της θήρας USB. Η προώθηση αυτή γίνεται με την συνεχόμενη σειριακή τύπωση της εντολής χειρισμού. Η τύπωση θα σταματήσει μόνο στην περίπτωση που ληφθεί μήνυμα επιβεβαίωσης από το Arduino, το οποίο εξασφαλίζει την επιτυχή λήψη της εντολής. Επιπλέον έχει υλοποιηθεί η βοηθητική υπολειτουργία «keep_rolling», η οποία βελτιώνει τον τρόπο κίνησης του οχήματος, όταν το σήμα του πομπού είναι εξασθενημένο. Ουσιαστικά ορίζει ένα κατώφλι, το οποίο για περιορισμένο αριθμό συνεχόμενων τιμών «0000» της «sensors» επιτρέπει την κίνηση του οχήματος, αντί να το ακινητοποιήσει. Προφανώς υπάρχει δυνατότητα ενεργοποίησης/απενεργοποίησης της συγκεκριμένης υπολειτουργίας.

L IR	R IR	FL IR	FR IR	Ενέργεια
0	0	0	0	Σταμάτημα λόγω έλλειψης σήματος
0	0	0	1	Κίνηση προς τα εμπρός
0	0	1	0	Κίνηση προς τα εμπρός
0	0	1	1	Κίνηση προς τα εμπρός
0	1	0	0	Στροφή δεξιά
0	1	0	1	Στροφή δεξιά
0	1	1	0	Στροφή δεξιά
0	1	1	1	Στροφή δεξιά
1	0	0	0	Στροφή αριστερά
1	0	0	1	Στροφή αριστερά
1	0	1	0	Στροφή αριστερά
1	0	1	1	Στροφή αριστερά
1	1	0	0	Σταμάτημα λόγω μη αποδεκτής τιμής
1	1	0	1	Σταμάτημα λόγω μη αποδεκτής τιμής
1	1	1	0	Σταμάτημα λόγω μη αποδεκτής τιμής
1	1	1	1	Σταμάτημα λόγω μη αποδεκτής τιμής

Πίνακας 11: Τιμές & ενέργειες για την «sensors»

Αναλυτικότερα η «follow me», μετά την αρχικοποίηση ορισμένων μεταβλητών, εισέρχεται σε έναν ατέρμονο βρόγχο, απ' τον οποίον θα εξέλθει μόνο με το πάτημα του κόκκινου κουμπιού. Η πρώτη ενέργεια είναι η σειριακή ανάγνωση του Arduino IR αισθητήρων και η ενημέρωση της μεταβλητής «sensors». Στην συνέχεια γίνεται έλεγχος της τιμής έντασης του Bluetooth «rssi». Στην περίπτωση που η τιμή δηλώνει ότι ο πομπός είναι κοντά, το όχημα ακινητοποιείται. Σε αντίθετη περίπτωση ελέγχεται αν η τιμή της «sensors» είναι «0». Εάν είναι, τότε είτε εκτελείται η «keep_rolling», είτε σημειώνονται οι εντολές ακινητοποίησης στις

ανάλογες μεταβλητές. Πρέπει να τονιστεί το γεγονός ότι σε αυτόν τον έλεγχο, και στους επόμενους που ακολουθούν, δεν εκτελούνται οι ενέργειες απόφασης αλλά αποθηκεύονται σε αντίστοιχες μεταβλητές («send_command1», «send_command2») για την μετέπειτα εκτέλεσή τους. Στην περίπτωση διαφορετικής τιμής από την «0», γίνονται οι έλεγχοι που εμφανίζονται στον **Πίνακας 11** και αναλόγως αποθηκεύονται οι αντίστοιχες εντολές. Τέλος γίνεται έλεγχος για εμπόδια στο μπροστινό μέρος του οχήματος (χρήση της «sonar_sens»). Αν υπάρχουν εκτελείται η λειτουργία αποφυγής εμποδίων, που αναλύεται στην επόμενη υποενότητα, ενώ σε αντίθετη περίπτωση εκτελούνται οι εντολές που είναι αποθηκευμένες στις μεταβλητές «send_command1» και «send_command2». Η λειτουργία της «follow me» παρουσιάζεται και στην **Εικόνα 46**.

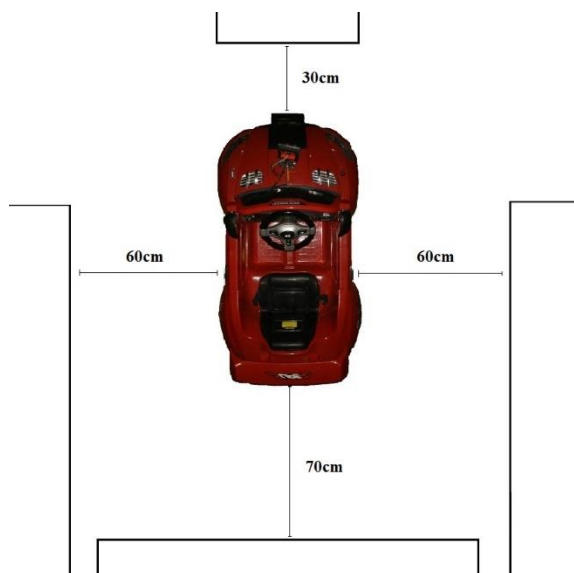


Εικόνα 46: Λειτουργία της Follow me

4.3.4 Λειτουργία αποφυγής εμποδίων

Η κλήση της λειτουργίας αποφυγής εμποδίων γίνεται μόνο από την λειτουργία «Follow me», που εκτελεί την αυτόνομη κίνηση του οχήματος. Η υλοποίηση της γίνεται μέσω της συνάρτησης «avoid_fun», η οποία προσφέρει την δυνατότητα αποφυγής μικρών και μεγαλύτερων εμποδίων. Συγκεκριμένα, μικρά θεωρούνται τα εμπόδια πλάτους μέχρι 60cm, ενώ μεγάλα αυτά που φτάνουν μέχρι 120cm. Με την διαδοχική κλήση της, το όχημα κινείται προς μία μόνο κατεύθυνση με σκοπό να αποφύγει μεγαλύτερα εμπόδια.

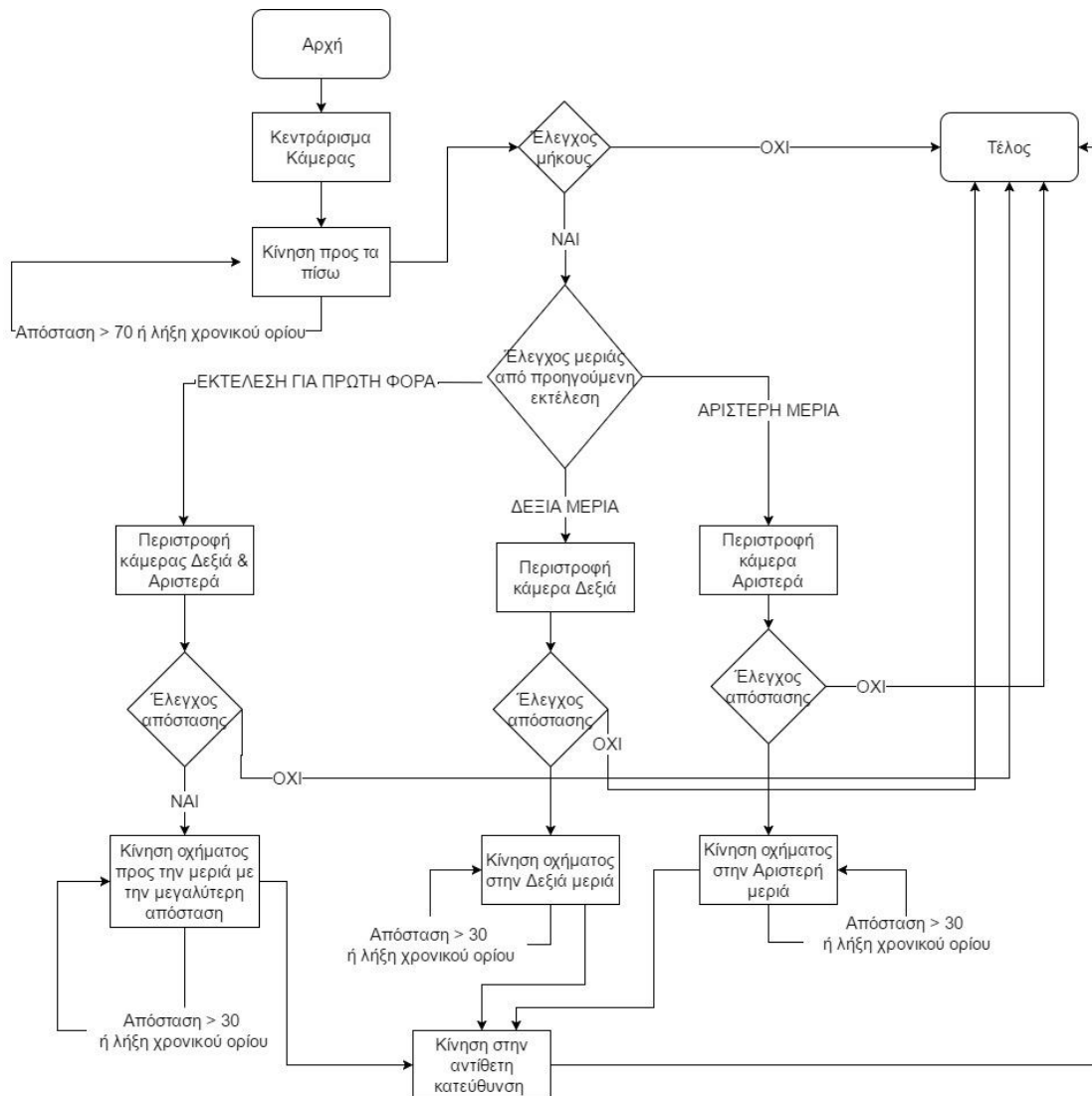
Αναλυτικότερα η συνάρτηση ξεκινάει με την επαναφορά της κάμερας στο κέντρο, και της κίνησης του οχήματος προς τα πίσω μέχρι την επίτευξη ασφαλούς απόστασης από το εμπόδιο. Σε περίπτωση λήψης του κωδικού «11», που μεταφράζεται σε μικρότερο εμπόδιο, η απόσταση αυτή υπολογίζεται όχι μέσω των αισθητήρων αλλά με χρονικό περιορισμό. Αυτά τα εμπόδια, σύμφωνα με τις συμβάσεις που έγιναν (**Εικόνα 14**) ανήκουν στην ομάδα των εμποδίων που δεν τα αντιλαμβάνονται οι αισθητήρες και το όχημα συγκρούεται. Παρόλα αυτά έγινε μια προσπάθεια υποτυπώδους αποφυγής τους. Μόλις εξασφαλιστεί η απαιτούμενη απόσταση το όχημα ακινητοποιείται. Στην συνέχεια ελέγχεται αν εξασφαλίζεται το απαιτούμενο μήκος (κενό μπρος/πίσω) (**Εικόνα 47**).



Εικόνα 47: Ελάχιστες αποστάσεις για την αποφυγή εμποδίων

Σε περίπτωση που δεν ικανοποιούνται οι ανάλογες αποστάσεις, η συνάρτηση τερματίζει. Διαφορετικά ο αλγόριθμος συνεχίζει με τον έλεγχο των αποστάσεων εκατέρωθεν του οχήματος. Για τον σκοπό αυτό η κάμερα περιστρέφεται μία φορά στις 180° (πλήρως αριστερά) αποθηκεύοντας την τιμή της απόστασης από τα αριστερά. Η ίδια διαδικασία επαναλαμβάνεται και για πλήρως δεξιά (0°). Στην συνέχεια η κάμερα κεντράρεται πάλι και ακολουθεί ο έλεγχος των τιμών που λήφθηκαν από αυτή την διαδικασία. Προφανώς αν και οι δύο τιμές είναι μικρότερες από την ελάχιστη προαπαιτούμενη, ο αλγόριθμος τερματίζει. Επιλέγεται η μεριά με την μεγαλύτερη απόσταση, ενώ στην σπάνια περίπτωση που είναι ίσες επιλέγεται η δεξιά, λόγω της κίνησης του οχήματος στον πίσω δεξιό τροχό. Με την επιλογή κατεύθυνσης ελιγμού η κάμερα περιστρέφεται στην αντίθετη μεριά και αρχίζει η κίνηση του οχήματος. Η κίνηση θα διακοπεί είτε λόγω εμφάνισης εμποδίου, είτε λόγω λήξης χρονικού ορίου. Τέλος η κάμερα κεντράρεται για τελευταία φορά, το τιμόνι περιστρέφεται από την αντίθετη κατεύθυνση και το όχημα κείται για λίγο ακόμα έτσι ώστε να περάσει ή να βρεθεί παράλληλα με το εμπόδιο. Σε αυτό το σημείο το εμπόδιο έχει αποφευχθεί με επιτυχία.

Εάν η κλήση της «avoid_fun» δεν είναι η πρώτη τότε το όχημα δεν ελέγχει και τις δύο μεριές του (δεξιά, αριστερά) αλλά μόνο εκείνη που είχε επιλέξει την προηγούμενη φορά. Η υπόλοιπη διαδικασία ακολουθεί κανονικά όπως περιγράφηκε προηγουμένως. Η λειτουργία του αλγόριθμου που αναλύθηκε φαίνεται στην **Εικόνα 48**.

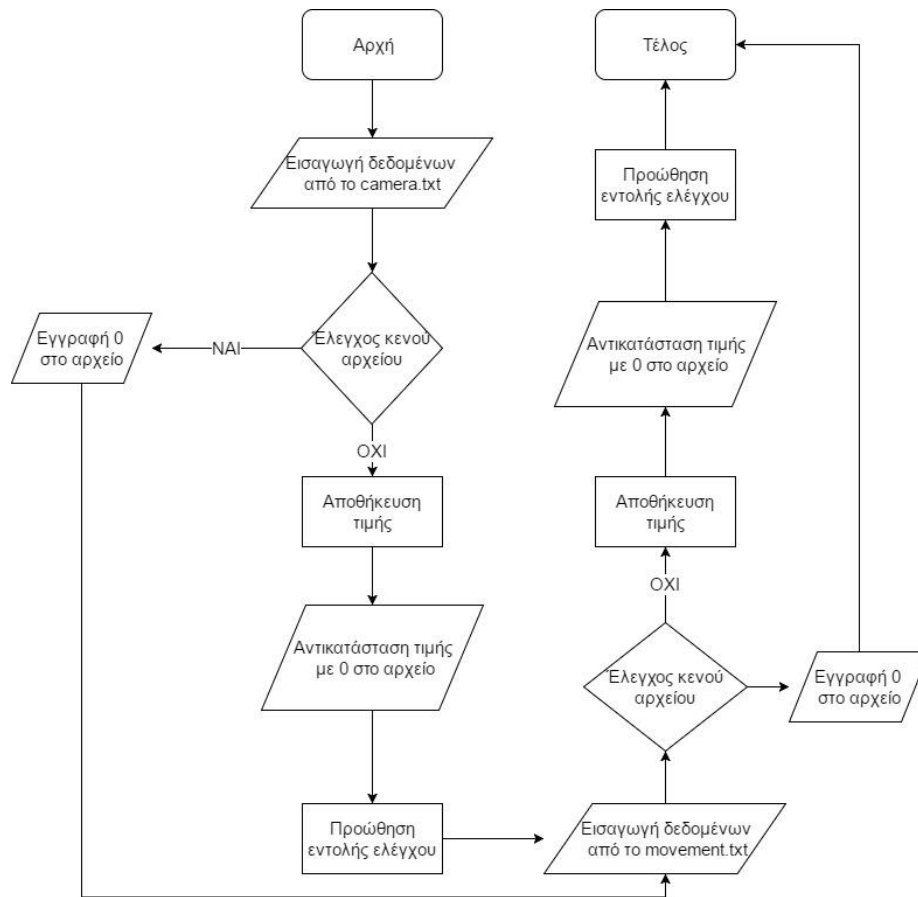


Εικόνα 48: Αλγόριθμος αποφυγής εμποδίων

4.3.5 Λειτουργία ελέγχου οχήματος

Εκτός από την λειτουργία «Follow me » που εμπεριέχει την αποφυγή εμποδίων, το κύριο πρόγραμμα παρέχει και την δυνατότητα ελέγχου του οχήματος. Ουσιαστικά πρόκειται για την συνάρτηση «fwdcom», η οποία προωθεί τους κωδικούς χειρισμού (**Πίνακας 7**) από τα προσωρινά αρχεία (camera.txt, movement.txt) στο Arduino ελέγχου. Εδώ πρέπει να πούμε ότι η υλοποίηση θα μπορούσε να πραγματοποιηθεί με ένα αρχείο, αλλά για λόγους ψευδοπαράλληλίας και καλύτερης απόκρισης του συστήματος, χρησιμοποιήθηκαν δύο. Το «camera.txt» περιλαμβάνει κωδικούς ελέγχου της κάμερας, ενώ το «movement.txt» αφορά την κίνηση του οχήματος. Σε κάθε κλήση της συνάρτησης «fwdcom» γίνεται ανάγνωση του αρχείου «camera.txt». Αν το αρχείο είναι κενό, εγγράφεται σε αυτό η τιμή «0» και ο αλγόριθμος

συνεχίζει στον επόμενο έλεγχο. Στην περίπτωση που διαβαστεί τιμή, αποθηκεύεται σε μια μεταβλητή, και στην συνέχεια αντικαθίσταται στο αρχείο με την τιμή «0». Τέλος ο κωδικός που είναι αποθηκευμένος στην μεταβλητή, προωθείται μέσω σειριακής τύπωσης στο Arduino ελέγχου. Η σειριακή τύπωση σε αυτό το σημείο γίνεται χωρίς την χρήση μηνύματος επιβεβαίωσης, μιας και ο χρήστης μπορεί να καταλάβει αν εκτελέστηκε ή όχι η εντολή που έστειλε. Η ίδια διαδικασία επαναλαμβάνεται για το αρχείο «movement.txt». Η λειτουργία της «fwdcom» παρουσιάζεται στην **Εικόνα 49** ενώ ο τρόπος εγγραφής των προσωρινών αρχείων ελέγχου (camera.txt, movement.txt) γίνεται από τον χρήστη μέσω SSH, όπως αναλύεται στην ενότητα που ακολουθεί.



Εικόνα 49: Λειτουργία της «fwdcom»

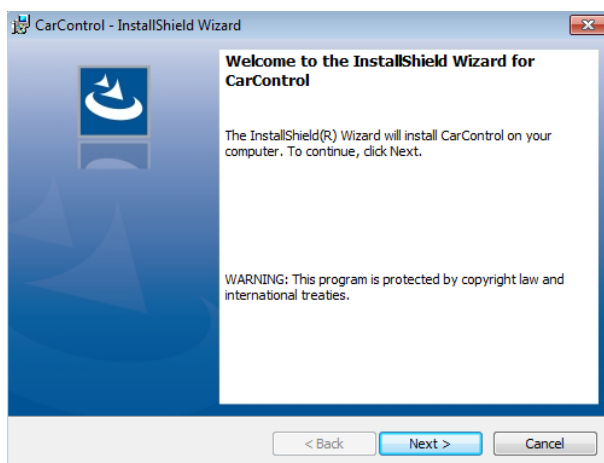
4.4 Λογισμικό στα Windows – εφαρμογή χρήστη

Για τον έλεγχο του οχήματος, εκτός από το λογισμικό – εξυπηρέτη που είδαμε στην προηγούμενη ενότητα, υλοποιήθηκε και λογισμικό - πελάτη που εκτελείται σε Windows. Σε αυτήν την ενότητα παρουσιάζεται ο τρόπος χρήσης της εφαρμογής αυτής και αναλύεται η λειτουργία της. Όπως είδαμε, η λειτουργία ελέγχου του λογισμικού στο Raspberry Pi προωθεί τις εντολές από τα προσωρινά αρχεία στο κατάλληλο Arduino Uno. Σκοπός της εφαρμογής - χρήστη είναι η εγγραφή των εντολών αυτών στα προσωρινά αρχεία.

Συγκεκριμένα υλοποιήθηκε ένα εκτελέσιμο αρχείο με όνομα CarControl.exe. Η σωστή εκτέλεση του CarControl.exe προϋποθέτει την ύπαρξη συγκεκριμένων αρχείων βιβλιοθήκης δυναμικής σύνδεσης (dll) και ειδικών αναδιανεμητέων Visual C++ πακέτων. Γι' αυτόν τον σκοπό δημιουργήθηκε ένας εγκαταστάτης (installer) της εφαρμογής. Ο εγκαταστάτης ελέγχει την διαθεσιμότητα των αναδιανεμητέων Visual C++ πακέτων και στην περίπτωση έλλειψής τους, τα εγκαθιστά. Επιπλέον τοποθετεί το εκτελέσιμο στον φάκελο επιλογής του χρήστη και εξάγει στο ίδιο σημείο τα dll.

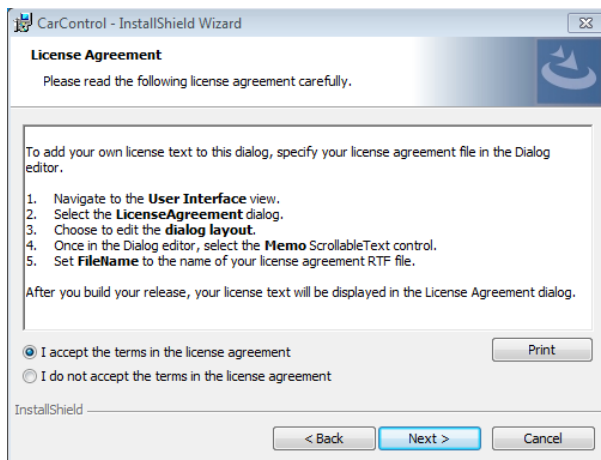
4.4.1 Εγκατάσταση του CarControl

Για την εγκατάσταση αρκεί να εκτελέσουμε το αρχείο CarControl_32_setup.exe. Με την εκτέλεσή του ανοίγει ο οδηγός εγκατάστασης (**Εικόνα 50**).



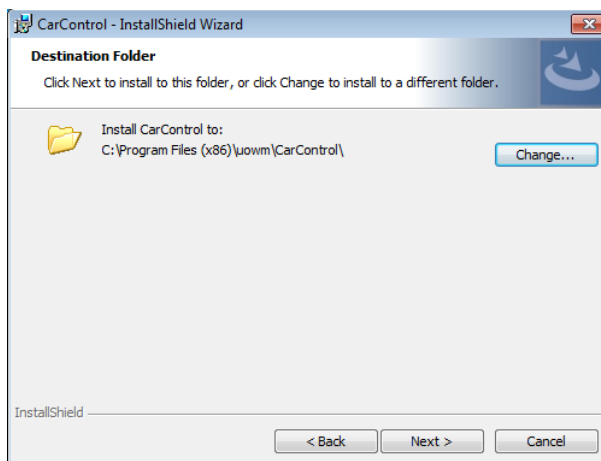
Εικόνα 50: Αρχική οθόνη εγκαταστάτη του CarControl

Με την επιλογή «Next» προχωράμε στην επόμενη οθόνη του εγκαταστάτη, που μας ενημερώνει για τα δικαιώματα χρήσης του συγκεκριμένου λογισμικού (**Εικόνα 51**).



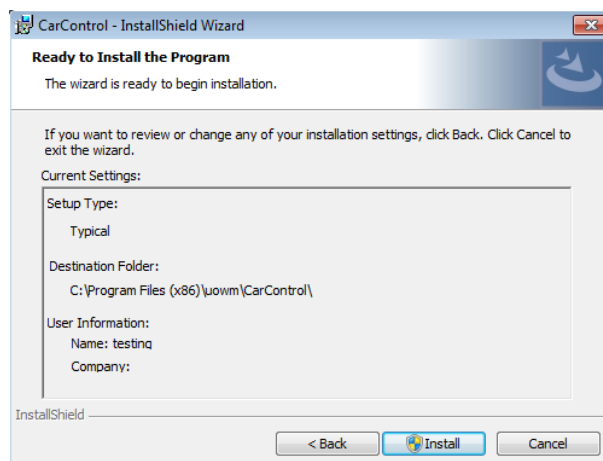
Εικόνα 51: Οθόνη του εγκαταστάτη συμφωνίας άδειας

Έπειτα εμφανίζεται η οθόνη τοποθεσίας προορισμού, στην οποία θα εγκατασταθεί η εφαρμογή. Ταυτόχρονα δίνεται η δυνατότητα στον χρήστη να επιλέξει την τοποθεσία που επιθυμεί (**Εικόνα 52**).



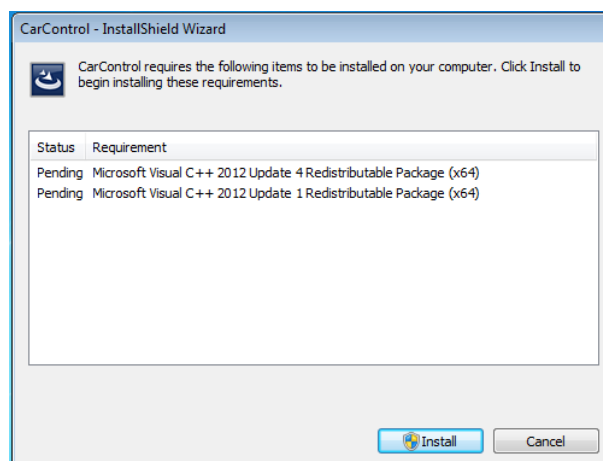
Εικόνα 52: Οθόνη τοποθεσίας εγκατάστασης

Στη συνέχεια εμφανίζεται η οθόνη επιβεβαίωσης, στην οποία εμφανίζονται όλες οι πληροφορίες σχετικά με την εγκατάσταση (**Εικόνα 53**).



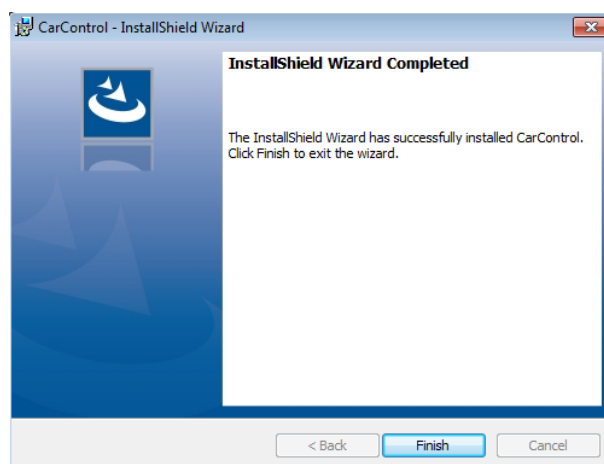
Εικόνα 53: Οθόνη επιβεβαίωσης εγκατάστασης

Σε αυτό το σημείο γίνεται ο έλεγχος ύπαρξης των αναδιανεμητέων Visual C++ πακέτων. Σε περίπτωση που κάποιο από τα απαιτούμενα λείπει, εμφανίζεται η αντίστοιχη οθόνη και με επιλογή του χρήστη, το σύστημα προχωράει στην εγκατάστασή τους (**Εικόνα 54**).



Εικόνα 54: Εγκατάσταση Visual C++ πακέτων

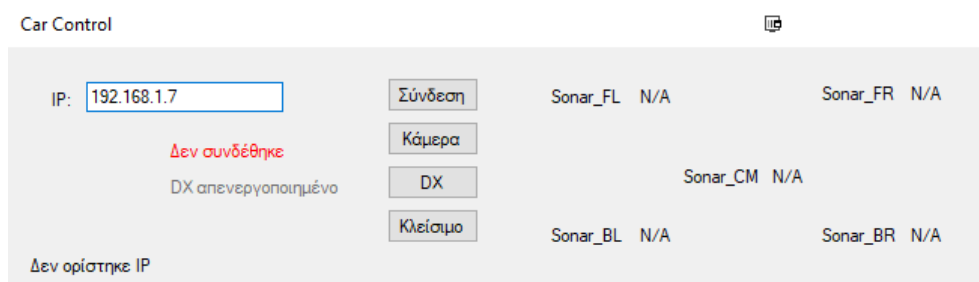
Αμέσως μετά γίνεται η εγκατάσταση του CarControl. Τέλος μόλις ολοκληρωθεί η εγκατάσταση εμφανίζεται η οθόνη ολοκλήρωσης με επιτυχία. Από αυτό το σημείο, το CarControl είναι έτοιμο για χρήση (**Εικόνα 55**).



Εικόνα 55: Οθόνη επιτυχίας εγκατάστασης

4.4.2 Χρήση & λειτουργία του CarControl

Απαραίτητη προϋπόθεση για τη χρήση του CarControl είναι η ύπαρξη χειριστηρίου (Gamepad/Joyrapd) στον Η/Υ. Το CarControl στην παρούσα υλοποίηση δεν χρησιμοποιεί άλλες μεθόδους εισαγωγής, όπως ποντίκι και πληκτρολόγιο, για τον έλεγχο του οχήματος. Με την εκτέλεση του CarControl.exe μας ανοίγει η κύρια οθόνη της εφαρμογής (Εικόνα 56).

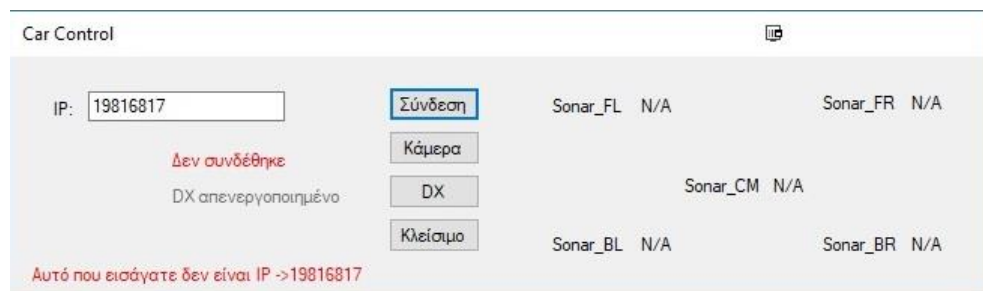


Εικόνα 56: Αρχική οθόνη του CarControl

Όπως φαίνεται πάνω αριστερά βρίσκεται το πλαίσιο, στο οποίο μπορούμε να εισάγουμε την IP του οχήματος. Ακριβώς από κάτω είναι η κατάσταση, της σύνδεσης και του DirectX. Κάτω αριστερά παρατηρείται το πλαίσιο μηνυμάτων σχετικά με την IP που εισάγει ο χρήστης. Η μεσαία στήλη αποτελείται από κουμπιά, τα οποία παρέχουν κάποιες δυνατότητες. Συγκεκριμένα από πάνω προς τα κάτω έχουμε το κουμπί «Σύνδεση», το οποίο εκτελεί SSH

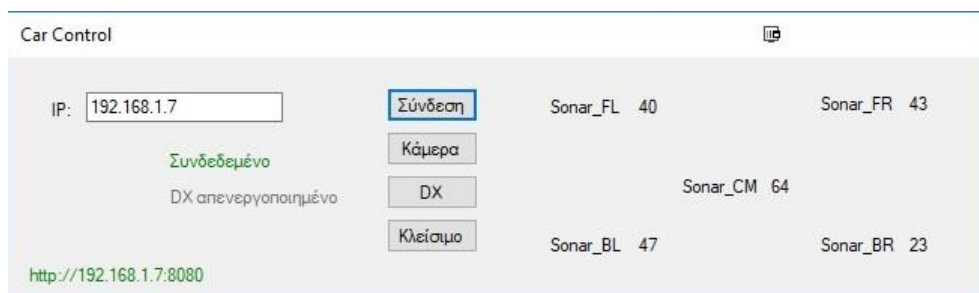
σύνδεση στην IP που βρίσκεται στο ανάλογο πλαίσιο. Στην συνέχεια το κουμπί «Κάμερα» ενεργοποιεί την ροή δεδομένων (stream) από την κάμερα του οχήματος. Για την εκτέλεση αυτής της ενέργειας απαιτείται ύπαρξη SSH σύνδεσης με το όχημα. Ακριβώς από κάτω υπάρχει το κουμπί «DX» το οποίο είναι υπεύθυνο για την ενεργοποίηση/απενεργοποίηση της ανάγνωσης χειριστηρίου με την χρήση DirectX Input. Τέλος έχει τοποθετηθεί το κουμπί «Κλείσιμο» το οποίο επιτρέπει τον σωστό τερματισμό της εφαρμογής, στέλνοντας το ανάλογο μήνυμα σε όλες τις διεργασίες της εφαρμογής. Στην δεξιά μεριά του παραθύρου είναι τοποθετημένα τα πεδία που αντιστοιχούν στους αισθητήρες απόστασης του οχήματος. Τα πεδία αυτά έχουν την ίδια διάταξη με τους αισθητήρες στο όχημα για καλύτερη και ευκολότερη κατανόηση. Αναλυτικά οι αισθητήρες είναι: ο μπροστά αριστερά (Sonar_FL), ο μπροστά δεξιά (Sonar_FR), ο αισθητήρας κάμερας (Sonar_CM), ο πίσω αριστερά (Sonar_BL) και ο πίσω δεξιά (Sonar_BR).

Αφού εισάγουμε την IP, με την επιλογή «Σύνδεση» εγκαθιδρύουμε SSH σύνδεση με το όχημα. Σε περίπτωση που δεν εισάχθηκε IP το σύστημα εμφανίζει μήνυμα λάθους (**Εικόνα 57**).



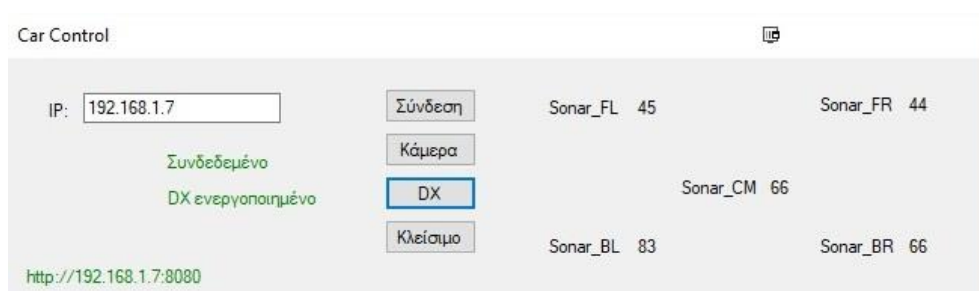
Εικόνα 57: Μήνυμα λάθους στο CarControl

Μόλις επιτευχθεί η σύνδεση είμαστε σε θέση να χειριστούμε το όχημα και να ενεργοποιήσουμε την ροή δεδομένων της κάμερας του οχήματος (**Εικόνα 58**).



Εικόνα 58: CarControl - επιτυχής σύνδεση

Σε οποιοδήποτε κατάσταση της εφαρμογής μπορούμε να ενεργοποιήσουμε την υποστήριξη DX, ανάλογα με τις δυνατότητες του χειριστηρίου που χρησιμοποιούμε (**Εικόνα 59**).



Εικόνα 59: CarControl - ενεργοποίηση DX

Οι ενέργειες των κουμπιών του χειριστηρίου δίνονται αναλυτικά στον **Πίνακας 12**. Η ορολογία των κουμπιών που χρησιμοποιείται αντιστοιχεί στο χειριστήριο Xbox360 και συγγενικών του. Επιπλέον για καλύτερη κατανόηση παρουσιάζεται και η διάταξή τους στην **Εικόνα 60**.

Κουμπιά		Ενέργεια	Εντολή
Rt	Δεξιά σκανδάλη	Κίνηση μπροστά	ms
Lt	Αριστερή σκανδάλη	Κίνηση πίσω	mb
Ls X-	Αριστερός μοχλός αριστερά	Στροφή αριστερά	ml
Ls X+	Αριστερός μοχλός δεξιά	Στροφή δεξιά	mr
Rs	Πάτημα δεξιού μοχλού	Κεντράρισμα κάμερας	cc
Rs X-	Δεξιός μοχλός αριστερά	Κίνηση κάμερας αριστερά	cl
Rs X+	Δεξιός μοχλός δεξιά	Κίνηση κάμερας δεξιά	cr
Rs Y-	Δεξιός μοχλός κάτω	Κίνηση κάμερας κάτω	cd
Rs Y+	Δεξιός μοχλός πάνω	Κίνηση κάμερας πάνω	cu

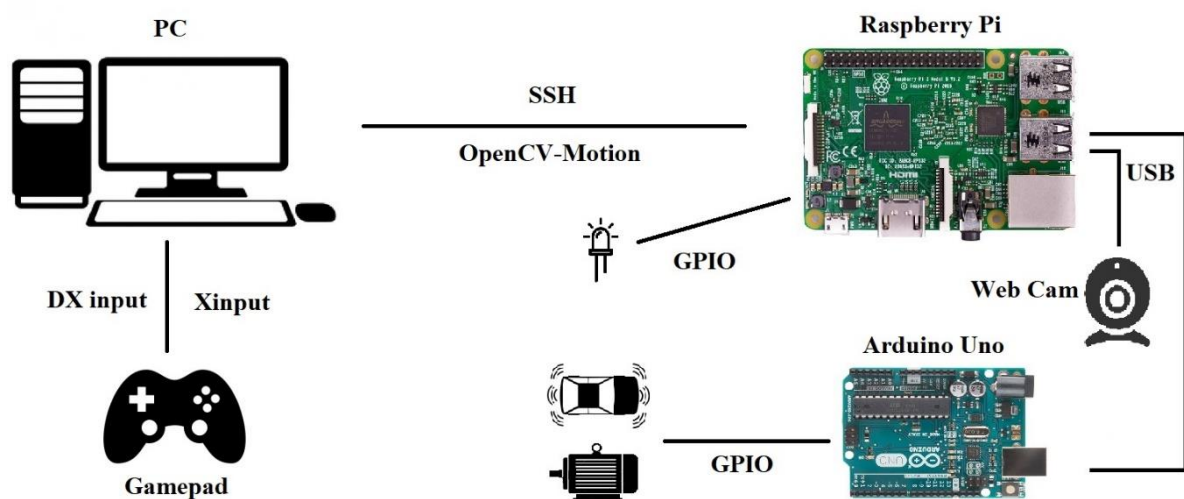
Y	Κουμπί Y	Ενεργοποίηση/απενεργοποίηση ροής δεδομένων από την κάμερα	-
B	Κουμπί B	Άναμμα/σβήσιμο μπλε φώτων	gpio write 0 1 / gpio write 0 0
A	Κουμπί A	Άναμμα/σβήσιμο λευκών φώτων	gpio write 1 1 / gpio write 1 0

Πίνακας 12: Ενέργειες κουμπιών χειριστηρίου



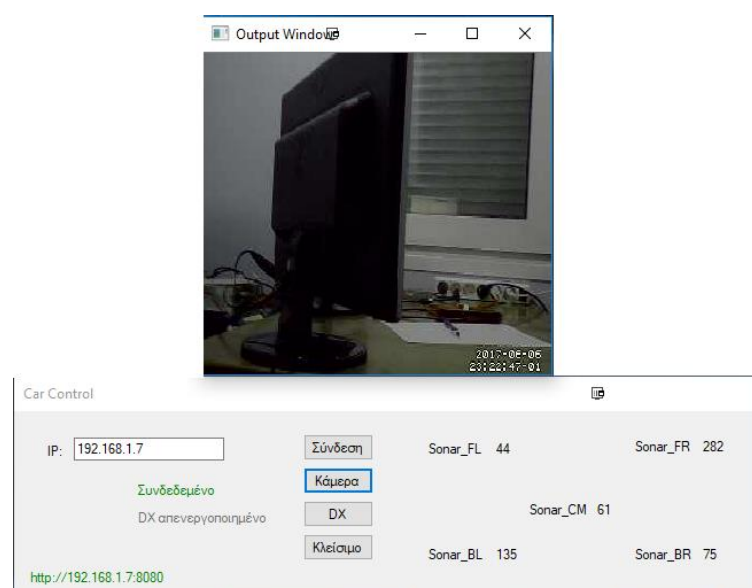
Εικόνα 60: Διάταξη χειριστηρίου

Εφόσον παρουσιάστηκε η χρήση του CarControl, ήρθε η στιγμή να αναλυθεί η λειτουργία του (**Εικόνα 61**). Σε γενικές γραμμές το σύστημα, μετά από επιτυχή SSH σύνδεση με το όχημα, διαβάζει περιοδικά την είσοδο του χειριστηρίου. Ανάλογα με την είσοδο, το CarControl θα γράψει την αντίστοιχη εντολή στα προσωρινά αρχεία του Raspberry Pi. Εξαίρεση αποτελούν οι εντολές χειρισμού των φώτων οι οποίες εκτελούνται απευθείας στο Raspberry Pi. Τόσο η εγγραφή των προσωρινών αρχείων, όσο και ο έλεγχος των φώτων, υλοποιούνται με εκτέλεση SSH εντολών.



Εικόνα 61: Συνολική λειτουργία ελέγχου οχήματος

Στην περίπτωση των προσωρινών αρχείων, η λειτουργία ελέγχου οχήματος προωθεί την εντολή στο κατάλληλο Arduino, όπου την εκτελεί (**4.3.5 Λειτουργία ελέγχου οχήματος**). Επιπλέον η CarControl με είσοδο, είτε απ' το χειριστήριο, είτε απ' το γραφικό περιβάλλον, μπορεί να εμφανίσει την ροή δεδομένων από την κάμερα του οχήματος σε ένα ξεχωριστό παράθυρο της εφαρμογής (**Εικόνα 62**). Τέλος υπάρχει μια διεργασία της CarControl, η οποία περιοδικά διαβάζει τις τιμές του αρχείου «sonar.txt» μέσω SSH. Με αυτόν τον τρόπο, η CarControl λαμβάνει τις τιμές των αισθητήρων και τις εμφανίζει στα αντίστοιχα πλαίσια.



Εικόνα 62: Ροή δεδομένων στην CarControl

Η συνολική υλοποίηση του CarControl περιλαμβάνει 6 αρχεία κώδικα σε C++. Το αρχικό αρχείο ονομάζεται Control Panel.cpp και έχει παραχθεί αυτόματα από τον σχεδιαστή γραφικού περιβάλλοντος του Visual Studio. Ουσιαστικά είναι το αρχείο που προετοιμάζει και ξεκινά την εφαρμογή με το γραφικό της περιβάλλον. Τα υπόλοιπα αρχεία φιλοξενούν ομαδοποιημένες συναρτήσεις και λειτουργίες του συστήματος. Το κύριο αρχείο είναι το main_gui.h, το οποίο αρχικοποιεί την εμφάνιση του γραφικού περιβάλλοντος. Στην συνέχεια ορίζει τις ενέργειες που εκτελεί το κάθε κουμπί, που είναι διαθέσιμο στο γραφικό περιβάλλον. Ένα μεγάλο μέρος του, όπως αυτό που περιγράφει την εμφάνιση του γραφικού περιβάλλοντος, είναι αυτόματα παραγόμενο από τον σχεδιαστή του Visual Studio. Τέλος στο παρόν λογισμικό χρησιμοποιήθηκε πολυνηματικός (multithreading) προγραμματισμός για την ταυτόχρονη εκτέλεση πολλών λειτουργιών. Παράλληλα, η πρόσβαση σε κοινόχρηστους πόρους του συστήματος γίνεται με την χρήση κλειδαριών (locks), για την αντιμετώπιση προβλημάτων ανταγωνισμού μεταξύ των διεργασιών.

4.4.3 Μεταγλώττιση της εφαρμογής

Για την δημιουργία του έργου (project) στο Visual Studio ακολουθήθηκε η διαδικασία: File -> New Project -> Visual C++ -> CLR -> CLR Empty Project. Στη συνέχεια δόθηκε το όνομα και η τοποθεσία του Project. Αυτό έχει ως αποτέλεσμα την δημιουργία ενός κενού project. Έπειτα με δεξί κλικ στο Project επιλέγουμε add -> new item -> Visual C++ -> UI -> Windows Forms. Με αυτόν τον τρόπο ανοίγει ο σχεδιαστής του γραφικού περιβάλλοντος της εφαρμογής. Επιπλέον με δεξί κλικ στην λύση (Solution) επιλέγουμε add -> existing project και προσθέτουμε το project του NvGamepad, που παρέχει η Nvidia για χρήση της Xinput. Με δεξί κλικ στο Project μας επιλέγουμε options (επιλογές). Σε αυτό το σημείο πρέπει να αλλάξουμε τις ιδιότητες του project και να εισάγουμε τις εξωτερικές βιβλιοθήκες που χρησιμοποιούμε. Συγκεκριμένα στο Configuration Properties -> General κάνουμε τις αλλαγές:

- Platform Toolset: Visual Studio 2015 (v140)
- Enable Managed Incremental Build: Yes
- Configuration Type: Application (.exe)
- Character Set: Use Multi-Byte Character Set
- Common Language Runtime Support: Common Language Runtime Support (/clr)

Στο Configuration Properties -> VC++ Directories στο πεδίο include Directories προσθέτουμε την διαδρομή *PATH_to_OpenCV\install\x86\vc14\bin*.

Στο Configuration Properties -> C/C++ κάνουμε τις εξής αλλαγές:

- Additional Include Directories: προσθέτουμε τις διαδρομές για:
 - libssh-0.7.1\include
 - nvgamepad-win-1.0\samples\GamepadSample
 - nvgamepad-win-1.0\include
 - opencv\install\include\opencv2
 - opencv\install\include\opencv
 - opencv\install\include\
- Basic Runtime Checks: Default
- Debug Information Format: Program Database (/Zi)
- Optimization: Disabled
- Preprocessor Definitions:
 - WIN32
 - _DEBUG
- SDL checks: Yes (/sdl)
- Use Unicode For Assembler Listing: Level3 (/W3)

Τέλος στο Configuration Properties -> Linker αλλάζουμε:

- Additional Dependencies: προσθέτουμε τα:
 - opencv_core310.lib
 - opencv_highgui310.lib
 - opencv_video310.lib
 - opencv_videoio310.lib
 - opencv_videostab310.lib
 - xinput9_1_0.lib
 - ssh.lib
 - dinput8.lib
 - dxguid.lib
- Additional Library Directories: προσθέτουμε τις διαδρομές για:
 - libssh-0.7.1\lib

-
- Διαδρομή του παρόντος project στον φάκελο Debug
 - opencv\install\x86\vc14\lib
 - Enable Incremental Linking: Yes (/INCREMENTAL)
 - Entry Point: Main
 - Generate Debug Info: Optimize debugging (/DEBUG)
 - Ignore Import Library: Yes
 - SubSystem: Console (/SUBSYSTEM:CONSOLE) για αποσφαλμάτωση ή Windows (/SUBSYSTEM:WINDOWS) αν δεν θέλουμε κονσόλα με μηνύματα αποσφαλμάτωσης

Ανάλογα πράττουμε και για το project NvGamepad, εφαρμόζοντας τις παρακάτω ρυθμίσεις. Στο Configuration Properties -> General κάνουμε τις αλλαγές:

- Target Name: libNvGamepadD
- Target Extension: .lib
- Platform Toolset: Visual Studio 2015 (v140)
- Configuration Type: Static library (.lib)

Ενώ τέλος στο Configuration Properties -> C/C++ τα εξής:

- Additional Include Directories: προσθέτουμε τις διαδρομές για:
 - Nvgamepad-win-1.0\src
 - Nvgamepad-win-1.0\include
- Debug Information Format: Program Database (/Zi)
- Enable function-Level Linking: Yes (/Oi)
- Optimization: Disabled (/Od)
- Preprocessor Definitions: προσθέτουμε τα
 - WIN32
 - _WIN32
 - _LIB
 - _DEBUG
 - PROFILE
 - _ITERATOR_DEBUG_LEVEL=0
- Warning Level: Level3 (/W3)

Με την εισαγωγή των παραπάνω ρυθμίσεων, το project μπορεί να φιλοξενήσει τα αρχεία με τους κώδικες και να μεταγλωττιστεί με επιτυχία. Τέλος για την προσθήκη του οδηγού εγκατάστασης της εφαρμογής προσθέτουμε ένα νέο Project στην παρούσα λύση. Αυτό επιτυγχάνεται με δεξί κλικ στο Solution -> add -> new project -> Other Project Types -> Setup and Deployment -> InstallShield Limited Edition Project. Αυτόματα ανοίγει ο οδηγός προσαρμογής του εγκαταστάτη. Στο πλαίσιο Application Files προσθέτουμε τα:

- opencv_core310.dll
- opencv_core310d.dll
- opencv_highgui310.dll
- opencv_imgcodecs310.dll
- opencv_imgcodecs310d.dll
- opencv_imgproc310.dll
- opencv_imgproc310d.dll
- opencv_ml310.dll
- opencv_objdetect310.dll
- opencv_videoio310.dll
- opencv_videoio310d.dll
- ssh.dll
- zlib.dll

Έτσι ο οδηγός εγκατάστασης θα τα τοποθετήσει στον ίδιο φάκελο με το εκτελέσιμο CarControl. Η ύπαρξη αυτών των αρχείων, στον ίδιο φάκελο με το εκτελέσιμο, είναι απαραίτητα για τη σωστή λειτουργία της εφαρμογής.

Επιπλέον στο Specify Application Data -> Redistributables προσθέτουμε:

- Microsoft Visual C++ 2010 Redistributable Package (x86)
- Microsoft Visual C++ 2012 Redistributable Package (x86)
- Microsoft Visual C++ 2012 Redistributable Package (x64)
- Microsoft Visual C++ 2012 Update 1 Redistributable Package (x86)
- Microsoft Visual C++ 2012 Update 1 Redistributable Package (x64)
- Microsoft Visual C++ 2012 Update 4 Redistributable Package (x86)
- Microsoft Visual C++ 2012 Update 4 Redistributable Package (x64)

-
- Microsoft Visual C++ 2015 Redistributable Package (x86)
 - Microsoft Visual C++ 2015 Redistributable Package (x64)

Με αυτόν τον τρόπο, ο εγκαταστάτης θα ελέγξει την υπάρξη των πακέτων αυτών στο σύστημα. Σε περίπτωση απουσίας κάποιου από αυτά, ο οδηγός θα το εγκαταστήσει όπως είδαμε στην ενότητα **4.4.1 Εγκατάσταση του CarControl**. Με την ολοκλήρωση των απαραίτητων ρυθμίσεων για την επιτυχή μεταγλώττιση του έργου, είμαστε σε θέση να προχωρήσουμε στην ανάλυση των επιμέρους λειτουργιών της εφαρμογής.

4.4.4 Σύνδεση SSH

Η επίτευξη της σύνδεσης SSH με το όχημα επιτυγχάνεται με την επιλογή του κουμπιού «Σύνδεση». Συγκεκριμένα με την επιλογή αυτή, το σύστημα παίρνει ως είσοδο το περιεχόμενο του πεδίου IP και ελέγχει αν έχει έγκυρη τιμή. Στην περίπτωση που η τιμή δεν είναι IP, η ενέργεια του κουμπιού εμφανίζει ανάλογο μήνυμα και τερματίζει. Διαφορετικά προσαρμόζει την IP σε IP ροής δεδομένων (προσθέτοντας στο τέλος της, την πόρτα «8080») και προχωρά την καθίδρυση της σύνδεσης.

Ο έλεγχος εγκυρότητας της τιμής IP γίνεται από την συνάρτηση «check_input» που βρίσκεται στο αρχείο «ssh_func.h». Η συνάρτηση αυτή χωρίζει την τιμή στα σημεία που υπάρχει «.». Έπειτα ελέγχει αν κάθε τμήμα της χωρισμένης μεταβλητής είναι αριθμός μεταξύ 0 - 255. Σε περίπτωση έγκυρης τιμής, η συνάρτηση είναι αληθής (επιστρέφει «True»), διαφορετικά ψευδής (επιστρέφει «False»).

Η επίτευξη της SSH σύνδεσης γίνεται με χρήση της συνάρτησης «create_ssh_connection» που βρίσκεται επίσης στο αρχείο «ssh_func.h». Η συνάρτηση αυτή είναι της βιβλιοθήκης libssh 0.7.1 και ουσιαστικά με χρήση της IP, του ονόματος χρήστη και του κωδικού του, δημιουργεί μια συνεδρία SSH σύνδεσης με το όχημα (εξυπηρέτη). Τέλος επιστρέφει το αντικείμενο της συνεδρίας, το οποίο αποθηκεύεται στην μεταβλητή κλήσης της συνάρτησης.

Κατά την καθίδρυση της σύνδεσης, δημιουργούνται δύο συνεδρίες SSH, η «sonar_ssh_session» και η «my_ssh_session». Έτσι η κάθε συνεδρία μπορεί να εκτελεί τις δικές της εντολές, παράλληλα και ανεξάρτητα από την άλλη. Στην συνέχεια εμφανίζεται μήνυμα επιτυχίας στο αντίστοιχο πλαίσιο του γραφικού περιβάλλοντος. Έπειτα ξεκινάει η εκτέλεση της διεργασίας «backgroundWorker1», η οποία είναι υπεύθυνη για την εμφάνιση και

ενημέρωση των τιμών των αισθητήρων απόστασης στο γραφικό περιβάλλον. Αμέσως μετά, ανάλογα με το αν έχει ενεργοποιηθεί ή όχι το DirectX Input, δημιουργείται το ανάλογο νήμα ανάγνωσης χειριστηρίου. Η ενέργεια αυτή γίνεται με την εκτέλεση της συνάρτησης «XinputRead» ή της «DxinputRead» αντίστοιχα. Τέλος σε περίπτωση που το κουμπί «Σύνδεση» πατηθεί όσο υπάρχει ήδη σύνδεση, τότε εμφανίζεται ανάλογο μήνυμα στην οθόνη του γραφικού περιβάλλοντος.

4.4.5 Ανάγνωση χειριστηρίου

Για την ανάγνωση χειριστηρίου υλοποιήθηκαν δύο συναρτήσεις, μιας και χρησιμοποιούνται δύο διαφορετικές τεχνολογίες πάνω σε αυτόν τον τομέα. Κατά την σύνδεση (4.4.4 Σύνδεση SSH) δημιουργείται το νήμα «padthread», το οποίο εκτελεί μία από τις δύο αυτές συναρτήσεις. Και οι δύο συναρτήσεις είναι τύπου «void», δηλαδή δεν επιστρέφουν κάποια τιμή στο πέρας τους.

Η XinputRead χρησιμοποιείται για την ανάγνωση χειριστηρίου με χρήση της τεχνολογίας Xinput. Αρχικά δημιουργείται το αντικείμενο «sGamepad» τύπου «gamepadXInput». Ακολουθούν δύο εμφωλευμένοι ατέρμονοι βρόγχοι, εκ των οποίων ο εξωτερικός είναι κενός, με εξαίρεση τον έλεγχο της σημαίας τερματισμού. Ο εσωτερικός ενημερώνει την κατάσταση του αντικειμένου «sGamepad» και ελέγχει την τιμή της. Η τιμή αυτή προσδιορίζει την κατάσταση των κουμπιών του χειριστηρίου. Με ένα πλήθος ελέγχων καλύπτονται όλες οι ενέργειες ελέγχου που εκτελεί το σύστημα (Πίνακας 12). Στην περίπτωση εγγραφής κάποιας εντολής σε προσωρινό αρχείο του Raspberry Pi, χρησιμοποιείται η εκτέλεση της εντολής διεργασίας ηχούς (echo) μέσω SSH. Η εκτέλεση αυτή γίνεται με την χρήση της συνάρτησης «shell_session», που βρίσκεται στο αρχείο «ssh_func.h».

Συγκεκριμένα η συνάρτηση αυτή παίρνει δύο ορίσματα, το «session» και το «my_command». Το πρώτο είναι αντικείμενο τύπου «ssh_session» και αναφέρεται στην συνεδρία που θέλουμε να χρησιμοποιήσουμε. Το δεύτερο είναι αλφαριθμητικό και περιέχει την εντολή που θέλουμε να εκτελεστεί στο Raspberry Pi του οχήματος. Η «shell_session» δημιουργεί και ανοίγει ένα κανάλι επικοινωνίας. Στην συνέχεια εκτελεί την εκάστοτε εντολή, και διαβάζει το αποτέλεσμα που θα τυπώνονταν στο τερματικό. Τέλος κλείνει το κανάλι επικοινωνίας και επιστρέφει την μεταβλητή SSH_OK, που περιέχει την πληροφορία της κατάστασης της σύνδεσης. Εάν η παραπάνω διαδικασία, σε κάποιο σημείο της αποτύχει, εμφανίζεται (στην κονσόλα κατά την διαδικασία αποσφαλμάτωσης) ανάλογο μήνυμα και

κλίνει το κανάλι επικοινωνίας. Παραδείγματος χάρη για την εγγραφή «ms» στο αρχείο «movement.txt» χρησιμοποιείται η εντολή: *echo ms > /home/onram/movement.txt*. Ανάλογη είναι η εντολή για την εγγραφή τιμών στο αρχείο «camera.txt». Τέλος για το σβήσιμο των λευκών φώτων δίνεται η εντολή: *gpio write 1 0*.

Ιδιάζουσα περίπτωση αποτελεί η ενεργοποίηση/απενεργοποίηση της ροής δεδομένων από την κάμερα του οχήματος. Η ενέργεια αυτή εκτελείται με τη χρήση του κουμπιού «Υ», από το χειριστήριο ή του κουμπιού «Κάμερα», από το γραφικό περιβάλλον. Η ενέργεια αυτή απαιτεί τον έλεγχο μοναδικότητας της κλήσης, δηλαδή αν εκτελείται αυτή η ενέργεια για πρώτη φορά ή όχι. Στην περίπτωση που εκτελούμε ενεργοποίηση της ροής δεδομένων για πρώτη φορά, δημιουργείται το νήμα «streamThread», το οποίο εκτελεί την συνάρτηση «stream» του αρχείου «video_stream.h». Έπειτα δίνονται οι απαραίτητες τιμές στις μεταβλητές που χρησιμοποιεί το νήμα που προαναφέρθηκε. Η διαδικασία ολοκληρώνεται με την εκτέλεση του νήματος. Στην περίπτωση που έχει ήδη ενεργοποιηθεί μια φορά η ροή δεδομένων, ελέγχεται αν στην παρούσα στιγμή είναι ενεργή ή όχι. Με αλλαγή των κατάλληλων μεταβλητών - σημαίας μεταβάλλεται η κατάσταση της ροής δεδομένων από ενεργή, σε μη και αντιστρόφως. Ίδια συμπεριφορά με αυτή που αναλύθηκε, παρουσιάζει το κουμπί «Κάμερα» του γραφικού περιβάλλοντος.

Αναλυτικά η συνάρτηση «stream» είναι τύπου «void» και δημιουργεί το αντικείμενο «ncap» τύπου «videoCapture» (χρήση της βιβλιοθήκης OpenCV). Στην συνέχεια ανοίγει η ροή δεδομένων του «ncap» με χρήση της δοσμένης IP (IP με την προσθήκη της πόρτας). Στην περίπτωση αποτυχίας εμφανίζεται ανάλογο μήνυμα και τερματίζει η συνάρτηση. Διαφορετικά ακολουθούν δύο εμφωλευμένοι ατέρμονοι βρόγχοι, εκ των οποίων ο εξωτερικός ελέγχει τη σημαία τερματισμού. Ο εσωτερικός συλλέγει, κάθε φορά, ένα καρέ από την συσκευή ροής δεδομένων και το εμφανίζει στο παράθυρο πολυμέσων της εφαρμογής. Κατά την αλλαγή των σημαιών για παύση της ροής δεδομένων, η εκτέλεση μεταφέρεται στον εξωτερικό βρόγχο. Με ανάλογη αλλαγή στις τιμές των σημαιών, η εκτέλεση μπορεί, είτε να ξαναμπει στον εσωτερικό (συνέχιση της ροής δεδομένων), ή να τερματίσει τελείως η συνάρτηση.

Για την ανάγνωση χειριστηρίου με DirectX χρησιμοποιείται η «DxinputRead», η οποία έχει παρόμοια συμπεριφορά με την «XinputRead». Η «DxinputRead» ξεκινάει με την δημιουργία ενός αντικειμένου συσκευής «DirectInput», ενώ ακολουθεί ο έλεγχος ύπαρξης τέτοιας συσκευής στον Η/Υ. Σε περίπτωση επιτυχίας ακολουθούν δύο εμφωλευμένοι ατέρμονοι βρόγχοι, εκ των οποίων ο εξωτερικός ελέγχει την ύπαρξη συσκευής «DirectInput»

και την τιμή της σημαίας τερματισμού. Ο εσωτερικός ανανεώνει και ελέγχει την τιμή της κατάστασης της συσκευής «DirectInput». Η υπόλοιπη διαδικασία είναι ίδια με αυτή της «XinputRead», όπως αναλύθηκε παραπάνω.

Η εναλλαγή, από τη χρήση της μίας τεχνολογίας ανάγνωσης χειριστηρίου, στην άλλη γίνεται με τη χρήση του κουμπιού «DX» του γραφικού περιβάλλοντος. Συγκεκριμένα με το πάτημα του «DX», ελέγχεται αρχικά εάν έχει επιτευχθεί σύνδεση ή όχι. Στην περίπτωση που δεν υπάρχει σύνδεση, η διαδικασία είναι απλή, εναλλάσσοντας τις τιμές των μεταβλητών - σημαίας για την επιλογή της συνάρτησης που θα εκτελεί το νήμα «padthread» κατά την δημιουργία του. Διαφορετικά, υπάρχει ήδη ένα νήμα «padthread», το οποίο εκτελεί, είτε την μία συνάρτηση, είτε την άλλη. Σε αυτή την περίπτωση με αλλαγή των κατάλληλων σημαιών, οδηγούμε το νήμα αυτό στην εκτέλεση του εξωτερικού βρόγχου της συνάρτησής του. Έτσι το νήμα αυτό είναι ουσιαστικά αδρανές. Στην συνέχεια δημιουργούμε ένα δεύτερο νήμα «padthread», το οποίο εκτελεί την άλλη συνάρτηση ανάγνωσης χειριστηρίου. Σε περίπτωση που υπάρχουν και τα δύο νήματα (αυτό που εκτελεί την «XinputRead» και αυτό που εκτελεί την «DxinputRead»), η εναλλαγή γίνεται με αντίστοιχη παύση-συνέχιση των νημάτων αυτών, μεταβάλλοντας τις τιμές των σημαιών τους. Τέλος, σε οποιαδήποτε περίπτωση, υπάρχει ενημέρωση και εναλλαγή του μηνύματος που εμφανίζεται στο γραφικό περιβάλλον της εφαρμογής.

4.4.6 Ανάγνωση τιμών αισθητήρων

Η εφαρμογή CarControl, εκτός από τον έλεγχο του οχήματος, προβάλλει τις τιμές των αισθητήρων απόστασης σε πραγματικό χρόνο στο γραφικό της περιβάλλον. Η λειτουργία αυτή ξεκινά την εκτέλεσή της μόλις επιτευχθεί η σύνδεση SSH. Αναλυτικά όπως αναφέρθηκε στο (4.4.4 Σύνδεση SSH) με την επίτευξη της σύνδεσης, ξεκινάει η εκτέλεση της διεργασίας «backgroundWorker1». Η «backgroundWorker1» ανήκει στην κλάση «BackgroundWorker» που εμπεριέχεται στο πλαίσιο .NET [53]. Η χρήση της «BackgroundWorker» κρίθηκε απαραίτητη, καθώς εστιάζει στην άμεση και σωστή ενημέρωση των τιμών στα πλαίσια του γραφικού περιβάλλοντος. Η λειτουργία της γίνεται με χρήση των δύο ενσωματωμένων συναρτήσεων «backgroundWorker1_DoWork» και «backgroundWorker1_ProgressChanged».

Η πρώτη εκτελεί περιοδικά τη συνάρτηση «sonaread» που βρίσκεται στο αρχείο «Sonar.h». Με αυτόν τον τρόπο παίρνει τις τιμές των αισθητήρων από το όχημα μέσω SSH. Στην συνέχεια οι τιμές αυτές τοποθετούνται σε έναν μονοδιάστατο πίνακα πέντε θέσεων. Τέλος

η κάθε τιμή του πίνακα προωθείται στην «backgroundWorker1_ProgressChanged». Η «backgroundWorker1_ProgressChanged» λαμβάνει την κάθε τιμή και, ανάλογα με την θέση της στον πίνακα, τοποθετείται στο αντίστοιχο πλαίσιο. Συγκεκριμένα από το πρώτο στοιχείο του πίνακα στο τελευταίο, οι τιμές αντιστοιχούν στους αισθητήρες: μπροστά δεξιός (FR), μπροστά αριστερός (FL), πίσω δεξιός (BR), πίσω αριστερός (BL) και κάμερας (CM).

Η συνάρτηση «sonaread» χρησιμοποιείται για την ανάγνωση του αρχείου «sonar.txt» μέσω SSH και επιστρέφει με τη μορφή διάνυσματος τις τιμές αυτές. Αρχικά εκτελεί μέσω SSH την εντολή συνένωσης αρχείων και εκτύπωση τους στην τυπική έξοδο (cat), ως εξής: *cat /home/onram/sonar.txt*. Για την εκτέλεσή της μέσω SSH χρησιμοποιείται η συνάρτηση «sonar_shell_session» του αρχείου «ssh_func.h». Η λειτουργία της «sonar_shell_session» είναι ίδια με αυτή της «shell_session» που είδαμε στην ενότητα **4.4.5 Ανάγνωση χειριστηρίου**. Σε αυτή την περίπτωση χρησιμοποιούμε τη δεύτερη συνεδρία SSH (ssh_session) που δημιουργήσαμε. Επιπλέον βασική διαφορά της «sonar_shell_session» με τη «shell_session» είναι ότι η πρώτη αντί να επιστρέφει την «SSH_OK», επιστρέφει το αλφαριθμητικό που τυπώνεται από την εκτέλεση της εντολής «*cat /home/onram/sonar.txt*». Το συγκεκριμένο αλφαριθμητικό αντιστοιχεί στις τιμές των αισθητήρων, που είναι καταχωρισμένες στο αρχείο «sonar.txt». Οι τιμές αυτές αποθηκεύονται με τη μορφή αλφαριθμητικού στη μεταβλητή «sonars_read». Στη συνέχεια οι τιμές της «sonars_read» μετατρέπονται σε αριθμούς και αποθηκεύονται στο διάνυσμα 5 θέσεων «sonar», το οποίο επιστρέφεται στο τέλος της συνάρτησης.

4.5 Σύνοψη Κεφαλαίου

Με την ανάλυση των λειτουργιών του CarControl οδηγηθήκαμε στο πέρας του τέταρτου κεφαλαίου. Συνοψίζοντας στο παρόν κεφάλαιο αναλύθηκε η συνολική λειτουργία του λογισμικού του συστήματος, καθώς και των επιμέρους υποσυστημάτων. Δόθηκαν διευκρινιστικές οδηγίες για την υλοποίηση ανάλογου έργου, καθώς και παραδείγματα χρήσης του παρόντος. Ακολουθεί το τελευταίο κεφάλαιο του συγγράμματος, το οποίο παρουσιάζει τα γενικά χαρακτηριστικά της όλης υλοποίησης. Παράλληλα αναλύονται τα πλεονεκτήματα και μειονεκτήματα της συγκεκριμένης υλοποίησης. Επιπλέον αναφέρονται τα σημαντικότερα προβλήματα που προέκυψαν και οι λύσεις που δόθηκαν. Τέλος δίνονται τα συμπεράσματα που προέκυψαν από την διεξαγωγή της παρούσας εργασίας, και αναφέρονται ιδέες για μεταγενέστερη εξέλιξη και αναβάθμιση του υπάρχοντος συστήματος.

Κεφάλαιο 5 – Σύνοψη συστήματος και συμπεράσματα

Ολοκληρώνοντας με την ανάλυση του λογισμικού μέρους, έχει ολοκληρωθεί η παρουσίαση του συστήματος. Εφόσον έχουν αναφερθεί όλες οι απαραίτητες πληροφορίες που πρέπει να γνωρίζει κάποιος, μπορούμε να μιλήσουμε για τα γενικά χαρακτηριστικά του συστήματος. Επιπλέον σε αυτό το κεφάλαιο παρουσιάζονται μετρήσεις σχετικά με το λογισμικό και υλικό μέρος του οχήματος. Ταυτόχρονα αναφέρονται τα πλεονεκτήματα και μειονεκτήματα της παρούσας υλοποίησης. Επιπρόσθετα παρουσιάζονται τα προβλήματα που προέκυψαν καθώς και οι λύσεις που δόθηκαν. Παράλληλα περιγράφονται οι περιορισμοί της παρούσας προσέγγισης, ενώ δίνονται προτάσεις για μελλοντικές επεκτάσεις. Τέλος αναφέρονται τα συμπεράσματα που προέκυψαν από τη συγκεκριμένη προσέγγιση.

5.1 Γενικά χαρακτηριστικά του συστήματος

Η λειτουργία ελέγχου του οχήματος είναι εφικτή, κατά την περίπτωση που το όχημα έχει πρόσβαση σε ένα δίκτυο υπολογιστών. Στο ίδιο δίκτυο πρέπει να βρίσκεται και ο Η/Υ του χρήστη. Σε αντίθετη περίπτωση, ο έλεγχος μπορεί να επιτευχθεί μέσω διαδικτύου με την προϋπόθεση, ότι τόσο ο χρήστης, όσο και το όχημα έχουν πρόσβαση σε αυτό. Επιπλέον για τον έλεγχο, πρέπει να υπάρχει από τη μεριά του χρήστη ένα χειριστήριο. Το σύστημα, λόγω χρήσης δύο τεχνολογιών ανάγνωσης χειριστηρίου, παρέχει μεγάλη υποστήριξη σε αυτό το τομέα.

Το όχημα με χρήση της διαδικτυακής κάμερας, που είναι εξοπλισμένο, παρέχει τη δυνατότητα παρακολούθησης μέσω αυτής. Για την παρακολούθηση μπορεί να χρησιμοποιηθεί, είτε η υλοποιημένη εφαρμογή του συστήματος (CarControl), είτε οποιοσδήποτε φυλλομετρητής. Παράλληλα στο CarControl παρέχονται μετρήσεις απ' όλους τους αισθητήρες απόστασης του οχήματος, δίνοντας μια πιο ολοκληρωμένη εικόνα για το χώρο που βρίσκεται το όχημα.

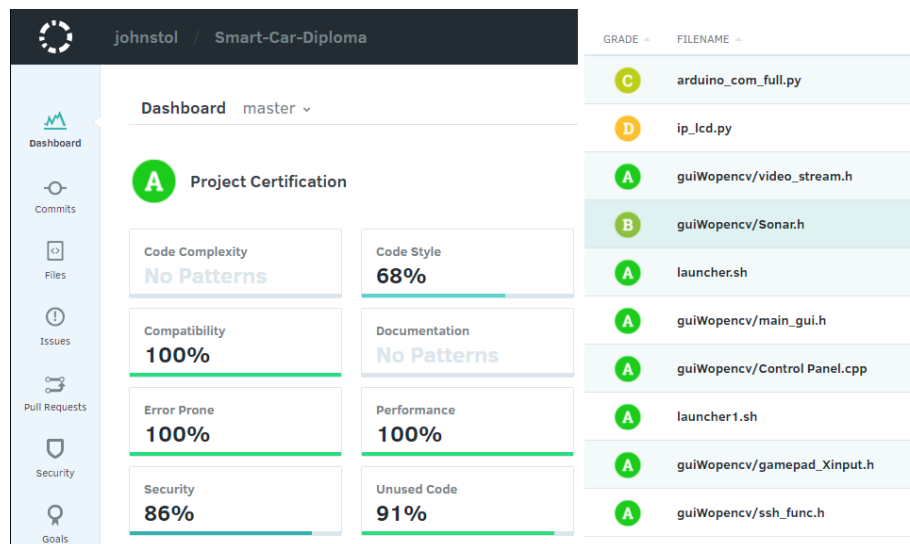
Οι αισθητήρες αυτοί είναι ικανοί για τον εντοπισμό εμποδίων σε απόσταση έως 2m. Το όχημα εκμεταλλευόμενο αυτή την πληροφορία, είναι σε θέση να ακινητοποιηθεί αυτόματα με σκοπό την αποτροπή σύγκρουσης. Επιπλέον διαθέτει ζεύγος αισθητήρων – πομπού, προσφέροντας τη δυνατότητα συνοδείας ατόμου. Κατά τη διάρκεια αυτής της διαδικασίας, το σύστημα είναι σε θέση να εντοπίσει εμπόδια, που βρίσκονται στην διαδρομή του και να τα αποφύγει.

5.1.1 Μετρήσεις συστήματος

Το σύστημα αποτελείται, τόσο από υλικό, όσο και από λογισμικό. Για την ορθότερη παρουσίαση των δυνατοτήτων του και της ποιότητας του, έγινε ένα σύνολο μετρήσεων. Οι μετρήσεις αυτές αφορούν τα δύο μέρη του συστήματος (υλικό, λογισμικό), τόσο μεμονωμένα, όσο και συνδυαστικά.

Συγκεκριμένα το όχημα, με το υλικό που χρησιμοποιήθηκε, κινείται με μέση ταχύτητα 0.6 m/s. Η ταχύτητα κίνησης επηρεάζεται άμεσα από ένα μεγάλο πλήθος παραγόντων, όπως το βάρος του οχήματος, η κλίση και η πρόσφυση της επιφάνειας κίνησης. Το όχημα κινείται ομαλά έχοντας φορτίο έως 15Kg, ενώ οριακή κίνηση επιτυγχάνεται στα 20Kg. Κατά τη διαδικασία συνοδείας, το όχημα μπορεί να ακολουθήσει το στόχο από απόσταση έως 2m. Τέλος το όχημα έχει αυτονομία κίνησης 90 λεπτά, και λειτουργίας 10 ώρες.

Το λογισμικό του συστήματος αποτελείται από 13 αρχεία πηγαίου κώδικα. Στα αρχεία αυτά γίνεται χρήση ποικίλων γλωσσών προγραμματισμού με τις παρακάτω αναλογίες: C++ 64.8%, Python 21.4%, Arduino (Wiring) 13.2%, shell 0.6%. Για την υλοποίηση ελέγχου του οχήματος από Windows, παράχθηκε το εκτελέσιμο CarControl.exe το οποίο έχει μέγεθος 261 KB. Για τη λειτουργία του εμπεριέχονται αρχεία βιβλιοθήκης δυναμικής σύνδεσης (dll), τα οποία οδηγούν το μέγεθος υλοποίησης στα 77,2MB. Για τη λειτουργία του CarControl απαιτούνται τα αναδιανεμητέα Visual C++ πακέτα, για τα οποία είναι υπεύθυνος ο υλοποιημένος εγκαταστάτης (**4.4.3 Μεταγλώττιση της εφαρμογής**). Το μέγεθος του εγκαταστάτη είναι 1.544KB. Κατά την εκτέλεση του CarControl καταναλώνονται 80MB μνήμης RAM, ενώ με τη χρήση της ροής δεδομένων καταναλώνονται έως και 150MB. Το μέγεθος αυτό μπορεί να αλλάξει ανάλογα με την ποιότητα του βίντεο. Το όλο λογισμικό αποτελείται από 2890 γραμμές κώδικα, 389 γραμμές σχολίων, 142 γραμμές κώδικα και σχολίων, ενώ περιλαμβάνει 22 συναρτήσεις. Τέλος το συνολικό λογισμικό έχει βαθμολογηθεί με A από την υπηρεσία βαθμολόγησης Codacy (**Εικόνα 63**).



Εικόνα 63: Βαθμολογία Codacy

Συνολικά το σύστημα, κατά τον εντοπισμό εμποδίου, ανταποκρίνεται στην ακινητοποίηση του οχήματος σε χρόνο μικρότερο του ενός δευτερολέπτου. Η ακινητοποίηση του οχήματος κατά την παρουσία υψηλής έντασης σήματος Bluetooth (άρα ο χρήστης βρίσκεται πολύ κοντά) είναι 0.5-1 s. Η ροή δεδομένων που λαμβάνει ο χρήστης από το όχημα μέσω δικτύου έχει καθυστέρηση 1-2 s, ενώ μεταβάλλεται ανάλογα με την ποιότητα της σύνδεσης. Τέλος η απόκριση του συστήματος κατά τον έλεγχο μέσω SSH είναι κάτω από 1s, η οποία επίσης εξαρτάται από την ποιότητα της σύνδεσης.

5.1.2 Ασφάλεια του συστήματος

Η παρούσα υλοποίηση περιλαμβάνει πλήθος μεθόδων προστασίας και ασφάλειας, τόσο του συστήματος, όσο και του γύρω περιβάλλοντος. Η ασφάλεια που εφαρμόστηκε είναι πολυεπίπεδη και έχει την έννοια, τόσο του αγγλικού όρου Security, όσο και του Safety. Επιπλέον έγιναν ενέργειες, έτσι ώστε να εξασφαλίζεται η προστασία του υλικού μέρους, αλλά και των ατόμων που έρχονται σε επαφή με αυτό. Επιπρόσθετα, λήφθηκαν τα απαραίτητα μέτρα, έτσι ώστε να μειωθεί όσο το δυνατόν περισσότερο η πιθανότητα πρόκλησης ατυχήματος. Με αυτόν τον τρόπο προστατεύεται, τόσο το υλικό του οχήματος, όσο και το περιβάλλον γύρω του. Παράλληλα χρησιμοποιήθηκαν τεχνικές, οι οποίες δεν επιτρέπουν τη χρήση του συστήματος από άτομα που δεν έχουν εξουσιοδότηση. Τέλος πρέπει να αναφερθεί, ότι όλα τα μέτρα ασφαλείας που υλοποιήθηκαν είναι στον προαπαιτούμενο βαθμό και σίγουρα υπάρχουν περιθώρια βελτίωσης.

Σημαντικό παράγοντα στην προστασία του υλικού συνέβαλε ο τρόπος εγκατάστασής του καθώς και η θέση του στο όχημα. Συγκεκριμένα το μεγαλύτερο μέρος του υλικού βρίσκεται στο μπροστινό μέρος του οχήματος, κάτω από το καπό. Με αυτόν τον τρόπο προστατεύεται από εξωτερικούς παράγοντες, όπως το να πέσει κάποιο ξένο αντικείμενο πάνω στον εξοπλισμό προκαλώντας τυχόν ζημιές. Επιπρόσθετα, η τοποθεσία του στη συγκεκριμένη περιοχή παρέχει περιορισμένη πρόσβαση σε φυσικά πρόσωπα, προστατεύοντας το από αυτά. Η λειτουργία αποτροπής πρόσκρουσης προσφέρει διπλή προστασία, τόσο από τη μεριά του περιβάλλοντος (να μη γίνει ζημιά στον χώρο χρήσης) , όσο και από αυτή του ίδιου του οχήματος (να προστατευτεί το αμάξωμα και γενικότερα το υλικό του). Σε περίπτωση πρόσκρουσης, έχει τοποθετηθεί ένας μαλακός προφυλακτήρας, ο οποίος αφενός μειώνει τη δύναμη πρόσκρουσης, αφετέρου ενημερώνει το όχημα για την πρόσκρουση. Με χρήση αυτής της πληροφορίας, το όχημα ακινητοποιείται αποφεύγοντας επιπλέον ατυχήματα, ενώ παράλληλα παίρνονται οι απαραίτητες αποφάσεις για αποφυγή του εμποδίου (στην περίπτωση που κάτι τέτοιο είναι εφικτό).

Σημαντική προσθήκη στην ασφάλεια του συστήματος αποτελεί η χρήση λέξης-κωδικού κατά τη λειτουργία συνοδείας ατόμου. Σε αυτή τη κατάσταση το όχημα δέχεται σήμα IR από τους αντίστοιχους αισθητήρες. Με τη χρήση της λέξης-κωδικού, το όχημα ελέγχει μόνο τους παλμούς αυτής της λέξης, απορρίπτοντας όλους τους υπόλοιπους που μπορεί να βρίσκονται στον ίδιο χώρο. Αυτό έχει ως αποτέλεσμα το όχημα να ακολουθεί μόνο τον κάτοχο του πομπού, και περιορίζονται τα λανθασμένα μηνύματα από τρίτους. Επιπλέον περιορίζεται ο αριθμός λανθασμένων μηνυμάτων του ίδιου του πομπού, που προέρχονται από αντανάκλαση της υπέρυθρης ακτινοβολίας στον χώρο. Αυτό ισχύει διότι κατά την λήψη τέτοιου σήματος, η πληροφορία έχει υποστεί αλλοίωση, επακόλουθο της οποίας είναι η μερική ή ολική απουσία της λέξης-κωδικού. Επομένως το συγκεκριμένο σήμα αντιμετωπίζεται, ως ξένο και απορρίπτεται. Στην ιδιάζουσα περίπτωση που ληφθεί αυτούσιο σήμα στους δύο ακριανούς αισθητήρες IR (δεξιός-αριστερός), το σύστημα το θεωρεί λάθος (πιθανό σήμα από αντανάκλαση) και ακινητοποιεί το όχημα.

Τέλος όσο αφορά τον έλεγχο του οχήματος, επειδή είναι απομακρυσμένη ενέργεια και ίσως προέρχεται μέσω διαδικτύου, χρησιμοποιήθηκε το πρωτόκολλο SSH. Αυτό παρέχει την απαραίτητη ασφάλεια και κρυπτογράφηση μεταξύ χρήστη και οχήματος. Η χρήση SSH προκαλεί μια μικρή καθυστέρηση μερικών δευτερολέπτων στην αρχή, για την εγκαθίδρυση της επικοινωνίας. Σε αυτή τη διαδικασία περιλαμβάνεται η είσοδος του χρήστη στο σύστημα του οχήματος. Έπειτα από την επίτευξη της σύνδεσης, η όλη επικοινωνία είναι κρυπτογραφημένη

και με αμελητέα καθυστέρηση. Το SSH χρησιμοποιείται ευρέως και θεωρείται αναγνωρισμένος τρόπος ασφαλούς επικοινωνίας μέσω διαδικτύου.

5.1.3 Ανάλυση SWOT

Η παρούσα υλοποίηση χαρακτηρίζεται ως οικονομική για τις παροχές της. Αποτελεί ένα αξιόπιστο και ασφαλές σύστημα. Η απόκριση του συστήματος βρίσκεται στα αποδεκτά πλαίσια, τόσο κατά τον χειρισμό του, όσο και κατά την αυτόνομη πορεία του. Υπάρχει η δυνατότητα κίνησης του οχήματος έχοντας επιπλέον βάρος. Η εκτεταμένη αυτονομία λειτουργίας, που αποτελεί ξεχωριστή ιδιότητα του συστήματος, παρέχει τη δυνατότητα παρακολούθησης ενός χώρου για μεγάλη χρονική διάρκεια. Επιπλέον η δυνατότητα αποφυγής εμποδίων προσφέρει διευκόλυνση στη χρήση του κατά τη συνοδεία ατόμου. Τέλος η λειτουργία αποτροπής σύγκρουσης προστατεύει, τόσο το όχημα, όσο και το γύρω περιβάλλον του.

Από την άλλη μεριά, λόγω περιορισμένων οικονομικών δαπανών, το σύστημα υστερεί από άποψη υλικού. Συγκεκριμένα το αμάξωμα είναι αρκετά δυσκίνητο, περιορίζοντας έτσι την κίνηση του στον χώρο και την αποφυγή εμποδίων. Επιπρόσθετα το όχημα έχει κίνηση σε έναν μόνο τροχό (στον πίσω δεξιά), με αποτέλεσμα να περιορίζεται η ταχύτητα κίνησής του και το βάρος που μπορεί να μεταφέρει. Τέλος οι αισθητήρες απόστασης λόγω της ποιότητας και του αριθμού τους, περιορίζουν τον αριθμό των εμποδίων που μπορεί το σύστημα να εντοπίσει. Αυτό έχει ως αποτέλεσμα περιπτώσεις σύγκρουσης, ενώ επηρεάζεται και η λειτουργία αποφυγής εμποδίων.

Το όχημα κατά τη λειτουργία του χρησιμοποιεί ηλεκτρομαγνητικά σήματα (IR, Wi-Fi), και υπέρηχους. Τυχόν παρεμβολές στον χώρο χρήσης του οχήματος, είτε λόγω καιρικών συνθηκών, είτε από άλλες συσκευές, έχουν ως αποτέλεσμα τη δυσλειτουργία του συστήματος. Επιπλέον, η κίνηση του οχήματος επηρεάζεται από το βάρος του, καθώς και από την κλίση και είδος του εδάφους. Σε περίπτωση χρήσης του σε ανώμαλο έδαφος τα αποτελέσματα δεν είναι σταθερά. Το όχημα προορίζεται, τόσο για χρήση σε εξωτερικό, όσο και σε εσωτερικό χώρο, με την προϋπόθεση ότι το έδαφος ικανοποιεί τους περιορισμούς που δόθηκαν. Η κατασκευή του οχήματος προσφέρει μερική προστασία από το νερό. Σε καμία περίπτωση δεν ενδείκνυται η χρήση του σε περιβάλλον με πολύ υψηλή υγρασία ή βροχή.

Οι αισθητήρες απόστασης εντοπίζουν μόνο λείες επιφάνειες κάθετες σε αυτούς ή με πολύ μικρή κλίση (max 15°). Σε οποιαδήποτε διαφορετική περίπτωση υπάρχει κίνδυνος

σύγκρουσης. Όσο αφορά τις μπαταρίες του οχήματος προτείνεται φόρτισή τους τουλάχιστον μία φορά τον μήνα, ανεξάρτητα από τη χρήση του οχήματος. Επιπρόσθετα, το σύστημα δεν προειδοποιεί όταν τελειώνει η μπαταρία του. Σε περίπτωση που η LCD φωτίζει αλλά όλα τα LEDs του οχήματος είναι σβηστά, πολύ πιθανόν το σύστημα να χρειάζεται φόρτιση. Σε διαφορετική περίπτωση ίσως βρίσκεται σε κάποια προβλεπόμενη κατάσταση (Πίνακας 9). Τέλος πρέπει να τονιστεί, ότι το σύστημα πρέπει να είναι απενεργοποιημένο κατά τη διάρκεια φόρτισής του.

5.2 Γενική χρήση και έλεγχοι του συστήματος

Το σύστημα παρουσιάστηκε στο φεστιβάλ βιομηχανικής πληροφορικής της Καβάλας «I2Fest» από τις 7/10/2016 έως τις 15/10/2016. Κατά τη διάρκεια του φεστιβάλ το σύστημα ήταν ενεργό καθημερινά για πάνω από 12 ώρες (όχι αυτόνομα). Αυτό το γεγονός αποτελεί σημαντικό έλεγχο της υλοποίησης. Το σύστημα παρατηρήθηκε από πλήθος κόσμου πριν και κατά τη διάρκεια του φεστιβάλ, ενώ έγινε επίδειξη της λειτουργίας του σε φοιτητές και καθηγητές. Αποτέλεσμα της όλης διαδικασίας, ήταν η επιβεβαίωση της ορθής λειτουργίας του συστήματος για μεγάλο χρονικό διάστημα, καθώς και η σταθερότητά του.

5.2.1 Χρήση του συστήματος

Το σύστημα επιτρέπει δύο τρόπους χρήσης, αποσκοπώντας στην ικανοποίηση δύο διαφορετικών αναγκών. Ο πρώτος τρόπος αφορά τον χειρισμό του οχήματος μέσω Wi-fi. Η διαδικασία είναι απλή και αποτελείται από τα παρακάτω βήματα.

Αρχικά τοποθετούμε τους δύο διακόπτες του οχήματος στην κατάσταση ON. Αυτό έχει ως αποτέλεσμα την τροφοδοσία του συστήματος με ρεύμα. Μόλις ολοκληρωθεί η εκκίνηση του συστήματος, ανάβει το κόκκινο LED του οχήματος. Στην συνέχεια παρατηρούμε την IP που εμφανίζεται στην LCD οθόνη του. Στην περίπτωση ελέγχου μέσω διαδικτύου, πρέπει να γίνουν οι κατάλληλες ρυθμίσεις στον δρομολογητή (router) του δικτύου. Εφόσον το όχημα έχει πάρει IP είμαστε σε θέση να χρησιμοποιήσουμε την εφαρμογή CarControl. Στην περίπτωση χρήσης τοπικού δικτύου, για την επικοινωνία, πρέπει να επιβεβαιωθεί ότι το όχημα και ο Η/Υ βρίσκονται στο ίδιο δίκτυο. Στη συνέχεια εισάγουμε την IP στο αντίστοιχο πλαίσιο της εφαρμογής και επιλέγουμε «Σύνδεση». Μόλις η σύνδεση επιτευχθεί εμφανίζεται αντίστοιχο μήνυμα στο γραφικό περιβάλλον. Από αυτό το σημείο και έπειτα, με χρήση χειριστηρίου,

μπορούμε να ελέγξουμε το όχημα. Στο γραφικό περιβάλλον εμφανίζονται όλες οι τιμές από τους αισθητήρες απόστασης, ενώ με την επιλογή «Κάμερα» ή «Υ» (από το χειριστήριο) μπορούμε να ενεργοποιήσουμε/απενεργοποιήσουμε τη ροή δεδομένων βίντεο.

Ο δεύτερος τρόπος λειτουργίας είναι η συνοδεία ατόμου. Σε αυτή την περίπτωση η χρήση είναι απλούστερη. Αρχικά, όπως και προηγουμένως, πρέπει να εκκινήσουμε το σύστημα τοποθετώντας τους δύο διακόπτες τροφοδοσίας στην θέση ON. Η εκκίνηση του συστήματος έχει ολοκληρωθεί μόλις ανάψει το κόκκινο LED. Με την ολοκλήρωση της εκκίνησης είμαστε σε θέση, σε οποιαδήποτε στιγμή, να ενεργοποιήσουμε την λειτουργία συνοδείας (Follow me). Η ενεργοποίηση γίνεται με το πάτημα του κουμπιού μπροστά από το πράσινο LED. Αυτό έχει ως συνέπεια να ανάψει το πράσινο LED και να ξεκινήσει η λειτουργία ακολουθίας. Επόμενο βήμα είναι να ενεργοποιηθεί ο πομπός. Αυτό γίνεται με τη μετατόπιση του διακόπτη που βρίσκεται πάνω στον πομπό. Εδώ να σημειωθεί ότι ο πομπός μπορεί να είναι και από πριν ενεργοποιημένος. Τέλος ο χρήστης μπορεί να κινηθεί, έχοντας τον πομπό στραμμένο μονίμως προς το όχημα και με διατήρηση της οπτικής επαφής μεταξύ αυτών των δύο.

Για την απενεργοποίηση του συστήματος πρέπει να βρίσκεται το όχημα στην κατάσταση απομακρυσμένου ελέγχου (αρχική κατάσταση μετά από την εκκίνηση). Εάν βρίσκεται στη λειτουργία συνοδείας, η μετάβαση στην επιθυμητή κατάσταση πραγματοποιείται με το πάτημα του κόκκινου κουμπιού. Όσο το όχημα βρίσκεται στην αρχική κατάσταση υπάρχουν δύο επιλογές. Η πρώτη απευθύνεται σε προγραμματιστές, τερματίζοντας την εκτέλεση του «Arduino_com_full.py» και όλων των διεργασιών-παιδιά του. Η χρήση αυτής της επιλογής γίνεται με παρατεταμένο πάτημα του κόκκινου κουμπιού μέχρι να σβήσει το κόκκινο LED. Τέλος η δεύτερη επιλογή απευθύνεται στο μέσο χρήστη και προκαλεί ολικό τερματισμό του λειτουργικού του οχήματος. Αυτό επιτυγχάνεται με το παρατεταμένο πάτημα του κόκκινου και πράσινου κουμπιού μέχρι να ανάψει το πράσινο LED. Μόλις σβήσουν και τα δύο LEDs (πρώτα το πράσινο και αρκετά αργότερα το κόκκινο) μπορούμε να κλείσουμε με ασφάλεια την παροχή ρεύματος (τοποθέτηση και των δύο διακοπών ρεύματος στην θέση OFF).

5.2.2 Έλεγχοι ορθής λειτουργίας – αποσφαλμάτωση

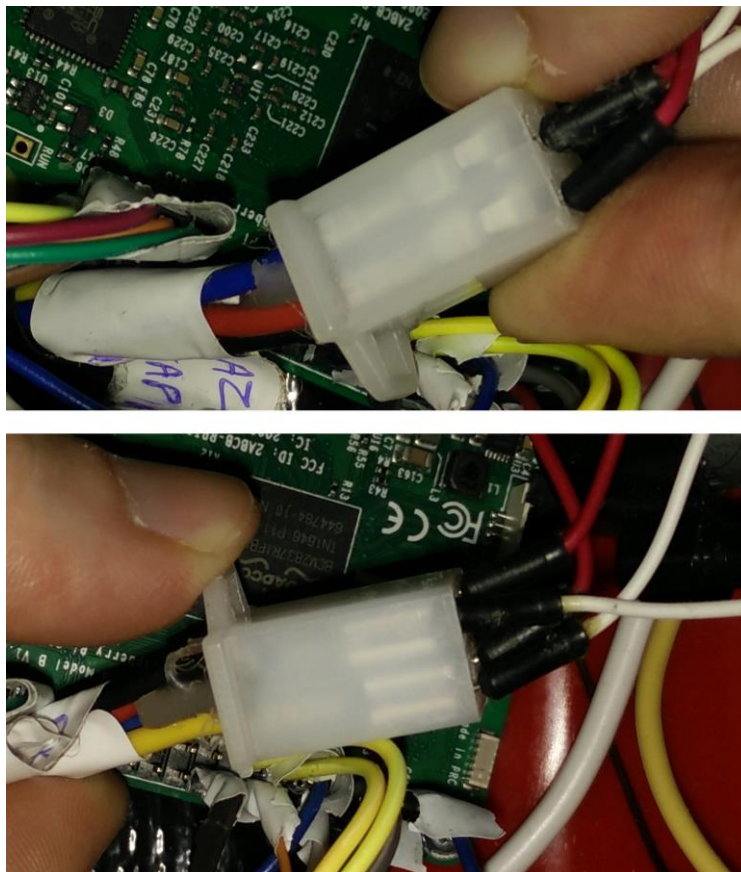
Λόγω της πολυπλοκότητας του συστήματος είναι αρκετά πιθανό και λογικό να εμφανιστούν προβλήματα κατά τη λειτουργία του. Για τα περισσότερα από αυτά ευθύνονται

αστοχίες του υλικού, της κατασκευής ή και του περιβάλλοντος χώρου. Σε αυτό το σημείο αναφέρονται τα πιο γνωστά, καθώς και τρόποι επίλυσής τους.

1. Έχει γίνει σύνδεση με SSH αλλά το όχημα δεν κινείται.
 - Σε αυτή την περίπτωση πρέπει να ελεγχθεί αν ο διακόπτης τροφοδοσίας του κινητήρα είναι στην θέση ON.
 - Πιθανό αίτιο επίσης αποτελεί η υφή και κλίση του εδάφους. Το όχημα κινείται ευκολότερα σε ομαλές και λείες επιφάνειες.
 - Το βάρος του οχήματος επηρεάζει αρνητικά την κίνηση του. Συνίσταται η απαλλαγή του οχήματος από περιττά βάρη.
 - Στην περίπτωση που ο διακόπτης τροφοδοσίας βρίσκεται στη σωστή θέση πολύ πιθανό να χρειάζεται επαναφόρτιση η μπαταρία του κινητήρα (η μικρή κάτω από το κάθισμα).
 - Εάν δεν λυθεί το πρόβλημα πρέπει να γίνει έλεγχος του κυκλώματος, και να επιβεβαιωθεί ότι όλα τα καλώδια είναι σωστά συνδεδεμένα (**Κύκλωμα του οχήματος με χρήση Raspberry Pi 3, Κύκλωμα του οχήματος με χρήση Raspberry Pi**).
2. Μετά το πέρας του χρόνου εκκίνησης δεν έχει ανάψει κανένα LED.
 - Εάν το λειτουργικό του Raspberry Pi εκτελείται, τότε πρέπει να ελεγχθεί εάν έχουν αποσυνδεθεί τα USB των Arduino.
 - Εάν το λειτουργικό δεν εκτελείται τότε πιθανότατα η μεγάλη μπαταρία χρειάζεται φόρτιση.
 - Στην περίπτωση που δεν ισχύει κάτι από τα προηγούμενα δύο, πρέπει να ελεγχθεί η κατάσταση του κυκλώματος και των συνδέσεων.
3. Το όχημα δεν εμφανίζει διεύθυνση IP αλλά μήνυμα «No IP».
 - Πρέπει πρώτα να ελεγχθεί εάν λειτουργεί το όλο σύστημα. Στην περίπτωση που το λειτουργικό εκτελείται, τότε πολύ πιθανόν το σημείο που βρίσκεται το όχημα να μην έχει καλή κάλυψη Wi-Fi. Το πρόβλημα μπορεί να επιλυθεί με την αλλαγή τοποθεσίας ή με τη χρήση ισχυρότερης κεραίας Wi-fi.
 - Σε περίπτωση μη λειτουργίας του συστήματος, κατά πάσα περίπτωση, η μεγάλη μπαταρία χρειάζεται επαναφόρτιση.

-
4. Το όχημα δεν εμφανίζει κανένα μήνυμα στην οθόνη. Αυτό είναι αποτέλεσμα προβλήματος στο υλικό του οχήματος. Πρέπει να ελεγχθεί η κατάσταση των υλικών του κυκλώματος και η συνδεσμολογία.
 5. Δεν είναι δυνατή η σύνδεση μέσω SSH ενώ το όχημα έχει κανονικά IP.
 - Πρέπει να επιβεβαιωθεί η λειτουργία του Raspberry Pi και η σύνδεση του οχήματος και του H/Y του χρήστη στο ίδιο δίκτυο.
 - Σε περίπτωση ελέγχου μέσω διαδικτύου πρέπει να ελεγχθούν οι ρυθμίσεις του δρομολογητή του δικτύου (router).
 - Εάν δεν έχει βρεθεί το πρόβλημα, μπορεί να χρησιμοποιηθεί η έκδοση του CarControl με λειτουργία αποσφαλμάτωσης. Έτσι μπορούν να χρησιμοποιηθούν τα ανάλογα μηνύματα για πιο κατευθυνόμενη έρευνα.
 6. Το όχημα δεν ακολουθεί τον πομπό.
 - Αρχικά πρέπει να ελεγχθεί εάν το όχημα μπορεί να κινηθεί όπως περιγράφεται στην πρώτη περίπτωση.
 - Εάν δεν είναι εφικτός αυτός ο έλεγχος, ή μέρος αυτού, μπορεί να βραχυκυκλωθεί απευθείας το κόκκινο καλώδιο με το μπλε, και το μαύρο με το κίτρινο (**Εικόνα 64**). Τα καλώδια αυτά βρίσκονται στο μπροστινό μέρος του οχήματος, κάτω από το καπό και πάνω στο Raspberry Pi. Τα καλώδια αυτά αντιστοιχούν στην μπαταρία (κόκκινο-μαύρο) και στον κινητήρα κίνησης (μπλε-κίτρινο). Με αυτό τον τρόπο βραχυκυκλώνουμε τον κινητήρα απευθείας στην μπαταρία κίνησής του. Εάν δεν περιστραφεί ο τροχός τότε πολύ πιθανό να χρειάζεται επαναφόρτιση η μπαταρία του.
 - Έπειτα πρέπει να επιβεβαιωθεί ότι βρίσκεται σε λειτουργία η συνάρτηση «Follow me». Αυτό ελέγχεται με το αν είναι αναμμένο το πράσινο LED.
 - Στην περίπτωση που είναι αναμμένο το μπλε LED το όχημα δεν θα κινηθεί μιας και δέχεται ισχυρή ένταση σήματος Bluetooth από τον πομπό. Συνίσταται απομάκρυνση ή απενεργοποίηση του Bluetooth του πομπού.
 - Τέλος πρέπει να ελεγχθεί εάν χρειάζονται αλλαγή οι μπαταρίες του πομπού. Η εξασθένηση της τάσης τους επηρεάζει την απόσταση λειτουργίας της συνοδείας.
 7. Δεν υπάρχει δυνατότητα παρακολούθησης της ροής δεδομένων βίντεο από την εφαρμογή CarControl.

- Σε αυτή την περίπτωση αρχικά ελέγχουμε την δυνατότητα παρακολούθησης μέσω φυλλομετρητή. Αυτό μπορεί να επιτευχθεί με την εισαγωγή της IP και της πόρτας 8080 στο κατάλληλο πλαίσιο. Θα πρέπει να ανοίξει η ροή δεδομένων.
- Σε αντίθετη περίπτωση πρέπει να επιβεβαιωθεί η σύνδεση της κάμερας με USB. Στην περίπτωση επανασύνδεσης της ίσως χρειαστεί επανεκκίνηση του συστήματος ή της διεργασίας Motion.



Εικόνα 64: Απευθείας βραχυκύκλωση κινητήρα κίνησης

5.3 Συμπεράσματα της προσέγγισης

Συνοψίζοντας, η παρούσα υλοποίηση αφορά την κατασκευή ενός αυτόνομου, μη επανδρωμένου οχήματος με δυνατότητα συνοδείας ατόμου και αποφυγής εμποδίων. Επιπλέον παρέχει τη δυνατότητα ασφαλούς χειρισμού του μέσω διαδικτύου. Καινοτόμο στοιχείο της αποτελεί η αναλογία μεταξύ χαμηλού κόστους και παροχής υπηρεσιών. Μπορούμε έτσι να συμπεράνουμε ότι το κόστος κατασκευής τέτοιων συστημάτων θα μειωθεί περαιτέρω.

Παράλληλα η παρούσα κατασκευή θεωρείται ευέλικτη, λόγω της δυνατότητας προσθήκης επιπλέον λειτουργιών. Επιπρόσθετα, ο χειρισμός του οχήματος γίνεται εύκολα με τη χρήση χειριστηρίου. Η γενικότερη χρήση της υλοποίησης θεωρείται εύκολη για τον μέσο χρήστη. Το χαμηλό κόστος, η πληθώρα λειτουργιών και η εύκολη χρήση μας οδηγούν στο συμπέρασμα ότι η ευρεία κατασκευή αυτόνομων οχημάτων για τον μέσο χρήστη είναι εφικτή.

5.4 Προβλήματα που προέκυψαν κατά την υλοποίηση

Κατά την υλοποίηση του συστήματος όπως είναι λογικό προέκυψαν αρκετά προβλήματα. Τα προβλήματα αυτά ανήκουν τόσο στον τομέα της βελτιστοποίησης, όσο και σε αυτόν της ασφάλειας. Για την επίτευξη ποιοτικής και ασφαλούς λειτουργίας του συστήματος, θεωρήθηκε αδήριτη ανάγκη η επίλυσή τους.

Αρχικό πρόβλημα ήταν η πολύ αργή απόκριση του συστήματος κατά τον έλεγχο του οχήματος. Συγκεκριμένα η απόκριση είχε καθυστέρηση αρκετών δευτερολέπτων για την εκτέλεση μιας εντολής. Επιπλέον συχνό φαινόμενο αποτελούσε η τυχαία απώλεια εντολών ελέγχου. Αίτια των προβλημάτων αυτών ήταν η χρήση αρχείων, που ήταν αποθηκευμένα στην κάρτα μνήμης SD, και η χρήση sh script για την προώθηση των εντολών. Το πρόβλημα αυτό λύθηκε με δύο ενέργειες. Αρχικά αντικαταστάθηκε το sh script με αντίστοιχη υλοποίηση σε Python. Επιπλέον τα αρχεία που χρησιμοποιήθηκαν προσαρτώνται πλέον στην μνήμη RAM του συστήματος κατά την εκκίνηση του. Έτσι λόγω της γρηγορότερης εγγραφής και ανάγνωσής τους από το `Arduino_com_full.py` επιλύεται το πρόβλημα, ενώ παράλληλα αυξάνεται και ο χρόνος ζωής της κάρτας μνήμης SD.

Ένα δεύτερο σημαντικό πρόβλημα ήταν η ανανέωση των τιμών των αισθητήρων απόστασης, που εμφανίζονται στα πεδία του γραφικού περιβάλλοντος της CarControl. Η λειτουργία αυτή γινόταν με χρήση μίας ξεχωριστής δευτερεύουσας διεργασίας. Η επίλυση δόθηκε με αντικατάσταση αυτής της διεργασίας με την `backgroundworker`, η οποία εξειδικεύεται στην διαρκή ενημέρωση πεδίων γραφικού περιβάλλοντος.

Επιπρόσθετα κατά τη λειτουργία της συνοδείας ατόμου παρατηρήθηκε η λήψη λανθασμένων σημάτων. Αυτό είχε ως αποτέλεσμα, πολλές φορές το όχημα να κινείται προς την κατεύθυνση της πηγής της παρεμβολής και όχι προς τον χρήστη. Λύση αποτέλεσε η κατασκευή ενός πομπού, ο οποίος στέλνει κωδικοποιημένα μηνύματα. Το όχημα προσαρμόστηκε στο να ακολουθεί μόνο τα μηνύματα που περιέχουν τον συγκεκριμένο κωδικό.

Με αυτόν τον τρόπο το σύστημα προστατεύεται από παρεμβολές άλλων συσκευών, ενώ ταυτόχρονα μειώθηκε σημαντικά η λήψη λανθασμένων μηνυμάτων από αντανακλάσεις.

Κατά τη διαδικασία αυτόνομης κίνησης εμφανίστηκε το πρόβλημα παράληψης της εκτέλεσης μερικών εντολών. Αίτιο αυτού του προβλήματος ήταν ο μικρός χρόνος μεσολάβησης μεταξύ δύο (ή περισσότερων) αποστολών εντολής από το Raspberry Pi στο Arduino. Για τη λύση αυτού του προβλήματος τηρώντας τη γρήγορη απόκριση του συστήματος, υιοθετήθηκε ένα πρωτόκολλο αποστολής με επιβεβαίωση. Με αυτόν τον τρόπο το σύστημα αποστέλλει την εντολή μέχρι να λάβει επιβεβαίωση λήψης.

Τέλος λόγω περιστροφής του τιμονιού με τη χρήση DC κινητήρα ήταν αδύνατος ο εντοπισμός της θέσης του από το σύστημα. Για την υλοποίηση λειτουργιών αυτονομίας κίνησης θεωρήθηκε απαραίτητη η πληροφορία αυτή. Για την επίτευξη αυτού του σκοπού προστέθηκε ένα πλαίσιο από κουμπιά στο κάτω μέρος του οχήματος. Με αυτόν τον τρόπο, ανάλογα με τη θέση του τιμονιού πατιέται το ανάλογο κουμπί που αντιστοιχεί στη συγκεκριμένη θέση.

5.5 Προτάσεις για επέκταση της υλοποίησης

Όπως έχει αναφερθεί σε αρκετά σημεία του παρόντος κειμένου η συγκεκριμένη υλοποίηση αποτελεί, εκτός των άλλων, βάση για μελλοντικές υλοποιήσεις από συναδέλφους. Αρχικά συνίσταται η αναβάθμιση στο υλικό μέρος της κατασκευής. Μια τέτοια αναβάθμιση θα μπορούσε να περιέχει την προσθήκη ενός δεύτερου κινητήρα κίνησης στον πίσω αριστερό τροχό του οχήματος. Αυτό θα προσέφερε μεγαλύτερη ταχύτητα, υψηλότερο όριο βάρους καθώς και ποιοτικότερη κίνηση του οχήματος. Παράλληλα θα μπορούσε να γίνει αντικατάσταση αυτών των κινητήρων με καλύτερους, έτσι ώστε να χρησιμοποιηθεί μόνο η μεγάλη μπαταρία του οχήματος. Τέλος για τον ιδανικό χειρισμό του τιμονιού προτείνεται η αντικατάσταση του DC κινητήρα με έναν σερβοκινητήρα.

Με την επίτευξη των παραπάνω καθίσταται δυνατή η βελτίωση των υπάρχοντων λειτουργιών και η προσθήκη περαιτέρω δυνατοτήτων. Μερικές πιθανές μελλοντικές επεκτάσεις αναφέρονται στις παρακάτω γραμμές.

Ενδιαφέρουσα υλοποίηση θα αποτελούσε, η προσθήκη GPS με σκοπό τη χρήση συντεταγμένων και υπολογισμού διαδρομής από έναν σταθμό εκκίνησης σε ένα σημείο προορισμού.

Επιπλέον θα μπορούσε το όχημα να μετατραπεί σε αγωνιστικό με τρόπο τέτοιο, ώστε να μπορεί να κατευθυνθεί με τη μέγιστη δυνατή ταχύτητα και αυτόνομα, σε μία κατάλληλα διαμορφωμένη πίστα.

Δυνατή θεωρείται η ανάπτυξη εφαρμογής για κινητά τηλέφωνα (π.χ. Android) με σκοπό τον έλεγχο του οχήματος. Η υλοποίηση μιας τέτοιας εφαρμογής θα μπορούσε να κάνει χρήση τόσο της οθόνης αφής, όσο και των χειριστηρίων που κάνει η παρούσα υλοποίηση (CarControl).

Τέλος, επέκταση με ιδιαίτερη σημασία αποτελεί η κατασκευή σταθμών επαναφόρτισης. Το σύστημα θα μπορεί να εντοπίζει τον πλησιέστερο σταθμό, και το όχημα θα κατευθύνεται αυτόνομα μέχρι αυτόν.

5.6 Σύνοψη κεφαλαίου

Εν κατακλείδι στο παρόν κεφάλαιο περιγράφηκε το σύνολο του συστήματος και των λειτουργιών του. Παρουσιάστηκαν οι τρόποι χρήσης του και οι έλεγχοι που πρέπει να γίνονται σε περίπτωση σφαλμάτων. Παράλληλα επισημάνθηκαν ορισμένα συμπεράσματα που προέκυψαν από την παρούσα υλοποίηση. Στη συνέχεια αναφέρθηκαν τα προβλήματα που προέκυψαν κατά την εκπόνηση της παρούσας διπλωματική εργασίας, ενώ τέλος δόθηκε μια λίστα με πιθανές μελλοντικές επεκτάσεις του συστήματος.

Βιβλιογραφία

- [1] "Unmanned ground vehicle," https://en.wikipedia.org/wiki/Unmanned_ground_vehicle, Μάιος 2017.
- [2] "Vehicular automation," https://en.wikipedia.org/wiki/Vehicular_automation, Μάιος 2017.
- [3] Internet of Things, https://en.wikipedia.org/wiki/Internet_of_things, Μάιος 2017.
- [4] «The Vespa Robots,» <http://newatlas.com/piaggio-gita/47646/>, Μάιος 2017.
- [5] Σ. Κ. Κ. Δ. Γ. Στέφανος Γκρίτζαλης, «Secure Shell,» σε *Ασφάλεια Δικτύων Υπολογιστών*, Αθήνα, Παπασωτηρίου, 2003.
- [6] «opencv,» <http://opencv.org/>, Μάιος 2017.
- [7] «DirectInput,» Microsoft, [https://msdn.microsoft.com/en-us/library/windows/desktop/ee416842\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ee416842(v=vs.85).aspx), Μάιος 2017.
- [8] «DirectX.DirectInput,» Microsoft, [https://msdn.microsoft.com/en-us/library/windows/desktop/bb318766\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb318766(v=vs.85).aspx), Μάιος 2017.
- [9] «XInput and DirectInput,» Microsoft, [https://msdn.microsoft.com/en-us/library/windows/desktop/ee417014\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ee417014(v=vs.85).aspx), Μάιος 2017.
- [10] «NvGamepad Library,» Nvidia, https://docs.nvidia.com/gameworks/content/gameworkslibrary/coresdk/nvgamepad_library.htm, Μάιος 2017.
- [11] «Visual C++,» Microsoft, <https://msdn.microsoft.com/en-us/library/60k1461a.aspx>, Μάιος 2017.
- [12] H. D. P.J. Deitel, C++ Προγραμματισμός, Αθήνα: Μ.Γκιούρδας, 2013.
- [13] «Getting Started with the .NET Framework,» Microsoft, <https://msdn.microsoft.com/en-us/library/hh425099.aspx>, Μάιος 2017.
- [14] «WHAT IS A RASPBERRY PI?,» RASPBERRY PI, <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>, Μάιος 2017.
- [15] «Python,» <https://el.wikipedia.org/wiki/Python>, Μάιος 2017.
- [16] «python - Linux man page,» <https://linux.die.net/man/1/python>, Μάιος 2017.
- [17] «Interpreted language,» https://en.wikipedia.org/wiki/Interpreted_language, Μάιος 2017.

-
- [18] «Introduction to Arduino,» Arduino, <https://www.arduino.cc/en/Guide/Introduction>, Μάιος 2017.
- [19] Δ. Δ. Πογαρίδης, Ενσωματωμένα Συστήματα - Ο μικροελεγκτής AVR, 2013.
- [20] Π. Ιωαννίδης, «Σχεδιασμός και μετατροπή μιας απλής οικιακής συσκευής σε επαναπρογραμματιζόμενη συσκευή IP,» Κοζάνη, 2015.
- [21] etix, «Arduino-IRremote,» <https://github.com/etix/Arduino-IRremote>, Φεβρουάριος 2017.
- [22] «NewPing,» <http://playground.arduino.cc/Code/NewPing>, Νοέμβριος 2016.
- [23] «L298N,» https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf, Μάιος 2017.
- [24] «Servo Motor SG90,» <http://www.micropik.com/PDF/SG90Servo.pdf>, Μάιος 2017.
- [25] «HCSR04,» <http://www.micropik.com/PDF/HCSR04.pdf>, Μάιος 2017.
- [26] «VS1838b,» <https://www.elecrow.com/download/Infrared%20receiver%20vs1838b.pdf>, Μάιος 2017.
- [27] «MWS-7.2-12,» <http://mplenergy.pl/wp-content/uploads/2016/01/MWS-7.2-12.pdf>, Μάιος 2017.
- [28] «LM2596,» <http://pdf.datasheetcatalog.com/datasheet/nationalsemiconductor/DS012583.PDF>, Μάιος 2017.
- [29] «3FM4_5,» http://www.ferlacar.com/Prodotti/Longway/UPS/3FM4_5_7.pdf, Μάιος 2017.
- [30] «Bluetooth BC417,» <https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/CSR-BC417-datasheet.pdf>, Μάιος 2017.
- [31] «Visual Studio,» Microsoft, <https://www.visualstudio.com/>, Μάιος 2017.
- [32] «Arduino IDE,» Arduino, <https://www.arduino.cc/en/main/software>, Μάιος 2017.
- [33] «About Notepad++,» <https://notepad-plus-plus.org/>, Μάιος 2017.
- [34] «sketchup,» <http://www.sketchup.com/>, Μάιος 2017.
- [35] «Win32DiskImager,» <http://wiki.hioproject.org/index.php?title=Win32DiskImager>, Μάιος 2017.
- [36] «RPi_Low-level_peripherals,» http://elinux.org/RPi_Low-level_peripherals, Μάιος 2017.
- [37] «H-bridge,» https://en.wikipedia.org/wiki/H_bridge, Μάιος 2017.
- [38] «arduino analog read,» <https://www.arduino.cc/en/Tutorial/AnalogInput>, Μάιος 2017.
- [39] «/IR333_A_datasheet,» https://cdn-shop.adafruit.com/datasheets/IR333_A_datasheet.pdf, Μάιος 2017.

-
- [40] «arduinoBoardUno,» <https://www.arduino.cc/en/main/arduinoBoardUno>, Μάιος 2017.
- [41] «Arduino-IRremote,» <https://github.com/z3t0/Arduino-IRremote>, Μάιος 2017.
- [42] «SimpleTimer,» <https://playground.arduino.cc/Code/SimpleTimer>, Μάιος 2017.
- [43] «Arduino-IRremote-multi,» <https://github.com/etix/Arduino-IRremote>, Φεβρουάριος 2017.
- [44] «raspi-config,» <https://www.raspberrypi.org/documentation/configuration/raspi-config.md>, Μάιος 2017.
- [45] Π. ΔΗΜΗΤΡΙΟΣ-ΝΕΚΤΑΡΙΟΣ, «Σχεδιασμός και Υλοποίηση οχήματος ελεγχόμενου μέσω Διαδικτύου,» Κοζάνη, 2014.
- [46] «wiringpi,» <https://projects.drogon.net/raspberry-pi/wiringpi/download-and-install/>, Μάιος 2017.
- [47] «motion,» <http://www.lavrsen.dk/foswiki/bin/view/Motion/MotionGuide>, Μάιος 2017.
- [48] «tmpfs,» <https://wiki.archlinux.org/index.php/tmpfs>, Μάιος 2017.
- [49] «openbsd,» <http://man.openbsd.org/fstab.5>, Μάιος 2017.
- [50] «mount man page,» <https://linux.die.net/man/8/mount>, Μάιος 2017.
- [51] «crontab,» <http://pubs.opengroup.org/onlinepubs/9699919799/utilities/crontab.html>, Μάιος 2017.
- [52] «bluetooth-proximity,» <https://github.com/ewenchou/bluetooth-proximity>, Μάιος 2017.
- [53] «backgroundworker,» [https://msdn.microsoft.com/en-us/library/system.componentmodel.backgroundworker\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.componentmodel.backgroundworker(v=vs.110).aspx), Μάιος 2017.