



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών

«ΠΛΗΡΟΦΟΡΙΚΗ»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη οχήματος τύπου rover με μηχανική όραση και χειρισμό μέσω Web API Development of a rover-type vehicle with artificial vision and web API management
Όνοματεπώνυμο Φοιτητή	ΑΝΤΩΝΙΟΣ ΣΠΑΝΟΣ
Πατρώνυμο	ΝΙΚΟΛΑΟΣ
Αριθμός Μητρώου	ΜΠΠΛ/17049
Επιβλέπων	ΘΕΜΙΣΤΟΚΛΗΣ ΠΑΝΑΓΙΩΤΟΠΟΥΛΟΣ, ΚΑΘΗΓΗΤΗΣ

Ημερομηνία Παράδοσης **Ιούλιος 2022**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Θ. Παναγιωτόπουλος
Καθηγητής

(υπογραφή)

Δ. Σωτηρόπουλος
Επίκουρος Καθηγητής

(υπογραφή)

Ι. Τασούλας
Επίκουρος Καθηγητής

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	4
ΕΛΛΗΝΙΚΑ	4
ENGLISH	4
ΕΙΣΑΓΩΓΗ	5
ΚΕΦΑΛΑΙΟ 1°	6
ΡΟΜΠΟΤ ΚΑΙ ΡΟΜΠΟΤΙΚΗ	6
ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ	7
ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ	7
ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ	8
ΚΕΦΑΛΑΙΟ 2°	10
ΚΑΤΑΣΚΕΥΗ	10
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ROVER	15
ΚΕΦΑΛΑΙΟ 3°	17
WEB USER INTERFACE	17
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ UI	20
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ OBJECT TRACKING	22
ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ	23
ΚΕΦΑΛΑΙΟ 4°	24
ΑΥΤΟΝΟΜΗ ΛΕΙΤΟΥΡΓΙΑ	24
ΧΕΙΡΟΚΙΝΗΤΗ ΛΕΙΤΟΥΡΓΙΑ	27
ΚΕΦΑΛΑΙΟ 5°	30
ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΠΑΡΟΥΣΙΑΣΤΗΚΑΝ	30
ΣΥΜΠΕΡΑΣΜΑΤΑ	30
ΒΙΒΛΙΟΓΡΑΦΙΑ	32

ΠΕΡΙΛΗΨΗ

ΕΛΛΗΝΙΚΑ

Τις προηγούμενες δεκαετίες, οι έρευνες για τα μη επανδρωμένα οχήματα κάθε τύπου (επίγεια, εναέρια, υποβρύχια) επικεντρώνονταν στις στρατιωτικές εφαρμογές και λιγότερο σε εμπορικές ή εκπαιδευτικές. Η αυξανόμενη τάση εμπορευματοποίησης των αυτόνομων οχημάτων από μεγάλες εταιρίες ενέτεινε την έρευνα στον ακαδημαϊκό χώρο. Η ανάπτυξη της τεχνολογίας και η ταυτόχρονη μείωση του κόστους διάφορων υπολογιστικών συστημάτων βοήθησαν στην διάδοσή τους. Η ανάπτυξη αισθητήρων ακριβείας σε πολλαπλές εφαρμογές και η ποικιλία πλατφορμών ανάπτυξης ρομποτικών εφαρμογών ανοιχτού κώδικα, συνέβαλαν δραστικά στην ανάπτυξη ενός ιδιαίτερα φιλικού περιβάλλοντος για τον ερευνητή και τον χρήστη. Στην παρούσα εργασία, παρουσιάζεται το αποτέλεσμα όλων αυτών των εξελίξεων, ένα μη επανδρωμένο όχημα γενικού σκοπού, ιδιοκατασκευή, στηριγμένο στον υπολογιστή πλακέτας Raspberry Pi, με πολλαπλούς αισθητήρες, μηχανική όραση και απομακρυσμένο έλεγχο (μέσω φυλλομετρητή υλοποιημένο με REST API).

ENGLISH

During the previous decades, research upon unmanned autonomous vehicles of every type (terrestrial, flying, submerged) was focused mainly on military applications and on a lesser degree on commercial or educational ones. However, the progressive commercialization of this kind of vehicles intensified research by major firms in the academic area. Technological advancements together with cost reduction of numerous computational resources helped to spread their appeal. The development in sensor accuracy and the diversity on open-source development platforms drastically contributed to a friendly environment for both the researcher and the user. In this dissertation, we present the culmination of this progress a general purpose unmanned autonomous vehicle based on the single board computer Raspberry Pi, equipped with numerous sensors, computer vision and remote control (browser based using RESTful API).

ΕΙΣΑΓΩΓΗ

Σε έναν αυξανόμενο τεχνοκρατικό κόσμο δημιουργήθηκε η ανάγκη ανάπτυξης νέων πεδίων στις επιστήμες που θα ανταποκρίνονται στις τρέχουσες απαιτήσεις. Μία απάντηση σε αυτό το ζητούμενο δίνει η ρομποτική, ένας σύγχρονος τεχνολογικός κλάδος της αυτοματοποίησης που έχει ως αντικείμενο τη μελέτη, το σχεδιασμό και τη λειτουργία των ρομπότ, όπως και την έρευνα για την περεταίρω ανάπτυξή τους. Οι εφαρμογές της εντοπίζονται σε διάφορους παραγωγικούς τομείς εστιασμένοι στη βιομηχανία, στην ιατρική, στην αεροναυπηγική και στην αεροδιαστημική. Η ραγδαία μείωση του κόστους απόκτησης του υλικού (hardware) και ο συνδυασμός συνεχώς αναπτυσσόμενων εργαλείων λογισμικού (software) ικανών για προσομοίωση εικονικών κόσμων και αποδέυση του υλικού από τον σχεδιασμό, καθιστούν τη ρομποτική πολύ πιο προσιτή.

Η αυξανόμενη τάση εμπορευματοποίησης των αυτόνομων οχημάτων από μεγάλες εταιρίες εντείνει την έρευνα στον ακαδημαϊκό χώρο. Η ανάπτυξη της τεχνολογίας και η ταυτόχρονη μείωση του κόστους διάφορων υπολογιστικών συστημάτων βοήθησαν στην διάδοση τους και στην ανάπτυξη αισθητήρων ακριβείας με πολλαπλές εφαρμογές. Επιπλέον, η ποικιλία πλατφορμών ανάπτυξης ρομποτικών εφαρμογών ανοιχτού κώδικα συνέβαλαν δραστικά στην ανάπτυξη ενός ιδιαίτερα φιλικού περιβάλλοντος για τον ερευνητή και τον χρήστη. Στην παρούσα εργασία, παρουσιάζεται το αποτέλεσμα όλων αυτών των εξελίξεων, ένα μη επανδρωμένο όχημα γενικού σκοπού, ιδιοκατασκευή, στηριγμένο στον υπολογιστή πλακέτας Raspberry PI, με πολλαπλούς αισθητήρες μηχανικής όρασης και απομακρυσμένο έλεγχο (μέσω φυλλομετρητή υλοποιημένο με REST API).

Σκοπός του αυτόνομου οχήματος που παρουσιάζεται στην εργασία αυτή είναι η εξοικείωση με τις προαναφερθείσες τεχνολογίες κάνοντας χρήση μιας πλατφόρμας ανοιχτού λογισμικού και ευρέως διαθέσιμων υλικών. Στόχος η υλοποίηση συστήματος αυτόνομου εντοπισμού ενός αντικειμένου στις 2 διαστάσεις (ενδεικτικά μιας κίτρινης μπάλας) αλλά και δυνατότητες τηλεχειρισμού μέσω φυλλομετρητή κάνοντας χρήση διεπαφής τύπου REST (REST API) σε μια εφαρμογή server - client μέσω του διαύλου διασύνδεσης Wi-Fi.

Στο κεφάλαιο 1 παρουσιάζονται βασικές έννοιες που θα χρησιμοποιηθούν για την υλοποίηση του οχήματος. Στην συνέχεια, στο κεφάλαιο 2 παρουσιάζεται η κατασκευή του οχήματος σαν φυσικό όχημα ενώ στο κεφάλαιο 3 παρουσιάζεται η σχεδίαση του λογισμικού και το υλικό που χρησιμοποιήθηκε. Στο κεφάλαιο 4 περιγράφεται το λογισμικό που δημιουργήθηκε και αναλύονται τα κύρια κομμάτια του κώδικα. Στο κεφάλαιο 5 παρουσιάζονται στιγμιότυπα από την λειτουργία του ρομπότ. Τέλος στο κεφάλαιο 6 παρουσιάζονται τα συμπεράσματα και τα προβλήματα που προέκυψαν κατά την ανάπτυξη της κατασκευής.

ΚΕΦΑΛΑΙΟ 1°

Για την κατασκευή ενός αυτόνομου οχήματος πρέπει να μελετηθούν πολλαπλά επιστημονικά πεδία. Η αυτόνομη πλοήγηση στον χώρο προϋποθέτει την κατανόηση του χώρου και την δυνατότητα μετάφρασης των οπτικών δεδομένων σε μαθηματικά μοντέλα υπολογισμού των διαθέσιμων κινήσεων, επιλογής της βέλτιστης και υλοποίησης της. Στην συνέχεια θα μελετηθούν κάποιες από τις εμπλεκόμενες έννοιες.

ΡΟΜΠΟΤ ΚΑΙ ΡΟΜΠΟΤΙΚΗ

Ως ρομπότ θεωρούμε όποια μηχανή δύναται να αλληλεπιδρά με το περιβάλλον της. Το ρομπότ είναι μια κατασκευή επαναπρογραμματιζόμενη και πολυλειτουργική, προικισμένη με αυτονομία, ικανή να εκπληρώσει τις προγραμματισμένες ενέργειες (μετακίνηση υλικών, εξαρτημάτων, εργαλείων και εξειδικευμένων διατάξεων) και χρησιμοποιείται για την εκτέλεση επικύνδυνων ή δύσκολων (για ανθρώπους) εργασιών ή για να αυτοματοποιήσει μονότονες επαναληπτικές εργασίες που πρέπει να γίνουν με μεγάλη ακρίβεια, σε ένα μεταβαλλόμενο περιβάλλον. Το Ρομπότ καθοδηγείται από ένα ηλεκτρονικό κύκλωμα και από κάποιο πρόγραμμα υπολογιστή, ενώ αποτελείται από δομικά μέρη (δοκοί, γρανάζια, ρόδες, άξονες...), ηλεκτρονικά (μονάδα επεξεργαστή, ελεγκτές, ολοκληρωμένα κυκλώματα, πυκνωτές, αντιστάσεις,...), αισθητήρες (χρώματος, φωτός, κίνησης, απόστασης, ...), μηχανικά μέρη, ηλεκτρικά μέρη, κινητήρες, μονάδες εξόδου (ηχεία, οθόνη, ενδεικτικές λυχνίες, ...) και από μία πηγή ενέργειας.

Τα ρομπότ θα μπορούσαν να διαχωριστούν σε τρεις κατηγορίες, ή αλλιώς “γενιές”. Τα ρομπότ πρώτης γενιάς χαρακτηρίζονται με περιορισμένη ευελιξία, διευθύνονται από τον άνθρωπο-χειριστή με απλά εργαλεία που επιτρέπουν, για παράδειγμα, την μετακίνηση επικύνδυνων αντικειμένων (όπως ραδιενεργών υλικών). Τα ρομπότ δεύτερης γενιάς είναι εφοδιασμένα με σταθερό πρόγραμμα δράσης και λαμβάνουν εντολές από κάποιο σύστημα ελέγχου. Τέλος, στην τρίτη γενιά ανήκουν τα ρομπότ που είναι εφοδιασμένα με δυνατότητες σύλληψης και επεξεργασίας αισθητήριων “πληροφοριών” από το περιβάλλον στόχο την εκτέλεση εργασιών. Στη παρούσα διατριβή, το ρομπότ κατατάσσεται στην τρίτη γενιά, καθώς το ρομπότ είναι δυνατόν να πλοηγηθεί με μεθόδους μηχανικής όρασης.

Κατά συνέπεια ένα ρομπότ είναι μια μηχανή που:

1. Αντιλαμβάνεται το περιβάλλον του.
2. Εκτελεί διάφορες αποστολές, όπως κίνηση ή χειρισμό.
3. Επαναπρογραμματίζεται, μπορεί δηλαδή να εκτελεί διαφορετικές αποστολές.
4. Είναι αυτόνομο ή/και αλληλεπιδρά με τους ανθρώπους.

Για να ανταπεξέλθει το ρομπότ στις παραπάνω απαιτήσεις εφοδιάζεται με συσκευές δράσης (actuators), όπως servo και κινητήρες σχηματίζοντας πόδια, αρθρώσεις και λαβές, αλλά και με αισθητήρες, όπως κάμερες, γυροσκόπια και επιταχυνσιόμετρα. Όπως είναι αναμμενόμενο, η ρομποτική συνδυάζει στοιχεία από πολλές καινοτομίες, όπως τη μηχανική

όραση, τη μηχανική μάθηση, την επεξεργασία εικόνας, την αναγνώριση προτύπων και την τεχνητή νοημοσύνη. Ακολούθως παρουσιάζονται οι κυρίαρχες έννοιες και επιστημονικά πεδία που θα μας απασχολήσουν ιδιαίτερα στην παρούσα εργασία.

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

Η αναγνώριση προτύπων (pattern recognition) είναι ένα επιστημονικό πεδίο της μηχανικής μάθησης που έχει ως στόχο την ανάπτυξη αλγορίθμων για την ταξινόμηση αντικειμένων σε κατηγορίες (κλάσεις). Ανάλογα με το είδος της εφαρμογής, τα προς ταξινόμηση αντικείμενα μπορεί να είναι εικόνες, βίντεο, κυματομορφές σημάτων ή οποιοδήποτε άλλο είδος μετρήσεων που τα διακρίνει κάποιο χαρακτηριστικό και χρειάζεται να ταξινομηθεί.

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

Επεξεργασία εικόνας ορίζουμε την αλγοριθμική επεξεργασία, ανάλυση και χειρισμό ψηφιακών δεδομένων εικόνας ή βίντεο. Κατά την διάρκεια της επεξεργασίας εικόνας, τόσο η είσοδος όσο και η έξοδος είναι δεδομένα εικόνας ή βίντεο (έγχρωμα, ασπρόμαυρα ή σε αποχρώσεις του γκριζου). Η επεξεργασία εικόνας ενέχει την έννοια της επεξεργασίας σήματος, καθώς η ψηφιακή εικόνα μπορεί να θεωρηθεί δισδιάστατο χωρικό σήμα και το βίντεο τρισδιάστατο χωροχρονικό σήμα. Η διαδικασία σχεδίασης και υλοποίησης των αλγορίθμων επεξεργασίας εικόνας απαιτεί πολύ καλές γνώσεις μαθηματικών.

Η επεξεργασία εικόνας μπορεί να διαχωριστεί στους παρακάτω τύπους:

- Γεωμετρικές μετατροπές: Αλλαγή στο μέγεθος ολόκληρης ή τμήματος της εικόνας, περιστροφή, παραμόρφωση, αλλαγή προοπτικής, αλλαγή ανάλυσης, κλπ.
- Χρωματικές μετατροπές και διορθώσεις: Αλλαγή χρωματικών τόνων μιας εικόνας, ρύθμιση φωτεινότητας, αντίθεσης, αλλαγή του χρωματικού χώρου (π.χ. από RGB σε CMYK).
- Συμπίεση και μετατροπή της μορφής αποθήκευσης μιας εικόνας στον υπολογιστή, π.χ. από μορφή .jpg σε μορφή .tif.
- Εφαρμογή φίλτρων με στόχο τη βελτίωση της ποιότητας της εικόνας ή τονισμό των γνωρισμάτων της (αφαίρεση αμυχών, εξάλειψη φαινομένου “κόκκινων ματιών”, ανίχνευση ακμών και τονισμός περιγραμμάτων, μείωση θορύβου, κλπ).
- Ανάμιξη δύο ή περισσότερων εικόνων ώστε να αποτελούν μία (φωτο-μοντάζ).
- Κατάτμηση μίας εικόνας σε περιοχές, με στόχο τον καθορισμό των τομέων ενδιαφέροντος στην εν λόγω εικόνα (Regions of Interest, ROI) όπως κατάτμηση σε προσκήνιο και παρασκήνιο.
- Αποκατάσταση με στόχο την εξαγωγή μιας εκδοχής της εικόνας από μια ενθόρυβη / θολωμένη / παραμορφωμένη εικόνα εισόδου.

ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ

Η μηχανική, υπολογιστική ή τεχνητή όραση είναι ένα επιστημονικό πεδίο της τεχνητής νοημοσύνης που σκοπό έχει την αλγοριθμική αναπαραγωγή της αίσθησης της όρασης, από κάποιον ηλεκτρονικό υπολογιστή. Η μηχανική όραση εμπλέκεται στη σχεδίαση και κατασκευή συστημάτων που λαμβάνουν και αναλύουν δεδομένα από ψηφιακές εικόνες τόσο θεωρητικά όσο και τεχνολογικά. Τα δεδομένα προς επεξεργασία δύνανται να είναι φωτογραφίες, βίντεο, όψεις απο πολλαπλές κάμερες, πολυδιάστατες εικόνες από ιατρικό σαρωτή, 3D σαρωτή κλπ. Τεχνητή όραση θα μπορούσε να θεωρηθεί η μετατροπή των δεδομένων από μια φωτογραφική μηχανή ή μια βιντεοκάμερα σε ένα συμπέρασμα ή μια νέα αναπαράσταση. Η μηχανική όραση σκοπό έχει την εφαρμογή θεωριών και μοντέλων στην κατασκευή μηχανικών συστημάτων με δυνατότητες παραπλήσιας της όρασης. Τα συστήματα αυτά μπορεί να έχουν εφαρμογές όπως:

1. Έλεγχος διαδικασιών (π.χ. σε βιομηχανίες ή ένα αυτόνομο όχημα).
2. Ανίχνευση συμβάντων (π.χ. οπτική επιτήρηση, εντοπισμός ουσιών).
3. Οργάνωση πληροφοριών (π.χ. ευρετηριοποίηση βάσεων δεδομένων και ακολουθιών εικόνων).
4. Εξομίωση αντικειμένων και περιβάλλοντων (π.χ. βιομηχανική επιθεώρηση, ιατρική ανάλυση εικόνας ή τοπογραφική εξομίωση).
5. Αλληλεπίδραση χρηστών με υπολογιστικά συστήματα (π.χ. ως είσοδος σε μια συσκευή επικοινωνίας ανθρώπου/μηχανής).

Η OpenCV (Open Source Computer Vision Library) είναι μια βιβλιοθήκη ελεύθερου λογισμικού η οποία αναπτύχθηκε από την Intel και αφορά στην επεξεργασία εικόνας. Δημιουργήθηκε για τη παροχή υπηρεσιών κάτω από μία ενιαία υποδομή για εφαρμογές μηχανικής όρασης καθώς και να χρησιμοποιηθεί δραστικά σε εμπορικά προϊόντα. Παρέχει μία μεσαίου έως υψηλού επιπέδου διασύνδεση εφαρμογών με περίπου τριακόσιες συναρτήσεις. Η ανάπτυξη επικοινωνίας ανθρώπου με υπολογιστή, η ανίχνευση, απομόνωση και αναγνώριση αντικειμένων, η ανίχνευση και αναγνώριση προσώπων, η κατανόηση και παρακολούθηση κίνησης είναι μερικά από τα πεδία μηχανικής όρασης που καλύπτει. Υποστηρίζει την C++, Python, Java και Matlab γλώσσες προγραμματισμού και λειτουργεί σε όλα τα λογισμικά. Windows, Linux, Android και MacOS. Οι αλγόριθμοι της OpenCV είναι βελτιστοποιημένοι για επεξεργαστές αρχιτεκτονικής Intel. Τέλος, η δημιουργία της OpenCV αποσκοπεί και στην δημιουργία μιας κοινότητας ανοικτού λογισμικού, σχετική με την μηχανική όραση, η οποία θα αναπτύσσει σύγχρονες μεθόδους επεξεργασίας εικόνας σε ένα συνεχώς αναπτυσσόμενο τεχνολογικό περιβάλλον.

Όνομα Πλακέτας	Raspberry Pi 4
Επεξεργαστής	Broadcom BCM2711B0
Αρχιτεκτονική	ARMv8-A (64/32-bit)
Πυρήνες	4× Cortex-A72 1.5 GHz
Κάρτα γραφικών	Broadcom VideoCore VI @ 500 MHz
Μνήμη RAM	DDR3 1GB (960MB στο σύστημα)
Ασύρματη επικοινωνία	10/100/1000 Mbit/s Ethernet, 802.11b/g/n/ac wireless, Bluetooth 5.0
GPIO Pins	17
Διαστάσεις	85.60 mm × 56.5 mm × 17mm
Βάρος	46g
Τροφοδοσία	5V / 1.5A

Πίνακας 1 Τεχνικά Χαρακτηριστικά Raspberry Pi

ΚΕΦΑΛΑΙΟ 2°

Στο παρόν κεφάλαιο παρουσιάζονται η φυσική κατασκευή και επιλογή των επιμέρους κομματιών με τα τεχνικά χαρακτηριστικά τους καθώς και η αρχιτεκτονική που ακολουθήθηκε κατά την ανάπτυξη του λογισμικού.

ΚΑΤΑΣΚΕΥΗ

Για την επίτευξη των προαναφερθέντων στόχων, φαίνεται πως η χρήση κάποιου απλού μικροελεγκτή (Arduino, ESP8266) δεν προσφέρει επαρκή επεξεργαστική ισχύ για επεξεργασία εικόνας σε πραγματικό χρόνο. Κατά συνέπεια, για την διασφάλιση της αυτονομίας του οχήματος χρησιμοποιήθηκε η πλακέτα Raspberry Pi 4.

Πρόκειται για έναν υπολογιστή πλακέτας (Single Board Computer - SBC), του οποίου τα τεχνικά χαρακτηριστικά αναφέρονται στον παρακάτω πίνακα. Καθότι πλήρης υπολογιστής έχει λειτουργικό σύστημα τύπου Linux (έκδοση Raspbian Buster Lite) και υποστηρίζει πολλές γλώσσες προγραμματισμού. Για τις ανάγκες της μηχανικής όρασης μέσω OpenCV χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python 3.

Για την απρόσκοπτη λειτουργία του Raspberry Pi χρησιμοποιήθηκε η βοηθητική πλακέτα Raspi UPS HAT V1.0, η οποία επιτρέπει και την ταυτόχρονη απομακρυσμένη παρακολούθηση της τάσης της μπαταρίας μέσω του προτύπου I2C. Στην πλακέτα αυτή τοποθετήθηκε μπαταρία 1 cell 3.7V 2900mAh.



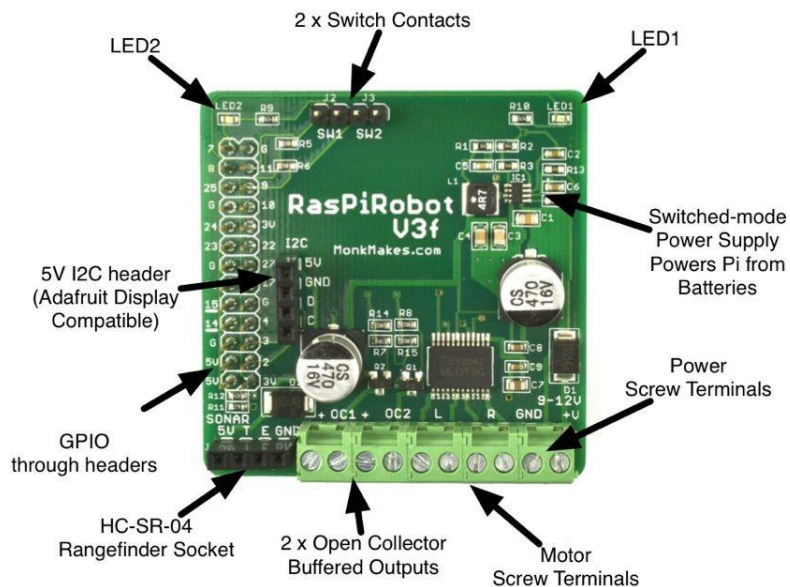
Εικόνα 1 Λεπτομέρειες από την πακέτα UPS

Για την σύλληψη εικόνας εγκαταστάθηκε το επίσημο περιφερειακό κάμερας Raspberry Pi Camera V2, του οποίου τα τεχνικά χαρακτηριστικά φαίνονται στον παρακάτω πίνακα.

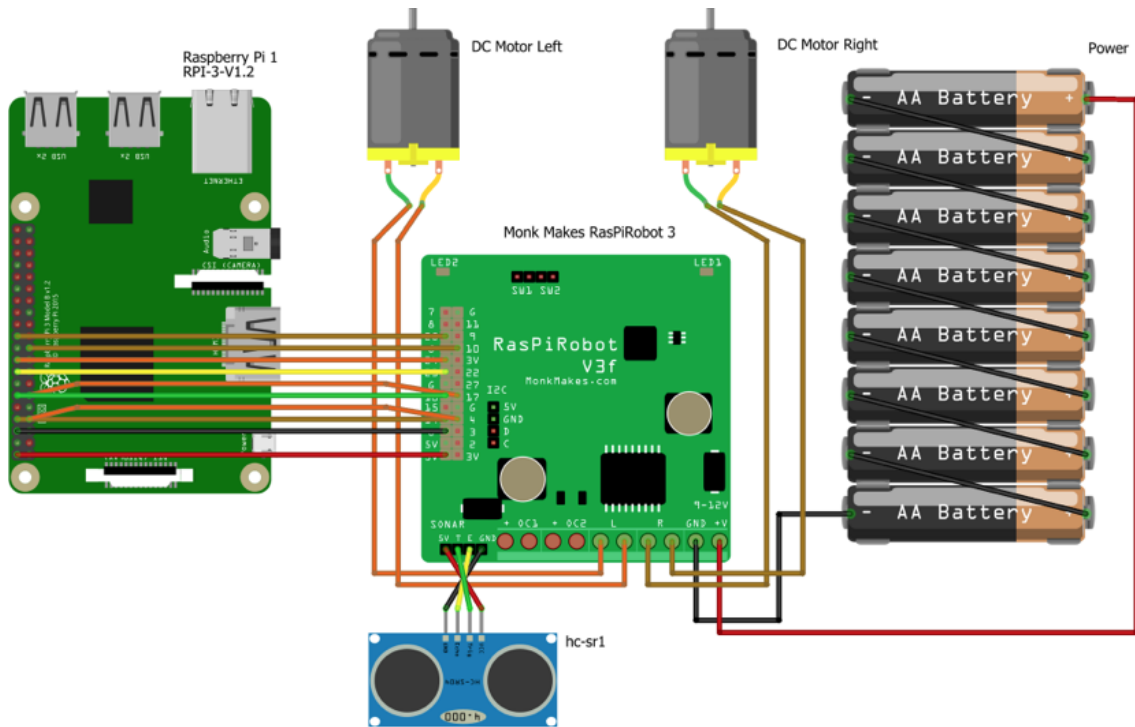
Όνομα Πλακέτας	Raspberry Pi Camera Module v2
Αισθητήρας	OmniVision OV5647
Μεγαapixel	5mp
Ανάλυση Αισθητήρα	Φωτογραφίας: 2592x1944 Βίντεο: 1080p 30fps
Σύνδεση	CSI
Διαστάσεις	25 mm × 20 mm × 9 mm

Πίνακας 2 Τεχνικά Χαρακτηριστικά Κάμερας

Για την κίνηση του rover εγκαταστάθηκαν τέσσερις κινητήρες με ενσωματωμένο κιβώτιο μείωσης ταχύτητας σχέσεως 150:1, οι οποίοι ελέγχονται μέσω του πρόσθετου RasPiRobot. Το πρόσθετο αυτό κουμπώνει πάνω στο Raspberry Pi στα GPIO pins και προσφέρει επιπλέον λειτουργικότητα.

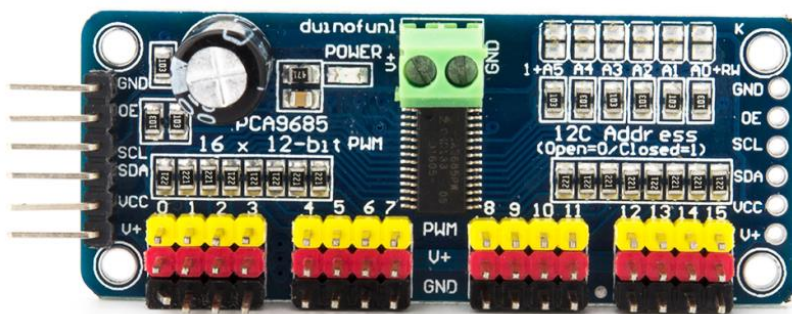


Εικόνα 2 Λεπτομέρειες από την πλακέτα RasPiRobot



Εικόνα 3 Ενδεικτική συνδεσμολογία της πλακέτας RasPiRobot

Στην πλακέτα αυτή επιπλέον συνδέεται μέσω I2C δίαυλου η βοηθητική πλακέτα PCA9685. Η πλακέτα αυτή επιτρέπει τον έλεγχο παραπάνω κινητήρων τύπου servo στους οποίους έχουν ανατεθεί επικουρικές κινήσεις (κίνηση κάμερας, δαγκάνα). Κατά την διάρκεια σεναρίου ρυμούλκησης, η πλοήγηση στον χώρο μπορεί να γίνει με αισθητήρα υπερήχων (HC-SR04) τοποθετημένο στην αντίστοιχη υποδοχή.



Εικόνα 4 Η πλακέτα PCA9685

Επιπλέον, για την κατασκευή της δαγκάνας χρησιμοποιήθηκε το kit της Actobotics. Για την κίνηση της κάμερας χρησιμοποιήθηκε ειδική μηχανική κατασκευή στην οποία συνδέθηκαν δύο κινητήρες τύπου servo MG90.



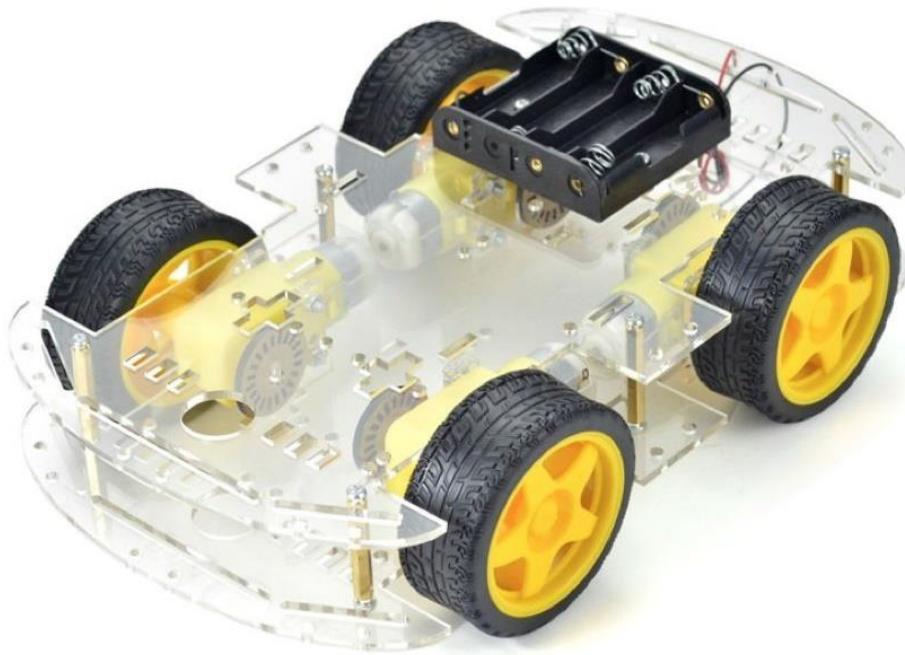
Εικόνα 5 Ο μηχανισμός δαγκάνας που χρησιμοποιήθηκε



Εικόνα 6 Ο μηχανισμός κίνησης της κάμερας

Η τροφοδοσία των κινητήρων κίνησης και servo ελέγχεται προγραμματιστικά μέσω αυτής της πλακέτας και προέρχεται από μια συστοιχία δύο μπαταριών τύπου 18650 1850mAh συνδεδεμένων σε σειρά, που πλήρως φορτισμένες αποδίδουν 8.4V. Οι μπαταρίες έχουν τοποθετηθεί σε συμμετρικά αντίθετη πλευρά με την δαγκάνα ώστε να υπάρχει συνέχεια πρόσφυση δύο αντίθετων τροχών.

Τα κομμάτια του οχήματος προήλθαν από ένα kit για τετρακίνητο όχημα από Plexiglass.

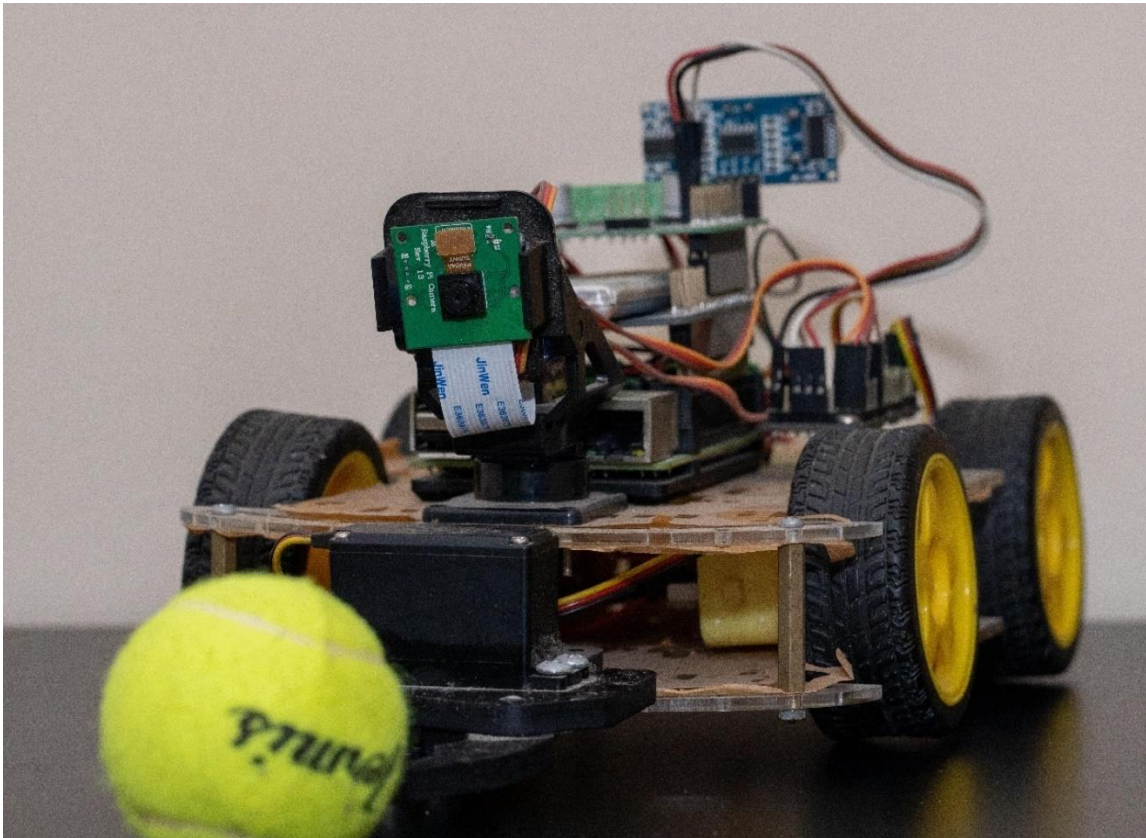


Εικόνα 7 Ο σκελετός και οι κινητήρες του οχήματος

Η συνδεσμολογία του οχήματος είναι η ακόλουθη:

- Οι πλακέτες έχουν στοιβαχθεί με την εξής σειρά: Raspberry PI > UPS HAT > RasPiRobot.
- Με αντιστοίχιση των pin του διαύλου I2C έχει συνδεθεί πάνω στο RasPiRobot η πλακέτα PCA9685 στην οποία με την σειρά τους συνδέονται τα βοηθητικά servo στις διαθέσιμες θύρες.
- Οι ακροδέκτες των μπαταριών έχουν συνδεθεί με την είσοδο ρεύματος του RasPiRobot.
- Οι κινητήρες στις αντίστοιχες εξόδους του RasPiRobot ανά πλευρά ζευγαρωμένοι. Η τάση τροφοδοσίας τους είναι μεταβλητή μέσω του λογισμικού.

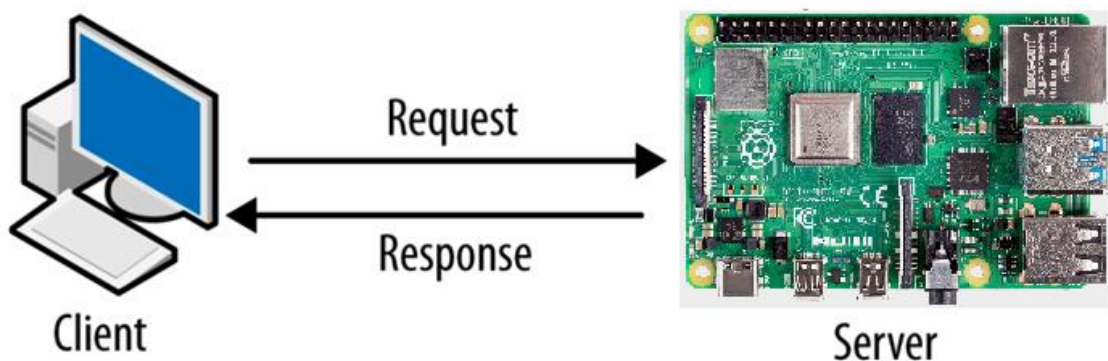
Για την εκμετάλλευση της κάμερας μέσω μηχανικής όρασης υπάρχουν πολλά frameworks από εταιρείες τεχνολογικού κολοσσούς όπως τα Microsoft Computer Vision, Google Cloud Vision API, Amazon Rekognition, ωστόσο το OpenCV έχει αγκαλιαστεί περισσότερο με την κοινότητα του Raspberry PI και υπάρχει μεγαλύτερη διαθέσιμη υποστήριξη σε οδηγούς εκμάθησης. Για το λόγο αυτό επιλέξαμε να κάνουμε τα πρώτα μας βήματα στο framework αυτό, χρησιμοποιώντας ως μέσο προγραμματισμού την γλώσσα Python 3. Η εγκατάσταση της βιβλιοθήκης δεν είναι και εύκολη καθώς υπάρχουν λεπτά σημεία, ενώ η μεταγλώττιση μπορεί να πάρει άνω των 3 ωρών στο Raspberry Pi και εξαρτάται άμεσα από την έκδοσή της πλακέτας. Μετά από αυτήν την χρονοβόρα διαδικασία ωστόσο είναι βελτιστοποιημένη για τον υπολογιστή πλακέτας PI.



Εικόνα 8 Η τελική κατασκευή

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ROVER

Για τον έλεγχο λειτουργίας και χειρισμού του rover έχει δημιουργηθεί γραφικό περιβάλλον με τεχνολογίες web και αρχιτεκτονικής client – server σε αρχιτεκτονικό στυλ REST.



Εικόνα 9 Μοντέλο επικοινωνίας

Το Representational State Transfer (REST) αποτελεί ένα αρχιτεκτονικό στυλ κατανεμημένων συστημάτων υπερμέσων. Αποτελείται από εξυπηρετητές (servers) και πελάτες (clients). Οι clients εκκινούν την αποστολή αιτημάτων στους servers, οι οποίοι με την σειρά τους τα επεξεργάζονται ώστε να επιστρέψουν κατάλληλες απαντήσεις. Σε κάθε αίτημα ή μήνυμα

προς τους εξυπηρετητές αποστέλλεται σε συγκεκριμένο αντικείμενο της εφαρμογής το οποίο ονομάζεται πόρος (resource). Κάθε πόρος χαρακτηρίζεται από μοναδική διεύθυνση στην οποία αντιστοιχίζεται κάθε αίτημα. Οι απαντήσεις αντιστοιχίζονται στον πελάτη σε αναπαραστάσεις συγκεκριμένων πόρων για την δεδομένη χρονική στιγμή του αιτήματος. Η ονομασία της αρχιτεκτονικής REST είναι παράγωγο λοιπόν της έννοιας της μεταφοράς κατάστασης στον πελάτη. Με κάθε νέο αίτημα επιστρέφεται μια ανάλογη απόκριση στον πελάτη με τη μορφή αναπαράστασης με αποτέλεσμα να μεταβαίνει αυτός σε μια νέα κατάσταση.

Πρωταρχικές αρχές του REST είναι:

- Μια υλοποίηση REST αποτελείται από συστατικά δικτύου (components) τα οποία διαιρούνται σε πόρους.

- Κάθε πόρος χαρακτηρίζεται από έναν μοναδικό καθολικό ταυτοποιητή (URI).

- Η μεταφορά κατάστασης μεταξύ του πελάτη και του πόρου πραγματοποιείται μέσα από μια ομοιόμορφη διεπαφή (uniform interface). Το πρότυπο της ανταλλαγής πληροφορίας χαρακτηρίζεται από ένα καθορισμένο σύνολο αυστηρά ορισμένων λειτουργιών και ένα σύνολο από τύπους περιεχομένου. Η χρήση αυτών των δύο αναδεικνύουν την σημασιολογία της επικοινωνίας. Στην παρούσα εργασία η μεταφορά αναπαραστάσεων των πόρων σε μορφή internet media types χρησιμοποιεί αντικείμενα τύπου JSON.

ΚΕΦΑΛΑΙΟ 3°

Στο παρόν κεφάλαιο παρουσιάζονται στιγμιότυπα από το λογισμικό που δημιουργήθηκε ενώ περιγράφονται και οι κύριοι αλγόριθμοι που υλοποιήθηκαν.

WEB USER INTERFACE

Η πρόσβαση στο γραφικό περιβάλλον γίνεται με σύνδεση TCP / IP σε τοπικό δίκτυο χρησιμοποιώντας ένα φυλλομετρητή όπως ο Google Chrome ή Mozilla Firefox. Απαιτείται να είναι ενεργή η εκτέλεση JS scripts.

Αρχικά, ο χειριστής μπορεί να επισκεφθεί τη σελίδα διαχείρισης αφού πρώτα συνδεθεί στο ίδιο τοπικό δίκτυο με το rover. Η διεύθυνση του rover είναι η <http://raspberrypi.local:5000>, το τμήμα της διεύθυνσης raspberrypi.local προκύπτει από το hostname που έχει δηλωθεί στο λειτουργικό σύστημα Raspbian και χρησιμοποιείται αυτόματα από την υπηρεσία mdns ή bonjour ανάλογα με το λειτουργικό σύστημα του υπολογιστή πελάτη.

Ο web application server που δημιουργείται κατά την εκτέλεση του προγράμματος του rover δέχεται συνδέσεις στη θύρα 5000 επιστρέφοντας την προεπιλεγμένη ιστοσελίδα διαχείρισης η οποία λειτουργεί με τεχνολογία ajax σε συνδυασμό με REST API.

Λειτουργίες που εκτελούνται μέσω του web interface:

1. Έλεγχος καλής λειτουργίας του Raspberry Pi/Server
 - Κατάσταση μπαταρίας.
 - Διαγνωστικά δικτύου (Σύνδεση Internet, διεύθυνση internet, διεύθυνση τοπικού δικτύου, όνομα δικτύου WIFI).
 - Διαγνωστικά κατάστασης λειτουργικού συστήματος (Χρήση CPU, RAM, DISK, LOAD συστήματος, UPTIME).
2. Manual Mode
 - Κίνηση προς 4 κατευθύνσεις.
 - Άνοιγμα και κλείσιμο της δαγκάνας στο μπροστινό μέρος του rover.
 - Περιστροφή της κάμερας κατά X και Y.
3. Live Stream Mode
 - Χρήση της κάμερας στο μπροστινό μέρος του rover για παρακολούθηση του περιβάλλοντος σε πραγματικό χρόνο.
 - Κίνηση του rover μέσω ενός εικονικού joystick που βρίσκεται στο κέντρο του βίντεο μέσω touch ή εντολών του mouse.
4. AI Mode
 - Ενεργοποίηση αυτόνομης κίνησης με χρήση του OpenCV (Computer Vision) με στόχο την κίνηση προς ένα συγκεκριμένο αντικείμενο, ένα κίτρινο μπαλάκι.

Εικόνες από το διαχειριστικό περιβάλλον που δημιουργήθηκε:

The screenshot shows a web browser window displaying the 'Web Rover Dashboard'. The browser's address bar shows 'Not secure | 192.168.1.106:5000'. The dashboard has a navigation menu at the top with 'Dashboard', 'Manual Mode', 'Live Stream Drive', 'AI Mode', and 'About'. The main content area is titled 'Web Rover Dashboard' and contains several sections:

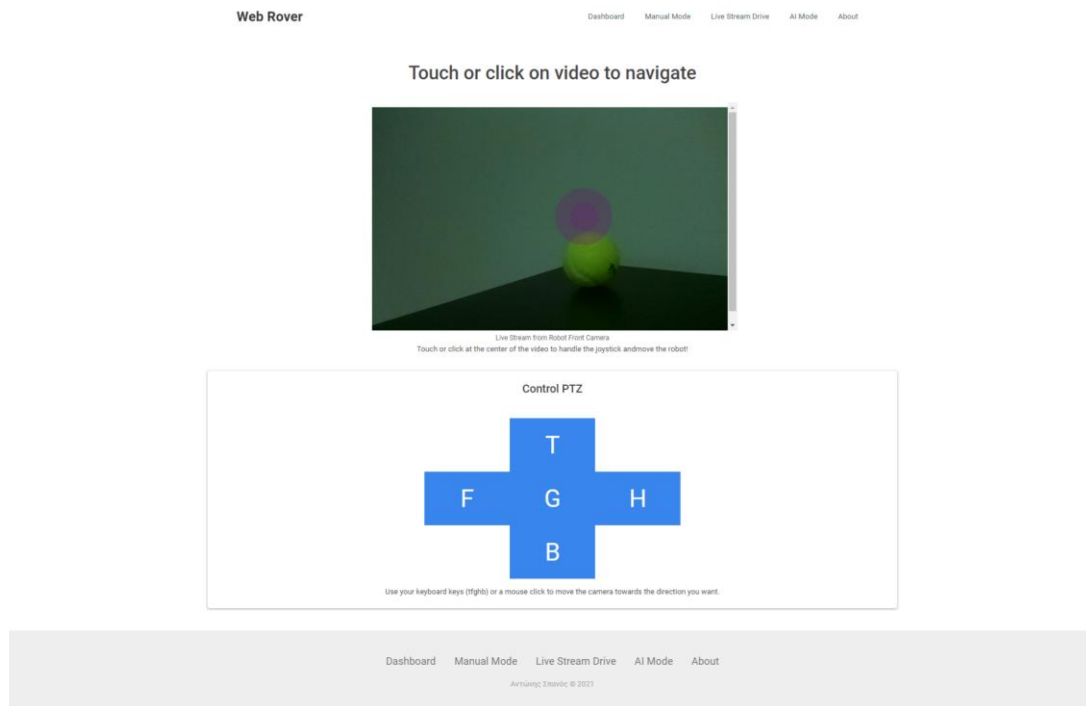
- RPI Battery Capacity:** A gauge showing 94% capacity.
- RPI Battery:** A table with 'Voltage: 3.02V' and 'Capacity: 94.02%'.
- RPI Status:** A table with 'Uptime: 0 days 0 hours 0 minutes', 'Ram Used: 14.3 %', 'CPU Used: 6.9 %', and 'Disk Used: 64.2 %'.
- Network Status:** A table with 'Online: true', 'WiFi: 62:0D:7F:4B:8C:2:4034:45668C', 'Local IP: 192.168.1.106', 'Internet IP: 77.49.82.155', and 'Server Port: 5000'.

Below these is an 'Actions' section with the heading 'Choose the mode you would like to access'. It features four cards:

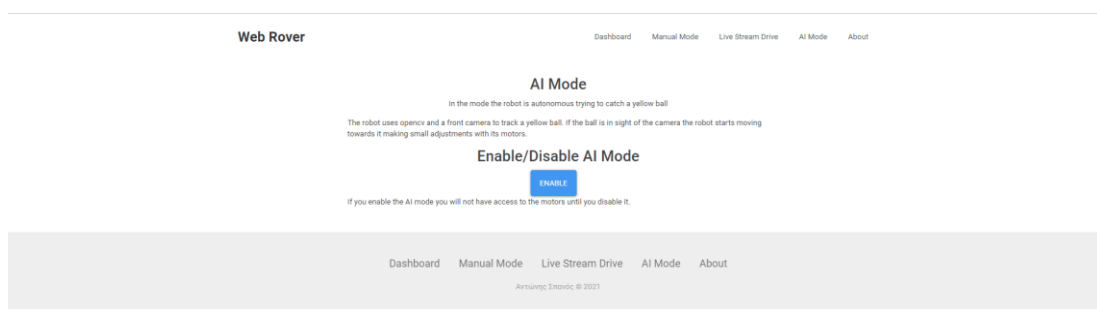
- Manual Mode:** Includes a hand icon and a 'Manual mode' button. Description: 'You can control the movement of the rover with your keyboard or mouse and toggle the front camera'.
- Live Stream Mode:** Includes a video camera icon. Description: 'Move the robot in realtime with a virtual joystick on top of a live video stream (front camera)'.
- AI Object Tracking Mode:** Includes an eye icon. Description: 'Let the robot track a yellow ball using its front camera'.
- Raspberry PI Diagnostics:** Includes an information icon. Description: 'Get an idea for the system performance'.

The dashboard footer contains the same navigation menu and a copyright notice: 'Αντώνιος Σπανός © 2021'.

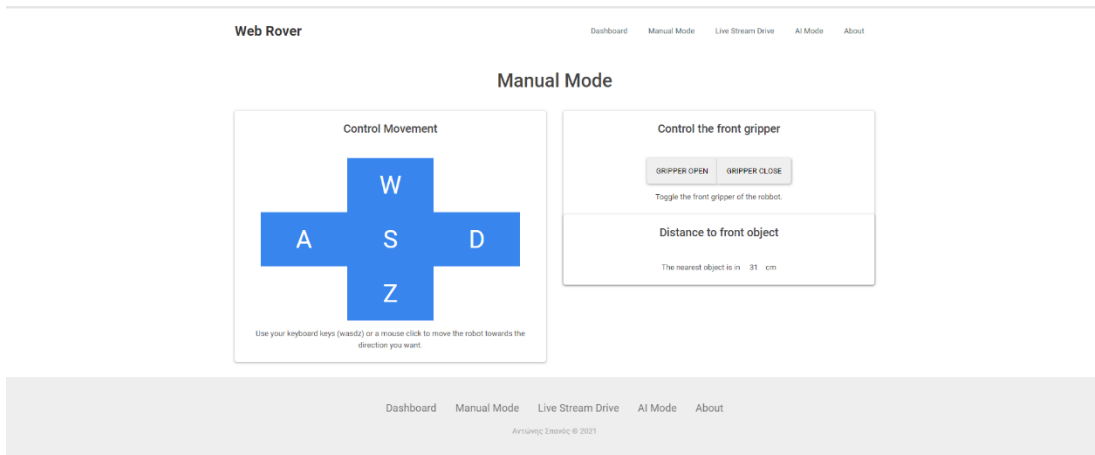
Εικόνα 10 Dashboard / Αρχική σελίδα



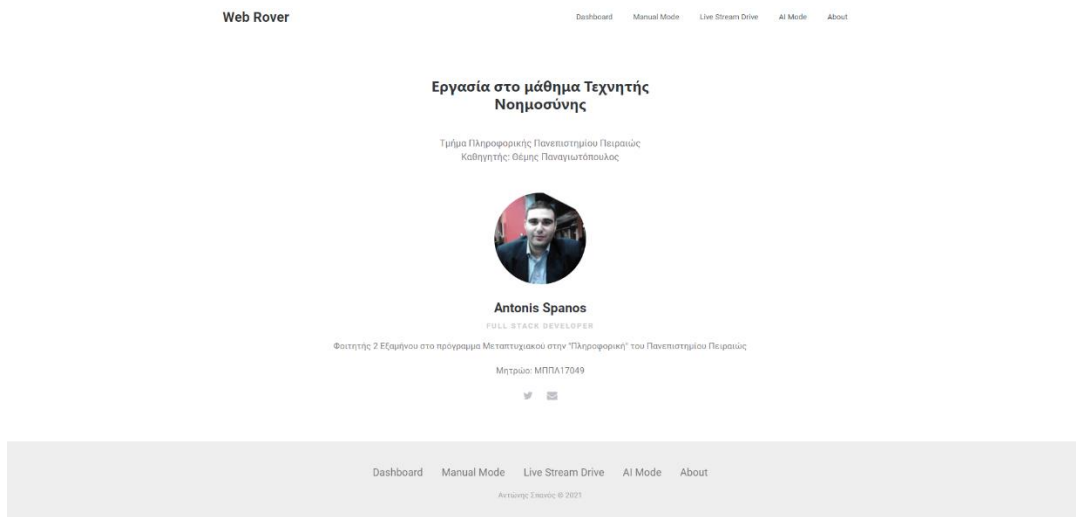
Εικόνα 11 Live Stream



Εικόνα 12 AI Mode / Αυτόνομη λειτουργία



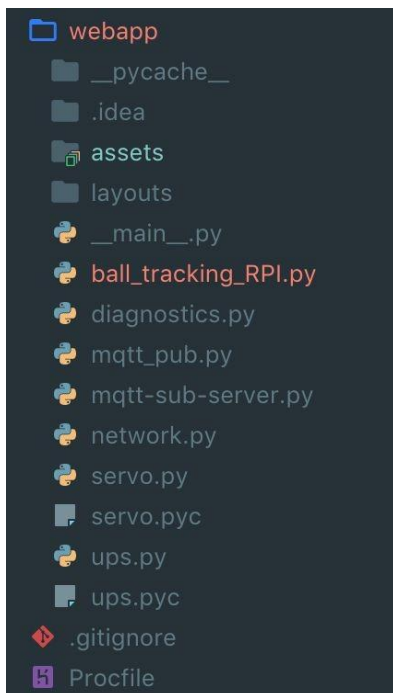
Εικόνα 13 Manual Mode / Κίνηση με κουμπιά



Εικόνα 13 Σελίδα Σχετικά-Επικοινωνίας

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ UI

Για την υλοποίηση του User Interface χρησιμοποιήθηκε η γλώσσα προγραμματισμού python σε συνδυασμό με HTML5, JavaScript και CSS.



Εικόνα 14 Τα αρχεία της εφαρμογής

Η εφαρμογή ξεκινάει με το αρχείο `__main__.py`.

Τα στατικά αρχεία όπως εικόνες, stylesheets (CSS) και αρχεία JavaScript (js) αποθηκεύονται στο φάκελο `assets`. Τα αρχεία HTML βρίσκονται στον φάκελο `layouts`.

Η κυρίως εφαρμογή αποτελείται από το αρχείο `__main__.py` το οποίο φορτώνει ως βιβλιοθήκες τα υπόλοιπα αρχεία python (`.py`) από τον φάκελο `webapp`.

Για την δημιουργία εξυπηρετητή χρησιμοποιούμε τη βιβλιοθήκη `bottle` η οποία είναι ένα micro-framework για εφαρμογές web. Το `bottle` είναι υπεύθυνο για να αντιστοιχίσει την κατάλληλη μέθοδο με το url που έχει δηλωθεί. Έτσι για παράδειγμα η αρχική σελίδα (dashboard) φορτώνει με βάση το παρακάτω κομμάτι κώδικα python

```
@get('/')
def index():
    return template('./webapp/layouts/dashboard', title="WebRover
Dashboard")
```

Δηλώνουμε αρχικά τον τρόπο που ο client/Browser θα ζητήσει κάτι από την εφαρμογή μας. Εδώ έχουμε ένα HTTP GET request στο / (root) της εφαρμογής το οποίο θα προκαλέσει την εκτέλεση της μεθόδου `index()` και εκείνη με τη σειρά της θα επιστρέψει στον client / Browser το αρχείο `html/webapp/layouts/dashboard.html` αφού περάσει ως όρισμα τον τίτλο της σελίδας.

Εδώ πρέπει να σημειώσουμε ότι η σελίδα `dashboard.html` αποτελείται από κλασικό κώδικα `html5` αλλά και μικρά κομμάτια κώδικα σε `Python`. Αυτού του τύπου οι σελίδες ονομάζονται `templates` και στη συγκεκριμένη εφαρμογή χρησιμοποιείται η μηχανή `SimpleTemplateEngine`.

Η κεντρική σελίδα όπως φαίνεται και από το πρώτο `screenshot` εμφανίζει πληροφορίες για την μπαταρία, το δίκτυο και το λειτουργικό σύστημα. Αυτές οι πληροφορίες εμφανίζονται στη σελίδα αφού ολοκληρωθεί η φόρτωση της μέσω κώδικα `JavaScript - Ajax`. Οι παραπάνω πληροφορίες ενημερώνονται σε σταθερά χρονικά διαστήματα χωρίς να χρειάζεται ανανέωση της σελίδας ή απλά με ένα κλικ του `mouse`. Ανάλογα με το είδος της πληροφορίας που χρειαζόμαστε από το λειτουργικό σύστημα (`Rasrbian`) έχει δημιουργηθεί το κατάλληλο αρχείο `python`.

Έτσι ο κώδικας που είναι υπεύθυνος για τις πληροφορίες του `UPS` βρίσκεται στο αρχείο `ups.py` και η κλάση λέγεται `UPS`. Ομοίως η κλάση `Servo` βρίσκεται στο αρχείο `servo.py`.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ OBJECT TRACKING

Η λειτουργία `object tracking` γίνεται μέσω κάμερας η οποία είναι τοποθετημένη στο μπροστινό τμήμα του `rover` επάνω σε ένα μηχανισμός `pan tilt`. Για την αναγνώριση αντικειμένων χρησιμοποιείται η βιβλιοθήκη `OpenCV` σε γλώσσα `Python`. Για την αυτόνομη λειτουργία, `AI-MODE` όπως την ονομάζουμε στο διαχειριστικό περιβάλλον `web`, δημιουργήθηκε το αρχείο `ball_tracking_RPI.py`

Το πρόγραμμα αναγνώρισης αντικειμένων δημιουργήθηκε ώστε να είναι ανεξάρτητο από τη `web` εφαρμογή. Η εφαρμογή `web` δημιουργεί ένα καινούριο `process` στο σύστημα όταν κάνουμε κλικ στο κουμπί `activate`, αν το ξαναπατήσουμε τότε η εφαρμογή `web` θα ψάξει τη διαδικασία `ball_tracking_RPI.py` στο σύστημα και θα την τερματίσει. Με τον παραπάνω τρόπο έχουμε τη δυνατότητα να ενεργοποιούμε /απενεργοποιούμε την αυτόνομη λειτουργία απομακρυσμένα.

Το αρχείο `ball_tracking_RPI.py` περιλαμβάνει μια κλάση `BallTracker` η οποία περιλαμβάνει τα δεδομένα και τις μεθόδους που είναι υπεύθυνες για την ανάλυση εικόνων (`frames`) αλλά και την κίνηση του `rover`.

Ένα αντικείμενο `BallTracker` χρειάζεται να γνωρίζει για την ύπαρξη του `rover` αλλά και της κάμερας. Συνεπώς δημιουργήθηκε το αντικείμενο `tracker` που είναι υπεύθυνο για την κίνηση και ανάλυση των εικόνων, χρησιμοποιώντας ένα αντικείμενο κάμερας (`camera = PiCamera()`) και για την κίνηση στον χώρο ένα αντικείμενο `rover` (`robot = rrb.RRB3(BATTERY_VOLTS, MOTOR_VOLTS)`)

Μετά τη δημιουργία του `BallTracker` (`tracker = BallTracker(robot, camera)`) αρκεί να καλέσουμε την μέθοδο `track_ball()` και να ξεκινήσει η ανάλυση κάθε `frame` για την αναζήτηση της μπάλας.

Ο αλγόριθμος που δημιουργήθηκε αναλύει κάθε frame του video και βρίσκει τις συντεταγμένες (x, y) του κέντρου του κύκλου που σχηματίζεται από την μπάλα. Ακολουθως γίνεται πρώτα έλεγχος στον άξονα των x για το αν η μπάλα βρίσκεται στο κέντρο της εικόνας. Αν δε βρίσκεται γίνεται διόρθωση με τους κινητήρες. Μόλις γίνουν οι διορθώσεις στον άξονα των x το robot σταματά στιγμιαία και ξεκινάει ο έλεγχος στον άξονα των y .

Βρέθηκε πειραματικά ότι οι συντεταγμένες του άξονα y έχουν σχέση ανάλογη με την απόσταση της μπάλας από την κάμερα, συνεπώς με κριτήριο την τιμή του y προστέθηκε η λειτουργία να ενεργοποιούνται οι κινητήρες ώστε το rover πλησιάζει το αντικείμενο.

ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ

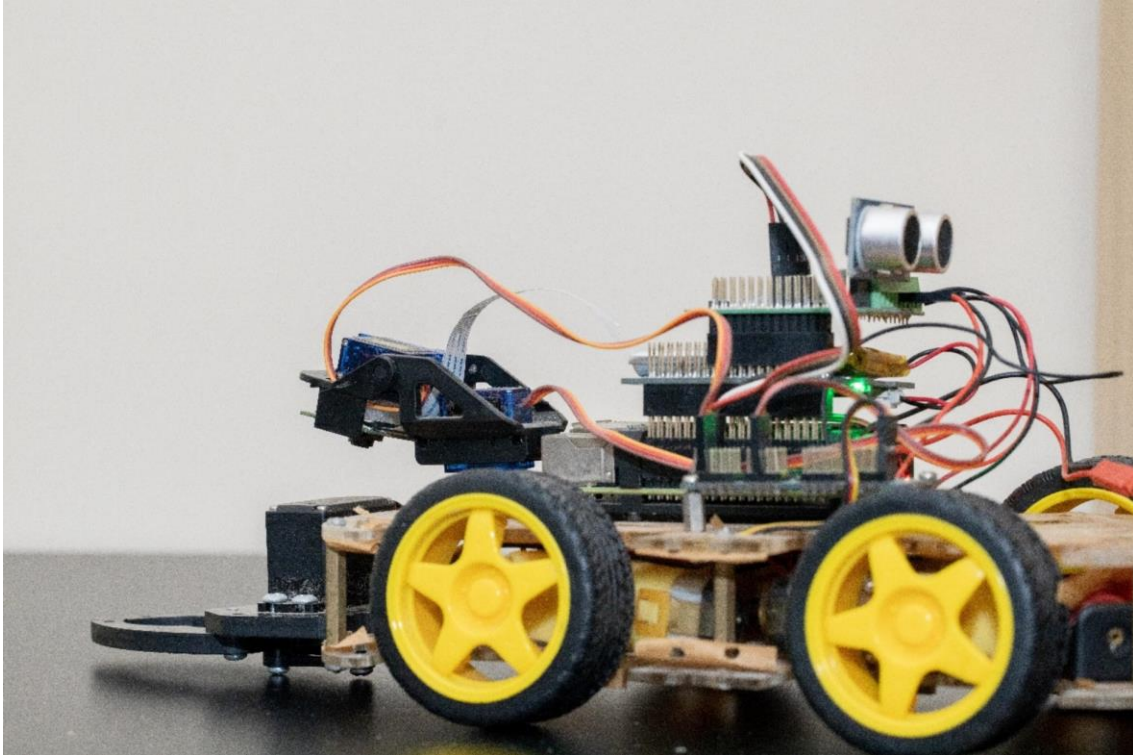
Για την ανάπτυξη της εργασίας χρησιμοποιήθηκαν μόνο δωρεάν λογισμικά ανοικτού κώδικα. Το λειτουργικό που χρησιμοποιήθηκε ήταν το Raspbian Buster, η γλώσσα προγραμματισμού Python 3.7 και σαν περιβάλλον προγραμματισμού το Visual Studio Code. Έγινε χρήση των Python βιβλιοθηκών:

- Bottle
- OpenCV
- Numpy
- Rrb3
- psutil, socket, subprocess, re, requests, json, time, smbus, os, sys

ΚΕΦΑΛΑΙΟ 4°

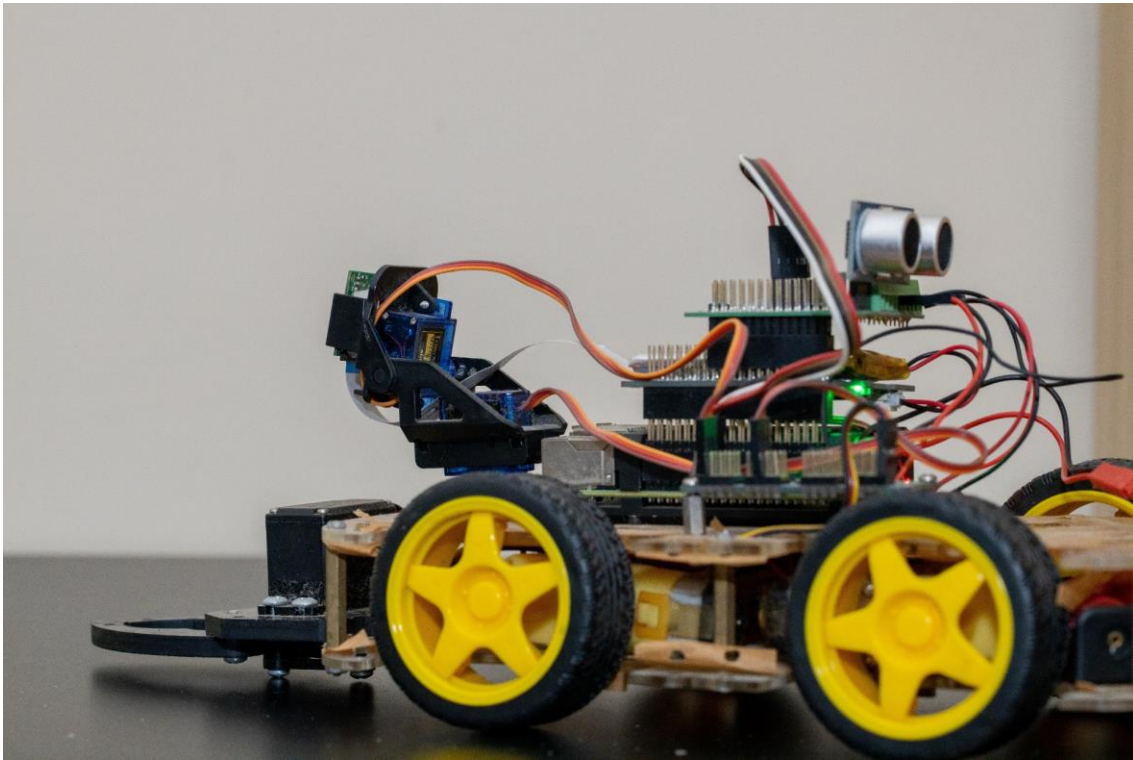
ΑΥΤΟΝΟΜΗ ΛΕΙΤΟΥΡΓΙΑ

Στο παρόν κεφάλαιο περιγράφεται μια ενδεικτική εκτέλεση του προγράμματος αναγνώρισης.



Εικόνα 15 Θέση εκκίνησης του Rover

Το web rover εκκινεί με την κάμερα προς τα κάτω σε μια τυχαία θέση στο επίπεδο. Με την βοήθεια της σελίδας Live Stream Mode και των πλήκτρων χειρισμού του PTZ μηχανισμού της κάμερας (T και F) οδηγούμε την κάμερα προς το κέντρο του rover και να κοιτάζει μπροστά.



Εικόνα 16 Τελική θέση της κάμερας του Rover

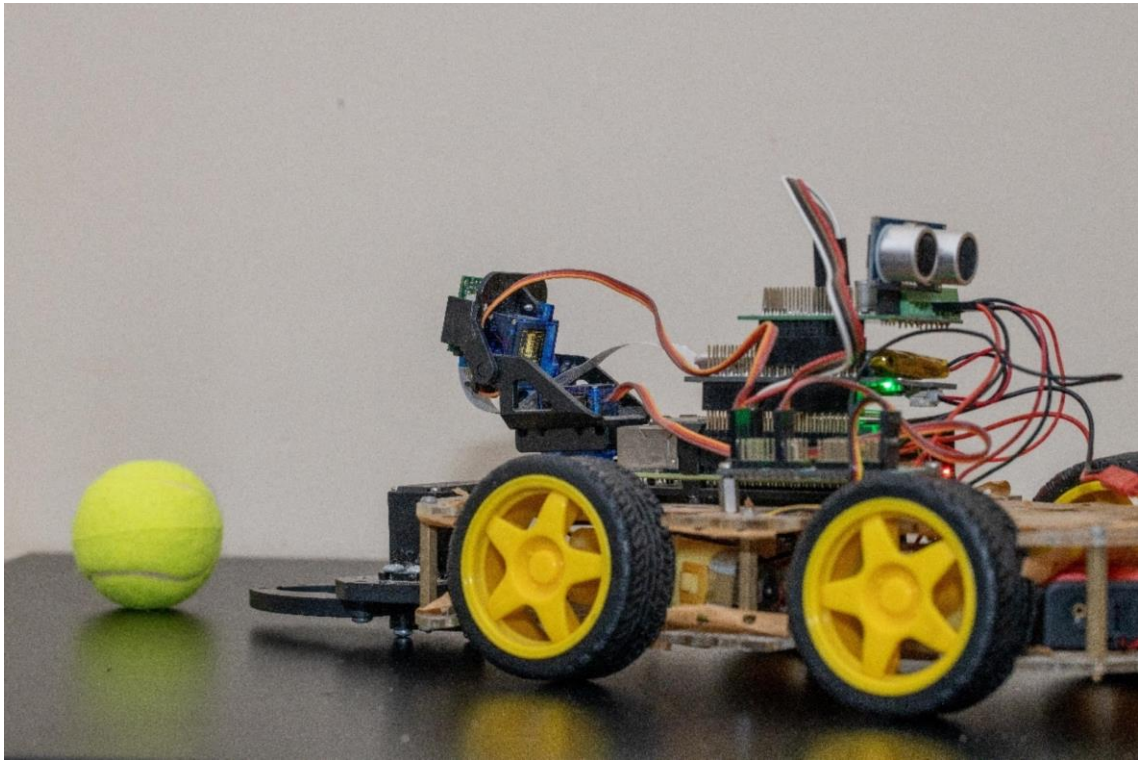
Στην συνέχεια, πλοηγούμαστε στην σελίδα AI-Mode και ενεργοποιούμε την τεχνητή νοημοσύνη. Αν έχουμε πρόσβαση στον server μπορούμε να δούμε τις παρακάτω εγγραφές από τον αλγόριθμο αναγνώρισης και τους υπολογισμούς του. Το rover σταματάει λίγο πιο μπροστά από την μπάλα.

```

creating a camera object
after raw capture. track_ball method
going left
x: 207.0 y: 392.5
found center, stoping
going forward
x: 296.0 y: 341.0
going right
x: 398.5 y: 289.2742004394531
going right
x: 363.0403747558594 y: 304.41363525390625
going right
x: 367.0929870605469 y: 278.0045471191406
going left
x: 262.6723937988281 y: 266.4621276855469
going left
x: 27.17009162902832 y: 260.67327880859375
going left
x: 7.5 y: 260.5
going left
x: 9.290698051452637 y: 264.7558288574219
going left
x: 15.0 y: 265.5
going left
x: 3.0 y: 254.5
going left
x: 74.8835220336914 y: 253.7982940673828
going right
x: 474.5 y: 227.0
going right
x: 565.0 y: 214.5
going left
x: 52.82758712768555 y: 243.6344757080078
going left
x: 19.5 y: 249.0
going right
x: 434.5 y: 205.0
going right
x: 543.0618286132812 y: 196.15721130371094
going right
x: 560.8104858398438 y: 190.0108642578125
going right
x: 563.1553344726562 y: 188.04776000976562
going right
x: 493.5 y: 192.0
found center, stoping
going forward
x: 338.5850830078125 y: 200.2510223388672
found center, stoping
going forward
x: 338.9324951171875 y: 234.17474365234375
found center, stoping
going forward
x: 344.5667724609375 y: 274.808837890625
found center, stoping
going forward
x: 348.425537109375 y: 311.4081115722656
going right
x: 368.18017578125 y: 330.63604736328125
going right
x: 382.29437255859375 y: 332.83392333984375

```

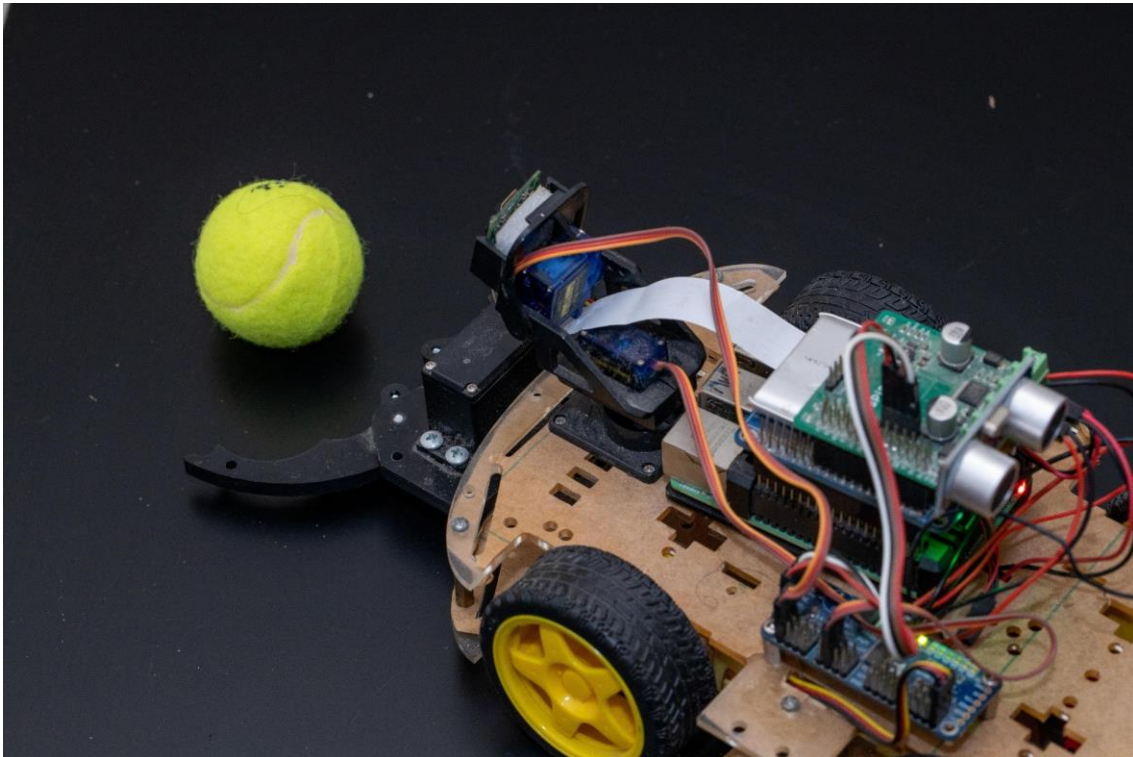
Εικόνα 17 Καταγραφές του server κατά την εκτέλεση της αναγνώρισης αντικειμένων.



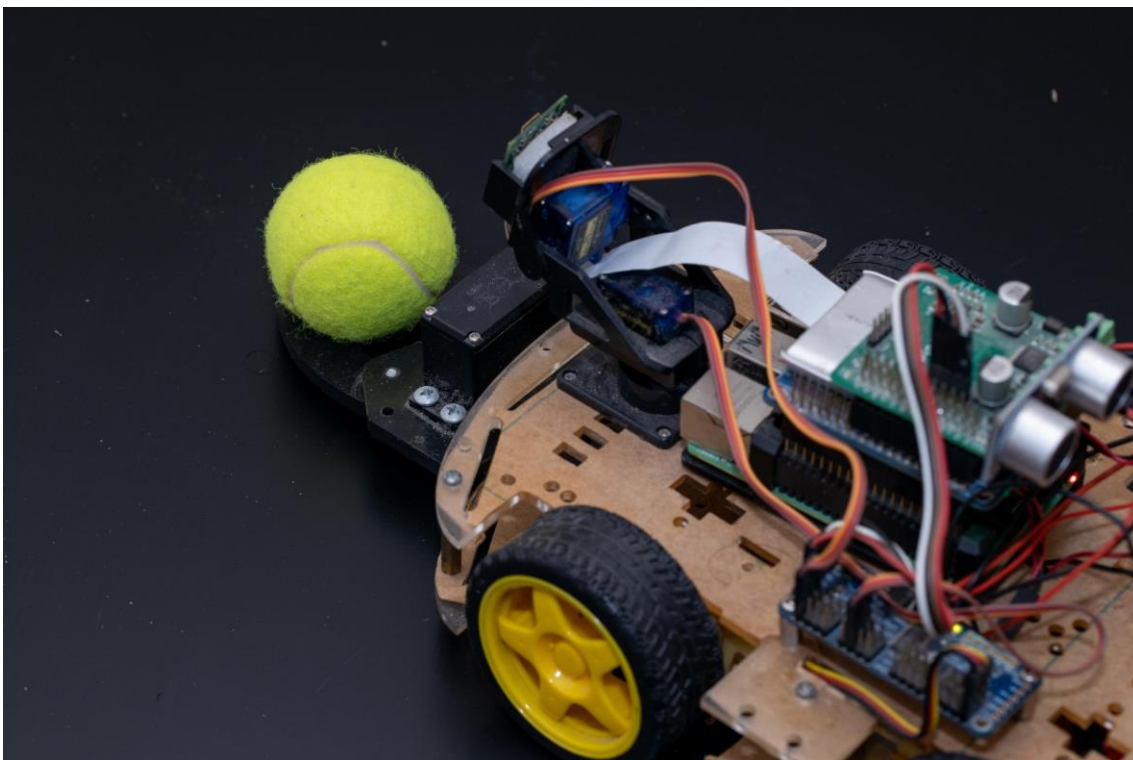
Εικόνα 18 Τελική θέση του rover.

ΧΕΙΡΟΚΙΝΗΤΗ ΛΕΙΤΟΥΡΓΙΑ

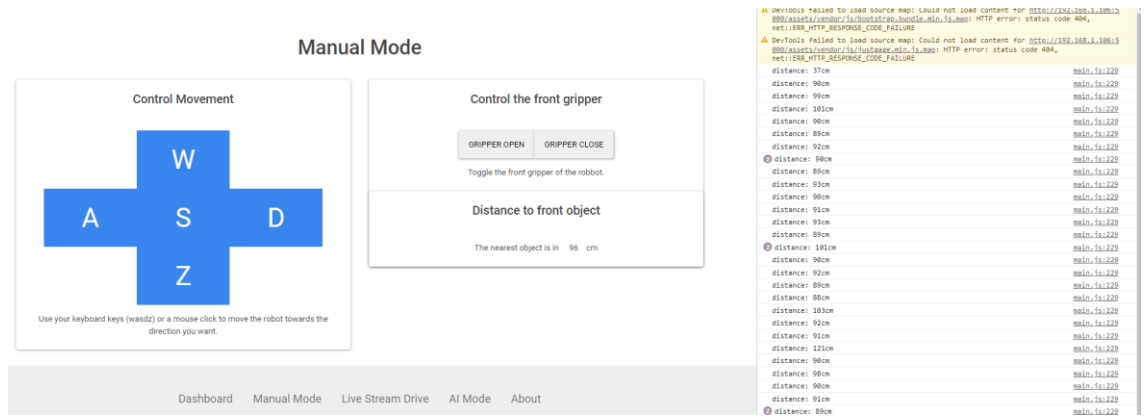
Στο παρόν κεφάλαιο περιγράφεται μια εκτέλεση κατά την χειροκίνητη λειτουργία. Κατά την διάρκεια της χειροκίνητης λειτουργίας το ρόβερ συμπεριφέρεται ως ένα τηλεκατευθυνόμενο αυτοκίνητο – παιχνίδι με χρήση WiFi. Στις παρακάτω εικόνες εμφανίζονται στιγμιότυπα από τις καταγραφές του server για τις εντολές κίνησης των μοτέρ, την διαρκή ενημέρωση της τιμής του αισθητήρα υπερήχων και την επιτυχή απόπειρα για λειτουργία της δαγκάνας.



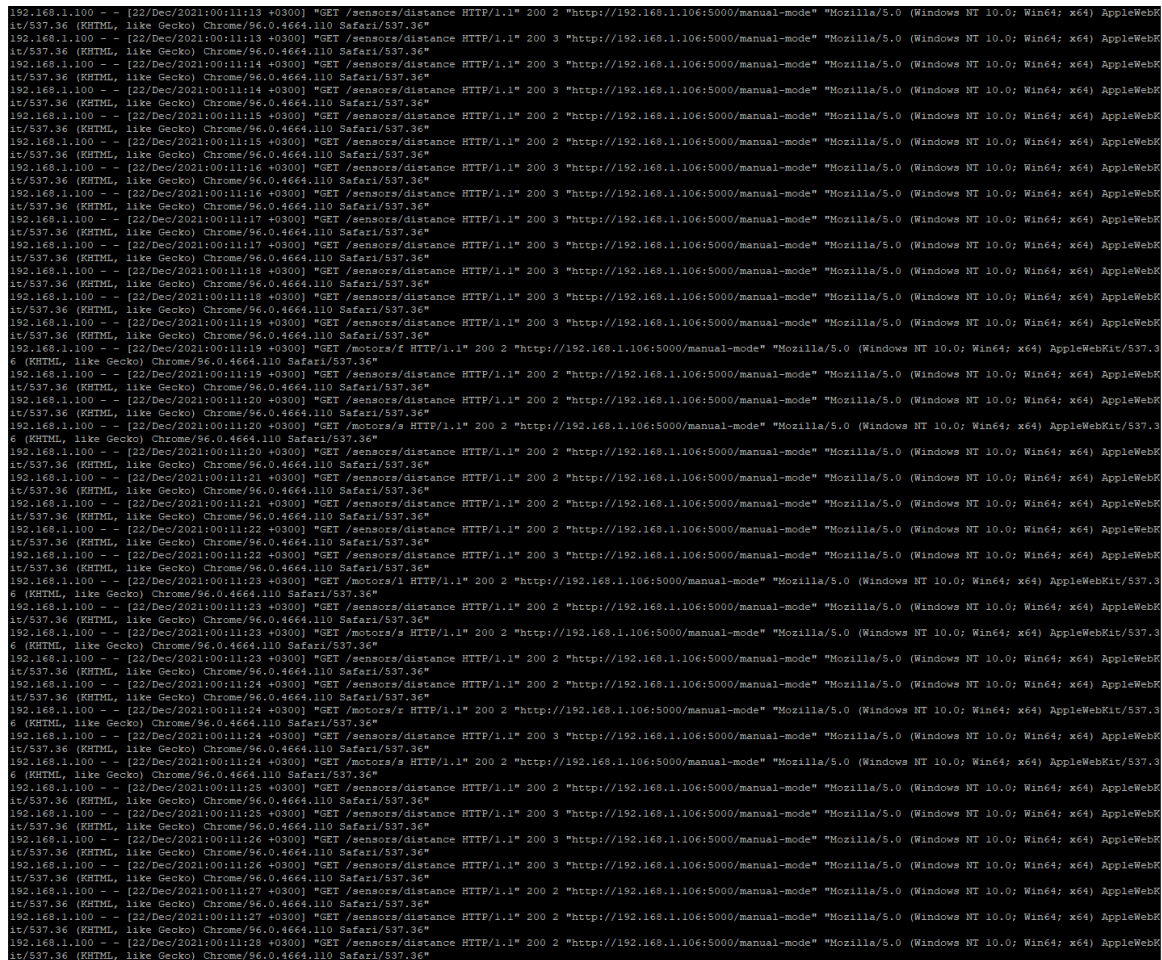
Εικόνα 19 Η δαγκάνα άνοιξε.



Εικόνα 20 Η δαγκάνα έκλεισε.



Εικόνα 21 Στιγμιότυπο της εκτέλεσης με ενεργά τα εργαλεία προγραμματιστή και καταγραφή των κλήσεων API.



Εικόνα 22 Στιγμιότυπο των καταγραφών των κλήσεων API στον server.

ΚΕΦΑΛΑΙΟ 5°

Στο παρόν κεφάλαιο περιγράφονται τα προβλήματα που αντιμετωπίστηκαν κατά την ανάπτυξη της κατασκευής και στην συνέχεια τα συμπεράσματα και μελλοντικές ενέργειες για την περαιτέρω ανάπτυξη της κατασκευής.

ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΠΑΡΟΥΣΙΑΣΤΗΚΑΝ

Κατά την εκπόνηση της εργασίας, αντιμετωπίστηκαν διάφορες δυσκολίες οι οποίες εστιάζονται κυρίως στην χαρτογράφηση. Για την δημιουργία ενός τρισδιάστατου χωρικού χάρτη (spatial 3d map) με χρήση οπτικών αισθητήρων (καμερών) απαιτείται η ύπαρξη δύο από αυτών.

Ωστόσο ο υπολογιστής πλακέτας Raspberry PI που χρησιμοποιεί η κατασκευή διαθέτει μόνο μια σύνδεση τύπου CSI. Κατά συνέπεια είναι αναγκαία η διασύνδεση δεύτερου αισθητήρα με χρήση άλλου διαύλου πχ τύπου USB. Στην περίπτωση αυτή πρέπει να υπάρχουν κοινές αναλύσεις μεταξύ των αισθητήρων. Λύση στο πρόβλημα αυτό προσφέρει η χρήση 2 ίδιων αισθητήρων μέσω διαύλου USB.

Η διασύνδεση αυτή ωστόσο με 2 ταυτόχρονες ροές βίντεο αρκετά υψηλής ανάλυσης παθαίνει κορεσμό σε σημείο που δεν είναι δυνατή η χρήση της μιας από της δύο συσκευές. Το πρόβλημα αυτό είναι ακόμα πιο έντονο λόγω της αρχιτεκτονικής του διαύλου USB στον επιλεγμένο υπολογιστή. Οι θύρες USB κάνουν χρήση δρομολογητή (hub) που διαμοιράζει τον κεντρικό δίαυλο με τον επεξεργαστή. Το πρόβλημα αυτό είναι πιθανόν να μην εμφανίζεται σε υπολογιστή πλακέτας με διαφορετική αρχιτεκτονική.

Ο κορεσμός του διαύλου USB με 2 ταυτόχρονες ροές βίντεο υψηλής ανάλυσης θα μπορούσε να αντιμετωπιστεί με χρήση ροής χαμηλότερης ανάλυσης πχ αντί για 1280x720 χρήση 640x480. Με τον τρόπο αυτό είναι δυνατή η προσπέλαση του προβλήματος μειωμένου εύρους ζώνης του USB ελεγκτή. Δεν υπερσκελίζεται όμως το πρόβλημα του καθολικού κλειστρου. Με τον όρο καθολικό κλείστρο εννοούμε την δυνατότητα τα λαμβάνουμε από τους αισθητήρες την ίδια χρονική στιγμή την αποτύπωση του χώρου. Λόγω της ψευδοπαράλληλης κάθε αισθητήρας παρουσιάζει ελαφρά διαφορετικό στιγμιότυπο του χώρου. Η διαφορά αυτή δημιουργεί ανακριβή δεδομένα προς επεξεργασία από τους αλγορίθμους χαρτογράφησης.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως γίνεται λοιπόν αντιληπτό η εξέλιξη της τεχνολογίας μας έχει επιτρέψει να κατασκευάζουμε μικρές κατασκευές ικανές να υλοποιούν συμπεριφορές που κάποτε απαιτούσαν το μέγεθος δωματίου. Σαν επέκταση του παρόντος ρομπότ θα μπορούσε να εξελιχθεί ο κώδικας του εντοπισμού να χρησιμοποιεί αυτοματισμό τύπου PID controller για που σαν αποτέλεσμα θα έχει την ομαλότερη και γρηγορότερη κίνηση του στον χώρο. Επιπλέον θα μπορούσε να γίνει διάσπαση των νημάτων που τρέχει το πρόγραμμα καθώς και προσθήκη επιταχυντή (πχ Coral AI Accelerator). Με τον τρόπο αυτό επιταχύνεται η λήψη και η επεξεργασία της εικόνας με αποτέλεσμα την ακριβότερη εκτέλεση του αλγορίθμου ειδικότερα σε περιπτώσεις που έχουμε κινούμενο αντικείμενο για παρακολούθηση. Τέλος, θα μπορούσε

να προστεθεί επιπλέον λειτουργικότητα επιστροφής στην βάση για φόρτιση π.χ., χαρτογράφησης χώρου ή και εντοπισμού ενός ευδιάκριτου αντικειμένου και μεταφορά του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΛΕΓΧΟΣ ΑΥΤΟΚΙΝΟΥΜΕΝΟΥ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΜΕ ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ ΚΑΙ ΧΡΗΣΗ ΔΕΙΚΤΩΝ ARUCO- ΣΑΦΙΚΟΥ ΕΥΣΤΑΘΙΑ - ΠΑΤΡΑ, ΟΚΤΩΒΡΙΟΣ 2018

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ, 3^Η ΕΚΔΟΣΗ, RAFAEL C. GONZALEZ-RICHARD E. WOODS, ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2011

DEEP LEARNING FOR COMPUTER VISION WITH PYTHON, ADRIAN ROSEBROCK, PYIMAGESEARCH, 2021

RASPIROBOTBOARD3 GITHUB (<https://github.com/simonmonk/raspirobotboard3>)

ADAFRUIT 16-CHANNEL 12-BIT PWM/SERVO DRIVER - I2C INTERFACE - PCA9685 (<https://www.adafruit.com/product/815>)

ΟΙ ΜΕΤΑΡΡΥΘΜΙΣΕΙΣ ΣΤΟ ΧΩΡΟ ΠΡΟΜΗΘΕΙΩΝ ΤΗΣ ΔΗΜΟΣΙΑΣ ΥΓΕΙΑΣ, ΚΩΣΤΟΥΛΑ-ΕΛΕΥΘΕΡΙΑ Ι. ΜΗΤΡΑΚΑ, ΟΣΕΛΟΤΟΣ, 2016