



ΣΤΡΑΤΙΩΤΙΚΗ ΣΧΟΛΗ ΕΥΕΛΠΙΔΩΝ  
Τμήμα Στρατιωτικών Επιστημών

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΔΙΔΡΥΜΑΤΙΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ  
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2016-17  
ΣΧΕΔΙΑΣΗ & ΕΠΕΞΕΡΓΑΣΙΑ

ΣΥΣΤΗΜΑΤΩΝ (SYSTEMS ENGINEERING)

(ΠΔ 96 / 2015 / ΦΕΚ 163Α' / 20.08.2014)



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
Σχολή Μηχανικών Παραγωγής & Διοίκησης

# ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Έλεγχος τηλεχειριζόμενου  
ρομποτικού οχήματος με τη βοήθεια  
κινητού τηλεφώνου (smartphone) ή  
φορητής συσκευής

Υπό:

ΔΗΜΗΤΡΙΟΥ ΜΠΟΤΩΝΗ

A.M.: 2014018012

ΑΠΡΙΛΙΟΣ 2017



Η Μεταπτυχιακή Διατριβή του Δημήτριου Μποτώνη εγκρίνεται:

**ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ**

Νικόλαος Δούκας, Επίκουρος Καθηγητής (Επιβλέπων) .....

Νικόλαος Μπάρδης, Αναπληρωτής Καθηγητής .....

Σάββας Πιπερίδης, Ε.ΔΙ.Π. ....

© Copyright υπό Δημήτριου Μποτώνη

Έτος 2017



# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

## ΠΕΡΙΛΗΨΗ

## ΕΙΣΑΓΩΓΗ

- §1. Στρατιωτικές εφαρμογές – το πρόβλημα
- §2. Σχετική έρευνα – αντίστοιχες εφαρμογές
- §3. Στόχοι της παρούσας εργασίας

## ΚΕΦΑΛΑΙΟ 1: Θεωρητικό υπόβαθρο

- §1. Παρουσίαση σχεδίασης
- §2. Τεχνολογίες υποστήριξης

## ΚΕΦΑΛΑΙΟ 2: Σχεδίαση - κατασκευή

- §1. Σχεδίαση
- §2. Επεξήγηση μεθόδου αποφυγής εμποδίων
- §3. Επεξήγηση μεθόδου επιστροφής στο φάρο υπέρυθρων

## ΚΕΦΑΛΑΙΟ 3: Πειράματα και αποτελέσματα

- §1. Μέτρηση ακρίβειας ανίχνευσης φάρου
- §2. Μέτρηση μέγιστης απόστασης προσέγγισης στο φάρο
- §3. Παρεμβολές από το ορατό φως – παρεμβολές από ισχυρές πηγές υπέρυθρων
- §4. Προσδιορισμός περιορισμών κίνησης
- §5. Περιορισμοί στην πυκνότητα και το μέγεθος των εμποδίων

## **ΚΕΦΑΛΑΙΟ 4: Αξιολόγηση των αποτελεσμάτων**

§1. Περιορισμοί

§2. Προτερήματα – σημεία υπεροχής

## **ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα και μελλοντική ανάπτυξη**

## **ΒΙΒΛΙΟΓΡΑΦΙΑ - ΙΣΤΟΤΟΠΟΙ**

### **ΠΑΡΑΡΤΗΜΑ Α: Εξαρτήματα**

### **ΠΑΡΑΡΤΗΜΑ Β: Συνδεσμολογία αισθητήρων – κινητήρων**

### **ΠΑΡΑΡΤΗΜΑ Γ: Κώδικας (Arduino Mega 2560 r3)**

### **ΠΑΡΑΡΤΗΜΑ Δ: Χειρισμός**

### **ΠΑΡΑΡΤΗΜΑ Ε: Πίνακες τεχνικών χαρακτηριστικών**





## ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή διατριβή έχει ως στόχο την κατασκευή ενός μη επανδρωμένου οχήματος το οποίο ελέγχεται με τηλεχειρισμό από φορητή συσκευή με λειτουργικό android (smartphone ή tablet) και θα μπορούσε να χρησιμοποιηθεί ως πλατφόρμα για τη διασύνδεση επιπλέον εξοπλισμού μελλοντικά.

Η κατασκευή έχει στρατιωτικό προσανατολισμό. Αυτό σημαίνει ότι το όχημα, εκτός από απλό τον τηλεχειρισμό, πρέπει να είναι ικανό να εκτελέσει κάποιες απλές αυτόνομες λειτουργίες, όπως η αυτόνομη περιπολία, η παρακολούθηση ενός φάρου και εκτέλεση πορείας προς αυτόν (είτε βρίσκεται σταθερά σε ένα σημείο είτε κινείται) και να ενσωματώνει μια υποτυπώδη τεχνική αποφυγής εμποδίων, οι οποίες θεωρούνται ως βασικό υπόβαθρο για την εκτέλεση πολλαπλών μελλοντικών αποστολών.

Τέλος, λόγω της αύξησης των επιχειρήσεων εντός κατοικημένων τόπων και κτιρίων, αποφασίστηκε το όχημα να κατασκευαστεί ως όχημα εσωτερικού χώρου.



# ΕΙΣΑΓΩΓΗ

## §1. Στρατιωτικές εφαρμογές – το πρόβλημα

Οι σύγχρονες επιχειρήσεις, σε μεγάλο ποσοστό, λαμβάνουν χώρα εντός κτιρίων και κατοικημένων τόπων γενικά. Για το σκοπό αυτό χρειάζονται μικρά οχήματα, με μεγάλη αυτονομία, με δυνατότητες τηλεχειρισμού και αυτόνομης κίνησης εντός εσωτερικών χώρων. Τα οχήματα αυτά θα πρέπει να μπορούν να αναλάβουν βοηθητικές αποστολές που θα διευκολύνουν τις επιχειρήσεις των τμημάτων που ενεργούν ή πρόκειται να ενεργήσουν στον εν λόγω χώρο, με σκοπό να μειώσουν τις απώλειες σε έμπυχο δυναμικό.

Τα χαρακτηριστικά που καλούνται να διαθέτουν αυτά τα οχήματα είναι κυρίως τα εξής:

1. Ικανοποιητική αυτονομία.
2. Ικανότητα κίνησης σε ένα χώρο με πολλά εμπόδια, που δεν είναι δυνατό να χαρτογραφηθεί εκ των προτέρων (ενδεχομένως και η αποστολή τους να είναι η χαρτογράφηση του χώρου αυτού).
3. Πρέπει να είναι εύκολα στη μεταφορά, άρα πρέπει να έχουν μικρές διαστάσεις και μικρό βάρος.
4. Πρέπει ο χειρισμός τους να είναι εύκολος και απλός, ώστε να μπορούν να χρησιμοποιηθούν σε συνθήκες μάχης από άτομα μη εξειδικευμένα στον τομέα της ρομποτικής.

5. Πρέπει να έχουν χαμηλό κόστος και να μην απαιτείται για την κατασκευή τους εξειδικευμένη τεχνολογία. Αυτό διότι λόγω της αποστολής τους και δεδομένου ότι θα χρησιμοποιούνται σε εχθρικά περιβάλλοντα, αναπόφευκτα θα έχουν μεγάλες απώλειες, οπότε θεωρούνται αναλώσιμα.
6. Ικανότητα μεταφοράς ενός περιορισμένου φορτίου (ώστε να μπορούν να εκτελέσουν πληθώρα αποστολών, ανάλογα με το φορτίο, όπως αναγνώριση, φύλαξη χώρου, ανίχνευση πολεμικών ουσιών ή ραδιενέργειας σε ένα χώρο κ.α.).

## §2. Σχετική Έρευνα – αντίστοιχες εφαρμογές

Ο τομές των επίγειων ρομποτικών οχημάτων εσωτερικού χώρου παρουσιάζει τεράστια εξέλιξη. Χαρακτηριστικά αναφέρονται ορισμένα παραδείγματα μικρών σε μέγεθος πειραματικών και επιχειρησιακών ρομποτικών οχημάτων και οι περισσότερες διαδεδομένες τεχνικές αυτόνομης αποφυγής εμποδίων:

### 1. iRobot 110 FirstLook:



2.1

Το iRobot 110 FirstLook (εικ. 2.1) είναι ένα μικρό ρομποτικό όχημα με μήκος περίπου 25 εκατοστά, και βάρος 2,26 κιλά. Χρησιμοποιεί ερπύστριες για την κίνησή του. Έχει στιβαρή σχεδίαση με σκοπό να επιτρέπει στο χειριστή του να το πετάξει σε κάποιο σημείο (μπορεί να το πετάξει από σκάλες ή να το εκτοξεύσει μέσα σε ένα κτίριο από το παράθυρο). Αντέχει σε πτώσεις 4,8 μέτρων περίπου σε τσιμέντο. Οι αισθητήρες του επιτρέπουν τον εντοπισμό

και την αναγνώριση κινδύνων σε κλειστούς χώρους, ενώ μπορεί να λειτουργεί υπό όλες τις καιρικές συνθήκες και υπό συνθήκες χαμηλής παρατήρησης. Οι αισθητήρες του περιλαμβάνουν τέσσερις κάμερες, μία σε κάθε πλευρά του οχήματος, οι οποίες λειτουργούν και στο φάσμα των υπέρυθρων. Στο «κατάρτι» του οχήματος βρίσκεται μία πέμπτη κάμερα με πεδίο ορατότητας 270°. Διαθέτει φώτα. Διαθέτει επίσης μηχανισμό που του επιτρέπει να υπερπηδά εμπόδια ύψους 17 περίπου εκατοστών (7 in). Είναι επίσης αδιάβροχο σε βάθος 1 μέτρου. Είναι τηλεχειριζόμενο, διαθέτοντας τρεις εναλλακτικές μορφές τηλεχειρισμού. Οι κύριες αποστολές που μπορεί να αναλάβει είναι: ανίχνευση εκρηκτικών ουσιών, καταστροφή εκρηκτικών ουσιών (Explosive Ordnance Disposal – EOD), ανίχνευση ΧΒΡΠ ουσιών, έλεγχος οχημάτων σε σημεία ελέγχου, παρατήρηση και ανίχνευση δρομολογίων εντός κτιρίων. Τέλος, το όχημα μπορεί να εξοπλιστεί με διάφορους επιπλέον αισθητήρες και εξοπλισμό ώστε να μπορέσει να εκτελέσει επιπλέον αποστολές (μηχανικός βραχίονας, εξειδικευμένες κάμερες και αισθητήρες κα) (23).

## 2. MLT-F “Jack Russel” – Compact Surveillamce Robot:

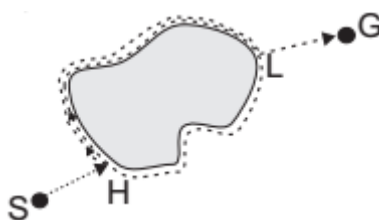


2.2

Το ρομποτικό αυτό όχημα (εικ. 2.2) είναι ένα πολύ μικρό όχημα με μήκος περίπου 11 εκατοστά και βάρος μικρότερο από 4 κιλά. Έχει σχεδιαστεί ώστε να μεταφέρεται εύκολα. Διαθέτει κάμερα στο εμπρόσθιο μέρος και μικρόφωνο. Έχει στιβαρή σχεδίαση με σκοπό να μπορεί να εκτοξεύεται μέσα σε κτίριο από παράθυρο (ή σε δυσπρόσιτες περιοχές) και αντέχει σε πτώσεις 3 περίπου μέτρων. Διαθέτει ερπύστριες και σύστημα υπέρβασης εμποδίων. Είναι τηλεχειριζόμενο, με δυνατότητα εναλλαγής μεθόδου τηλεχειρισμού. Χρησιμοποιείται κυρίως για αναγνώριση και επιτήρηση εσωτερικού χώρου στα πλαίσια επιχειρήσεων εντός κτιρίου (24).

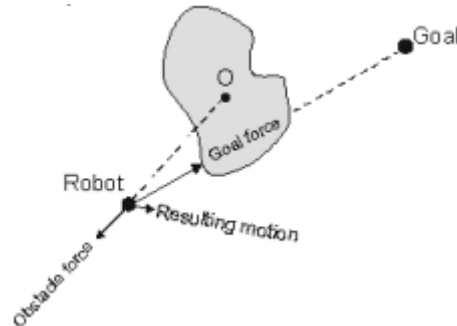
3. Όσον αφορά στις μεθόδους αποφυγής εμποδίων από ρομποτικά οχήματα σε πραγματικό χρόνο, χρησιμοποιώντας χαμηλού κόστους υπέρυθρους αισθητήρες ή αισθητήρες υπερήχων, αυτές μπορούν συνοπτικά να περιγραφούν ως εξής (2):

- Ο απλούστερος αλγόριθμος ονομάζεται «αλγόριθμος του εντόμου» (the bug algorithm): όταν το όχημα συναντά ένα εμπόδιο, απλά κινείται κυκλικά γύρω από το εμπόδιο έως ότου βρει το σημείο που απέχει λιγότερο από το σκοπό του και μετά κινείται προς το στόχο.



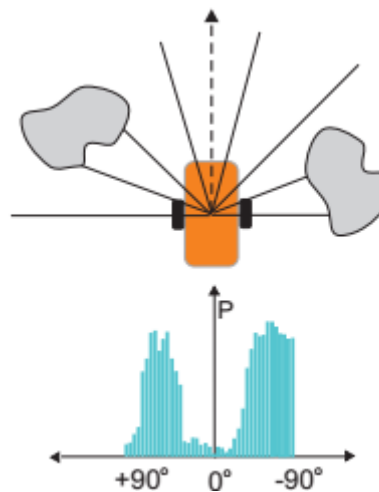
- Αλγόριθμος Δυναμικού Πεδίου (Potential Field Algorithm): ο αλγόριθμος αυτός υποθέτει ότι το όχημα κατευθύνεται από θεωρητικές

δυνάμεις που το ελκύουν ή το απωθούν από το στόχο του. Η διαδρομή του είναι αποτέλεσμα της συνισταμένης των δυνάμεων αυτών.



Αυτός ο αλγόριθμος εντάσσεται στο γενικότερο πλαίσιο του σχεδιασμού διαδρομής (path planning) και αποτελεί υπο-εργασία του.

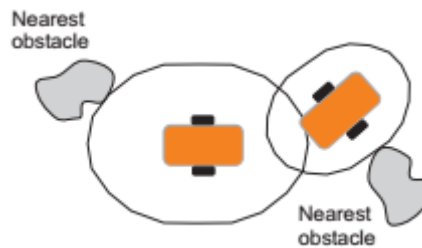
- Αλγόριθμος Ιστογράμματος Πεδίου Διανυσμάτων (Vector Field Histogram – VFH): αρχικά δημιουργούνται ιστογράμματα για αριθμό μετρήσεων αισθητήρων, όπως στο ακόλουθο σχήμα.



Ο άξονας x αντιπροσωπεύει τις γωνίες των μετρήσεων των αισθητήρων, ενώ ο άξονας y αντιπροσωπεύει την πιθανότητα να υπάρχει εμπόδιο στην αντίστοιχη κατεύθυνση. Βάσει αυτού υπολογίζεται η ύπαρξη διαδρόμων που μπορεί να κινηθεί το όχημα και επιλέγεται ο βέλτιστος.



- Τεχνική «ζώνης φουσαλίδας» (bubble band technique): βάσει των χαρακτηριστικών του οχήματος και των μετρήσεων των αισθητήρων, ορίζεται μια «φουσαλίδα» που περιέχει το μέγιστο χώρο που μπορεί να κινηθεί το όχημα, προς οποιαδήποτε κατεύθυνση. Μια αλληλουχία (ζώνη) τέτοιων φουσαλίδων χρησιμοποιείται για την δημιουργία μιας διαδρομής ανάμεσα σε ένα αρχικό και ένα τελικό σημείο.



- Φυσικά, υπάρχουν πολλές ακόμα τεχνικές αποφυγής εμποδίων, οι οποίες δεν μπορούν να αναγραφούν όλες.

### §3. Στόχοι της παρούσας εργασίας

Ο στόχος της παρούσας εργασίας είναι να κατασκευαστεί μια πλατφόρμα που να ενσωματώνει τα χαρακτηριστικά των αυτόνομων ρομποτικών οχημάτων της προηγούμενης παραγράφου. Για κάθε ένα από τα χαρακτηριστικά, κατά απόλυτη αντιστοιχία με την παράγραφο 1:

1. Η αυτονομία επιτυγχάνεται με τη χρήση ηλεκτρικών και ηλεκτρονικών εξαρτημάτων που δεν καταναλώνουν πολλή ενέργεια. Ακόμα και με τη χρήση απλών μπαταριών του εμπορίου το όχημα διαθέτει επαρκή ενέργεια για παρατεταμένη λειτουργία.
2. Το όχημα ενσωματώνει κάποιες απλές λειτουργίες αποφυγής εμποδίων, χρησιμοποιώντας υπέρυθρους αισθητήρες, αισθητήρα υπερήχων και απτικούς αισθητήρες (μια απλοποιημένη μέθοδος σε σχέση με τις τεχνικές που παρουσιάστηκαν στην παράγραφο 2).
3. Οι διαστάσεις και το βάρος του οχήματος είναι όσο ενός μεσαίου προς μικρού τηλεκατευθυνόμενου παιχνιδιού, με αντίστοιχο βάρος.
4. Ο χειρισμός του οχήματος βασίζεται στο πρωτόκολλο Bluetooth, ενώ δεν απαιτεί εξειδικευμένο τηλεχειριστήριο (αρκεί ένα smartphone ή tablet με λειτουργικό σύστημα android ή iOS). Οι λειτουργίες του είναι προσβάσιμες μέσω ενός απλού interface 9 πλήκτρων.
5. Το όχημα είναι κατασκευασμένο αποκλειστικά από υλικά εμπορίου (commercial off-the-shelf). Οι μπαταρίες που χρησιμοποιούνται είναι απλές μπαταρίες εμπορίου.
6. Δεδομένου ότι ένα φορτίο επιπλέον αισθητήρων ή μια απλή κάμερα έχουν πλέον μικρό βάρος, το όχημα μπορεί να μεταφέρει μια πληθώρα επιπλέον

φορτίων εάν χρειαστεί (π.χ. μια μικρή κάμερα, κάρτα SD για αποθήκευση δεδομένων, αισθητήρες ΧΒΡΠ κ.α.).

# ΚΕΦΑΛΑΙΟ 1

## Θεωρητικό Υπόβαθρο

### §1. Παρουσίαση σχεδίασης

Το όχημα σχεδιάστηκε με σκοπό να εκτελεί τρεις βασικές λειτουργίες, οι οποίες κρίνονται βασικές για την εκτέλεση οποιασδήποτε αποστολής μπορεί να ανατεθεί στο όχημα (στις ενδεικτικές αποστολές, αναγράφονται και πιθανά επιπλέον φορτία που θα ήταν απαραίτητα για την εκτέλεση των αποστολών αυτών):

#### 1. Απλή τηλεκατεύθυνση χωρίς καμία αυτόνομη λειτουργία.

##### Ενδεικτικές αποστολές:

- επιτήρηση χώρου (κάμερα)
- διερεύνηση συγκεκριμένης τοποθεσίας από τον χειριστή του οχήματος (κάμερα)
- μεταφορά φορτίου σε συγκεκριμένη τοποθεσία (κυτίο μεταφοράς)
- ναρκοθέτηση – καταστροφή στόχου (εκρηκτικό φορτίο)
- ανίχνευση και καταγραφή πολεμικών χημικών – βιολογικών ουσιών ή ραδιενέργειας (αντίστοιχοι αισθητήρες – σύστημα αποθήκευσης δεδομένων όπως κάρτα SD)
- Έλεγχος οχημάτων σε σημεία ελέγχου (με χρήση ευρυγώνιας κάμερας στην οροφή του οχήματος).

2. Αυτόνομη παρακολούθηση ενός στόχου, ο οποίος επισημαίνεται με ένα φάρο (ο στόχος θα μπορούσε να είναι είτε κινητός είτε ακίνητος), για ένα ορισμένο χρονικό διάστημα.

Ενδεικτικές αποστολές:

- αυτόματη επιστροφή σε κάποιο αρχικό σημείο,
- αυτόματη επιστροφή σε σημείο φόρτισης (επαναφορτιζόμενες μπαταρίες, σύστημα φόρτισης και σταθμός πρόσδεσης)
- παρακολούθηση ενός άλλου ρομποτικού οχήματος ώστε να είναι δυνατή η καθοδήγηση πολλαπλών οχημάτων από ένα χειριστή
- παρακολούθηση σημασμένου στόχου ή κατεύθυνση προς σημασμένο σημείο (σύστημα σήμανσης από τον χειριστή)

3. Εκτέλεση αυτόνομης περιπολίας μέσα σε ένα συγκεκριμένο εσωτερικό χώρο για ένα ορισμένο χρονικό διάστημα.

Ενδεικτικές αποστολές:

- επιτήρηση χώρου (κάμερα)
- λειτουργία φύλαξης (κάμερα, αισθητήρες κίνησης)
- απαγόρευση περιοχής (εκρηκτικό φορτίο, αισθητήρες κίνησης).

καταγραφή συνθηκών εντός αποθήκης (σύστημα αισθητήρων, σύστημα αποθήκευσης δεδομένων όπως κάρτα SD).

## §2. Τεχνολογίες υποστήριξης

### 1. Πομποί υπέρυθρων (IR Transmitter)

Οι πομποί υπέρυθρων (IR Transmitters) είναι διόδοι εκπομπής φωτός (light emitting diode – LED) που εκπέμπουν υπέρυθρη ακτινοβολία. Για χάρη συντομίας, αναγράφονται και ως IR LED. Η ακτινοβολία τους είναι αόρατη για το ανθρώπινο μάτι. Υπάρχουν διαφορετικές κατηγορίες πομπών υπέρυθρων, ανάλογα με το μήκος κύματος στο οποίο εκπέμπουν και την ισχύ εκπομπής τους (10). Στην 1.2.1 φαίνεται ένα τυπικό IR LED.



1.2.1

### 2. Δέκτες υπέρυθρων

Οι υπέρυθροι δέκτες καλούνται και υπέρυθροι αισθητήρες, αφού ανιχνεύουν την εκπεμπόμενη ακτινοβολία από έναν πομπό υπέρυθρων. Συναντώνται ως φωτοδιόδοι και φωτοτρανζίστορ. Οι υπέρυθρες φωτοδιόδοι διαφέρουν από τις κανονικές φωτοδιόδους διότι ανιχνεύουν μόνο υπέρυθρη ακτινοβολία. Όπως και με τους πομπούς υπέρυθρων, οι δέκτες υπέρυθρων διαφέρουν ανάλογα με το μήκος κύματος της ακτινοβολίας που μπορούν να ανιχνεύσουν και διάφορα τεχνικά χαρακτηριστικά λειτουργίας.

Όταν οι πομποί υπέρυθρων συνδυάζονται με τους δέκτες υπέρυθρης ακτινοβολίας, θα πρέπει το μήκος κύματος της ακτινοβολίας που εκπέμπει ο πομπός να είναι ίδιο με το μήκος κύματος που μπορεί να ανιχνεύσει ο δέκτης, ώστε να μπορεί ο πομπός να είναι ορατός στον δέκτη (10). Στην 1.2.2 φαίνεται μία τυπική φωτοδίοδος.



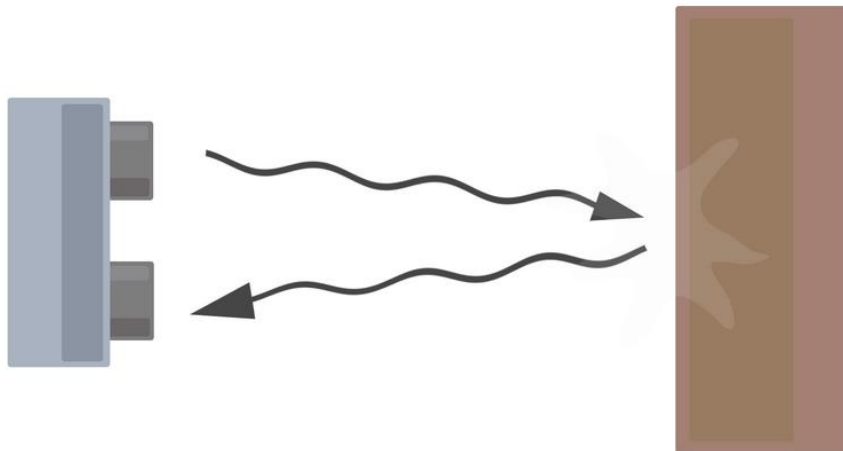
1.2.2

### 3. Αισθητήρες Μέτρησης Απόστασης Υπερήχων

Οι αισθητήρες μέτρησης απόστασης υπερήχων είναι αισθητήρες που μετρούν την απόσταση από ένα αντικείμενο χρησιμοποιώντας κύματα υπερήχων. Εκπέμπουν ένα ηχητικό κύμα με συγκεκριμένη συχνότητα και ανιχνεύουν την ανάκλαση του κύματος όταν αυτή επιστρέφει. Μετρώντας το χρόνο ανάμεσα στην εκπομπή και την ανίχνευση της ανάκλασης, υπολογίζεται η απόσταση ανάμεσα στον αισθητήρα και κάποιο αντικείμενο, χρησιμοποιώντας τον τύπο:

$$\text{απόσταση} = \text{ταχύτητα του ήχου} \times \text{χρόνος} \div 2$$

κατά τον τρόπο που απεικονίζεται στην 1.2.3 (27).



1.2.3

#### 4. Πρωτόκολλο Bluetooth

Το πρωτόκολλο Bluetooth είναι ένα τυποποιημένο πρωτόκολλο για αποστολή και λήψη δεδομένων μέσω μιας ασύρματης σύνδεσης στα 2.4 GHz. Χρησιμοποιείται για ασύρματη, μικρής εμβέλειας μεταφορά δεδομένων μέσω ηλεκτρονικών συσκευών, με μικρές ενεργειακές απαιτήσεις και με μικρό κόστος.

Η εμβέλεια του πρωτοκόλλου Bluetooth περιορίζεται σε αποστάσεις μικρότερων των 100 μέτρων και συστήνεται για μεταφορά σχετικά μικρού όγκου δεδομένων (11).



## ΚΕΦΑΛΑΙΟ 2

# Σχεδίαση - Κατασκευή

### §1. Σχεδίαση

Το όχημα σχεδιάστηκε ώστε:

- Να ανιχνεύει ένα φάρο υπέρυθρης ακτινοβολίας, να κατευθύνεται προς αυτόν και να σταματά μερικά εκατοστά μπροστά του. Αν υπάρχει ενδιάμεσα κάποιο ή κάποια εμπόδια, να τα αποφύγει, χρησιμοποιώντας μια απλή τεχνική αποφυγής εμποδίων. Εάν περάσει χρόνος ενός λεπτού και δεν έχει βρει το φάρο, σταματά.
- Να περιπολεί αυτόνομα σε ένα εσωτερικό χώρο για δύο λεπτά, αποφεύγοντας εμπόδια και καλύπτοντας το χώρο.
- Να είναι δυνατή η πλήρης τηλεκατεύθυνση του οχήματος με τη χρήση μιας απλής φορητής συσκευής (smartphone ή tablet) με λειτουργικό android ή iOS.

Η σχεδίαση έχει ως εξής:

1. **Σκελετός - κινητήρες:** το αυτόνομο ρομποτικό όχημα αποτελείται από ένα τυποποιημένο μεταλλικό σασί 4X4 με τέσσερις ηλεκτροκινητήρες, όπου ο κάθε ηλεκτροκινητήρας κινεί ένα τροχό. Ο ελεγκτής που χρησιμοποιείται για τη σύνδεση των κινητήρων με τον υπολογιστή είναι το Adafruit Motor Shield V2 for Arduino.

2. **Αισθητήρες:** το όχημα διαθέτει τους εξής αισθητήρες:

- Έναν αισθητήρα υπερήχων για την ανίχνευση εμποδίων εμπρός.
- Τέσσερις αισθητήρες υπέρυθρων τοποθετημένους σε μια υπερυψωμένη κατασκευή περιμετρικά (ιστίο), ώστε να ανιχνεύει την κατεύθυνση στην οποία βρίσκεται ο φάρος.
- Έναν υπέρυθρο αισθητήρα προσέγγισης για την ανίχνευση της απόστασης του φάρου στο εμπρόσθιο τμήμα.
- Δύο προφυλακτήρες (απτικοί αισθητήρες), ώστε να αναγνωρίζει και να αντιδρά σε προσκρούσεις στα πλάγια του οχήματος.

3. **Ελεγκτές:** ο υπολογιστής που χρησιμοποιείται για την επεξεργασία των δεδομένων των αισθητήρων, τον προγραμματισμό του οχήματος και τον τηλεχειρισμό του είναι ένας μικροελεγκτής Funduino Mega 2560, συμβατός με Arduino Mega 2560 r3. Ο ελεγκτής που χρησιμοποιείται για τη σύνδεση των κινητήρων με τον υπολογιστή είναι το Adafruit Motor Shield V2 for Arduino.

4. **Τηλεχειρισμός:** το σύστημα τηλεχειρισμού που έχει επιλεγεί είναι το Adafruit Bluefruit LE UART, το οποίο μεταφέρει στον υπολογιστή εντολές από συσκευή με λειτουργικό android ή iOS.

5. **Τροφοδοσία:** οι κινητήρες τροφοδοτούνται από 5 μπαταρίες AA (7,5V). Ο υπολογιστής και οι αισθητήρες τροφοδοτούνται από μπαταρία 9V.

6. **Φάρος:** Ο φάρος διαθέτει 2 τύπους πομπών υπέρυθρης ακτινοβολίας:

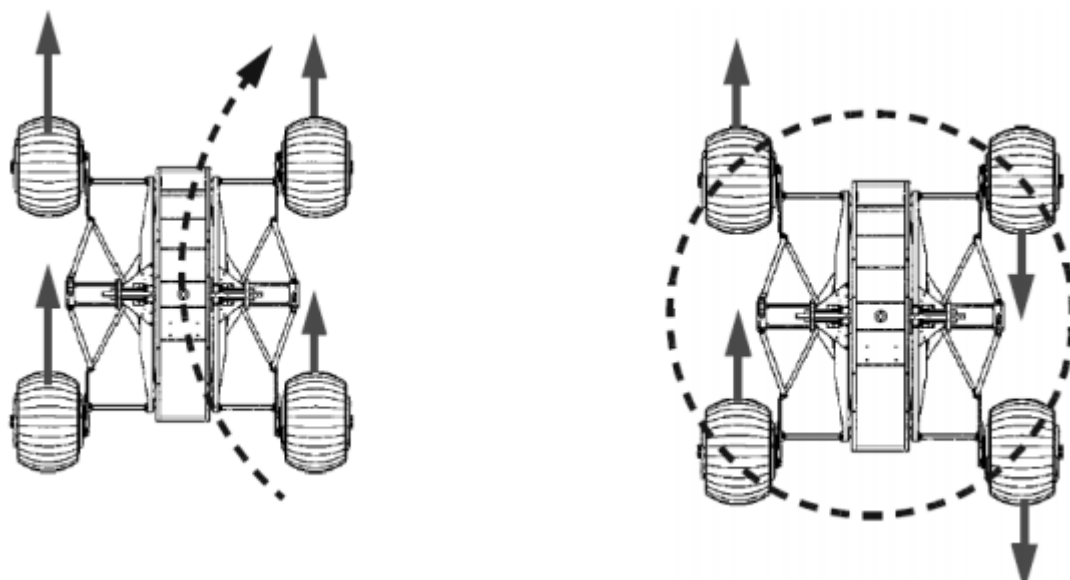
- Πολλαπλούς πομπούς υπέρυθρων υψηλής έντασης, οι οποίοι ανιχνεύονται από το όχημα και προσδιορίζουν τη θέση του φάρου σε σχέση με αυτό.

- Πολλαπλούς πομπούς υπέρυθρων χαμηλής έντασης, οι οποίοι ανιχνεύονται από το όχημα όταν προσεγγίσει σε ικανοποιητική απόσταση τον φάρο (μερικά εκατοστά).

Ο φάρος τροφοδοτείται από τέσσερις μπαταρίες ΑΑ.

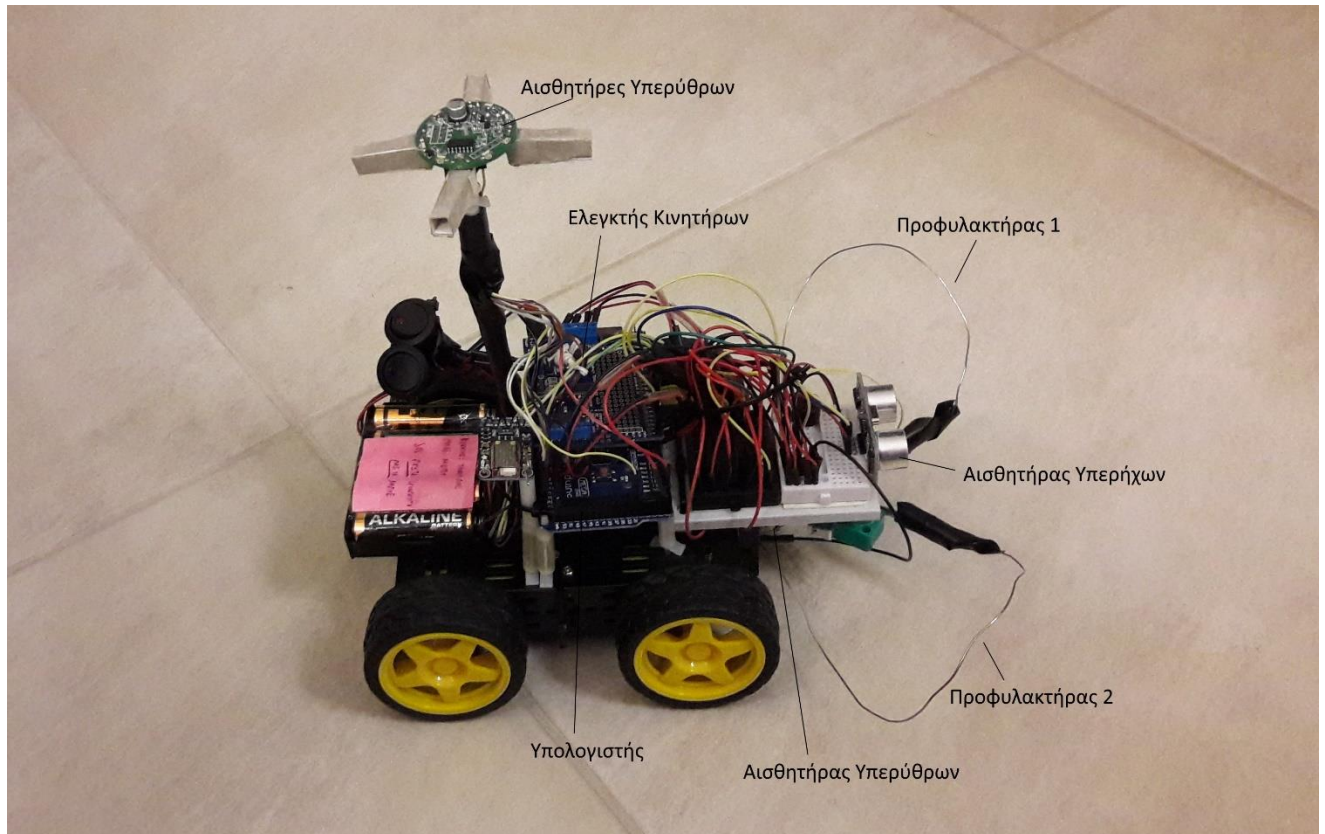
Λεπτομερής λίστα των εξαρτημάτων που χρησιμοποιούνται, τεχνικά χαρακτηριστικά, διαγράμματα λογικής και ο πλήρης κώδικας παρατίθενται στα παραρτήματα.

Ο τύπος της κίνησης που χρησιμοποιείται είναι η οδήγηση με ολίσθηση τροχών (1), όπως φαίνεται σχηματικά στην εικόνα 2.1.1

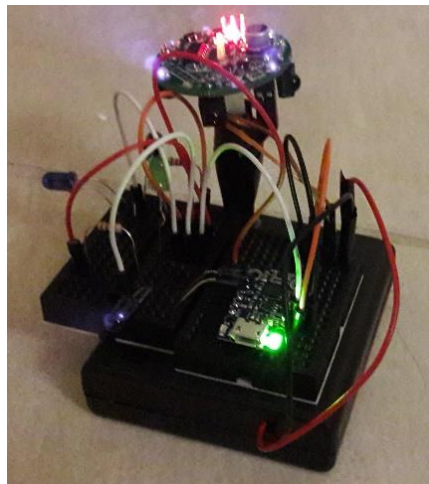


2.1.1

Στις φωτογραφίες 2.1.2 και 2.1.3 απεικονίζονται το όχημα και ο φάρος ολοκληρωμένα.



### 2.1.2

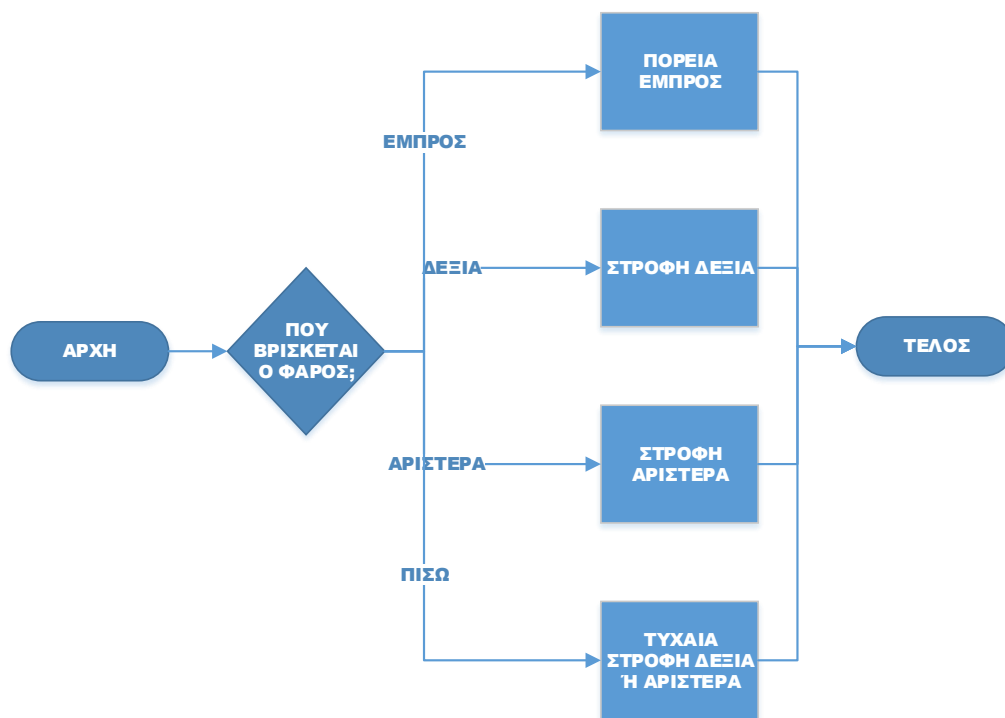


### 2.1.3

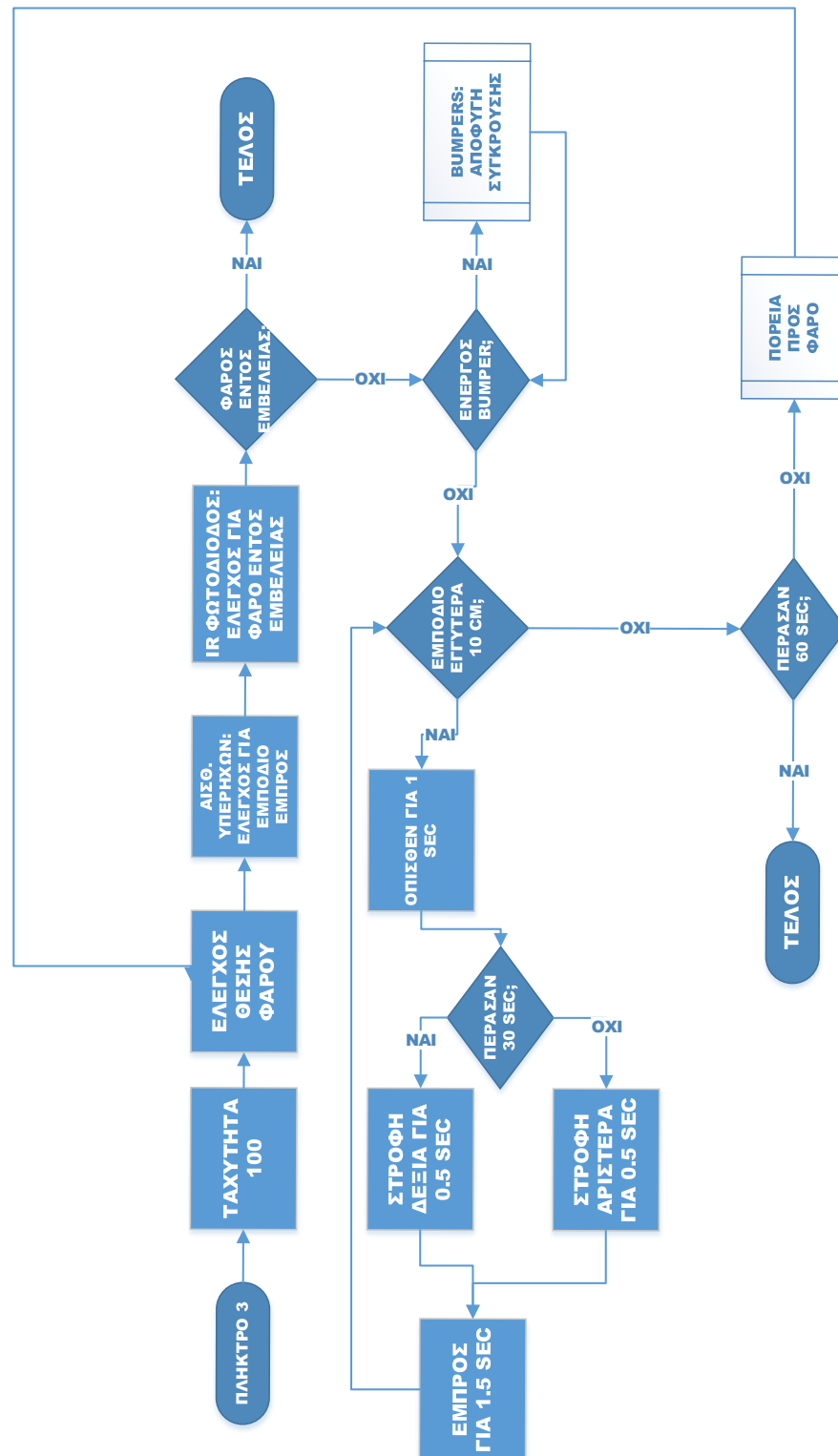
## §2. Επεξήγηση μεθόδου αποφυγής εμποδίων

Η λογική που διέπει τη μέθοδο αποφυγής εμποδίων με τον αισθητήρα υπερήχων καθώς αναζητά το φάρο αναλύεται στα διαγράμματα 2.2.1 και 2.2.2 και περιγράφεται ως εξής:

1. Αρχικά το όχημα στρέφει προς το φάρο.
2. Προχωρά προς το φάρο.
3. Εάν ο αισθητήρας υπερήχων εντοπίσει εμπόδιο σε απόσταση μικρότερη των 10 εκατοστών, αναστρέφει πορεία για χρονικό διάστημα 1 δευτερολέπτου.
4. Στρέφει αριστερά ή δεξιά (ανάλογα με το χρονικό διάστημα που έχει παρέλθει από την ενεργοποίηση της εντολής) για 0,5 δευτερόλεπτα.
5. Κινείται εμπρός για 1,5 δευτερόλεπτα, με αποτέλεσμα να βρεθεί δίπλα στο εμπόδιο.
6. Κινείται πάλι προς το φάρο (και η διαδικασία είναι δυνατόν να επαναληφθεί).



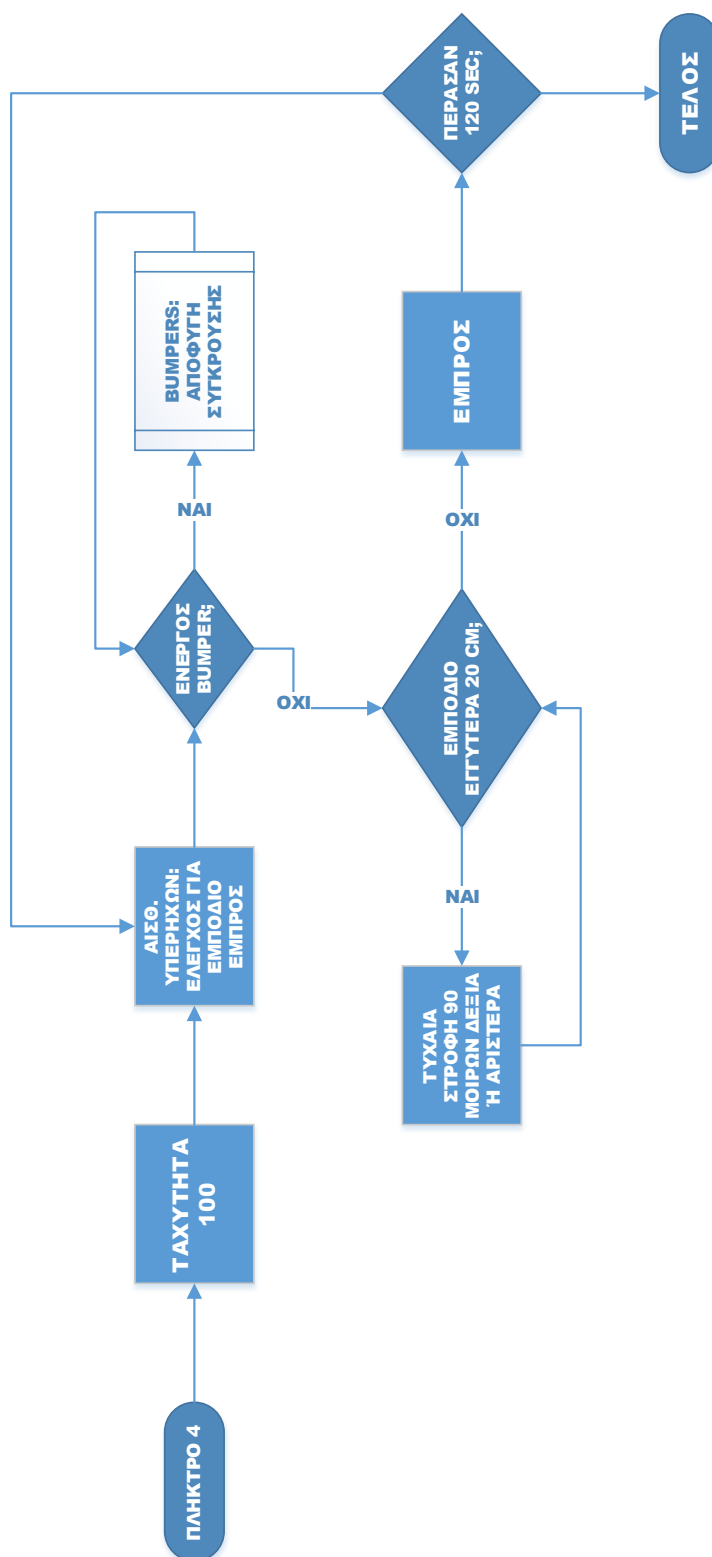
2.2.1



### 2.2.2

Η λογική που διέπει τη λειτουργία περιπολίας και την αποφυγή εμποδίων κατά τη διάρκεια της αναλύεται στο διάγραμμα 2.2.3 και αναλύεται ως εξής:

1. Αρχικά το όχημα κινείται εμπρός.
2. Εάν ο αισθητήρας υπερήχων εντοπίσει εμπόδιο σε απόσταση μικρότερη των 20 εκατοστών, αναστρέφει πορεία για χρονικό διάστημα 1 δευτερολέπτου.
3. Στρέφει αριστερά ή δεξιά τυχαία (βασιζόμενο σε γεννήτρια τυχαίων αριθμών) για 0,5 δευτερόλεπτα.
4. Κινείται εμπρός και η διαδικασία επαναλαμβάνεται.
5. Η λειτουργία παύει με την πάροδο 120 sec. Αυτό διότι έχει υπολογιστεί ότι με την ταχύτητα κίνησης του οχήματος και με μια λογική πυκνότητα εμποδίων (όπως αναλύεται παρακάτω) το όχημα μπορεί μέσα στο χρόνο αυτό να καλύψει πλήρως χώρο μεγέθους μιας μεγάλης αίθουσας.

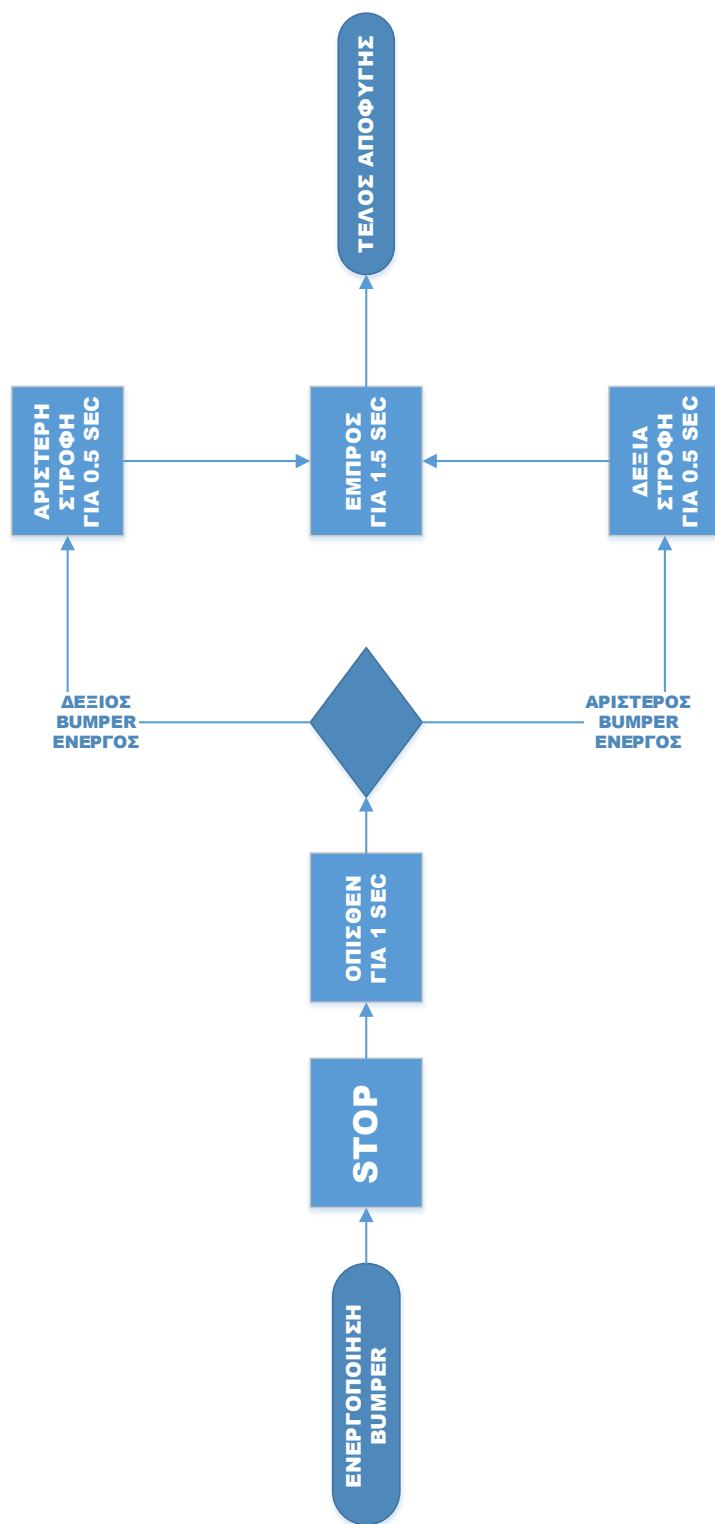


### 2.2.3



Και οι δύο λειτουργίες μπορεί να διακοπούν και να ενεργοποιηθεί μέσω των προφυλακτήρων η αποφυγή εμποδίου από την αριστερή ή τη δεξιά πλευρά, ανάλογα με τον προφυλακτήρα που ενεργοποιήθηκε. Η λογική που διέπει την αποφυγή του δεξιού προφυλακτήρα (και αντίστοιχα του αριστερού) αναλύεται στο διάγραμμα 2.2.4 και περιγράφεται ως εξής:

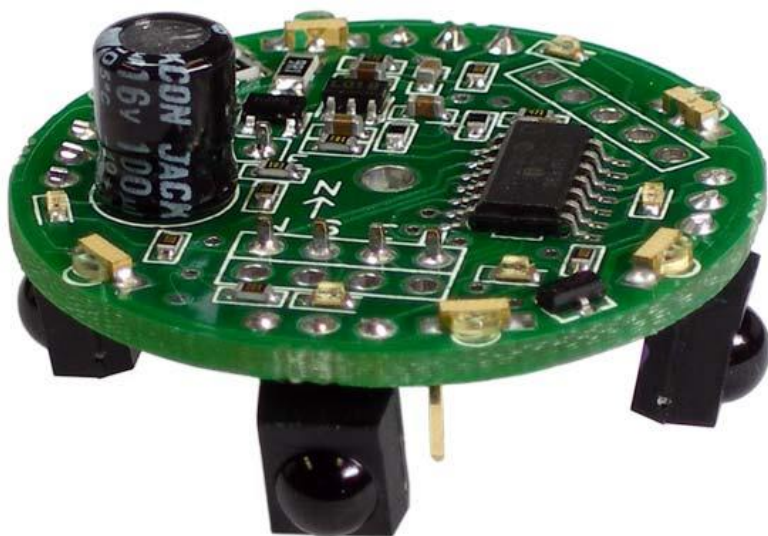
1. Καθώς το όχημα κινείται, ενεργοποιείται ο δεξιός προφυλακτήρας.
2. Διακόπτεται η κίνηση.
3. Οπισθοπορεία για 1 δευτερόλεπτο.
4. Στροφή αριστερά (δεξιά για τον αριστερό προφυλακτήρα) για 0,5 δευτερόλεπτα.
5. Εμπροσθοπορεία για 1,5 δευτερόλεπτα, με αποτέλεσμα να βρεθεί δίπλα στο εμπόδιο.
6. Συνεχίζεται η κίνηση κανονικά.



2.2.4

### §3. Επεξήγηση μεθόδου επιστροφής στο φάρο υπέρυθρων

Η μέθοδος αυτή βασίζεται στη χρήση του ζεύγους φάρων Pololu IR Beacon Transceiver.



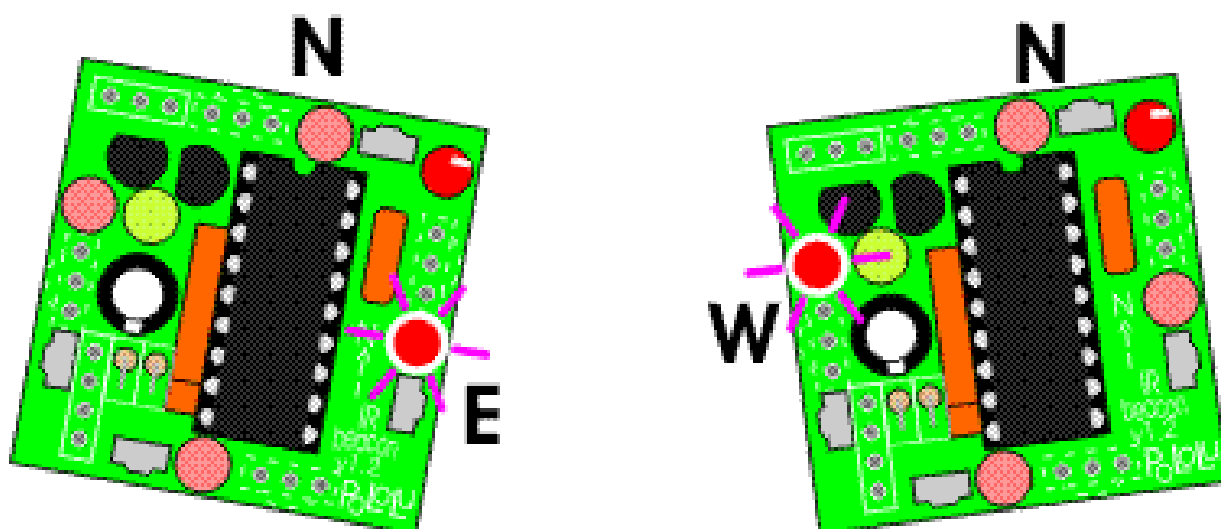
2.3.1

Στις εικόνες 2.3.1 και 2.3.2 εικονίζεται μία συσκευή Pololu IR Beacon Transceiver. Περιμετρικά της κυκλικής πλακέτας έχουν τοποθετηθεί έξι υπέρυθρα LED και τέσσερις αισθητήρες υπέρυθρων (IR). Οι συσκευές χρησιμοποιούνται σε ζεύγη. Οι αισθητήρες IR της μίας ανιχνεύουν τα LED της άλλης. Εξάγουν μια

αναλογική τιμή για κάθε αισθητήρα IR, ώστε να είναι δυνατός ο προσδιορισμός για κάθε συσκευή της πλευράς με την οποία «βλέπει» την άλλη. Σε κάθε συσκευή οι αισθητήρες IR ονομάζονται «Βόρειος», «Νότιος», «Ανατολικός» και «Δυτικός». Η κάθε συσκευή διαθέτει 4 αναλογικές εξόδους, μία για κάθε αισθητήρα IR.



2.3.2



2.3.3

Στην εικόνα 2.3.3, η πρώτη συσκευή αντιλαμβάνεται ότι ο «Ανατολικός» αισθητήρας IR της (E) έχει οπτική επαφή με τα LED της δεύτερης συσκευής. Η δεύτερη συσκευή αντιλαμβάνεται ότι ο «Δυτικός» αισθητήρας IR της (W) έχει οπτική επαφή με τα LED της πρώτης.

Στο ιστίο του οχήματος βρίσκεται τοποθετημένη η μία από τις δύο συσκευές, ενώ στο μηχανισμό του φάρου βρίσκεται η δεύτερη. Το όχημα, γνωρίζοντας σε ποια πλευρά του βρίσκεται η δεύτερη συσκευή, στρέφει προς αυτή και κινείται προς το φάρο.

Σε περίπτωση που κατά τη διάρκεια της κίνησης ανιχνευθεί εμπόδιο, ακολουθείται η διαδικασία αποφυγής εμποδίων που έχει περιγραφεί στην προηγούμενη παράγραφο (26).

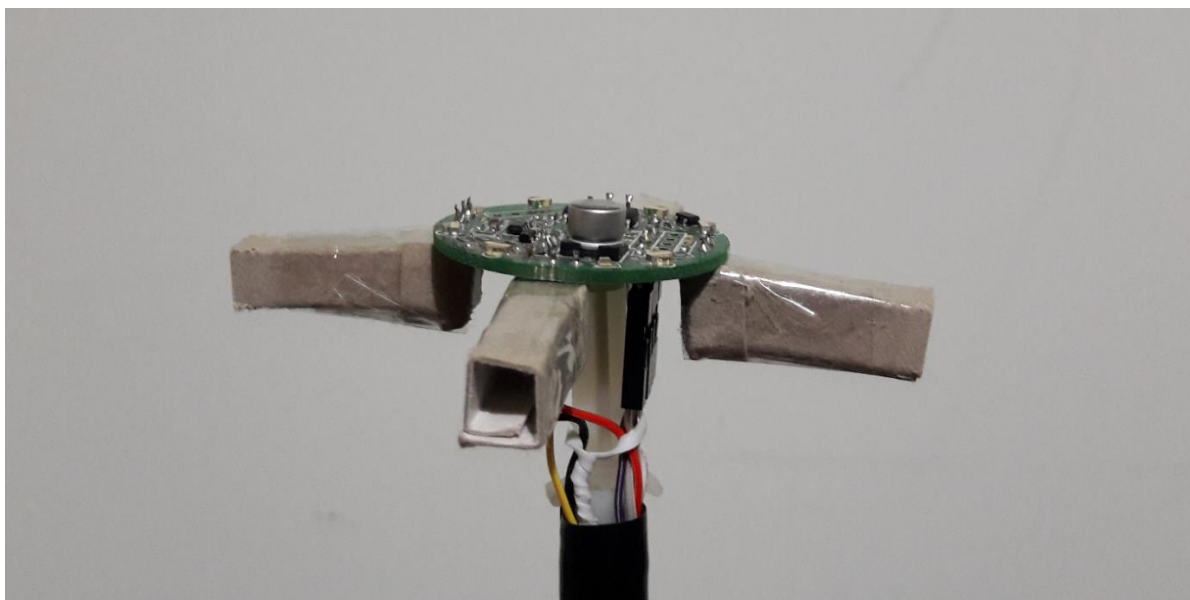
## ΚΕΦΑΛΑΙΟ 3

# Πειράματα και αποτελέσματα

### §1. Μέτρηση ακρίβειας ανίχνευσης φάρου

Κατά τη διάρκεια πολλαπλών μετρήσεων, έγινε αντιληπτό ότι μειωνόταν η ακρίβεια εντοπισμού του φάρου, ειδικά σε μικρές αποστάσεις. Ο αισθητήρας που βρισκόταν πάνω στο ιστίο του οχήματος εντόπιζε συχνά σε λανθασμένη κατεύθυνση τον αισθητήρα του φάρου.

Για τη διόρθωση του προβλήματος αυτού προστέθηκαν στον αισθητήρα του ιστίου του οχήματος μικρές χάρτινες κατασκευές (χοάνες) που περιορίσαν το οπτικό πεδίο των αισθητήρων υπέρυθρων, όπως εικονίζεται στη φωτογραφία 3.1.1:



3.1.1

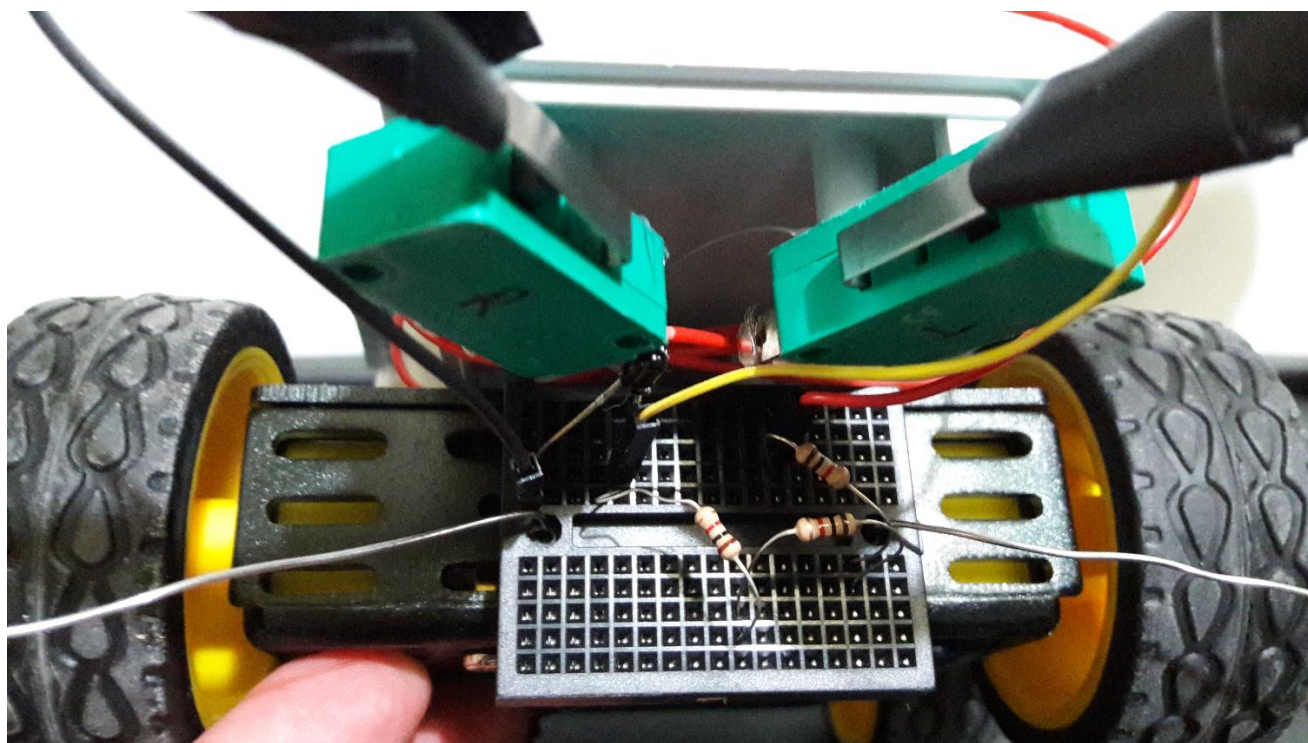
Ως αποτέλεσμα, βελτιώθηκε η ακρίβεια εντοπισμού της κατεύθυνσης του φάρου ακόμα και σε αποστάσεις έως 10 εκατοστών.

Τέλος, διαπιστώθηκε ότι οποιαδήποτε απόσταση και αν υπάρχει μεταξύ του ζεύγους των αισθητήρων υπέρυθρων (φάρος και ιστίο), οι αναλογικές τιμές που εξάγονται είναι ίδιες (οι αναλογικές τιμές που εξάγονται δεν αντιστοιχούν στην ισχύ με την οποία δέχεται υπέρυθρη ακτινοβολία ο αισθητήρας). Αυτό είχε σαν αποτέλεσμα να μην μπορεί να χρησιμοποιηθεί το ζεύγος rololu για τον προσδιορισμό της απόστασης μεταξύ του οχήματος και του φάρου.

## §2. Μέτρηση μέγιστης απόστασης προσέγγισης στο φάρο

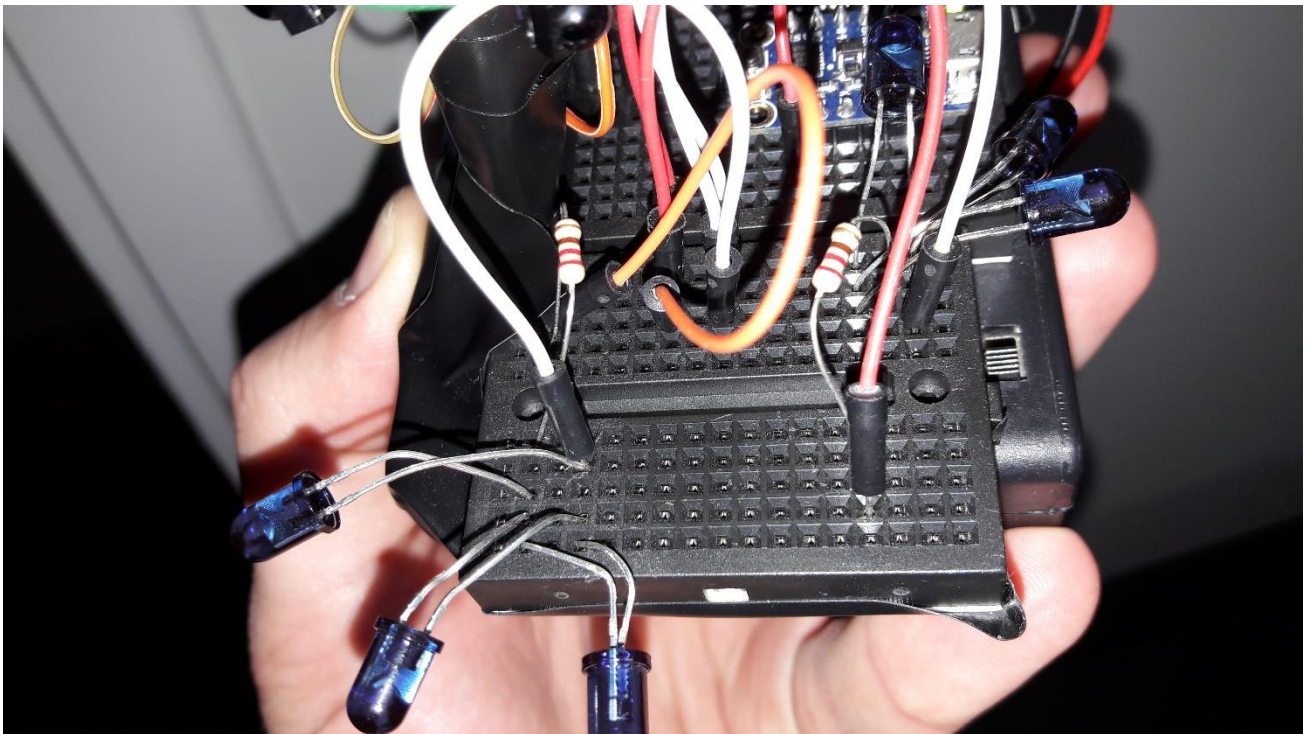
Η αυτόνομη λειτουργία της προσέγγισης στο φάρο, σταματά όταν το όχημα βρεθεί σε απόσταση 10 εκατοστών από αυτόν. Αυτό επιτυγχάνεται με τον περιορισμό της έντασης των μικρών λυχνιών LED υπέρυθρων. Οι διατάξεις φαίνονται στις φωτογραφίες 3.1.2, 3.1.3 και 3.1.4.

Οι διακόπτες ενεργοποιούν έκατος ένα interrupt στον κώδικα, οπότε το όχημα ξεκινά αυτομάτως τη διαδικασία αποφυγής εμποδίου. Οι διακόπτες είναι απλοί μηχανικοί διακόπτες ON/OFF.

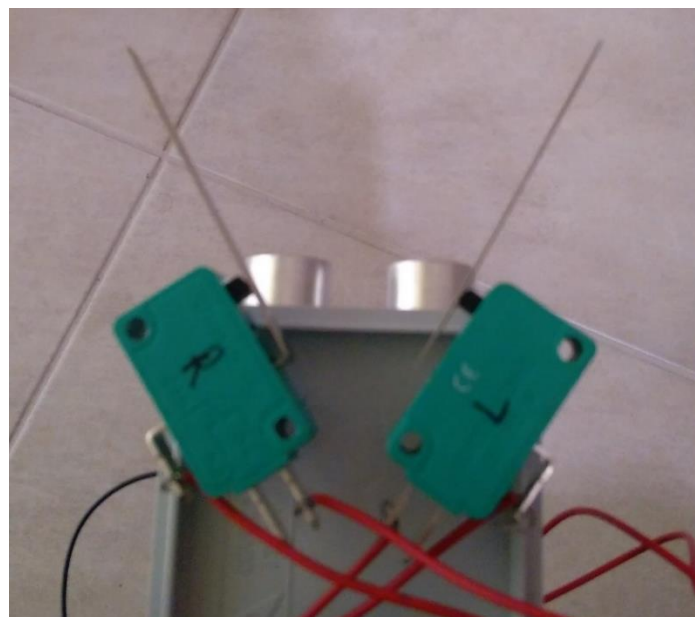


3.1.2





3.1.3



3.1.4



Οι αντιστάσεις που χρησιμοποιούνται είναι  $220\Omega$ . Στο όχημα βρίσκονται 3 αντιστάσεις συνδεδεμένες σε σειρά (συνολικά  $660\Omega$ ), ενώ στο φάρο η κάθε λυχνία περιορίζεται από 1 αντίσταση.

Η απόσταση ανίχνευσης του φάρου, εάν μεταβληθεί η αντίσταση της διάταξης του φάρου, φαίνεται στον παρακάτω πίνακα 3.1.5:

ΑΝΤΙΣΤΑΣΗ LED	ΑΠΟΣΤΑΣΗ ΑΝΙΧΝΕΥΣΗΣ
$220\Omega$	10 cm
$440\Omega$	02 cm
$660\Omega$	Απαιτείται επαφή οχήματος - φάρου

*3.1.5*

### §3. Παρεμβολές από το ορατό φως – παρεμβολές από ισχυρές πηγές υπέρυθρων

Κατά τη διάρκεια των δοκιμών και κατόπιν μετρήσεων σε διάφορες συνθήκες φωτισμού και δεν παρατηρήθηκε καμία διαφορά ανάλογα με τις συνθήκες αυτές. Δοκιμάστηκαν οι εξής συνθήκες:

- Σκοτεινό δωμάτιο
- Δωμάτιο με φωτισμό από φωτιστικά LED
- Δωμάτιο με φωτισμό με λαμπτήρες πυρακτώσεως
- Δωμάτιο που φωτίζεται με το φως του ηλίου
- Ανοικτός χώρος κατά τη διάρκεια της ημέρας

Έγινε προσπάθεια παρεμβολών από ισχυρή σχετικά πηγή υπέρυθρων. Για το σκοπό αυτό χρησιμοποιήθηκε τηλεχειριστήριο τηλεόρασης. Οι παρεμβολές ήταν επιτυχημένες, και κατόρθωσαν να δημιουργήσουν σύγχυση στον αισθητήρα. Παρ' όλα αυτά, εξ' αιτίας των χάρτινων κατασκευών στους αισθητήρες του ιστίου χρειαζόταν οπτική επαφή μεταξύ ενός αισθητήρα και του τηλεχειριστηρίου μέσα από τη χοάνη ώστε να υπερिσχύσει το σήμα της παρεμβολής.

Τέλος έγινε προσπάθεια παρεμβολής των αισθητήρων υπέρυθρων χρησιμοποιώντας φως λέιζερ και ισχυρό φως LED απ' ευθείας στους αισθητήρες. Και οι δύο προσπάθειες δεν κατέληξαν σε παρεμβολή κανενός αισθητήρα.

## §4. Προσδιορισμός περιορισμών κίνησης

Δοκιμάστηκε η δυνατότητα κίνησης του οχήματος σε διαφορετικά εδάφη και διαπιστώθηκαν ορισμένοι περιορισμοί στην κίνησή του, κυρίως λόγω των διαθέσιμων τροχών. Τα αποτελέσματα παρουσιάζονται στον πίνακα 3.1.6:

ΕΔΑΦΟΣ	ΑΠΟΤΕΛΕΣΜΑ
Πλακάκι – λείο έδαφος (τυπικό δάπεδο εσωτερικού οικίας)	Κανένας περιορισμός
Τσιμέντο (όχι απόλυτα λεία στρωμένο)	Δυσκολία στην κίνηση λόγω κακής πρόσφυσης
Χαλί λεπτό (τυπικό χαλί διαδρόμου)	Δυσκολία στην κίνηση λόγω τριβής
Χαλί χοντρό (τυπικό χαλί σαλονιού)	Μεγάλη δυσκολία στην κίνηση λόγω τριβής
Φλοκάτη	Αδυναμία κίνησης
Πλάκα πεζοδρομίου – πλάκα μη λεία	Δυσκολία στην κίνηση άλλοτε λόγω κακής πρόσφυσης και άλλοτε λόγω τριβής

### 3.1.6

Δεν πραγματοποιήθηκε δοκιμή του οχήματος σε υγρό έδαφος, σε λάσπη και χώμα λόγω της ευαισθησίας των ηλεκτρονικών τμημάτων του.

Όσον αφορά στις αυτόνομες λειτουργίες του οχήματος, δοκιμάστηκαν διάφορες ταχύτητες. Οι τελική ταχύτητα που επιλέχθηκε ήταν η ταχύτητα των 100 rpm. Σε μικρότερες ταχύτητες από αυτή, η το όχημα δυσκολευόταν να κινηθεί ακόμα και σε λείο έδαφος. Σε μεγαλύτερες ταχύτητες από τα 100 rpm, το όχημα αντιδρούσε καθυστερημένα στις εντολές που λάμβανε από τον αισθητήρα υπερήχων και επίσης οι απτικές επαφές με εμπόδια δημιουργούσαν ζημιές στους προφυλακτήρες.

## §5. Περιορισμοί στην πυκνότητα και το μέγεθος των εμποδίων

### 1. Λειτουργία αυτόνομης προσέγγισης φάρου

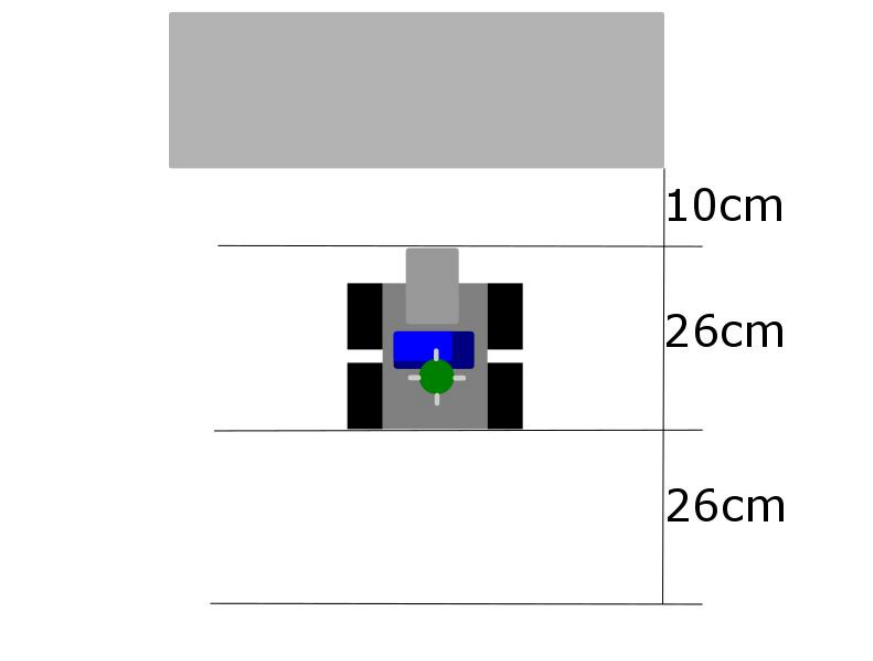
Κατά την αποφυγή εμποδίων, το όχημα εκτελεί αρχικά οπισθοπορεία για 1 sec. Με την ορισμένη ταχύτητα 100 rpm σε ιδανικό έδαφος, αυτό φέρνει το όχημα πίσω κατά 1 μήκος οχήματος (26 cm). Λαμβάνοντας υπ' όψη και το γεγονός ότι η μέγιστη απόσταση ανίχνευσης ενός εμποδίου από τον υπέρυθρο αισθητήρα είναι 10cm, τότε καταλήγουμε στο συμπέρασμα ότι για μια τυπική αποφυγή εμποδίου το όχημα απαιτεί ιδανικά απόσταση μεταξύ εμποδίων  $26+26+10$  cm, δηλαδή 62cm.

### 2. Λειτουργία αυτόνομης περιπολίας

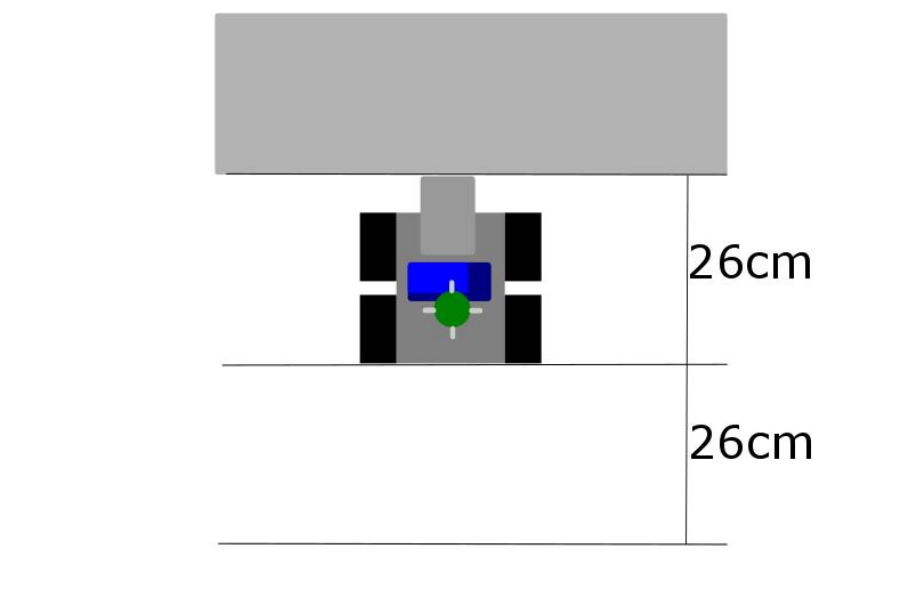
Στη λειτουργία αυτόνομης περιπολίας η μέγιστη απόσταση ανίχνευσης εμποδίου από τον υπέρυθρο αισθητήρα είναι 20cm. Το όχημα τότε εκτελεί επιτόπια στροφή, άρα δεν χρειάζεται μεγαλύτερο χώρο από 20 cm.

Όμως, οι προφυλακτήρες όταν έρθουν σε επαφή με εμπόδιο, δίνουν εντολή για εκτέλεση οπισθοπορείας για 1 sec, δηλαδή ένα μήκος οχήματος. Άρα το όχημα απαιτεί ιδανικά απόσταση μεταξύ εμποδίων  $26+26$  cm, δηλαδή 52 cm.

Τα παραπάνω φαίνονται στα σχήματα 3.1.7 (περιορισμοί σε λειτουργία προσέγγισης φάρου) και 3.1.8 (περιορισμοί σε λειτουργία αυτόνομης περιπολίας).



3.1.7



3.1.8

Τέλος, πρέπει να επισημανθεί ότι λόγω της φύσης των αισθητήρων, το όχημα δεν μπορεί να διακρίνει λεπτά κατακόρυφα εμπόδια (π.χ. λεπτά πόδια επίπλων, συρματοσχοίνα ή σύρματα κ.τ.λ.).

## ΚΕΦΑΛΑΙΟ 4

# Αξιολόγηση των αποτελεσμάτων

### §1. Περιορισμοί

#### 1. Σιάλες – κατακόρυφα εμπόδια

Το όχημα αδυνατεί λόγω κατασκευής να υπερβεί κατακόρυφα εμπόδια, έστω και μικρού ύψους. Οι σιάλες επίσης αποτελούν εμπόδιο, είτε πρόκειται για ανωφέρεια είτε πρόκειται για κατωφέρεια.

Για την επίλυση του προβλήματος αυτού εξετάζονται δύο λύσεις:

- Προσαρμογή αισθητήρα υπερήχων ή υπέρυθρων στο εμπρόσθιο τμήμα του οχήματος, με κλίση προς το έδαφος, ώστε να μπορεί να μετρηθεί η υψομετρική διαφορά με το επίπεδο κίνησης του οχήματος. Με αυτό τον τρόπο θα είναι δυνατή η ανίχνευση σιάλας μπροστά από το όχημα.
- Χρήση μηχανισμού υπερπήδησης εμποδίων (flipper) ή χρήση μηχανισμού κατακόρυφων ερπυστριών στο εμπρόσθιο τμήμα του οχήματος. Αυτός ο μηχανισμός ενδεχομένως να μπορέσει να χρησιμοποιηθεί και σε περίπτωση ανατροπής του οχήματος.

#### 2. Εμπόδια: για να είναι διακριτά πρέπει να είναι συγκεκριμένων διαστάσεων

Ο αισθητήρας υπερήχων εμφανίζει δυσκολία στη διάκριση λεπτών εμποδίων (λεπτά πόδια επίπλων, λεπτών κατακόρυφων σωληνώσεων κ.τ.λ.). Οι πιθανές λύσεις που εξετάζονται είναι οι εξής:

- Χρήση αισθητήρων μεγαλύτερης διακριτικότητας

- Χρήση πολλαπλών αισθητήρων και συνδυασμός των μετρήσεών τους (συνδυασμός πολλαπλών αισθητήρων υπερήχων ή συνδυασμός αισθητήρων υπερήχων με υπέρυθρους αισθητήρες)
- Χρήση αισθητήρα υπέρυθρων προσαρμοσμένο σε σερβοκινητήρα, ώστε να σαρώνει το χώρο μπροστά από το όχημα και να διακόπτει την κίνηση όταν διακρίνει εμπόδιο.

### 3. Δυσκολία κίνησης σε συγκεκριμένες επιφάνειες, όπως χαλιά, τραχιές επιφάνειες, κ.α.

Η λύση που προτείνεται στην περίπτωση αυτή είναι η χρήση ερπυστριών αντί των τροχών, προσδίδοντας έτσι στο όχημα καλύτερη πέδηση και κύλιση.

### 4. Το όχημα δεν έχει στρατιωτικές προδιαγραφές

Σε σχέση με τα οχήματα που περιεγράφηκαν στην παράγραφο 2 της εισαγωγής, καθίσταται προφανές ότι το εξεταζόμενο όχημα δεν είναι τελικό προϊόν ούτε είναι επιχειρησιακό.

Τα οχήματα που εξετάζονται στην εισαγωγή έχουν στρατιωτικές προδιαγραφές και αποτελούν επιχειρησιακά μοντέλα. Το πιο αξιοσημείωτο χαρακτηριστικό τους είναι η αυξημένη αντοχή, που τους επιτρέπει ακόμα και να εκτοξευτούν σε ένα σημείο. Αυτό προϋποθέτει στιβαρή κατασκευή, τόσο εξωτερικά όσο και εσωτερικά, χρήση εξαρτημάτων συγκεκριμένων προδιαγραφών, όπως και δυνατότητα επαναφοράς σε κανονική θέση μετά από ανατροπή. Επίσης, θεωρείται αυτονόητο ότι ένα όχημα στρατιωτικών προδιαγραφών θα πρέπει να είναι ανθεκτικό σε δύσκολες περιβαλλοντικές συνθήκες (βροχή, υγρασία, σκόνη, λάσπη). Οι αισθητήρες των οχημάτων αυτών είναι υψηλής ποιότητας και έχουν την ικανότητα να αποδίδουν σε αντίξοες συνθήκες.



Για τη μετατροπή του οχήματος σε όχημα στρατιωτικών προδιαγραφών απαιτείται η ολική επανασχεδιάσή του βασιζόμενη σε ίδιες μεν αρχές, αλλά με βελτιστοποιημένη κατασκευή και αναβαθμισμένους αισθητήρες.

## §2. Προτερήματα – σημεία υπεροχής

### 1. Απλότητα κατασκευής – τιμή – αναλωσιμότητα

Το όχημα έχει κατασκευαστεί με φθηνά εξαρτήματα που προέρχονται από το λιανικό εμπόριο. Κατά συνέπεια, είναι φθηνό και αναλώσιμο. Είναι ήδη ικανό να διεξάγει απλές αποστολές αναγνώρισης και περιπολίας εάν προσαρμοστεί σε αυτό μια κάμερα. Εάν αφαιρεθεί ο ιστός και στη θέση του προσαρμοστεί φακός και κάμερα, μπορεί να εκτελέσει αποστολές επιθεώρησης οχήματος σε σημείο ελέγχου.

Το όχημα είναι απλό στη χρήση και δεν απαιτεί καμία εξειδικευμένη γνώση. Ο τηλεχειρισμός του μπορεί να εκτελεστεί από οποιοδήποτε smartphone ή tablet, εφόσον εγκατασταθεί μια απλή εφαρμογή σε αυτό.

### 2. Αυτόνομες λειτουργίες

Οι αυτόνομες λειτουργίες προσδίδουν στο όχημα τη δυνατότητα να εκτελέσει αποστολές χωρίς να απαιτείται η συνεχής επίβλεψή του. Παραδείγματος χάριν, θα ήταν δυνατή η χρήση κάμερας και κάρτας αποθήκευσης SD ώστε να καταγράφεται η δραστηριότητα σε έναν εκτεταμένο εσωτερικό χώρο χωρίς την επίβλεψη από τον χειριστή του. Αυτή η δυνατότητα θα μπορούσε να συνδυαστεί με τη λειτουργία επιστροφής στο φάρο υπέρυθρων, ώστε το όχημα να μπορεί, αφού υλοποιηθεί κατάλληλη μετατροπή του, να φορτίζει τις μπαταρίες του σε προκαθορισμένο χώρο φόρτισης. Με αυτό τον τρόπο, ένας χειριστής θα μπορούσε να επιβλέπει περισσότερα από ένα οχήματα.

## ΚΕΦΑΛΑΙΟ 5

# Συμπεράσματα και μελλοντική ανάπτυξη

Αν και πρόκειται για μια φθηνή πειραματική συσκευή με περιορισμένες αυτόνομες δυνατότητες, το όχημα μπορεί να εξελιχθεί σε πλατφόρμα πολλαπλών ρόλων, ακόμα και αν δεν ανακατασκευαστεί ώστε να έχει στρατιωτικές προδιαγραφές. Για το άμεσο μέλλον προτείνονται οι εξής εξελίξεις:

### 1. Μετατροπή σε ερπυστριοφόρο όχημα

Κατ' αρχήν προτείνεται η μετατροπή του οχήματος σε ερπυστριοφόρο (αλλαγή των τροχών με ερπύστριες). Αυτό θα προσδώσει στο όχημα καλύτερη συμπεριφορά και επιδόσεις σε εδάφη που συναντά δυσκολία κίνησης.

Μπορεί να εξεταστεί η χρήση σασί με σύστημα υπερπήδησης εμποδίων (flippers). Αυτό θα προσδώσει στο όχημα τη δυνατότητα να υπερπηδά μικρά εμπόδια.

### 2. Εξέταση προσθήκης επιπλέον αισθητήρων υπερήχων – υπέρυθρων

Θα πρέπει να εξεταστεί η προσθήκη επιπλέον αισθητήρων. Αρχικά θα μπορούσαν να προσαρμοστούν αισθητήρες υπερήχων περιμετρικά του οχήματος, ώστε να παρέχουν ολόπλευρη κάλυψη με σκοπό τη λήψη περισσότερου περίπλοκων αποφάσεων για την αποφυγή εμποδίων. Επίσης η προσαρμογή ενός αισθητήρα

υπερήχων εμπρός με κλίση προς το έδαφος, ώστε να ανιχνεύεται η ύπαρξη κενού εμπρός από το όχημα.

Έπειτα, θα μπορούσε να εξεταστεί η προσθήκη ενός αισθητήρα υπέρυθρων προσαρμοσμένο σε σερβοκινητήρα, ώστε να ανιχνεύει ακόμα και μικρότερα ή λεπτότερα εμπόδια.

### **3. Εξέταση της διαμόρφωσης του φορτίου που πρέπει να φέρει, ανάλογα με το είδος της αποστολής που ανατίθεται να εκτελέσει**

Ανάλογα με το είδος της αποστολής που θα του ανατεθεί, θα πρέπει να εξεταστεί και η διαμόρφωση του φορτίου του οχήματος. Ενδεικτικά αναφέρονται ορισμένες αποστολές και προτεινόμενα φορτία:

- Περιπολία – αναγνώριση περιοχής (αυτόνομη και με απλή τηλεκατεύθυνση) – επιθεώρηση οχήματος σε σημείο ελέγχου: Εμπρόσθια κάμερα, κάρτα αποθήκευσης δεδομένων SD, φώτα LED περιμετρικά του οχήματος.
- Ανίχνευση ραδιενέργειας και πολεμικών χημικών ουσιών: Εμπρόσθια κάμερα, κάρτα αποθήκευσης δεδομένων SD, αισθητήρας Geiger με αυτόνομη πηγή ενέργειας, δεσμίδα αισθητήρων ΧΠΟ (ο συνδυασμός των αισθητήρων για την ανίχνευση χημικών πολεμικών ουσιών γίνεται λαμβάνοντας υπ' όψη τις αναμενόμενες απειλές ή τις απειλές που θεωρούνται σοβαρότερες).
- Χαρτογράφηση δωματίου: περιμετρικοί αισθητήρες υπέρυθρων, κάρτα αποθήκευσης SD, χρήση του φάρου υπέρυθρων ως σημείο αναφοράς.
- Αναζήτηση εκρηκτικών: Εμπρόσθια κάμερα. Μπορεί να εξεταστεί η προσαρμογή μικρού βραχίονα (μετατροπή με σκοπό την εκτέλεση απλών αποστολών EOD).

- Μεταφορά φορτίου στα πλαίσια της ομάδας μάχης: Εμπρόσθια κάμερα, σχάρα μεταφοράς, αντικατάσταση των κινητήρων με ισχυρότερους.

#### 4. Σμίκρυνση του οχήματος

Προτείνεται η επανασχεδίαση του οχήματος με σκοπό τη μείωση του μεγέθους του. Εάν το όχημα επανασχεδιαστεί ώστε να διαθέτει μικρότερο μέγεθος (μικρότερο των 10 εκατοστών – ίδιο μέγεθος με μια χειροβομβίδα και πολύ μικρότερο βάρος) διατηρώντας τις ίδιες δυνατότητες όσον αφορά στις αυτόνομες λειτουργίες και με την προθήκη κάμερας, θα είναι δυνατή η χρήση ενός ή περισσότερων ως αναλώσιμα οχήματα αναγνώρισης από τον κάθε μαχητή ατομικά, σε συνδυασμό με το κινητό του τηλέφωνο. Δηλαδή, θα μπορούσε ο κάθε μαχητής να φέρει ένα μικρό όχημα αναγνώρισης στον φόρτο του.

Θα μπορούσε επίσης να εξεταστεί η μεταφορά του οχήματος εντός κελύφους, το οποίο θα το προστατεύει από πτώσεις. Ο μαχητής θα μπορούσε να ενεργοποιεί μηχανισμό στο προστατευτικό κέλυφος αντίστοιχο με αυτόν μιας χειροβομβίδας και να το εκτοξεύει στο σημείο που επιθυμεί. Μετά την πτώση το κέλυφος θα απελευθερώνει το όχημα ώστε να εκτελέσει την αποστολή που επιθυμεί ο χειριστής του.

## Βιβλιογραφία - Ιστότοποι

1. Παρουσίαση Δρ. Νικόλαου Τσουρβελούδη «Εντροχα Ρομποτικά Οχήματ (ΕΡΟ) Κινηματική», όπως παρουσιάστηκε το 2016 στη ΣΣΕ, στα πλαίσια του μαθήματος «Μη Επανδρωμένα Ρομποτικά Οχήματα», στο Β' Εξάμηνο του Διδρυματικού Διατμηματικού Προγράμματος Μεταπτυχιακών Σπουδών «Σχεδίαση και Επεξεργασία Συστημάτων».
2. Simple, Real-Time Obstacle Avoidance Algorithm for Mobile Robots, των Ioan Susnea, Viorel Minzu, Grigore Vasiliu. Department of Control Engineering, University "Dunarea de Jos", Δεκέμβριος 2009.
3. DG02S-A130GEARMOTOR datasheet της DAGU Hi-Tech Electronic Co., LTD
4. DG012-BV Multi chassis 4WD kit (basic vesrsion) printed manual
5. Pololu IR Beacon Quick-Start Sheet της Pololu Corporation
6. Adafruit Motor Shield V2 for Arduino (created by lady ada), από την ιστοσελίδα <https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/overview> (29/7/16)
7. Product User's Manual – HC-SR04 Ultrasonic Sensor της Cytron Technologies
8. Phototransistor Products Specification HPTB3b-14D της HUEY JANN ELECTRONICS INDUSTRY CO., LTD.
9. 5mm Round Standard T-1 ¾ Type Infrared LED Technical Data Sheet της Lucky Light Electronics Co., Ltd.

10. <http://www.electronicshub.org/ir-sensor/> (26/03/17)
11. <https://learn.sparkfun.com/tutorials/bluetooth-basics> (23/06/17)
12. <http://www.hobbytronics.co.uk> (31/03/17)
13. <http://www.bustatech.com> (31/03/17)
14. <https://learn.adafruit.com> (31/03/17)
15. <https://www.adafruit.com> (31/03/17)
16. <http://www.electronics.com> (31/03/17)
17. <http://shallowsky.com> (31/03/17)
18. <https://www.pololu.com> (31/03/17)
19. <http://www.electro-tech-online.com> (31/03/17)
20. <http://www.electronics.com> (31/03/17)
21. <http://www.instructables.com> (31/03/17)
22. <https://www.elprocus.com> (31/03/17)
23. <http://www.army-technology.com/projects/irobot-110-firstlook-robot/>  
(02/04/17)
24. <http://sdractical.com/MLT-Jack-Russell/> (02/04/17)
25. <https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/overview>  
(02/04/17)
26. <https://www.pololu.com/product/701> (04/04/17)
27. [http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic\\_sensor/1.html#basicunderstanding](http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html#basicunderstanding) (23/6/17)

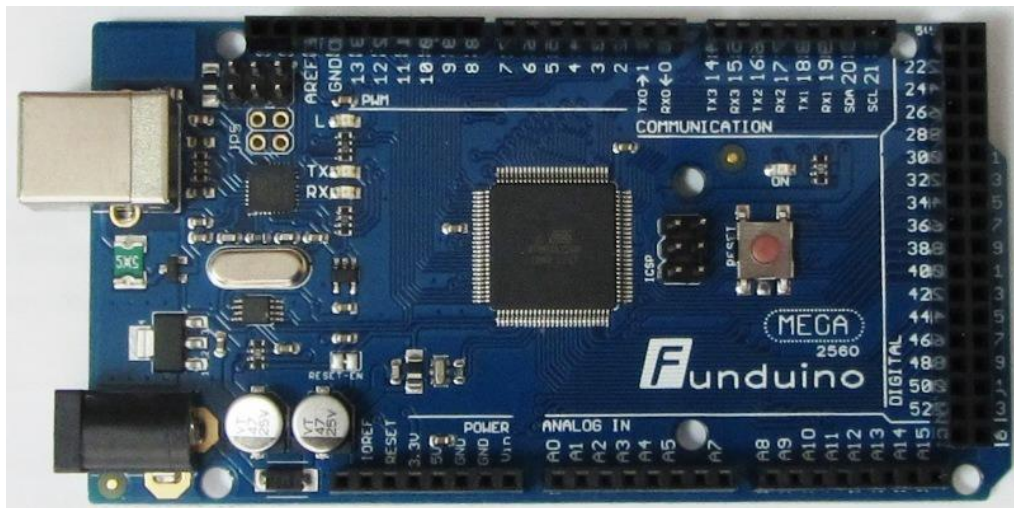
## ΠΑΡΑΡΤΗΜΑ Α΄

### Εξαρτήματα

- Πλαίσιο και κινητήρες από το σετ Multi-chassis 4WD kit με 4 κινητήρες (DC) DG02S-A130 Gearmotor (12)

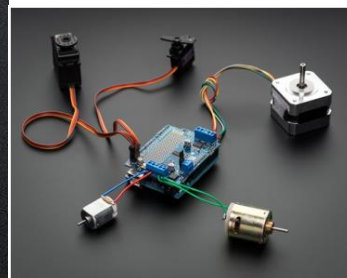
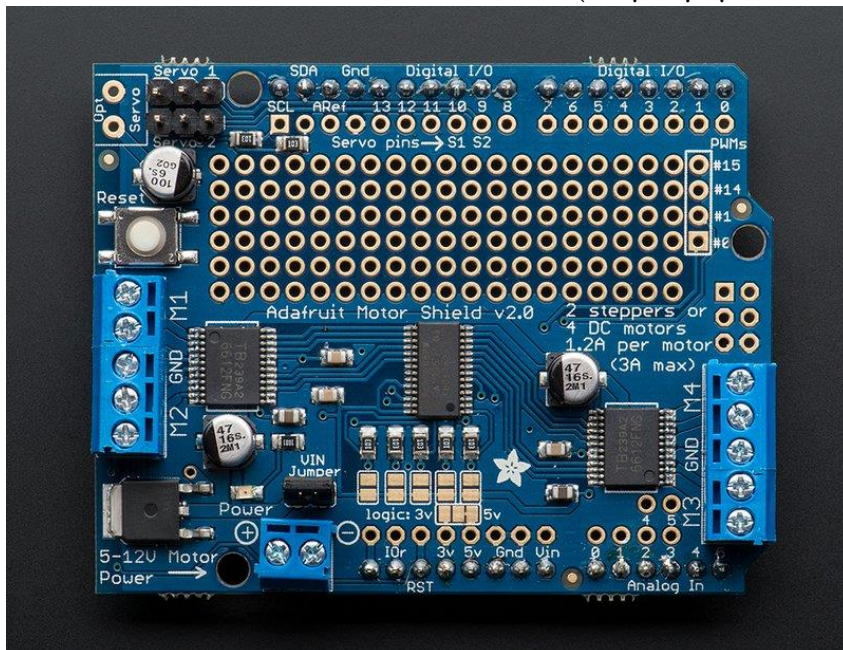


- Arduino Mega clone (Funduino Mega 2560) (13)

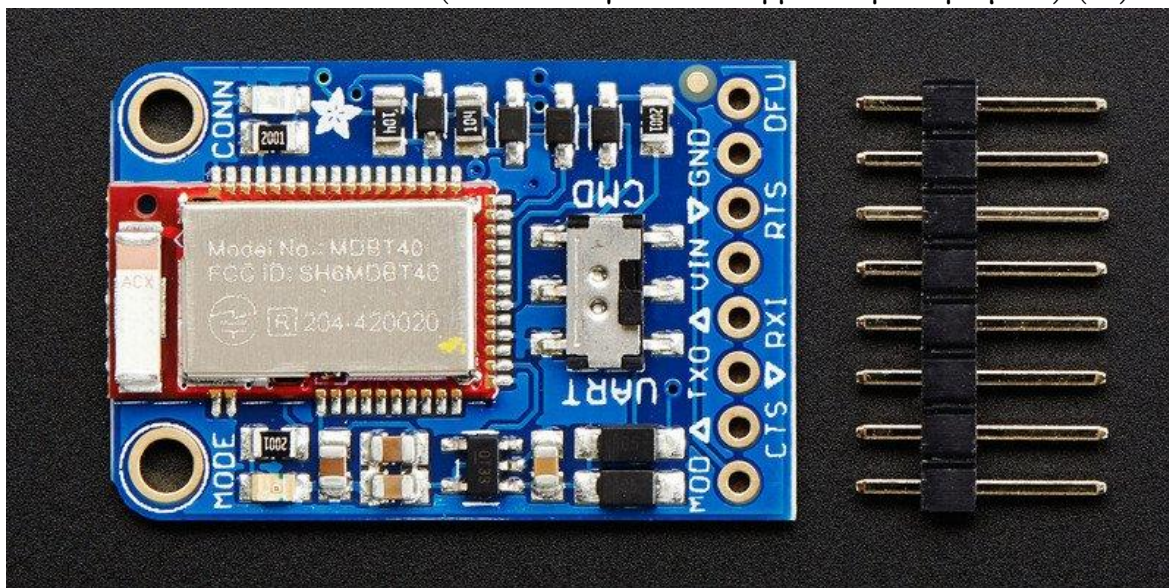




- Adafruit Motor Shield V2 for Arduino (ελεγκτής για τους κινητήρες) (14)



- Adafruit Bluefruit LE UART (Bluetooth για σύνδεση με κινητό τηλέφωνο) (14)



- HC-SR04 Ultrasonic range sensor (αποστασιόμετρο υπερήχων) (16)



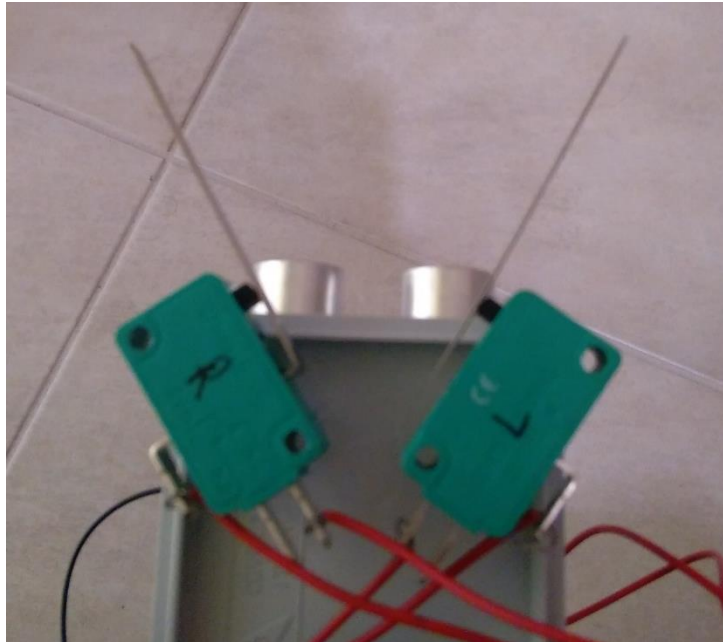
- Buzzer (17)



- Pololu IR beacon receiver set (18)



- Δύο διακόπτες (για το σύστημα bumper)



- Απλή Φωτοдиодος (για το σύστημα προσέγγισης) (19)



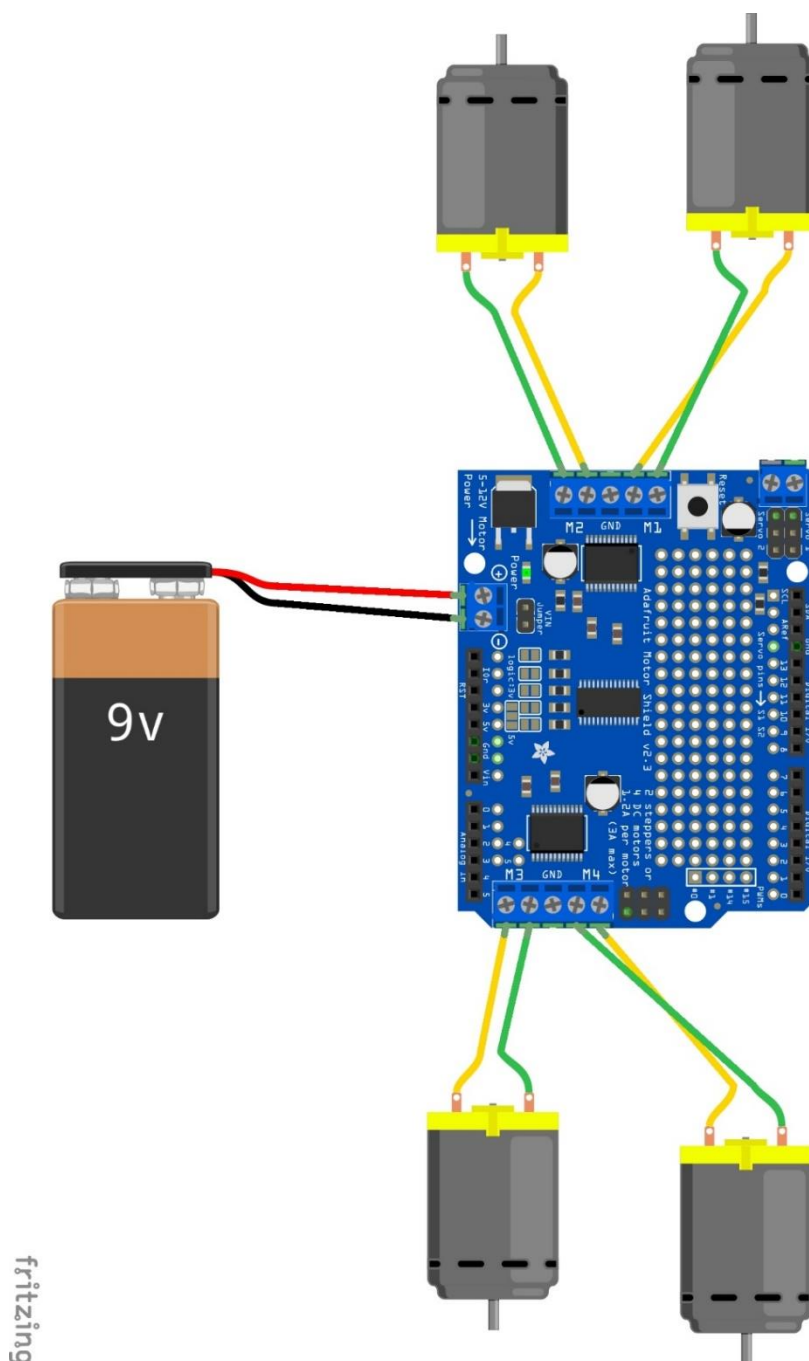
- Mini breadboards, καλώδια, αντιστάσεις, απλοί διακόπτες on-off κτλ. (16, 21, 22)

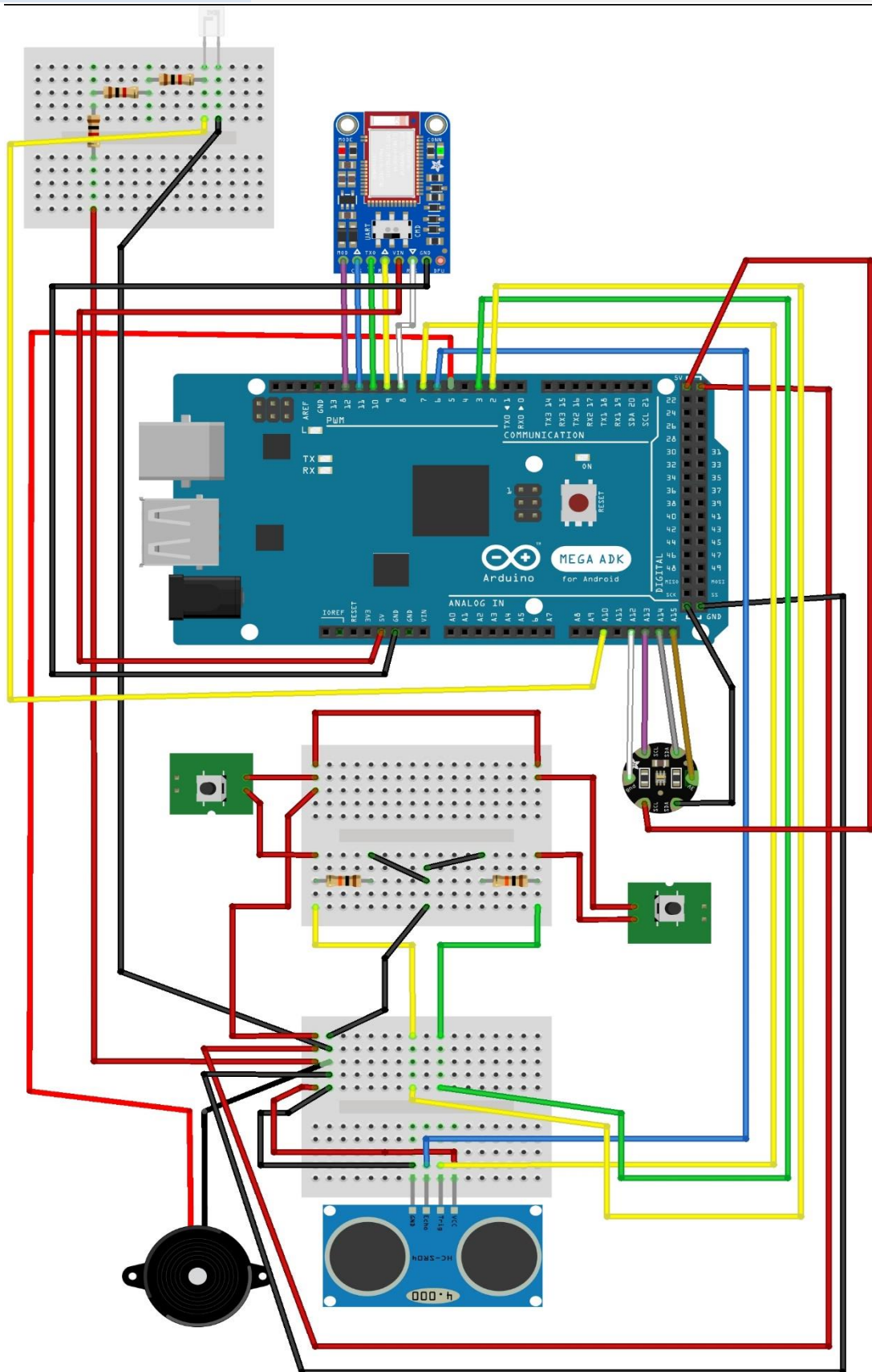


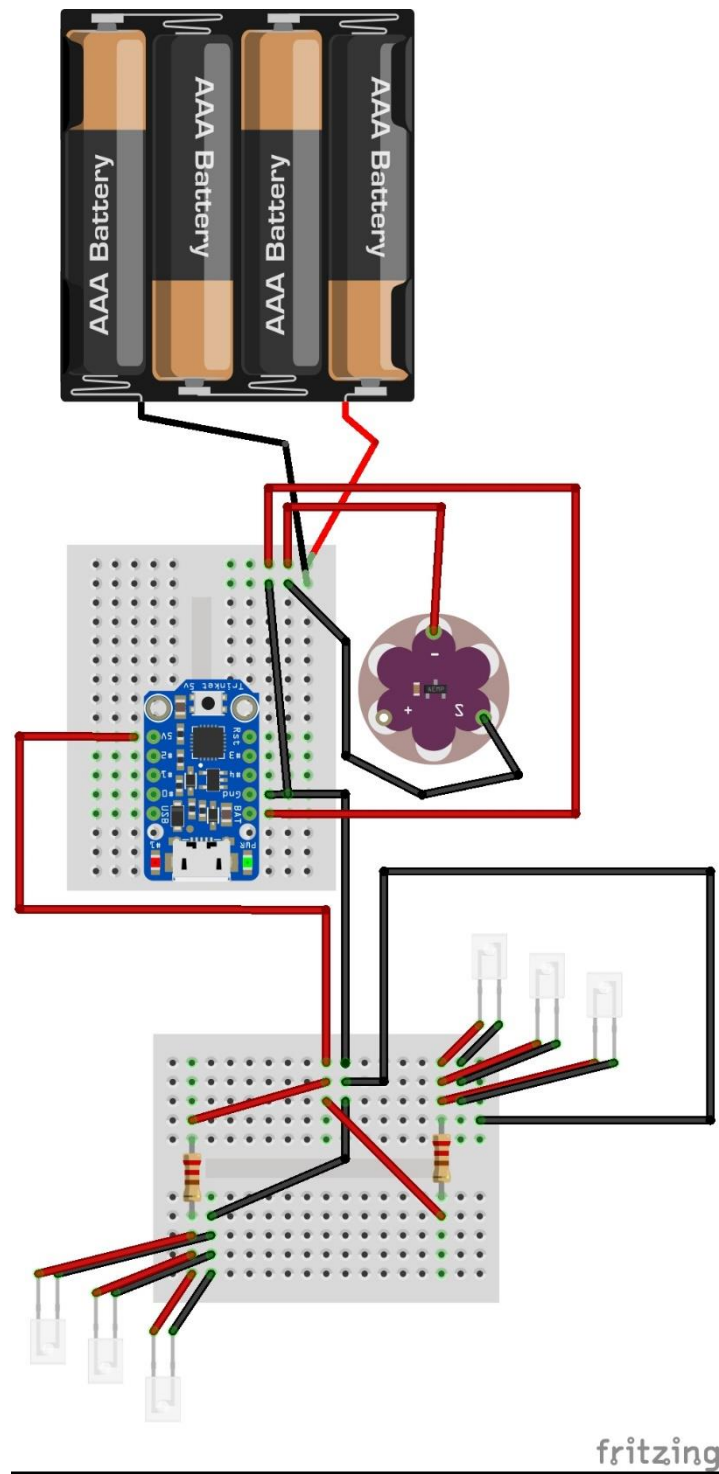


## ΠΑΡΑΡΤΗΜΑ Β΄

Συνδεσμολογία αισθητήρων - κινητήρων







### Επεξήγηση συνδεσμολογίας αισθητήρων

- Το Adafruit Bluetooth LE UART είναι το Bluetooth module που χρησιμοποιείται για τη μεταβίβαση εντολών από τη συσκευή android προς το όχημα. Χρησιμοποιεί τα digital pins 8, 9, 10, 11, 12 στο Arduino. Επίσης χρειάζεται 5V και pin γείωσης (GND).
- Ο αισθητήρας υπερήχων HC-SR04 χρησιμοποιεί τα digital pins 6,7 στο Arduino. Επίσης χρειάζεται 5V και pin γείωσης (GND).
- Το buzzer χρησιμοποιεί το digital pin 5. Επίσης χρειάζεται 5V και pin γείωσης (GND).
- Τα bumpers χρησιμοποιούν τα digital pins 2, 3 (για το Arduino Mega είναι interrupt pins).
- Το Pololu IR beacon χρησιμοποιεί τα analog pins 12, 13, 14, 15, το digital pin 31 (enable pin, προαιρετικά) και επίσης χρειάζεται 5V και pin γείωσης.
- Ο αισθητήρας προσέγγισης χρησιμοποιεί το analog pin 10. Χρειάζεται επίσης 5V και pin γείωσης.
- Το motor shield δεσμεύει τα digital pins 20, 21.

### Επεξήγηση αυτοσχέδιου αισθητήρα προσέγγισης

Ο φάρος (beacon) που χρησιμοποιείται για την προσέγγιση του οχήματος εκπέμπει, εκτός της IR ακτινοβολίας του Pololu IR set, μια μικρής ισχύος ακτινοβολία IR από λυχνίες IR σε διαφορετικό μήκος κύματος. Το απλό phototransistor, με τη συγκεκριμένη συνδεσμολογία αντιστάσεων, μπορεί να την ανιχνεύσει σε απόσταση περίπου 10 εκατοστών, οπότε το όχημα θεωρεί ότι προσέγγισε με επιτυχία το φάρο.

# ΠΑΡΑΡΤΗΜΑ Γ'

## Κώδικας (Arduino Mega 2560 r3)

Παρακάτω παρατίθεται ο πλήρης κώδικας, για Arduino Mega 2560. Έχει συνταχθεί με τον compiler Arduino έκδοσης 1.6.11. Μαζί με τον παρακάτω κώδικα πρέπει να συμπεριληφθούν και τα συνοδευτικά modules BluefruitConfig.h και packetParser.cpp, τα οποία περιλαμβάνονται στις βιβλιοθήκες του Adafruit Bluefruit LE UART.

```
//MOTORSHIELD SETTINGS
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWM_ServoDriver.h"

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();

//Create DC Motor objects
Adafruit_DCMotor *motor1 = AFMS.getMotor(1);
Adafruit_DCMotor *motor2 = AFMS.getMotor(2);
Adafruit_DCMotor *motor3 = AFMS.getMotor(3);
Adafruit_DCMotor *motor4 = AFMS.getMotor(4);

//HC-SR04 Ultrasonic Sensor Settings
const int echoPin = 6;
const int trigPin = 7;
long duration, distance;

//IR Receiver pins
const int westPin = A13;
const int southPin = A15;
const int eastPin = A12;
const int northPin = A14;
const int enablePin = 31;

//Bumper pins and variables
const int leftpin = 2;
const int rightpin = 3;
const int bumperout = 13;
volatile int avoidright = 0;
volatile int avoidleft = 0;

//Buzzer Pin
const int buzzPin = 5;

//Distance from beacon IR photodiode
const int disPin = A10;

//Set inrange variable (beacon in range or not)
```



```
boolean inrange = false;

//Set initial speed to lowest value
int speed = 4;
int RUNAT = 0;

//BLUETOOTH SETTINGS
#include <string.h>
#include <Arduino.h>
#include <SPI.h>
#if not defined (_VARIANT_ARDUINO_DUE_X_) && not defined (_VARIANT_ARDUINO_ZERO_)
  #include <SoftwareSerial.h>
#endif
#include "Adafruit_BLE.h"
#include "Adafruit_BluefruitLE_SPI.h"
#include "Adafruit_BluefruitLE_UART.h"
#include "BluefruitConfig.h"
#define BUFSIZE 128
#define VERBOSE_MODE true
#define BLE_READPACKET_TIMEOUT 500
#define BLUEFRUIT_SWUART_RXD_PIN 9
#define BLUEFRUIT_SWUART_TXD_PIN 10
#define BLUEFRUIT_UART_CTS_PIN 11
#define BLUEFRUIT_UART_RTS_PIN 8
#ifdef Serial1
  #define BLUEFRUIT_HWSERIAL_NAME Serial1
#else
  #define BLUEFRUIT_UART_MODE_PIN 12
  #define FACTORYRESET_ENABLE 0
  #define MINIMUM_FIRMWARE_VERSION "0.6.6"
  #define MODE_LED_BEHAVIOUR "MODE"
  SoftwareSerial bluefruitSS = SoftwareSerial(BLUEFRUIT_SWUART_TXD_PIN,
BLUEFRUIT_SWUART_RXD_PIN);
  Adafruit_BluefruitLE_UART ble(bluefruitSS, BLUEFRUIT_UART_MODE_PIN,
BLUEFRUIT_UART_CTS_PIN, BLUEFRUIT_UART_RTS_PIN);

void error(const __FlashStringHelper*err) {
  Serial.println(err);
  while (1);
}

uint8_t readPacket(Adafruit_BLE *ble, uint16_t timeout);
float parsefloat(uint8_t *buffer);
void printHex(const uint8_t * data, const uint32_t numBytes);
extern uint8_t packetbuffer[];

void setup(void) {

  //Allocate bumper pins and declare interrupts and variables
  pinMode (leftpin, INPUT);
  pinMode (rightpin, INPUT);
  pinMode(bumperout, OUTPUT);
  digitalWrite(bumperout, LOW);
  attachInterrupt(digitalPinToInterrupt(rightpin), rightbumper, LOW);
  attachInterrupt(digitalPinToInterrupt(leftpin), leftbumper, LOW);

  //Allocate IR Receiver pins
  pinMode(westPin, INPUT);
  pinMode(southPin, INPUT);
```

```
pinMode(eastPin, INPUT);
pinMode(northPin, INPUT);

//Allocate ultrasonic sensor pins
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

//Allocate Buzzer pin
pinMode(buzzPin, OUTPUT);

//Allocate distance from beacon sensor PIN
pinMode(disPin, INPUT);

//Start the motor shield
AFMS.begin();

//Set initial motor speeds
motor1->setSpeed(100);
motor2->setSpeed(100);
motor3->setSpeed(100);
motor4->setSpeed(100);

//Bluetooth Settings
char Dir = "RELEASE";
while (!Serial); // required for Flora & Micro
delay(500);
Serial.begin(9600);
/* Initialise the module */
Serial.print(F("Initialising the Bluefruit LE module: "));

if ( !ble.begin(VERBOSE_MODE) )
{
  error(F("Couldn't find Bluefruit, make sure it's in CoMmanD mode & check wiring?"));
}
Serial.println( F("OK!") );
if ( FACTORYRESET_ENABLE )
{
  Serial.println(F("Performing a factory reset: "));
  if ( ! ble.factoryReset() ){
    error(F("Couldn't factory reset"));
  }
}
ble.echo(false);
Serial.println("Requesting Bluefruit info:");
ble.info();
Serial.println(F("Please use Adafruit Bluefruit LE app to connect in Controller mode"));
Serial.println(F("Then activate/use the sensors, color picker, game controller, etc!"));
Serial.println();
ble.verbose(false); // debug info is a little annoying after this point!
while (! ble.isConnected()) {
  delay(500);
}
Serial.println(F("*****"));
if ( ble.isVersionAtLeast(MINIMUM_FIRMWARE_VERSION) )
{
  Serial.println(F("Change LED activity to " MODE_LED_BEHAVIOUR));
  ble.sendCommandCheckOK("AT+HWMODELED=" MODE_LED_BEHAVIOUR);
}
Serial.println( F("Switching to DATA mode!") );
ble.setMode(BLUEFRUIT_MODE_DATA);
```

```
Serial.println(F("*****"));

//Shuffle random function for random number later
randomSeed(analogRead(0));
}

void loop(void) {

  //Bumper input initially OFF
  digitalWrite(bumperout, LOW);

  //IR beacon initially set out of range for the loop
  inrange = false;

  //Bluetooth Settings
  uint8_t len = readPacket(&ble, BLE_READPACKET_TIMEOUT);
  if (len == 0) return;

  //Color
  if (packetbuffer[1] == 'C') {
    uint8_t red = packetbuffer[2];
    uint8_t green = packetbuffer[3];
    uint8_t blue = packetbuffer[4];
    Serial.print ("RGB #");
    if (red < 0x10) Serial.print("0");
    Serial.print(red, HEX);
    if (green < 0x10) Serial.print("0");
    Serial.print(green, HEX);
    if (blue < 0x10) Serial.print("0");
    Serial.println(blue, HEX);
  }

  //Buttons
  if (packetbuffer[1] == 'B') {
    uint8_t buttnum = packetbuffer[2] - '0';
    boolean pressed = packetbuffer[3] - '0';

    //Initial Reads from sensors

    //Read beacon variables
    volatile int N = analogRead(northPin);
    volatile int S = analogRead(southPin);
    volatile int E = analogRead(eastPin);
    volatile int W = analogRead(westPin);

    //Ultrasonic pulse away - calculations
    digitalWrite(trigPin, LOW);
    delay(10);
    digitalWrite(trigPin, HIGH);
    delay(10);
    digitalWrite(trigPin, LOW);

    //Calculate the duration of the interval
    duration=pulseIn(echoPin, HIGH);

    //Calculate the distance from sensor (duration/29/2 gives cm)
    distance=duration/29/2;

    //Display on Serial Monitor
```

```
Serial.print(distance);
Serial.print(" cm");
Serial.println();
delay(50);

//Is the beacon in range?
int range = analogRead(disPin);
if (range < 900) {
    inrange = true;
}

//Directions (arrow buttons)
if (pressed) {
    switch(buttnum){
        case 5:
            motor1->run(FORWARD);
            motor2->run(FORWARD);
            motor3->run(FORWARD);
            motor4->run(FORWARD);
            Serial.println ("FORWARD");
            break;
        case 6:
            motor1->run(BACKWARD);
            motor2->run(BACKWARD);
            motor3->run(BACKWARD);
            motor4->run(BACKWARD);
            Serial.println ("BACKWARD");
            break;
        case 8:
            motor1->run(BACKWARD);
            motor2->run(BACKWARD);
            motor3->run(FORWARD);
            motor4->run(FORWARD);
            Serial.println ("ROTATE RIGHT");
            break;
        case 7:
            motor1->run(FORWARD);
            motor2->run(FORWARD);
            motor3->run(BACKWARD);
            motor4->run(BACKWARD);
            Serial.println ("ROTATE LEFT");
            break;
    }
}

//Increase Speed (button 1)
case 1:
    ++speed;
    RUNAT = speed * 25;
    motor1->setSpeed(RUNAT);
    motor2->setSpeed(RUNAT);
    motor3->setSpeed(RUNAT);
    motor4->setSpeed(RUNAT);
    Serial.print ("SPEED SET TO ");
    Serial.println (RUNAT);
    for (int x= 0; x <= speed; x++) {
        digitalWrite(buzzPin, HIGH);
        delay(50);
        digitalWrite(buzzPin, LOW);
        delay(50);
    }
    if (speed > 9) {
```

```
        speed = 9;
    }
    break;

//Reduce Speed (button 2)
case 2:
    --speed;
    RUNAT = speed * 25;
    motor1->setSpeed(RUNAT);
    motor2->setSpeed(RUNAT);
    motor3->setSpeed(RUNAT);
    motor4->setSpeed(RUNAT);
    Serial.print ("SPEED SET TO ");
    Serial.println (RUNAT);
    for (int x= 0; x <= speed; x++) {
        digitalWrite(buzzPin, HIGH);
        delay(50);
        digitalWrite(buzzPin, LOW);
        delay(50);
    }
    if (speed < 0) {
        speed = 0;
    }
    break;

//Autonomous RTB
case 3:
    // Set a medium speed (best for practical purposes)
    motor1->setSpeed(100);
    motor2->setSpeed(100);
    motor3->setSpeed(100);
    motor4->setSpeed(100);
    // Start the loop
    for(int time = 0; time < 600; time++){

        // Check for the beacon's location - get readings from analog IR sensors
        volatile int N = analogRead(northPin);
        volatile int S = analogRead(southPin);
        volatile int E = analogRead(eastPin);
        volatile int W = analogRead(westPin);

        // Check possible obstacle distance from ultrasonic sensor
        // Ultrasonic pulse away - calculations
        digitalWrite(trigPin, LOW);
        delay(10);
        digitalWrite(trigPin, HIGH);
        delay(10);
        digitalWrite(trigPin, LOW);
        //Calculate the duration of the interval
        duration=pulseIn(echoPin, HIGH);
        //Calculate the distance from sensor (duration/29/2 gives cm)
        distance=duration/29/2;

        //Is the beacon in range?
        range = analogRead(disPin);
        if (range < 900) {
            inrange = true;
            Serial.println("BEACON IN RANGE");
            digitalWrite(buzzPin, HIGH);
            delay(500);
        }
    }
}
```

```
digitalWrite(buzzPin, LOW);  
delay(500);  
}  
  
if (inrange == true) {  
  motor1->run(RELEASE);  
  motor2->run(RELEASE);  
  motor3->run(RELEASE);  
  motor4->run(RELEASE);  
  inrange = false;  
  break;  
} else if (avoidright >= 5) {  
  Serial.println("RIGHT BUMPER ACTIVATED");  
  digitalWrite(buzzPin, HIGH);  
  motor1->run(RELEASE);  
  motor2->run(RELEASE);  
  motor3->run(RELEASE);  
  motor4->run(RELEASE);  
  delay(500);  
  motor1->run(BACKWARD);  
  motor2->run(BACKWARD);  
  motor3->run(BACKWARD);  
  motor4->run(BACKWARD);  
  delay(1000);  
  motor1->run(FORWARD);  
  motor2->run(FORWARD);  
  motor3->run(BACKWARD);  
  motor4->run(BACKWARD);  
  delay(500);  
  motor1->run(FORWARD);  
  motor2->run(FORWARD);  
  motor3->run(FORWARD);  
  motor4->run(FORWARD);  
  delay(1500);  
  digitalWrite(buzzPin, LOW);  
  avoidright = 0;  
} else if (avoidleft >= 5) {  
  Serial.println("LEFT BUMPER ACTIVATED");  
  digitalWrite(buzzPin, HIGH);  
  motor1->run(RELEASE);  
  motor2->run(RELEASE);  
  motor3->run(RELEASE);  
  motor4->run(RELEASE);  
  delay(500);  
  motor1->run(BACKWARD);  
  motor2->run(BACKWARD);  
  motor3->run(BACKWARD);  
  motor4->run(BACKWARD);  
  delay(1000);  
  motor1->run(BACKWARD);  
  motor2->run(BACKWARD);  
  motor3->run(FORWARD);  
  motor4->run(FORWARD);  
  delay(500);  
  motor1->run(FORWARD);  
  motor2->run(FORWARD);  
  motor3->run(FORWARD);  
  motor4->run(FORWARD);  
  delay(1500);  
  digitalWrite(buzzPin, LOW);
```

```
avoidleft = 0;
}else if (distance < 10) {
motor1->run(BACKWARD);
motor2->run(BACKWARD);
motor3->run(BACKWARD);
motor4->run(BACKWARD);
delay(1000);
    if (time < 300){
        motor1->run(BACKWARD);
        motor2->run(BACKWARD);
        motor3->run(FORWARD);
        motor4->run(FORWARD);
        delay(500);
    }
    if (time >= 300) {
        motor1->run(FORWARD);
        motor2->run(FORWARD);
        motor3->run(BACKWARD);
        motor4->run(BACKWARD);
        delay(500);
    }
motor1->run(FORWARD);
motor2->run(FORWARD);
motor3->run(FORWARD);
motor4->run(FORWARD);
delay(1500);
motor1->run(RELEASE);
motor2->run(RELEASE);
motor3->run(RELEASE);
motor4->run(RELEASE);
} else if (N > 500 && S > 500 && E < 900 && W > 500) {
motor1->run(BACKWARD);
motor2->run(BACKWARD);
motor3->run(FORWARD);
motor4->run(FORWARD);
Serial.println("EAST");
} else if (N > 500 && S > 500 && E >500 && W < 900) {
motor1->run(FORWARD);
motor2->run(FORWARD);
motor3->run(BACKWARD);
motor4->run(BACKWARD);
Serial.println("WEST");
} else if (N < 900 && S > 500 && E > 500 && W > 500) {
motor1->run(FORWARD);
motor2->run(FORWARD);
motor3->run(FORWARD);
motor4->run(FORWARD);
Serial.println("NORTH");
} else if (N > 500 && S < 900 && E > 500 && W > 500) {
int randomNumber = random(100);
if (randomNumber < 50) {
    motor1->run(BACKWARD);
    motor2->run(BACKWARD);
    motor3->run(FORWARD);
    motor4->run(FORWARD);
    delay(1800);
    motor1->run(RELEASE);
    motor2->run(RELEASE);
    motor3->run(RELEASE);
    motor4->run(RELEASE);
}
```

```
    } else {
        motor1->run(FORWARD);
        motor2->run(FORWARD);
        motor3->run(BACKWARD);
        motor4->run(BACKWARD);
        delay(1800);
        motor1->run(RELEASE);
        motor2->run(RELEASE);
        motor3->run(RELEASE);
        motor4->run(RELEASE);
    }
    Serial.println("SOUTH");
}
digitalWrite(bumperout, LOW);
delay(100);
}
break;

//Autonomous Patrol for 2 mins (button 4)
case 4:
    motor1->setSpeed(100);
    motor2->setSpeed(100);
    motor3->setSpeed(100);
    motor4->setSpeed(100);

// Start the loop
for (int time = 0; time < 1200; time++) {

    // Ultrasonic pulse away - calculations
    digitalWrite(trigPin, LOW);
    delay(10);
    digitalWrite(trigPin, HIGH);
    delay(10);
    digitalWrite(trigPin, LOW);
    //Calculate the duration of the interval
    duration=pulseIn(echoPin, HIGH);
    //Calculate the distance from sensor (duration/29/2 gives cm)
    distance=duration/29/2;

    //Bumper turns (microcrashes)
    if (avoidright >= 5) {
        Serial.println("RIGHT BUMPER ACTIVATED");
        digitalWrite(buzzPin, HIGH);
        motor1->run(RELEASE);
        motor2->run(RELEASE);
        motor3->run(RELEASE);
        motor4->run(RELEASE);
        delay(500);
        motor1->run(BACKWARD);
        motor2->run(BACKWARD);
        motor3->run(BACKWARD);
        motor4->run(BACKWARD);
        delay(1000);
        motor1->run(FORWARD);
        motor2->run(FORWARD);
        motor3->run(BACKWARD);
        motor4->run(BACKWARD);
        delay(500);
        motor1->run(FORWARD);
        motor2->run(FORWARD);
```



```
motor3->run(FORWARD);
motor4->run(FORWARD);
delay(1500);
digitalWrite(buzzPin, LOW);
avoidright = 0;
}
if (avoidleft >=5) {
  Serial.println("LEFT BUMPER ACTIVATED");
  digitalWrite(buzzPin, HIGH);
  motor1->run(RELEASE);
  motor2->run(RELEASE);
  motor3->run(RELEASE);
  motor4->run(RELEASE);
  delay(500);
  motor1->run(BACKWARD);
  motor2->run(BACKWARD);
  motor3->run(BACKWARD);
  motor4->run(BACKWARD);
  delay(1000);
  motor1->run(BACKWARD);
  motor2->run(BACKWARD);
  motor3->run(FORWARD);
  motor4->run(FORWARD);
  delay(500);
  motor1->run(FORWARD);
  motor2->run(FORWARD);
  motor3->run(FORWARD);
  motor4->run(FORWARD);
  delay(1500);
  digitalWrite(buzzPin, LOW);
  avoidleft = 0;
}

//Turn when faced with obstacle (ultrasonic)
if (distance < 20){

  //Randomly turn left or right
  int randomNumber = random(100);
  if (randomNumber < 50) {
    motor1->run(RELEASE);
    motor2->run(RELEASE);
    motor3->run(RELEASE);
    motor4->run(RELEASE);
    delay(500);
    motor1->run(BACKWARD);
    motor2->run(BACKWARD);
    motor3->run(FORWARD);
    motor4->run(FORWARD);
    delay(1000);
    motor1->run(RELEASE);
    motor2->run(RELEASE);
    motor3->run(RELEASE);
    motor4->run(RELEASE);
  } else {
    motor1->run(RELEASE);
    motor2->run(RELEASE);
    motor3->run(RELEASE);
    motor4->run(RELEASE);
    delay(500);
    motor1->run(FORWARD);
  }
}
```

```
        motor2->run(FORWARD);
        motor3->run(BACKWARD);
        motor4->run(BACKWARD);
        delay(1000);
        motor1->run(RELEASE);
        motor2->run(RELEASE);
        motor3->run(RELEASE);
        motor4->run(RELEASE);
    }
} else {
    motor1->run(FORWARD);
    motor2->run(FORWARD);
    motor3->run(FORWARD);
    motor4->run(FORWARD);
}
digitalWrite(bumperout, LOW);
digitalWrite(buzzPin, HIGH);
delay(50);
digitalWrite(buzzPin, LOW);
delay(50);
}
motor1->run(RELEASE);
motor2->run(RELEASE);
motor3->run(RELEASE);
motor4->run(RELEASE);
digitalWrite(buzzPin, HIGH);
delay(500);
digitalWrite(buzzPin, LOW);
delay(500);
break;
default:
    motor1->run(RELEASE);
    motor2->run(RELEASE);
    motor3->run(RELEASE);
    motor4->run(RELEASE);
    break;
}
} else {
    motor1->run(RELEASE);
    motor2->run(RELEASE);
    motor3->run(RELEASE);
    motor4->run(RELEASE);
}
}

// GPS Location
if (packetbuffer[1] == 'L') {
    float lat, lon, alt;
    lat = parsefloat(packetbuffer+2);
    lon = parsefloat(packetbuffer+6);
    alt = parsefloat(packetbuffer+10);
    Serial.print("GPS Location\t");
    Serial.print("Lat: "); Serial.print(lat, 4); // 4 digits of precision!
    Serial.print('\t');
    Serial.print("Lon: "); Serial.print(lon, 4); // 4 digits of precision!
    Serial.print('\t');
    Serial.print(alt, 4); Serial.println(" meters");
}

// Accelerometer
```

```
if (packetbuffer[1] == 'A') {
    float x, y, z;
    x = parsefloat(packetbuffer+2);
    y = parsefloat(packetbuffer+6);
    z = parsefloat(packetbuffer+10);
    Serial.print("Accel\t");
    Serial.print(x); Serial.print('\t');
    Serial.print(y); Serial.print('\t');
    Serial.print(z); Serial.println();
}

// Magnetometer
if (packetbuffer[1] == 'M') {
    float x, y, z;
    x = parsefloat(packetbuffer+2);
    y = parsefloat(packetbuffer+6);
    z = parsefloat(packetbuffer+10);
    Serial.print("Mag\t");
    Serial.print(x); Serial.print('\t');
    Serial.print(y); Serial.print('\t');
    Serial.print(z); Serial.println();
}

// Gyroscope
if (packetbuffer[1] == 'G') {
    float x, y, z;
    x = parsefloat(packetbuffer+2);
    y = parsefloat(packetbuffer+6);
    z = parsefloat(packetbuffer+10);
    Serial.print("Gyro\t");
    Serial.print(x); Serial.print('\t');
    Serial.print(y); Serial.print('\t');
    Serial.print(z); Serial.println();
}

// Quaternions
if (packetbuffer[1] == 'Q') {
    float x, y, z, w;
    x = parsefloat(packetbuffer+2);
    y = parsefloat(packetbuffer+6);
    z = parsefloat(packetbuffer+10);
    w = parsefloat(packetbuffer+14);
    Serial.print("Quat\t");
    Serial.print(x); Serial.print('\t');
    Serial.print(y); Serial.print('\t');
    Serial.print(z); Serial.print('\t');
    Serial.print(w); Serial.println();
}
}

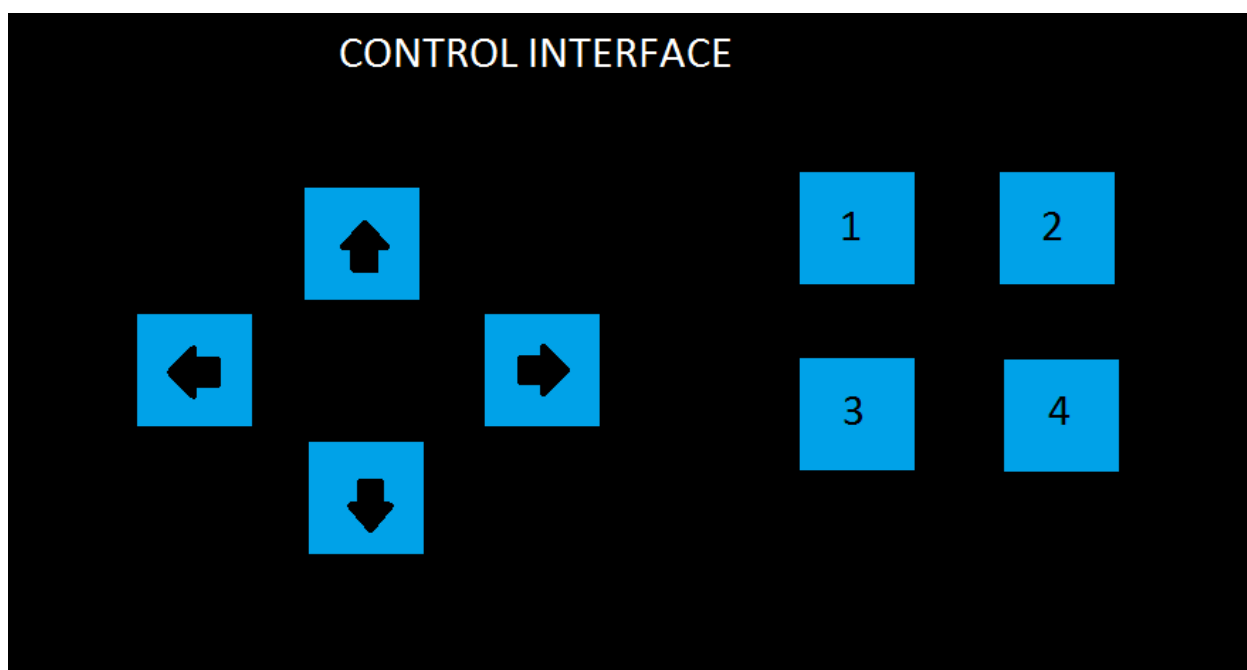
void rightbumper() {
    avoidright = ++avoidright;
}

void leftbumper() {
    avoidleft = ++avoidleft;
}
```

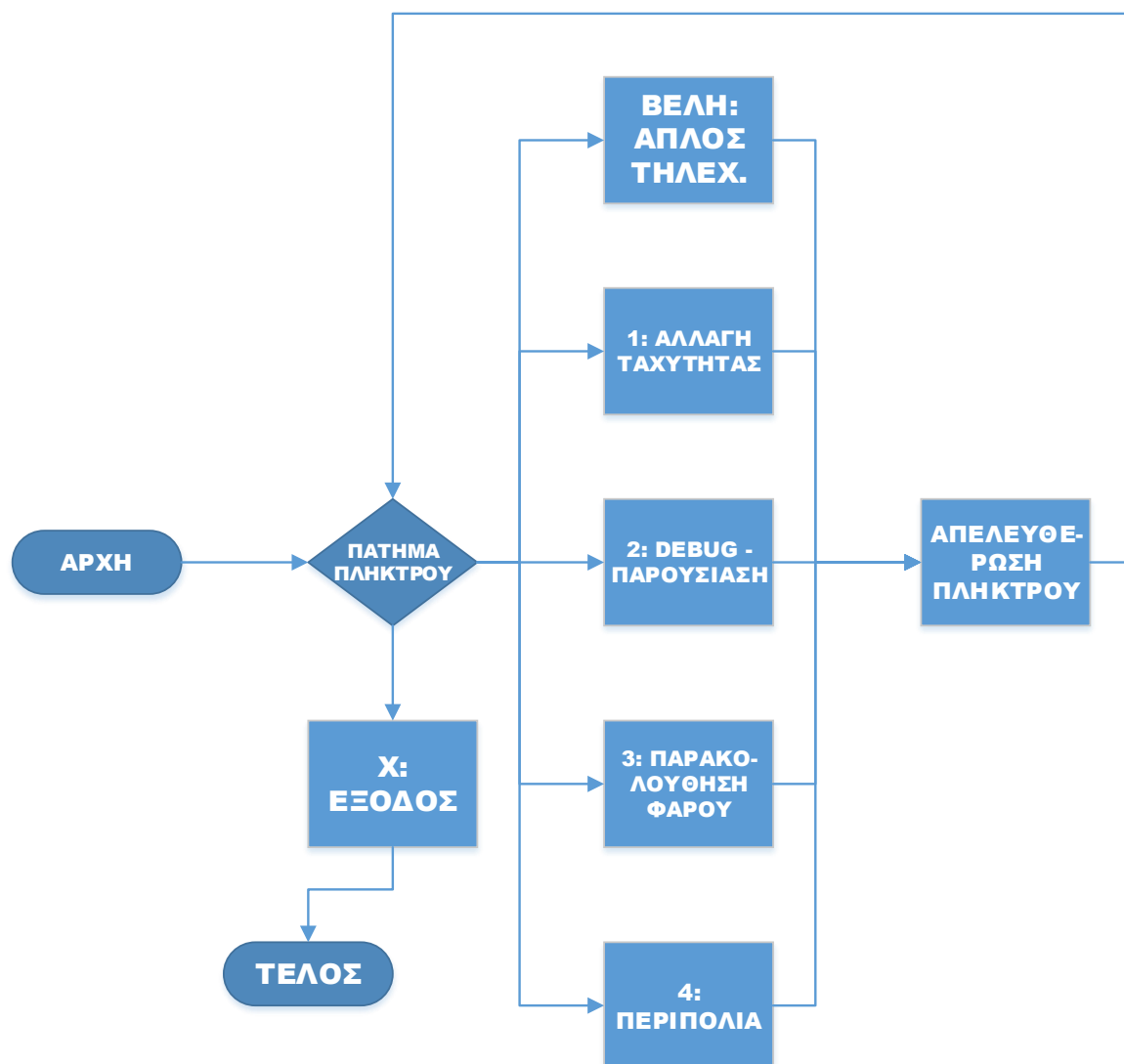
## ΠΑΡΑΡΤΗΜΑ Δ'

### Χειρισμός

Κατ' αρχήν ο χρήστης πρέπει να εγκαταστήσει στη συσκευή του την εφαρμογή Bluefruit LE (από το Android Playstore εφ' όσον πρόκειται για συσκευή με λειτουργικό σύστημα android ή από το Apple Store εφ' όσον πρόκειται για συσκευή με λειτουργικό iOS). Αφού ενεργοποιήσει τη λειτουργία Bluetooth της συσκευής του και εκκινήσει την εφαρμογή, θα κληθεί να συνδέσει το όχημα με τη φορητή συσκευή. Αφού υλοποιηθεί αυτό το βήμα, θα πρέπει να επιλέξει τη λειτουργία "Control Pad" από το μενού που θα εμφανιστεί στη συσκευή του. Μόλις κάνει την επιλογή αυτή, εμφανίζεται στην οθόνη του το χειριστήριο του οχήματος:



- **Πλήκτρα διεύθυνσης:** κίνηση εμπρός, πίσω, περιστροφή δεξιά, περιστροφή αριστερά (απλή τηλεκατεύθυνση)
- **Πλήκτρο 1:** αύξηση ταχύτητας ανά 25 rpm (έως 225 rpm)
- **Πλήκτρο 2:** μείωση ταχύτητας ανά 25 rpm (έως 0 rpm)
- **Πλήκτρο 3:** αυτόνομη ανίχνευση φάρου και παρακολούθησή για 1 λεπτό σε εσωτερικό χώρο με εμπόδια
- **Πλήκτρο 4:** αυτόνομη λειτουργία περιπολίας για 2 λεπτά σε εσωτερικό χώρο με εμπόδια



## ΠΑΡΑΡΤΗΜΑ Ε΄

### Πίνακες Τεχνικών Χαρακτηριστικών

#### Διαστάσεις Οχήματος - Επιδόσεις

Μήκος	26 cm
Πλάτος	15.5 cm
Ύψος	23.5 cm
Βάρος	900 gr
Μέγιστη Ταχύτητα Κινητήρων	225 rpm (περιορισμένη από το λογισμικό)

#### Χαρακτηριστικά Κινητήρων (3)

Τάση λειτουργίας	3-6V
Απώλεια λειτουργίας	3A-6V 1,5A-3V
Κατάσταση άνευ φορτίου	125mA (170mA max)
Gearbox Ratio	48:1
Ταχύτητα Περιστροφής (3V)	65 rpm

#### Pololu IR Beacon (5)

Συχνότητα IR	56 kHz
Ρυθμός Ανανέωσης	20 Hz
Εμβέλεια Ανίχνευσης	15 cm – 6 m
Απαιτούμενη Τάση (παροχή)	6-16V
Τάση εξόδου (δεδομένα)	0-5V

**Σύστημα Φωτοδιόδου – Λαμπτήρων IR (8, 9)**

Μήκος Κύματος	940nm
Γωνία θέασης	20°
Απόσταση Ανίχνευσης	10cm

**Αισθητήρας Υπερήχων HC-SR04 (7)**

Τροφοδοσία	5V
Ένταση λειτουργίας	15mA
Απόσταση Ανίχνευσης	2cm – 4m
Ανάλυση	0.3cm
Γωνία Μέτρησης	30°