

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Τσώλης Ορέστης

A.M. 713242017049

Εισηγητής: Βογιατζής Ιωάννης, Καθηγητής

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

(Κενό φύλλο)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

**Τσώλης Ορέστης
Α.Μ. 713242017049**

Εισηγητής:

Βογιατζής Ιωάννης, Καθηγητής

Εξεταστική Επιτροπή:

Σταύρος Φατούρος, Αν. Καθηγητής

Γεώργιος Λεντάρης, Επ. Καθηγητής

Ημερομηνία εξέτασης

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

(Κενό φύλλο)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Τσώλης Ορέστης του Εμμανουήλ, με αριθμό μητρώου 713242017049 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

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

**Ο Δηλών
Τσώλης Ορέστης**



Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

(Κενό φύλλο)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές μου, που μου έδειξαν κατανόηση για και υπομονή κατά την διάρκεια της διπλωματικής μου εργασίας, καθώς βρισκόμουν σε δύσκολες συνθήκες.

Θα ήθελα επίσης να ευχαριστήσω τους γονείς μου, για την άνευ όρων αλληλεγγύη που μου προσέφεραν καθ' όλη την διάρκεια των σπουδών μου, και της διπλωματικής εργασίας.

Τέλος, θέλω να ευχαριστήσω θερμά τον κ. Jan-Willem van Bentum για την εκπληκτική κατανόηση και υποστήριξη που μου προσέφερε κατά το χρονικό διάστημα εκπόνησης της παρούσας εργασίας καθώς παρέμενα στην Ουτρέχτη, Ολλανδία

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

(Κενό φύλλο)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Περίληψη

Την τελευταία δεκαετία, έχουν κάνει την εμφάνισή τους στην αγορά, πολλαπλά οχήματα με λειτουργία αυτόματης οδήγησης, με ολίγη ή ακόμη και μηδαμινή ανθρώπινη παρέμβαση. Αυτά τα συστήματα συνδυάζουν πολλαπλών ειδών αισθητήρες, κάμερες και αλγορίθμους για να ενισχύσουν την επίγνωση του περιβάλλοντος του οχήματος, και να υποστηρίξουν οδηγούς στην πολύωρη οδήγηση που απαιτεί πλήρης αφοσίωση καθ' όλη την διάρκεια της μετακίνησης. Η απουσία της προσοχής, ακόμη και στιγμιαία, μπορεί έχει αποτέλεσμα σοβαρά ατυχήματα, ακόμη και απώλεια ζωής. Επίσης, τελευταία εξερευνώνται περαιτέρω και οι χρήσεις της αυτόνομης οδήγησης στην αυτοματοποίηση μετακίνησης αγαθών και προσώπων χωρίς ανθρώπινη επίβλεψη

Η παρούσα διπλωματική εργασία ασχολείται με την κατασκευή ενός αυτοκινούμενου οχήματος το οποίο χρησιμοποιεί σύστημα εντοπισμού λωρίδας και ευθυγράμμισης σε αυτήν, σε συνδυασμό με αισθητήρες απόστασης για τον έλεγχο της περιμέτρου του οχήματος για την αποφυγή εμποδίων και συγκρούσεων. Το τελικό αποτέλεσμα αποτελεί ένα απλό και ευέλικτο σύστημα αυτόνομης οδήγησης που ανοίξει την πόρτα σε πολλές δυνατότητες.

Abstract

In the last decade, multiple vehicles with self-driving capabilities that require little to no human intervention, have made their debut in the market. Those systems combine multiple kinds of sensors, cameras and algorithms to strengthen the environmental awareness of the vehicle, and to provide support to drivers that find themselves driving for long periods of time that require full attention during the whole period. The lapse in focus, even momentarily, can be the cause of serious accidents and even the loss of life. Also, lately there has been a surge in interest in the uses of autonomous driving in the full automation of the transport of goods and people, without human supervision.

The present thesis focuses on the creation of a self driving vehicle using a lane detection system that centers the vehicle in the lane, in combination with distance sensors for the monitoring of the immediate surroundings of the vehicle for collision and obstacle avoidance. The final result is a simple and flexible self-driving system that opens the door to a multitude of possibilities.

Επιστημονική περιοχή: Σύστημα αυτόνομης οδήγησης

Λέξεις κλειδιά: Raspberry Pi, Arduino, Αισθητήρες απόστασης, Εντοπισμός λωρίδας

Περιεχόμενα

Περίληψη	9
Abstract	9
Περιεχόμενα	10
1. Εισαγωγή.....	13
1.1. Αντικείμενο διπλωματικής εργασίας.....	14
2. Σχετικές πληροφορίες	15
2.1. Ιστορία της αυτόνομης οδήγησης	15
2.2. Το Raspberry Pi	17
2.3. Το Arduino.....	18
2.4. Η κάμερα	20
2.5. Αισθητήρες απόστασης	20
2.6. Κινητήρες	22
3. Αρχιτεκτονική οχήματος	26
3.1. Αρχιτεκτονική υλικού.....	26
3.2. Αρχιτεκτονική λογισμικού	27
3.3 Σύνδεση στο όχημα εξ' αποστάσεως, μέσω VPN	29
4. Θεωρητικό υπόβαθρο χρησιμοποιηθέντων αλγορίθμων	30
4.1. Γκαουσιανό θόλωμα – Gaussian Blur	30
4.2. Αλγόριθμος Canny	33
4.3. Μετασχηματισμός Hough	39
5. Υλοποίηση κώδικα	44
5.1. Εφαρμογή κινητής συσκευής	44
5.1.1 Λειτουργία χειροκίνητης κίνησης “m”	46
5.1.2 Λειτουργία εντοπισμού λωρίδας “s”	49
5.1.3 Λειτουργία ελεύθερης περιπλάνησης “f”	52
5.2 Λειτουργίες του Raspberry Pi	53
5.2.1 Επικοινωνία με εξωτερικές συσκευές	53
5.2.2 Εντοπισμός λωρίδας	54
5.3. Λειτουργίες του Arduino	62
5.4 Πειράματα λειτουργιών οχήματος	64
6. Συμπεράσματα και μελλοντικές βελτιώσεις	67
6.1 Συμπεράσματα.....	67
6.1.1 Συμπεράσματα για το όχημα	67
6.1.2 Συμπεράσματα για το λογισμικό	67
6.2 Βελτιώσεις συστήματος	68
6.2.1 Βελτιώσεις στο όχημα	68
6.2.2 Βελτιώσεις στο λογισμικό	69
7. Παράρτημα.....	70
7.1. Εγκατάσταση απαραίτητων βιβλιοθηκών στο Raspberry Pi.....	70
7.2 Δημιουργία του τοπικού δικτύου - hotspot - του Raspberry Pi.....	70
7.3 Εγκατάσταση του προγράμματος Zerotier	72
7.4 Εκτέλεση κώδικα.....	73
8. Πηγές και βιβλιογραφία	75

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

(Κενό φύλλο)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

1. Εισαγωγή

Πρόσφατα παρατηρείται ένα έντονο κέντρισμα ενδιαφέροντος για τον τομέα της Τεχνητής Νοημοσύνης (AI). Με την επένδυση σε αυτόν τον τομέα, η ανθρωπότητα ανακαλύπτει τις ατελείωτες δυνατότητες προσφέρει σε όλους τους τομείς της ζωής. Υπό την ομπρέλα της τεχνητής νοημοσύνης συμπεριλαμβάνεται και η αυτόνομη οδήγηση έχει την ικανότητα να επηρεάσει πολλαπλούς τομείς της καθημερινότητάς μας, όπως η αυτοκινητοβιομηχανία, υγεία, πρόνοια, πολεοδομία, κυκλοφορία, μεταφορά και άλλους τομείς. Κύρια παραδείγματα έργων, πειραμάτων και προϊόντων που χρησιμοποιούν αυτόνομη οδήγηση αποτελούν:

- **Η προς το παρόν υπό κατασκευή πόλη της Σαουδικής Αραβίας “Νίομ”:** Στα πλάνα “Vision 2030” της Σαουδικής Αραβίας για την απεξάρτησης από το πετρέλαιο, περιλαμβάνεται η πρωτοποριακή πόλη “Νίομ”. Σκοπεύεται να χρησιμοποιηθούν όλες τις τελευταίες τεχνολογίες που έχει να προσφέρει η ανθρωπότητα, συμπεριλαμβανομένων αυτόνομα μέσα μαζικής μεταφοράς.[1]
- **Πειράματα της Google υπό την εταιρία “Waymo”:** Επίσημα πειράματα με αυτόνομα αμάξια άρχισαν το 2009[2]. Οι πρώτες δοκιμασίες προσφέροντας περιορισμένες υπηρεσίες ταξί για εθελοντές, από αυτόνομα αμάξια, ξεκίνησαν το 2017 στην Φοίνιξ, Αριζόνα[3]. Το 2020 η Waymo ανακοίνωσε υπηρεσίες “ρομπο-ταξί” ανοικτές για το ευρύτερο κοινό.[4]. Τα πιο πρόσφατα σχέδια της εταιρίας περιλαμβάνουν επέκταση υπηρεσιών στο Σαν Φρανσίσκο[5] και Λος Άντζελες[6], Καλιφόρνια, και επέκταση των υπηρεσιών στην μητροπολιτική περιοχή της Φοίνιξ, Αριζόνα[7].
- **Τα αμάξια της Τέσλα:** Πιθανώς το πιο διαδεδομένο παράδειγμα ενός διαδεδομένου αυτόνομου αμαξιού είναι από την Τέσλα. Καθώς η Τέσλα έχει πουλήσει εκατομμύρια αμάξια[8] τα οποία χρησιμοποιούνται σε πυκνοκατοικημένες περιοχές, η εταιρία αυτή βρίσκει τον εαυτό της στην πρώτη γραμμή ανάπτυξης λογισμικού και συστημάτων που καθιστούν την αυτόνομη οδήγηση εφικτή σε απρόβλεπτα περιβάλλοντα. Λόγω τούτου, έχουν κληθεί να αντιμετωπίσουν πολλαπλές νομικές διαμάχες σημασιολογικού χαρακτήρα[9] με θέμα το επίπεδο αυτονομίας[10] των αμαξιών της εταιρίας.

Εκτός των προαναφερθέντων, υπάρχουν αρκετοί παίκτες σε όλον τον πλανήτη που προσπαθούν να φτάσουν το ορόσημο του “πρώτου αυτόνομου οχήματος επιπέδου 3+[10]” στην αγορά. Είναι σίγουρο πως κάθε όραμα του μέλλοντος περιέχει αυτόνομη οδήγηση κάποιου βαθμού.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

1.1. Αντικείμενο διπλωματικής εργασίας

Στόχος της διπλωματικής εργασίας είναι η κατασκευή ενός αυτοκινούμενου οχήματος το οποίο εντοπίζει λωρίδα δρόμου και προσπαθεί να εντοπίσει και εμπόδια (όπως αμάξια), χρησιμοποιώντας εξελιγμένους αλγόριθμους image processing. Το όχημα χρησιμοποιεί δύο ηλεκτρικούς κινητήρες DC. Ως υπολογιστικά κέντρα για το όχημα χρησιμοποιήθηκαν το embedded board, Raspberry Pi 4B 4GB, σε συνδυασμό με την πλακέτα μικροελεγκτή Arduino Mega 2560. Για τον εντοπισμός αντικειμένων στο περιβάλλον χρησιμοποιήθηκαν δύο αισθητήρες απόστασης με υπερήχους HC-SR04 για τον εντοπισμό εμποδίων και μια κάμερα OdSeven Camera HD camera OV5647 5Mpx για εντοπισμό της λωρίδας και αναγνώριση εμποδίων για τις πλαϊνές πλευρές της λωρίδας. Τα δύο υπολογιστικά κέντρα επικοινωνούν μεταξύ τους μέσω USB καλωδίου. Η κάμερα είναι συνδεδεμένη με το Raspberry Pi, με την ανάπτυξη του κώδικα για τον αναγνώριση της λωρίδας έγινε με την γλώσσα Python. Οι αισθητήρες απόστασης και οι κινητήρες ελέγχονται από το Arduino Mega, με τον κώδικα ελέγχου αυτών να είναι γραμμένη στην γλώσσα Wiring. Σε συνδυασμό με το Raspberry Pi και το Arduino Mega, έχει υλοποιηθεί και μια εφαρμογή Android για τον χειροκίνητο έλεγχο του οχήματος. Για την εφαρμογή αυτή χρησιμοποιήθηκε η γλώσσα Kotlin με την πλατφόρμα του Android Studio. Το smartphone στο οποίο έχει δοκιμαστεί ο κώδικας είναι το Realme C25Y.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

2. Σχετικές πληροφορίες

2.1. Ιστορία της αυτόνομης οδήγησης

Το πρώτο πείραμα με όχημα το οποίο δεν ελέγχεται άμεσα από άνθρωπο ήρθε το 1925, από τον ηλεκτρολόγο μηχανικό Francis P. Houdina[12], στην Νέα Υόρκη στις Η.Π.Α. Το πείραμα αποτελούνταν από 2 αμάξια. Το ένα αμάξι είχε έναν δέκτη ραδιοσημάτων, και στο 2ο υπήρχε ο πομπός. Τα ραδιοσήματα από τον πομπό κινούσαν μικρούς ηλεκτρικούς κινητήρες που κατεύθυναν κάθε κίνηση του 1ου αυτοκινήτου.[11] Με αυτόν τον τρόπο, γινόταν η οδήγηση του 1ου αμαξίου από απόσταση, χωρίς άμεση ανθρώπινη παρέμβαση. Το αμάξι αυτό κατάφερε και οδήγησε στους πυκνούς δρόμους της Νέας Υόρκης για ένα μικρό χρονικό διάστημα, ωστόσο το τηλεκατευθυνόμενο αμάξι έπεσε πάνω σε ένα άλλο όχημα γεμάτο με φωτογράφους που κατέγραφαν το γεγονός. Ύστερα από αυτό, η εταιρία Achen Motor χρησιμοποίησε την εφεύρεση του Francis P. Houdina υπό το όνομα “Phantom Auto” σε δύο επιδείξεις, μία το 1926 στους δρόμους του Μίλγουοκι, Ουισκόνσιν[13], και την δεύτερη στους δρόμους του Φρέντερικμπουργκ, Βιρτζίνια[14].

Σχεδιασμοί και υλοποιήσεις πρώιμων συστημάτων υποβοήθησης οδηγού άρχισαν την δεκαετία του 1950. Το 1953, η εταιρία RCA έφτιαξε ένα σύστημα στο οποίο ένα αμάξι-μινιατούρα καθοδηγούνταν από καλώδια παραβρισκόμενα στο πάτωμα ενός εργαστηρίου. Ικανοποιημένοι με τα αποτελέσματα, προχώρησαν οι δοκιμασίες σε μεγέθη κανονικών διαστάσεων[15]. Το 1957, στην διασταύρωση των αυτοκινητοδρόμων 77 και Νεμπράσκα 2, δίπλα στην πόλη Λίνκολν, Νεμπράσκα των Η.Π.Α[16], χρησιμοποιήθηκαν καλώδια, αισθητήρες και πηνία ενσωματωμένοι στον δρόμο και πεζοδρόμιο. Το πειραματικό αμάξι ήταν εξοπλισμένο με δύο πηνία στον μπροστινό προφυλακτήρα, τα οποία θα επικοινωνούσαν μέσω μαγνητικής επαγωγής με τους αισθητήρες στον δρόμο. Επίσης στο ταμπλό του αυτοκινήτου υπήρχε ένας δείκτης ο οποίος έδειχνε προς το κέντρο της λωρίδας. Ο οδηγός του αμαξίου αυτού κατάφερε και οδήγησε ευθεία, χωρίς να έχει οπτική επαφή με τον δρόμο, και καθώς αργότερα, το αμάξι του πλησίασε ένα άλλο όχημα, ενεργοποιήθηκαν αυτόματα συναγερμοί ήχου και φωτός ως προειδοποίηση κινδύνου προς τον οδηγό.

Το μοτίβο του ελέγχου λειτουργιών του αμαξίου με την χρήση καλωδίων και αισθητήρων ενσωματωμένων στον δρόμο συνέχισε μέχρι το 1980, όπου ο Γερμανός μηχανικός Ernst Dickmanns και η ομάδα του από το Πανεπιστήμιο της Bundeswehr του Μονάχου, σχεδίασαν ένα σύστημα αυτόνομης οδήγησης με την χρήση καμερών και διάφορων αισθητήρων. Το σύστημα αυτό επέτρεψε σε ένα βαν Mercedes-Benz T2 να φτάσει ταχύτητα 96 χιλιομέτρων ανά ώρα σε ένα άδειο κομμάτι γερμανικού αυτοκινητόδρομου, χωρίς την παρέμβαση του οδηγού[17]. Παράλληλα στις Η.Π.Α, το έργο “ALV” χρηματοδοτούμενο από τον οργανισμό Υπουργείου Άμυνας “DARPA”, με την συμβολή πολλαπλών ερευνητικών και εκπαιδευτικών σωμάτων, χρησιμοποίησε LIDAR, κάμερες και αυτόνομο ρομποτικό έλεγχο, για την αυτόνομη οδήγηση ενός οχήματος σε περιβάλλον εντός αλλά και εκτός δρόμου με υψηλές και χαμηλές ταχύτητες.[18] Το 1989, ένα από τα πανεπιστήμια που συμμετείχαν στο “ALV”, το Πανεπιστήμιο Carnegie Mellon, πρωτοστάτησε την χρήση των Νευρωνικών Δικτύων για την αυτόνομη οδήγηση οχημάτων[19], θέτοντας έτσι την βάση για τις σύγχρονες υλοποιήσεις συστημάτων αυτόνομης οδήγησης.

Την δεκαετία του ‘90, με μια χρηματοδότηση 749,000,000€ στα πλαίσια του προγράμματος “Prometheus” για έρευνα στο πεδίο της αυτόνομης οδήγησης, του διακυβερνητικού οργανισμού για έρευνα και ανάπτυξη “Eureka”, με σημαντικό ρόλο να παίζει ξανά ο Ernst Dickmanns, κατασκευάστηκαν δύο τροποποιημένα αμάξια Mercedes “VaMP” και “VITA-2” με λειτουργίες αυτομάτου ελέγχου του τιμονιού, του γκαζιού και των φρένων μέσω εντολών υπολογιστή που βασίζονται σε αξιολόγηση ακολουθιών εικόνων πραγματικού χρόνου[20][21]. Τα αμάξια αυτά κατάφεραν να οδηγήσουν με κανονική κίνηση σε δρόμο 3 λωρίδων, πάνω

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

από 1000 χιλιόμετρα, με ταχύτητες έως 130 χιλιόμετρα την ώρα. Αν και η οδήγηση ήταν ημιαυτόνομη με ανθρώπινες παρεμβάσεις, έδειξαν αυτόνομη οδήγηση σε ελεύθερες λωρίδες, οδήγηση κομβία και αλλαγές λωρίδας με αυτόνομη προσπέραση άλλων αυτοκινήτων[22]. Την ίδια περίπου περίοδο έλαβαν χώρα πειράματα παρόμοιου τύπου σε άλλες μέρη του πλανήτη, όπως το πείραμα του καθηγητή Han Min-Hong του Πανεπιστημίου της Κορέας στην Νότια Κορέα[23], στο Ηνωμένο Βασίλειο τα σύστημα αυτόματης πλοήγησης Cruise Control και αυτόματη ευθυγράμμιση οχήματος στην λωρίδα που ανέπτυξε η Lucas Industries με συγχρηματοδότηση από την Jaguar Cars και το Υπουργείο Εμπορίου και Βιομηχανίας του Ηνωμένου Βασιλείου[24][25], το σχέδιο “ARGO” στην Ιταλία[26], το σχέδιο “Navlab” του Πανεπιστημίου Carnegie Mellon, Πίτσμπουργκ, Πενσυλβάνια[27][28], τα αυτόματα λεωφορεία της Ολλανδίας “ParkShuttle”[29] και πολλά άλλα.

Με την εισαγωγή του 2000, στις Η.Π.Α, λαμβάνουν χώρα τρεις διαγωνισμοί, “First DARPA Grand Challenge” το 2004[30], “Second DARPA Grand Challenge” το 2005[31] και “Third DARPA Grand Challenge” το 2007[32]. Κάθε διαγωνισμός είχε μια δοκιμασία για την οποία πολλαπλά πανεπιστήμια και ερευνητές έπρεπε να κατασκευάσουν οχήματα με πλήρης αυτονομία στις κινήσεις τους. Οι δύο πρώτοι διαγωνισμοί έλαβαν χώρα στην έρημο Μοχάβε των Η.Π.Α και ο τρίτος έλαβε χώρα στην αστική περιοχή της αεροπορικής βάσης “George Air Force Base”. Οι τρεις διαγωνισμοί αυτό όχι μόνο έφεραν καινοτομία στον τομέα της αυτόνομης οδήγησης, αλλά κίνησε το ενδιαφέρον επενδυτών και ερευνητών, φέρνοντας τον τομέα αυτόν πιο κοντά στο κέντρο προσοχής.

Στα τέλη του 2009 με αρχές του 2010, αρχίζει η έκρηξη καινοτομίας στον τομέα. Γίγαντες της τεχνολογίας και αυτοκινητοβιομηχανίας όπως: Google, General Motors, Ford, Mercedes Benz, Volkswagen, Audi, Nissan, Toyota, BMW, Volvo και πολλαπλά εκπαιδευτικά ιδρύματα όπως το Τεχνικό Πανεπιστήμιο του Μπράουνσβαϊγκ, Ανοικτό Πανεπιστήμιο του Βερολίνου, Πανεπιστήμιο της Πάρμα, Ινστιτούτο Τεχνολογίας της Καρλσρούης και πολλά άλλα, δοκιμάζουν δικά τους οχήματα με διαφορετικό βαθμό αυτόνομης οδήγησης σε κλειστά και ελεγχόμενα περιβάλλοντα, αλλά και στους δρόμους διαφόρων πόλεων. Το 2014, η Γαλλική εταιρία “Induct Technology” φέρνει το πρώτο όχημα με δυνατότητες αυτόνομης οδήγησης στην αγορά με το “Navia”. Το όχημα αυτό δεν χρησιμοποιείται σε πολυσύχναστους αστικούς δρόμους, αλλά σε ανοικτά περιβάλλοντα όπως πανεπιστημιούπολεις[33].

Τον Οκτώβρη του 2015 ξεκινάει η σύγχρονη εποχή της αυτόνομης οδήγησης με την Τέσλα να κυκλοφορεί την έκδοση 7 του λογισμικού της στα αμάξια “Model S” στις Η.Π.Α. που περιελάμβανε τη δυνατότητα αυτόνομης οδήγησης[34]. Έως σήμερα, Σεπτέμβρη του 2023, έχουν υπάρξει μύριες εξελίξεις και νέα οχήματα που κατέχουν αυτονομία επιπέδου 2+[10], όπως τα οχήματα της Waymo που προσφέρουν τις υπηρεσίες “ρομπο-ταξί” στο Σαν Φρανσίσκο της Καλιφόρνιας τον Απρίλιο του 2022[35] και η εμφάνιση του νέου αμαξίου της Mercedes-Benz που κατέχει επιπέδου 3 αυτονομία τον Μάιο του 2022[36]. Το σημερινό τοπίο της αγοράς, καθώς συνεχώς εξελισσόμενο, έχει μπει στο στάδιο των νομικών διαμαχών. Αρκετοί Κ.Ο.Κ σε όλον τον πλανήτη δεν είναι ακόμη εξοπλισμένοι να αντιμετωπίσουν ατυχήματα με οχήματα με αυτόνομη οδήγηση. Μια εξαίρεση σε αυτό είναι η Ιαπωνία, που τον Απρίλιο του 2023 έκανε αναθεωρήσεις στη νομοθεσία της για την αδειοδότηση οχημάτων με αυτονομία επιπέδου 4[37]. Εκτός από νομικά θέματα, οχήματα με τέτοιες λειτουργίες είναι ακόμη υπερβολικά ακριβά για τον μέσο άνθρωπο. Έτσι οι εταιρίες σκοπεύουν στην “εκλαϊκείωση” των προϊόντων αυτών, κάνοντάς τα πιο προσιτά για το κοινό.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

2.2. Το Raspberry Pi

Η ιστορία του Raspberry Pi αρχίζει στο Πανεπιστήμιο του Κέιμπριτζ της Αγγλίας το 2006. Το σχέδιο αρχικά ήταν η διάδοση της γνώσης σε φοιτητές και μαθητές, όσον αφορά την πρακτική λειτουργικότητα υπολογιστικών συστημάτων, όχι μόνο την θεωρητική επίγνωση. Μια ομάδα καθηγητών και ερασιτεχνών με πρωτεργάτη τον Eben Upton, ξεκίνησαν να σχεδιάζουν έναν υπολογιστή στο μέγεθος παλάμης χεριού, ο οποίος θα ήταν προσιτός και εύχρηστος από το μη-εξειδικευμένο κοινό. Το 2011, ο David Braben ανακοινώνει ένα πρωτότυπο μεγέθους ενός απλού USB stick, ο οποίος είχε μια θύρα USB για την σύνδεση πληκτρολογίου ή ποντικιού και μια θύρα HDMI για την σύνδεση οθόνης[38]. Με υποσχέσεις ότι η τιμή θα είναι κοντά στα 25\$, ο φτηνός προγραμματισμός σύντομα θα γινόταν πραγματικότητα. Τον Φεβρουάριο του 2012, το πρώτο μοντέλο του υπολογιστή “Raspberry Pi 1 Model B” με την μορφή που γνωρίζουμε σήμερα βγαίνει στην αγορά στην τιμή των μόλις 35\$. Το μοντέλο αυτό είχε επεξεργαστή ARM11 700MHz, κάρτα γραφικών VideoCore IV, 256 MB SDRAM, HDMI και θύρα 3.5mm για οθόνη και ήχο, και μια θύρα Ethernet για σύνδεση στο διαδίκτυο[39]. Με την πάροδο του χρόνου, η εταιρία ανακοινώνει συνεχείς αναβαθμίσεις με νέες εκδόσεις του υπολογιστή να βγαίνουν στην αγορά σχεδόν κάθε έτος.

Το μοντέλο που χρησιμοποιήθηκε στην εργασία είναι το “Raspberry Pi 4B”. Συγκεκριμένα, έχει τα εξής χαρακτηριστικά[40]:



Εικόνα 2.2.1 Raspberry Pi 4B

Πηγή: https://en.m.wikipedia.org/wiki/File:Raspberry_Pi_4_Model_B_-_Side.jpg

- Επεξεργαστής: Broadcom BCM2711 τετραπύρηνος Cortex-A72 (ARMv8) 64-bit SoC, 1.5 GHz.
- Μνήμη RAM: 4GB LPDDR4-3200 SDRAM.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

- Κάρτα γραφικών: VideoCore VI GPU
- Βίντεο: Δύο θύρες microHDMI που υποστηρίζουν ανάλυση έως και 4K, 60fps.
- Ήχος: 3.5mm audio jack, as well as audio over HDMI.
- Θύρες USB: 2×USB3.0, 2×USB2.0.
- Δίκτυο: Θύρα Gigabit Ethernet RJ45 , με υποστήριξη Wi-Fi 802.11ac (2.4GHz και 5GHz) συν Bluetooth 5.0 μέσω ενσωματωμένου chip.
- Αποθηκευτικός χώρος: Κάρτα MicroSD.
- GPIO: 40 ακίδες GPIO για την σύνδεση εξωτερικών εξαρτημάτων και αισθητήρων.
- Τροφοδοσία: 5V, έως 3A, μέσω μιας θύρας USB-C.
- Power over Ethernet (PoE): Υποστήριξη με το κατάλληλο συμπλήρωμα PoE HAT.
- Λειτουργικό σύστημα: Raspberry Pi OS Bullseye (32-bit) .
- Διαστάσεις: 85.6mm × 56.5mm.

2.3. Το Arduino

Η ιστορία του Arduino ξεκινά στις αρχές της δεκαετίας του 2000 όταν μια ομάδα μηχανικών και σχεδιαστών στο Interaction Design Institute Ivrea (IDII) στην Ivrea της Ιταλίας, άρχισε να αναπτύσσει μια πλατφόρμα για να διευκολύνει τον προγραμματισμό και τα ηλεκτρονικά για τους μη εξοικειωμένους, όπως σχεδιαστές και καλλιτέχνες. Το 2003, ο τότε φοιτητής Hernando Barragán δημιούργησε την πλατφόρμα “Wiring” ως μέρος της διπλωματικής του εργασίας για τις σπουδές του στο IDII. Η Wiring είναι μια πλατφόρμα που περιλαμβάνει ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) και μια γλώσσα προγραμματισμού ανοιχτού κώδικα. Η πλατφόρμα αυτή χρησιμοποιείται για τη δημιουργία λογισμικού για μικροελεγκτές και ενσωματωμένα συστήματα και βασίζεται στη γλώσσα προγραμματισμού “Processing”[41] .

Το 2005, ο Massimo Banzi, με τον David Mellis και τον David Cuartielles, επεκτείνουν το έργο της Wiring προσθέτοντας υποστήριξη για τον φθηνότερο μικροελεγκτή ATmega8. Το νέο έργο ονομάστηκε Arduino. Η πρώτη πλακέτα ανάπτυξης που κυκλοφόρησε στην αγορά σε μεγάλο βαθμό, ήταν η “Arduino Diecimila” το 2007[42]. Με την πάροδο των χρόνων, υπήρχαν πολλαπλές εκδόσεις και αναβαθμίσεις στην πλακέτα.

Σήμερα, υπάρχουν τεσσάρων ειδών πλακέτας Arduino[43]:

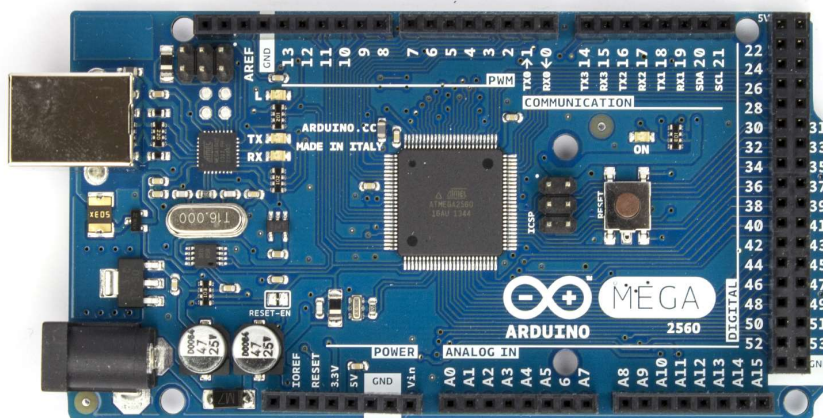
1. **Σειρά Nano:** Η σειρά αυτή έχει ως κύριο χαρακτηριστικό το μικρό αποτύπωμα της πλακέτας. Αυτές οι πλακέτες διαθέτουν επίσης ένα σύνολο ενσωματωμένων αισθητήρων, όπως θερμοκρασίας, υγρασίας, πίεσης, ήχου και άλλα. Μπορούν επίσης να προγραμματιστούν με MicroPython και υποστηρίζουν τη Μηχανική Μάθηση.
2. **Σειρά MKR:** Οι πλακέτες της σειράς αυτής εστιάζουν σε έργα Δικτύου των Πραγμάτων (IoT) Οι πλακέτες αυτές έχουν σχεδιαστεί για να επεκτείνουν τις λειτουργίες της των κύριων πλακετών από άλλες σειρές, όπως αισθητήρες περιβάλλοντος, GPS, Ethernet, έλεγχος κινητήρα και έλεγχο RGB φωτεινών ενδείξεων. Περιλαμβάνουν ενσωματωμένες επιλογές συνδεσιμότητας όπως Wi-Fi, GSM ή

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

LoRa. Επίσης, έχουν χαμηλή κατανάλωση ενέργειας, καθιστώντας τις κατάλληλες για έργα με μπαταρία και εξοικονόμηση ενέργειας.

3. **Σειρά Classics:** Η σειρά με τις πιο γνωστές πλακέτες που έχει παράξει η εταιρία. Στην σειρά αυτή ανήκει το διάσημο Arduino UNO R3, Arduino Zero, και την πιο πρόσφατη πλακέτα, Arduino Uno R4 σε δύο εκδόσεις, Arduino Uno R4 Minima και Arduino Uno R4 WiFi. Οι πλακέτες αυτής της σειράς προσφέρουν ευελιξία και ευκολία στην χρήση τους. Διαθέτουν θύρες για σειριακή επικοινωνία με άλλες συσκευές και εισόδους για διαφόρων ειδών ηλεκτρονικών συσκευών και αισθητήρων, και όλα αυτά σε προσιτή τιμή.
4. **Σειρά Mega:** Οι πλακέτες της σειράς αυτής είναι παρόμοιες με την σειρά Classics. Η διαφορά τους είναι πως οι πλακέτες της Mega σειράς είναι αρκετά μεγαλύτερες και με δυνατότερη υπολογιστική ισχύ. Έτσι έχουν μεγαλύτερη ευελιξία στο πλήθος συσκευών που μπορούν να συνδεθούν με αυτές, αλλά και στον τύπο διεργασιών που μπορούν να εκτελέσουν.

Στην εργασία χρησιμοποιήθηκε το “Arduino Mega 2560 R3”. Συγκεκριμένα, έχει τα εξής χαρακτηριστικά[44]:



Εικόνα 2.3.1 Arduino Mega 2560 R3

Πηγή:

https://a.scdn.gr/images/sku_main_images/007301/7301825/20150812135726_arduino_mega_2560_rev3.jpeg

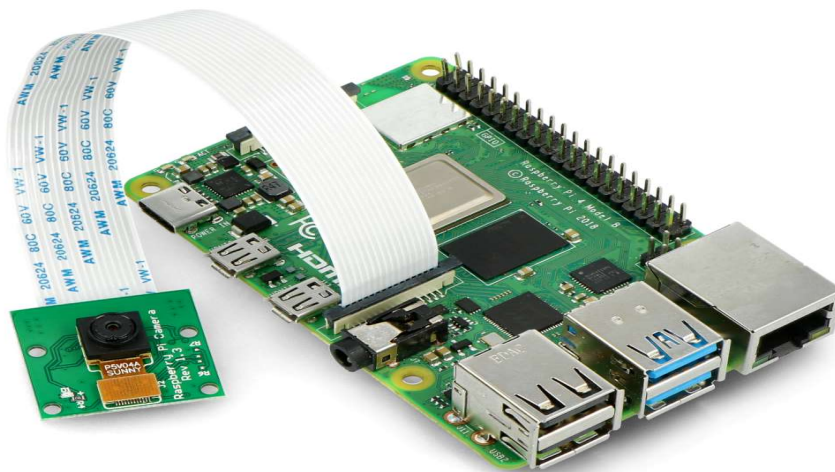
- Μικροελεγκτής: ATmega2560
- Τροφοδοσία: 5V, έως 200A, μέσω της θύρας USB-B.
- Ψηφιακές Ακίδες εισόδου/εξόδου: 54 (από τις οποίες 15 παρέχουν έξοδο PWM)
- Αναλογικές Ακίδες εισόδου: 16
- Μνήμη Flash: 256 KB εκ των οποίων τα 8 KB χρησιμοποιούνται από το bootloader

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

- SRAM: 8 KB
- EEPROM: 4 KB
- Συχνότητα ρολογιού: 16 MHz
- Πλήθος διαύλων UART: 4
- Δίαυλος SPI: 1
- Δίαυλος I2C: 1
- Θύρα USB: 1
- Power over Ethernet (PoE): Υποστήριξη με το κατάλληλο συμπλήρωμα PoE Shield.
- Διαστάσεις: 101.52mm × 53.3mm.

2.4. Η κάμερα

Η κάμερα που χρησιμοποιήθηκε στην εργασία είναι η “Arducam 5MP OV5647”. Το κύριο κριτήριο επιλογής ήταν τα πολλαπλά προσόντα της κάμερας για μια αρκετά προσιτή τιμή. Η κάμερα αυτή χρησιμοποιεί αισθητήρα εικόνας CMOS OV5647 της OmniVision ο οποίος είναι και γνωστός για την υψηλή απόδοσή του σε περιβάλλοντα χαμηλού φωτός. Υποστηρίζει 8-10bit δεδομένα εικονών RGB σε αναλύσεις από 320×240 έως 2592×1944[45]



Εικόνα 2.4.1 Η κάμερα συνδεδεμένη στο Raspberry Pi 4B

Πηγή: https://botland.store/img/art/inne/05619_16.jpg

2.5. Αισθητήρες απόστασης

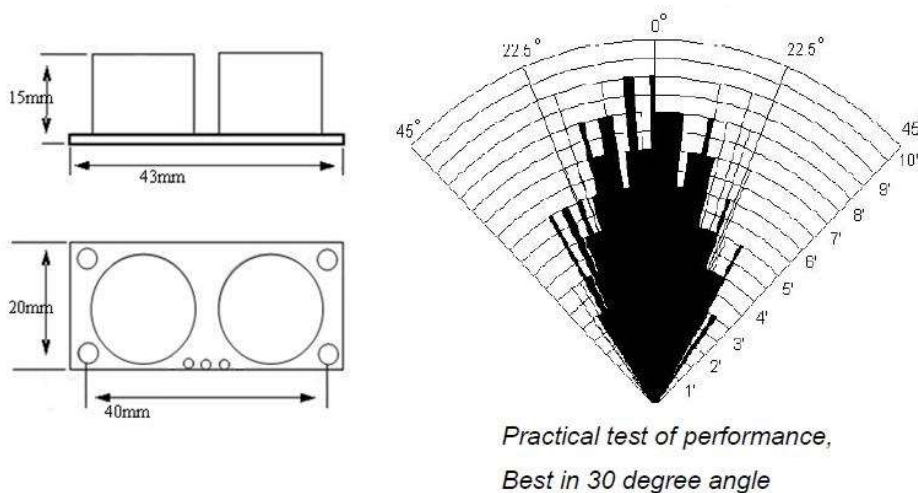
Για τους αισθητήρες απόστασης χρησιμοποιήθηκε ο αισθητήρας υπερήχων HC-SR04. Ο αισθητήρας αυτός είναι ευρέως χρησιμοποιούμενος καθώς είναι προσιτός στην τιμή του, απλός στην χρήση του, και μικρός σε

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος μέγεθος. Χρησιμοποιούνται υπερηχητικά κύματα για τον προσδιορισμό της απόστασης μεταξύ του αισθητήρα και ενός αντικειμένου με αξιοσημείωτη ακρίβεια 0.3cm. Έχει εμβέλεια από 2cm έως και 400cm και έχει την ικανότητα να μετρήσει ακριβείς αποστάσεις σε γωνία 30°[46].



Εικόνα 2.5.1 Ο αισθητήρας HC-SR04

Πηγή: <https://www.amazon.nl/HC-SR04-Ultrasonic-Afstandsmeting-Transducer-Arduino/dp/B08LYX2H7L>



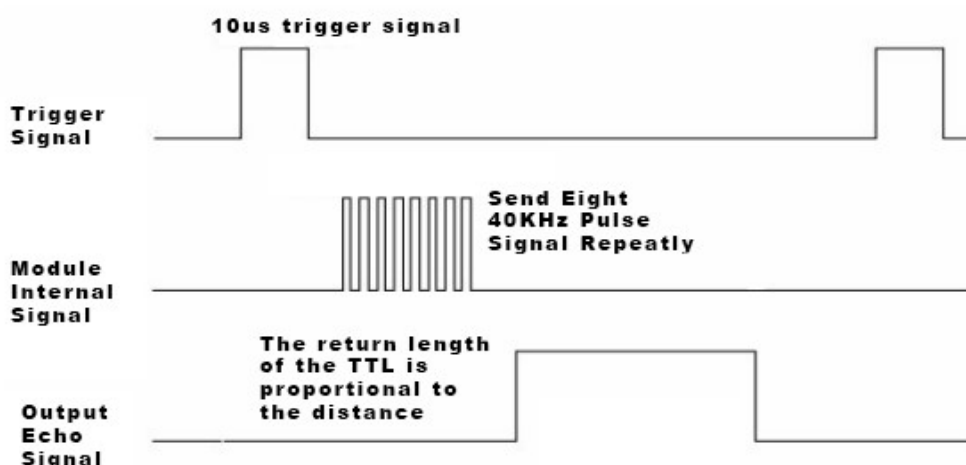
Εικόνα 2.5.2 Μέγεθος και γωνία βέλτιστης απόδοσης του αισθητήρα

Πηγή: https://botland.store/index.php?controller=attachment&id_attachment=476

Συγκεκριμένα, ο αισθητήρας αυτός διαθέτει 4 ακίδες. Μία για την τροφοδοσία(Vcc), μια για την γείωση(GND), την ακίδα για τον σκανδαλισμό του πομπού(Trig) και την ακίδα εισόδου του δέκτη(Echo). Τροφοδοτείται από τάση 5V με κατανάλωση περίπου 15mA. Για την χρήση του αισθητήρα, πρέπει οι ακίδες του να συνδεθούν με ένα Arduino. Στην συνέχεια, για να ενεργοποιήσουμε τον πομπό, παράγουμε ένα σήμα

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

λογικό HIGH στην ακίδα Trig το οποίο παραμένει στην υψηλή κατάσταση για τουλάχιστον 10μs και ύστερα το σήμα πρέπει να πέσει σε λογικό LOW. Την στιγμή που το η ακίδα Trig λάβει αυτό το σήμα, ο πομπός παράγει 8 κύματα υπερήχων 40KHz. Όταν παραχθούν τα ηχητικά κύματα, ξεκινάει ένας εσωτερικός μετρητής χρόνου σε μs. Την στιγμή που ο δέκτης ανιχνεύσει την ηχώ (το ανακλώμενο ηχητικό κύμα που παράχθηκε νωρίτερα) από το αντικείμενο ενδιαφέροντος, σταματά το χρονόμετρο και παράγεται ένα σήμα στην ακίδα Echo, του οποίου η διάρκεια παραμένει στο λογικό HIGH ανάλογα με τον χρόνο που έκανε η ηχώ να φτάσει τον δέκτη.



Εικόνα 2.5.3 Χρονοδιάγραμμα των σημάτων του αισθητήρα

Πηγή: <https://www.semanticscholar.org/paper/Arduino-Based-Ultrasonic-Radar-System-using-Matlab-Jain/45a15c6b323fe0ac89947df3b565b533a2d7e869/figure/1>

2.6. Κινητήρες

Για την κίνηση του οχήματος χρησιμοποιήθηκαν δύο TT DC κινητήρες, και για την οδήγησή τους, χρησιμοποιήθηκε ο "L298N Motor Driver Board (ST1112)" της Iduino.

Οι κινητήρες είναι απλής χρήσης και έχουν τα εξής χαρακτηριστικά[47]:

- Μέγιστη τάση κινητήρα: 6V
- Ροπή: 0.8 kg*cm (0.78 Nm)
- Ταχύτητα κινητήρα χωρίς φορτίο 90 +/- 10 rpm
- Κατανάλωση ρεύματος κινητήρα χωρίς φορτίο: 190 mA (250 mA max.)
- Κατανάλωση ρεύματος κινητήρα στον σταθερό άξονα : 1A

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 2.6.1 Οι κινητήρες

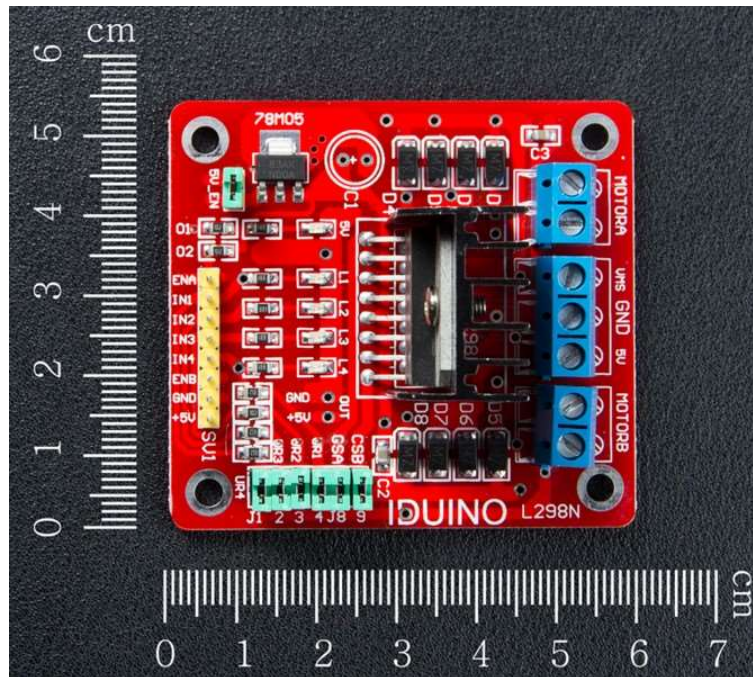
Πηγή: <https://www.amazon.com/misppro-20Pcs-Geared-Bracket-Support/dp/B0C48ZMNNM>

Ο Motor controller των κινητήρων έχει τα εξής χαρακτηριστικά[48]:

- Οδηγητής: L298N two-channel H-bridge
- Τάση οδηγητή: 5V~35V
- Μέγιστο ρεύμα οδηγητή: 2A
- Τάση λογικής εξόδου Vss: 5V~7V (με το εσωτερικό τροφοδοτικό 5V)
- Ρεύμα λογικής εξόδου: 0~36mA
- Επίπεδα λογικής: Low -0.3V~1.5V, High: 2.3V~Vss
- Μέγιστη ισχύς: 25W (Θερμοκρασία 75°C)
- Κατάλληλη θερμοκρασία: -25°C~130°C

Με την χρήση του οδηγητή μπορούμε να τροφοδοτήσουμε τους κινητήρες με τάση και ρεύμα που κανονικά το Arduino και το Raspberry Pi δεν έχουν την ικανότητα να παράξουν. Συνδέοντας τον οδηγητή με το Arduino, μπορούμε να τον ελέγξουμε δίνοντάς του σήματα ελέγχου από το Arduino, και η τροφοδοσία των κινητήρων παραμένει απομονωμένη μεταξύ του οδηγητή και της τροφοδοσίας.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



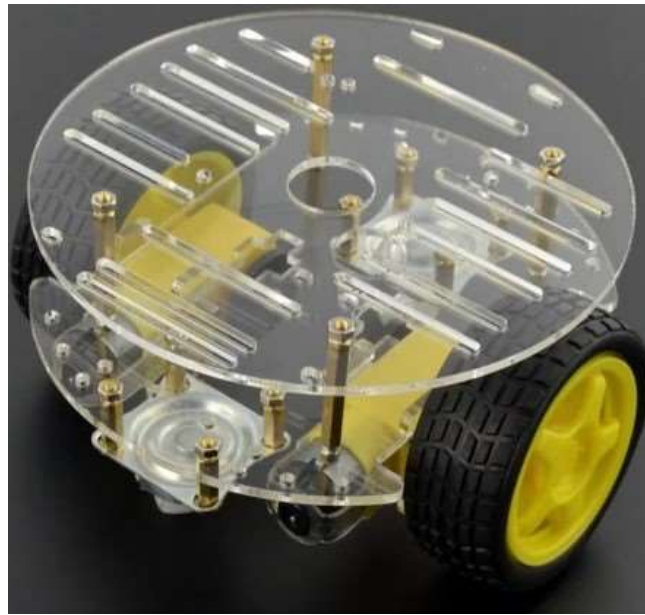
Εικόνα 2.6.2 Ο οδηγητής των κινητήρων

Πηγή: <https://asset.conrad.com/media10/add/160267/c1/-/en/001525437ML01/manual-1525437-iduino-st-1112-motor-controller-1-pcs.pdf>

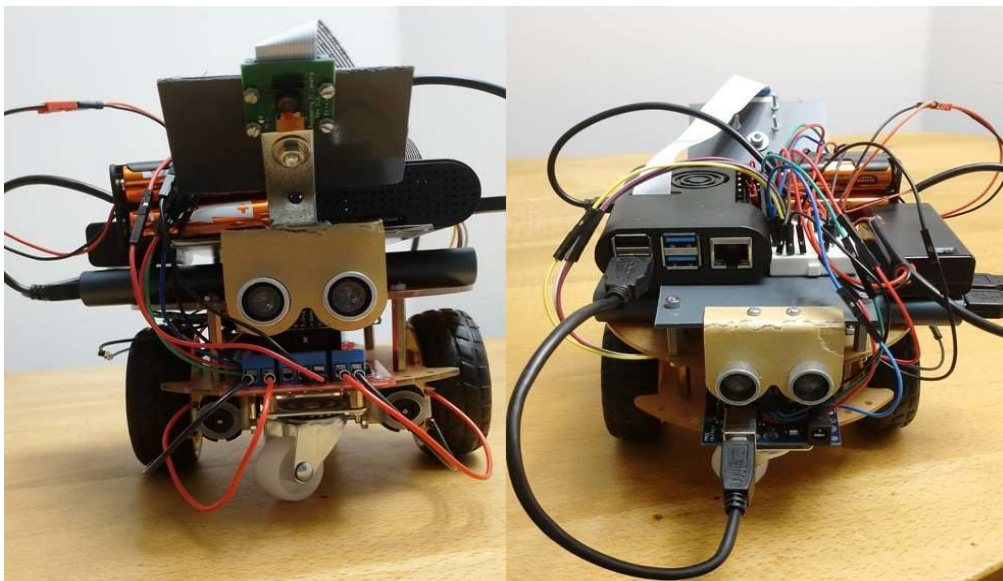
2.7. Ο σκελετός του οχήματος

Για το όχημα χρησιμοποιήθηκε ένα σασί ρομπότ δύο ορόφων. Στο σασί αυτό χωράνε πολλαπλών μεγεθών εξαρτήματα, πλακέτες και μπαταρίες, καθώς το σασί επεκτείνεται προς τον κατακόρυφο και όχι προς τον οριζόντιο. Αυτό δίνει στο σασί αρκετή ευελιξία για την τοποθέτηση εξαρτημάτων. Το σασί περιλαμβάνει και 2 ρόδες οι οποίες προσδένονται στους κινητήρες. Στο σασί έγιναν και DIY τροποποιήσεις, επεκτείνοντας τον διαθέσιμο χώρο. Αυτές οι τροποποιήσεις συμπεριλαμβάνουν την προσθήκη μιας επιπλέον πλατφόρμας από πλαστικό PVC στην οποία τοποθετήθηκε το Raspberry Pi, και μια βάση επίσης από PVC στην οποία μπόρεσε να βιδωθεί η κάμερα. Κάτω από αυτήν την πλατφόρμα τοποθετήθηκε μια μπαταρία από την οποία τροφοδοτούμε το Raspberry Pi και τους κινητήρες, και στην πρώτη πλατφόρμα τοποθετήθηκαν το Arduino Mega και ο L298N.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 2.7.1 Το σασί χωρίς τα εξαρτήματα και τις τροποποιήσεις



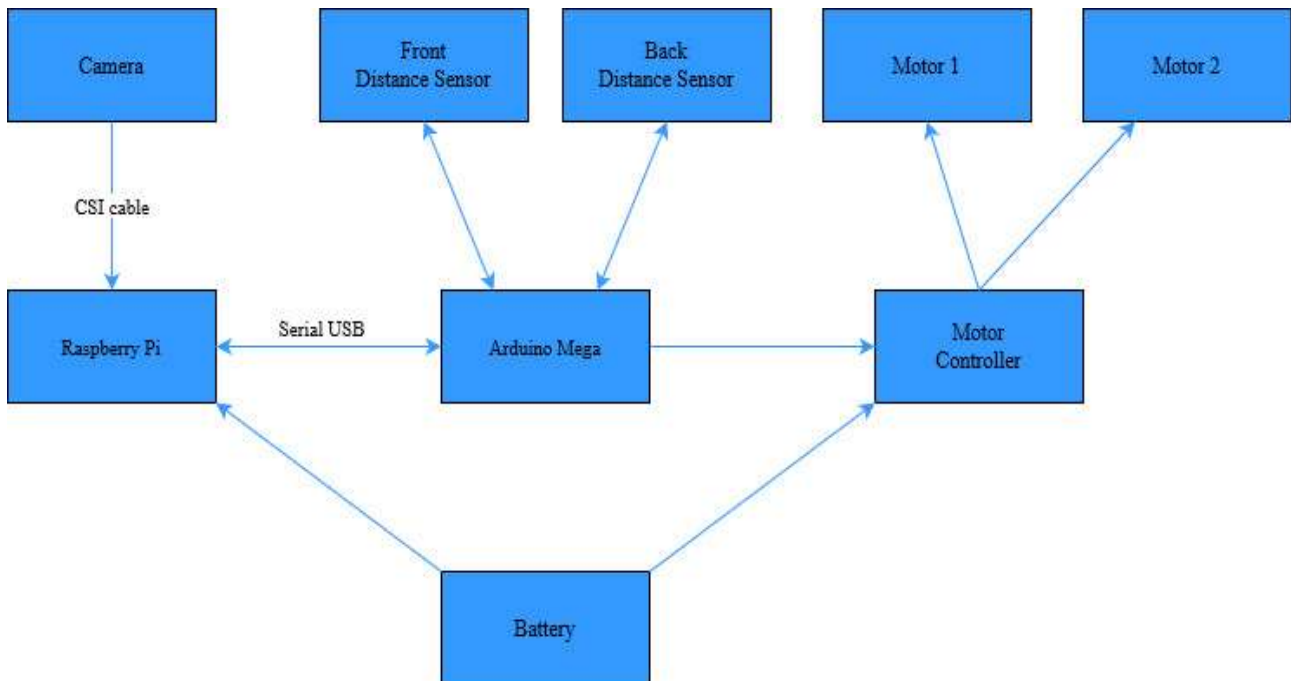
Εικόνα 2.7.2 Φορτωμένο σασί και τις τροποποιήσεις (Αριστερά, η μπροστινή όψη και δεξιά, η πίσω)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

3. Αρχιτεκτονική οχήματος

3.1. Αρχιτεκτονική υλικού

Παρακάτω παρουσιάζεται block διάγραμμα που αναπαριστά τις συνδέσεις μεταξύ των εξαρτημάτων του οχήματος:



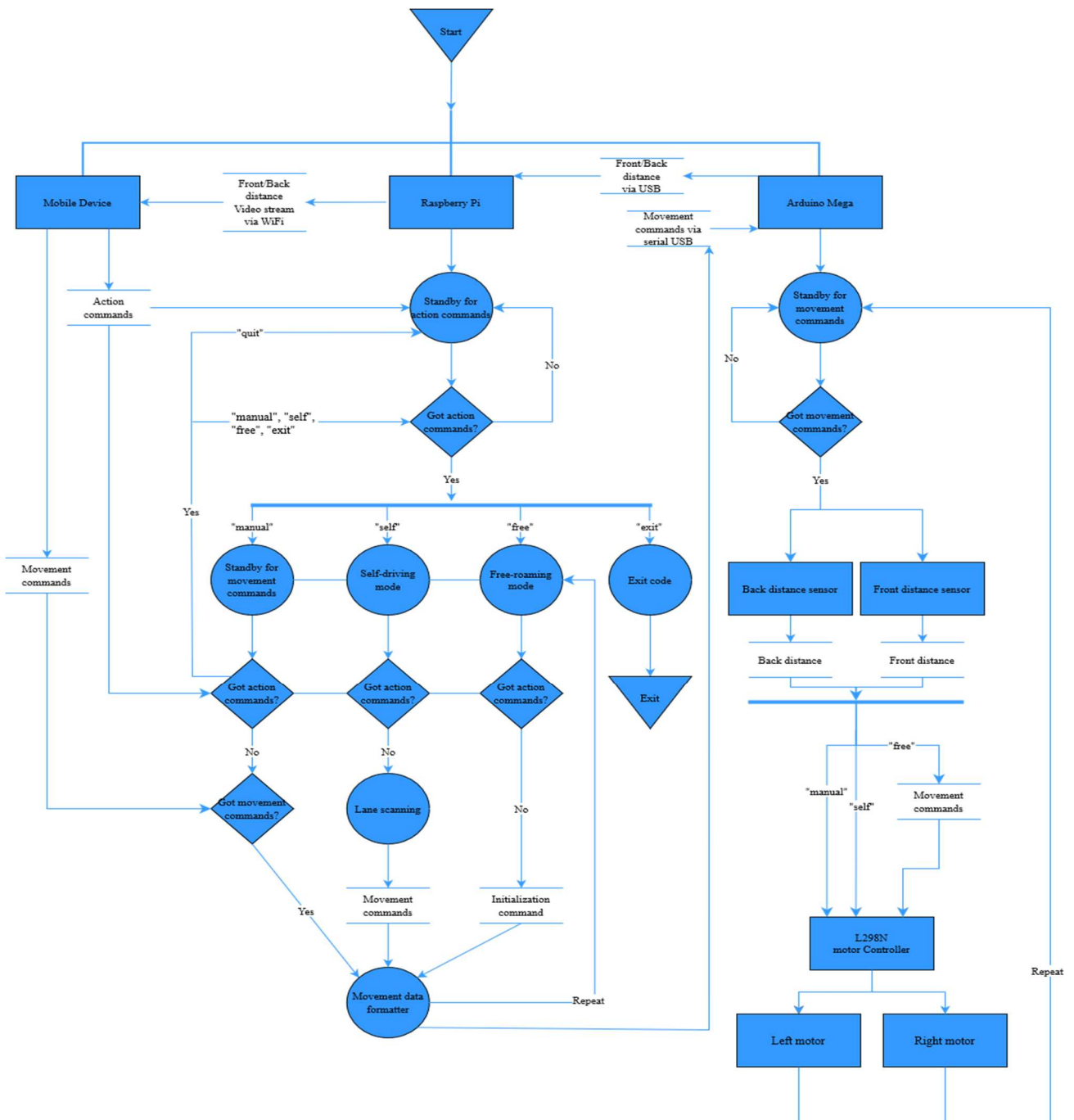
Εικόνα 3.1.1 Η αρχιτεκτονική του υλικού

Στην Εικόνα 3.1.1 παρατηρήται η αρχιτεκτονική του υλικού του οχήματος. Το όχημα διαθέτει δύο κέντρα επεξεργασίας. Το Raspberry Pi είναι συνδεδεμένο με την κάμερα μέσω καλωδίου CSI χρησιμοποιώντας την ενσωματωμένη θύρα MIPI CSI-2. Στο Arduino Mega έχουν συνδεθεί ο οδηγός κινητήρων L298N και οι δύο αισθητήρες HC-SR04. Οι αισθητήρες τροφοδοτούνται από τις θύρες 5V και GND που διαθέτει το Arduino, καθώς για την μεταφορά δεδομένων, έχουν συνδεθεί στους ψηφιακούς ακροδέκτες. Όσον αφορά τον L298N, οι ακίδες IN1, IN2, IN3, IN4 και GND είναι συνδεδεμένες με ψηφιακούς ακροδέκτες του Arduino. Στην ίδια σειρά, η ακίδα 5V δεν έχει συνδεθεί καθώς είναι έξοδος 5V και μπορούμε να έχουμε τέτοια έξοδο από το Arduino. Οι ακίδες ENA, ENB έχουν συνδεθεί με αναλογικούς ακροδέκτες. Οι εισοδοί καλωδίων Motor A και Motor B έχουν συνδεθεί με τους ακροδέκτες καλωδίων των δύο κινητήρων. Η είσοδος Vms έχει συνδεθεί άμεσα με την μπαταρία και η είσοδος GND έχει συνδεθεί με την γείωση του Arduino. Παρομοίως και εδώ, ο ακροδέκτης 5V δεν έχει παραμείνει κενός καθώς με αυτόν δίνουμε στον L298N είσοδο 5V για την λογική της πλακέτας. Η πλακέτα διαθέτει έναν μικρό βραχυκυκλωτήρα “5V_EN”, ο οποίος όταν είναι βραχυκυκλωμένος, η πλακέτα χρησιμοποιεί την είσοδο Vms για να τροφοδοτήσει την λογική της. Στην εργασία, ο 5V_EN είναι βραχυκυκλωμένος, έτσι ώστε να μην επιβαρύνουμε το Arduino ή το Raspberry Pi με την επιπλέον τροφοδοσία της λογικής της πλακέτας L298N.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

3.2. Αρχιτεκτονική λογισμικού

Στην παρακάτω Εικόνα 3.1.2, παρατηρείται η ροή πληροφορίας του συνολικού συστήματος, συμπεριλαμβανομένου και της εφαρμογής της κινητής συσκευής.



Εικόνα 3.2.1 Ροή δεδομένων του συστήματος

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Στην Εικόνα 3.2.1 παρατηρείται η αρχιτεκτονική του λογισμικού του οχήματος. Για την εκκίνηση των διεργασιών απαιτούνται συγκεκριμένα βήματα. Αρχικά πρέπει να ενεργοποιηθεί το Raspberry Pi το οποίο παράλληλα τροφοδοτεί το Arduino. Όταν το Raspberry Pi είναι πλήρως ενεργοποιημένο, πρέπει να συνδεθούμε στην συσκευή μέσω SSH και να εκτελέσουμε ένα script, με το οποίο γίνεται ανέβασμα του κώδικα του Arduino και η εκκίνηση του κύριου αρχείου εκτέλεσης κώδικα του Raspberry Pi. Μπορούμε να εντοπίσουμε το πέρας του script αυτού, όταν ενεργοποιηθούν οι φωτεινές ενδείξεις του L298N.

Την στιγμή που τελειώσει η διαδικασία της εκκίνησης, το Raspberry Pi και το Arduino μπαίνουν σε κατάσταση αναμονής εντολών. Σε αυτό το σημείο, πρέπει να χρησιμοποιηθεί η κινητή εφαρμογή για την αποστολή εντολών στο όχημα. Όταν το Raspberry Pi παραλάβει μια εντολή από την κινητή εφαρμογή, τότε για κάθε ξεχωριστή εντολή θα μπει σε έναν βρόχο εκτέλεσης διεργασιών. Υπάρχει η δυνατότητα της εξόδου από τον συγκεκριμένο βρόχο ή και από τον κώδικα σαν σύνολο με την χρήση ειδικών εντολών από την εφαρμογή. Στην περίπτωση που χρησιμοποιηθεί η εντολή της εξόδου από τον βρόχο εκτέλεσης, τότε το Raspberry Pi θα ξαναμπει στην κατάσταση αναμονής νέων εντολών από το κινητό. Σε αυτήν την κατάσταση δεν είναι δυνατή η χρήση του οχήματος χειροκίνητα καθώς και η αυτόνομη κίνηση, έως ότου ο χρήστης στείλει την κατάλληλη εντολή στο όχημα. Στην περίπτωση της εξόδου από τον κώδικα, το Raspberry Pi θα τερματίσει όλες τις λειτουργίες του και θα βρεθεί σε κατάσταση πλήρης αδράνειας. Για την επανεκκίνηση του κώδικα, θα πρέπει να ξαναγίνει σύνδεση στο Raspberry Pi μέσω SSH, και να εκτελεστεί το script που τον ξεκινά από την αρχή.

Παράλληλα, το Arduino από την στιγμή που τροφοδοτηθεί και γίνει upload ο κώδικας του, παραμένει μόνιμα σε κατάσταση αναμονής εντολών από το Raspberry Pi, έως ότου το Raspberry Pi απενεργοποιηθεί πλήρως. Το Arduino, περιμένει εντολές μιας συγκεκριμένης μορφής. Η εντολή αυτή περιλαμβάνει πληροφορίες για τις κατάλληλες ρυθμίσεις που πρέπει να πάρουν οι κινητήρες για την ομαλή κίνηση του οχήματος. Την στιγμή που παραλάβει εντολές κίνησης το Arduino θα ξεκινήσει τις διεργασίες εκτέλεσης της εντολής, και παράλληλα θα ξεκινήσει την σάρωση εμπρός και πίσω του οχήματος για τυχόν εμπόδια, επιστρέφοντας την απόσταση στο Raspberry Pi.

Εδώ, η εφαρμογή της κινητής συσκευής αποτελεί το όργανο με το οποίο μπορεί ο χρήστης να επικοινωνήσει άμεσα τις λειτουργίες του οχήματος με την χρήση κουμπιών. Οι συνοπτικές λειτουργίες αποτελούν την επιλογή μεταξύ τριών μεθόδων κίνησης, την έξοδο από μια μέθοδο κίνησης, και την έξοδο από τον κώδικα. Η κινητή συσκευή πρέπει να συνδεθεί στο τοπικό δίκτυο που δημιουργεί το Raspberry Pi, μέσω WiFi. Με την σύνδεση αυτή δύναται να επικοινωνήσει ομαλά με το Raspberry Pi.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

3.3 Σύνδεση στο όχημα εξ' αποστάσεως, μέσω VPN

Ένας πιθανός παρατηρητής μπορεί συνδεθεί εξ' αποστάσεως στο όχημα και να παραλάβει την ροή βίντεο που παράγεται από το Raspberry Pi. Αυτό μπορεί να επιτευχθεί με την χρήση του εργαλείου freemium ανοικτού λογισμικού, Zerotier VPN.

Το ZeroTier λειτουργεί ως εικονικό Ethernet Switch επιπέδου 2 στο cloud, δημιουργώντας μια εικονική επικάλυψη δικτύου πάνω από τα υπάρχοντα φυσικά δίκτυα. Οι συσκευές που συμμετέχουν σε ένα τέτοιο δίκτυο δημιουργούν άμεσες συνδέσεις P2P μέσω κρυπτογραφημένων καναλιών επικοινωνίας, παρακάμπτοντας τους παραδοσιακούς μηχανισμούς δρομολόγησης, επιτρέποντας την αποτελεσματική μετάδοση δεδομένων μεταξύ των κόμβων ανεξάρτητα από τη φυσική τους θέση.

Κάθε εικονικό δίκτυο που δημιουργείται διαθέτει έναν Κέντρικό Κόμβο-Ελεγκτή(Central Controller Node). Αυτός ο κόμβος λειτουργεί ως η κεντρική αρχή που διαχειρίζεται τα δίκτυα ZeroTier. Διαχειρίζεται την εκχώρηση διευθύνσεων, τον έλεγχο ταυτότητας και τις πολιτικές δικτύου. Η επικοινωνία με τον ελεγκτή πραγματοποιείται μέσω του RESTful API του ZeroTier. Κάθε συσκευή που συμμετέχει στο δίκτυο, που ονομάζονται κόμβοι, διαθέτουν ένα μοναδικό ζεύγος δημόσιου/ιδιωτικού κλειδιού 256-bit που βασίζεται στην κρυπτογραφία Curve25519, διασφαλίζοντας ασφαλή επικοινωνία και επαλήθευση ταυτότητας. Κατά την είσοδο στο δίκτυο, οι κόμβοι δημιουργούν και ανταλλάσσουν κρυπτογραφικά κλειδιά. Ο κόμβος στέλνει ένα αίτημα που περιέχει το δημόσιο κλειδί του στον κόμβο-ελεγκτή για αποδοχή. Ο ελεγκτής επαληθεύει την ταυτότητα του κόμβου επικυρώνοντας την ψηφιακή υπογραφή χρησιμοποιώντας το δημόσιο κλειδί του, παραχωρώντας πρόσβαση, εάν εξουσιοδοτηθεί από τον διαχειριστή του δικτύου, τον χρήστη. Μετά τον επιτυχή έλεγχο ταυτότητας, ο κόμβος-ελεγκτής εκχωρεί μια διεύθυνση ZeroTier 40-bit (παρόμοια με μια διεύθυνση IPv6) στον κόμβο. Αυτές οι διευθύνσεις είναι μοναδικές στο δίκτυο ZeroTier.

Τα πακέτα δεδομένων ενσωματώνονται στις προσαρμοσμένες κεφαλίδες πρωτοκόλλου του ZeroTier, που περιέχουν διευθύνσεις ZeroTier προέλευσης και προορισμού, επιτρέποντας την απρόσκοπτη δρομολόγηση μέσω του δικτύου επικάλυψης. Το ZeroTier χρησιμοποιεί το Cryptokey Routing (CKR) για να δημιουργήσει κρυπτογραφημένα κανάλια επικοινωνίας, απευθείας μεταξύ κόμβων με βάση τις εκχωρημένες διευθύνσεις ZeroTier, διασφαλίζοντας ασφάλεια και απόρρητο από άκρο σε άκρο.

Το ZeroTier, με τα ισχυρά κρυπτογραφικά του πρωτόκολλα, την αποκεντρωμένη αρχιτεκτονική P2P και τα κρυπτογραφημένα κανάλια επικοινωνίας, δημιουργεί ένα ασφαλές, επεκτάσιμο και ευέλικτο δίκτυο επικάλυψης. Η τεχνική του ικανότητα επιτρέπει την απρόσκοπτη συνδεσιμότητα, καθιστώντας το μια ευέλικτη λύση για διάφορες εφαρμογές μηχανικής δικτύου, από την απομακρυσμένη πρόσβαση έως την ενοποίηση κατανεμημένων συστημάτων.

Στην συγκεκριμένη εργασία, το Raspberry Pi έχει συνδεθεί σε ένα δίκτυο Zerotier [Παράρτημα 7.3]. Ο χρήστης/παρατηρητής μπορεί να παρακολουθήσει, από απόσταση, τις ροές βίντεο του Raspberry Pi, άμα έχει μπει στο ίδιο δίκτυο, και πατήσει τον εξής σύνδεσμο στον περιηγητή διαδικτύου επιλογής: **“http://10.147.19.76:5001/remote_stream”**.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

4. Θεωρητικό υπόβαθρο χρησιμοποιηθέντων αλγορίθμων

4.1. Γκαουσιανό θόλωμα – Gaussian Blur

Προτού γίνει ο εντοπισμός ακμών σε μια ασπρόμαυρη εικόνα, πρέπει να εφαρμοστεί Gaussian Blur. Είναι μια αρκετά χρησιμοποιούμενη λειτουργία στα πεδία της υπολογιστικής όρασης και γραφικών υπολογιστών. Ο κύριος στόχος του αλγορίθμου αυτού είναι η εξάλειψη του θορύβου της εικόνας και της λεπτομέρειας. Το οπτικό αποτέλεσμα αυτού του αλγορίθμου σε μια εικόνα είναι ένα “θόλωμα”, όμοιο με το θόλωμα της ανθρώπινης όρασης εξαιτίας της μυωπίας ή πρεσβυωπίας, ή και με την εμφάνιση μιας φωτογραφίας με λανθασμένη ρύθμιση του φακού της κάμερας.

Στον κώδικα, ο αλγόριθμός αυτός καλείται ως εξής:

```
# Blurring image to reduce noise  
blurred_image = cv2.GaussianBlur(image, (5,5), 0)
```

Η εντολή αυτή δημιουργεί έναν πίνακα $M \times M$, με M το μέγεθος κάθε πλευράς του πίνακα. Στην συγκεκριμένη περίπτωση $M = 5$. Η τιμή 0 που εμφανίζεται στην εντολή είναι η τιμή σ . Η εντολή αυτή κανονικά δέχεται και την τιμή σ_y , αλλά καθώς δεν έχει οριστεί, θα πάρει τιμή του σ . Αν και οι δύο τιμές σ_x και σ_y είναι 0, τότε υπολογίζονται αυτόματα από τον εξής τύπο:

$$(Συνάρτηση 4.1.1) \sigma = 0.3 \cdot ((M - 1) \cdot 0.5 - 1) + 0.8$$

Ο πίνακας που δημιουργείται περιέχει τιμές οι οποίες ακολουθούν την Γκαουσιανή συνάρτηση κατανομής. Ο τύπος που χρησιμοποιείται για τον υπολογισμό των τιμών αυτών είναι ο εξής:

$$(Συνάρτηση 4.1.2) G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Στην συνάρτηση 4.2, οι τιμές x, y αναπαριστούν τις συντεταγμένες του πίνακα. Τυπικά ο πίνακας έχει ως σημείο $(0,0)$ το κέντρο του πίνακα. Άρα σε έναν πίνακα 5×5 , οι τιμές x, y μπορούν να πάρουν τιμές $[-2, 2]$ Ύστερα θα εφαρμοστεί συνελικτικά πάνω στην εικόνα ενδιαφέροντος. Κάθε εικονοστοιχείο της εικόνας θα περαστεί από μια τιμή που του αντιστοιχεί στον πίνακα συνέλιξης.

Η νέα τιμή κάθε εικονοστοιχείου ορίζεται σε έναν σταθμισμένο μέσο όρο των τιμών των γειτονικών εικονοστοιχείων αυτού του εικονοστοιχείου. Η τιμή του κεντρικού εικονοστοιχείου λαμβάνει το μεγαλύτερο βάρος (με την υψηλότερη Γκαουσιανή τιμή) και τα γειτονικά εικονοστοιχεία λαμβάνουν μικρότερες τιμές καθώς αυξάνεται η απόστασή τους από το κεντρικό εικονοστοιχείο. Αυτό έχει ως αποτέλεσμα ένα θάμπωμα που διατηρεί τα ακμές καλύτερα από άλλα, πιο ομοιόμορφα φίλτρα θολώματος.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 4.1.1 Παράδειγμα εφαρμογής του Γκαουσιανού φίλτρου
Πηγή: <https://commons.wikimedia.org/w/index.php?curid=5949583>

Στην Εικόνα 4.1.1 φαίνονται δύο παρόμοιες εικόνες. Η πάνω εικόνα περιέχει θόρυβο χάρη στον τρόπο εκτύπωσης της φωτογραφίας, χρησιμοποιώντας εικόνα ενδιάμεσων τόνων. Αυτός ο τρόπος αναπαράστασης εικόνας χρησιμοποιεί μικροσκοπικές τελείες οι οποίες αλλάζουν μέγεθος και πυκνότητα ανάλογα με την ανοιχτόχρωμη ή σκουρόχρωμη απόχρωση που πρέπει να παρουσιαστεί. Στην κάτω εικόνα παρατηρείται η εξομάλυνση του θορύβου με την χρήση του Γκαουσιανού φίλτρου.



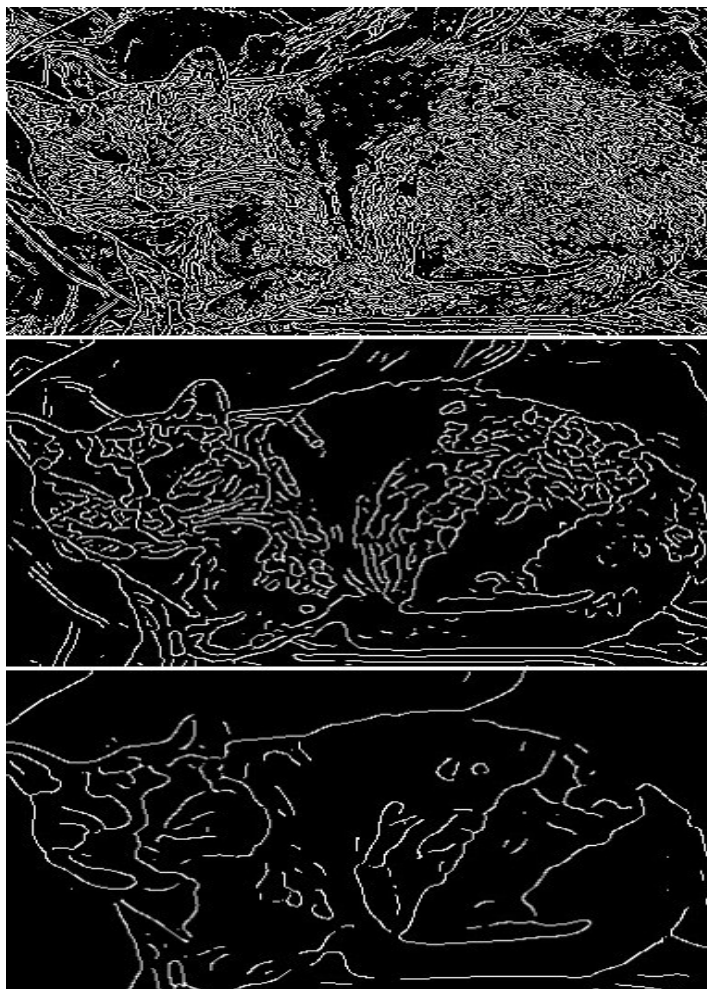
Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Εικόνα 4.1.2 Παράδειγμα χρήσης διαφορετικών τιμών τυπικής απόκλισης

Πηγή: <https://commons.wikimedia.org/w/index.php?curid=41790217>

Όπως παρατηρείται στην Εικόνα 4.1.2, η αλλαγή στην τιμή του σ έχει μεγάλη επίπτωση στο “θόλωμα” της εικόνας. Στην περίπτωση μας, χρησιμοποιούμε τον αλγόριθμο αυτόν ως στάδιο προ-επεξεργασίας καθώς στο επόμενο στάδιο πρόκειται να εντοπίσουμε ακμές στην εικόνα μας.

Οι εικόνες συχνά περιέχουν θόρυβο ή μικρές ανωμαλίες που δεν αποτελούν μέρος των πραγματικών άκρων. Το θάμπωμα Gauss βοηθά στην εξομάλυνση αυτών των ανωμαλιών, μειώνοντας τον θόρυβο που διαφορετικά θα μπορούσαν να ανιχνευθούν ως ακμές. Αντί να υπάρχουν απότομες μεταβάσεις, που μπορεί να είναι θόρυβος, το θάμπωμα επιτρέπει στους αλγόριθμους ανίχνευσης ακμών να εντοπίζουν ομαλότερες και πιο συνεχείς μεταβάσεις στην ένταση. Αυτό βοηθά στην απόκτηση καθαρότερων, πιο καθορισμένων άκρων, οδηγώντας σε καλύτερο εντοπισμό.



Σχέδιο 4.1.3 Εντοπισμός ακμών μετά από Γκαουσιανό φιλτράρισμα διαφόρων βαθμών

Πηγή: <https://commons.wikimedia.org/w/index.php?curid=10581259>

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

4.2. Αλγόριθμος Canny

Οι γραμμές της λωρίδας εντοπίζονται από τον αλγόριθμο αναγνώρισης ακμών Canny. Ο αλγόριθμος αυτός έχει δύο αποτελέσματα. Το πρώτο είναι ο εντοπισμός ακμών στην εικόνα και το δεύτερο είναι η μείωση του όγκου της πληροφορίας που περιέχει μια εικόνα καθώς μετά την εφαρμογή του αλγορίθμου, η εικόνα περιέχει μόνο τις εντοπισμένες ακμές. Αυτό συμβάλλει στην επιτάχυνση της επεξεργασίας της εικόνας ενδιαφέροντος.

Ο αλγόριθμος αυτός αναπτύχθηκε από τον John F. Canny το 1986. Ο Canny παρατήρησε πως οι απαιτήσεις για την ανίχνευση άκρων σε διαφορετικά συστήματα όρασης είναι σχετικά παρόμοιες. Έτσι, κατασκευάστηκε ένας αλγορίθμος ανίχνευσης ακμών για την αντιμετώπιση αυτών των απαιτήσεων που μπορεί να εφαρμοστεί σε ένα ευρύ φάσμα καταστάσεων. Τα κεντρικά κριτήρια για την ανίχνευση ακμών είναι τα εξής:

1. Η επίτευξη χαμηλού ποσοστού σφάλματος στην ανίχνευση ακμών είναι ζωτικής σημασίας, διασφαλίζοντας την ακριβή λήψη όσο το δυνατόν περισσότερων άκρων που απεικονίζονται στην εικόνα.
2. Τα σημεία ακμών που προσδιορίζονται από τον χειριστή πρέπει να ευθυγραμμίζονται με ακρίβεια με το κέντρο της ανιχνευμένης ακμής.
3. Κάθε άκρο που υπάρχει στην εικόνα θα πρέπει να αναγνωρίζεται μόνο μία φορά και θα πρέπει να καταβάλλονται προσπάθειες για την αποφυγή λανθασμένων άκρων που προκαλούνται από το θόρυβο της εικόνας.

Αρχικά είναι δεδομένο πως ο αλγόριθμος αυτός θα εντοπίσει ακμές σε μια ασπρόμαυρη εικόνα. Αυτό σημαίνει πως τα εικονοστοιχεία της εικόνας αυτής, για τον υπολογιστή, μπορούν να κατέχουν τιμές από 0 έως 255, με το 0 να υποδεικνύει πλήρες μαύρο χρώμα, και το 255 να υποδεικνύει πλήρες λευκό χρώμα. Η εικόνα έτσι μπορεί να παρουσιαστεί σαν έναν πίνακα με διαστάσεις ίδιες με αυτές της εικόνας, και σε κάθε τιμή του πίνακα να περιέχεται ένας αριθμός που κυμαίνεται στο διάστημα [0,255]. Οι ακμές σε αυτήν την περίπτωση ορίζονται ως οι περιοχές στις οποίες οι τιμές γειτονικών εικονοστοιχείων παρουσιάζουν μεγάλη μεταβολή, ως προς τις τιμές τους, μεταξύ τους. Η ευαισθησία του αλγορίθμου σε αυτήν την μεταβολή μπορεί να ρυθμιστεί προγραμματιστικά.

Στον κώδικα, ο αλγόριθμος αυτός καλείται ως εξής:

```
# Detecting edges in the whole image
edge_image = cv2.Canny(blurred_image, 80, 130)
```

Όπως αναφέρθηκε στην υποενότητα 4.1, το πρώτο βήμα που πρέπει να πραγματοποιηθεί για την ανίχνευση ακμών είναι η εφαρμογή του Γκαουσιανού φίλτρου για την μείωση του θορύβου που είναι παρόν στην εικόνα. Όπως φαίνεται στην Εικόνα 4.1.3, εφαρμογές του φίλτρου αυτού με διαφορετικές ρυθμίσεις μπορούν να δώσουν άκρως διαφορετικά αποτελέσματα. Αν εφαρμοστεί το θόλωμα με χαμηλή τιμή σ ή με μικρό μέγεθος του συνελκτικού πίνακα, τότε το αποτέλεσμα θα είναι μια εικόνα θολωμένη, αλλά όχι αρκετά έτσι ώστε να εξαλειφθεί ο θόρυβος, και ανάλογα με τον στόχο που αναζητούμε, ίσως να μην έχουμε ικανοποιητικό επίπεδο θαμπώματος στην εικόνα. Έτσι στην συνέχεια, μπορεί ο αλγόριθμος Canny να εντοπίσει ακμές σε σημεία που δεν μας ενδιαφέρουν, ή να εντοπίσει υπεραρκετές ακμές εξαιτίας του θορύβου.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Ύστερα από την εφαρμογή του θολώματος, εκτελούμε την εντολή που αναφέρεται παραπάνω, χρησιμοποιώντας ως είσοδο την εικόνα που παρήχθη από το Γκαουσιανό θόλωμα. Ο υπολογιστής θα ακολουθήσει τα παρακάτω βήματα για την ανίχνευση ακμών στην εικόνα:

1. Εύρεση κλίσης μεταβολής της έντασης κάθε εικονοστοιχείου της εικόνας:

Εύρεση του ποσού, ρυθμού μεταβολής και κατεύθυνση μεταβολής της τιμής των εικονοστοιχείων σε ένα διάστημα.

Στο βήμα αυτό χρησιμοποιούνται φίλτρα εύρεσης της πρώτης παραγώγου ως προς την μεταβολή τιμής των εικονοστοιχείων, για κάθε εικονοστοιχείο. Για τον σκοπό αυτόν υπάρχουν διάφορα φίλτρα όπως Roberts cross, Prewitt, Kayyali, Sobel. Στην εντολή που καλείται στην παρούσα εργασία, χρησιμοποιείται ο Sobel πυρήνας. Αυτή η μέθοδος χρησιμοποιεί δύο πίνακες των οποίων οι τιμές αντιπροσωπεύουν βάρη ή συντελεστές που χρησιμοποιούνται κατά τη διαδικασία δισδιάστατης συνέλιξης τους με την αρχική εικόνα, για τον προσεγγιστικό υπολογισμό της μεταβολής τιμών εικονοστοιχείων στοιχειωδών κομματιών εικόνας. Οι δύο πίνακες αυτοί έχουν την εξής μορφή:

$$(Συνάρτηση 4.2.1)G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A$$

$$(Συνάρτηση 4.2.2)G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

Ορίζουμε το A ως την εικόνα πηγής και το G_x και το G_y είναι δύο εικόνες που σε κάθε σημείο περιέχουν τις οριζόντιες (G_x) και κατακόρυφες (G_y) προσεγγίσεις παραγώγων των αλλαγών τιμής των εικονοστοιχείων. Το σύμβολο “*” σε αυτήν την περίπτωση, εκφράζει την πράξη της δισδιάστατης συνέλιξης.

- Οριζόντιος πυρήνας Sobel G_x (Συνάρτηση 4.2.1): Ο πίνακας αυτός ανιχνεύει οριζόντιες αλλαγές στην ένταση τιμών εικονοστοιχείων. Οι τιμές -1 και 1 υποδεικνύουν την ανίχνευση αλλαγών στην ένταση από δεξιά προς τα αριστερά. Το μεγαλύτερο βάρος 2 στην κεντρική στήλη τονίζει περισσότερο τη συμβολή του κεντρικού εικονοστοιχείου, σηματοδοτώντας πιο δυνατές οριζόντιες ακμές. Οι τιμές 0 δεν συμβάλλουν στην παρατηρητέα αλλαγή οριζόντιων αλλαγών τιμών, λειτουργώντας ως ουδέτερα στοιχεία στον υπολογισμό.
- Κάθετος πυρήνας Sobel G_y (Συνάρτηση 4.2.2): Ο πίνακας αυτός ανιχνεύει κάθετες αλλαγές στην ένταση τιμών εικονοστοιχείων. Οι τιμές -1 και 1 υποδεικνύουν την ανίχνευση αλλαγών στην ένταση από πάνω προς τα κάτω. Το υψηλότερο βάρος 2 στη μεσαία γραμμή τονίζει περισσότερο τη συμβολή του κεντρικού εικονοστοιχείου, τονίζοντας ισχυρότερα κάθετες ακμές. Οι τιμές 0 δεν συμβάλλουν στην κάθετη κλίση.

Οι δύο πίνακες αυτοί θα εφαρμοστούν “σειριακά” πάνω στην εικόνα πηγή, εκτελώντας την πράξη της συνέλιξης με όλα τα εικονοστοιχεία. Η πράξη αυτή πραγματοποιείται πολλαπλασιάζοντας κάθε στοιχείο των πυρήνων Sobel, G_x και G_y με την αντίστοιχη γειτονιά τιμών των εικονοστοιχείων της εικόνας, αθροίζοντας το γινόμενο και θέτοντας το αποτέλεσμα στο κέντρο της εικόνας εξόδου.

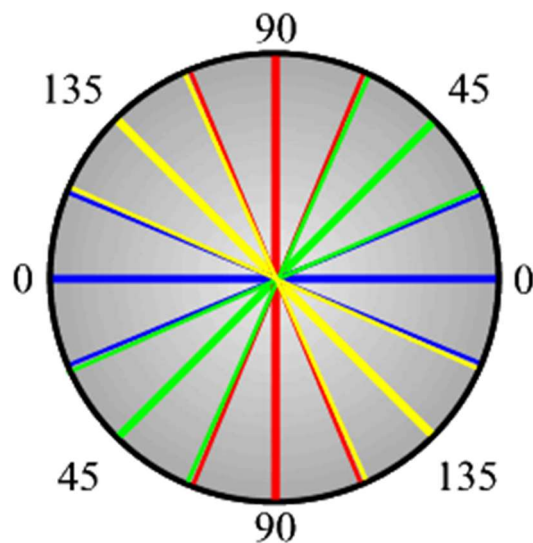
Ύστερα, για κάθε εικονοστοιχείο της εικόνας εξόδου μπορούμε να υπολογίσουμε τα διανύσματα των αλλαγών τιμών των εντάσεων των εικονοστοιχείων όπως και την κατεύθυνση του διανύσματος με τους παρακάτω τύπους:

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

$$(Συνάρτηση 4.2.3) G = \sqrt{Gx^2 + Gy^2}$$

$$(Συνάρτηση 4.2.4) \theta = \tan^{-1} \left(\frac{Gy}{Gx} \right)$$

Η κατεύθυνση της κλίσης είναι πάντα κάθετη στα άκρα. Η γωνία κατεύθυνσης της άκρης στρογγυλεύεται σε μία από τις τέσσερις γωνίες που αντιπροσωπεύουν την κατακόρυφη, την οριζόντια και τις δύο διαγώνιες (0° , 45° , 90° και 135°). Μια κατεύθυνση άκρης που πέφτει σε κάθε περιοχή χρώματος θα οριστεί σε μια συγκεκριμένη τιμή γωνίας, για παράδειγμα, θ σε $[0^\circ, 22.5^\circ]$ ή $[157.5^\circ, 180^\circ]$ θα μετατραπούν σε 0° .



Εικόνα 4.2.1 Όρια γωνιών των διανυσμάτων και οι αντίστοιχες τιμές στρογγυλοποίησης
Πηγή: https://sbme-tutorials.github.io/2018/cv/notes/4_week4.html

Σε αυτήν την περίπτωση, αν $\theta = 0$, τότε γνωρίζουμε πως πρόκειται για μια κατακόρυφη ακμή που είναι πιο “φωτεινή” στη δεξιά ή αριστερή πλευρά, ανάλογα με τις τιμές των Gx , Gy .

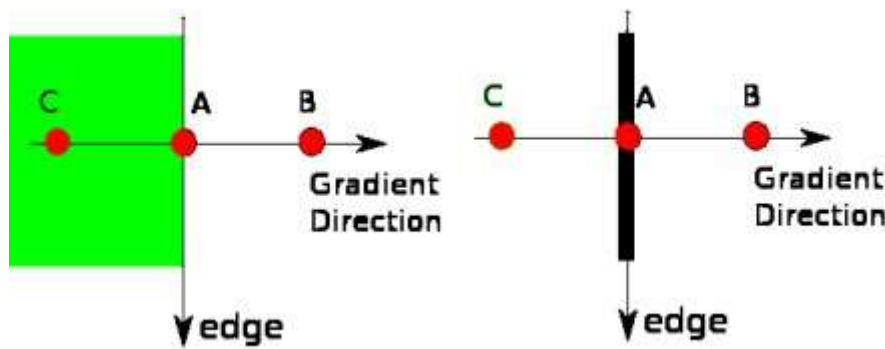


Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Εικόνα 4.2.2 Στα αριστερά, μια εικόνα ύστερα από εφαρμογή Γκαουσιανού φίλτρου
Στα δεξιά, μια εικόνα ύστερα από την εύρεση κλίσεων μεταβολής έντασης
Πηγή: https://sbme-tutorials.github.io/2018/cv/notes/4_week4.html

2. Εξάλειψη των μη μέγιστων τιμών:

Αφού ληφθεί το μέγεθος και η κατεύθυνση της κλίσης, γίνεται πλήρης σάρωση της εικόνας για να αφαιρεθούν τυχόν ανεπιθύμητα εικονοστοιχεία που μπορεί να μην αποτελούν μέρη ακμής. Για αυτό, σε κάθε εικονοστοιχείο, ελέγχεται εάν είναι ένα τοπικό μέγιστο στη γειτονιά του προς την κατεύθυνση της κλίσης. Συγκεκριμένα:



Εικόνα 4.2.3 Εξάλειψη των μη μέγιστων τιμών
Πηγή: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

Συγκρίνεται η ένταση ακμής του τρέχοντος εικονοστοιχείου με την ένταση ακμής του εικονοστοιχείου στις θετικές και αρνητικές κατευθύνσεις κλίσης.

Εάν η ένταση ακμής του τρέχοντος εικονοστοιχείου είναι η μεγαλύτερη σε σύγκριση με τα άλλα εικονοστοιχεία στη μάσκα με την ίδια κατεύθυνση (για παράδειγμα, ένα εικονοστοιχείο που δείχνει προς την κατεύθυνση $y'y$ θα συγκριθεί με το εικονοστοιχείο πάνω και κάτω από αυτό στον κατακόρυφο άξονα), η τιμή θα διατηρηθεί. Διαφορετικά, η τιμή θα καταργηθεί.

Με την Εικόνα 4.2.3 ως παράδειγμα:

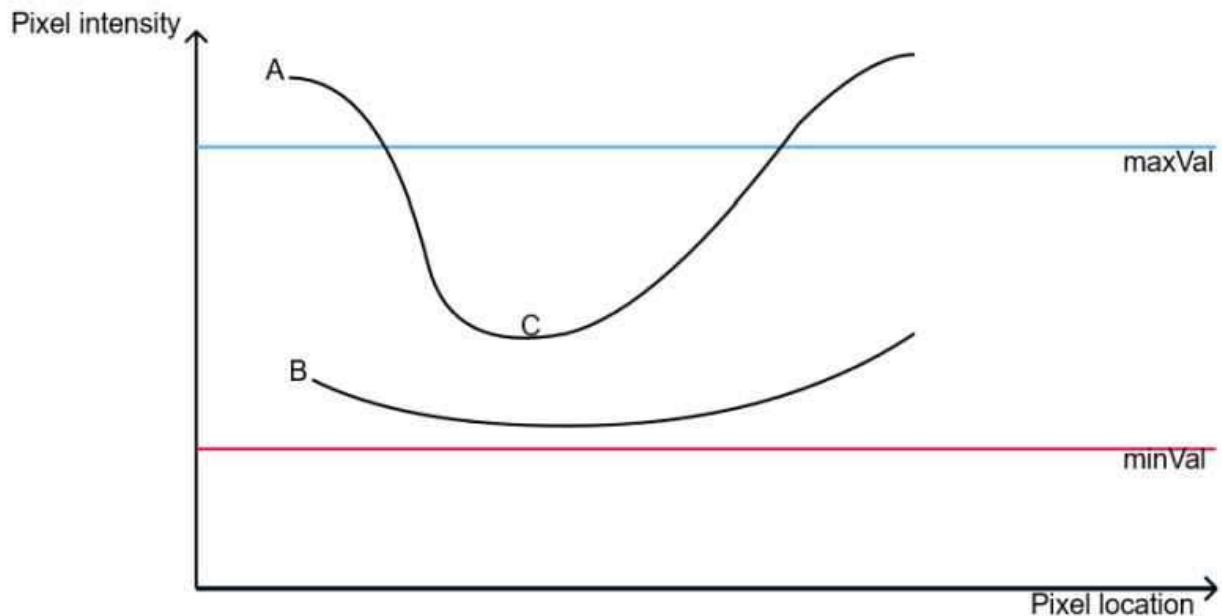
Το σημείο A βρίσκεται σε μια ακμή (σε κατακόρυφη κατεύθυνση). Η κατεύθυνση της κλίσης είναι κάθετη ως προς την ακμή. Τα σημεία B και C βρίσκονται στην ίδια κατεύθυνση κλίσης. Έτσι το σημείο A ελέγχεται με τα σημεία B και C για να ελεγχθεί αν το A σχηματίζει τοπικό μέγιστο. Εάν ναι, θεωρείται για το επόμενο στάδιο, διαφορετικά, καταστέλλεται (μηδενίζεται).

3. Κατωφλίωση τιμών και φιλτράρισμα ακμών

Αυτό το στάδιο καθορίζει ποιες από όλες οι ακμές είναι πραγματικά ακμές και ποιες όχι. Για αυτό, χρησιμοποιούνται δύο τιμές κατωφλίου, $minVal$ και $maxVal$. Οποιοσδήποτε ακμές με κλίση έντασης μεγαλύτερη από το $maxVal$ είναι βέβαιο ότι θα είναι άκρες και εκείνες κάτω από το $minVal$ είναι βέβαιο ότι δεν είναι άκρες, επομένως απορρίπτονται. Εκείνοι που βρίσκονται ανάμεσα σε αυτά τα δύο κατώφλια

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

ταξινομούνται ακμές ή μη ακμές με βάση τη συνδεσιμότητα τους. Εάν συνδέονται με εικονοστοιχεία "σίγουρης ακμής", θεωρούνται ότι αποτελούν μέρος των ακμών. Διαφορετικά, απορρίπτονται.

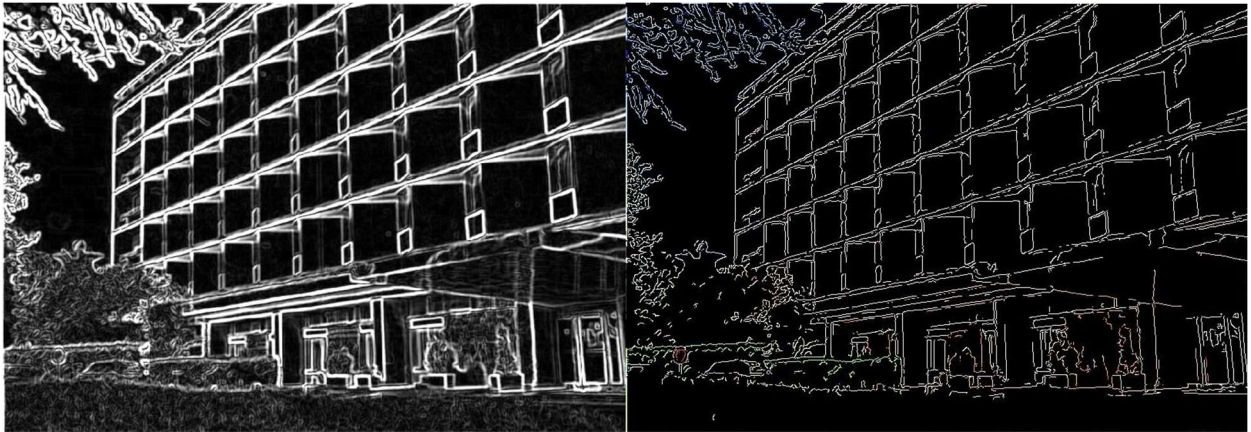


Εικόνα 4.2.4 Διάγραμμα τιμών εντάσεων εικονοστοιχείων με τα δύο κατώφλια

Πηγή: <https://wttech.blog/blog/2022/edge-detection-and-processing-using-canny-edge-detector-and-hough-transform/>

Με την Εικόνα 4.2.4 ως παράδειγμα, παρατηρείται ότι το άκρο A είναι πάνω από την τιμή $maxVal$, επομένως θεωρείται ως "σίγουρη ακμή". Αν και η ακμή C είναι κάτω από το $maxVal$, συνδέεται με την ακμή A, έτσι ώστε να θεωρείται επίσης έγκυρη ακμή και να έχουμε αυτήν την πλήρη καμπύλη. Αλλά η ακμή B, αν και είναι πάνω από το $minVal$ και βρίσκεται στην ίδια περιοχή με αυτήν της ακμής C, δεν συνδέεται με καμία "σίγουρη άκρη", οπότε απορρίπτεται. Επομένως, είναι πολύ σημαντικό να επιλέξουμε $minVal$ και $maxVal$ ανάλογα για να έχουμε το σωστό αποτέλεσμα. Αυτό το στάδιο αφαιρεί επίσης τους θορύβους των μικρών εικονοστοιχείων με την υπόθεση ότι οι άκρες είναι μεγάλες γραμμές.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 4.2.5 Αποτέλεσμα κατωφλίωσης

Πηγή: https://sbme-tutorials.github.io/2018/cv/notes/4_week4.html

Παρατηρώντας την Εικόνα 4.2.5, φαίνεται το αισθητό αποτέλεσμα της κατωφλίωσης στην μείωση της έντασης των εντοπισμένων ακμών, όπως και στην μείωση του θορύβου γύρω από τις ακμές. Τελικό αποτέλεσμα είναι η ακριβής επισήμανση ακμών στην εικόνα.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

4.3. Μετασχηματισμός Hough

Ύστερα από τον εντοπισμό των ακμών στην εικόνα, πρέπει να εντοπιστούν οι γραμμές που σχηματίζουν οι ακμές που έχουν δημιουργηθεί.

Στην αυτοματοποιημένη ανάλυση ψηφιακών εικόνων, συχνά παρουσιάζεται το πρόβλημα του εντοπισμού απλών σχημάτων, όπως ευθείες, κύκλοι ή ελλείψεις. Σε πολλές περιπτώσεις, μπορεί να χρησιμοποιηθεί ένας αλγόριθμος εντοπισμού ακμών ως στάδιο προ-επεξεργασίας, έτσι ώστε να αναγνωριστούν σημεία, ή γραμμές μέσα στην εικόνα πηγή, οι οποίες συμπίπτουν με τα σχήματα ενδιαφέροντος. Εξαιτίας όμως του θορύβου στην εικόνα ή των ατελειών στους αλγορίθμους εντοπισμού ακμών, είναι πιθανό να υπάρχουν κενά ή ελλειπείς εικονοστοιχεία στο σχήμα ενδιαφέροντος, όπως και χωρικές αποκλίσεις μεταξύ της ιδανικής γραμμής/κύκλου/έλλειψης και των θορυβωδών ακραίων σημείων, όπως αυτά προκύπτουν από τον αλγόριθμο εντοπισμού ακμών. Για αυτούς τους λόγους, συχνά πρέπει να πραγματοποιείται ομαδοποίηση των εξαγόμενων σημείων σε ένα σύνολο γραμμών ή οποιουδήποτε σχήματος είναι απαραίτητο. Ο σκοπός του μετασχηματισμού Hough είναι να αντιμετωπίσει αυτό το πρόβλημα καθιστώντας δυνατή την εκτέλεση ομαδοποιήσεων σημείων ακμών σε υποψήφια αντικείμενα εκτελώντας μια διαδικασία ψηφοφορίας σε ένα σύνολο παραμετροποιημένων αντικειμένων εικόνας.

Ο αλγόριθμος αυτός λειτουργεί αναγνωρίζοντας πολύπλοκες γραμμές στις φωτογραφίες με αντιστοίχιση σημείων ακμών στο χώρο Hough. Έτσι είναι απαραίτητο να χρησιμοποιηθεί αλγόριθμος εντοπισμού ακμών, όπως ο προαναφερθέν αλγόριθμο Canny.

Είναι δεδομένο πως, στο καρτεσιανό σύστημα συντεταγμένων, οι εξισώσεις ευθειών έχουν την μορφή (Συνάρτηση 4.3.1) $y = a \cdot x + b$. Αξίζει να σημειωθεί πως, καθώς αναφερόμαστε σε εξίσωση, μια τιμή y , αντιστοιχεί σε μία και μόνο μία τιμή του x .

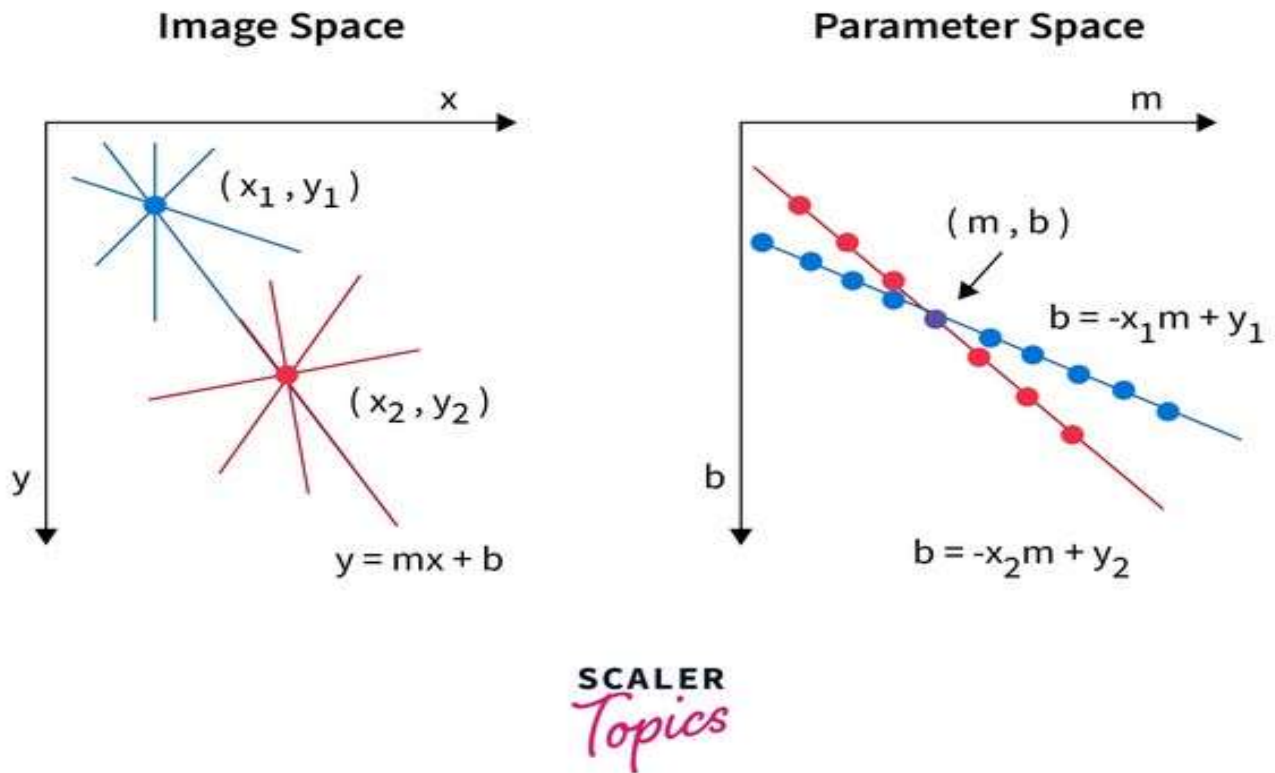
Μετασχηματίζοντας την Συνάρτηση 4.3.1, μπορούμε να αναπαραστήσουμε την συνάρτηση αυτή στο παραμετρικό σύστημα συντεταγμένων, έτσι ώστε να έχουμε μια συνάρτηση μεταξύ των a και b .

(Συνάρτηση 4.3.2) $b = -x \cdot a + y$. Έχοντας αυτόν τον τύπο, μπορούμε να αναπαραστήσουμε μια ευθεία στο καρτεσιανό σύστημα συντεταγμένων με άξονες x' , y' , στο παραμετρικό σύστημα συντεταγμένων με άξονες a' , b' . Με αυτόν τον τρόπο, μια ευθεία στο καρτεσιανό σύστημα παρουσιάζεται ως ένα σημείο στο παραμετρικό σύστημα.

Επίσης δεδομένο είναι πως από ένα σημείο στο καρτεσιανό σύστημα περνούν άπειρες ευθείες. Καθώς επίσης από πριν γνωρίζουμε πως όλες οι ευθείες, στο παραμετρικό σύστημα εμφανίζονται ως σημεία, συμπεραίνουμε το σύνολο όλων άπειρων ευθειών στο καρτεσιανό σύστημα θα σχηματίζουν μια ένα σύνολο σημείων, που με την σειρά τους, σχηματίζουν μια ευθεία.

Έστω υποθέτουμε δύο σημεία (x_1, y_1) και (x_2, y_2) στο καρτεσιανό σύστημα συντεταγμένων. Γνωρίζουμε πως από κάθε ένα από αυτά τα δύο σημεία περνούν άπειρες ευθείες, αλλά υπάρχει μία και μόνο μία ευθεία που περνά από τα δύο σημεία ταυτόχρονα. Έτσι στο παραμετρικό σύστημα, θα έχουμε δύο ευθείες, οι οποίες συμβολίζουν τις δύο οικογένειες σημείων που περνούν από τα δύο σημεία στο καρτεσιανό σύστημα, οι οποίες ευθείες θα τέμνονται. Το σημείο τομής αυτών των δύο ευθειών συμβολίζει την μια ευθεία που περνά από τα δύο σημεία στο καρτεσιανό σύστημα.

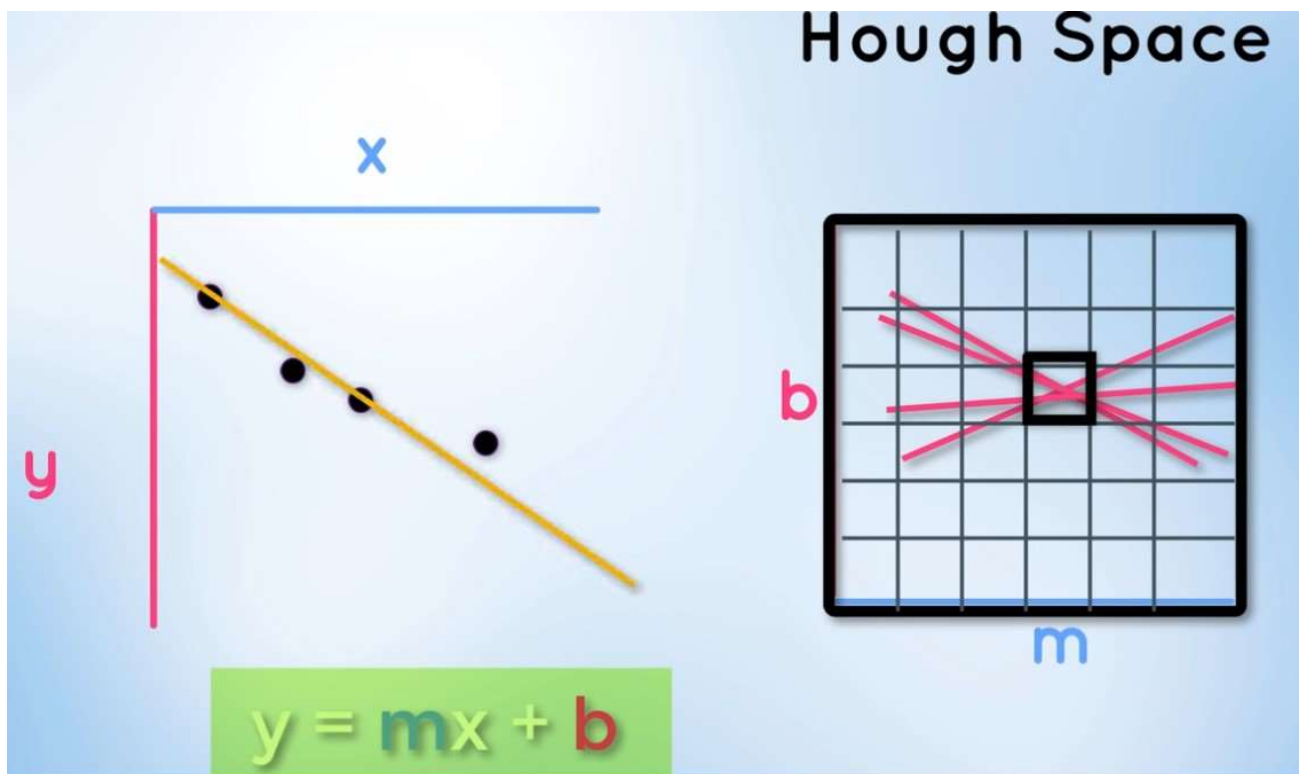
Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 4.3.1 Καρτεσιανό και παραμετρικό σύστημα συντεταγμένων
Πηγή: <https://www.scaler.com/topics/hough-transform-in-image-processing/>

Υπό κανονικές συνθήκες, εντοπίζονται πολλαπλά σημεία ενδιαφέροντος στο καρτεσιανό σύστημα, χάρη στους αλγορίθμους εντοπισμού ακμών, και είναι πρακτικά αδύνατο να υπάρχει μία και μόνο μία ευθεία που να περνάει από όλα τα σημεία που έχουν εντοπισθεί. Αυτό σημαίνει πως στο παραμετρικό σύστημα θα υπάρχουν πολλαπλές ευθείες οι οποίες, καθώς τείνουν προς ένα κοινό σημείο, το σημείο αυτό δεν είναι κοινό δεν είναι το ίδιο για όλες τις ευθείες, καθιστώντας αδύνατο να βρεθεί κοινή ευθεία στο καρτεσιανό σύστημα. Αυτό το πρόβλημα μπορεί να λυθεί με τον εξής τρόπο:

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 4.3.2 Εύρεση της ευθείας με την καλύτερη εφαρμογή ως προς τα σημεία ενδιαφέροντος

Πηγή: <https://youtu.be/eLTLtUVuuy4?si=GgdwINre4gYH0s7c&t=2721>

Έστω έχουμε 4 σημεία στο καρτεσιανό σύστημα. Αυτά τα σημεία αναπαριστώνται ως ευθείες στο παραμετρικό σύστημα. Καθώς δεν υπάρχει μια ευθεία που να τέμνει ταυτόχρονα αυτά τα σημεία, υπάρχουν ευθείες που να τέμνουν τα 4 αυτά σημεία ως προς 2 κάθε φορά. Έτσι δεν υπάρχει και σημείο τομής κοινό προς όλες τις ευθείες, αλλά υπάρχουν σημεία τομής για ζευγάρια ευθειών στο παραμετρικό σύστημα. Έτσι, μπορούμε να προσεγγίσουμε ένα υποθετικό σημείο τομής, κοινό προς όλες τις ευθείες στο παραμετρικό σύστημα, χωρίζοντας τους άξονες b ' b και m ' m σε στοιχειώδεις κομμάτια. Έστερα μετρούμε τα σημεία τομών μεταξύ ζευγαριών ευθειών και σημειώνουμε τις συντεταγμένες της “περιοχής” στην οποία σημειώνεται το μεγαλύτερο πλήθος τομών μεταξύ ευθειών. Αποτέλεσμα είναι ένα σημείο (b, m) το οποίο μπορεί να εκφράσει προσεγγιστικά μια ευθεία στο καρτεσιανό σύστημα, η οποία τείνει να περάσει από όλα τα σημεία ταυτόχρονα.

Με αυτήν την προσέγγιση είναι δυνατό να βρεθούν ευθείες με βάση σημείων και ευθειών που υπάρχουν στο καρτεσιανό σύστημα, αλλά με μια εξαίρεση.

Γνωρίζοντας τον ορισμό της συνάρτησης, μια κατακόρυφη ευθεία, για παράδειγμα $x = 5$, δεν υφίσταται ως συνάρτηση. Έτσι, για να γίνει δυνατός ο εντοπισμός γραμμών με κατακόρυφες ευθείες στο καρτεσιανό σύστημα, πρέπει να αλλάξει ο τρόπος αναπαράστασης των συναρτήσεων στο πολικό σύστημα συντεταγμένων.

Σε αυτό το σύστημα, οι συναρτήσεις δεν εκφράζονται σύμφωνα με την Συνάρτηση 4.3.1, αλλά εκφράζονται σύμφωνα με την παρακάτω συνάρτηση:

$$(Συνάρτηση 4.3.3) r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

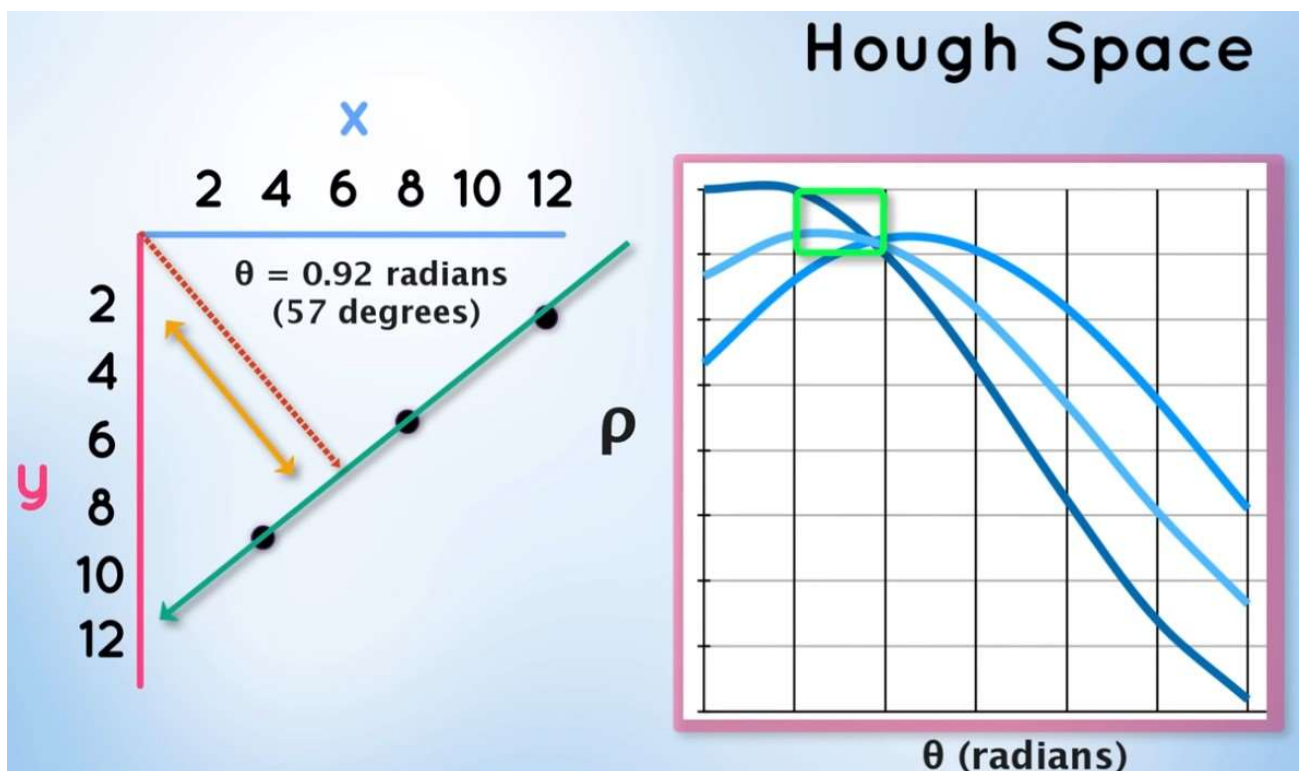
Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Με “ r ”: Το μήκος της ευθείας που σχηματίζεται μεταξύ της αρχής των αξόνων $x'x$, $y'y$ και του σημείου στο οποίο η ευθεία αυτή τέμνει την ευθεία ενδιαφέροντος κάθετα, που ονομάζεται ακτινική συντεταγμένη ή απλώς ακτίνα.

Και με “ θ ”: Την γωνία που σχηματίζει η ακτίνα του σημείου με έναν από τους δύο κύριους άξονες συντεταγμένων, συνήθως τον άξονα $x'x$, η οποία ονομάζεται γωνιακή συντεταγμένη ή αζιμούθιο.

Με αυτόν τον τρόπο αναπαράστασης, δεν υπάρχει πρόβλημα με τις κατακόρυφες ευθείες, καθώς όλες οι ευθείες περιγράφονται με γωνία και απόσταση από το κέντρο των αξόνων.

Αναπαριστώντας ένα σημείο χρησιμοποιώντας πολικές συντεταγμένες στο παραμετρικό σύστημα με άξονες (r, θ) , δεν λαμβάνουμε ευθείες, αλλά ημιτονοειδείς γραμμές.



Εικόνα 4.3.3 Εύρεση της ευθείας με την καλύτερη εφαρμογή ως προς τα σημεία ενδιαφέροντος, με την χρήση πολικών συντεταγμένων

Πηγή: <https://youtu.be/eLTLtUVuuy4?si=kdxvqNleIWsL7duo&t=3044>

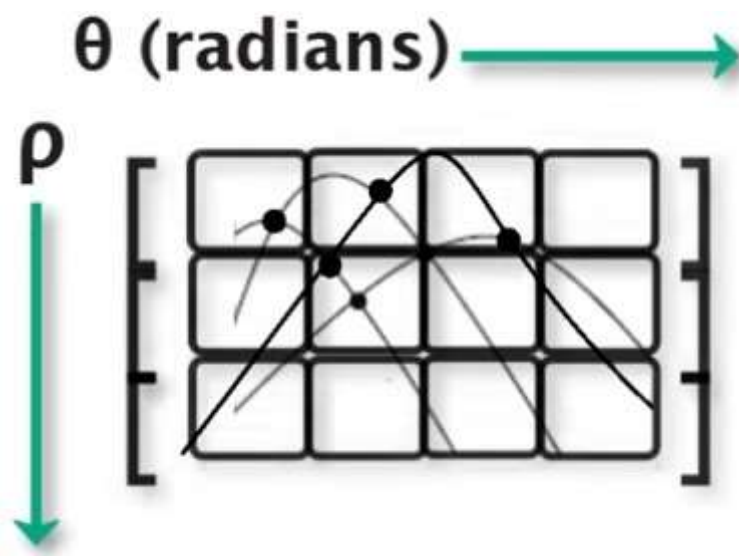
Έστω έχουμε 3 σημεία στο καρτεσιανό σύστημα συντεταγμένων και επιθυμούμε την ομαδοποίησή τους σε μια ευθεία. Με την χρήση πολικών συντεταγμένων, αυτά τα σημεία αναπαρίστανται ως ημιτονοειδείς γραμμές στον μετασχηματισμένο σύστημα συντεταγμένων. Χρησιμοποιώντας την ίδια μέθοδο “ψήφου” που αναφέρθηκε προηγουμένως, διαπιστώνουμε πως οι τρεις γραμμές αυτές τέμνονται σε μια συγκεκριμένη περιοχή. Γνωρίζοντας τις τιμές (r, θ) που αντιστοιχεί στην περιοχή με τις περισσότερες τομές μεταξύ γραμμών, μπορούμε να εκφράσουμε την προσεγγιστική ευθεία που περιγράφει τα τρία αρχικά σημεία.

Η εντολή χρησιμοποιείται στον κώδικα ως εξής:

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

```
# Detecting small lines in our edge image
side_lines_points = cv2.HoughLinesP(side_edge_image, 1, np.pi/180, 5,
np.array([]), minLineLength=10, maxLineGap=5)
```

Στην εντολή, εκτός από την πηγή εικόνα για τον αλγόριθμο, ορίζουμε και τις παραμέτρους με τις οποίες χωρίζουμε τον παραμετρικό χώρο έτσι ώστε να μπορούμε να μετρήσουμε τομές μεταξύ γραμμών μέσα στους χώρους που σχηματίζονται. Επίσης ορίζονται και το ελάχιστο αποδεκτό μήκος γραμμών, από το οποίο και κάτω, δεν θεωρούμε ότι έχει εντοπισθεί γραμμή, όπως και το μέγιστο μήκος μεταξύ ξεχωριστών γραμμών κάτω από το οποίο οι γραμμές θα αναπαρασταθούν ως μια γραμμή, και πάνω από το οποίο οι γραμμές θα παραμείνουν ξεχωριστές.



Εικόνα 4.3.4 Αναπαράσταση του διαχωρισμού του παραμετρικού πεδίου
Πηγή: <https://youtu.be/eLTLtUVuuy4?si=djKG0jWfRVDpHARy&t=3139>

Όπως φαίνεται στην Εικόνα 4.3.4, χωρίζουμε τον παραμετρικό χώρο έτσι ώστε να μπορούμε να μετρήσουμε τομές μεταξύ γραμμών μέσα στους χώρους που σχηματίζονται. Ο $y'y$ άξονας εξαρτάται από την τιμή r , σε πλήθος εικονοστοιχείων, που δώσουμε στην εντολή, στην περίπτωση αυτή $r = 1$, και ο $x'x$ άξονας εξαρτάται από την τιμή θ , μετρημένο σε ακτίνια, που σε αυτήν την περίπτωση, έχουμε ορίσει τιμή 1, από τον υπολογισμό $\theta = \pi/180$. Όσο πιο μικρές μονάδες χρησιμοποιήσουμε, τόσο πιο ακριβής θα είναι η καταμέτρηση τομών μεταξύ γραμμών. Και τέλος ορίζουμε το ελάχιστο πλήθος τομών πάνω από το οποίο θεωρούμε πως έχουμε βρει μια ευθεία στο καρτεσιανό σύστημα που περιγράφει προσεγγιστικά όλες τις γραμμές που τέμνονται στο παραμετρικό σύστημα. Σε αυτήν την περίπτωση, ορίζουμε κατώτατο όριο 5.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5. Υλοποίηση κώδικα

5.1. Εφαρμογή κινητής συσκευής

Ύστερα, μπορούμε να χρησιμοποιήσουμε την κινητή εφαρμογή για να ελέγξουμε το όχημα. Παρουσιάζεται παρακάτω η όψη της εφαρμογής.



Εικόνα 5.1.1 Αριστερά, η εφαρμογή προτού συνδεθεί στο Raspberry Pi, και δεξιά, αφού συνδεθεί

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

Η κινητή εφαρμογή θα παραμείνει σε κατάσταση αναμονής, όπως φαίνεται στην Εικόνα 4.1.1, μέχρι το κινητό να συνδεθεί στο Raspberry Pi, όταν αυτό έχει ξεκινήσει την λειτουργία του κώδικα. Επίσης στην εικόνα φαίνονται και τα κουμπιά που στέλνουν τις απαραίτητες εντολές στο Raspberry Pi, όπως και δύο πλαίσια, στα οποία φαίνονται ροές βίντεο από την κάμερα του Raspberry Pi. Στο πάνω πλαίσιο, φαίνεται η ροή της κάμερας με κάποιες παραπάνω πληροφορίες, και στο κάτω πλαίσιο φαίνεται η επεξεργασμένη ροή βίντεο που χρησιμοποιεί το Raspberry Pi καθώς αυτό οδηγεί αυτόνομα.

Το Raspberry Pi βρίσκεται σε κατάσταση αναμονής εντολών από την κινητή εφαρμογή. Η εφαρμογή μπορεί να στείλει τις παρακάτω εντολές στο Raspberry Pi:

- “m”, η οποία εντολή σημαίνει “Manual”. Με αυτήν την εντολή μπορούμε να καταλάβουμε χειροκίνητο έλεγχο του οχήματος. Πατώντας το κουμπί “m”, ενεργοποιούνται οι δύο ρυθμιστικές μπάρες. Η μπάρα στα αριστερά της οθόνης ρυθμίζει την ταχύτητα του οχήματος. Η μπάρα που βρίσκεται στο κάτω μέρος της εφαρμογής ρυθμίζει την πηδαλιούχηση του οχήματος. Κινώντας τον κόμβο που βρίσκεται σε κάθε μπάρα, μπορούμε να στείλουμε εντολές κίνησης στο όχημα ανάλογα με τις ανάγκες μας.
- “s”, η οποία εντολή σημαίνει “Self”. Με αυτήν την εντολή μπορούμε να ενεργοποιήσουμε την λειτουργία της αυτόνομης οδήγησης και του εντοπισμού λωρίδας με την χρήση της κάμερας. Όταν πατηθεί το κουμπί αυτό, το Raspberry Pi θα ξεκινήσει τον αλγόριθμο εντοπισμού του κέντρου της λωρίδας που έχει ρυθμιστεί να εντοπίζει. Παράλληλα, θα ενεργοποιηθεί η ροή βίντεο στο κάτω πλαίσιο, το οποίο θα δείχνει την επεξεργασμένη ροή βίντεο για την χρήση του εντοπισμού λωρίδας.
- “F”, η οποία εντολή σημαίνει “Free”. Με την εντολή αυτή ενεργοποιείται η λειτουργία της ελεύθερης περιπλάνησης. Με αυτήν την λειτουργία, το όχημα θα κινηθεί προς τα μια κατεύθυνση αποφεύγοντας εμπόδια, εφόσον παρουσιαστούν, με την χρήση τυποποιημένων κινήσεων αποφυγής.
- “e”, η οποία εντολή σημαίνει “Exit”. Πατώντας το κουμπί αυτό, δίνουμε εντολή στο Raspberry Pi να τερματίσει την εκτέλεση του κώδικα. Για να επανεκκινηθεί ο κώδικας πρέπει να συνδεθούμε ξανά με SSH και να εκτελέσουμε το script που ξεκινάει τον κώδικα του Raspberry Pi. Σε αυτήν την περίπτωση δεν χρειάζεται να ανεβάσουμε κώδικα προς το Arduino, καθώς αυτό παραμένει σε κατάσταση αναμονής εντολών.
- “q”, η οποία εντολή σημαίνει “Quit”. Αυτή η εντολή δεν έχει επίπτωση στην λειτουργία του κώδικα καθώς το Raspberry Pi βρίσκεται σε κατάσταση αναμονής εντολών από την εφαρμογή. Όταν το Raspberry Pi βρίσκεται σε κατάσταση εκτέλεσης διεργασιών λόγω των εντολών “m”, “s”, “F”, και ύστερα πατηθεί το κουμπί “q”, τότε το Raspberry Pi θα μηδενίσει την ταχύτητα και θα επιστρέψει στην κατάσταση αναμονής εντολών από την εφαρμογή.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5.1.1 Λειτουργία χειροκίνητης κίνησης “m”

Οι δύο ρυθμιστικές μπάρες μπορούν να παράξουν ακέραιες τιμές για την δημιουργία μιας εντολής κίνησης προς τους κινητήρες του οχήματος για την ταχύτητα και κατεύθυνση κίνησης.

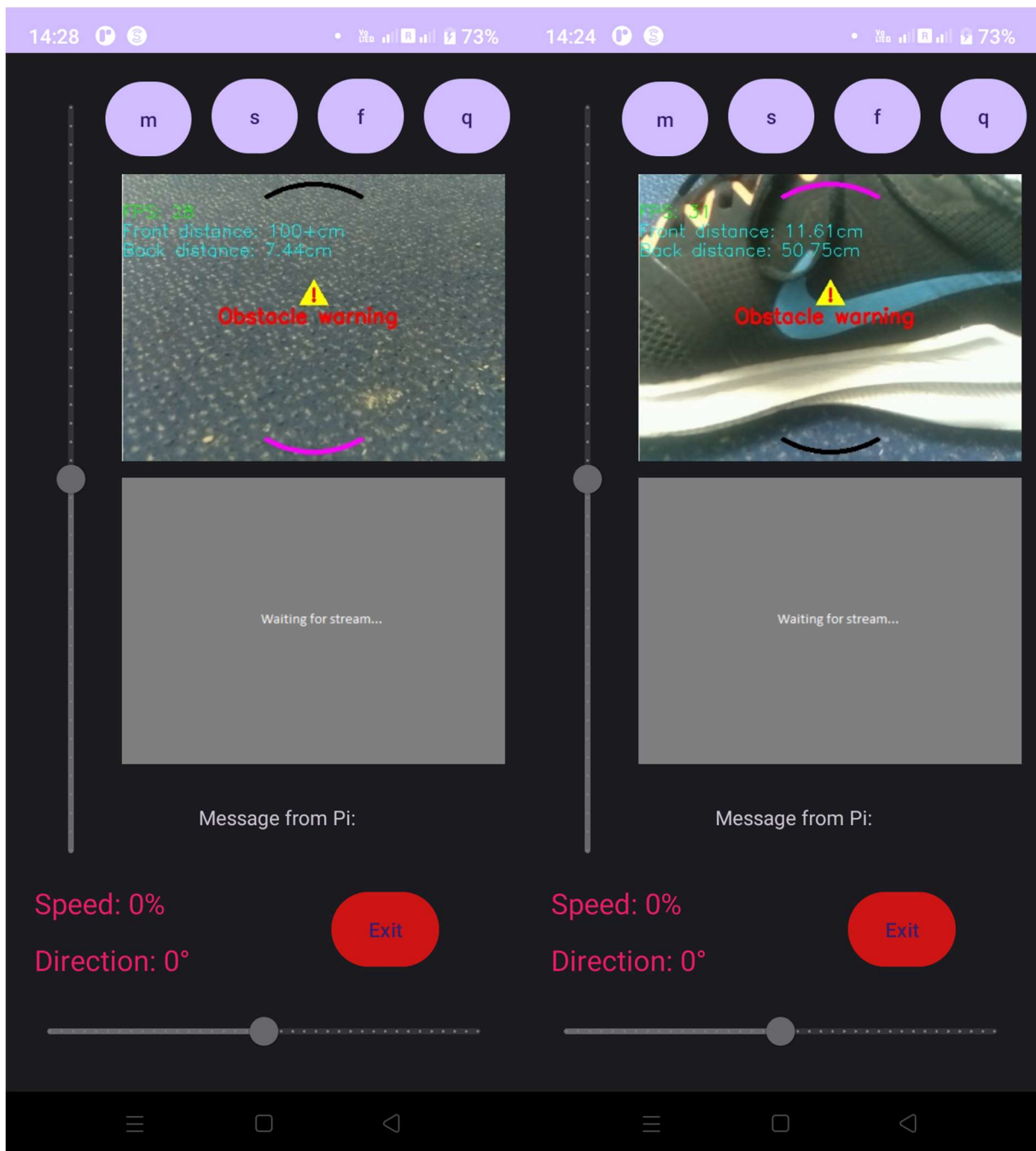
Οι ακέραιες τιμές που μπορεί να παράξει η αριστερή μπάρα ταχύτητας βρίσκονται στην περιοχή [-100, 100]. Ο αριθμός αυτός δίνει το ποσοστό τοις εκατό της ταχύτητας του οχήματος. Για παράδειγμα, άμα η τιμή βρίσκεται στο 77%, οι κινητήρες θα κινηθούν με το 77% της τελικής τους ταχύτητας. Αν η τιμή γίνει αρνητική, τότε το όχημα θα ξεκινήσει να οδεύει προς τα όπισθεν. Αν η τιμή γίνει 0, τότε το όχημα θα φρενάρει. Οι ακέραιες τιμές που μπορεί να παράξει η κάτω μπάρα κατεύθυνσης, βρίσκονται στην περιοχή [-180, 180]. Ο αριθμός αυτός δίνει τις μοίρες που πρέπει να στρίψει το όχημα. Άμα ο αριθμός βρίσκεται στο -180, τότε το όχημα θα στρίψει αριστερόστροφα με τον δεξιό κινητήρα να λειτουργεί πλήρως και τον αριστερό κινητήρα να μην κινείται καθόλου. Αν η τιμή γίνει 0, τότε το όχημα οδεύει προς τα εμπρός.

Με την εκκίνηση αυτής της λειτουργίας, το Raspberry Pi θα ξεκινήσει επίσης να στέλνει ροή βίντεο από την κάμερα με την οποία είναι συνδεδεμένο. Στην ροή βίντεο, περιλαμβάνεται επίσης το πλήθος των καρτέ ανά δευτερόλεπτο της ροής, με πράσινο χρώμα, στην πάνω αριστερή γωνία της πάνω ροής.

Επιπλέον, εμφανίζονται δύο δείκτες στην πάνω και κάτω άκρη της ροής. Αυτοί οι δείκτες σε σχήμα τόξου, αλλάζουν το χρώμα τους, από μαύρο σε ροζ, ανάλογα με την απόσταση που εντοπίζουν οι αισθητήρες απόστασης του Arduino. Το χρώμα του επάνω τόξου εξαρτάται από την μπροστινή απόσταση του οχήματος, και το κάτω τόξο εξαρτάται από την οπίσθια απόσταση του οχήματος. Σε απόσταση 40 εκατοστών, οι δείκτες γίνονται πλήρως μαύροι, δίνοντας έτσι την απώλεια κινδύνου σύγκρουσης με ένα εμπόδιο. Όσο η απόσταση γίνεται πιο μικρή, που σημαίνει πως υπάρχει αντικείμενο σε πορεία κρούσης με το όχημα, τα τόξα αλλάζουν σταδιακά χρώμα προς το ροζ, δείχνοντας οπτικά ότι πλησιάζουμε μια σύγκρουση. Σε απόσταση 10 εκατοστών, τα τόξα γίνονται πλήρως ροζ.

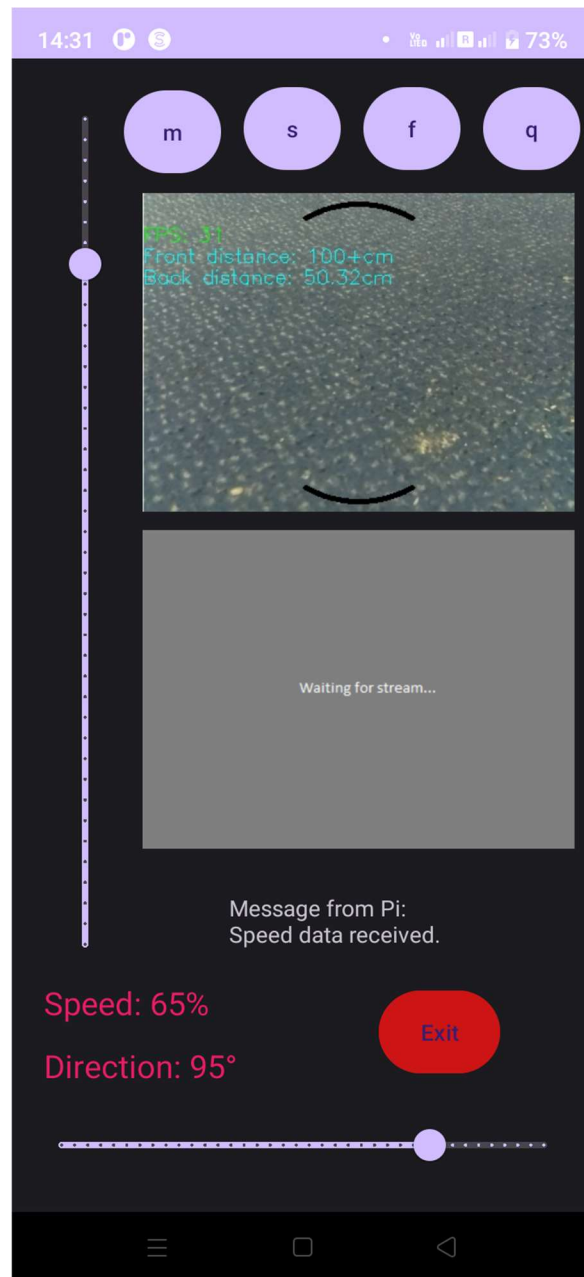
Μια επιπλέον λειτουργία είναι η προειδοποίηση σύγκρουσης. Αν οι μπροστινή ή όπισθεν αποστάσεις γίνουν 15 εκατοστά και λιγότερο, τότε θα παρουσιαστεί προειδοποίηση στην ροή του βίντεο, πως ένα αντικείμενο βρίσκεται αρκετά κοντά στο όχημα, και κινδυνεύουμε να συγκρουστούμε με το όχημα. Στην χειροκίνητη κίνηση, το όχημα δεν θα λάβει κάποια δράση για την αποφυγή σύγκρουσης. Ο χρήστης της εφαρμογής είναι πλήρως υπεύθυνος για την κίνηση του οχήματος. Το όχημα απλά παραδίδει σήματα και προειδοποιήσεις για το άμεσο περιβάλλον του.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 5.1.2 Η κινητή εφαρμογή με τους δείκτες απόστασης και την προειδοποίηση σύγκρουσης

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 5.1.3 Η κινητή εφαρμογή κατά την διάρκεια χειροκίνητης οδήγησης

Για την έξοδο από αυτήν την λειτουργία, ο χρήστης μπορεί να πατήσει “q” για να μπει το Raspberry Pi σε λειτουργία αναμονής, “e” για να γίνει έξοδος από τον κώδικα, ή “s”, “f” για να μπει το Raspberry Pi σε λειτουργία διαφορετικής κίνησης. Την στιγμή που ο χρήστης βγει από την λειτουργία χειροκίνητης κίνησης, τότε οι μπάρες θα πάρουν τιμή 0 και ύστερα, θα απενεργοποιηθούν.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

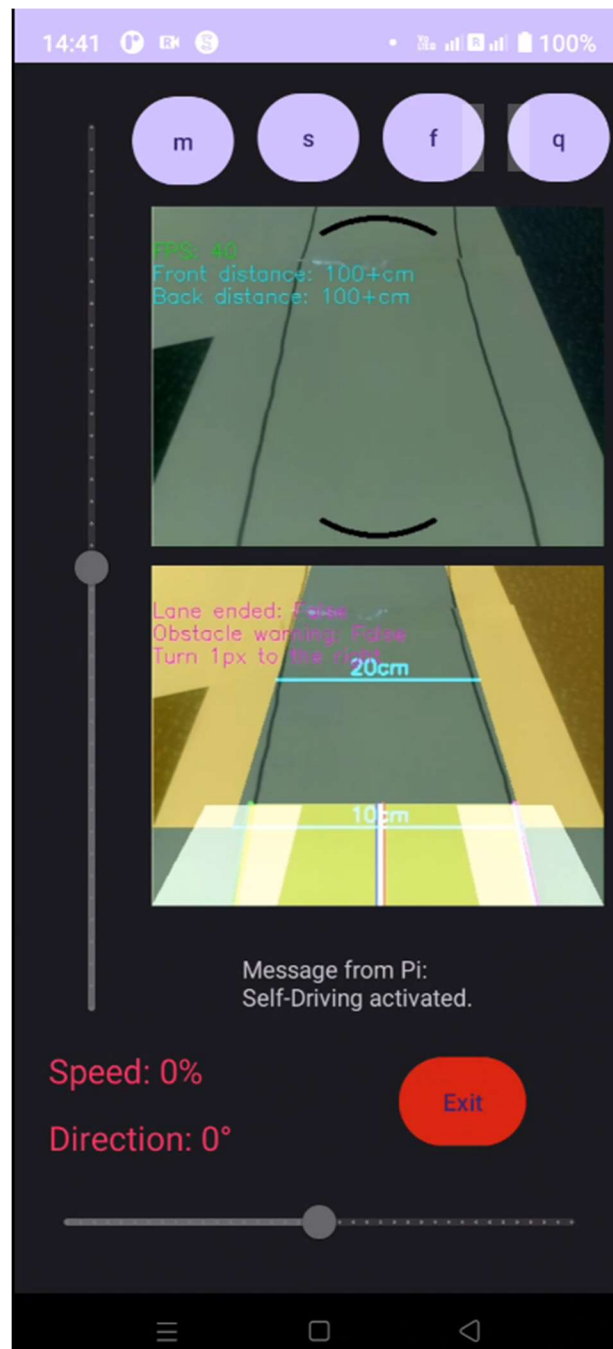
5.1.2 Λειτουργία εντοπισμού λωρίδας “s”

Όταν ο χρήστης μπει σε αυτήν την λειτουργία, αμέσως ενεργοποιείται η δεύτερη ροή βίντεο. Σε αυτήν την ροή παρουσιάζεται το βίντεο που παράγεται από τον αλγόριθμο εντοπισμού λωρίδας. Ο αλγόριθμος έχει την ικανότητα να εντοπίσει λωρίδες κανονικού δρόμου, όπως και αυτοσχέδιες λωρίδες από χαρτί A4 και τις γραμμές της λωρίδας να είναι ζωγραφισμένες με μαρκαδόρο.



Εικόνα 4.1.4 Ο αλγόριθμος κατά την διάρκεια εντοπισμού πραγματικής λωρίδας

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 5.1.5 Η κινητή εφαρμογή κατά την διάρκεια εντοπισμού αυτοσχέδιας λωρίδας

Όπως φαίνεται στην Εικόνα 4.1.5, στην ροή που εμφανίζεται στο κάτω πλαίσιο, τα σημεία στα οποία αναγνωρίζονται τα όρια της λωρίδας χρωματίζονται πράσινα και ροζ για την αριστερή και δεξιά αντίστοιχα πλευρά. Οι περιοχές στις οποίες ο αλγόριθμος θα εντοπίσει γραμμές της λωρίδας, εμφανίζονται ως λευκά ημιδιαφανές ορθογώνια. Επίσης, τα πορτοκαλί ημιδιαφανές ορθογώνια είναι υπεύθυνα για τον εντοπισμό εμποδίων στην μπροστινή πλευρά του οχήματος. Οι δείκτες του κέντρου του οχήματος και του κέντρου της λωρίδας φαίνονται ως μπλέ και κόκκινες γραμμές αντίστοιχα, στην κάτω πλευρά της ροής. Στο πάνω μέρος της ροής φαίνονται και οι εξής πληροφορίες:

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

- “Lane ended”: Αυτός ο δείκτης ενημερώνει τον χρήστη όταν το όχημα εντοπίσει το τέλος της λωρίδας που προσπαθεί να ακολουθήσει. Η τιμή του δείκτη είναι “True” ή “False”. Όταν η τιμή είναι “False”, τότε το όχημα ακόμη γνωρίζει πως υπάρχει διελεύσιμη λωρίδα και θα συνεχίσει να κινείται. Όταν το όχημα δεν μπορεί πια να “δει” την λωρίδα, τότε ο δείκτης θα πάρει την τιμή “True”. Σε αυτήν την περίπτωση το όχημα θα φρενάρει και θα περιμένει από τον χρήστη διαφορετική εντολή κίνησης, ή θα παραμείνει στην αυτή κατάσταση αναμονής έως ότου εντοπιστεί λωρίδα.
- “Obstacle warning”: Αυτός ο δείκτης ενημερώνει τον χρήστη όταν το όχημα βρίσκεται σε πορεία σύγκρουσης με κάποιο εμπόδιο πάνω στην λωρίδα. Παρομοίως με τον δείκτη “Lane ended”, ο παρόν δείκτης επίσης παίρνει τιμές “True” ή “False”. Οι αισθητήρες απόστασης του Arduino συνεχώς επιστρέφουν την μπροστινή και οπίσθια απόσταση από το όχημα. Όταν η μπροστινή απόσταση γίνει 12 εκατοστά ή λιγότερη, τότε ο δείκτης Obstacle warning θα γίνει True. Σε αυτήν την περίπτωση το όχημα θα φρενάρει, και θα παραμείνει ακίνητο μέχρι το εμπόδιο να αφαιρεθεί ή ο χρήστης δώσει εντολή διαφορετικής κίνησης. Σε όλες τις άλλες περιπτώσεις, ο δείκτης έχει τιμή False.
- Turn amount: Αυτό το κομμάτι αναφέρει την απόσταση του κέντρου της λωρίδας από το κέντρο του οχήματος. Ανάλογα με την τοποθεσία του οχήματος μέσα στην λωρίδα, ενημερώνει τον χρήστη πόσο δεξιά ή αριστερά πρέπει να μετακινηθεί το όχημα για να βρεθεί στο κέντρο της λωρίδας.

Για την έξοδο από αυτήν την λειτουργία, όπως και στον χειροκίνητο έλεγχο, ο χρήστης μπορεί να πατήσει οποιοδήποτε κουμπί είναι διαθέσιμο, εκτός από το “s”, καθώς το όχημα ήδη οδηγεί αυτόνομα.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5.1.3 Λειτουργία ελεύθερης περιπλάνησης “F”

Όταν ο χρήστης μπει σε αυτήν την λειτουργία, η ροή βίντεο που εμφανίζεται στο κάτω πλαίσιο σταματά, καθώς ο αλγόριθμος εντοπισμού λωρίδας παύει την λειτουργία του. Σε αυτήν την λειτουργία, μόνο η πάνω ροή παρουσιάζεται.

Σε αυτήν την μέθοδο κίνησης, το όχημα θα κινηθεί χωρίς κάποια συγκεκριμένη κατεύθυνση. Ο κύριος στόχος όμως είναι η αποφυγή εμποδίων. Το όχημα θα ξεκινήσει να κινείται προς τα εμπρός με πλήρης ταχύτητα. Κατά την διάρκεια της οδήγησης, οι αισθητήρες απόστασης συνεχώς σαρώνουν τον χώρο εμπρός και πίσω από το όχημα.

Αν κατά την οδήγηση το όχημα εντοπίσει ένα εμπόδιο να βρίσκεται σε πορεία σύγκρουσης από τα εμπρός, σε απόσταση 25 εκατοστών, τότε το όχημα θα ξεκινήσει τους ελιγμούς αποφυγής σύγκρουσης. Αρχικά θα ξεκινήσει να στρίβει δίνοντας στους κινητήρες ένα ποσό “στροφής” που πρέπει να εκτελέσουν. Όσο η μπροστινή απόσταση από το όχημα δεν μειώνεται, τόσο περισσότερο είναι και το ποσό της στροφής των κινητήρων. Αν το εμπόδιο δεν βρίσκεται μπροστά από το όχημα πια, τότε το όχημα συνεχίζει την ομαλή του κίνηση προς τα εμπρός.

Αν το εμπόδιο ακόμη βρίσκεται μπροστά από το όχημα, ολοένα και πλησιάζει, και φτάσει να βρίσκεται σε απόσταση 10 εκατοστών, τότε το όχημα θα φρενάρει επιτόπου. Ύστερα θα ελεγχτεί η απόσταση της οπίσθιας πλευράς του οχήματος, άμα είναι δυνατή η κίνηση προς τα πίσω. Αν υπάρχει εμπόδιο σε απόσταση 25 εκατοστών ή λιγότερο πίσω από το όχημα, τότε θα ξεκινήσει να περιστρέφεται γύρω από τον εαυτό του σαρώνοντας το άμεσο περιβάλλον για τυχόν σημεία στα οποία μπορεί το όχημα να πορευτεί ελεύθερα. Αν η πίσω πλευρά είναι κενή από εμπόδια, τότε το όχημα, όσο η πίσω πλευρά συνεχίζει να είναι κενή από εμπόδια, θα κινηθεί προς τα όπισθεν μέχρι η εμπρός απόσταση να γίνει τουλάχιστον 25 εκατοστά. Όταν αυτό πραγματοποιηθεί, τότε το όχημα θα περιστραφεί γύρω από τον εαυτό του, ψάχνοντας για σημεία χωρίς εμπόδια. Αν βρεθεί αυτό το σημείο, τότε ξεκινά την ομαλή κίνησή του προς τα εμπρός.

Η έξοδος από την λειτουργία αυτή γίνεται παρόμοια με τις άλλες δύο λειτουργίες.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5.2 Λειτουργίες του Raspberry Pi

5.2.1 Επικοινωνία με εξωτερικές συσκευές

Για το όχημα, το Raspberry Pi αποτελεί το κέντρο επικοινωνίας μεταξύ του χρήστη μέσω της κινητής εφαρμογής, και του Arduino.

Για την διασύνδεση του Raspberry Pi με την κινητή συσκευή, χρησιμοποιήθηκε σύνδεση WiFi. Το Raspberry Pi εξέπεμπε σήμα ασύρματης σύνδεσης “Hotspot” στο οποίο δίκτυο μπορούσε η κινητή συσκευή να συνδεθεί και να ξεκινήσει η επικοινωνία τους.

Η επικοινωνία πραγματοποιήθηκε με την χρήση Sockets. Το Raspberry Pi καθώς είναι ο πομπός του WiFi, περιέχει τον διακομιστή στον οποίο θα συνδεθεί η κινητή συσκευή για να ξεκινήσει η επικοινωνία. Στον δίαυλο που δημιουργείται, η κινητή συσκευή στέλνει εντολές κίνησης, εντολές λειτουργίας, όπως και εντολές εξόδου κώδικα. Από την άλλη πλευρά, το Raspberry Pi, με κάθε εντολή που παραλαμβάνει από την κινητή συσκευή, απαντά με ένα μήνυμα το οποίο πιστοποιεί στο κινητό όπως και στον χρήστη πως η επικοινωνία πραγματοποιήθηκε επιτυχώς.

Το Raspberry Pi, παράλληλα περιέχει και έναν δεύτερο διακομιστή ο οποίος είναι υπεύθυνος για την αποστολή ροών βίντεο της κάμερας και την έξοδο του αλγορίθμου εντοπισμού λωρίδας. Οι εικόνες που παράγονται στο Raspberry Pi κωδικοποιούνται σε μορφή Base64 πριν αποσταλούν προς την κινητή συσκευή. Η κωδικοποίηση αυτή μετατρέπει τις εικόνες σε αλφαριθμητικούς χαρακτήρες [A-Za-z0-9+/=]. Με τον τρόπο αυτό, είναι δυνατή η εύκολη διαχείριση της ροής βίντεο σαν String. Έτσι, ανάλογα με την τρέχουσα λειτουργία κίνησης, το Raspberry Pi έχει την ικανότητα να στείλει μόνο το βίντεο που πρόκειται για το πάνω πλαίσιο της κινητής εφαρμογής, και όταν πατηθεί το κουμπί ενεργοποίησης του εντοπισμού της λωρίδας, απλά γίνεται “επικόλληση” του String των επεξεργασμένων εικόνων στο String των εικόνων που παράγονται από την κάμερα.

Το Raspberry Pi είναι επίσης υπεύθυνο για την επικοινωνία με το Arduino. Εδώ, η ανταλλαγή πληροφορίας γίνεται μέσω USB καλωδίου, που εξυπηρετήσει διπλό σκοπό. Ο ένας είναι η τροφοδοσία του Arduino, και ο δεύτερος είναι η αποστολή και λήψη εντολών κίνησης και τιμές απόστασης που μετρήθηκαν από τους αισθητήρες συνδεδεμένοι με το Arduino.

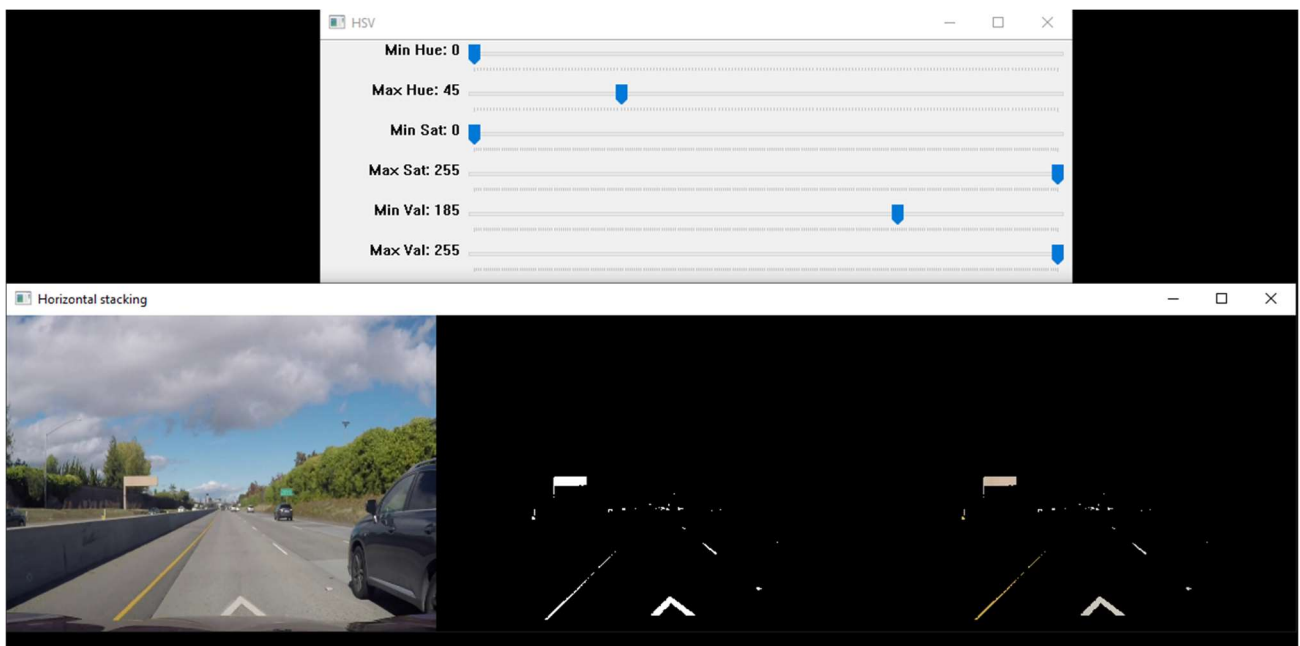
Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5.2.2 Εντοπισμός λωρίδας

- 1ο Βήμα

Ο εντοπισμός της λωρίδας απαιτεί κάποια συγκεκριμένα βήματα για την ολοκλήρωσή του.

Αρχικά, πρέπει να εντοπιστεί ο χώρος ενδιαφέροντος, που στην τωρινή περίπτωση είναι μια λωρίδα λευκού χρώματος. Για αυτήν την διαδικασία χρησιμοποιείται εργαλείο καταφλίου χρώματος. Το Raspberry Pi μετατρέπει εικόνες από την ροή βίντεο της κάμερας από BGR σε HSV, καθώς διάφορα φάσματα χρωμάτων είναι πιο εύκολα εντοπιζόμενα σε HSV μορφή. Σε BGR μορφή, μια μικρή αλλαγή στις τιμές χρωμάτων αλλάζει δραστικά το φάσμα εντοπιζόμενων χρωμάτων.



Εικόνα 5.2.1 Εργαλείο εντοπισμού χρωμάτων σε εικόνα

Με μπάρες του εργαλείου μπορούμε να τροποποιήσουμε τις τιμές απόχρωσης, κορεσμού και φωτεινότητας (Hue, Saturation, Lightness) τις οποίες θέλουμε να φιλτράρουμε. Πειραματικά, αλλάζουμε τις τιμές των μπαρών, και ανάλογα με τις εικόνες που αλλάζουν χρώμα απευθείας, χωρίς να χρειαστεί να γίνει επανεκκίνηση του εργαλείου, εντοπίζουμε το χρώμα που θέλουμε. Με αυτό το εργαλείο, η λωρίδα που εντοπίζουμε χρωματίζεται λευκή, και υπόλοιπα χρώματα εμφανίζονται ως μαύρα.

Με αυτό το εργαλείο γίνεται ο εντοπισμός κάθε είδους χρώματος. Αν η λωρίδα που πρέπει να εντοπιστεί είναι πράσινης ή κίτρινης απόχρωσης, τότε πρέπει απλά να αλλάξουν οι τιμές των μπαρών στο εργαλείο. Έτσι το όχημα έχει την ικανότητα να εντοπίσει κάθε είδους αντικείμενα, οποιουδήποτε χρώματος.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

- 2ο Βήμα

Στην συνέχεια, πρέπει να ορίσουμε τις περιοχές ενδιαφέροντος (Region of Interest, ROIs) στην εικόνα που επεξεργαζόμαστε. Οι 4 περιοχές που μας ενδιαφέρουν είναι δύο για την αναγνώριση γραμμών λωρίδας, μια περιοχή για κάθε πλευρά, και δύο περιοχές για την αναγνώριση εμποδίων στην μπροστινή περιφέρεια του οχήματος, επίσης μια περιοχή για κάθε πλευρά.

Για τις περιοχές ενδιαφέροντος γραμμών λωρίδας, ορίζουμε δύο τρίγωνα των οποίων μια πλευρά βρίσκεται στο σημείο που αναμένεται να εντοπισθεί γραμμή λωρίδας. Τα σημεία που ορίζουν την πλευρά αυτή μεταβάλλονται ανάλογα με το ύψος και την μετατόπιση της κάμερας, όπως και το είδος του δρόμου, καθώς ένας τοπικός δρόμος μιας αστικής γειτονιάς δεν θα είναι ίδιου μεγέθους και τύπου σε σχέση με έναν δρόμο μιας μεγάλης αστικής αρτηρίας. Η κορυφή των τριγώνων συμπίπτουν στο ίδιο σημείο. Η συντεταγμένες του σημείου αυτού επίσης δεν είναι σταθερές. Η $y'y$ συντεταγμένη του σημείου εξαρτάται από την κλίση των γραμμών της λωρίδας, όπως και την τοποθεσία της κάμερας στο όχημα και η $x'x$ συντεταγμένη του σημείου συμπίπτει με αυτήν του κέντρου της λωρίδας. Σε συνδυασμό με αυτά τα σημεία, εισάγουμε και ένα όριο ύψους στα τρίγωνα που έχουμε σχηματίσει. Έτσι από τα τρίγωνα, σχηματίζουμε δύο τραπέζια των οποίων το σχήμα αλλάζει ανάλογα με το κέντρο της λωρίδας. Με αυτό μπορούμε να προκαλέσουμε τα τραπέζια αυτά να “ακολουθούν” την κάμψη της λωρίδας.

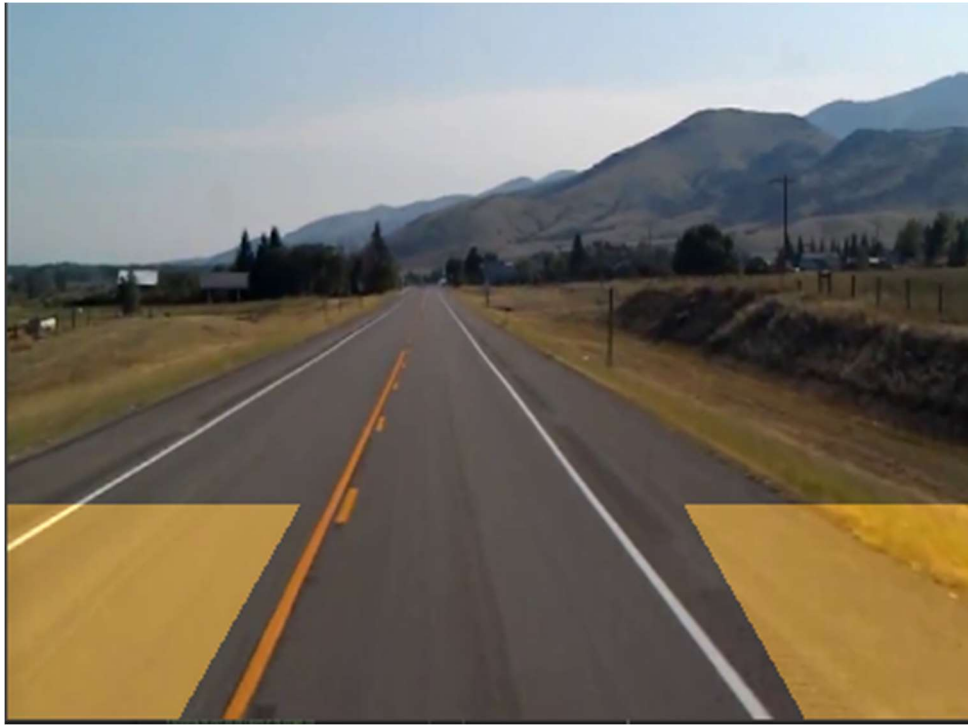


Εικόνα 5.2.2 Παράδειγμα περιοχής ενδιαφέροντος γραμμών λωρίδας

Ύστερα, προθέτουμε την περιοχή ενδιαφέροντος εμποδίων. Παρομοίως με την περιοχή ενδιαφέροντος της λωρίδας, τα σημεία των περιοχών αυτών δεν είναι σταθερά, καθώς εξαρτώνται από την τοποθέτηση της κάμερας. Σε κάθε περίπτωση, οι περιοχές αυτές έχουν σχήμα τραπέζιου τα οποία βρίσκονται ακριβώς έξω από τις λωρίδες του δρόμου. Έτσι δίνουμε την ικανότητα στο όχημα να έχει επίγνωση του κοντινού του

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

περιβάλλοντος για την παρουσία εμποδίων. Εφόσον οι περιοχές αυτές εξαρτώνται από τις λωρίδες του δρόμου, πρέπει να γίνεται πειραματική θέση των σημείων με κάθε διαφορετικό περιβάλλον που βρίσκεται το όχημα.

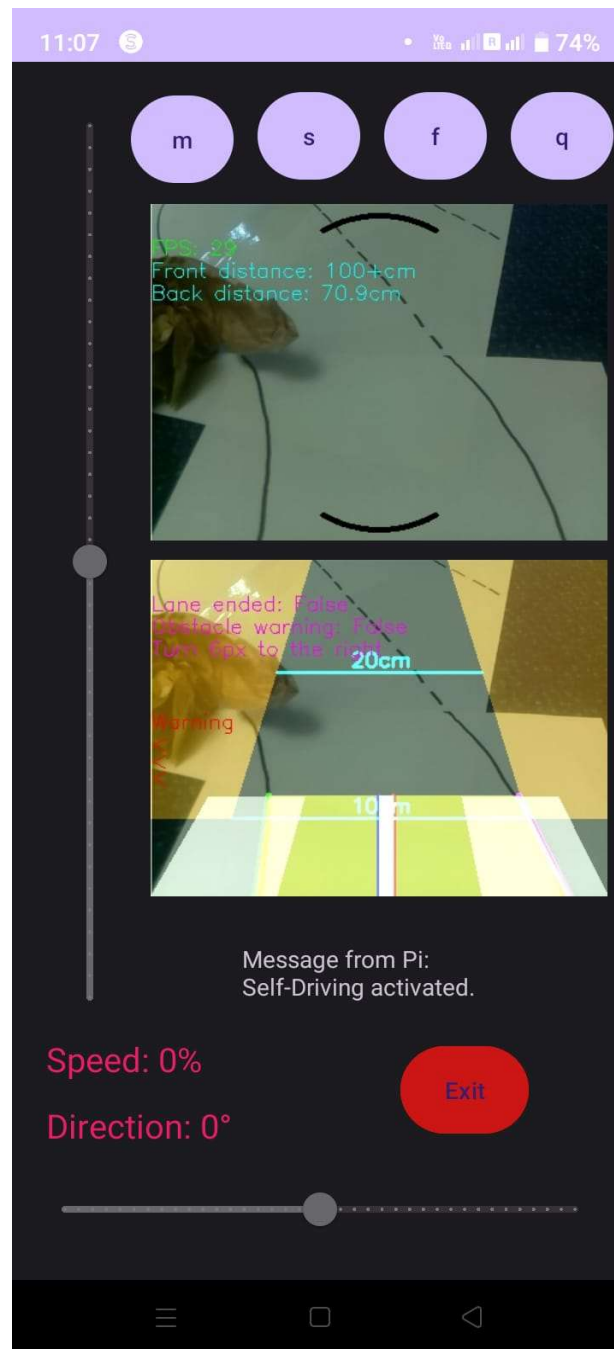


Εικόνα 5.2.3 Παράδειγμα περιοχής ενδιαφέροντος εμποδίων

Οι περιοχές αυτές δεν επεκτείνονται προς το κέντρο της λωρίδας, καθώς το όχημα που έχει κατασκευαστεί διαθέτει και αισθητήρες απόστασης, οι οποίοι έχουν τοποθετηθεί έτσι ώστε να μπορούν να μετρούν απόσταση από την εμπρόσθια οπίσθια πλευρά του οχήματος. Έτσι, για να μην είναι καλυμμένη η εμπρόσθια πλευρά μονάχα από μια πολύ μικρή γωνία “αίσθησης” που διαθέτει ο αισθητήρας απόστασης, υλοποιούνται και η ανίχνευση εμποδίων στις παράπλευρες περιοχές.

Οι περιοχές ενδιαφέροντος λωρίδας και εμποδίων εφαρμόζονται πάνω στην ασπρόμαυρη εικόνα που έχουμε κατασκευάσει στο 1ο βήμα. Έτσι μπορούμε να οριοθετήσουμε ποια κομμάτια της εικόνας θέλουμε να επεξεργαστούμε, και με ποιόν τρόπο.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

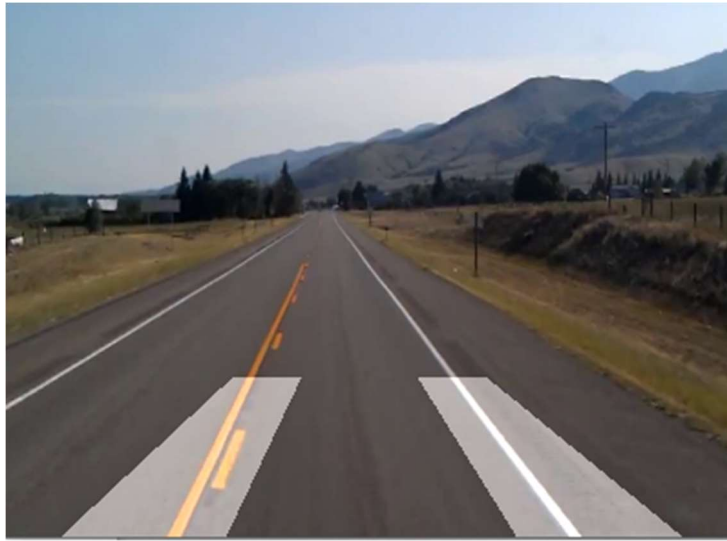


Εικόνα 5.2.4 Παράδειγμα εντοπισμού εμποδίου

- 3ο Βήμα

Το επόμενο βήμα για την αναγνώριση λωρίδας είναι ο εντοπισμός των γραμμών. Ο εντοπισμός των γραμμών πρέπει να γίνει σε μια ασπρόμαυρη εικόνα, όπως έχουμε κατασκευάσει στο 2ο βήμα. Σε αυτό το στάδιο, έχουμε μια ασπρόμαυρη εικόνα με συγκεκριμένες περιοχές στις οποίες πρέπει να εντοπίσουμε γραμμές.

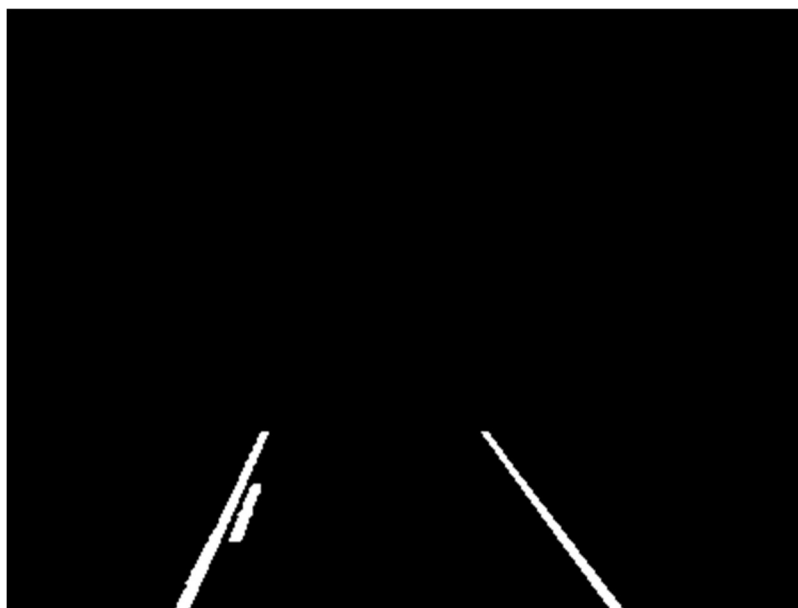
Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 5.2.5 Παράδειγμα αρχικής εικόνας με το ROI της λωρίδας

Ο εντοπισμός γραμμών γίνεται με τα εξής στάδια:

1. Εφαρμογή Γκαουσιανού θολώματος (Gaussian Blur) στην εικόνα. Το βήμα αυτό εξουδετερώνει τον θόρυβο που είναι παρόν στα εικονοστοιχεία της εικόνας λόγω της κακής ποιότητας της κάμερας που χρησιμοποιείται ή από τον τρόπο συμπίεσης βίντεο στην περίπτωση εφαρμογής του αλγορίθμου σε βίντεο αντί για πραγματική χρήση κάμερας. Έτσι μπορούμε να μειώσουμε το ποσό των “άκρων” που προκύπτουν από τους παραπάνω λόγους, και να κρατήσουμε μόνο τις “άκρες” που εμφανίζονται φυσικά.



Εικόνα 5.2.6 Φιλτράρισμα χρωμάτων και εφαρμογή Gaussian Blur στο ROI λωρίδας

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

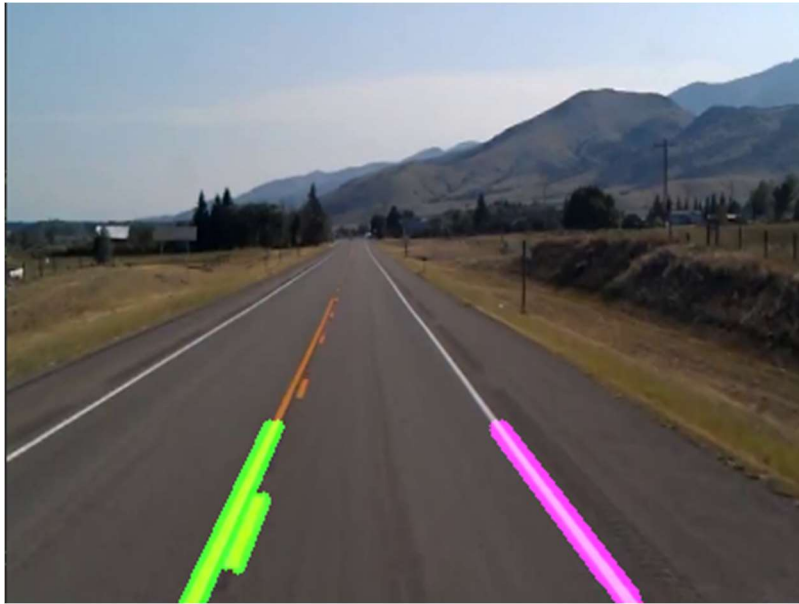
2. Εφαρμογή αλγορίθμου Canny. Ο αλγόριθμος αυτός εντοπίζει τις “άκρες” σε μια εικόνα, οι οποίες στην πραγματικότητα είναι σημεία στα οποία υπάρχει μεγάλη διαφορά μεταξύ των τιμών χρώματος γειτονικών εικονοστοιχείων. Καθώς οι διαχωριστικές γραμμές μιας λωρίδας έχουν έντονο χρώμα, και σχεδόν πάντα, διαφορετική απόχρωση από το υπόλοιπο οδόστρωμα, μπορούμε να εντοπίσουμε τις περιοχές στις οποίες βρίσκονται οι διαχωριστικές γραμμές της λωρίδας, και κατ’ επέκταση, το κέντρο της λωρίδας. Επιπλέον, με αυτόν τον τρόπο μπορούμε να εντοπίσουμε εμπόδια στις παράπλευρες περιοχές ενδιαφέροντος. Καθώς έχουμε αγνοήσει το χρώμα του οδοστρώματος, αν κατά την διάρκεια του αλγορίθμου παρουσιαστεί ένα αντικείμενο με διαφορετικό χρώμα σε σχέση με το οδόστρωμα, τότε θα εντοπισθούν “άκρες” και στην θέση του αντικειμένου που μπήκε στην περιοχή του ενδιαφέροντος εμποδίων.



Εικόνα 5.2.7 Εφαρμογή αλγορίθμου Canny

3. Μετά την εφαρμογή του αλγορίθμου Canny, εφαρμόζουμε τον μετασχηματισμό Hough στην εικόνα με τις ακμές. Ο αλγόριθμος αυτός εντοπίζει γραμμές σε μια δεδομένη εικόνα. Εφαρμόζοντας τον μετασχηματισμό αυτόν, μπορούμε να παράξουμε το σύνολο των γραμμών που είναι παρόν στην εικόνα εφαρμογής. Οι γραμμές παρουσιάζονται με την μορφή τεσσάρων τιμών που αναπαριστούν τις x' και y' συντεταγμένες των δύο άκρων μιας ευθείας. Γνωρίζοντας τις απαραίτητες συντεταγμένες για την αναπαράσταση των γραμμών που μας ενδιαφέρουν, μπορούμε να τις αποτυπώσουμε πάνω στην εικόνα που μας ενδιαφέρει.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 5.2.8 Αναπαράσταση γραμμών από τον μετασχηματισμό Hough

Παρομοίως με την αναγνώριση των διαχωριστικών γραμμών της λωρίδας, ο μετασχηματισμός Hough θα εφαρμοστεί και για την αναγνώριση γραμμών για στις περιοχές ενδιαφέροντος εμποδίων.

4. Το επόμενο βήμα περιλαμβάνει την σύντηξη των πολλαπλών μικρών γραμμών που έχουμε εντοπίσει για τις διαχωριστικές γραμμές της λωρίδας, και η αναπαράστασή τους ως δύο γραμμές, μια για κάθε πλευρά της λωρίδας. Οι δύο γραμμές αυτές είναι ο μέσος όρος των γραμμών τις αντίστοιχης πλευράς της λωρίδας. Με αυτόν τον τρόπο έχουμε την ικανότητα να αναπαραστήσουμε τις γραμμές της λωρίδας με πιο κομψό τρόπο, όπως και να υπολογίσουμε το κέντρο της λωρίδας πιο εύκολα, έχοντας μόνο 2 γραμμές ως τα όρια της λωρίδας.



Εικόνα 5.2.9 Δημιουργία μέσου όρου γραμμών

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5. Τέλος, πρέπει να εντοπίσουμε το κέντρο της λωρίδας , και την οπτική αναπαράσταση της διαδικασίας. Αρχικά, μπορούμε εύκολα να βρούμε το κέντρο της λωρίδας, ως το μέσο σημείο μεταξύ των τελικών σημείων την δύο γραμμών που έχουμε αναπαραστήσει στην παραπάνω εικόνα. Καθώς οι γραμμές αυτές θα βρίσκονται συνεχώς πάνω στις διαχωριστικές γραμμές της λωρίδας, ανεξαρτήτως από το πού βρίσκεται το όχημα, μπορούμε να έχουμε συνεχώς επίγνωση του πόσο πρέπει να στρίψει το όχημα έτσι ώστε να βρεθεί στο κέντρο της λωρίδας.
- Για την αναπαράσταση της διαδικασίας εντοπισμού λωρίδας, χρωματίζουμε τις ξεχωριστές περιοχές ενδιαφέροντος με διαφορετικό χρώμα, όπως και τον χώρο μεταξύ των δύο γραμμών για να γίνει πιο ορατός ο δρόμος που ακολουθεί το όχημα. Σε αυτόν τον χώρο, επίσης χρωματίζουμε και με δύο στήλες, το κέντρο του οχήματος ως μπλέ, και το κέντρο της λωρίδας με κόκκινο χρώμα. Η περιοχή ανάμεσα στις δύο στήλες αυτές επίσης χρωματίζεται με έντονο χρώμα για την οπτική απεικόνιση του ποσού “στροφής” που πρέπει το όχημα να πάρει για να βρεθεί στο κέντρο της λωρίδας.
- Προσθέτουμε επίσης και τις πληροφορίες τέλους λωρίδας και προειδοποίησης εμποδίων, μαζί με λεκτική ένδειξη του ποσοστού στροφής που πρέπει να παρθεί.
- Στο τέλος έχουμε την εξής εικόνα:



Εικόνα 5.2.10 Τελικό στιγμιότυπο του εντοπισμού λωρίδας

Αξίζει να σημειωθεί πως κατά την διάρκεια λειτουργίας της αυτόνομης οδήγησης, τα βήματα αυτά θα εκτελούνται συνεχώς, με είσοδο τις εικόνες που παράγονται από την κάμερα του οχήματος, και ως έξοδο εικόνα όπως φαίνεται στην Εικόνα 5.2.10.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5.3. Λειτουργίες του Arduino

Όπως αναφέρθηκε και στην υποενότητα 5.2.1, το Raspberry Pi και το Arduino επικοινωνούν μεταξύ τους μέσω σειριακής επικοινωνίας με το καλώδιο USB, το οποίο τροφοδοτεί και το Arduino.

Μέσω αυτής της επικοινωνίας, το Arduino παραλαμβάνει τις εντολές κίνησης από το Raspberry, και επιστρέφει μήνυμα επιτυχημένης επικοινωνίας με την εντολή που πρόκειται να εκτελέσει, όπως και τις μετρήσεις των αισθητήρων απόστασης που έχουν τοποθετηθεί στην εμπρόσθια και πίσω πλευρά του οχήματος.

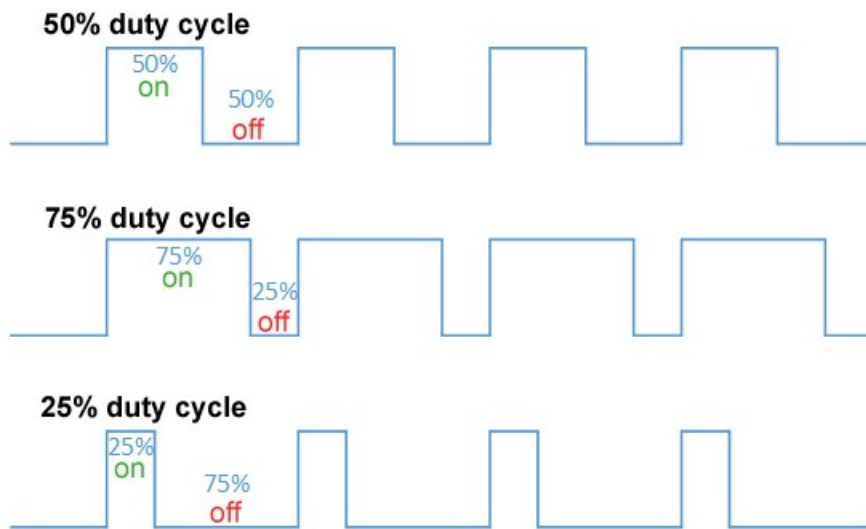
Την στιγμή που το Arduino παραλάβει τις εντολές κίνησης, αρχίζει η διαδικασία επεξεργασίας των εντολών με σκοπό την τελική εφαρμογή τους. Τα παρακάτω βήματα αποτελούν την ροή πληροφορίας που ακολουθούν οι εντολές:

1. Το Arduino παραλαμβάνει τις εντολές με την εξής μορφή: “rfs_50_-43”.
Το r είναι το αρχικό γράμμα του πακέτου με το οποίο ελέγχεται άμα έχει γίνει σωστή παραλαβή πακέτου πληροφορίας. Το δεύτερο γράμμα του πακέτου “f ή b” σηματοδοτεί την εμπρόσθια (forward) ή όπισθεν (backward) κίνηση του οχήματος. Το τρίτο γράμμα το οποίο μπορεί να είναι ‘f’, ‘s’, ή ‘m’, από τις τρεις λειτουργίες κίνησης που μπορεί να έχει το όχημα, “Free roaming” κίνηση ελεύθερης περιπλάνησης, “Self-Driving”, αυτόνομη οδήγηση και κεντράρισμα στην εντοπισμένη λωρίδα και “Manual Driving”, χειροκίνητη οδήγηση με την χρήση της κινητής εφαρμογής για την δημιουργία εντολών κίνησης από τον χρήστη. Οι τελευταίοι δύο αριθμοί δίνουν τις τιμές της ταχύτητας και κατεύθυνσης. Στο τωρινό παράδειγμα, το Arduino παραλαμβάνει σωστό πακέτο κίνησης, με εμπρόσθια κίνηση με την αυτόνομη κίνηση ενεργοποιημένη, με 50% δύναμη στους κινητήρες και κατεύθυνση -43, που σημαίνει ότι το όχημα θα στρίψει αριστερά.
2. Στο δεύτερο βήμα, το Arduino θα διασπάσει την εντολή στα συστατικά μέρη της για να γίνει η σωστή ανάγνωσή της, διαχωρίζοντας την εντολή με την χρήση των ‘_’ χαρακτήρων.
3. Ύστερα μετατρέπονται τα ψηφία ταχύτητας και κατεύθυνσης από αλφαριθμητικούς χαρακτήρες σε ακέραιους αριθμούς.
4. Τέλος ελέγχονται οι λειτουργίες που πρέπει να εκτελεστούν έτσι ώστε το Arduino να ελέγξει ορθώς τους κινητήρες μέσω του οδηγητή L298N.

Η τελική εφαρμογή των εντολών αλλάζει ανάλογα με τον τωρινό τρόπο κίνησης (m, s, f) όπως και εξωτερικούς παράγοντες.

Όταν το όχημα κινείται με αυτόνομη ή χειροκίνητη οδήγηση, τότε η εφαρμογή είναι απλή. Το Arduino θα επαναχαράξει την τιμή της ταχύτητας από 0 έως 100 που παραλαμβάνεται από το Raspberry σε τιμές από 0 έως 255, συνυπολογίζοντας το ποσοστό της στροφής που πρέπει να παρθεί. Ύστερα, ο αριθμός που προκύπτει, εφαρμόζεται πάνω στον οδηγητή L298N με την μορφή PWM (Pulse Width Modulation) σημάτων. Καθώς ο οδηγητής έχει μόνιμη τροφοδοσία, το Arduino αποφασίζει πόσο “συχνά” οι κινητήρες θα λαμβάνουν τροφοδοσία από τον οδηγητή, δίνοντας έτσι την αυξομείωση της ταχύτητας που παρατηρείται στους κινητήρες.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 5.3.1 Διαμόρφωση εύρους παλμών PWM

Πηγή: https://commons.wikimedia.org/wiki/File:Duty_Cycle_Examples.png

Ανάλογα με την τιμή της ταχύτητας από 0 έως 255, τότε και οι κινητήρες θα τροφοδοτούνται αναλόγως από 0 έως 100% της περιόδου ενός κύκλου, όπως παρατηρείται στην Εικόνα 5.3.1.

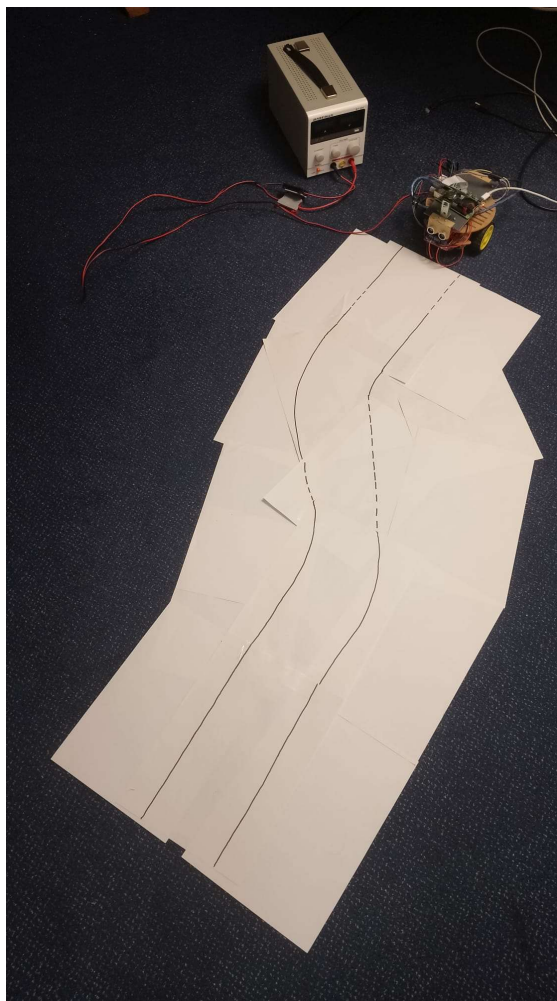
Όταν το όχημα κινείται υπό την λειτουργία της ελεύθερης περιπλάνησης, τότε η κίνηση του οχήματος επηρεάζεται και από τις μετρήσεις που έχουν παρθεί από τους αισθητήρες απόστασης. Τα παρακάτω βήματα αποτελούν την σειρά ελέγχων που εκτελεί το Arduino για την αποφυγή εμποδίων:

1. Αρχικά, το Arduino θα προσπαθήσει να κινηθεί προς τα εμπρός. Έτσι, ελέγχεται η μέτρηση του εμπρόσθιου αισθητήρα. Αν η τιμή του αισθητήρα είναι μεγαλύτερη από την ασφαλής απόσταση που ορίζουμε προγραμματιστικά, τότε το όχημα θα προχωρήσει προς τα εμπρός έως ότου η μέτρηση του μπροστινού αισθητήρα γίνει μικρότερη από την ασφαλής απόσταση.
2. Αν το όχημα βρεθεί στην περίπτωση ενός εμποδίου να βρίσκεται σε ασφαλής απόσταση, αλλά όχι στην απόσταση κινδύνου, τότε το όχημα θα προσπαθήσει να αποφύγει το εμπόδιο στρίβοντας δεξιά ή αριστερά, ταυτόχρονα με την εμπρόσθια κίνησή του, μέχρι το εμπόδιο να αποφευχθεί ή το εμπόδιο να μπει στην απόσταση κινδύνου.
3. Αν ένα εμπόδιο μπει στην απόσταση κινδύνου σύγκρουσης με το όχημα, το όχημα θα σταματήσει αμέσως τις κινήσεις του, και θα ξεκινήσει ο έλεγχος της περιμέτρου για πιθανούς τρόπος κίνησης. Αρχικά ελέγχεται η απόσταση του οπίσθιου αισθητήρα απόστασης, αν υπάρχει εμπόδιο σε απόσταση κινδύνου. Αν υπάρχει περιθώριο κίνησης προς τα όπισθεν, τότε το όχημα θα κινηθεί προς τα πίσω έως να βρεθεί σε ένα σημείο στο οποίο η εμπρόσθια απόσταση είναι ασφαλής, όπως και η πίσω απόσταση δεν βρίσκεται στην απόσταση κινδύνου. Σε αυτό το σημείο, το όχημα θα περιστραφεί γύρω από τον εαυτό του μέχρι να βρεθεί μέρος στο οποίο η μπροστινή απόσταση είναι ασφαλής. Μετά από το κομμάτι αυτό, τα βήματα συνεχίζονται κυκλικά.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

5.4 Πειράματα λειτουργιών οχήματος

Οι λειτουργίες του οχήματος δοκιμάστηκαν με την χρήση απλών οικιακών αντικειμένων. Συγκεκριμένα, για την λειτουργία αυτόνομης οδήγησης χρησιμοποιήθηκαν δύο “δρόμοι” κατασκευασμένοι από κανονικά χαρτιά A4 και συγκολλημένα μεταξύ τους με κολλητική ταινία. Οι διαχωριστικές γραμμές του δρόμου σχηματίστηκαν χειροποίητα με την χρήση μαρκαδόρου, όπως φαίνεται παρακάτω.



Εικόνα 5.4.1 Ο πρώτος διάδρομος δοκιμασίας για το όχημα

Ο πρώτος διάδρομος περιέχει απλά μια μικρή καμπύλη στην μέση της, αλλά επίσης περιέχει και διακεκομμένες γραμμές, δείχνοντας την λειτουργία του αλγορίθμου εντοπισμού λωρίδας χωρίς συνεχόμενες διαχωριστικές γραμμές. Ο δρόμος αυτός δεν έχει μεγάλο μήκος καθώς στην παρούσα κατάσταση τα καλώδια που τροφοδοτούν το όχημα ήταν αρκετής μικρού μήκους και υπήρχε περιορισμός στην απόσταση ομαλής οδήγησης χωρίς τα καλώδια να ασκούν πίεση στο όχημα εξαιτίας των μηκών τους, αλλάζοντας την κατεύθυνση πορείας.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



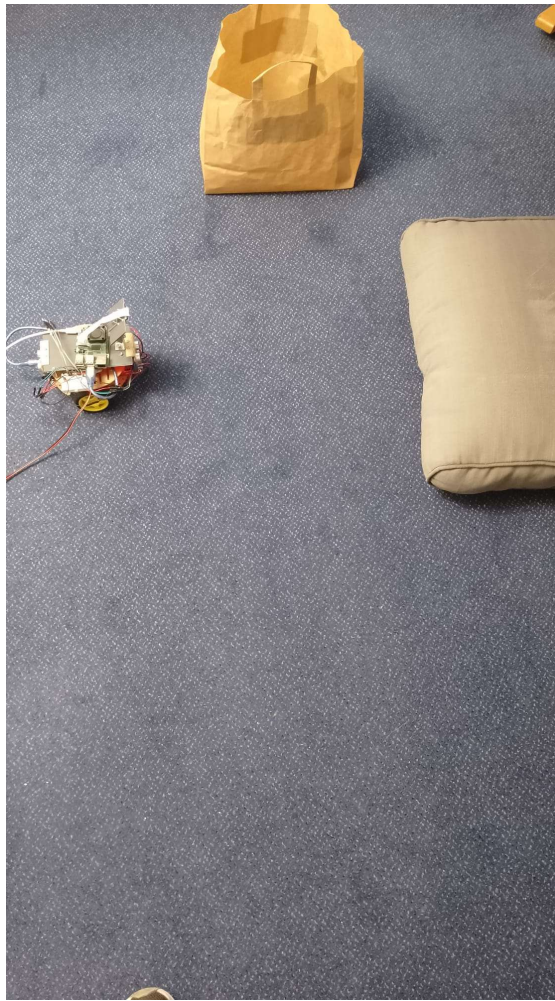
Εικόνα 5.4.2 Ο δεύτερος διάδρομος δοκιμασίας για το όχημα

Παρομοίως με τον πρώτο δρόμο, και ο δεύτερος δεν έχει μεγάλο μήκος για τους ίδιους προαναφερθέντες λόγους. Ο διάδρομος αυτός περιέχει δύο κλειστές καμπύλες στην μέση της και διακεκομμένες γραμμές.

Για την δοκιμή της λειτουργία ελεύθερης πλοήγησης, χρησιμοποιήθηκαν ξανά οικιακά αντικείμενα τα οποία ήταν άμεσα διαθέσιμα, καθώς σε αυτήν την λειτουργία η μόνη απαίτηση που υπάρχει από αυτά είναι η λειτουργία τους ως εμπόδια.

Για τους ίδιους λόγους με τις δοκιμασίες της αυτόνομης οδήγησης, δεν χρησιμοποιήθηκαν πολλαπλά εμπόδια για την δημιουργία “λαβυρίνθου”, καθώς τα καλώδια θα επηρέαζαν την κίνηση του οχήματος.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 5.4.3 Η δοκιμασία της ελεύθερης περιπλάνησης

Σε αυτήν την δοκιμασία το όχημα ξεκινάει την οδήγηση όταν το μαξιλάρι βρίσκεται στην ασφαλής περιοχή. Πλησιάζοντας, το όχημα θα εκτελέσει μανούβρες αποφυγής. Αν ύστερα κινηθεί προς τα δεξιά, τότε το όχημα αποφεύγει το εμπόδιο. Αν κινηθεί προς τα αριστερά, τότε το όχημα θα εντοπίσει αμέσως εμπόδιο στην επικίνδυνη ζώνη, ενεργοποιώντας την λειτουργία σάρωσης του περιβάλλοντος για πιθανές διεξόδους. Στα βίντεο επίδειξης του οχήματος, σε αυτήν την λειτουργία, παρουσιάζονται και απότομα νέα εμπόδια μπροστά από το όχημα, για την επίδειξη της παρούσας λειτουργίας.

Για την δοκιμασία της χειροκίνητης κίνησης, δεν χρειάστηκε η χρήση εμποδίων, καθώς ο έλεγχος του οχήματος εξαρτάται από τον χειριστή.

Οι σχετικές επιδείξεις των λειτουργιών με το πραγματικό όχημα, όπως και δοκιμή λειτουργίας του αλγορίθμου εντοπισμού λωρίδας πραγματικού δρόμου, υπάρχουν σε μορφή βίντεο.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

6. Συμπεράσματα και μελλοντικές βελτιώσεις

6.1 Συμπεράσματα

6.1.1 Συμπεράσματα για το όχημα

Καθώς το όχημα που χρησιμοποιήθηκε, επιλέχθηκε πρωτίστως χάρη στην προσιτή τιμή του και μικρό χρόνο αποστολής, ο διαθέσιμος χώρος για τις πλακέτες απεδείχθη ανεπαρκής, κάνοντας κάποιες τροποποιήσεις αναγκαίες.

Για παράδειγμα, για την κάμερα, καθώς δεν υπήρχε κατάλληλο σημείο τοποθέτησης, έπρεπε να χρησιμοποιηθεί μια υπερυψωμένη μεταλλική βάση, στην οποία προσαρτήθηκε κομμάτι πλαστικού PVC. Στο κομμάτι αυτό τοποθετήθηκε η κάμερα με την χρήση 4 βιδών.

Παρομοίως για τις πλακέτες του Raspberry Pi, Arduino, L298N, μπαταριών και μικρής πλακέτας σύνδεσης καλωδίων, ο χώρος 2 ορόφων που προσέφερε το όχημα δεν ήταν επαρκής. Έτσι, χρησιμοποιήθηκε κομμάτι PVC πλαστικού για την δημιουργία τρίτου ορόφου.

Επίσης, αξίζει να σημειωθεί πως οι κινητήρες με τους οποίους το όχημα ήλθε, απεδείχθησαν μη επαρκείς για το βάρος του οχήματος με μπαταρίες, καθώς έπρεπε να χρησιμοποιηθεί ένα powerbank για την τροφοδοσία του Raspberry Pi, όπως και ξεχωριστές μπαταρίες, συνδεδεμένες σε σειρά με το powerbank, για την τροφοδοσία των κινητήρων και του L298N. Καθώς η κίνηση του οχήματος ήταν δυνατή, έπρεπε να χρησιμοποιηθεί το 100% της ισχύς των κινητήρων, η οποία κίνηση δεν ήταν επαρκώς αποδεκτή, με το όχημα να αδυνατεί να στρίψει και να κινηθεί ευθεία με ακρίβεια. Για τον λόγο αυτόν αφαιρέθηκαν οι μπαταρίες και χρησιμοποιήθηκαν εξωτερικά τροφοδοτικά. Με τα τροφοδοτικά αυτά η κίνηση, καθώς έγινε πιο ευέλικτη, τα καλώδια που εισήχθησαν παρουσίασαν πίεση στο όχημα, με αυτά να τείνουν να παραμείνουν ίσια, ενώ το όχημα να τείνει να στρίψει, μειώνοντας ξανά την ακρίβεια στροφών. Και ακόμη και χωρίς τις μπαταρίες, οι κινητήρες απεδείχθησαν αδύναμοι στην ικανότητά τους να κινήσουν το όχημα, καθώς έπρεπε να χρησιμοποιηθεί αρκετή ισχύς για την κίνηση. Αποτέλεσμα είναι το όχημα να κινείται γρήγορα, μειώνοντας την ικανότητά του να διαχειριστεί λεπτές κινήσεις, ή να μην κινείται καθόλου.

6.1.2 Συμπεράσματα για το λογισμικό

Σε γενικές γραμμές το λογισμικό αποδείχτηκε ικανό ως προς την ευελιξία, αναγνωσιμότητα και αρθρωτότητα του.

Χάρη στην χρήση της γλώσσας προγραμματισμού Python στο Raspberry Pi, η βελτιστοποίηση του κώδικα έγινε αρκετά πιο βατή. Βέβαια, ο κώδικας πρέπει να εκτελεί συνεχώς πολλαπλές παράλληλες επεξεργαστικά ακριβές διεργασίες ταυτόχρονα, οπότε φάνηκαν περιορισμοί στην υψηλή επίδοσή του, ειδικά όταν το όχημα κινείται με την αυτόνομη οδήγηση, και ταυτόχρονα υπάρχει παρατηρητής συνδεδεμένος στο όχημα. Με τον όγκο των διεργασιών που πρέπει να εκτελεστούν ταυτόχρονα, η απόδοση του Raspberry Pi ήταν επαρκής, αλλά δεν παύει να είναι μη ικανοποιητική από την οπτική γωνία ενός χρήστη που περιμένει άμεση ανταπόκριση από εντολές της κινητής εφαρμογής, όπως και αλλαγές στο περιβάλλον του οχήματος. Για την καταπολέμηση αυτού του φαινομένου, δεν χρησιμοποιήθηκε νευρωνικό δίκτυο για την αναγνώριση λωρίδας δρόμου, καθώς αυτό θα παρουσίαζε αρκετό υπολογιστικό φόρτο με αποτέλεσμα το σύστημα να ανταποκρινόταν ανεπαρκώς. Έτσι χρησιμοποιήθηκαν αποκλειστικά αλγόριθμοι επεξεργασία εικόνας με την χρήση της βιβλιοθήκης OpenCV.

Εξαιτίας αυτού, η λειτουργία εντοπισμού λωρίδας, καθώς αποδίδει ικανοποιητικά, πολλές φορές δίνει λανθασμένα αποτελέσματα. Για παράδειγμα, ο εντοπισμός των γραμμών της λωρίδας, όπως και η παρουσία εμποδίων λειτουργεί με τον εντοπισμό γραμμών με την χρήση του αλγορίθμου Hough. Κατ' επέκταση, αυτές

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

οι λειτουργίες επεξεργάζονται εικόνες στις οποίες έχει εντοπισθεί η απόχρωση των γραμμών της λωρίδας. Έτσι, αν στον δρόμο ή και στο περιβάλλον παρουσιαστούν σχήματα και εμπόδια με την ίδια απόχρωση όπως οι διαχωριστικές γραμμές της λωρίδας, τότε ο κώδικας θα εντοπίσει αυτά τα αντικείμενα ως γραμμές της λωρίδας. Ειδικά ο τωρινός τρόπος εντοπισμού εμποδίων στο περιβάλλον του οχήματος που βασίζεται στο πλήθος των γραμμών που εντοπίζονται στα πλαίσια που βρίσκονται στις άκρες της εικόνας, είναι αρκετά ευαίσθητος στον θόρυβο, δίνοντας συχνά λανθασμένους δείκτες παρουσίας εμποδίων.

Ο κώδικας που χρησιμοποιήθηκε για την κατασκευή της κινητής εφαρμογής και τις λειτουργίες του Arduino απέδωσαν ικανοποιητικά, καθώς για το μεγαλύτερο κομμάτι της λειτουργίας τους, περιμένουν εντολές από τον χρήστη ή από το Raspberry Pi, το οποίο είναι το πιο αργό εξάρτημα του συστήματος.

6.2 Βελτιώσεις συστήματος

6.2.1 Βελτιώσεις στο όχημα

Το πρώτο σημείο αναβάθμισης του συστήματος είναι πιθανός το υλικό που χρησιμοποιείται. Αρχικά, θα ήταν επιθυμητό να χρησιμοποιηθούν τέσσερις κινητήρες αντί για δύο δίνοντας έτσι υψηλότερη ισχύ και επαφή με το έδαφος, με αποτέλεσμα την ομαλότερη κίνηση. Επίσης, μπορεί να χρησιμοποιηθεί και διαφορετικός σκελετός για το όχημα που παρέχει περισσότερο χώρο για εξαρτήματα, μετακινώντας το κέντρο βάρους πιο κοντά στο κέντρο του οχήματος. Η αύξηση στην ισχύ μπορεί να επιτρέψει στο όχημα να φορτώσει και τις μπαταρίες πάνω στον σκελετό, αντί να εξαρτάται από εξωτερικά τροφοδοτικά των οποία τα καλώδια παρουσιάζουν εμπόδιο για το όχημα.

Μια επιπλέον αναβάθμιση αποτελεί η χρήση ταχύτερης έκδοσης του Raspberry Pi, ή οποιουδήποτε υπολογιστή αντίστοιχου μεγέθους και τιμής, που διαθέτουν επεξεργαστές νεότερης ή ταχύτερης τεχνολογίας. Με την κυκλοφορία του νέου Raspberry Pi 5 στην αγορά, είναι αναμενόμενη η μείωση της απόκρισης του συστήματος σε αλλαγές στο περιβάλλον όπως και η ταχύτερη επεξεργασία εικόνων, ή ακόμα και η ικανότητα χρήσης νευρωνικών δικτύων για την αύξηση της ακρίβειας του εντοπισμού της λωρίδας, εντοπισμού εμποδίων, όπως και την αναγνώριση παραπάνω λεπτομερειών για τα οχήματα που βρίσκονται στο άμεσο περιβάλλον, όπως η αλλαγή πορείας τους, επιβράδυνση και επιτάχυνσή τους.

Χρήση πιο εξελιγμένων αισθητήρων απόστασης. Στην παρούσα εργασία έχουν χρησιμοποιηθεί οι HC-SR04 αισθητήρες υπερήχων. Καθώς είναι αξιόπιστοι έως μια απόσταση και αρκετά προσιτοί, παρουσιάζουν αδυναμία στο εύρος της λειτουργίας τους. Έχουν την ικανότητα να εντοπίσουν εμπόδια μόνο σε μια μικρή γωνία που βρίσκεται ακριβώς μπροστά από τον αισθητήρα. Η χρήση ενός προηγμένου συστήματος LiDAR μπορεί να χρησιμοποιηθεί για να εντοπισθεί, όχι μόνο μια πολύ μικρή γωνία, αλλά μια μεγάλη περιοχή που βρίσκεται γύρω από τον αισθητήρα.

Αναβάθμιση της κάμερας. Η κάμερα που χρησιμοποιείται είναι ικανοποιητική για τις λειτουργίες της, αλλά κάποιες φορές παρουσιάζονται προβλήματα με τις αλλαγές στον φωτισμό του περιβάλλοντος, ή την παρουσία φωτεινών αντικειμένων σε σκοτεινό περιβάλλον. Μια κάμερα η οποία δεν παρουσιάζει αυτά τα προβλήματα μπορεί να αυξήσει την αξιοπιστία του συστήματος.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

6.2.2 Βελτιώσεις στο λογισμικό

Η γλώσσα που χρησιμοποιήθηκε για το Raspberry είναι η Python. Καθώς αυτή η γλώσσα είναι ευέλικτη και εύκολη στην χρήση της, δεν είναι η σωστή γλώσσα για συστήματα στα οποία η ταχεία ανταπόκριση σε αλλαγές στο περιβάλλον είναι κρίσιμη. Για αυτόν τον λόγο, η γλώσσα της C++ θα έδινε μεγάλη αύξηση στην ταχύτητα επεξεργασίας.

Χρήση νευρωνικών δικτύων για την αύξηση της ακρίβειας εντοπισμού λωρίδας, όπως και την καλύτερη αναγνώριση εμποδίων στο άμεσο περιβάλλον του οχήματος. Αυτό το κομμάτι προαπαιτεί την χρήση ταχύτερου επεξεργαστή, ή ταχύτερης γλώσσας προγραμματισμού, όπως C++.

Κατασκευή Web-Server πάνω στο Raspberry Pi, ο οποίος θα δίνει την ικανότητα σε χρήστες να ελέγξουν το όχημα χωρίς την χρήση της κινητής εφαρμογής, αλλά μέσω του προγράμματος περιήγησης που διαθέτουν οι κινητές συσκευές. Τώρα ένας χρήστης πρέπει να διαθέτει την κινητή εφαρμογή για τον έλεγχο του οχήματος. Χωρίς την εφαρμογή, ένας χρήστης μπορεί μόνο να είναι παθητικός παρατηρητής με πρόσβαση στις ροές βίντεο που παράγει το όχημα.

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

7. Παράρτημα

7.1. Εγκατάσταση απαραίτητων βιβλιοθηκών στο Raspberry Pi

Το λειτουργικό σύστημα που χρησιμοποιεί το Raspberry Pi σε αυτήν την εργασία είναι το Raspbian GNU/Linux 11 (Bullseye) 32-bit. Εφόσον το λειτουργικό είναι 32-bit, πρέπει να χρησιμοποιηθεί συγκεκριμένη έκδοση της γλώσσας Python. Στην παρούσα εργασία χρησιμοποιήθηκε η Python 3.9.2 [GCC 10.2.1 20210110], καθώς άλλες εκδόσεις παρουσίασαν προβλήματα συμβατότητας με τις απαραίτητες βιβλιοθήκες. Η συγκεκριμένη έκδοση Python, εγκαταστάθηκε από την ιστοσελίδα:

“<https://www.python.org/downloads/source/>”

Στην συνέχεια εγκαθιστώνται οι ακόλουθες βιβλιοθήκες:

- pip: 23.2.1
- OpenCV: opencv-contrib-python 4.7.0.72
- Numpy: numpy 1.19.5
- Flask: flask 1.1.2
- Flask-Cors: flask_cors 4.0.0
- Flask-Sockets: flask-sockets 0.2.1
- gevent: gevent 23.9.1
- gevent-websocket: geventwebsocket 0.10.1

Η εγκατάσταση γίνεται με την εκτέλεση της εντολής :

“python -m pip install X==Y”, όπου X είναι το όνομα της βιβλιοθήκης, και Y είναι η έκδοση της βιβλιοθήκης. Η μόνη διαφορά παρατηρείται στην αναβάθμιση του pip, καθώς γίνεται με την εξής εντολή:
“pip install --upgrade pip==23.2.1”

Γίνεται επίσης εγκατάσταση του Arduino IDE 32-bit για Linux από τον παρακάτω σύνδεσμο:

<https://www.arduino.cc/en/software>

7.2 Δημιουργία του τοπικού δικτύου - hotspot - του Raspberry Pi

Αρχικά πρέπει να εκτελεστεί η εντολή *“sudo apt-get install hostapd dnsmasq”*, *“sudo apt-get install systemctl”*, και δημιουργηθούν τρία αρχεία για να λειτουργήσει το hotspot:

1. Στον φάκελο *“/etc/hostapd/”*, δημιουργούμε αρχείο *“hostapd.conf”* με το εξής περιεχόμενο:
interface=wlan0
driver=nl80211
ssid=“Το όνομα του δικτύου του Raspberry Pi”
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

```
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase"Ο κωδικός του δικτύου"
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

2. Στον φάκελο `/etc/` δημιουργούνται δύο αρχεία:
"dnsmasq.conf" με περιεχόμενο:
interface=wlan0
dhcp-range="Αρχική διεύθυνση IP"-"Τελική διεύθυνση IP",24h
dhcp-option=option:classless-static-route,"IP Διεύθυνση δικτύου"/"Μάσκα δικτύου","IP Διεύθυνση του Raspberry Pi στο δίκτυο"

Στο κάτω μέρος του αρχείου `dhcpcd.conf` προσθέτουμε το παρακάτω κομμάτι:

```
interface wlan0
nohook wpa_supplicant
inform "IP Διεύθυνση του Raspberry Pi στο δίκτυο"/"Μάσκα δικτύου"
static route add - net "IP Διεύθυνση δικτύου"/"Μάσκα δικτύου" gw "IP Διεύθυνση του Raspberry Pi στο δίκτυο" metric 10
noipnr6rs
```

3. Ύστερα, για την ενεργοποίηση ή απενεργοποίηση του δικτύου, μπορούμε να γράψουμε μικρά script για τον έλεγχο αυτού:

```
"hotspot_control.sh":
#!/bin/bash
hotspot_status = $(sudo systemctl is-active hostapd)
```

```
if [[ "hotspot_status" == "active" ]]; then
./stop_hotspot.sh
else
./start_hotspot.sh
fi
```

```
"start_hotspot.sh":
#!/bin/bash
sudo systemctl start hostapd
sudo systemctl start dnsmasq
```

```
"stop_hotspot.sh":
#!/bin/bash
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

7.3 Εγκατάσταση του προγράμματος Zerotier

Με το πρόγραμμα αυτό, μπορεί ένας παρατηρητής από οπουδήποτε στον πλανήτη, να παρακολουθήσει τις ροές βίντεο που παράγει το Raspberry Pi κατά την διάρκειά του. Για αυτό χρειάζεται να εγκαταστηθεί το πρόγραμμα Zerotier, και ύστερα το Raspberry Pi και η συσκευή του παρατηρητή να βρίσκονται στο ίδιο δίκτυο Zerotier.

Για την εγκατάσταση ακολουθούμε τα εξής βήματα:

1. Αρχικά εκτελούμε τις δύο εντολές για να πιστοποιήσουμε πως το Raspberry Pi είναι αναβαθμισμένο:
"sudo apt update", "sudo apt upgrade"
2. Ύστερα ακολουθούμε με τις δύο εντολές:
"curl
https://raw.githubusercontent.com/zerotier/ZeroTierOne/master/doc/contact%40zerotier.com.gpg | gpg --dearmor | sudo tee
/usr/share/keyrings/zerotierone-archive-keyring.gpg >/dev/null"
και "RELEASE=\$(lsb_release -cs)"
3. Ύστερα ανανεώνουμε τον σύνδεσμο που διαθέτει το λειτουργικό σύστημα του Raspberry Pi για το package του Zerotier:
"echo "deb [signed-by=/usr/share/keyrings/zerotierone-archive-keyring.gpg] http://download.zerotier.com/debian/\$RELEASE \$RELEASE main" | sudo tee /etc/apt/sources.list.d/zerotier.list"
4. Εκτελούμε ξανά "sudo apt update".
5. Τέλος, "sudo apt install -y zerotier-one"

Για να προσθέσουμε το Raspberry Pi στο δίκτυο που διαθέτουμε, πρέπει πρώτα να φτιάξουμε έναν λογαριασμό στην ιστοσελίδα του Zerotier, "https://www.zerotier.com/" και να κατασκευάσουμε ένα δίκτυο ακολουθώντας τις οδηγίες της ιστοσελίδας.

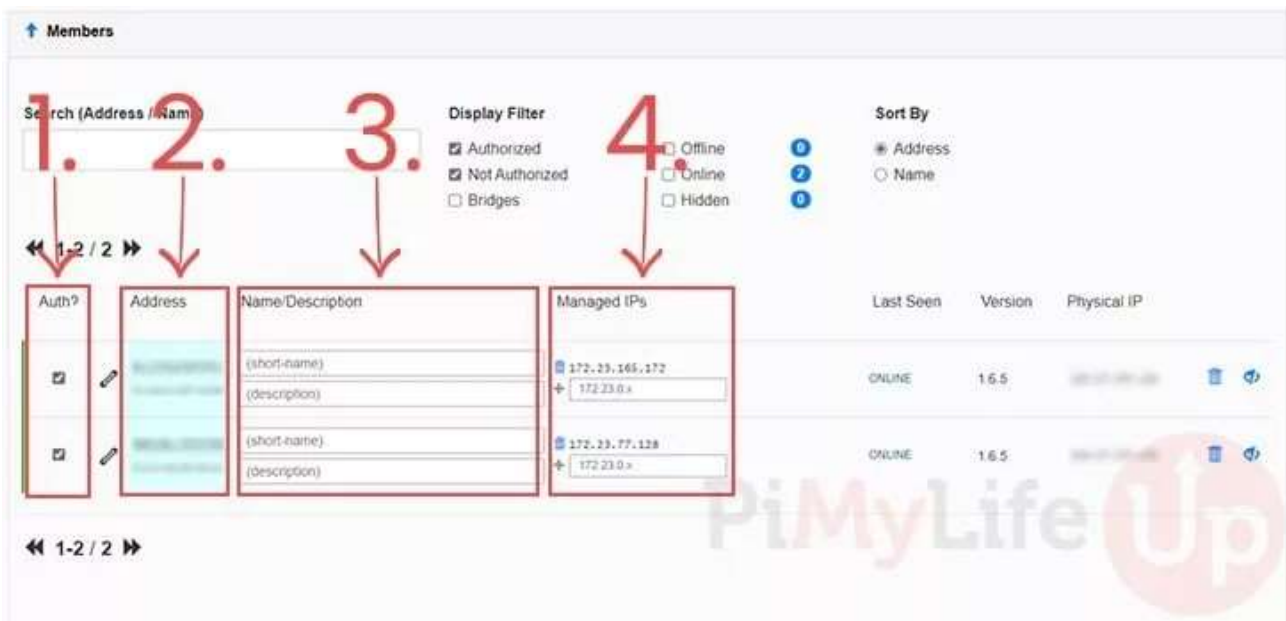
Αφού κατασκευαστεί το δίκτυό μας, πληκτρολογούμε στο τερματικό του Raspberry Pi την εξής εντολή:

```
"sudo zerotier-cli join [Network ID]". Αν το Raspberry Pi προστεθεί στο δίκτυό μας χωρίς λάθη, τότε θα τυπωθεί από κάτω "200 join OK".
```

Τέλος, ανάλογα με το πως έχουμε κατασκευάσει το δίκτυό μας στο Zerotier, ίσως χρειαστεί να πιστοποιήσουμε την συσκευή που μόλις μπήκε στο δίκτυό μας. Για να βεβαιωθούμε, πάμε στον σύνδεσμο

```
"https://my.zerotier.com/network/[Network ID]" και πιστοποιούμε την συσκευή μας, και δίνουμε μια διεύθυνση IP στο Raspberry Pi.
```


Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος



Εικόνα 7.3.1 Κεντρικός πίνακας ελέγχου του δικτύου Zerotier

Πηγή: <https://pimylifeup.com/raspberry-pi-zerotier/>

Στην Εικόνα 7.3.1, παρατηρούνται κάποιες στήλες. Η πρώτη στήλη ρυθμίζει τις πιστοποιήσεις των συσκευών που εισέρχονται στο δίκτυό μας. Η δεύτερη στήλη περιέχει μοναδικές Zerotier διευθύνσεις συσκευή που είναι μοναδικές για κάθε συσκευή στο δίκτυο. Στην στήλη 3 ρυθμίζουμε το όνομα και περιγραφή της συσκευής ενδιαφέροντος. Στην τέταρτη στήλη μπορούμε να ρυθμίσουμε την διεύθυνση IP της συσκευής ενδιαφέροντος. Ανάλογα με τις ρυθμίσεις του δικτύου μας, αυτή η στήλη μπορεί να ρυθμίζεται αυτόματα, ή πρέπει να την ρυθμίσουμε χειροκίνητα.

Με αυτά τα βήματα, μπορούμε να συνδεθούμε στο Raspberry Pi από απόσταση, χωρίς κάποιον άμεσο τρόπο σύνδεσης, με μόνο τον μόνο περιορισμό την παρουσία σύνδεσης στο διαδίκτυο από το Raspberry Pi και την συσκευή που χρησιμοποιεί ο παρατηρητής για την σύνδεσή του στο Raspberry Pi.

7.4 Εκτέλεση κώδικα

Για την εκτέλεση του κώδικα, χρησιμοποιούνται τα εξής αρχεία, τα οποία μπορούν να εκτελεστούν μέσω SSH σύνδεσης με το Raspberry Pi με την εντολή:

(ssh “Όνομα του Raspberry Pi@”IP Διεύθυνση του Raspberry Pi”), ή κανονικά με την χρήση ποντικιού ή εντολής μέσα από το τερματικό του Raspberry Pi:

```
“start_pi_sh”:  
#!/bin/bash
```

```
python -u “Τοποθεσία του main.py αρχείου του κωδικά”  
exit 0
```

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

“start_arduino.sh”:

```
#!/bin/bash
```

```
# Path to the Arduino CLI executable
```

```
ARDUINO_CLI="Τοποθεσία του Arduino executable"
```

```
# Path to the Arduino sketch folder
```

```
SKETCH_PATH="Τοποθεσία του main.ino αρχείου του κωδικα"
```

```
# Upload the sketch to the Arduino board
```

```
$ARDUINO_CLI --upload $SKETCH_PATH --port /dev/ttyACM0
```

```
exit 0
```

“upload_code.sh”:

```
#!/bin/bash
```

```
# Path to the Arduino CLI executable
```

```
ARDUINO_CLI="/home/orestts/Desktop/arduino-1.8.19/arduino"
```

```
# Path to the Arduino sketch folder
```

```
SKETCH_PATH="/home/orestts/Desktop/Thesis/Arduino/Code/main/main.ino"
```

```
# Upload the sketch to the Arduino board
```

```
$ARDUINO_CLI --upload $SKETCH_PATH --port /dev/ttyACM0
```

```
python -u /home/orestts/Desktop/Thesis/Pi/main.py
```

```
exit 0
```

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

8. Πηγές και βιβλιογραφία

[1]: "The future of mobility" neom.com

<https://www.neom.com/en-us/our-business/sectors/mobility> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[2]: "Waymo story" waymo.com

<https://waymo.com/about/#story> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[3]: A. J. Hawkins "Waymo is first to put fully self-driving cars on US roads without a safety driver" theverge.com

<https://www.theverge.com/2017/11/7/16615290/waymo-self-driving-safety-driver-chandler-autonomous> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[4]: J. White "Waymo opens driverless robo-taxi service to the public in Phoenix" reuters.com

<https://www.reuters.com/article/us-waymo-autonomous-phoenix-idUSKBN26T2Y3> (τελευταία επίσκεψη 01 Σεπτεμβρίου 2023)

[5]: P. Dave "Waymo seeks permit to sell self-driving car rides in San Francisco" reuters.com

<https://www.reuters.com/technology/waymo-seeks-permit-sell-self-driving-car-rides-san-francisco-2022-12-13/> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[6]: J. Elias "Waymo says it plans to launch its self-driving service in Los Angeles" cnbc.com

<https://www.cnbc.com/2022/10/19/waymo-says-it-plans-to-launch-a-ride-hailing-service-in-los-angeles.html> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[7]: K. Korosec "Waymo doubles robotaxi service area in Phoenix in bid to grow driverless trips tenfold" techcrunch.com

<https://techcrunch.com/2023/05/04/waymo-doubles-robotaxi-service-area-in-phoenix-in-bid-to-grow-driverless-trips-tenfold/> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[8]: R. Carter "How Many Teslas Have Been Sold? (2023 Update)" moneytransfers.com

<https://moneytransfers.com/news/2022/05/30/how-many-teslas-have-been-sold> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[9]: J. Mulach "Tesla admits its semi-autonomous driving tech is not the world's most advanced" drive.com.au

<https://www.drive.com.au/news/tesla-full-self-driving-level-2-autonomous/> (τελευταία επίσκεψη 06 Σεπτεμβρίου 2023)

[10]: J. Shuttleworth "SAE Standards News: J3016 automated-driving graphic update" sae.org

<https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic> (τελευταία επίσκεψη 01 Σεπτεμβρίου 2023)

[11]: "Science: Radio Auto", 1925 time.com/

<https://content.time.com/time/subscriber/article/0,33009,720720,00.html> (τελευταία επίσκεψη 07 Σεπτεμβρίου 2023)

[12]: C. Engelking "The 'Driverless' Car Era Began More Than 90 Years Ago" discovermagazine.com

<https://www.discovermagazine.com/technology/the-driverless-car-era-began-more-than-90-years-ago> (τελευταία επίσκεψη 07 Σεπτεμβρίου 2023)

[13]: The Milwaukee Sentinel "PHANTOM AUTO WILL TOUR CITY" 8 Δεκεμβρίου 1926

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

<https://web.archive.org/web/20181010202440/https://news.google.com/newspapers?id=unBQAAAAIIBAJ&sjid=QQ8EAAAAIIBAJ&pg=7304,3766749> (τελευταία επίσκεψη 07 Σεπτεμβρίου 2023)

[14]: The Free Lance-Star “*Phantom Auto to Be Operated Here*” 17 Ιουνίου 1932
<https://news.google.com/newspapers?id=PthNAAAAIIBAJ&sjid=yYoDAAAAIIBAJ&pg=6442,3879017>
(τελευταία επίσκεψη 07 Σεπτεμβρίου 2023)

[15]: “*Driving Back through the History of Autonomous Travel*” cseengineermag.com
<https://cseengineermag.com/driving-back-through-the-history-of-autonomous-travel/> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[16]: N. Hicks “*Nebraska tested driverless car technology 60 years ago*” journalstar.com
https://journalstar.com/news/local/govt-and-politics/nebraska-tested-driverless-car-technology-years-ago/article_a702fab9-cac3-5a6e-a95c-9b597fdab078.html (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[17]: J. Delcker “*The man who invented the self-driving car (in 1986)*” politico.eu
<https://www.politico.eu/article/delf-driving-car-born-1986-ernst-dickmanns-mercedes/> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[18]: R. D. Leighty “*DARPA ALV SUMMARY*”, U. S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060-5546, Report Documentation Page, 1η Μαρτίου 1986
Defense Technical Information Center - discover.dtic.mil
<https://apps.dtic.mil/sti/pdfs/ADA167472.pdf> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[19]: D. A. Pomerleau, “ALVINN, an autonomous land vehicle in a neural network”. Carnegie Mellon University, 30-Jun-2018, doi: 10.1184/R1/6603146.v1.
ALVINN, an autonomous land vehicle in a neural network (cmu.edu)

[20]: A. Oagana “*A Short History of Mercedes-Benz Autonomous Driving Technology*” autoevolution.com
<https://www.autoevolution.com/news/a-short-history-of-mercedes-benz-autonomous-driving-technology-68148.html> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[21]: D. Rostcheck “*The Self-Driving Car — from 1994*” medium.com
<https://medium.com/@davidrostcheck/the-self-driving-car-from-1994-fb1ec617bd5a> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[22]: “*The PROMETHEUS project launched in 1986: Pioneering autonomous driving*” automotiveworld.com
<https://www.automotiveworld.com/news-releases/prometheus-project-launched-1986-pioneering-autonomous-driving/> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[23]: C. Lee “*The professor who built a self-driving car decades ago*” taipeitimes.com
<https://www.taipeitimes.com/News/biz/archives/2021/04/04/2003755032> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[24]: P. Marks “*Driverless cars ready to hit our roads*” newscientist.com
<https://www.newscientist.com/article/mg21328585-300-driverless-cars-ready-to-hit-our-roads/>
(τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[25]: Raglan Tribe, Ηνωμένο Βασίλειο, “*Robot Car | Raglan Tribe*” (23 Φεβρουαρίου 2013)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

(τελευταία επίσκεψη 08 Σεπτεμβρίου 2023) [Online Video]. Διαθέσιμο:
<https://www.youtube.com/watch?v=AILZhcnpXYI>

[26]: Πανεπιστήμιο της Πάρμα, Τμήμα Μηχανικών και Πληροφορικής “*The ARGO Project*”
<https://www.argo.ce.unipr.it/argo/english/index.html> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[27]: Πανεπιστήμιο Carnegie Mellon, Ινστιτούτο Ρομποτικής “*NO HANDS ACROSS AMERICA*”
https://www.cs.cmu.edu/afs/cs/usr/tjochem/www/nhaa/nhaa_home_page.html (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[28]: Πανεπιστήμιο Carnegie Mellon, Ινστιτούτο Ρομποτικής “*Navlab: The Carnegie Mellon University Navigation Laboratory*”
<https://www.cs.cmu.edu/afs/cs/project/alv/www/index.html> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[29]: en.wikipedia.org “*ParkShuttle*”
<https://en.wikipedia.org/wiki/ParkShuttle> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[30]: en.wikipedia.org “*DARPA Grand Challenge (2004)*”
[https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_\(2004\)](https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_(2004)) (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[31]: en.wikipedia.org “*DARPA Grand Challenge (2005)*”
[https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_\(2005\)](https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_(2005)) (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[32]: en.wikipedia.org “*DARPA Grand Challenge (2007)*”
[https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_\(2007\)](https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_(2007)) (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[33]: M. Maisto “*Induct Now Selling Navia, First Self-Driving Commercial Vehicle*” eweek.com
<https://archive.ph/20140107184549/http://www.eweek.com/innovation/induct-now-selling-navia-first-self-driving-commercial-vehicle.html/#selection-388.0-388.1> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[34]: V. Vijayenthiran “*Tesla Version 7.0 Software Update Activates First Autopilot Features*” motorauthority.com
https://www.motorauthority.com/news/1100476_tesla-version-7-0-software-update-activates-first-autopilot-features (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[35]: N. Grant “*Waymo, Chasing Cruise, Plans Fully Driverless Rides in San Francisco*” bloomberg.com
<https://www.bloomberg.com/news/articles/2022-03-30/google-s-waymo-to-offer-public-fully-driverless-rides-in-san-francisco> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[36]: B. Wessling “*Mercedes rolls out Level 3 autonomous driving tech in Germany*” therobotreport.com
<https://www.therobotreport.com/mercedes-rolls-out-level-3-autonomous-driving-tech-in-germany/> (τελευταία επίσκεψη 08 Σεπτεμβρίου 2023)

[37]: A. Asagami “*Level-4 Self-Driving Cars to Hit Japanese Streets for the First Time*” japan-forward.com
<https://japan-forward.com/level-4-self-driving-cars-to-hit-japanese-streets-for-the-first-time/> (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)

Κατασκευή αυτοκινούμενου οχήματος με Raspberry-Pi, Arduino, με εφαρμογή κινητής συσκευής για τον έλεγχο λειτουργιών του συστήματος

[38]: K. Parrish “*Tiny USB-Sized PC Offers 1080p HDMI Output*” tomshardware.com
<https://www.tomshardware.com/news/Raspberry-Pi-David-Braben-Ubuntu-9-OLPC-Railroad-Tycoon,12709.html> (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)

[39]: “*The History of Raspberry Pi*” deviceplus.com
<https://www.deviceplus.com/raspberry-pi/the-history-of-raspberry-pi/#OriginStory> (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)

[40]: Raspberry Pi Trading Ltd. “*Raspberry Pi 4 Computer Model B*” (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023) [Online]. Διαθέσιμο: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf>

[41]: H. Barragán “*The Untold History of Arduino*”
<https://arduinohistory.github.io/> (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)

[42]: J. M. Hughes “*Arduino: A Technical Reference*” oreilly.com
<https://www.oreilly.com/library/view/arduino-a-technical/9781491934319/ch01.html> (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)

[43]: arduino.cc “*Arduino Hardware*”
<https://www.arduino.cc/en/hardware> (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)

[44]: arduino.cc “*Arduino Mega 2560 Rev3*”
<https://store.arduino.cc/products/arduino-mega-2560-rev3> (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)

[45]: OmniVision Technologies “*OV5647 Datasheet*” (τελευταία επίσκεψη 13 Σεπτεμβρίου 2023)
[Online]. Διαθέσιμο:
https://www.arducam.com/downloads/modules/RaspberryPi_camera/OV5647DS.pdf

[46]: botland.store “*Ultrasonic distance sensor HC-SR04 2-200cm - justPi*”
<https://botland.store/ultrasonic-distance-sensors/1420-ultrasonic-distance-sensor-hc-sr04-2-200cm-justpi-5903351241366.html> (τελευταία επίσκεψη 14 Σεπτεμβρίου 2023)

[47]: botland.store “*Chassis Round 2WD - Robot Chassis with DC Motor Drive*”
<https://botland.store/chassis-for-robots/7243-chassis-round-2wd-robot-chassis-with-dc-motor-5904422310028.html> (τελευταία επίσκεψη 14 Σεπτεμβρίου 2023)

[48]: openplatform.cc “*Stepper Motor Controller Module, ST1112*”
<http://www.openplatform.cc/index.php/home/index/details/apiid/28> (τελευταία επίσκεψη 14 Σεπτεμβρίου 2023)