



**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**

**Σχολή Τεχνολογικών Εφαρμογών - Τμήμα Μηχανικών  
Πληροφορικής**

## **Πτυχιακή εργασία**

**Ανάπτυξη Ρομποτικής Πλατφόρμας και RTOS**

**Δεληγιάννης Ιωάννης (Α.Μ. 2232)**

**Επιβλέπων Καθηγητής: Παπαδουράκης Γιώργος**

**Ημερομηνία Παρουσίασης: Ιούλιος 2014**

# Ευχαριστίες



## Abstract

Robotics is the branch of technology that deals with the design, construction, operation and application of robots and computer systems for their control, sensory feedback, and information processing. These technologies deal with automated machines that can take the place of humans, in hazardous or manufacturing processes, or simply just resemble humans.

Many of today's robots are inspired by nature contributing to the field of bio-inspired robotics. The concept in creation of machines that could operate autonomously dates back to classical times, but research into the functionality and potential uses of robots did not grow substantially until the 20th century.

Throughout history, robotics has been often seen to mimic human behavior, and often manage tasks in a similar fashion. Today, robotics is a rapidly growing field, as we continue to research, design, and build new robots that serve various practical purposes, whether domestically, commercially, or militarily. Many robots do jobs that are hazardous to people such as defusing bombs, exploring shipwrecks, and mines.

Though a significant percentage of robots in commission today are either human controlled, or operate in a static environment, there is an increasing interest in robots that can operate autonomously in a dynamic environment. These robots require some combination of navigation hardware and software in order to traverse their environment.

This document refers to the study and implementation of a flexible robotic system according to the needs in remote areas [areas where it is impossible to move a man without the use of protective equipment]. It would be appropriate to make a reference to the previous model [ i025 ] which was constructed in order to determine the most suitable operating system as well the procedure sharing algorithm analogous to the priority and the time for completion.

Inaccessible areas, such as nuclear power plants to specific sites and areas affected by earthquakes, often necessary autonomous intelligent systems in order to solve specific or abstract tasks. This problem occurs when the risk is high enough for a human in order to solve it. Although this is a prototype robot [proof of concept] it may not be able to handle such situations.

Our first goal is to create a stable real time Operating System in both microcontrollers in order to start the development of independent applications that could handle each sensor, process the results of them and create a reaction of the robot. According to our needs we should collect as much information as possible about the external environment in order to have the most suitable reaction and of course to eliminate the possibility of damage that could happened in case of wrong decision.

## Σύνοψη

Η παρούσα πτυχιακή αναφέρεται στη μελέτη και στην υλοποίηση ενός ευέλικτου ρομποτικού συστήματος καθώς και στην εφαρμογή ενός υβριδικού λειτουργικού συστήματος στηριζόμενο στις αρχές των Real-Time Embedded Operating Systems. Βασικός στόχος είναι η σύνθεση των παραπάνω και η σωστή λειτουργία τους.

Θα ήταν σκόπιμο να γίνει μια αναφορά στο προηγούμενο μοντέλο [i025], το οποίο κατασκευάστηκε με σκοπό να καθορίσει το κατάλληλο λειτουργικό σύστημα, καθώς και τον αλγόριθμο καταμερισμού διαδικασιών ανάλογα με την προτεραιότητα και το χρόνο ολοκλήρωσής τους.

Στο σύνολο της μελέτης θα παρουσιαστούν διαδοχικά τα βήματα επιλογής υλικού καθώς και κατασκευής της ρομποτικής πλατφόρμας όπως επίσης και οι αλγόριθμοι που χρησιμοποιήθηκαν για το λειτουργικό σύστημα ,την συλλογή δεδομένων και τη κίνηση.

Σκοπός της εργασίας είναι η ανάδειξη τέτοιων εφαρμογών με στόχο την χρήση τους σε περιοχές δυσπρόσιτες για τον άνθρωπο όπως σε ατυχήματα πυρηνικών εργοστασίων όπου υπάρχει διαρροή ραδιενέργειας ή περιοχές που επλήγησαν από σεισμούς. Σε τέτοιες περιπτώσεις πάντα είναι χρήσιμη η μετάβαση αυτόνομων κινητών συστημάτων για την εύρεση και την επίλυση προβλημάτων ή ακόμα και για την διαπίστωση κάποιων ήδη υπαρχουσών καταστάσεων.

## Πίνακας Περιεχομένων

|  |    |
|--|----|
| 1. Εισαγωγή.....   | 1  |
| 1.1 Η Πιθανοτική Ρομποτική.....                          | 2  |
| 1.4 Δομή Εργασίας.....                                   | 4  |
| 2. Μεθοδολογία.....                                      | 5  |
| 2.1 Ανάλυση προβλήματος.....                             | 5  |
| 3. Μηχανολογική - Ηλεκτρονική Μελέτη και Υλοποίηση.....  | 7  |
| 3.1 Εξωτερική κατασκευή.....                             | 7  |
| 3.2 Βηματικός Κινητήρας - Servo.....                     | 11 |
| 3.3 Ηλεκτρονικά Στοιχεία.....                            | 13 |
| 3.4 Ανάλυση τροφοδοσίας ενέργειας.....                   | 15 |
| 3.6 Αισθητήρες και ολοκληρωμένα κυκλώματα.....           | 16 |
| 3.6.1 Μικροεπεξεργαστής.....                             | 18 |
| 3.6.2 Μικροεπεξεργαστής χειρισμού κινητήρων.....         | 21 |
| 3.6.3 Αισθητήρας Απόστασης.....                          | 25 |
| 3.6.4 Αισθητήρας μαγνητικών πεδίων.....                  | 29 |
| 3.6.5 Επιταχυνσιόμετρο.....                              | 30 |
| 3.6.6 Αισθητήρας ανίχνευσης Διοξειδίου του Άνθρακα.....  | 32 |
| 3.6.7 Παγκόσμιο Σύστημα Εντοπισμού Θέσης.....            | 33 |
| 3.6.8 Συσκευή Επικοινωνίας.....                          | 35 |
| 3.6.9 Συσκευή Αποθήκευσης σε micro SD.....               | 37 |
| 3.6.10 Συνολικό Κύκλωμα.....                             | 39 |
| 3.8 Αποτέλεσμα Μηχανολογικής – Ηλεκτρονικής μελέτης..... | 40 |
| 4. Κινηματική Μελέτη Ρομποτικού Συστήματος.....          | 43 |
| 4.1 Αλγόριθμος Denavit-Hartenberg.....                   | 46 |
| 4.2 Προσανατολισμός τελικής Θέσης.....                   | 47 |
| 4.3 Παρουσίαση κίνησης.....                              | 48 |
| 5. Ανάλυση Λογισμικού.....                               | 55 |
| 5.1 Λογισμικό Ρομποτικής Πλατφόρμας.....                 | 56 |
| 5.1.1 Λειτουργικό Σύστημα.....                           | 58 |
| 5.1.1.1 Διεργασίες.....                                  | 60 |
| 5.1.1.2 Αμοιβαίος αποκλεισμός (Mutual exclusion).....    | 62 |
| 5.1.3 Εφαρμογές.....                                     | 65 |
| 5.2 Λογισμικό Επικοινωνίας.....                          | 67 |
| 5.2.1 Περισσότερες Πληροφορίες.....                      | 67 |

|   |    |
|---|----|
| 5.2.2 Επεξήγηση Διεπαφής της εφαρμογής .....      | 68 |
| 5.2.3 Χρήση της Εφαρμογής i026 Control Panel..... | 69 |
| 6. Αποτελέσματα.....                              | 72 |
| 6.1 Παρουσίαση Δείγματος Λειτουργίας.....         | 72 |
| 6.2 Συμπεράσματα .....                            | 76 |
| 6.3 Μελλοντικές Εργασίες και Επεκτάσεις .....     | 76 |
| 7. Βιβλιογραφία .....                             | 77 |

## Πίνακας Εικόνων

|  |    |
|--|----|
| Εικόνα 1 : Πιθανοτική Ρομποτική .....                                | 2  |
| Εικόνα 2 : Πρότυπο Σχέδιο AutoCAD (1).....                           | 7  |
| Εικόνα 3 : Πρότυπο Σχέδιο AutoCAD (2).....                           | 8  |
| Εικόνα 4 : Πρότυπο Σχέδιο AutoCAD (3).....                           | 8  |
| Εικόνα 5 : Σχέδια Κοπής Βάσης.....                                   | 9  |
| Εικόνα 6 : Μέρη Αλουμινίου (σώμα) .....                              | 10 |
| Εικόνα 7 : Μέρη Αλουμινίου(πόδι) .....                               | 10 |
| Εικόνα 8 : Σερβοκινητήρες .....                                      | 11 |
| Εικόνα 9 : Σερβοκινητήρες (Σχέδιο - Σύνδεση) .....                   | 12 |
| Εικόνα 10 : Κλίση του σερβοκινητήρα.....                             | 13 |
| Εικόνα 11 : Μπαταρία Li-Poli 2200mAh .....                           | 14 |
| Εικόνα 12 : Πυκνωτής .....   | 15 |
| Εικόνα 13 : Πυκνωτής (σύμβολο στο λογικό διάγραμμα).....             | 15 |
| Εικόνα 14 : Σταθεροποιητής τάσης .....                               | 15 |
| Εικόνα 15 : Σταθεροποιητής τάσης (σύμβολο στο λογικό διάγραμμα)..... | 15 |
| Εικόνα 16 : Teensy 3.0 (πλευρά 1) .....                              | 20 |
| Εικόνα 17 : Teensy 3.0 (πλευρά 2) .....                              | 20 |
| Εικόνα 18 : Σήματα PWM .....   | 21 |
| Εικόνα 19 : Servocontroller Pololu Mini Maestro 12.....              | 22 |
| Εικόνα 20 : Servocontroller Pololu Mini Maestro 12 (Συνδέσεις) ..... | 23 |
| Εικόνα 21 : Συχνότητες ήχου.....                                     | 25 |
| Εικόνα 22 : Ultrasonic SRF05 .....                                   | 25 |
| Εικόνα 23 : Αρχή λειτουργίας SONAR σχηματικά.....                    | 26 |
| Εικόνα 24 : SRF05 Συνδέσεις.....                                     | 27 |
| Εικόνα 25 : SRF05 Χρονισμός .....                                    | 28 |
| Εικόνα 26 : SRF05 Δέσμη Υπερήχων .....                               | 28 |
| Εικόνα 27 : Μαγνητόμετρο HMC6352.....                                | 29 |
| Εικόνα 28 : ADXL 335.....  | 31 |
| Εικόνα 29 : Gas Sensor .....   | 32 |
| Εικόνα 30 : GPS EM406α.....  | 33 |
| Εικόνα 31 : Xbee Pro 900 .....                                       | 36 |
| Εικόνα 32 : Xbee Breakout Board .....                                | 36 |
| Εικόνα 33 : XBee Breakout Board (USB) .....                          | 36 |

|   |    |
|---|----|
| Εικόνα 34 : Συσκευή καταγραφής OpenLog .....                          | 37 |
| Εικόνα 35 : Ηλεκτρονικό Κύκλωμα (1).....                              | 39 |
| Εικόνα 36 : Ηλεκτρονικό Κύκλωμα (2).....                              | 39 |
| Εικόνα 37 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (1) ..... | 40 |
| Εικόνα 38 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (2) ..... | 40 |
| Εικόνα 39 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (3) ..... | 41 |
| Εικόνα 40 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (4) ..... | 41 |
| Εικόνα 41 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (5) ..... | 42 |
| Εικόνα 42 : Κινηματική μελέτη βραχίονα .....                          | 43 |
| Εικόνα 43 : Γεωμετρική άποψη μεγεθών D-H .....                        | 44 |
| Εικόνα 44 : Προσδιορισμός Άκρου Βραχίονα.....                         | 47 |
| Εικόνα 45 : Timelapse Κινήσεων (1).....                               | 49 |
| Εικόνα 46 : Timelapse Κινήσεων (2).....                               | 50 |
| Εικόνα 47 : Timelapse Κινήσεων (3).....                               | 51 |
| Εικόνα 48 : Timelapse Κινήσεων (4).....                               | 52 |
| Εικόνα 49 : Timelapse Κινήσεων (5).....                               | 53 |
| Εικόνα 50 : Timelapse Κινήσεων (6).....                               | 54 |
| Εικόνα 51 : Καταμερισμός Χρόνου.....                                  | 60 |
| Εικόνα 52 : Παράδειγμα συνδεδεμένης λίστας.....                       | 62 |
| Εικόνα 53 : i026 Control Panel.....                                   | 68 |
| Εικόνα 54 : i026 Control Panel Δείγμα Χρήσης (1) .....                | 69 |
| Εικόνα 55 : i026 Control Panel Δείγμα Χρήσης (2) .....                | 69 |
| Εικόνα 56: i026 Control Panel Δείγμα Χρήσης (3) .....                 | 70 |
| Εικόνα 57 : i026 Control Panel Δείγμα Χρήσης (4) .....                | 71 |
| Εικόνα 58 : Φωτογραφία Ρομποτ (1).....                                | 72 |
| Εικόνα 59 : Φωτογραφία Ρομποτ (2).....                                | 73 |
| Εικόνα 60 : Φωτογραφία Ρομπότ (3).....                                | 74 |
| Εικόνα 61 : Φωτογραφία Ρομπότ (4).....                                | 74 |
| Εικόνα 62 : Φωτογραφία Ρομποτ (5).....                                | 74 |



## Παράρτημα Πινάκων

|  |    |
|--|----|
| Πίνακας 1 : Βήματα Μελέτης και Υλοποίησης .....                  | 5  |
| Πίνακας 2 : Χαρακτηριστικά μπαταρίας 7.4V Lipo 2200mAh .....     | 14 |
| Πίνακας 3 : Διαστάσεις Pololu Servo Controller .....             | 22 |
| Πίνακας 4 : Χαρακτηριστικά Pololu Servo Controller .....         | 23 |
| Πίνακας 5 : Χαρακτηριστικά HMC5883L .....                        | 29 |
| Πίνακας 6 : Διαστάσεις HMC5883L .....                            | 29 |
| Πίνακας 7 : Εφαρμογές του ADXL335 .....                          | 31 |
| Πίνακας 8 : Κανόνες Επιπέδου Εφαρμογής (NMEA) .....              | 34 |
| Πίνακας 9 : Επεξήγηση παραδείγματος NMEA πρωτοκόλλου .....       | 35 |
| Πίνακας 10 : Συνδέσεις EM406a .....                              | 35 |
| Πίνακας 11 : Χαρακτηριστικά XBee Pro 900 .....                   | 36 |
| Πίνακας 12 : Χαρακτηριστικά OpenLog .....                        | 37 |
| Πίνακας 13 : Διαστάσει OpenLog .....                             | 37 |
| Πίνακας 14 : Βασικές Εντολές OpenLog .....                       | 38 |
| Πίνακας 15 : Ερμηνία μεγεθών D-H .....                           | 45 |
| Πίνακας 16 : Αρχεία πηγαίου κώδικα Λειτουργικού Συστήματος ..... | 56 |
| Πίνακας 17 : Αρχεία πηγαίου κώδικα Οδηγών Συσκευών .....         | 57 |
| Πίνακας 18 : Αρχεία πηγαίου κώδικα Εφαρμογών .....               | 57 |

## Πίνακας Πηγαίου Κώδικα – Αλγορίθμων

|  |    |
|--|----|
| Κώδικας 1 : Παράδειγμα επικοινωνίας Servocontroller (Compact)..... | 24 |
| Κώδικας 2 : Παράδειγμα NMEA μηνύματος .....                        | 34 |
| Αλγόριθμος 3 : Λογικά βήματα αλγορίθμου Denavit - Hartenberg ..... | 46 |
| Αλγόριθμος 4 : Λογικά βήματα αλγορίθμου Raymond.....               | 63 |

# 1. Εισαγωγή

Η απαλλαγή από μονότονες και χειρωνακτικές εργασίες αποτελούσε πάντα ανθρώπινη επιθυμία. Στον αιώνα μας, η δυνατότητα πραγματοποίησης ενός τέτοιου στόχου άρχισε να φαίνεται εφικτή με την ανάπτυξη των αυτοματισμών και ειδικότερα της ρομποτικής. Σύμφωνα με το Ινστιτούτο Ρομποτικής της Αμερικής, ρομπότ είναι ένας αναπρογραμματιζόμενος και πολυλειτουργικός χωρικός μηχανισμός σχεδιασμένος να μετακινεί υλικά, αντικείμενα, εργαλεία ή εξειδικευμένες συσκευές με κατάλληλες μεταβλητά προγραμματιζόμενες κινήσεις, που στοχεύουν στη βελτίωση της απόδοσης μιας σειράς εργασιών.

Ο όρος ρομπότ πρωτοεμφανίζεται σε ένα θεατρικό έργο επιστημονικής φαντασίας του Τσέχου συγγραφέα Κάρελ Τσάπεκ το 1921 και προέρχεται από τη σλαβική λέξη *robota* που σημαίνει εργάτης. Η αυτοματοποίηση της παραγωγικής διαδικασίας στις βιομηχανίες μαζικής παραγωγής αντικαθιστά τους ανθρώπους με εξειδικευμένες μηχανές που εκτελούν μια προκαθορισμένη σειρά εργασιών στα προϊόντα που παράγονται.

Η έρευνα στην περιοχή της ρομποτικής εκτείνεται κυρίως σε τρεις κατευθύνσεις. Η πρώτη, αφορά την εφαρμογή ή/και την ανάπτυξη τεχνικών ελέγχου για τη βελτίωση της απόδοσης των ρομπότ. Η δεύτερη, σχετίζεται με την εφαρμογή και την ανάπτυξη λογισμικού για τη διαχείριση των εργασιών των ρομπότ. Η τρίτη συνδέεται με τη σχεδίαση υλικού υπολογιστών για την επικοινωνία με έναν κόμβο μεγάλης επεξεργαστικής ικανότητας και την παρουσία συλλογών δεδομένων από τους αισθητήρες των ρομπότ. Είναι γεγονός πως η ρομποτική ωφελείται από τις εξελίξεις βασικών κλάδων τεχνολογίας και θετικών επιστήμων όπως η ηλεκτρολογία, η ηλεκτρονική, η μηχανολογία, τα μαθηματικά και επιστήμες με θέμα την ανάπτυξη λογισμικού.

Η έρευνα του τομέα της Ρομποτικής παρουσιάζει τα τελευταία χρόνια εξαιρετική ανάπτυξη. Ειδικότερα, είναι σημαντικό να αναφερθούν οι ερευνητικές δραστηριότητες με στόχο την δημιουργία εξελιγμένων αλγορίθμων ελέγχου, καθώς και οι δραστηριότητες που σχετίζονται με την εξέλιξη συστημάτων Τεχνητής Νοημοσύνης, όπως οι ειδικοί αλγόριθμοι *fuzzy logic* που χρησιμοποιούνται για την σύνθεση των δεδομένων από τους αισθητήρες ενός ρομπότ.

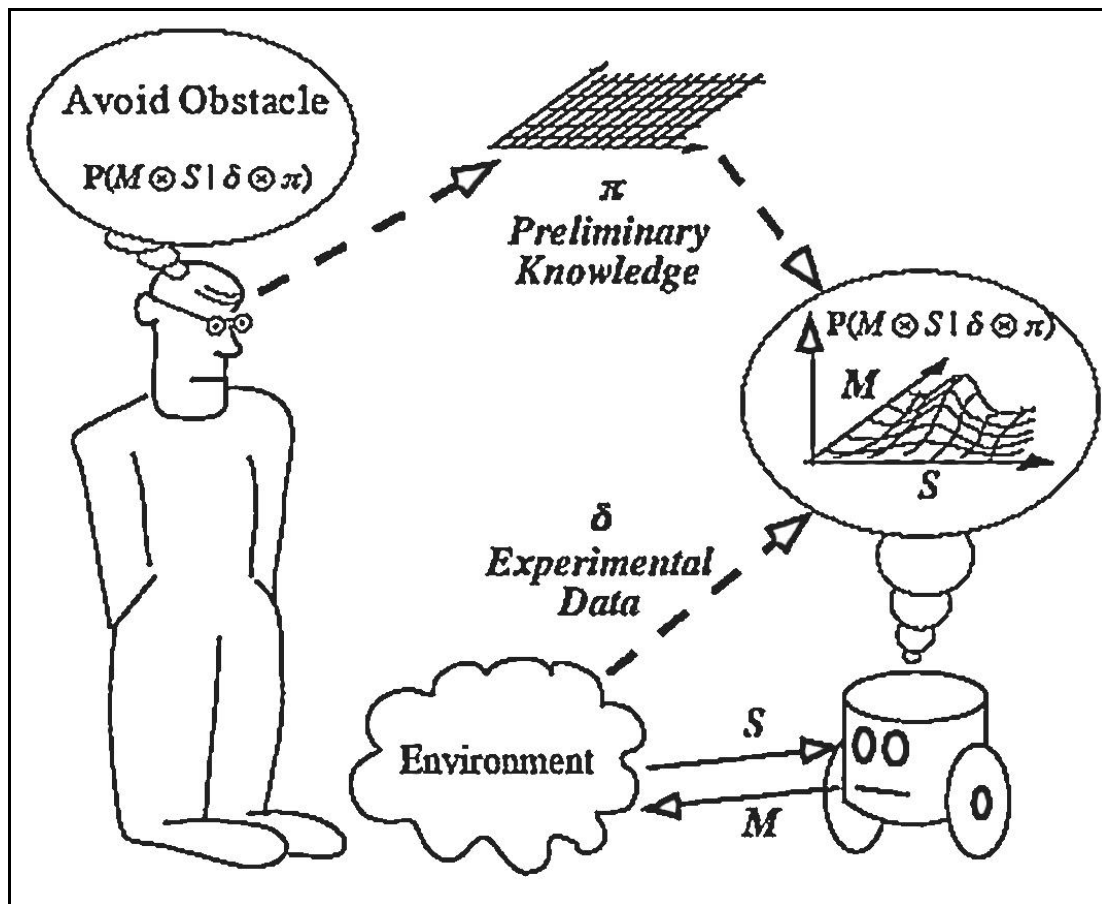
Ένας πολύ σημαντικός κλάδος που αναπτύσσεται τα τελευταία χρόνια είναι τα αυτόνομα ρομποτικά συστήματα εξερεύνησης και συλλογής δεδομένων χωρίς την επίβλεψη του ανθρώπου ακόμα και σε περιοχές δυσπρόσιτες για αυτόν.

Το πλαίσιο της εργασίας θα ασχοληθεί με την ανάπτυξη και υλοποίηση του βασικού μέρους ενός τέτοιου συστήματος με σκοπό την ανάδειξη της χρησιμότητας μιας τέτοιας πλατφόρμας στην καθημερινή ζωή του ανθρώπου αλλά και την χρηστικότητα που μπορεί να προσφέρει ακόμα και σε περιπτώσεις ανάγκης, όπως σεισμοί κτλ.

## 1.1 Η Πιθανοτική Ρομποτική

Η πιθανοτική ρομποτική αποτελεί μια υποπεριοχή της ρομποτικής που αφορά την αντίληψη και τον έλεγχο. Στηρίζεται σε στατιστικές τεχνικές για την αναπαράσταση των πληροφοριών και τη λήψη αποφάσεων. Με τον τρόπο αυτό, λαμβάνει υπόψη την αβεβαιότητα που προκύπτει στις περισσότερες σύγχρονες εφαρμογές της ρομποτικής.

Τα τελευταία χρόνια, οι πιθανοτικές τεχνικές έχουν γίνει ένα από τα κυρίαρχα υποδείγματα για τη σχεδίαση αλγορίθμων στη ρομποτική. Στην παρούσα εργασία θα προσπαθήσουμε να δημιουργήσουμε ένα τέτοιο σύστημα ώστε να μπορεί να υποστηριχθεί η αλγοριθμική δυσκολία που προκύπτει από τους αλγόριθμους που σχετίζονται με τις πιθανότητες. Πιο απλά θα δημιουργήσουμε ένα ρομπότ το οποίο θα μπορέσει να υποστηρίξει τον κανόνα του Bayes και την χρονική επέκταση του που είναι γνωστή ως φίλτρα Bayes. Αυτός ο μαθηματικός σκελετός είναι ενιαίος και αποτελεί τον κοινό πυρήνα των πιθανοτικών αλγορίθμων.



Εικόνα 1 : Πιθανοτική Ρομποτική

Λαμβάνοντας υπόψιν πως η ρομποτική είναι μια επιστήμη που ασχολείται με την αντίληψη και τον χειρισμό του φυσικού κόσμου μέσω συσκευών που ελέγχονται από υπολογιστές, μπορούμε να κατανοήσουμε την σημαντικότητα της αβεβαιότητας ως δεδομένο στους αλγορίθμους απόφασης άρα και την σπουδαιότητα της χρήσης τέτοιων αλγορίθμων σε ρομπότ εξερευνητές.

## 1.2 Περίληψη Εργασίας

Η παρούσα πτυχιακή εργασία έχει ως σκοπό, την μελέτη και υλοποίηση ενός κινητού ρομποτικού συστήματος από την αρχή, χρησιμοποιώντας πολλές αρχές της μηχανολογίας, της ηλεκτρολογίας και ειδικότερα του προγραμματισμού ενσωματωμένων υπολογιστικών συστημάτων.

Θα μπορούσαμε να διακρίνουμε την εργασία σε 3 μέρη:

**Το κατασκευαστικό μέρος** που αποτελεί την μελέτη ,ανάλυση και δημιουργία των μηχανικών μερών του ρομπότ, καθώς και των ηλεκτρικών – ηλεκτρονικών εξαρτημάτων. Ο κεντρικός κορμός του ρομπότ, δημιουργήθηκε με την χρήση μιας βασικής αρχής της μηχανολογίας όπου προβάλλεται η συμμετρία σε 2 άξονες. Το υλικό που χρησιμοποιήθηκε είναι το αλουμίνιο λόγω των σημαντικών ιδιοτήτων που προσφέρει ως μέταλλο. Τέλος, χρησιμοποιώντας ηλεκτρονικά εξαρτήματα, όπως μικροελεγκτές, TLL chips, πυκνωτές ,αντιστάσεις κλπ. δημιουργήθηκε η μητρική πλακέτα και γενικότερα η υπολογιστική μονάδα του ρομπότ.

Το δεύτερο μέρος είναι **το λειτουργικό κομμάτι** που αποτελεί την μελέτη ,ανάλυση και δημιουργία του λειτουργικού συστήματος καθώς και των εφαρμογών λειτουργίας του ρομπότ. Χρησιμοποιώντας τις γλώσσες προγραμματισμού C και C++ δημιουργήθηκε το λειτουργικό σύστημα και οι εφαρμογές, τα οποία αποτελούν αναπόσπαστο κομμάτι του ρομπότ. Και με την χρήση της C# δημιουργήθηκε η εφαρμογή υπολογιστή για την συλλογή δεδομένων από το ρομπότ.

Τέλος, **το κινηματικό μέρος** αφορά την ανάλυση και δημιουργία διαφόρων απλών αλγορίθμων κίνησης. Με ένα ολοκληρωμένο κινηματικό μοντέλο αποσπάστηκαν σημαντικά δεδομένα τα οποία χρησιμοποιήθηκαν για την δημιουργία αλγοριθμικών βημάτων επιδιώκοντας την κίνηση της ρομποτικής πλατφόρμας.

## 1.3 Σκοπός και Στόχοι Εργασίας

Στην σημερινή εποχή ο κόσμος μας περικλύεται από ρομπότ. Η χρήση τους είναι τόσο εκτενής που τα συναντούμε καθημερινά χωρίς πολλές φορές να τα αναγνωρίζουμε. Ωστόσο ακόμα δεν υπάρχουν αρκετά ανεπτυγμένες εφαρμογές ρομποτικής για την εξερεύνηση απροσπέλαστων περιοχών.

Με αυτή την σκέψη η έρευνα πάνω στον εν λόγω τομέα αποκτά ιδιαίτερο ενδιαφέρον. Αν και η δημιουργία μιας τέτοιας πλατφόρμας θέλει χρόνια δουλειάς και το κόστος είναι ιδιαίτερα περιοριστικό ,στην παρούσα πτυχιακή εργασία θα μελετηθούν τα βήματα για την υλοποίηση αυτής της ιδέας.

Βασικός σκοπός της εργασίας είναι η παρουσίαση βήμα προς βήμα της κατασκευής ενός τετράποδου ρομπότ. Την μηχανολογική μελέτη για την κατασκευή μιας βάσης φιλοξενίας της υπολογιστικής μονάδας, την επιλογή ηλεκτρονικών εξαρτημάτων για την δημιουργία του

«εγκεφάλου» ,τον προγραμματισμό ενός real-time λειτουργικού συστήματος εφαρμογών ελέγχου ,συντονισμού και συλλογής δεδομένων καθώς και μια εφαρμογής επικοινωνίας με τον ηλεκτρονικό υπολογιστή για την ανάλυση των δεδομένων.

## 1.4 Δομή Εργασίας

### **Κεφάλαιο 2 : Μεθοδολογία**

Στο κεφάλαιο αυτό αναφέρονται αναλυτικά το θέμα της πτυχιακής και οι τεχνολογίες που χρησιμοποιήθηκαν καθώς και οι διάφοροι μέθοδοι που ακολουθήθηκαν για την υλοποίηση της ρομποτικής πλατφόρμας ,του λειτουργικού συστήματος ,των εφαρμογών συλλογής δεδομένων και την εφαρμογή διεπαφής με σκοπό την προβολή δεδομένων και επικοινωνίας με το ρομπότ.

[ σελίδες 5-6 ]

### **Κεφάλαιο 3 : Μηχανολογική - Ηλεκτρονική Μελέτη και Υλοποίηση**

Παρουσιάζεται η μελέτη και η ανάλυση της μηχανολογικής κατασκευής, η επιλογή υλικού όπως επίσης αναλυτικές πληροφορίες για τα ηλεκτρονικά εξαρτήματα που χρησιμοποιήθηκαν καθώς και η μελέτη - δημιουργία του ηλεκτρονικού κυκλώματος.

[ σελίδες 7-42 ]

### **Κεφάλαιο 4 : Κινηματική Μελέτη Ρομποτικού Συστήματος**

Αναφέρεται η κινηματική μελέτη της ρομποτικής πλατφόρμας με πληροφορίες για τον αλγόριθμο Denavit – Hartenberg, τον τρόπο εύρεσης τελικής θέσης και τις βασικές θέσεις του ρομπότ.

[ σελίδες 43-54 ]

### **Κεφάλαιο 5 : Ανάλυση Λογισμικού**

Στην ανάλυση λογισμικού μελετώνται, το λειτουργικό σύστημα του ρομπότ ,οι εφαρμογές ελέγχου και συλλογής δεδομένων καθώς και η εφαρμογή προβολής δεδομένων στον υπολογιστή του χρήστη. Επίσης παρουσιάζονται τμήματα πηγαίου κώδικα και πρότυπα εφαρμογών – οδηγών συσκευών.

[ σελίδες 55-70 ]

### **Κεφάλαιο 6 : Αποτελέσματα**

Στο τελευταίο αυτό κεφάλαιο παραθέτουμε τα συμπεράσματα από τη διαδικασία υλοποίησης της πτυχιακής εργασίας για τη χρησιμότητά τους σε μελλοντικές εργασίες.

[ σελίδες 71-77 ]

## 2. Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε είναι γνωστή ως trial and error χρησιμοποιώντας πάντα κάποιες βασικές αρχές από διάφορους τομείς της τεχνολογίας όπως η μηχανολογία, η ηλεκτρολογία και κυρίως του προγραμματισμού ηλεκτρονικών κυκλωμάτων και υπολογιστών.

Ο παρακάτω πίνακας παρουσιάζει τα βήματα μελέτης και υλοποίησης :

| Βήματα Μελέτης και Υλοποίησης     |                 |   |   |   |   |   |   |  |   |   |   |   |  |   |   |
|-----------------------------------|-----------------|---|---|---|---|---|---|--|---|---|---|---|--|---|---|
| Εργασία                           | Σειρά Ενεργειών |   |   |   |   |   |   |  |   |   |   |   |  |   |   |
| Ανάλυση Συνολικής Εργασίας        | ■               |   |   |   |   |   |   |  |   |   |   |   |  |   |   |
| Μελέτη Αποτελεσμάτων              |                 | ■ |   |   |   |   |   |  |   |   |   |   |  | ■ | ■ |
| Εύρεση Υλικών – Εξαρτημάτων       |                 |   | ■ | ■ |   |   | ■ |  |   | ■ |   |   |  |   |   |
| Μελέτη Υλικών – Εξαρτημάτων       |                 |   |   | ■ |   |   |   |  |   |   |   |   |  |   |   |
| Υλοποίηση Μηχανικών Μερών         |                 |   |   |   | ■ | ■ |   |  |   |   |   |   |  |   |   |
| Έλεγχος Μηχανικών Μερών           |                 |   |   |   | ■ | ■ |   |  |   |   |   |   |  |   |   |
| Υλοποίηση Ηλεκτρονικού Κυκλώματος |                 |   |   |   |   |   |   |  | ■ |   |   |   |  |   |   |
| Έλεγχος Ηλεκτρονικού Κυκλώματος   |                 |   |   |   |   |   |   |  | ■ | ■ |   |   |  |   |   |
| Υλοποίηση Λογισμικού              |                 |   |   |   |   |   |   |  |   |   | ■ |   |  |   |   |
| Αποσφαλμάτωση Λογισμικού          |                 |   |   |   |   |   |   |  |   |   | ■ | ■ |  | ■ | ■ |
| Αποσφαλμάτωση Συνόλου             |                 |   |   |   |   |   |   |  |   |   |   | ■ |  | ■ | ■ |

Πίνακας 1 : Βήματα Μελέτης και Υλοποίησης

Η θεωρία που χρησιμοποιήθηκε κυρίως σε αυτή την εργασία και μας βοήθησε και στην κατεύθυνση που πήρε η δημιουργία της αφορά κυρίως την ρομποτική και τα ενσωματωμένα λειτουργικά συστήματα. Τον ορισμό τους, τις κατηγορίες και δυνατότητές τους καθώς και τα πλεονεκτήματα και μειονεκτήματα που προκύπτουν από τη χρήση τους.

### 2.1 Ανάλυση προβλήματος

Για την μελέτη και την ανάλυση του συνόλου της πτυχιακής εργασίας προαπαιτηθηκε ο χωρισμός της σε τμήματα έτσι ώστε ο καθορισμός των εργασιών που έπρεπε να γίνουν και η αναζήτηση πληροφοριών για κάθε σημείο της εργασίας να επιτευχθεί χωρίς προβλήματα.

## **Κατασκευή**

Στην κατασκευή την ρομποτικής πλατφόρμας προσπαθήσαμε να πετύχουμε την συμμετρία και το μειωμένο βάρος. Ο λόγος είναι η καλύτερη λειτουργία και η μικρότερη κατανάλωση ρεύματος. Χρησιμοποιώντας διάφορα υλικά και σχηματισμούς αποφασίστηκε τελικά η καλύτερη δυνατή λύση και σαφώς εφικτή για υλοποίηση.

Η μελέτη και η δημιουργία του ηλεκτρονικού κυκλώματος ήταν ένα στάδιο όπου έπρεπε να συνυπολογιστούν πολλές παράμετροι όπως ο διαθέσιμος χώρος και η λειτουργικότητα των αισθητήρων.

## **Λογισμικό**

Στο λογισμικό οι παράμετροι που ληφθήκαν υπόψιν ήταν η ταχύτητα και η σταθερότητα. Όντας ένα σύνολο εφαρμογών από την αρχή ήταν αναγκαία η σοβαρή μελέτη των λειτουργικών συστημάτων. Εκτός από τον προγραμματισμό της ρομποτικής πλατφόρμας ήταν αναγκαία η δημιουργία και μιας εφαρμογής υπολογιστή για την επικοινωνία και την αποσφαλμάτωση διαφόρων λειτουργιών του ρομπότ.

## **Κινηματική**

Η κινηματική μελέτη βασίστηκε στον αλγόριθμο Denavit – Hartenberg. Τα αποτελέσματα του αλγόριθμου αυτού χρησιμοποιήθηκαν για την δημιουργία συλλογών κίνησης οι οποίες τοποθετήθηκαν σε μία εφαρμογή που εκτελείται στο ρομπότ.



### 3. Μηχανολογική - Ηλεκτρονική Μελέτη και Υλοποίηση

Σε αυτό το κεφάλαιο θα δούμε αναλυτικά τα βήματα υλοποίησης της πτυχιακής εργασίας σε κατηγορίες ,παρέχοντας πληροφορίες για την κατασκευή και τα εξαρτήματα, καθώς και τον τρόπο λειτουργίας του λειτουργικού συστήματος που χρησιμοποιείται στο συγκεκριμένο πρότυπο ρομπότ.

#### 3.1 Εξωτερική κατασκευή<sup>1</sup>

Η εξωτερική κατασκευή, δηλαδή ο «σκελετός» της ρομποτικής πλατφόρμας, δημιουργήθηκε με την χρήση φύλλων αλουμινίου. Το συγκεκριμένο υλικό επιλέχθηκε λόγω της ανθεκτικότητας του βάρους και γενικά των ιδιοτήτων του οι οποίες υπερέχουν σε σύγκριση με τα υπόλοιπα γνωστά μέταλλα. Επίσης ένα ακόμα σημαντικό κριτήριο επιλογής ήταν η εύκολη επεξεργασία του με απλά εργαλεία.

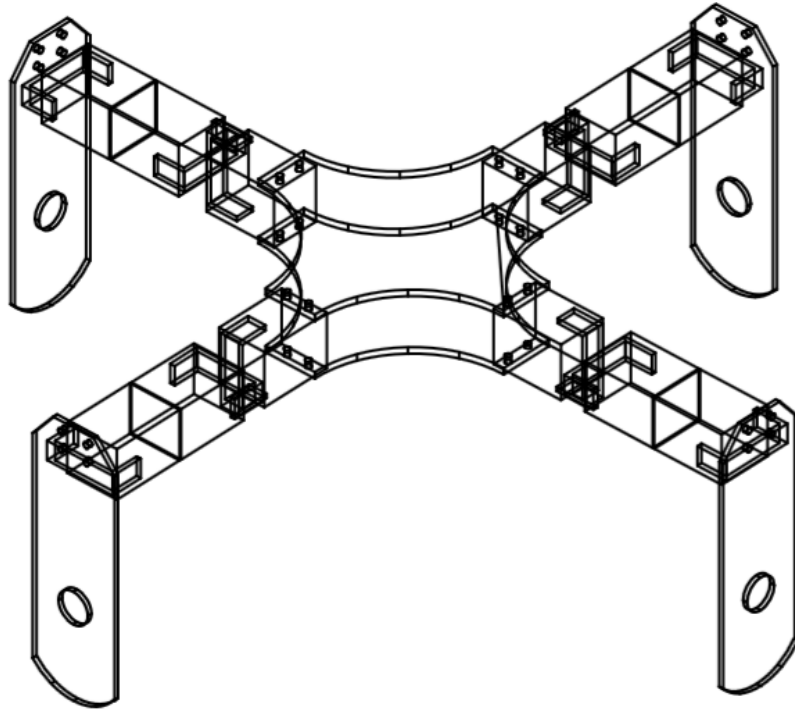
##### Πρότυπο Σχέδιο AutoCAD

Πριν προχωρήσουμε στα σχέδια κοπής, ας δούμε το πρότυπο έτσι όπως σχεδιάστηκε στο Autodesk AutoCAD.

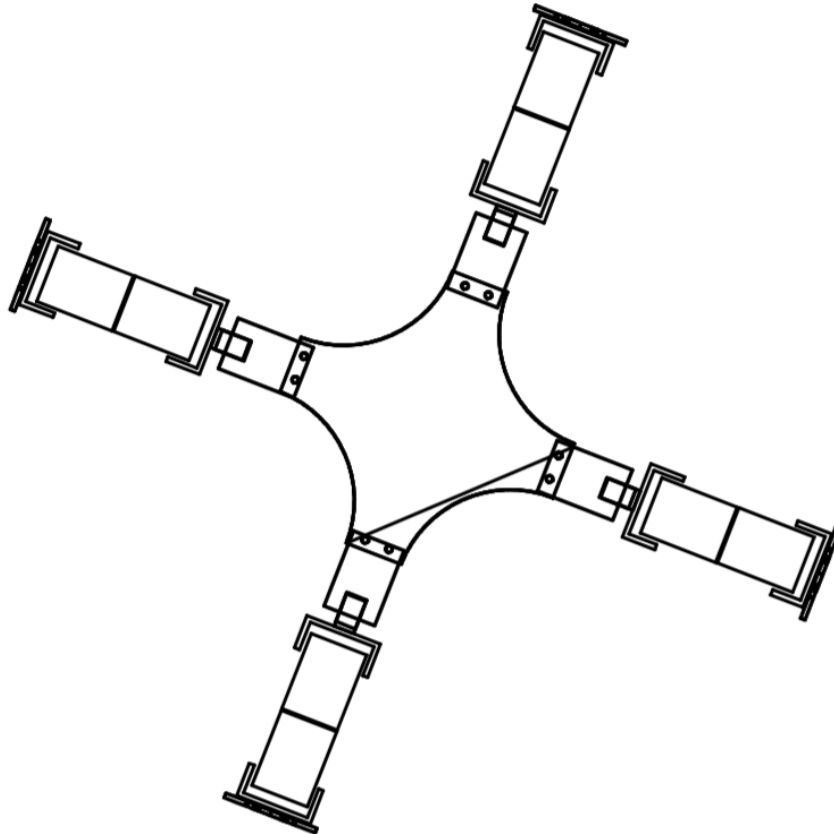


Εικόνα 2 : Πρότυπο Σχέδιο AutoCAD (1)

<sup>1</sup> Η ολοκληρωμένη κατασκευή υπάρχει σε επόμενο υποκεφάλαιο



Εικόνα 3 : Πρότυπο Σχέδιο AutoCAD (2)



Εικόνα 4 : Πρότυπο Σχέδιο AutoCAD (3)

## Σχέδια κοπής

Το πρώτο βήμα που ακολουθήσαμε ήταν η δημιουργία ενός σχεδίου για την βάση και τα «πόδια» της τετράποδης πλατφόρμας.



Εικόνα 5 : Σχέδια Κοπής Βάσης

## Τα μέρη αλουμινίου της Πλατφόρμας

Όλα τα μέρη περάστηκαν από διαδικασία απλής διάβρωσης και βαφής<sup>2</sup>. (Ιδιότητες αλουμινίου)



Εικόνα 6 : Μέρη Αλουμινίου (σώμα)



Εικόνα 7 : Μέρη Αλουμινίου(πόδι)

---

<sup>2</sup> Τα υλικά που χρησιμοποιήθηκαν δεν αναγράφονται.

## 3.2 Βηματικός Κινητήρας - Servo

Οι βηματικοί κινητήρες είναι μικρές συσκευές που έχουν έναν εξωτερικό άξονα. Αυτός ο άξονας μπορεί να μετακινηθεί σε διάφορες θέσεις αν αποσταλεί στον servo ένα κωδικοποιημένο σήμα. Όσο υπάρχει αυτό το σήμα στην γραμμή εισόδου του βηματικού κινητήρα τόσο αυτός θα διατηρεί τον άξονα του σε μια συγκεκριμένη θέση. Όταν αλλάζει το σήμα προκαλεί στο servo μεταβολή στην γωνία του άξονα του. Σε πρακτικές εφαρμογές τα servo 's χρησιμοποιούνται σε τηλεχειριζόμενα αεροπλάνα, αυτοκίνητα και στην ρομποτική. Οι κινητήρες αυτοί έχουν ένα εξαιρετικά μικρό μέγεθος αλλά είναι αρκετά ισχυροί για το μέγεθος τους.



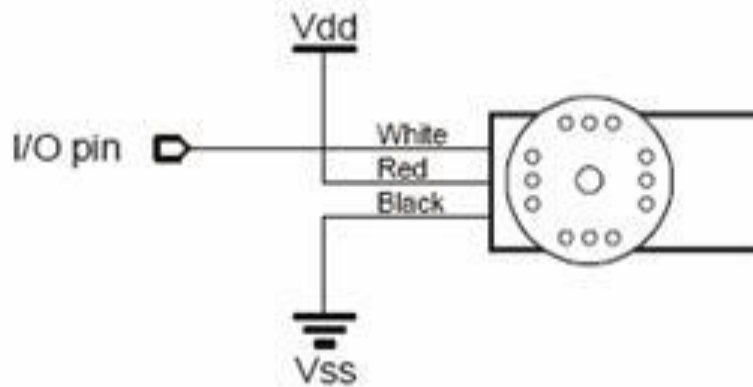
Εικόνα 8 : Σερβοκινητήρες

Ο στάνταρ βηματικός κινητήρας παρέχει κίνηση του άξονα από 0ο έως 180ο μοίρες. Δύο είναι τα βασικά μέρη ενός βηματικού κινητήρα το κύκλωμα ελέγχου και ο ηλεκτρομαγνητικός κινητήρας που μετατρέπει τα ηλεκτρικά φορτία σε μηχανική ενέργεια.

Η τάση που απαιτεί για την λειτουργία του είναι 6 Volt DC. Ο άξονας του μπορεί να μετακινηθεί από την θέση 0ο στη θέση 180° σε 1.2 sec. Αυτό ορίζεται και ως η ταχύτητα του κινητήρα. Ζυγίζει περίπου 85gr και έχει ροπή στρέψης 11.40 kg-cm.

### Διασύνδεση Servo

Για την διασύνδεση ενός απαιτούνται 1 Pin Εισόδου, μια γραμμή για την τάση Vdd και μια γραμμή για την γείωση αυτού. Αυτά φαίνονται στο σχήμα 2, όπως επίσης και τα χρώματα που αντιστοιχούν σε κάθε γραμμή (White, Red, Black).



Εικόνα 9 : Σερβοκινητήρες (Σχέδιο - Σύνδεση)

Στη γραμμή Vdd παρέχεται τάση από 5 ως 7.4 Volt και η γραμμή Vss συνδέεται με την γείωση του κυκλώματος. Η γραμμή I/O συνδέεται με κάποιο Pin ελέγχου (PWM) και ονομάζεται γραμμή ελέγχου.

### Προγραμματισμός Servo

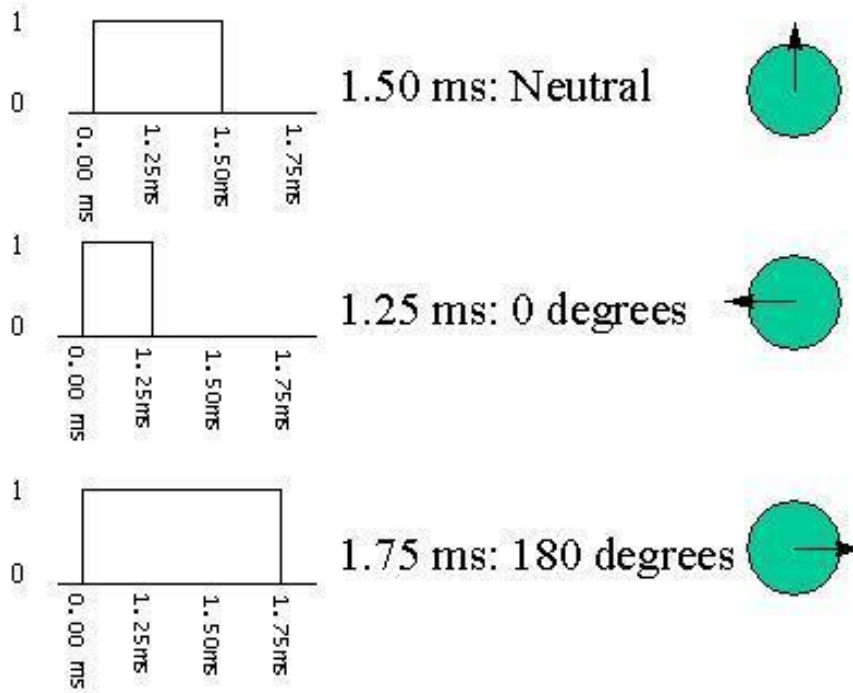
Το servo για να ενεργοποιηθεί χρειάζεται να σταλεί κάποιος παλμός στην γραμμή ελέγχου του. Το κύκλωμα ελέγχου έχει ενσωματωμένη μια μεταβλητή αντίσταση που συνδέεται με τον εξωτερικό άξονα. Αυτή η αντίσταση επιτρέπει στο κύκλωμα ελέγχου να παρακολουθεί την τρέχουσα θέση στην οποία βρίσκεται ο άξονας. Αν ο άξονας είναι στην σωστή θέση τότε ο ηλεκτρομαγνητικός κινητήρας σταματά να λειτουργεί. Αν το κύκλωμα διαπιστώσει ότι η γωνία που βρίσκεται ο άξονας δεν είναι η σωστή τότε θα στρέψει τον άξονα στην σωστή κατεύθυνση μέχρις ότου ευθυγραμμιστεί με την γωνία που πρέπει.

Ένας βηματικός κινητήρας servo δεν μπορεί να στραφεί πέρα από 0ο έως 180ο μοίρες καθώς έχει μια ειδική ασφάλεια στο κιβώτιο ταχυτήτων του που σταματάει οποιαδήποτε κίνηση πέρα από αυτό το διάστημα.

Το καλώδιο ελέγχου είναι αυτό που θα ορίσει στο κύκλωμα ελέγχου την γωνία. Η γωνία ορίζεται από την χρονική διάρκεια του παλμού που στέλνεται στο κύκλωμα ελέγχου μέσω του καλωδίου από κάποιο Pin Εισόδου / Εξόδου. Ο κινητήρας servo περιμένει να εμφανιστεί ένας παλμός κάθε 20milliseconds.

Το χρονικό πλάτος (μήκος) του παλμού θα καθορίσει πόσο θα γυρίσει ο κινητήρας. Ένας παλμός για παράδειγμα της τάξεως του 1.5 millisecond, θα αναγκάσει τον κινητήρα να στραφεί στην θέση των 90 μοιρών. Αν ο παλμός είναι μικρότερος του 1.5 millisecond τότε ο κινητήρας θα στραφεί προς την θέση των 0ο μοιρών. Αν ο παλμός που θα σταλεί από το κύκλωμα είναι μεγαλύτερος του 1.5 millisecond τότε ο κινητήρας θα στραφεί σε κάποια θέση κοντινότερα στην θέση των 180° μοιρών.

Όπως φαίνεται και στην εικόνα 4 το πλάτος των παλμών ορίζει την θέση του κινητήρα.



Εικόνα 10 : Κλίση του σερβοκινητήρα

### 3.3 Ηλεκτρονικά Στοιχεία

Το ηλεκτρονικό κύκλωμα που κατασκευάστηκε για την υλοποίηση της ρομποτικής πλατφόρμας αποτελείται από 4 μπαταρίες Ιόντων λιθίου , 2 ηλεκτρολυτικούς πυκνωτές μεγάλης χωρητικότητας, 2 ηλεκτρολυτικούς πυκνωτές μικρής χωρητικότητας και 2 σταθεροποιητές τάσης .

#### Μπαταρίες Ιόντων Λιθίου

Οι μπαταρίες ιόντων λιθίου χρησιμοποιούνται σήμερα σε ένα τεράστιο εύρος ηλεκτρικών συσκευών, από τα κινητά τηλέφωνα μέχρι τα υβριδικά οχήματα. Μπορούν να αποθηκεύουν μεγάλα ποσά ενέργειας ανά μονάδα βάρους (περιέχουν υψηλότερη πυκνότητα ενέργειας). Έτσι παρατείνεται η διάρκεια ζωής της μπαταρίας, ενώ είναι πιο ελαφριά αφού το λίθιο είναι το πιο ελαφρύ μέταλλο.

Η επαναφόρτιση των μπαταριών ιόντων λιθίου μπορεί να γίνει οποιαδήποτε στιγμή χωρίς να είναι αναγκαία η πλήρης φόρτιση, πράγμα που απαιτείται από τις άλλες μπαταρίες για καλύτερη απόδοση. Για αυτό τον λόγο εξασφαλίζουν καλύτερη απόδοση στα οχήματα και μεγάλη αυτονομία σε σύγκριση με τις μπαταρίες μολύβδου και έχουν διπλάσια αυτονομία και πενταπλάσια διάρκεια ζωής. Ο όρος «μπαταρία λιθίου» ή «συσσωρευτής λιθίου» αναφέρεται σε μια κατηγορία

μπαταριών με διαφορετική σύνθεση, περιλαμβάνοντας πολλούς τύπους καθοδικών ηλεκτροδίων και ηλεκτρολυτών.



Εικόνα 11 : Μπαταρία Li-Poli 2200mAh

| Χαρακτηριστικά              |                           |
|-----------------------------|---------------------------|
| 7.4V 2-cell pack            | 2200mAh χωρητικότητα      |
| 20C συνεχόμενης αποφόρτισης | 8AWG bare discharge leads |
| 1000 κύκλοι ζωής            | DC2.1 Βύσμα Σύνδεσης      |
| Μέγεθος: 103x34x15mm        | Βάρος: 130g               |

Πίνακας 2 : Χαρακτηριστικά μπαταρίας 7.4V Lipo 2200mAh

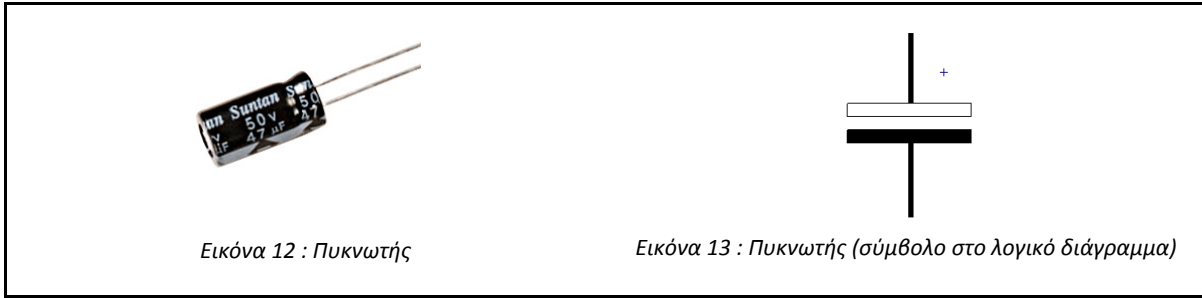
## Ηλεκτρολυτικός Πυκνωτής

Ο ηλεκτρολυτικός πυκνωτής είναι ένας τύπος πυκνωτή του οποίου η μία πλάκα που χρησιμοποιείται είναι κατασκευασμένη από ένα ιονισμένο υγρής μορφής σκεύασμα. Συνήθως έχουν μεγαλύτερη χωρητικότητα σε αναλογία με τον όγκο τους από ότι έχουν άλλοι τύποι πυκνωτών. Η αξία τους είναι ανεκτίμητη όσον αφορά την χρήση σε κυκλώματα υψηλών ρευμάτων και χαμηλών συχνοτήτων.

Ειδικά, η χρήση τους σε τροφοδοτικά ως φίλτρα για το φιλτράρισμα της Τάσης που εξάγεται από ένα κύκλωμα σταθεροποίησης είναι πολύ συχνή. Οι ηλεκτρολυτικοί πυκνωτές έχουν μεγάλη χωρητικότητα και μας επιτρέπουν να κατασκευάσουμε φίλτρα που έχουν πολύ χαμηλές συχνότητες αποκοπής.

Το σύμβολο στο λογικό διάγραμμα και μια αναπαράσταση του πραγματικού ηλεκτρονικού στοιχείου φαίνονται στις παρακάτω εικόνες:





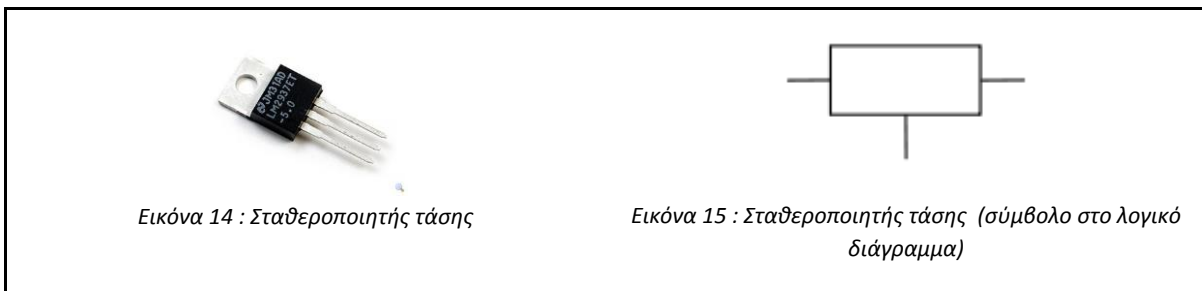
Εικόνα 12 : Πυκνωτής

Εικόνα 13 : Πυκνωτής (σύμβολο στο λογικό διάγραμμα)

### Σταθεροποιητής Τάσης

Σταθεροποιητής τάσης είναι ένα ηλεκτρονικό στοιχείο το οποίο χρησιμοποιείται για να μετατρέπει μια ασταθή τάση σε μια τάση σταθερής τιμής. Υπάρχουν διαφόρων ειδών σταθεροποιητές, για εναλλασσόμενες τάσεις αλλά και για συνεχείς. Αν η τάση εξόδου είναι πολύ χαμηλή τότε το κύκλωμα ελέγχου εσωτερικά του σταθεροποιητή αναγκάζεται να παράγει μεγαλύτερη τάση. Η αντίθετη διαδικασία, δηλαδή αν η τάση εξόδου είναι υψηλή, θα παραχθεί μικρότερη τάση, ισχύει μόνο για μερικούς σταθεροποιητές.

Δίνεται μεγάλη σημασία κατά την κατασκευή του κυκλώματος ελέγχου των σταθεροποιητών καθώς πρέπει να παράγει σταθερή τάση και να έχει γρήγορη απόκριση. Πράγματα τα οποία είναι αντιστρόφως ανάλογα. Το σύμβολο στο λογικό διάγραμμα και μια εικόνα του πραγματικού ηλεκτρονικού στοιχείου φαίνονται παρακάτω:



Εικόνα 14 : Σταθεροποιητής τάσης

Εικόνα 15 : Σταθεροποιητής τάσης (σύμβολο στο λογικό διάγραμμα)

## 3.4 Ανάλυση τροφοδοσίας ενέργειας

Ένα σημαντικό μέρος για την σωστή λειτουργία της πλατφόρμας είναι ο υπολογισμός της κατανάλωσης που έχει στο σύνολο.

Υπολογίζοντας πώς κάθε βραχίονας «πόδι» χρειάζεται 4 κινητήρες. Και κάθε κινητήρας έχει ως μέγιστη κατανάλωση 0.8A ,συμπεραίνουμε πως η κατανάλωση ρεύματος της βάσης είναι 9.6A μέγιστο. Σε αυτό το υποσύνολο χρειάζεται να συμπεριληφθείς και η κατανάλωση ρεύματος του ηλεκτρονικού κυκλώματος «εγκέφαλος» η οποία στο μέγιστο μπορεί να είναι 2.4A.

Άρα στο σύνολο η μέγιστη (αν και απίθανη εκδοχή) κατανάλωση που μπορεί να έχει η ρομποτική πλατφόρμα είναι 12Α.

Για αυτό το λόγο οι μπαταρίες που επιλέχθηκαν έχουν μεγάλο ρυθμό εκφόρτισης όπως επίσης μπορεί να θεωρηθεί απαραίτητο ένα κύκλωμα σταθεροποίησης της τάσης. Αναλυτικά για το κύκλωμα θα μιλήσουμε στα επόμενα κεφάλαια.

### 3.6 Αισθητήρες και ολοκληρωμένα κυκλώματα

Κάθε ρομποτική πλατφόρμα βασίζεται στα εξωτερικά ερεθίσματα που δέχεται από το περιβάλλον και ύστερα στον υπολογισμό τους για την απόδοση μίας ενέργειας. (*event-driven computational model*).

Η ρομποτική αίσθηση είναι ένας κλάδος της ρομποτικής επιστήμης που προορίζεται να δώσει στο ρομπότ δυνατότητες αντίληψης, έτσι ώστε τα ρομπότ να είναι πιο ανθρώπινα. Η ρομποτική αντίληψης δίνει κυρίως στο ρομπότ τη δυνατότητα να δει, να αγγίξει, να ακούσει όπως για παράδειγμα αν θέλουμε να προχωρήσουμε χρησιμοποιούμε αλγόριθμους που απαιτούν περιβαλλοντική ανάδραση.

Ένας αισθητήρας (*ονομάζεται επίσης ανιχνευτής*) είναι ένας μετατροπέας που μετρά μια φυσική ποσότητα και την μετατρέπει σε ένα σήμα το οποίο μπορεί να διαβαστεί από έναν παρατηρητή ή από (*σήμερα κυρίως ηλεκτρονικά*) ένα μέσο.

Για παράδειγμα, ένα θερμόμετρο υδραργύρου μετατρέπει τη μετρούμενη θερμοκρασία σε διαστολή και συστολή ενός υγρού το οποίο μπορεί να διαβαστεί σε ένα βαθμονομημένο γυάλινο σωλήνα. Ένα θερμοζεύγος μετατρέπει τη θερμοκρασία σε μια τάση εξόδου η οποία μπορεί να διαβαστεί από ένα βολτόμετρο. Για την ακρίβεια, οι περισσότεροι αισθητήρες έχουν βαθμονομηθεί σύμφωνα με γνωστές μετρήσεις όπως μέτρα ,εκατοστά, βαθμούς Κελσίου κλπ.

#### Ταξινόμηση των αισθητήρων

Για την καλύτερη κατανόηση του τρόπου λειτουργίας των αισθητήρων και των ιδιοτήτων τους είναι απαραίτητη η ταξινόμηση τους. Ως κριτήρια ταξινόμησης μπορούν να χρησιμοποιηθούν πολλά από τα χαρακτηριστικά τους, όπως ο τρόπος ένδειξης, η αρχή λειτουργίας και το ερέθισμα που δέχονται.

Ως προς τον τρόπο ένδειξης οι αισθητήρες διακρίνονται σε αναλογικούς και ψηφιακούς.

**Αναλογικοί :** Είναι οι αισθητήρες που μετατρέπουν μηχανικά μεγέθη π.χ. μήκη διαδρόμων σε ηλεκτρικά σήματα τάσεως ή εντάσεως ρεύματος.

**Ψηφιακοί :** Είναι οι αισθητήρες που συλλαμβάνουν το προς μέτρηση μέγεθος με αριθμητικό τρόπο και παράγουν ψηφιακά σήματα με δύο μόνο τιμές.

Ως προς την αρχή λειτουργίας τους οι αισθητήρες διακρίνονται σε:

*Μαγνητικούς, Ηλεκτρομαγνητικούς, Θερμικούς, Πιεζοηλεκτρικούς, Πιεζοαντίσταση, Επαγωγικούς, Χωρικούς, Οπτικούς κλπ.*

Ως προς το ερέθισμα το οποίο δέχονται και μετρούν οι αισθητήρες διακρίνονται σε :

*Θέσης – Μετακίνησης , θερμοκρασίας , Δύναμης , Ταχύτητα Επιτάχυνσης*

### Αποκλίσεις αισθητήρα

Εάν ο αισθητήρας δεν είναι ιδανικός (κάτι που συνήθως ισχύει), μπορούν να παρατηρηθούν διάφοροι τύποι αποκλίσεων:

- Η ευαισθησία μπορεί να διαφέρουν στην πράξη από την καθορισμένη τιμή. Αυτό ονομάζεται σφάλμα ευαισθησίας, (ο αισθητήρας λειτουργεί ακόμα γραμμικά ανεξάρτητα με το σφάλμα).
- Δεδομένου ότι το φάσμα του σήματος εξόδου είναι πάντα περιορισμένο, το σήμα εξόδου θα φτάσει τελικά στο ελάχιστο ή στο μέγιστο όταν η μετρούμενη τιμή υπερβαίνει τα όρια. Το εύρος πλήρους κλίμακας καθορίζει τις μέγιστες και ελάχιστες τιμές της μετρούμενης ιδιότητας.
- Αν το σήμα εξόδου δεν είναι μηδέν, όταν η μετρούμενη τιμή είναι μηδέν, ο αισθητήρας έχει μετατόπιση ή μεροληψία. Αυτό ορίζεται ως η έξοδος του αισθητήρα σε μηδενική είσοδο.
- Εάν η ευαισθησία δεν είναι σταθερή σε όλο το εύρος του αισθητήρα ονομάζεται, μη γραμμικότητα. Συνήθως αυτό ορίζεται από το πόσο η έξοδος διαφέρει από την ιδανική συμπεριφορά σε όλο το εύρος του αισθητήρα, και συχνά σημειώνεται ως ποσοστό του πλήρους φάσματος.
- Εάν η απόκλιση προκαλείται από μια ταχεία μεταβολή της μετρούμενης ιδιότητας με την πάροδο του χρόνου, υπάρχει ένα δυναμικό σφάλμα.
- Μακροχρόνια αλλαγή της τιμής εξόδου με σταθερή είσοδο συνήθως δείχνει μια αργή υποβάθμιση των ιδιοτήτων του αισθητήρα σε ένα μεγάλο χρονικό διάστημα.
- Ο θόρυβος είναι μια τυχαία απόκλιση του σήματος που μεταβάλλεται στο χρόνο.
- Η υστέρηση είναι ένα σφάλμα που προκαλείται όταν η μετρούμενη ιδιότητα αντιστρέφει την κατεύθυνση, αλλά υπάρχει κάποια πεπερασμένη υστέρηση στο χρόνο για την

ανταπόκριση του αισθητήρα, δημιουργώντας μια διαφορετική αντιστάθμιση σφάλματος σε μία κατεύθυνση από ό, τι στην άλλη.

- Εάν το σήμα παρακολουθείται ψηφιακά, υπάρχει περιορισμός της συχνότητας δειγματοληψίας και μπορεί να προκαλέσει κάποιο δυναμικό λάθος (μεταβλητότητα σφάλματος).
- Ο αισθητήρας μπορεί σε κάποιο βαθμό να είναι ευαίσθητος και σε άλλες περιβαλλοντικές ιδιότητες εκτός της μετρούμενης ιδιότητας. Αυτό συνήθως προκαλεί σφάλματα στην απόδοση και ακρίβεια του αισθητήρα.

### Ανάλυση (Resolution)

Η ανάλυση της τιμής που μπορεί να προσφέρει ο κάθε αισθητήρας είναι η ελάχιστη αλλαγή η οποία μπορεί να ανιχνευθεί. Η ανάλυση σχετίζεται με την ακρίβεια με την οποία γίνεται η μέτρηση.

## 3.6.1 Μικροεπεξεργαστής

Ένα βασικό εξάρτημα των ηλεκτρονικών συσκευών αλλά και του συνόλου αποτελεί ο μικροεπεξεργαστής. Ένας **μικροεπεξεργαστής** περιλαμβάνει τις περισσότερες ή όλες τις λειτουργίες μιας κεντρικής μονάδας επεξεργασίας (ΚΜΕ) ενός ηλεκτρονικού σε ένα ενιαίο ολοκληρωμένο κύκλωμα (IC). Οι πρώτοι μικροεπεξεργαστές εμφανίστηκαν στις αρχές της δεκαετίας του 1970 και χρησιμοποιήθηκαν σε ηλεκτρονικές αριθμομηχανές. Η ενσωμάτωση των μικροεπεξεργαστών σε άλλες συσκευές, όπως τερματικά, εκτυπωτές κλπ., ακολούθησε σχετικά γρήγορα. Με χρήση ενός οκτάμπιτου μικροεπεξεργαστή, κατασκευάστηκε ο πρώτος μικροϋπολογιστής γενικού σκοπού στα μέσα της δεκαετίας του 1970. Η ραγδαία ανάπτυξη της τεχνολογίας των μικροεπεξεργαστών που ακολούθησε συνδέεται με τις αυξημένες απαιτήσεις από γλώσσες προγραμματισμού υψηλού επιπέδου.

Ένας σύγχρονος μικροεπεξεργαστής αποτελείται από τις ακόλουθες μονάδες.

- **Μονάδα αποκωδικοποίησης (Decoding Unit)**
- **Αριθμητική και Λογική Μονάδα (Arithmetic and Logical Unit, ALU):** Η μονάδα στην οποία εκτελούνται μία προς μία οι αριθμητικές ή λογικές πράξεις, όπως υπαγορεύονται από τις εντολές που έχουν δοθεί στον υπολογιστή.
- **Καταχωρητές (Registers):** Μικρά κελιά μνήμης στο εσωτερικό του επεξεργαστή, που χρησιμοποιούνται για την προσωρινή αποθήκευση των δεδομένων, καθώς αυτά υφίστανται επεξεργασία. Οι καταχωρητές διαφέρουν ανάλογα με τον τύπο του επεξεργαστή και τον κατασκευαστή, τόσο ως προς την οργάνωση όσο και ως προς τη χωρητικότητά τους.

- **Μονάδα ελέγχου (Control Unit):** Ελέγχει τη ροή δεδομένων από και προς την ALU, τους καταχωρητές, τη μνήμη και τις περιφερειακές μονάδες εισόδου/εξόδου.
- **Μονάδα προσκόμισης (Fetch Unit):** Μεταφέρει τις εντολές από τη μνήμη στον επεξεργαστή.
- **Μονάδα προστασίας (Protection Unit):** Εξασφαλίζει το αποδεκτό της κάθε διεργασίας που εκτελεί ο επεξεργαστής, ώστε να μη τροποποιούνται δεδομένα που δεν πρέπει ή να μην εκτελούνται μη αποδεκτές εντολές, όπως π.χ. διαίρεση αριθμού με το μηδέν.

Στην συγκεκριμένη εργασία χρησιμοποιήθηκε ο μικροεπεξεργαστής **ARM Cortex M4** ο οποίος βρίσκεται σε διάφορες έτοιμες πλακέτες όπως το **Teensy 3.0**. Όπως έχει προαναφερθεί και σε προηγούμενα κεφάλαια ,το μέγεθος και το βάρος είναι ένας σημαντικότερος παράγοντας για την σωστή λειτουργία της ρομποτικής πλατφόρμας.

Ωστόσο η επεξεργαστική ικανότητα δεν πρέπει να αποτελέσει εξαίρεση λόγο της πολυπλοκότητας των εφαρμογών που επρόκειτο να αναλυθούν αργότερα. Για αυτό το λόγο η καλύτερη και πιο προσιτή επιλογή που υπήρξε και επιλέχθηκε είναι η πλακέτα **Teensy 3.0** με επεξεργαστή **ARM Cortex M4**.

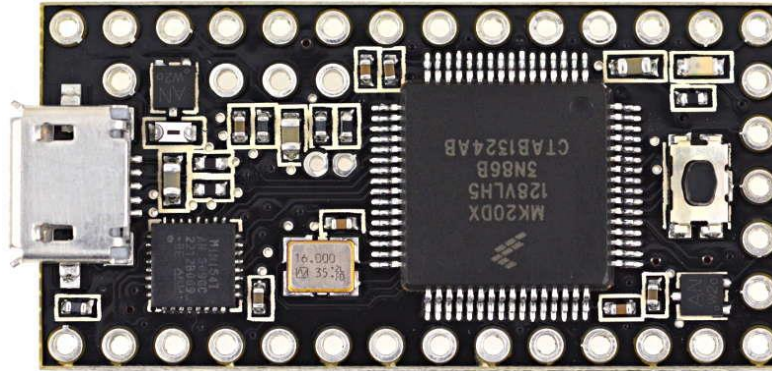
Το **Teensy 3.0** είναι μια breadboard\* εύκολης ανάπτυξη εφαρμογών με πολύ καλά χαρακτηριστικά. Φέρει έναν 32 bit μικροεπεξεργαστή **ARM Cortex** στο μίγμα ώστε να μπορεί να υποστηρίξει την υπολογιστική απαίτηση του ρομπότ που δημιουργούμε.

Παρέχει έναν **boot loader** με αποτέλεσμα να μην απαιτείται εξωτερικός προγραμματιστής και απλά συνδέοντας το με USB στον υπολογιστή να «φορτώνουμε» τα προγράμματα. Ο προγραμματισμός της πλακέτας Teensy έγινε με την χρήση του Microsoft Visual Studio χρησιμοποιώντας ένα δωρεάν plugin ,το Micro Visual Studio. Εφαρμόζοντας **μία ανάμειξη της C με την C++** μπορούσαμε να δημιουργήσουμε μέχρι και ολόκληρο λειτουργικό σύστημα.

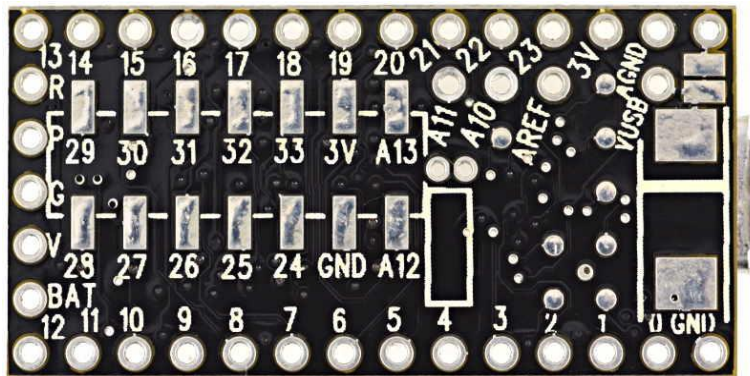
Ο **επεξεργαστής** στο teensy έχει επίσης πρόσβαση **στη θύρα USB** και μπορεί να μιμηθεί οποιοδήποτε είδος συσκευής USB, καθιστώντας το ιδανικό **για USB-MIDI και άλλα έργα HID**. Ο επεξεργαστής **32 bit** φέρνει μερικά άλλα στοιχεία, όπως πολλαπλά κανάλια άμεσης προσπέλασης μνήμης, αρκετές ADCs υψηλής ανάλυσης και **I2S ψηφιακή διεπαφή ήχου**. Υπάρχουν επίσης **4 ξεχωριστά χρονόμετρα συν ένα χρονόμετρο καθυστέρησης**. Επιπρόσθετα, μπορεί να παρέχει τάση στο σύστημα **3.3V, 100mA**.

Όλη αυτή η λειτουργικότητα έχει εμπλακεί σε έναν πίνακα 1.4 x 0.7 ιντσών με όλες τις κεφαλίδες σε ένα πλέγμα 0.1 ".

Στις παρακάτω εικόνες βλέπουμε την πλακέτα Teensy 3.0.



Εικόνα 16 : Teensy 3.0 (πλευρά 1)



Εικόνα 17 : Teensy 3.0 (πλευρά 2)

| Χαρακτηριστικά   |
|--|
| 32 bit ARM Cortex-M4 72 MHz CPU (M4 = DSP extensions)              |
| 256K Flash Memory, 64K RAM, 2K EEPROM                              |
| 21 High Resolution Analog Inputs (13 bits usable, 16 bit hardware) |
| 34 Digital I/O Pins (5V tolerance on Digital Inputs)               |
| 12 PWM outputs   |
| 7 Timers for intervals/delays, separate from PWM                   |
| USB with dedicated DMA memory transfers                            |
| 3 UARTs (serial ports)   |
| SPI, I2C, I2S, CAN Bus, IR modulator                               |
| I2S (for high quality audio interface)                             |
| Real Time Clock (with user-added 32.768 crystal and battery)       |
| 4 general purpose DMA channels (separate from USB)                 |
| Touch Sensor Inputs  |

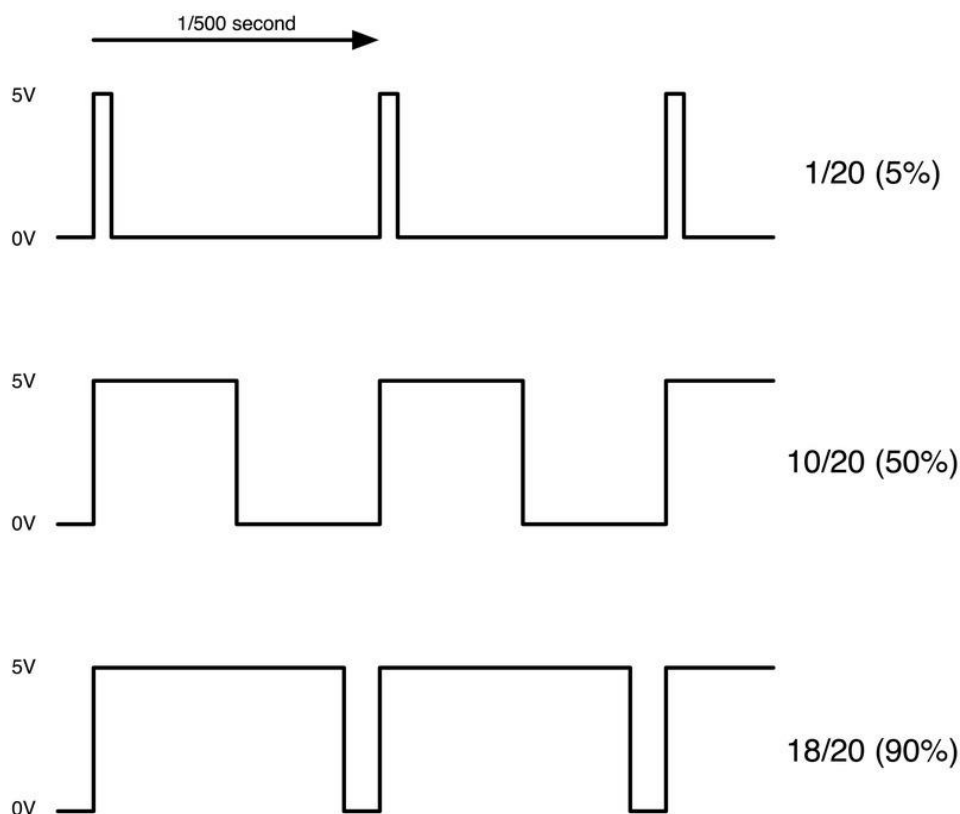
### 3.6.2 Μικροεπεξεργαστής χειρισμού κινητήρων

Ένα βασικό μέρος των ηλεκτρονικών εξαρτημάτων που χρησιμοποιούμε είναι ο μικροεπεξεργαστής χειρισμού κινητήρων. Θεωρείται μια πλακέτα με έναν επεξεργαστή και μερικά περιφερειακά κυκλώματα η οποία εξειδικεύεται στην μετάδοση PWM σημάτων προς τους ηλεκτρικούς κινητήρες του ρομπότ.

#### **PWM (Pulse-width modulation)**

Πριν τη παρουσίαση της πλακέτας χειρισμού των κινητήρων θα ήταν χρήσιμο να αναλυθεί ο όρος PWM (*Pulse-width modulation*). Τα PWM είναι κυκλώματα που ελέγχουν το εύρος και την συχνότητα παλμού. Αυτά τα στοιχεία είναι βασικά για τον έλεγχο καταναλώσεων συνεχούς τάσης. Με τον έλεγχο του εύρους παλμού και συχνότητας γίνεται οδήγηση κινητήρων DC motor, λαμπτήρων, ωμικών φορτίων, πηνίων, ηλεκτρομαγνητών κλπ. Πιο απλά το 'Pulse Width Modulation' (*PWM*) είναι μια τεχνική ελέγχου της ενέργειας.

Το παρακάτω διάγραμμα δείχνει το σήμα από έναν από τους ακροδέκτες PWM.



Εικόνα 18 : Σήματα PWM

Με αυτό τον τρόπο μπορούμε να δημιουργούμε τάσεις ανάμεσα των 0V και 5V. Δηλαδή αν θέλουμε να παράγουμε μία τάση 2.5V αρκεί να έχουμε τον διακόπτη ανοιχτό για το 50% του χρόνου.

## 12-channel Mini Maestro

Το 12-channel Mini Maestro είναι μία πλακέτα χειρισμού ηλεκτρικών κινητήρων (servo) με ιδιαίτερες επιδόσεις και πλεονεκτήματα όπως ο χειρισμός μέσω USB θύρας αλλά και εσωτερικά scripts για αποθήκευση κάποιων patterns. Επίσης καλύπτει πολλές δυνατότητες για τον χειρισμό των servo αφού παρέχει ανάλυση σήματος στα 0.25μs και αλγορίθμους ομαλοποίησης της κίνησης είτε με σταθερή ταχύτητα αλλαγής θέσης είτε με σταθερή επιτάχυνση.



Εικόνα 19 : Servocontroller Pololu Mini Maestro 12

Ένα κύριο χαρακτηριστικό του είναι το μέγεθος που αποτελεί βασικό παράγοντα επιλογής για την εργασία καθώς και οι επιδόσεις που παρέχει για τον χειρισμό των ηλεκτρικών κινητήρων.

Ας δούμε αναλυτικά μερικά σημαντικά χαρακτηριστικά του Pololu Servo Controller 12 :

| Διαστάσεις |                 |
|------------|-----------------|
| Πλάτος     | 1.10" (2.79 cm) |
| Μήκος      | 1.42" (3.61 cm) |
| Βάρος      | 4.2 g           |

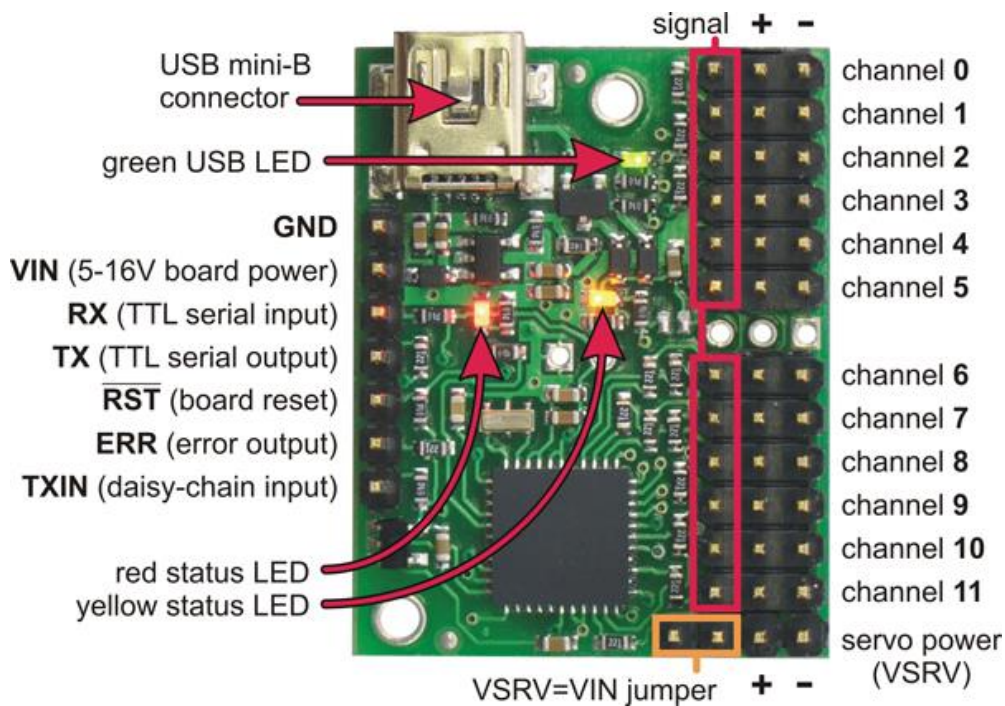
Πίνακας 3 : Διαστάσεις Pololu Servo Controller



| Γενικά Χαρακτηριστικά               |                  |
|-------------------------------------|------------------|
| Κανάλια μετάδοσης                   | 12               |
| Ρυθμός Μετάδοσης Δεδομένων (Serial) | 300 - 200000 bps |
| Μικρότερη δυνατή τάση λειτουργίας   | 5 V              |
| Μέγιστη δυνατή τάση λειτουργίας     | 16 V             |
| Ρυθμός μετάδοσης Παλμών             | 1-333 Hz         |
| Εύρος Παλμού                        | 64-4080 μs       |

Πίνακας 4 : Χαρακτηριστικά Pololu Servo Controller

Η παρακάτω εικόνα εμφανίζει αναλυτικά την πλακέτα Pololu Servo Controller 12.



Εικόνα 20 : Servocontroller Pololu Mini Maestro 12 (Συνδέσεις)

Η επικοινωνία επιτυγχάνεται με την αποστολή πακέτων εντολών που αποτελείται από μια μόνο byte εντολή ακολουθούμενη από τυχόν ψηφιολέξεις δεδομένων που η εντολή απαιτεί.

Το 12-channel Mini Maestro ανταποκρίνεται σε 3 υπο-πρωτόκολλα:

### **Compact υπό-πρωτόκολλο**

Αυτή είναι η απλούστερη και πιο συμπαγής έκδοση από τα 3 πρωτόκολλα.

Για παράδειγμα, αν θέλουμε να θέσουμε τον στόχο ενός βηματικού κινητήρα στα 1500 μs, θα μπορούσαμε να στείλουμε την παρακάτω ακολουθία από bytes:

```
hex: 0x84, 0x00, 0x70, 0x2E  
decimal: 132, 0, 112, 46
```

*Εξίσωση 1 : Παράδειγμα επικοινωνίας Servocontroller (Compact)*

Το 0x84 byte είναι η εντολή Set Target, το πρώτο byte δεδομένων 0x00 είναι ο αριθμός θέσης του κινητήρα, και τα τελευταία δύο bytes δεδομένων περιέχουν το στόχο σε μονάδες μικροδευτερολέπτων.

### **Pololu υπό-πρωτόκολλο**

Το πρωτόκολλο αυτό είναι συμβατό με το σειριακό πρωτόκολλο που χρησιμοποιείται και από άλλους σειριακούς σερβοελεγκτές.

### **Mini SSC υπό-πρωτόκολλο**

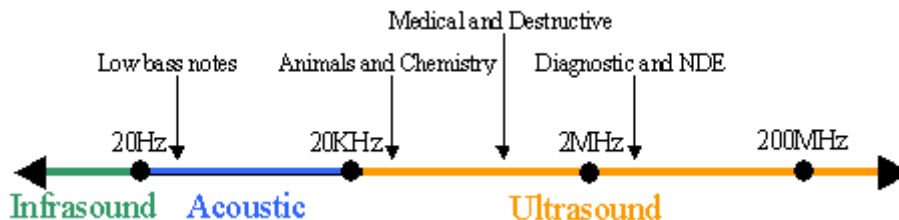
Αυτό το πρωτόκολλο επιτρέπει να ελέγχουμε έως και 254 διαφορετικά servos από τη σύνδεση πολλαπλών σερβο-ελεγκτών μαζί. Χρειάζονται μόνο 3 σειριακές bytes για να οριστεί ο στόχος ενός βηματικού κινητήρα, οπότε το πρωτόκολλο αυτό είναι καλό, αν πρέπει να στείλουμε πολλές εντολές γρήγορα.

### 3.6.3 Αισθητήρας Απόστασης

Οι αισθητήρες υπερήχων λειτουργούν με την ίδια αρχή που λειτουργούν τα ραντάρ και τα σόναρ. Εκτιμούν την απόσταση ενός στόχου λαμβάνοντας υπόψη τους την αντανάκλαση ενός ραδιοκύματος ή ενός ηχητικού σήματος πάνω στο στόχο. Δημιουργούν υψηλής συχνότητας κύματα και χρησιμοποιώντας το επιστρεφόμενο σήμα καθορίζουν την απόσταση ή ακόμα και την ταχύτητα του στόχου. Για να το επιτύχουν αυτό χρησιμοποιούν τον χρόνο που έκανε το σήμα για να καλύψει την απόσταση από τον αισθητήρα στο αντικείμενο και πίσω. Εφαρμογές τους θα βρούμε σε ένα μεγάλο εύρος τεχνολογιών από την μέτρηση της διεύθυνσης και της ταχύτητας του ανέμου έως και την απεικονιστική ιατρική.

#### Υπέρηχος

Σαν υπέρηχο καθορίζουμε εκείνο το κύμα το οποίο βρίσκεται πάνω από την μέγιστη συχνότητα που μπορεί να ακούσει το ανθρώπινο αυτί, η οποία είναι  $20 \text{ Hz} < f < 20\,000 \text{ Hz}$ .



Εικόνα 21 : Συχνότητες ήχου

Παρόλο πάντως που εμείς δεν τους ακούμε κάποια ζώα μπορούν και να τους ακούν αλλά και να τους χρησιμοποιούν. Χαρακτηριστικά παραδείγματα που μας το δείχνουν αυτό είναι η κίνηση των νυχτερίδων και η σφυρίχτρα που χρησιμοποιείται για τους σκύλους. Οι νυχτερίδες μάλιστα μπορούν να αντιληφθούν τον στόχο που κυνηγούν χάρη στο φαινόμενο Doppler.

#### Ο αισθητήρας υπερήχων SR05

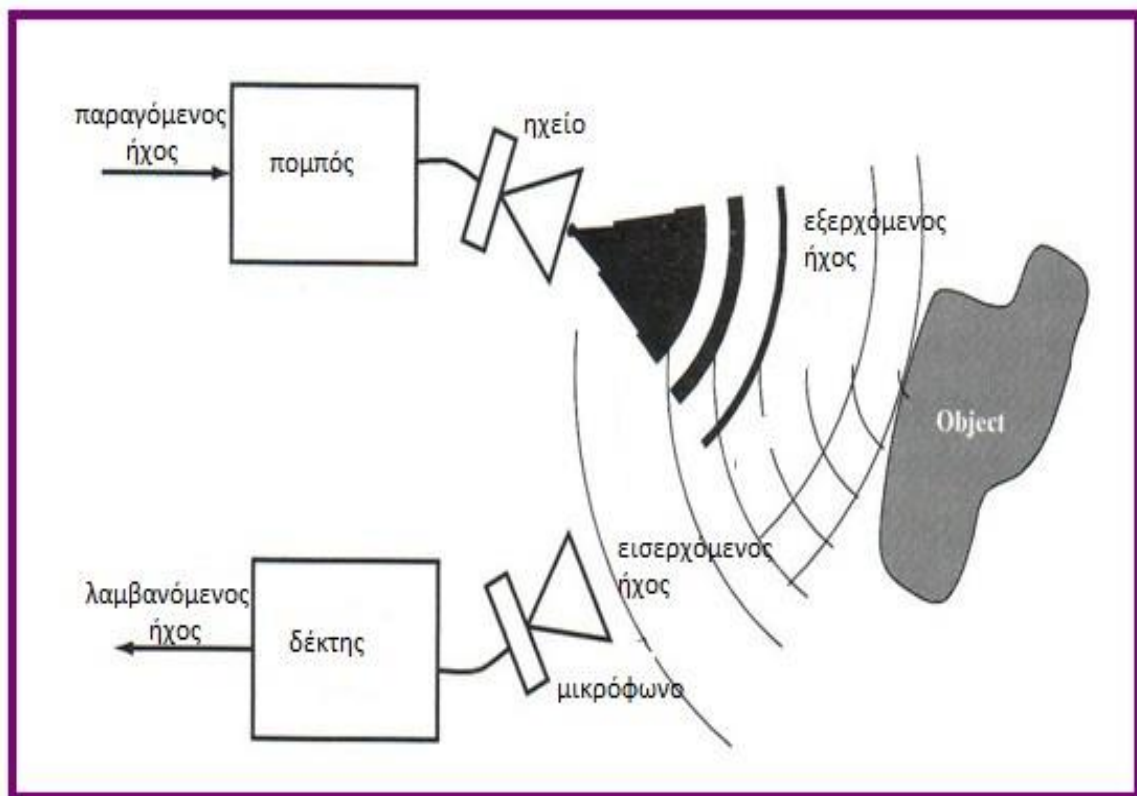


Εικόνα 22 : Ultrasonic SRF05

Ο αισθητήρας υπερήχων SR05 καθιστά το ρομπότ ικανό να μετρά απόσταση από ένα αντικείμενο και να αντιδρά στην κίνηση. Είναι αποτελεσματικό στον εντοπισμό αντικειμένων σε απόσταση

από 0 έως 2 μέτρα με ακρίβεια 3 εκατοστών. Η διαδικασία έχει ως εξής. Ηχητικά κύματα εκπέμπονται από τη μία (πομπός) από τις δύο πορτοκαλί οπές που βρίσκονται στην πρόσοψη του αισθητήρα. Ανακλώνονται πάνω στο αντικείμενο που βρίσκεται στο εύρος εντοπισμού και επιστρέφουν στην άλλη οπή(δέκτης). Έτσι καθορίζεται η απόσταση ή η κίνηση του αντικειμένου. Ο υπολογισμός γίνεται μέσω μαθηματικού τύπου. Η ταχύτητα του ηχητικού κύματος πολλαπλασιάζεται με το μισό του χρόνου που χρειάζεται το κύμα να επιστρέψει στον αισθητήρα. Αυτή είναι και η απόσταση του αντικειμένου.

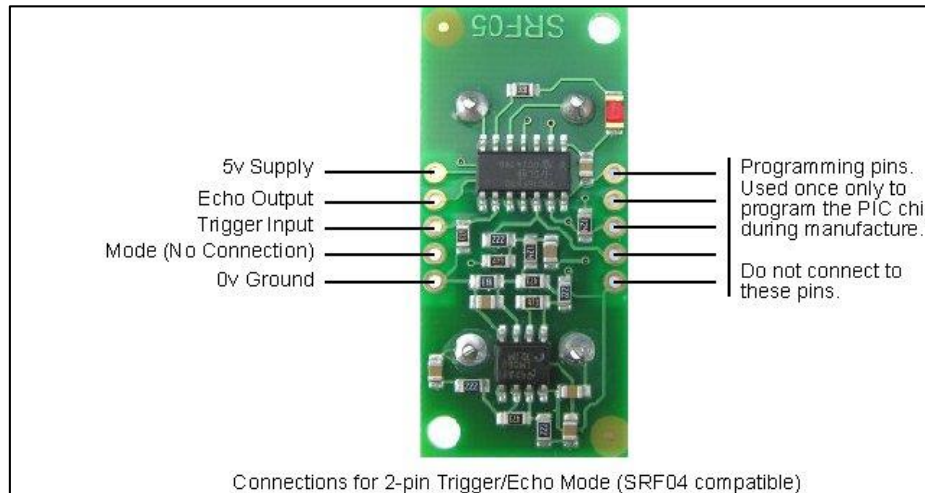
Το SRF05 είναι ένα εξελικτικό βήμα από την SRF04, και έχει σχεδιαστεί για να αυξηθεί η ευελιξία και το εύρος, και να μειωθεί ακόμα περισσότερο το κόστος. Ως εκ τούτου, το SRF05 είναι πλήρως συμβατό με το SRF04. Το εύρος αυξάνεται από 3 μέτρα σε 4 μέτρα. Ένας τρόπος λειτουργίας επιτρέπει στο SRF05 να χρησιμοποιεί ένα μόνο pin σαν trigger και receiver. Βέβαια παρέχεται και ο κλασικός τρόπος λειτουργίας με ξεχωριστό trigger pin και receiver pin, όπως το SRF04. Το SRF05 περιλαμβάνει μια μικρή καθυστέρηση πριν από την παλμική ηχώ για να δώσει την δυνατότητα χρήσης του και σε πιο αργούς ελεγκτές.



Εικόνα 23 : Αρχή λειτουργίας SONAR σχηματικά

## **Mode 1 - SRF04 compatible - Separate Trigger and Echo**

Αυτή η λειτουργία χρησιμοποιεί ξεχωριστά pins για trigger και receiver και είναι ο πιο απλός τρόπος λειτουργίας. Για να χρησιμοποιηθεί αυτή η λειτουργία, απλά αφήστε το mode pin χωρίς σύνδεση - το SRF05 έχει μια εσωτερική pull up αντίσταση σε αυτό το pin.



Εικόνα 24 : SRF05 Συνδέσεις

## **Mode 2 - Single pin for both Trigger and Echo**

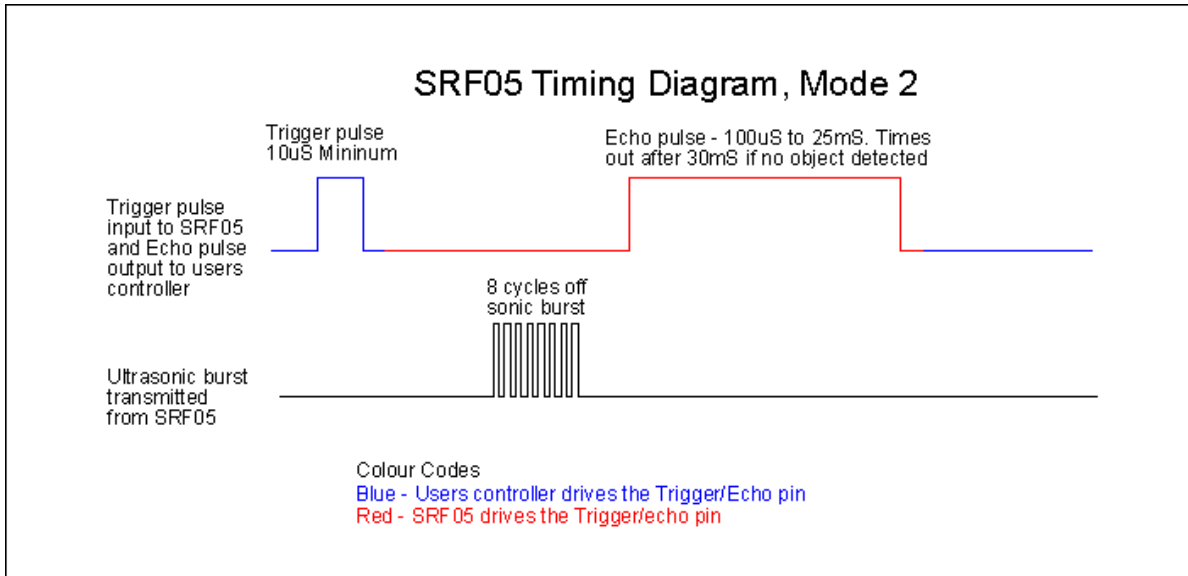
Αυτή η λειτουργία χρησιμοποιεί ένα ενιαίο pin για trigger και receiver. Για να χρησιμοποιήσετε αυτή τη λειτουργία, συνδέστε το mode pin με τον ακροδέκτη γείωσης 0V. Το σήμα ηχούς εμφανίζεται στο ίδιο pin με το σήμα ενεργοποίησης.

### **Υπολογισμός Απόστασης**

Τα διαγράμματα χρονισμού SRF05 φαίνεται από τα παραπάνω για κάθε λειτουργία. Το μόνο που πρέπει να παρέχει είναι ένα σύντομο παλμό 10us στην είσοδο trigger για να ξεκινήσει το φάσμα. Το SRF05 θα στείλει ένα ξέσπασμα 8 κύκλων υπερήχων με συχνότητα 40kHz. Στη συνέχεια, περιμένει για μια ηχώ, και μόλις ανιχνεύσει την πρώτη ειδοποιεί. Το receiver pin, επιστρέφει ένα παλμό του οποίου το πλάτος είναι ανάλογο με την απόσταση που ανιχνεύθηκε.

Το πλάτος του παλμού διαιρώντας το με 58 δίνει την απόσταση σε εκατοστά ή διαιρώντας το 148 δίνει την απόσταση σε ίντσες.  $\mu S/58 = \text{cm}$  ή  $\mu S/148 = \text{ίντσες}$ .

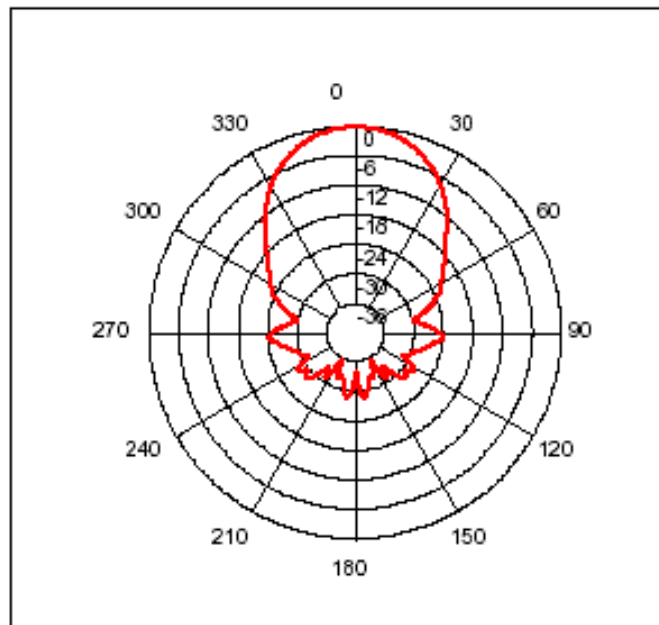
Το SRF05 μπορεί να ενεργοποιηθεί κάθε 50ms. Θα πρέπει να περιμένετε 50ms πριν από το επόμενο trigger, ακόμη και αν εντοπιστεί ένα κοντινό αντικείμενο και η ηχώ του παλμού είναι μικρότερη. Αυτό γίνεται για να διασφαλιστεί πως δεν θα υπάρξει παρεμβολή από προηγούμενη μέτρηση.



Εικόνα 25 : SRF05 Χρονισμός

### Μοτίβο της Δέσμης Υπερήχων

Το μοτίβο της δέσμης SRF05 είναι κωνικό με το πλάτος της δοκού να είναι συνάρτηση του εμβαδού επιφανείας.



Εικόνα 26 : SRF05 Δέσμη Υπερήχων

### 3.6.4 Αισθητήρας μαγνητικών πεδίων

Τα περισσότερα συστήματα πλοήγησης χρησιμοποιούν σήμερα κάποιο είδος πυξίδας για να καθορίσει την κατεύθυνση τους. Χρησιμοποιώντας το μαγνητικό πεδίο της Γης, οι ηλεκτρονικές πυξίδες που βασίζονται σε αισθητήρες μαγνητικής αντίστασης (MR) έχουν την δυνατότητα ανίχνευσης της περιστροφής μέχρι και 0,1 μοίρας .



Εικόνα 27 : Μαγνητόμετρο HMC6352

Αυτό είναι ένα breakout board της Honeywell HMC5883L, μια ψηφιακή πυξίδα τριών αξόνων. Η επικοινωνία με το HMC5883L είναι απλή και όλα γίνονται μέσω διασύνδεσης I2C. Δεν υπάρχει καμία ρυθμιστική αρχή τροφοδοσίας ρεύματος, οπότε θα πρέπει να παρέχεται ρυθμιζόμενη τάση από 2.16-3.6VDC.

| Χαρακτηριστικά        |
|-----------------------|
| I2C Διασύνδεση        |
| 2.16-3.6VDC           |
| Χαμηλή κατανάλωση     |
| 5 milli-gauss ανάλυση |

Πίνακας 5 : Χαρακτηριστικά HMC5883L

| Διαστάσεις |                  |
|------------|------------------|
| Πλάτος     | 0.7" (17.78cm)   |
| Μήκος      | 0.7" (x17.78 cm) |
| Βάρος      | 3.2 g            |

Πίνακας 6 : Διαστάσεις HMC5883L

## 3.6.5 Επιταχυνσιόμετρο

### Τι είναι το Επιταχυνσιόμετρο;

Το Επιταχυνσιόμετρο είναι μια ηλεκτρομηχανική συσκευή που έχει την ικανότητα να μετρά δυνάμεις επιτάχυνσης. Αυτές οι δυνάμεις μπορεί να είναι στατικές, όπως είναι η επιτάχυνση της βαρύτητας, η δυναμικές όταν προκαλούνται – προέρχονται από αλλαγές στην ταχύτητα ή στην διεύθυνση της κίνησης (επιταχύνσεις, επιβραδύνσεις, στροφές).

### Που χρησιμοποιείται;

Χρησιμοποιείται κυρίως σε αδρανειακά συστήματα πλοήγησης, που συνήθως αποκαλούνται Inertial Navigation System (INS), ή Inertial Reference System (IRS), ή Inertial Reference Unit (IRU), ή Air Data Inertial Reference Unit (ADIRU), τα οποία συστήματα συνδυάζοντας την χρήση Επιταχυνσιόμετρων Γυροσκοπίων και ηλεκτρονικών κυκλωμάτων, μπορούν να υπολογίζουν την θέση ενός οχήματος στον χώρο καθώς και την στάση και ταχύτητα αυτού με πολύ μεγάλη ακρίβεια, παράμετροι που είναι απολύτως απαραίτητοι για την λειτουργία του αυτόματου πιλότου και την ναυσιπλοΐα αεροπλάνων πλοίων και υποβρυχίων.

Το επιταχυνσιόμετρο χρησιμοποιείται επίσης για την μέτρηση και καταγραφή των επιταχύνσεων στους τρεις άξονες των αεροσκαφών από τον καταγραφέα των στοιχείων της πτήσης, για την ενεργοποίηση του αερόσακου στα αυτοκίνητα, και σε οποιαδήποτε άλλη εφαρμογή απαιτείται η μέτρηση της επιτάχυνσης.

### Πως λειτουργεί;

Υπάρχουν διάφοροι τρόποι λειτουργίας ενός επιταχυνσιόμετρου. Ένας τρόπος είναι η αξιοποίηση του πιεζοηλεκτρικού φαινομένου. Αυτά χρησιμοποιούν πιεζοκρύσταλο ο οποίος πιέζεται από μάζα ανάλογη της επιτάχυνσης που δέχεται αυτή και παράγει τάση λόγω πιεζοηλεκτρικού φαινομένου ανάλογη της επιτάχυνσης.

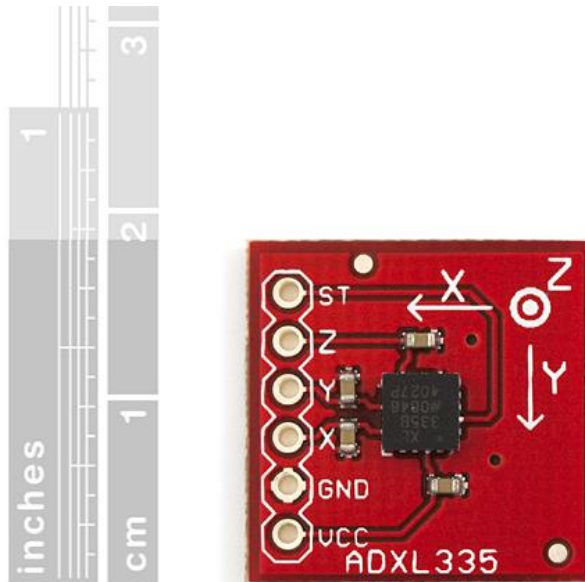
Ένας άλλος τρόπος είναι η μέτρηση αλλαγών στην χωρητικότητα ενός πυκνωτή. Ο πυκνωτής αποτελεί μέρος της ανάρτησης μάζας και η τιμή του μεταβάλλεται με την κίνηση αυτής της μάζας. Η αξιοποίηση της μεταβολής της χωρητικότητας του πυκνωτή μας παρέχει μέτρηση επιτάχυνσης.

Υπάρχουν και άλλοι τρόποι όπως η χρήση της αλλαγής της αντίστασης κάποιου υλικού ανάλογη της πίεσης που δέχεται, η χρήση φυσαλίδας ζεστού αέρα, και η χρήση των ιδιοτήτων του φωτός.



## ADXL 335

Το ADXL335 είναι ένα μικρό, χαμηλής ισχύος, επιταχυνσιόμετρο 3 αξόνων με εξόδους σήματος κλιμακούμενης τάσης. Μετρά την επιτάχυνση με ανάλυση τάξης του  $\pm 3G$ . Μπορεί να μετρήσει τη στατική επιτάχυνση της βαρύτητας σε εφαρμογές ανίχνευσης κλίσης, καθώς και τη δυναμική επιτάχυνση που προκύπτει από την κίνηση, χτυπημάτων ή κραδασμών.



Εικόνα 28 : ADXL 335

Το εύρος ζώνης μπορεί να επιλεγθεί ανάλογα με την εφαρμογή, με μια σειρά από 0,5 Hz έως 1600 Hz για X και Y άξονες, και μια σειρά από 0,5 Hz έως 550 Hz για τον άξονα Z.

| Εφαρμογές                             |                              |
|---------------------------------------|------------------------------|
| Κινητές συσκευές                      | Συστήματα τυχερών παιχνιδιών |
| Προστασία της μονάδας δίσκου          | Σταθεροποίηση εικόνας        |
| Αθλητισμός και συσκευές για την υγεία |                              |

Πίνακας 7: Εφαρμογές του ADXL335

### 3.6.6 Αισθητήρας ανίχνευσης Διοξειδίου του Άνθρακα

Ο αισθητήρας αερίου ανιχνεύει αέρια υδρογόνου. Είναι σε θέση να δώσει συναγερμό όταν υπάρχουν διαρροές καπνού / αερίου. Το ποτενσιομετρο μπορεί να ρυθμίσει την ευαισθησία του αισθητήρα. Αυτός ο αισθητήρας θα επιστρέψει μια αναλογική τιμή που αντιπροσωπεύει την ένταση του αερίου που ανιχνεύεται.



| Χαρακτηριστικά          |
|-------------------------|
| Τύπος: Αναλογικός       |
| Ανιχνευτής: QM-NG1      |
| Τροφοδοσία ρεύματος: 5V |
| Κατανάλωση: 100mA       |

Εικόνα 29 : Gas Sensor

### 3.6.7 Παγκόσμιο Σύστημα Εντοπισμού Θέσης

Το GPS (Global Positioning System), Παγκόσμιο Σύστημα Θεσιθεσίας είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης, το οποίο βασίζεται σε ένα "πλέγμα" εικοσιτεσσάρων δορυφόρων της Γης, στους οποίους υπάρχουν ειδικές συσκευές, που ονομάζονται "δέκτες GPS". Οι δέκτες αυτοί παρέχουν ακριβείς πληροφορίες για τη θέση ενός σημείου, το υψόμετρό του, την ταχύτητα και την κατεύθυνση της κίνησης του. Επίσης, σε συνδυασμό με ειδικό λογισμικό χαρτογράφησης μπορούν να απεικονίσουν γραφικά τις πληροφορίες αυτές.



Εικόνα 30 : GPS EM406α

Το σύστημα εντοπισμού θέσης GPS σχηματίζει ένα παγκόσμιο δίκτυο, με εμβέλεια που καλύπτει ξηρά, θάλασσα και αέρα. Εξαιτίας αυτής της έκτασής του, είναι απαραίτητος ο διαχωρισμός του σε επιμέρους τμήματα όπου πραγματοποιούνται όλες οι λειτουργίες του αλλά και ο συντονισμός του.

Όλοι οι δορυφόροι βρίσκονται σε ύψος 12.552 μιλίων (20.200 χιλιομέτρων) πάνω από την επιφάνεια της θάλασσας και εκτελούν δύο περιστροφές γύρω από τη Γη κάθε 24ωρο

#### Πρωτόκολλο NMEA

Το πρωτόκολλο NMEA 0183 (ή NMEA για συντομία) ορίζει το σύνολο συνδυαζόμενων ηλεκτρικών στοιχείων αλλά και προδιαγραφών για την επικοινωνία μεταξύ ηλεκτρονικών συσκευών όπως sonars, βυθόμετρα, αυτόματους πιλότους αλλά και GPS δέκτες. Κυρίως στην απόλυτα εμπορική του εκδοχή βρήκε εφαρμογή σε GPS δέκτες που προορίζονται για το ευρύ κοινό. Η αρχή που όρισε για πρώτη φορά και ελέγχει αυτό το πρωτόκολλο είναι η National Marine Electronics Association στις Η.Π.Α.

Το NMEA 0183 είναι ένα πρότυπο το οποίο χρησιμοποιεί απλές ASCII σειριακές επικοινωνίες και ορίζει πώς θα γίνει η μετάδοση των δεδομένων από έναν "πομπό" σε ένα "δέκτη" μια δεδομένη χρονική στιγμή (τη φορά). Η έξοδος του NMEA είναι τύπου EIA-422A, αλλά για τους περισσότερους σκοπούς μπορεί να θεωρηθεί συμβατό με RS-232. Χρησιμοποιεί ρυθμό διαμεταγωγής **4800 bps, με 8 data bits**, χωρίς ισοτιμία (no parity) και ένα **stop bit( 8N1 )**. Κάθε ακολουθία δεδομένων ξεκινά με το σύμβολο του **δολαρίου (\$)** και ολοκληρώνεται με ένα Carriage Return (<CR><LF>). Τα δεδομένα που υπάρχουν ενδιάμεσα χωρίζονται με το κόμμα. Μπορεί να προστεθεί επιπλέον και η ακολουθία **checksum** για την επιβεβαιωμένη λήψη από τον δέκτη των δεδομένων.

Με την χρήση των ενδιάμεσων επεκτάσεων, ένας πομπός μπορεί να έχει αμφίδρομη επικοινωνία με πολλούς ακροατές, και χρησιμοποιώντας πολυπλέκτες, πολλαπλοί αισθητήρες μπορούν να επικοινωνούν διαδοχικά μέσω μιας και μόνο θύρας ενός H/Y

| Κανόνες Επιπέδου Εφαρμογής ( Application Layer )   |
|--|
| Κάθε μήνυμα ξεκινά με το σύμβολο του δολαρίου  |
| Οι επόμενοι 5 χαρακτήρες είναι αναγνωριστικά για τον τύπο του μηνύματος  |
| Όλα τα πεδία των δεδομένων διαχωρίζονται με το κόμμα   |
| Ο πρώτος χαρακτήρας του τελευταίου πεδίου δεδομένων είναι ο αστερίσκος   |
| Ο αστερίσκος προμηνύει την εκπομπή ενός διηγήσιμου <b>checksum</b> σε μορφή δεκαεξαδική ( <b>HEX</b> ). Το checksum είναι το αποκλειστικό <b>OR (XOR)</b> όλων των χαρακτήρων μεταξύ του δολαρίου και του αστερίσκου ( <b>\$...*</b> ) |
| Οι χαρακτήρες <CR><LF> ορίζουν το τέλος του συνολικού μηνύματος  |

Πίνακας 8 : Κανόνες Επιπέδου Εφαρμογής (NMEA)

Ένα παράδειγμα. Η NMEA μορφή για έναν συναγερμό ότι ένα σκάφος πλησίασε σε κάποιο προκαθορισμένο σημείο ( Waypoint ) έχει την εξής μορφή:

```
$GPRAM,A,A,0.10,N,WPTNME*32
```

Κώδικας 2 : Παράδειγμα NMEA μηνύματος

| Απάντηση | Πληροφορία  |
|----------|---|
| GP       | Ταυτότητα(ID) του πομπού («GP» για μονάδα GPS, «GL» για μονάδα <u>GLONASS</u> ) |
| AAM      | Συναγερμός Άφιξης   |
| A        | Κύκλος άφιξης   |

|        |  |
|--------|--|
| A      | Κάθετη θέση προς την επιφάνεια                       |
| 0.10   | Κλάσμα Κύκλου  |
| N      | Ναυτικά Μίλια  |
| WPTNME | Το όνομα του προκαθορισμένου σημείου (WayPoint Name) |
| *32    | Τα δεδομένα Checksum                                 |

Πίνακας 9 : Επεξήγηση παραδείγματος NMEA πρωτοκόλλου

| Περιγραφή Συνδέσεως |  |
|---------------------|--|
| Σύνδεση             | Περιγραφή  |
| VIN                 | Αυτή είναι η κύρια παροχή DC για 4.5V ~ 6.5 DC ισχύς εισόδου.  |
| TX                  | Αυτό είναι το κύριο κανάλι μεταδίδει για την έξοδο των δεδομένων πλοήγησης και μέτρησης λογισμικό πλοήγησης του χρήστη ή ο χρήστης γράφει λογισμικό.   |
| RX                  | Αυτό είναι το κύριο κανάλι λήψης για την λήψη εντολών λογισμικού στην πλακέτα του κινητήρα από το λογισμικό SiRFdemo ή από το χρήστη γραπτή λογισμικό. |
| GND                 | GND παρέχει το έδαφος για το διοικητικό συμβούλιο του κινητήρα. Συνδέστε όλους τους λόγους.  |
| 1PPS                | Αυτό το pin παρέχει ένα παλμό ανά δευτερόλεπτο εξόδου από το κεντρικό κύκλωμα για τον συγχρονισμό με το χρόνο GPS.                                     |

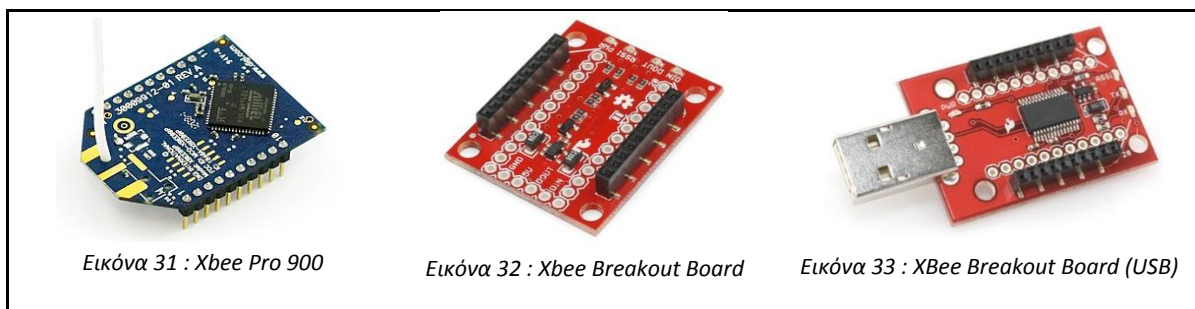
Πίνακας 10 : Συνδέσεις EM406a

### 3.6.8 Συσκευή Επικοινωνίας

Η μονάδα 900 RF XBee-PRO είναι ιδανική για low-latency point-to-multipoint εφαρμογές δικτύωσης. Ικανές point-to-point, peer-to-peer και point-to-multipoint δικτύωσης, η XBee-900 είναι ιδανική για λύσεις όπου η διεύθυνση RF και απόλυτη απόσταση μετάδοσης είναι υψίστης σημασίας για την εφαρμογή.

Ως μέλος της οικογένειας XBee των προϊόντων RF, η μονάδα είναι εύκολη στη χρήση, μοιράζεται ένα κοινό αποτύπωμα, και αξιοποιεί την λειτουργία Digi για Drop-in (άμεσης σύνδεσης) λύσεων δικτύωσης, συμπεριλαμβανομένων πύλες και προσαρμογείς. Η ρύθμιση δεν είναι αναγκαία για την out-of-the-box επικοινωνιών RF.

Η προεπιλεγμένη ρύθμιση παραμέτρων του module υποστηρίζει ένα ευρύ φάσμα των εφαρμογών του συστήματος δεδομένων. Προηγμένες ρυθμίσεις μπορούν να υλοποιηθούν με τη χρήση απλής σειριακής εντολής AT.

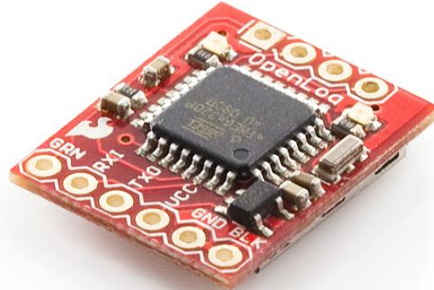


| <b>Χαρακτηριστικά</b>   |
|---|
| Επεξεργαστής : <i>ADF7023 transceiver, Cortex-M3 EFM32G230 @ 28 MHz</i>   |
| Συχνότητα Λειτουργίας : <i>902 to 928 MHz,</i>  |
| RF Ρυθμός Μετάδοσης Δεδομένων : <i>10 Kbps or 200 Kbps</i>  |
| Εμβέλεια εσωτερικών χώρων : <i>10 Kbps: 2000 ft (610 m); 200 Kbps: 1000 ft (305 m)</i>                              |
| Εμβέλεια Εξωτερικών χώρων : <i>10 Kbps: 9 miles (14 km); 200 Kbps: 4 miles (6.5 km)</i>                             |
| Ισχύς Μετάδοσης : <i>24 dBm (250 mW)</i>  |
| Ευσαιθησία Δέκτη : <i>-101 dBm @ 200 Kbps, -110 dBm @ 10 Kbps</i>   |
| Τάση Λειτουργίας : <i>2.1 to 3.6 VDC</i>  |
| Κατανάλωση ενέργειας κατά την μετάδοσης : <i>215 mA</i>   |
| Κατανάλωση ενέργειας κατά την λήψη : <i>29 mA</i>   |
| Κατανάλωση ενέργειας σε κατάσταση αναμονής : <i>2.5 uA</i>  |
| Συνδεσιμότητα : <i>UART (3V), SPI</i>   |
| Υποστηριζόμενες Δυκτιακες τοπολογίες : <i>DigiMesh, Repeater, Point-to-Point, Point-to-Multipoint, Peer-to-Peer</i> |
| Διασπορά Φάσματος : <i>FHSS</i>   |

Πίνακας 11 : Χαρακτηριστικά XBee Pro 900

### 3.6.9 Συσκευή Αποθήκευσης σε micro SD

Το OpenLog είναι ένας καταγραφέας δεδομένων ανοικτού κώδικα. Με την χρήση μιας SD κάρτας παρέχει την δυνατότητα δημιουργίας αρχείων για την συλλογή πληροφοριών.



Εικόνα 34 : Συσκευή καταγραφής OpenLog

| Χαρακτηριστικά   |  |
|--|--|
| Σύνδεση με χαμηλού κόστους κάρτες micro SD FAT16/32 έως 16GB     |  |
| Απλή διεπαφή εντολών   |  |
| Παραμετρικές ρυθμοί baud (2400 έως 115200bps)                    |  |
| Διαμόρφωση της μονάδας από το αρχείο config ή το σύστημα μενού   |  |
| Ισχύς, γείωση, και RX-I είναι οι ελάχιστες συνδέσεις             |  |
| Επαναπρογραμματιζόμενες ATmega328 χρησιμοποιώντας το Arduino IDE |  |
| Δύο LED δείχνουν την κατάσταση γραπτώς                           |  |
| Τάση εισόδου από 3.3V σε 12V                                     |  |
| 2mA κατάσταση απλής λειτουργίας                                  |  |
| 6mA στο μέγιστο ρυθμό εγγραφής                                   |  |

Πίνακας 12 : Χαρακτηριστικά OpenLog

| Διαστάσεις |              |
|------------|--------------|
| Πλάτος     | 0.6" (15mm)  |
| Μήκος      | 0.75" (19mm) |
| Ύψος       | 0.16" (4mm)  |

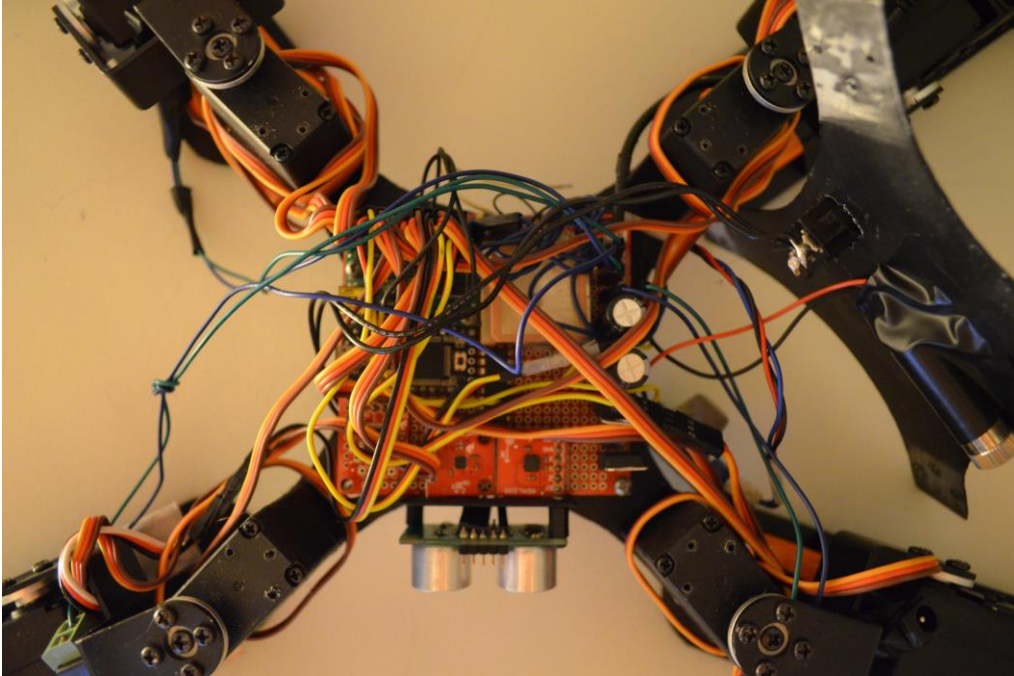
Πίνακας 13 : Διαστάσει OpenLog

| <b>Βασικές Εντολές</b>   |   |
|--------------------------|---|
| <b>Εντολή</b>            | <b>Περιγραφή</b>  |
| <b>new file</b>          | Δημιουργεί ένα νέο αρχείο (όνομα αρχείου) στον τρέχοντα κατάλογο. Μόνο τυπικά ονόματα αρχείων 8.3 υποστηρίζεται. "12345678.123" είναι αποδεκτό. "123456789.123" δεν είναι.  |
| <b>append file</b>       | Προσαρτά κείμενο στο τέλος του αρχείου. Serial δεδομένα στη συνέχεια διαβάζονται από το UART σε ένα ρεύμα και δεν είναι ορατός. Αυτός είναι ο πιο κοινός τρόπος για να χρησιμοποιήσει το καταγραφικό. Για να βγείτε από αυτή την κατάσταση, να στείλετε την εντολή Ctrl + z (ASCII 26) και OpenLog θα επιστρέψει στο «>» εντολών. Αν το αρχείο δεν υπάρχει θα δημιουργηθεί. |
| <b>write file OFFSET</b> | Γράφει το κείμενο στο αρχείο ξεκινώντας από τη θέση OFFSET μέσα στο αρχείο. Το κείμενο διαβάζεται από το UART, γραμμή προς γραμμή, και επανέλαβε πίσω. Για να βγείτε από αυτή την κατάσταση, να στείλει μια κενή γραμμή.  |
| <b>rm file</b>           | Διαγράφει το αρχείο από τον τρέχοντα κατάλογο   |
| <b>size file</b>         | Εμφανίζει το μέγεθος του αρχείου σε bytes.  |
| <b>ls</b>                | Εμφανίζει το περιεχόμενο του τρέχοντος καταλόγου  |
| <b>md directory</b>      | Δημιουργεί ένα υπο κατάλογο στον τρέχοντα κατάλογο  |
| <b>cd directory</b>      | Αλλαγή σε ένα υπο κατάλογο  |
| <b>rm directory</b>      | Αφαιρέστε μια υπο κατάλογο  |
| <b>read file</b>         | Διαβάζει ένα αρχείο   |

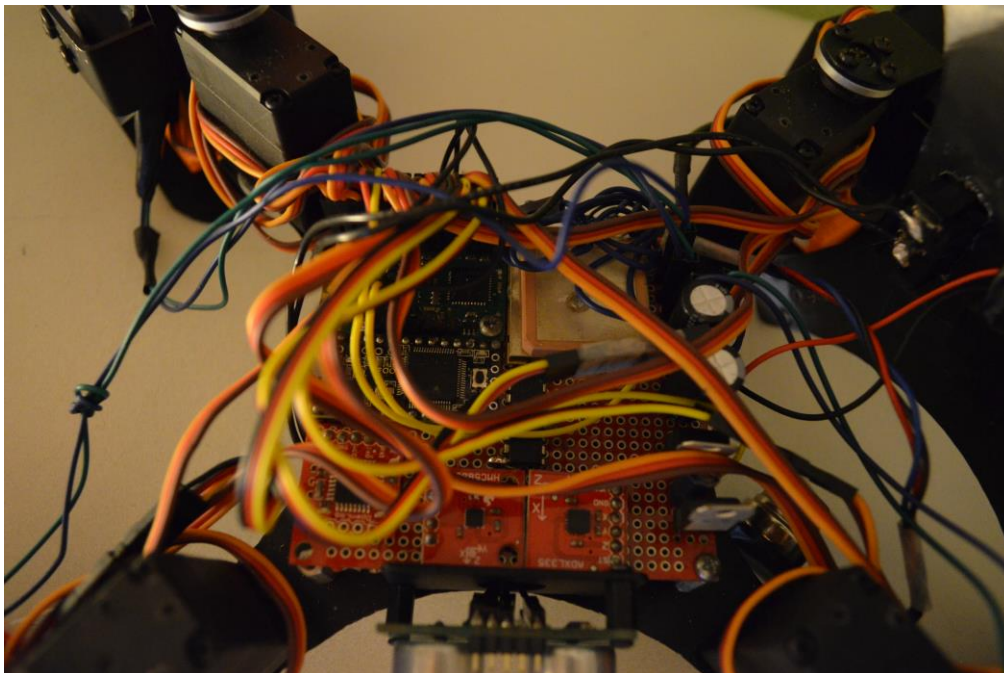
Πίνακας 14 : Βασικές Εντολές OpenLog



### 3.6.10 Συνολικό Κύκλωμα

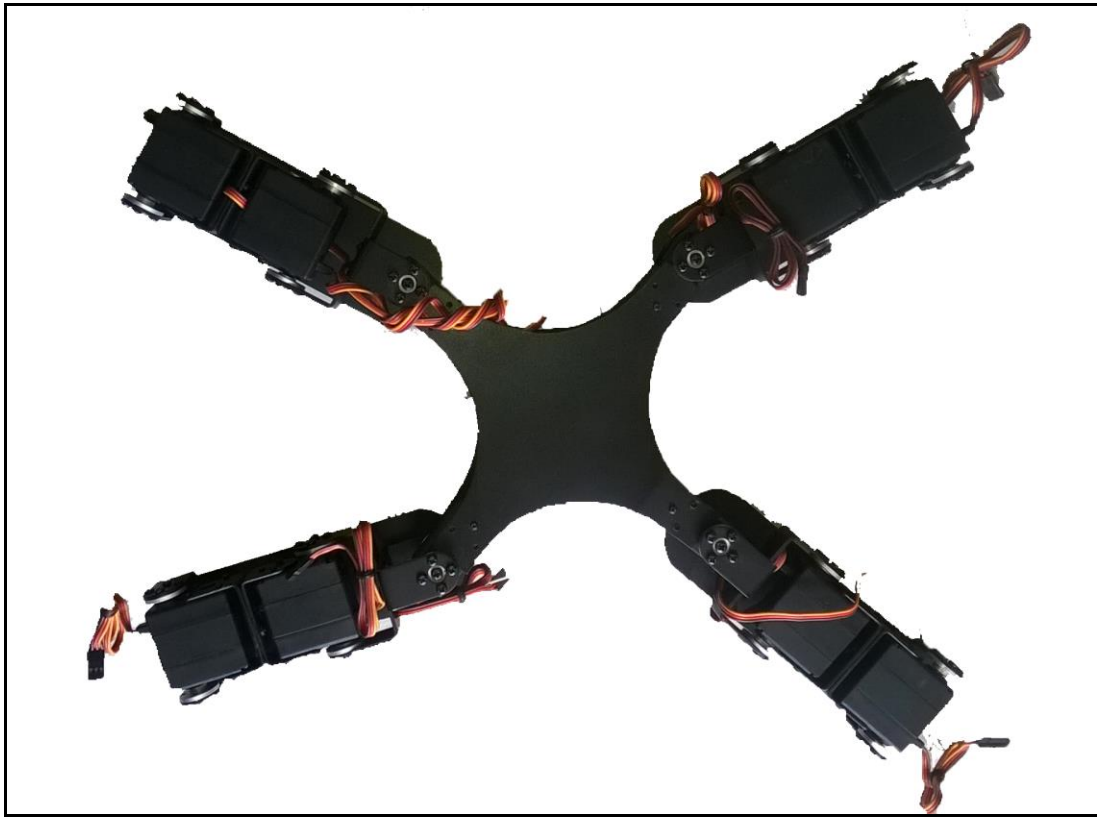


Εικόνα 35 : Ηλεκτρονικό Κύκλωμα (1)

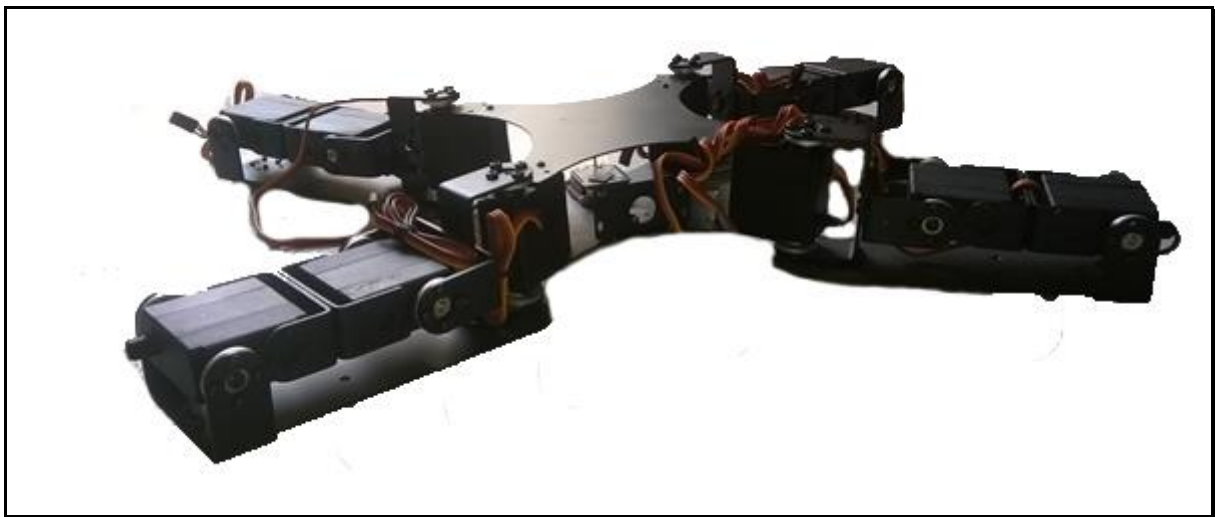


Εικόνα 36 : Ηλεκτρονικό Κύκλωμα (2)

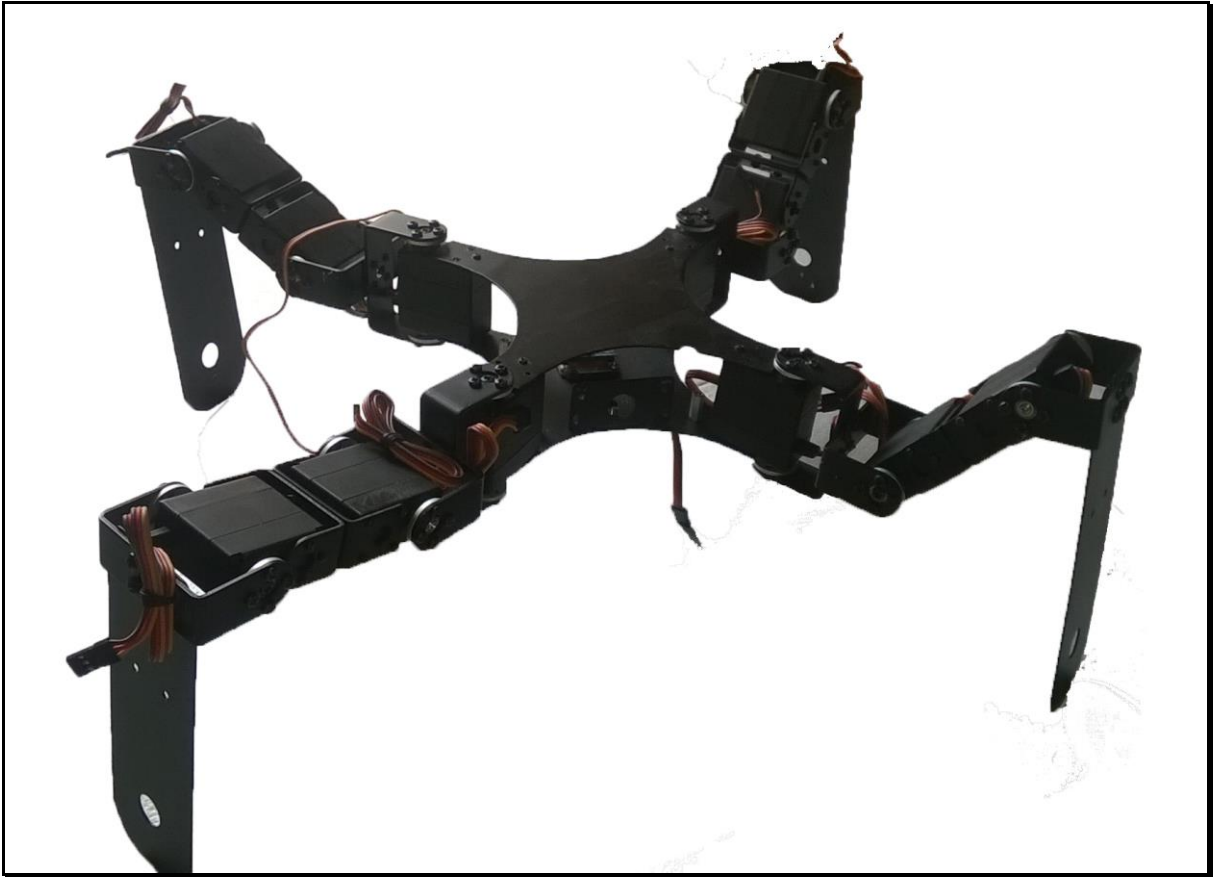
### 3.8 Αποτέλεσμα Μηχανολογικής – Ηλεκτρονικής μελέτης



Εικόνα 37 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (1)



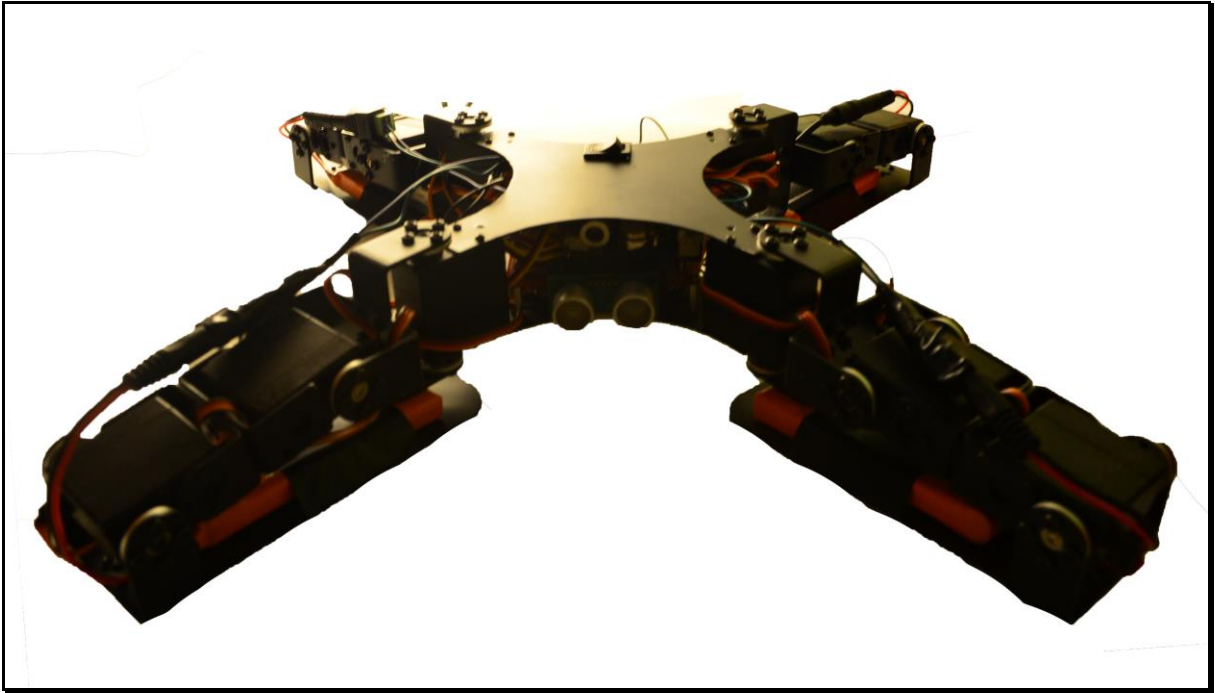
Εικόνα 38 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (2)



Εικόνα 39 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (3)

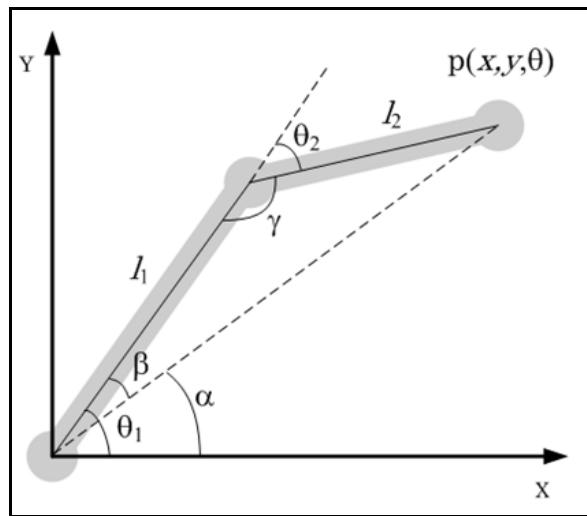


Εικόνα 40 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (4)



Εικόνα 41 : Αποτέλεσμα Μηχανολογικής - Ηλεκτρονικής Μελέτης (5)

## 4. Κινηματική Μελέτη Ρομποτικού Συστήματος



Εικόνα 42 : Κινηματική μελέτη βραχίονα

### Εισαγωγή

Τα κινηματικά προβλήματα που σχετίζονται με ένα ρομποτικό βραχίονα είναι δύο:

**Ευθύ κινηματικό πρόβλημα.** Με δεδομένες τις μεταβλητές των αρθρώσεων ενός ρομποτικού βραχίονα, να προσδιοριστούν η θέση και ο προσανατολισμός του εργαλείου τελικής δράσης του ρομποτικού βραχίονα ως προς το ακίνητο σύστημα συντεταγμένων.

**Αντίστροφο κινηματικό πρόβλημα.** Με δεδομένη τη θέση και τον προσανατολισμό του εργαλείου τελικής δράσης του ρομποτικού βραχίονα, ως προς το ακίνητο σύστημα συντεταγμένων του χώρου εργασίας, να προσδιοριστούν οι μεταβλητές των αρθρώσεων του.

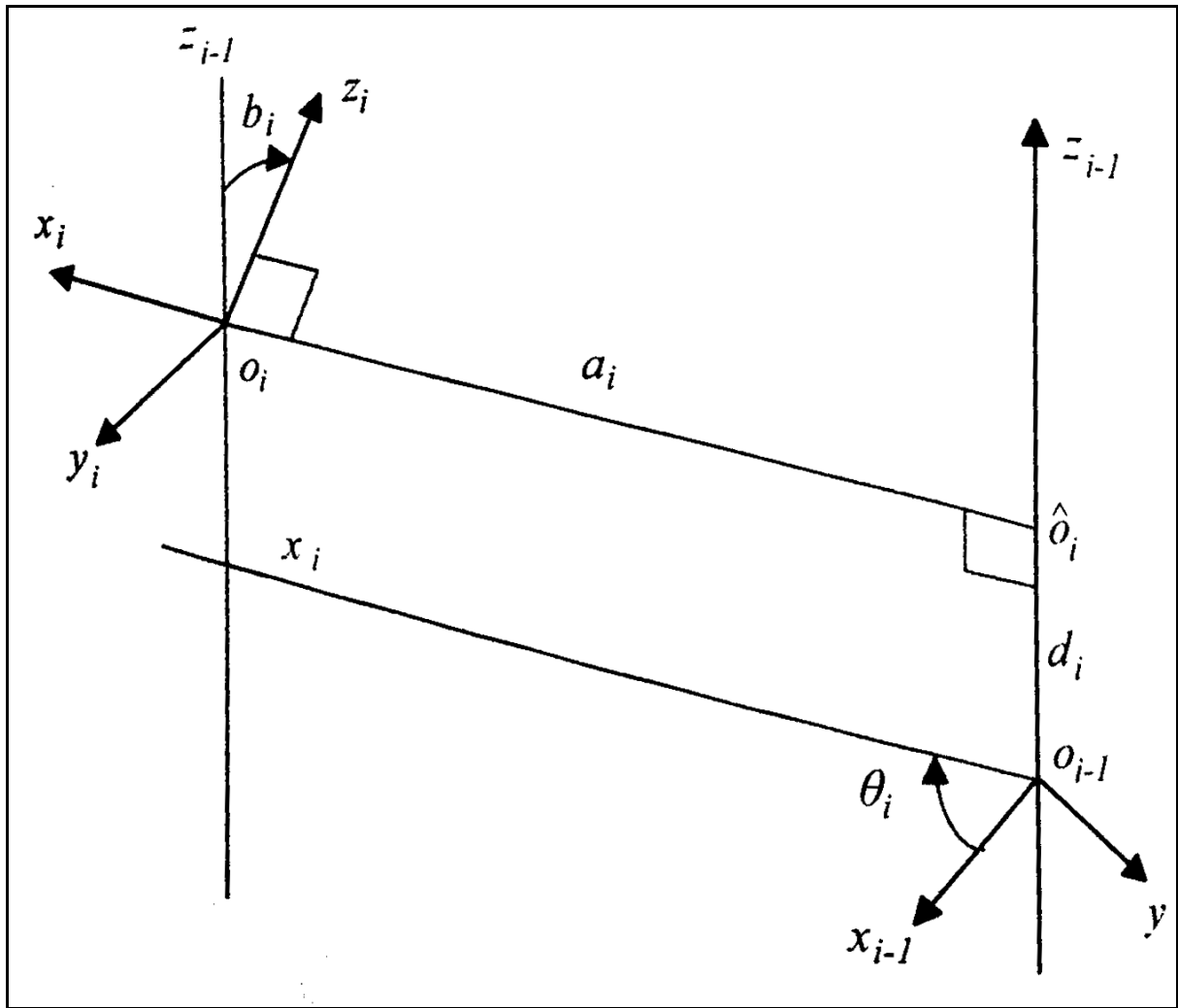
### Κινηματική Αλυσίδα

Η κινηματική ενός ρομπότ μπορεί να αναλυθεί στα επιμέρους κινηματικά προβλήματα των συνδέσμων του. Η αλυσίδα συνδέσμων και αρθρώσεων ενός ρομπότ αντιστοιχεί σε μια αλυσίδα κινηματικών προβλημάτων που καλείται κινηματική αλυσίδα.

Η μελέτη της κινηματικής αλυσίδας στηρίζεται στην ακόλουθη βασική παραδοχή: Κάθε άρθρωση έχει μόνο ένα βαθμό ελευθερίας (γωνιά ή μετατόπιση) που συμβολίζεται με τη μεταβλητή  $Q_i$ . Ο δείκτης  $i$  αντιστοιχεί στη  $i$ -οστή άρθρωση του ρομπότ.

**Σημείωση:** Η παραπάνω παραδοχή δεν περιορίζει τη γενικότητα του προβλήματος, με την έννοια ότι κάθε άρθρωση με βαθμούς ελευθερίας περισσότερους του ενός μπορεί να αναλυθεί σε επιμέρους αρθρώσεις ενός βαθμού ελευθερίας (Για παράδειγμα, μια σφαιρική άρθρωση μπορεί να αναλυθεί σε δυο στρωφικές αρθρώσεις)

Η Μέθοδος Denavit – Hartenberg (D - H) είναι η πιο διαδεδομένη συστηματική μέθοδος που επιλύει το ευθύ κινηματικό πρόβλημα θέσης, δηλαδή προσδιορίζει τη θέση του εργαλείου τελικής δράσης ως προς το χώρο εργασίας. Ο προσδιορισμός της θέσης του εργαλείου τελικής δράσης γίνεται προσδιορίζοντας τις συντεταγμένες ενός προκαθορισμένου σημείου του, έστω A ως προς το χώρο εργασίας. Σε αυτο το σημείο πρέπει να σημειωθεί ότι πριν την εφαρμογή της μεθόδου Denavit – Hartenberg (D - H) είναι αναγκαία η παρουσίαση του ακόλουθου θεωρήματος, το οποίο θέτει τη γεωμετρική άποψη των μεγεθών (D - H).



Εικόνα 43 : Γεωμετρική άποψη μεγεθών D-H

**Θεώρημα:** Έστω δυο δεξιόστροφα ορθοκανονικά συστήματα συντεταγμένων  $\{O_{i-1} X_{i-1} Y_{i-1} Z_{i-1}\}$  και  $\{O_i X_i Y_i Z_i\}$ , τέτοια ώστε ο άξονας  $X_i$  του δεύτερου συστήματος συντεταγμένων να είναι κάθετος στον άξονα  $Z_{i-1}$  του πρώτου συστήματος συντεταγμένων. Ο ομογενής μετασχηματισμός  $H^i_{i-1}$  αναλύεται σαν γινόμενο τεσσάρων μετασχηματισμών

$$H_{i-1}^i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,\alpha_i} Rot_{x,b_i} =$$

$$\begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \alpha_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{b_i} & -s_{b_i} & 0 \\ 0 & s_{b_i} & c_{b_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{b_i} & s_{\theta_i}s_{b_i} & \alpha_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{b_i} & -c_{\theta_i}s_{b_i} & \alpha_i s_{\theta_i} \\ 0 & s_{b_i} & c_{b_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Όπου  $Rot_{(a,b)}$  συμβολίζει τον 4 x 4 ομογενή μετασχηματισμό που αντιστοιχεί σε στροφή  $\beta$  γύρω από τον άξονα  $a$  και  $Trans_{(a,b)}$  συμβολίζει τον 4 x 4 ομογενή μετασχηματισμό που αντιστοιχεί σε μετατόπιση  $\beta$  κατά μήκος του άξονα  $a$ .

Η ανάλυση που παρουσιάστηκε στο αποτέλεσμα του θεωρήματος, εξαρτάται από τέσσερεις μεταβλητές (μεγέθη D - H): Το μήκος  $\alpha_i$ , τη στρέψη  $\beta_i$ , το περιθώριο  $d_i$  και τη γωνιά  $\theta_i$ . Η ερμηνεία των μεγεθών αυτών δίνεται στη παραπάνω εικόνα.

### Ειδικότερα ορίζονται:

|   |   |
|---|---|
| 1 | Η γωνιά $\theta_i$ να είναι η γωνιά μεταξύ των αξόνων $Z_{i-1}$ και $Z_i$ με στροφή γύρω από τον άξονα $Z_{i-1}$ , με φορά από τον άξονα $X_{i-1}$ προς τον άξονα $X_i$ και πρόσημο σύμφωνο με τον κανόνα του δεξιόστροφου κανόνα κοχλίας.      |
| 2 | Το περιθώριο $d_i$ να είναι η μετατόπιση κατά μήκος του άξονα $Z_{i-1}$ από το σημείο $O_{i-1}$ στο σημείο $O_i$ (Σημείο τομής του άξονα $Z_{i-1}$ με το άξονα $X_i$ ), με πρόσημο που καθορίζεται από τη φορά του άξονα $Z_{i-1}$ .            |
| 3 | Το μήκος $\alpha_i$ να είναι η μετατόπιση κατά μήκος του άξονα $X_i$ από το σημείο $O_i$ στο σημείο $O_i$ , με πρόσημο που καθορίζεται από τη φορά του άξονα $X_i$ .  |
| 4 | Η στρέψη $\beta_i$ είναι η γωνιά μεταξύ των αξόνων $Z_{i-1}$ και $Z_i$ με στροφή γύρω από τον άξονα $X_i$ , με φορά από τον άξονα $Z_{i-1}$ προς τον άξονα $Z_i$ και το πρόσημο να είναι σύμφωνο με τον κανόνα του δεξιόστροφου κανόνα κοχλίας. |

Πίνακας 15 : Ερμηνεία μεγεθών D-H

**Σημείωση:** Στο μετασχηματισμό του θεωρήματος δεν εμφανίζεται κανείς ομογενής μετασχηματισμός που να συνδέει με τον άξονα  $y$ . Αυτό οφείλεται στο γεγονός ότι ο μοναδικός προσδιορισμός των αξόνων  $x$  και  $z$  'οδηγεί' αυτόματα και στο μοναδικό του άξονα  $y$ .

Ο αλγόριθμος που ακολουθεί, γνωστός και ως αλγόριθμος D - H θεμελιώθηκε στις εργασίες Denavit και Hartenberg (1955) Uicker (1964) κ.α.



## 4.1 Αλγόριθμος Denavit-Hartenberg

Θεωρώντας ένα ρομποτικό σύστημα  $n+1$  συνδέσμων και  $n$  αρθρώσεων. Ο αλγόριθμος D-H για τον προσδιορισμό των συντεταγμένων ενός σημείου του εργαλείου τελικής δράσης, ως προς το χώρο εργασίας, αποτελείται από τα ακόλουθα 8 βήματα:

|   |  |
|---|--|
| 1 | Επιλέγονται οι άξονες $Z_0, Z_1, Z_2, \dots, Z_n$ των συστημάτων συντεταγμένων κάθε συνδέσμου, ώστε οι $n$ πρώτοι ( $Z_0, Z_1, Z_2, \dots, Z_{n-1}$ ) να διέρχονται από τους άξονες των αντίστοιχων αρθρώσεων 1, ..., $n$ και ο τελευταίος άξονας $Z_n$ να διέρχεται από σημείο, έστω $A$ , του εργαλείου τελικής δράσης, η θέση του οποίου είναι σημαντική για την εφαρμογή που εκτελεί ο ρομποτικός βραχίονας. Η φορά των αξόνων που αντιστοιχούν σε πρισματικές αρθρώσεις επιλέγεται προς τη φορά της στροφής θετικής αύξησης της μεταβλητής άρθρωσης. Η φορά των αξόνων που αντιστοιχούν σε στροφικές αρθρώσεις καθορίζεται από τη φορά της στροφής που αντιστοιχεί σε θετική γωνιά σύμφωνα με τον κανόνα του δεξιόστροφου κανόνα κοχλίας. |
| 2 | Επιλέγεται το σύστημα συντεταγμένων $\{O_0 X_0 Y_0 Z_0\}$ της βάσης ως προς τα $O_0, X_0$ και $Y_0$ . Από το σύστημα αυτό προσδιορίζεται το σύστημα συντεταγμένων του συνδέσμου 1 ακολουθώντας τα βήματα 3 – 5. Επαναληπτικά, υπολογίζεται από το σύστημα συντεταγμένων του συνδέσμου $i$ , ακολουθώντας τα βήματα 3 – 5.  |
| 3 | Σχηματίζεται η ευθεία που είναι κοινή κάθετος στους άξονες $Z_{i-1}$ και $Z_i$ . Το σημείο τομής της ευθείας με τον άξονα $Z_i$ είναι η αρχή των αξόνων $O_i$ του συστήματος συντεταγμένων $i$ . Το αντίστοιχο σημείο τομής με τον άξονα $Z_{i-1}$ συμβολίζεται με $O_{i-1}$ (είναι κ στην παραπάνω εικόνα). Σημειώνεται ότι το κέντρο $O_i$ ορίζεται διαφορετικά μόνο στη περίπτωση που οι άξονες είναι παράλληλοι (ή ταυτίζονται). Τότε, για $i=1, \dots, n-1$ η αρχή του συστήματος συντεταγμένων $i$ ορίζεται να είναι το σημείο τομής του άξονα $Z_i$ με άρθρωση $i+1$ . Αν $i=n$ , το σημείο $O_i$ ταυτίζεται με το σημείο $A$ .   |
| 4 | Καθορίζεται ο άξονας $X_i$ ώστε να βρίσκεται πάνω στη κοινή κάθετο των αξόνων $Z_{i-1}$ και $Z_i$ και να ξεκινά από το σημείο $O_i$ . Όταν οι άξονες $Z_{i-1}$ και $Z_i$ είναι παράλληλοι, ο άξονας $X_i$ έχει φορά από τον άξονα $Z_{i-1}$ προς τον άξονα $Z_i$ . Αν οι άξονες $Z_{i-1}$ και $Z_i$ τέμνονται, τότε ο άξονας $X_i$ ορίζεται στην ευθεία που είναι κάθετη στο επίπεδο ( $Z_{i-1} Z_i$ ) με φορά σύμφωνη με τον κανόνα του δεξιόστροφου κοχλίας, ακολουθώντας τη μικρότερη γωνιά από τον άξονα $Z_{i-1}$ προς τον $Z_i$ .  |
| 5 | Ο άξονας $Y_i$ επιλέγεται ώστε να σχηματίσει με τους άξονες $Z_i$ και $X_i$ δεξιόστροφο ορθογώνιο σύστημα συντεταγμένων.   |
| 6 | Σχηματίζεται για κάθε σύνδεσμο, πίνακας με τα δεδομένα του μήκους $a_i$ , της στρέψης $b_i$ και της γωνίας $\theta_i$ .  |
| 7 | Με βάση τα δεδομένα του βήματος 6, υπολογίζονται για κάθε ζεύγος γειτονικών συνδέσμων οι πίνακες ομογενούς μετασχηματισμού που ορίζονται στον τύπο του θεωρήματος. Με βάση τον τύπο για $i=0, j=n$ υπολογίζεται ο ομογενής μετασχηματισμός $H^n_0$ .   |
| 8 | Υπολογίζεται η θέση του σημείου $A$ όπου $p^*n$ είναι ο $4 \times 1$ πίνακας που περιλαμβάνει τις συντεταγμένες του σημείου $A$ ως προς το σύστημα συντεταγμένων που είναι προσαρμοσμένο στο $n$ -οστό σύνδεσμο και $p^*0$ είναι ο $4 \times 1$ πίνακας που περιλαμβάνει τις γενικευμένες συντεταγμένες του σημείου $A$ ως προς το σύστημα συντεταγμένων που είναι προσαρμοσμένο στη βάση. Συνήθως το σύστημα συντεταγμένων που είναι προσαρμοσμένο στο $n$ -οστό σύνδεσμο επιλέγεται ώστε η αρχή του να ταυτίζεται με το σημείο $A$ , οπότε ισχύει $P^*n = [0 \ 0 \ 0 \ 1]$ .   |

Αλγόριθμος 3 : Λογικά βήματα αλγορίθμου Denavit - Hartenberg

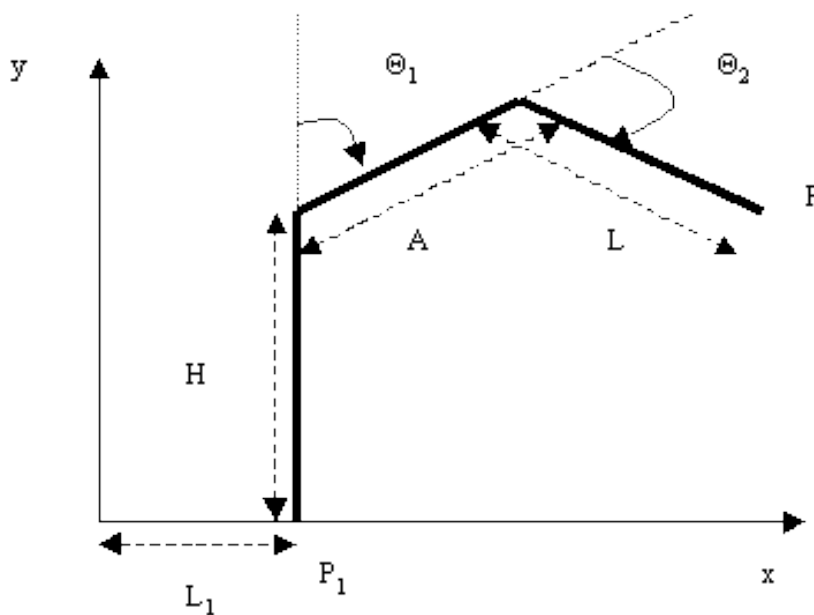


## 4.2 Προσανατολισμός τελικής Θέσης

Σύμφωνα με τον αλγόριθμο D – H που παρουσιάστηκε στις προηγούμενες σελίδες, προσδιορίζεται αναλυτικά η θέση του εργαλείου τελικής δράσης ως προς ακίνητο σύστημα συντεταγμένων στο χώρο εργασίας. Σε πολλές ρομποτικές εφαρμογές για την ορθή εκτέλεση μιας εργασίας απαιτείται εκτός από τη γνώση της θέσης του εργαλείου τελικής δράσης και η γωνιά του προσανατολισμού του. Έτσι προσδιορίζεται όχι μόνο η θέση του εργαλείου τελικής δράσης αλλά και η κατεύθυνση στην οποία αυτό δρα.

Ο προσδιορισμός της θέσης και του προσανατολισμού ενός στερεού σώματος ανάγεται στη γνώση της θέσης τριών σημείων του στερεού σώματος, τα οποία δε βρίσκονται στην ίδια ευθεία. Έστω ότι με  $S_A^*$ ,  $S_B^*$  και  $S_\Gamma^*$  συμβολίζονται οι  $4 \times 1$  πίνακες γενικευμένων συντεταγμένων τριών σημείων A, B και Γ (Τα οποία δεν είναι συνευθειακά) του εργαλείου τελικής δράσης ως προς το n-οστό σύστημα συντεταγμένων. Είναι προφανές ότι εφαρμόζοντας τον αλγόριθμο για καθένα από τα τρία σημεία, προσδιορίζονται οι  $4 \times 1$  πίνακες P\*A, P\*B και P\*Γ των γενικευμένων συντεταγμένων των τριών σημείων του χώρου εργασίας. Έτσι από τους πίνακες P\*A, P\*B και P\*Γ προσδιορίζεται μονοσήμαντα η θέση και ο προσανατολισμός του εργαλείου τελικής δράσης.

Είναι σημαντικό να σημειωθεί ότι σε περίπτωση που η κίνηση του ρομποτικού βραχίονα γίνεται σε δυο διαστάσεις, τότε ο προσανατολισμός του εργαλείου τελικής δράσης έχει νόημα κίνησης μόνο στο επίπεδο κίνησης του ρομποτικού βραχίονα. Στη περίπτωση αυτή αρκεί να επιλεγούν μόνο δυο σημεία του εργαλείου τελικής δράσης, τα οποία να βρίσκονται επάνω στο επίπεδο κίνησης και μετά να εφαρμοστεί ο αλγόριθμος D – H στις γενικευμένες συντεταγμένες και των δυο σημείων.



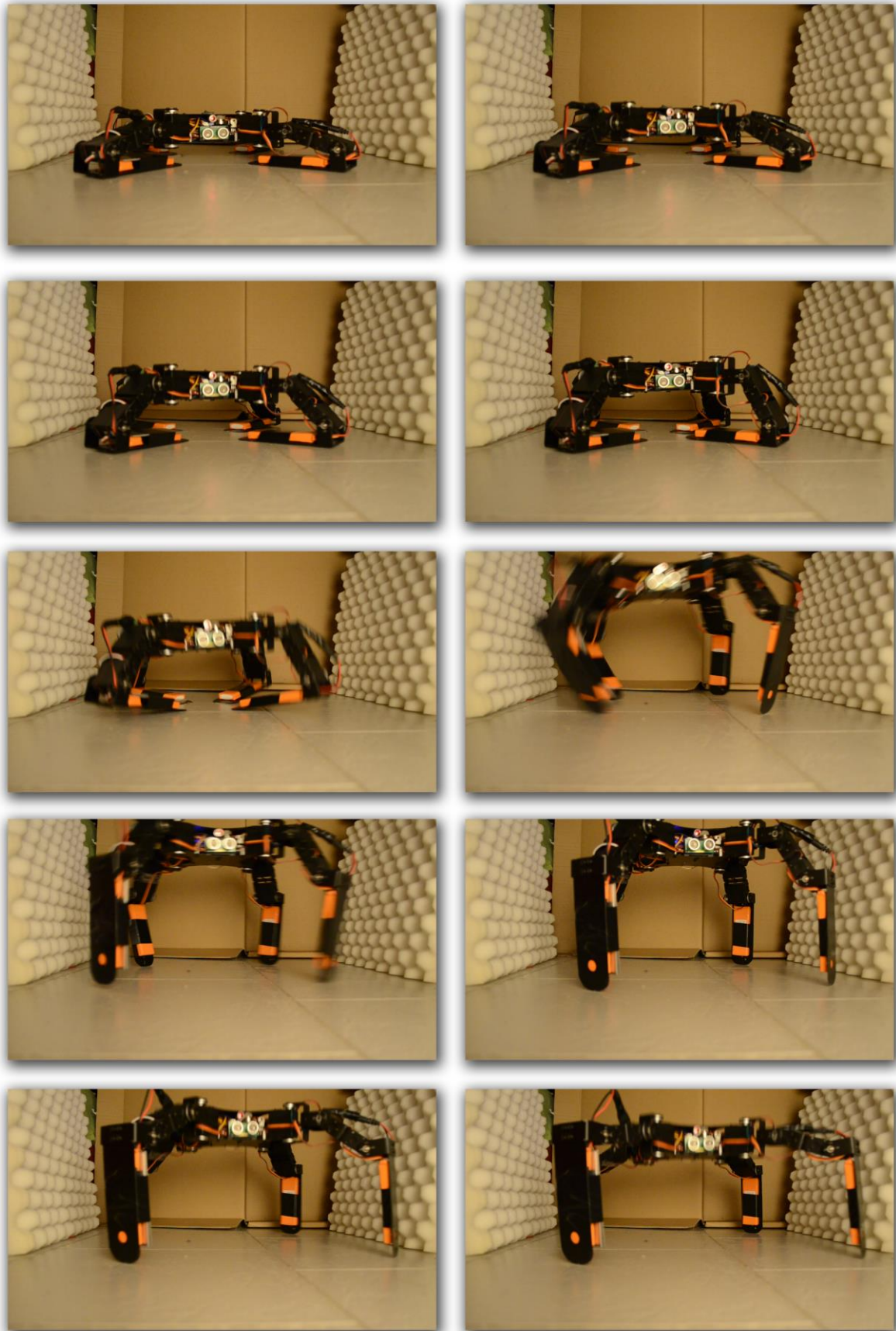
Εικόνα 44 : Προσδιορισμός Άκρου Βραχίονα

## 4.3 Παρουσίαση κίνησης

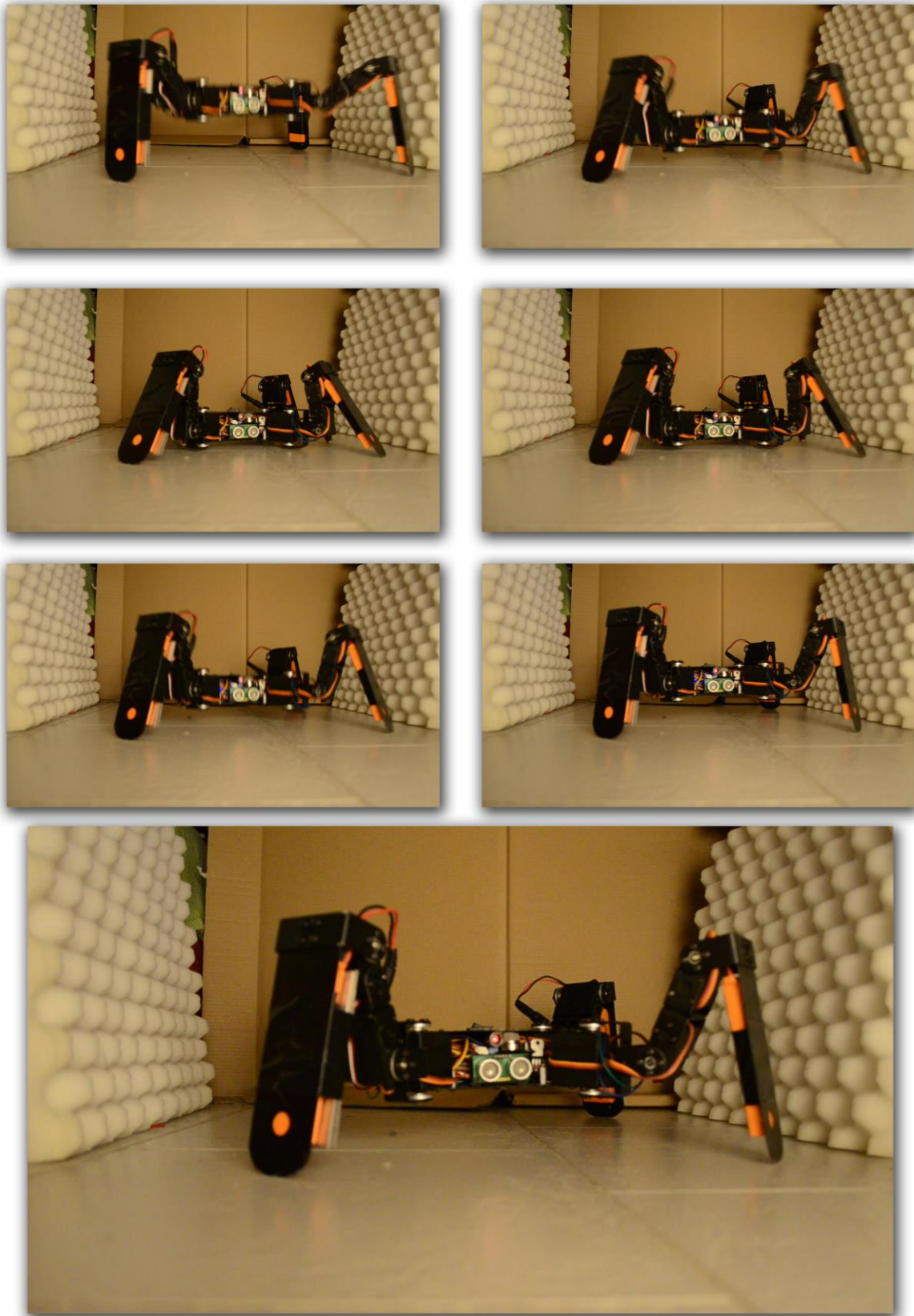
Το κινηματικό μοντέλο που χρησιμοποιούμε είναι βασισμένο στη μέθοδο Denavit - Hartenberg. Με την μέθοδο αυτή, για μία κινηματική αλυσίδα  $n$  αρθρώσεων, έχουμε  $n + 1$  συστήματα συντεταγμένων, με  $O_0$  το σύστημα της βάσης και  $O_n$  το σύστημα της άκρης.

Πρακτικά στο ρομπότ χρησιμοποιήθηκαν τα αποτελέσματα των συναρτήσεων ως προκαθορισμένες θέσεις. Αυτό έγινε για λόγους οικονομίας επεξεργαστικής ισχύος από τον πολλαπλασιασμό πινάκων. Ωστόσο το πρόβλημα που αντιμετωπίζει μια τέτοια κατασκευή είναι η μάζα και η άνιση κατανομή της στα μέρη του σκελετού, όπου περιορίζουν τις θέσεις ισορροπίας και δυσκολεύουν την κίνηση.

Παρακάτω παρουσιάζονται τρεις διαφορετικές κινήσεις :

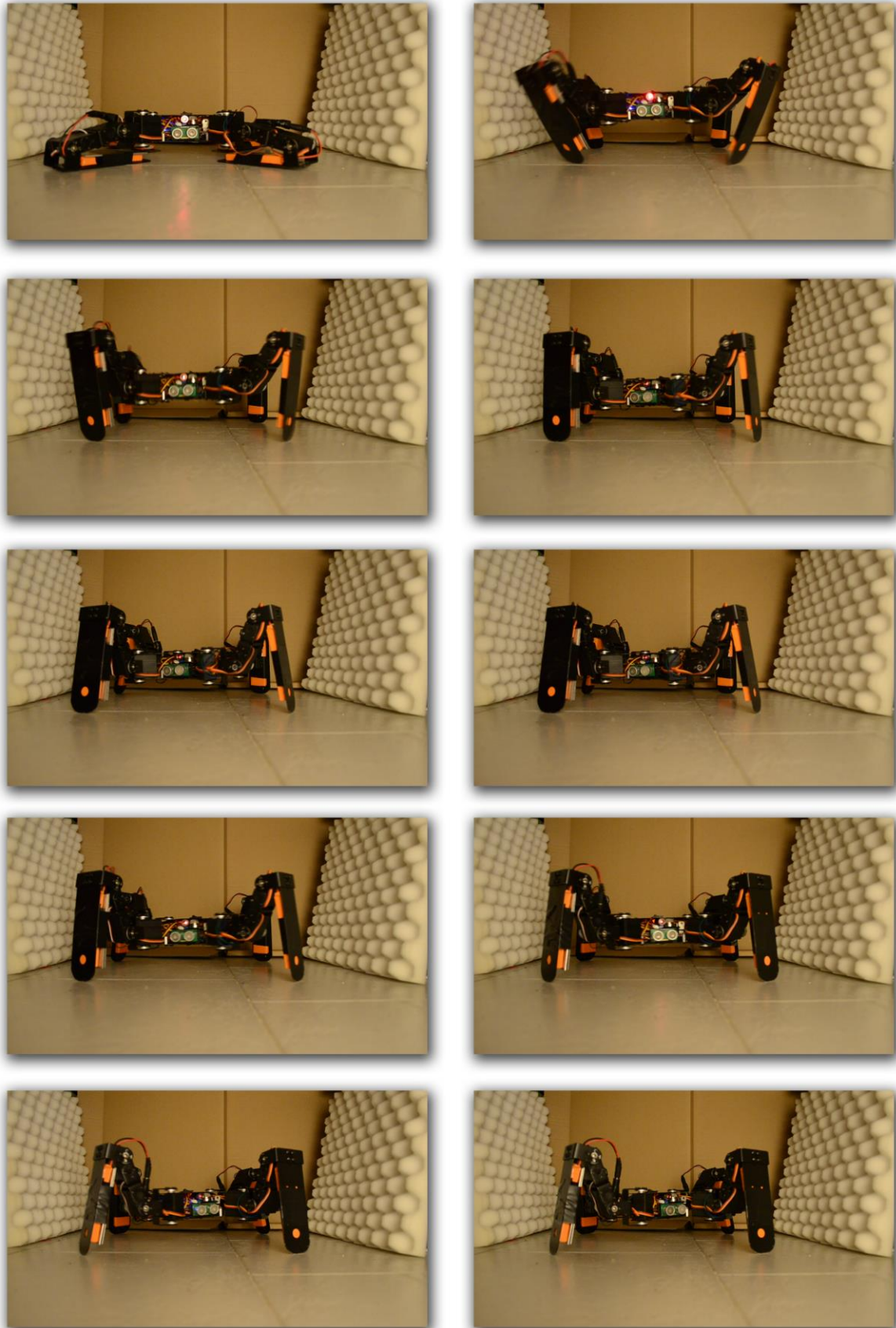


Εικόνα 45 : Timelapse Κινήσεων (1)

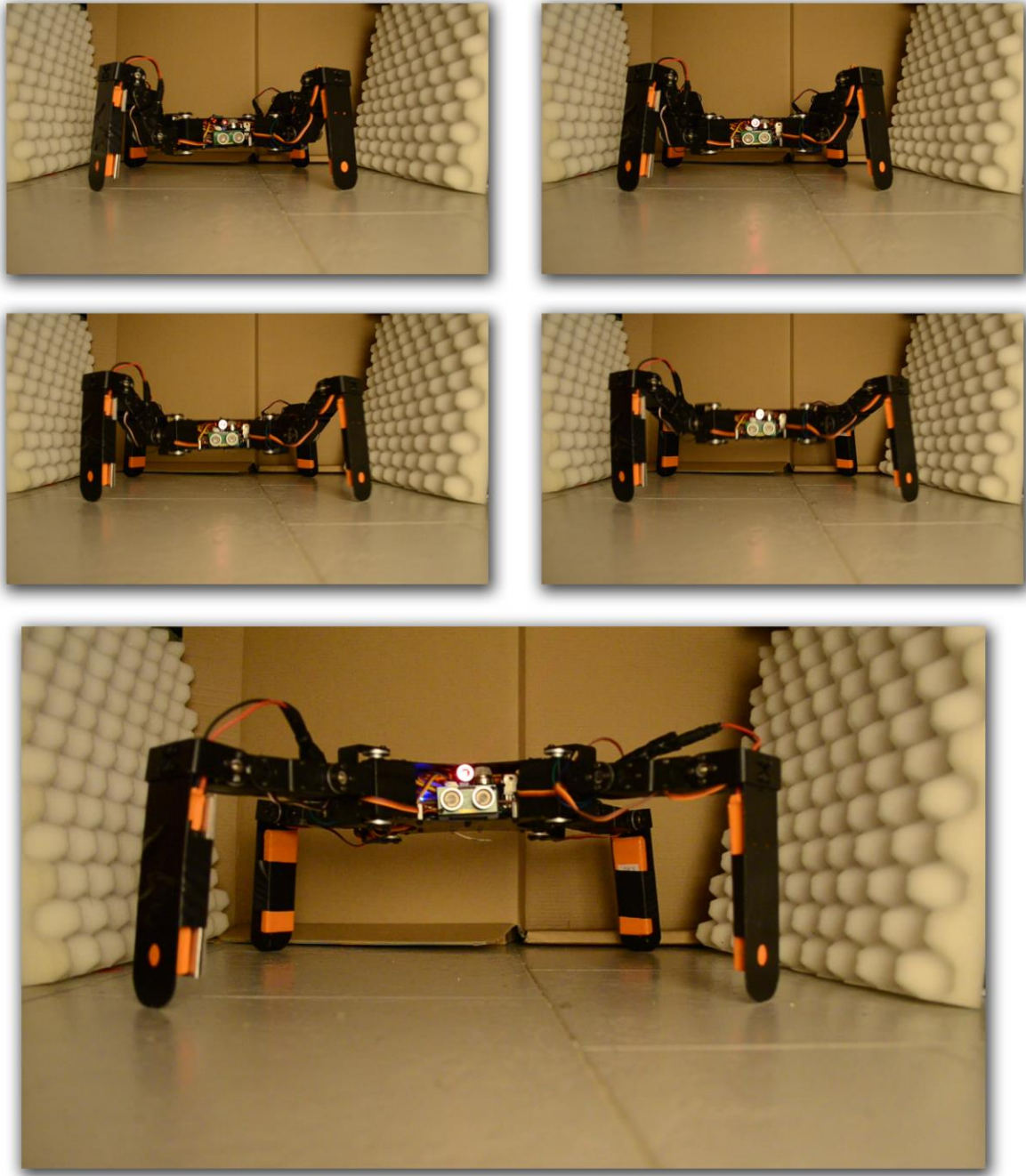


Εικόνα 46 : Timelapse Κινήσεων (2)



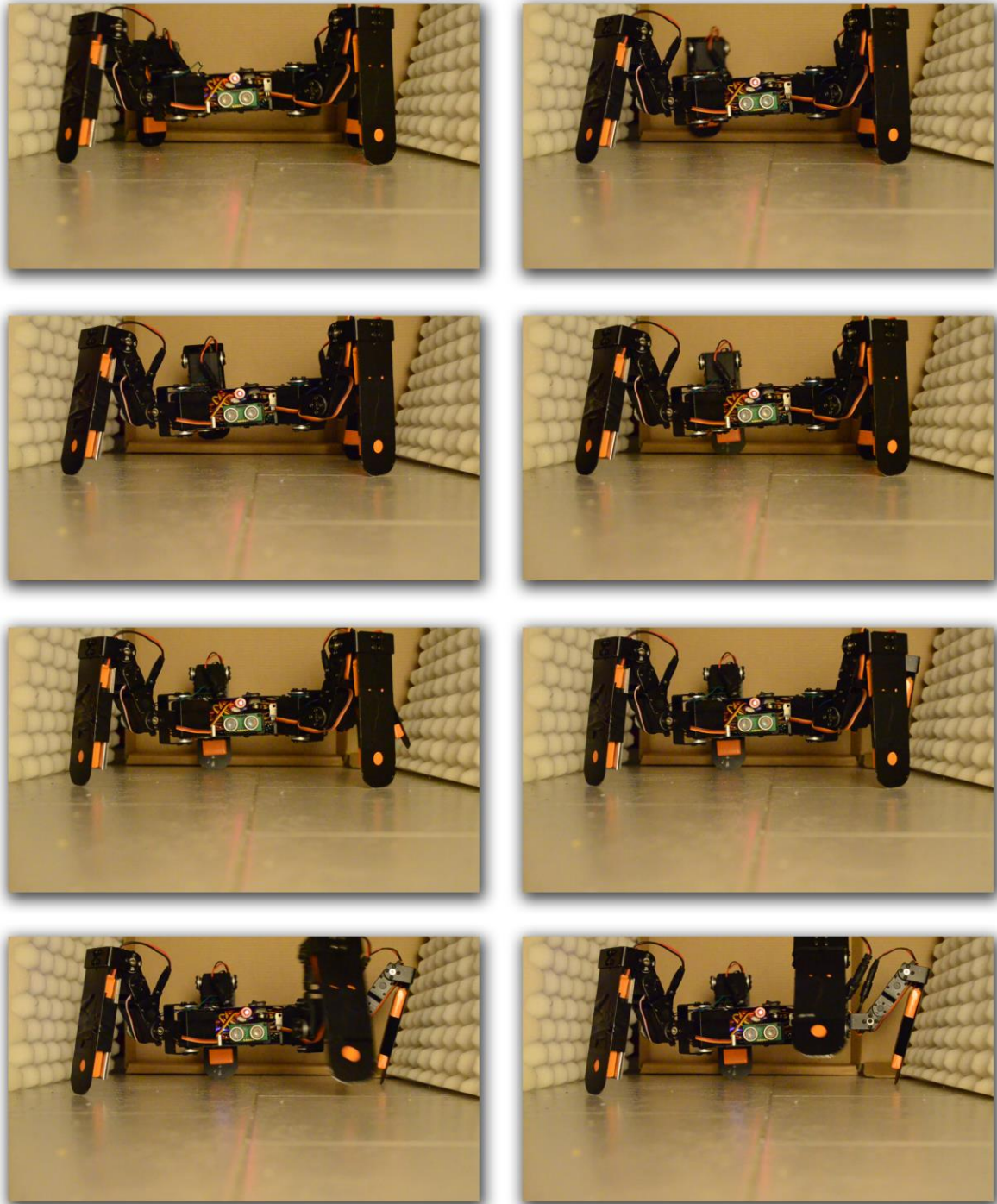


Εικόνα 47 : Timelapse Κινήσεων (3)

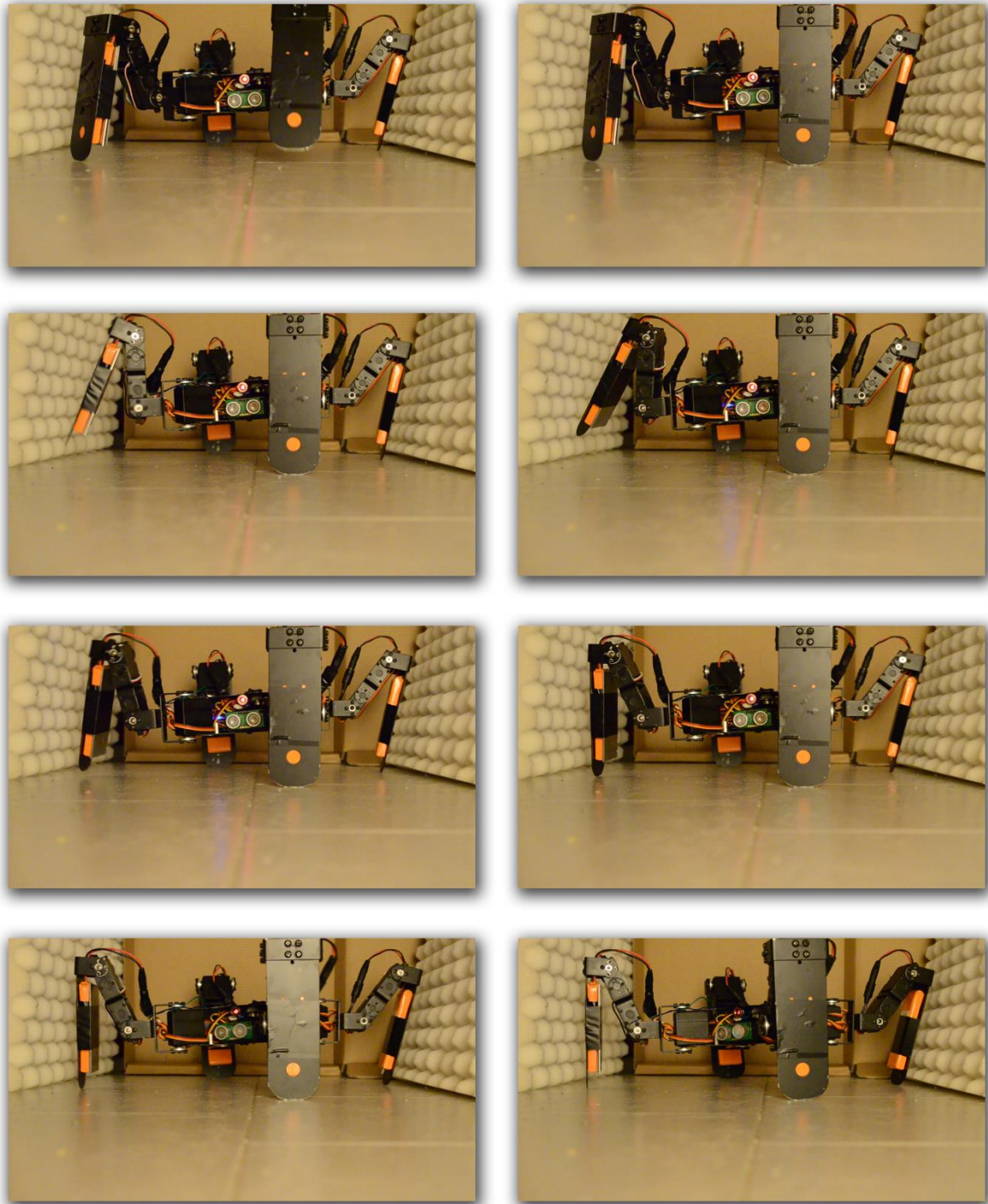


Εικόνα 48 : Timelapse Κινήσεων (4)





Εικόνα 49 : Timelapse Κινήσεων (5)



Εικόνα 50 : Timelapse Κινήσεων (6)



## 5. Ανάλυση Λογισμικού

Μια βασική πτυχή της εργασίας ήταν η ανάλυση και η δημιουργία λογισμικού κατάλληλου να υποστηρίξει πολλές διαφορετικές στο είδος απαιτήσεις. Για την κατανόηση, τον προσδιορισμό και την έκφραση των απαιτήσεων από το λογισμικό είναι απαραίτητο ένα ιδεατό μοντέλο (conceptual model) των διεργασιών του συστήματος στο οποίο θα λειτουργήσει το λογισμικό.

Σύνθετες κατασκευές, όπως το λογισμικό, απαιτούν τη δημιουργία ενός σχεδίου πριν αρχίσει η πραγματική υλοποίηση. Το σχέδιο αυτό βοηθά στην ελαχιστοποίηση των διορθώσεων και των αλλαγών κατά τη διάρκεια της κωδικοποίησης. Ο σχεδιασμός του λογισμικού βασίζεται στη δόμησή του σε μακροσκοπικό (σύστημα, υποσυστήματα, προγράμματα) και σε μικροσκοπικό (σύνθετες δομικές μονάδες, απλές δομικές μονάδες) επίπεδο. Η σχεδίαση ενός συστήματος ξεκινά από το αρχιτεκτονικό σχέδιο το οποίο παριστάνει τη μακροσκοπική δομή του λογισμικού. Σε επόμενο στάδιο ακολουθεί η λεπτομερής σχεδίαση η οποία παριστάνει τη μικροσκοπική δομή του λογισμικού. Η σχεδίαση βασίζεται πάνω στα αποτελέσματα της ανάλυσης των απαιτήσεων του συστήματος.

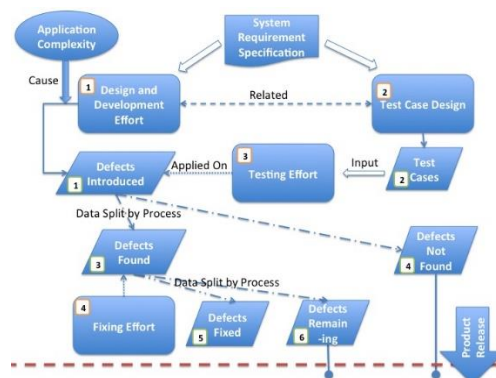
Το συγκεκριμένο μέρος της εργασίας (Ανάλυση Λογισμικού) χρειάστηκε να χωριστεί σε 2 μεγάλες υποκατηγορίες :

### Λογισμικό ρομποτικής πλατφόρμας

Όντας ένα μεγάλο πεδίο χωρίστηκε σε 3 μικρότερα κομμάτια όπως το λειτουργικό σύστημα, τους οδηγούς συσκευών και τέλος τις εφαρμογές. Ένας πολύ σημαντικός παράγοντας αυτής της κατηγοριοποίησης είναι η εύκολη αποσφαλμάτωση και ο γρήγορος και διακριτός υπολογισμός αποδόσεων. Αναλυτικά για κάθε μέρος παρουσιάζονται πληροφορίες στα επόμενα κεφάλαια.

### Λογισμικό Υπολογιστή

Το λογισμικό που χρειάστηκε να δημιουργηθεί στον υπολογιστή είναι μία εφαρμογή με πολύ καθορισμένες απαιτήσεις και μικρότερης πολυπλοκότητας με αποτέλεσμα να μπορέσουμε να το θεωρήσουμε ως ένα σύνολο και να μην διαχωριστεί σε υποδιαίστερα τμήματα.



## 5.1 Λογισμικό Ρομποτικής Πλατφόρμας

Όπως προαναφέρθηκε μια από τις δύο κεντρικές κατηγορίες Λογισμικού είναι το λογισμικό της ρομποτικής πλατφόρμας η οποία χρειάστηκε να διασπαστεί σε 3 τμήματα.

### Λειτουργικό σύστημα

Περιλαμβάνει όλες τις βασικές λειτουργίες που χρειάζεται το Λειτουργικό Σύστημα του ρομπότ, όπως βιβλιοθήκες αρχικοποίησης και μεθόδους διαχείρισης εφαρμογών και events.

### Οδηγοί συσκευών

Αποτελείται από αρχεία τα οποία αναφέρονται στην επικοινωνία και την ανάλυση των δεδομένων των διαφόρων συσκευών (αισθητήρες κτλ.) που υπάρχουν στο ρομπότ. Μέσω αυτών των οδηγών οι εφαρμογές έχουν πρόσβαση σε αυτά τα δεδομένα. Οι οδηγοί συνδέονται άμεσα με το λειτουργικό σύστημα και αποτελούν βασικό μέρος του τρόπου λειτουργίας του ρομπότ.

### Εφαρμογές

Αναφέρονται σε οποιαδήποτε λειτουργία – ανάλυση – απόφαση παρθεί από την ρομποτική πλατφόρμα. Σκοπός τους είναι η συλλογή και αξιοποίηση των δεδομένων για την λήψη αποφάσεων. Δεν έχουν καμία άμεση συσχέτιση με το λειτουργικό σύστημα. Συνήθως λειτουργούν κατά ομάδες με αποτέλεσμα τον διακριτό ρόλο τους σε κάθε ενέργεια.

### Αναλυτικά τα αρχεία που αποτελούν κάθε τμήμα :

| <b>Λειτουργικό σύστημα : [5.1.1 Λειτουργικό Σύστημα]</b> |   |
|--|---|
| Kernel.h   | Περιλαμβάνει τις βασικές λειτουργίες του πυρήνα του λειτουργικού συστήματος. Καθώς και την ενεργοποίηση των Timers. |
| Kernel.cpp   |   |
| Kernel_Applications.h                                    | Αποτελείται από τις μεθόδους και τις αρχικοποιήσεις της διαχείρισης των εφαρμογών από το λειτουργικό σύστημα.       |
| Kernel_Applications.cpp                                  |   |
| Kernel_Clock.h   | Διαχείριση του χρόνου λειτουργίας της ρομποτικής πλατφόρμας.  |
| Kernel_Clock.cpp   |   |
| Kernel_Definitions.h                                     | Αρχικοποιήσεις που χρειάζεται το λειτουργικό σύστημα.   |
| Kernel_Definitions.cpp                                   |   |
| Kernel_Events.h  | Αποτελείται από τις μεθόδους και τις αρχικοποιήσεις της διαχείρισης των events από το λειτουργικό σύστημα.          |
| Kernel_Events.cpp  |   |
| Kernel_Scheduler.h                                       | Διαχειρίζεται την εκτέλεση των εφαρμογών  |
| Kernel_Scheduler.cpp                                     |   |

Πίνακας 16 : Αρχεία πηγαίου κώδικα Λειτουργικού Συστήματος

| <b>Οδηγοί συσκευών : [5.1.2 Οδηγοί Υλικού]</b> |   |
|--|---|
| Hardware.h                                     | Βασικές αρχικοποιήσεις για τους οδηγούς των συσκευών          |
| Hardware.cpp                                   |   |
| Hardware_Accelerometer.h                       | Οδηγός Συσκευής του accelerometer                             |
| Hardware_Accelerometer.cpp                     |   |
| Hardware_Compass.h                             | Οδηγός Συσκευής μαγνητόμετρου                                 |
| Hardware_Compass.cpp                           |   |
| Hardware_Gas.h                                 | Οδηγός Συσκευής αισθητήρα αέρος                               |
| Hardware_Gas.cpp                               |   |
| Hardware_GPS.h                                 | Οδηγός Συσκευής GPS   |
| Hardware_GPS.cpp                               |   |
| Hardware_SDCard.h                              | Οδηγός Συσκευής για την διαχείριση του αποθηκευτικού χώρου SD |
| Hardware_SDCard.cpp                            |   |
| Hardware_Servocontroller.h                     | Οδηγός Συσκευής για το Servocontroller                        |
| Hardware_Servocontroller.cpp                   |   |
| Hardware_Ultrasonic.h                          | Οδηγός Συσκευής Ultrasonic                                    |
| Hardware_Ultrasonic.cpp                        |   |
| Hardware_XBee.h                                | Οδηγός Συσκευής επικοινωνίας                                  |
| Hardware_XBee.cpp                              |   |

Πίνακας 17 : Αρχεία πηγαίου κώδικα Οδηγών Συσκευών

| <b>Εφαρμογές : [5.1.3 Εφαρμογές]</b> |  |
|--------------------------------------|--|
| Software.h                           | Βασικές αρχικοποιήσεις για τις εφαρμογές.  |
| Software.cpp                         |  |
| Software_Sensors.h                   | Εφαρμογή διαχείρισης των αισθητήρων : GAS , Ultrasonic , GPS , Accelerometer , Compass |
| Software_Sensors.cpp                 |  |
| Software_SDCard.h                    | Εφαρμογή διαχείρισης ανάκτησης - αποθήκευσης δεδομένων                                 |
| Software_SDCard.cpp                  |  |
| Software_Servocontroller.h           | Εφαρμογή διαχείρισης κινήσεων  |
| Software_Servocontroller.cpp         |  |
| Olimpia.h                            | Εφαρμογή επικοινωνίας  |
| Olimpia.cpp                          |  |

Πίνακας 18 : Αρχεία πηγαίου κώδικα Εφαρμογών

## 5.1.1 Λειτουργικό Σύστημα

Λειτουργικό σύστημα ή ΛΣ (αγγλ. Operating System ή OS) ονομάζεται στην επιστήμη της πληροφορικής το λογισμικό του υπολογιστή που είναι υπεύθυνο για τη διαχείριση και τον συντονισμό των εργασιών, καθώς και την κατανομή των διαθέσιμων πόρων. Το λειτουργικό σύστημα παρέχει ένα θεμέλιο, ένα μεσολαβητικό επίπεδο λογικής διασύνδεσης μεταξύ λογισμικού και υλικού, διαμέσου του οποίου οι εφαρμογές αντιλαμβάνονται εμμέσως τον υπολογιστή. Μια από τις κεντρικές αρμοδιότητες του λειτουργικού συστήματος είναι η διαχείριση του υλικού, απαλλάσσοντας έτσι το λογισμικό του χρήστη από τον άμεσο και επίπονο χειρισμό του υπολογιστή και καθιστώντας ευκολότερο τον προγραμματισμό τους. Σχεδόν όλοι οι υπολογιστές (παλάμης, επιτραπέζιοι, υπερυπολογιστές, ακόμη και παιχνιδιομηχανές) χρησιμοποιούν έναν τύπο λειτουργικού συστήματος.

Αναλυτικότερα ως λειτουργικό σύστημα (ΛΣ) χαρακτηρίζεται μία συλλογή βασικών προγραμμάτων, η οποία ελέγχει τη λειτουργία του υπολογιστή συνολικά και χρησιμοποιείται ως υπόβαθρο για την εκτέλεση όλων των υπόλοιπων προγραμμάτων, τη διαχείριση των περιφερειακών συσκευών και την εξασφάλιση της επικοινωνίας μεταξύ χρήστη και υπολογιστή. Στην πράξη πρόκειται για ένα επίπεδο λογισμικού που μεσολαβεί μεταξύ του υλικού και των εκτελούμενων προγραμμάτων σε έναν ηλεκτρονικό υπολογιστή. Αποτελείται από ένα σύνολο μηχανισμών μέσω των οποίων επιτυγχάνεται αυτόματη διαχείριση των πόρων ενός υπολογιστή και ελεγχόμενη κατανομή τους στις εκτελούμενες εφαρμογές, έτσι ώστε οι τελευταίες να είναι σε θέση να προσπελάσουν εύκολα τους πόρους και τις συσκευές του συστήματος χωρίς να χρειάζεται να γνωρίζουν με ακρίβεια τη δομή του υποκείμενου υλικού, αλλά και ώστε πολλαπλές εφαρμογές να μπορούν να εκτελούνται ταυτόχρονα χωρίς να έρχονται σε διένεξη μεταξύ τους ή με τον υπολογιστή.

Το λειτουργικό σύστημα περιέχει κώδικα χαμηλού επιπέδου, αποκλειστικά για την αρχιτεκτονική του επεξεργαστή στην οποία εκτελείται (γραμμένο είτε σε κάποια κατάλληλη γλώσσα προγραμματισμού υψηλού επιπέδου, όπως η C, είτε απευθείας σε συμβολική γλώσσα), που αναλαμβάνει την υλοποίηση όλων αυτών των μηχανισμών. Ο εν λόγω κώδικας, ο οποίος κατά την κανονική λειτουργία του υπολογιστή είναι αποθηκευμένος στη μνήμη, ονομάζεται «πυρήνας» και ο κώδικας των εκτελούμενων εφαρμογών («κώδικας χρήστη») δεν έχει άμεση πρόσβαση σε αυτόν. Ο πυρήνας όμως παρέχει μία προγραμματιστική διασύνδεση, τις κλήσεις συστήματος που προαναφέρθηκαν, με την οποία τα προγράμματα χρήστη μπορούν να καλούν με ελεγχόμενο τρόπο διαδικασίες που εξάγει ο πυρήνας και παρέχουν υπηρεσίες στον καλούντα.

Η σύνταξη, η σημασιολογία και η ονοματολογία των κλήσεων συστήματος διαφέρει από ΛΣ σε ΛΣ, με αποτέλεσμα ένα εκτελέσιμο πρόγραμμα συνήθως να μπορεί να τρέξει μόνο σε ένα ΛΣ. Ο κώδικας χρήστη δεν μπορεί να προσπελάσει μόνος του τους πόρους του υπολογιστή (π.χ. δίσκους, μνήμη συστήματος, περιφερειακά, δίκτυο κλπ.) αλλά μπορεί να ζητήσει ό,τι χρειάζεται (π.χ. άνοιγμα ενός αρχείου στον δίσκο) από τον πυρήνα μέσω των κλήσεων συστήματος.

Οι βασικοί μηχανισμοί ενός λειτουργικού συστήματος αφορούν τη διαχείριση της εκτέλεσης των προγραμμάτων χρήστη (μέσω του μηχανισμού των διεργασιών και - στα συστήματα τα οποία

υποστηρίζουν **ταυτοχρονισμό / πολυδιεργασία / πολυπρογραμματισμό** - του κατάλληλου χρονοπρογραμματισμού τους), της επικοινωνίας μεταξύ τους (μέσω των μηχανισμών εικονικής μνήμης και διαδιεργασιακής επικοινωνίας) και των δεδομένων που αυτές χρησιμοποιούν (μέσω του συστήματος αρχείων). Με τον τρόπο που είναι υλοποιημένοι οι εν λόγω μηχανισμοί διασφαλίζουν συνήθως τόσο την ασφάλεια του πυρήνα απέναντι στον κώδικα χρήστη, όσο και την ασφάλεια μεταξύ των προγραμμάτων χρήστη, ώστε κανένα να μην παρεμποδίζει ή να επηρεάζει αρνητικά την πρόσβαση των άλλων στους πόρους του συστήματος.

## **Τι Είναι Ενσωματωμένα Συστήματα**

Ένα ενσωματωμένο λειτουργικό σύστημα είναι ένα λειτουργικό σύστημα για ενσωματωμένα συστήματα υπολογιστών. Αυτά τα λειτουργικά συστήματα έχουν σχεδιαστεί για να είναι συμπαγής, αποδοτικά στην χρήση των πόρων και αξιόπιστα, εγκαταλείποντας πολλές λειτουργίες που παρέχουν μη ενσωματωμένα λειτουργικά συστήματα ηλεκτρονικών υπολογιστών, τα οποία δεν μπορούν να χρησιμοποιηθούν από τις εξειδικευμένες εφαρμογές που τρέχουν. Συχνά αναφέρονται επίσης ως λειτουργικά συστήματα πραγματικού χρόνου, και ο όρος RTOS χρησιμοποιείται συχνά ως συνώνυμο.

Συνήθως, το υλικό τρέχει ένα ενσωματωμένο λειτουργικό σύστημα είναι πολύ περιορισμένες σε πόρους όπως η μνήμη RAM και ROM συνέπεια, τα συστήματα γίνονται για ενσωματωμένο hardware τείνουν να είναι πολύ συγκεκριμένα, πράγμα που σημαίνει ότι λόγω των διαθέσιμων πόρων (χαμηλό σε σύγκριση με μη ενσωματωμένα συστήματα).αυτά Τα συστήματα δημιουργήθηκαν για να καλύψουν συγκεκριμένες εργασίες ή πεδία.

Η επιλογή της επίλυσης ενός βασικού προβλήματος για την σωστή λειτουργία ενός ρομπότ είναι σαφώς ο διαμοιρασμός του χρόνου σε πολλές λειτουργίες.

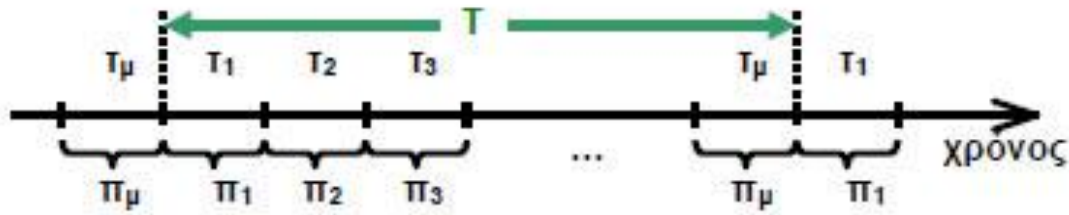
Αναλύοντας τις απαιτήσεις που έχει ένα τέτοιο σύστημα μπορούμε ευθέως να παρατηρήσουμε κάποιες ιδιομορφίες σε σχέση με τα γνωστά συστήματα (πχ. Υπολογιστής γραφείου-H/Y) ,και να καταλήξουμε σε ένα λειτουργικό σύστημα το οποίο βασίζεται σε πραγματικό χρόνο.

Είναι ένα multitasking λειτουργικό σύστημα που στοχεύει στην εκτέλεση εφαρμογών πραγματικού χρόνου, χρησιμοποιώντας εξειδικευμένους αλγόριθμους χρονοπρογραμματισμού, ώστε να πετύχουμε ένα ντετερμινιστικό χαρακτήρα συμπεριφοράς.

Το κύριο χαρακτηριστικό του λειτουργικού είναι η γρήγορη και προβλέψιμη αντίδρασή του στα διάφορα events ,όπως και είτε να είναι driven-events είτε time-sharing σχεδιασμένο.

## **Λειτουργικό σύστημα I026OSv0.3**

Το συγκεκριμένο λειτουργικό σύστημα στηρίζεται στην φιλοσοφία των ενσωματωμένων συστημάτων προσπαθώντας ταυτόχρονα να μιμηθεί βασικές χαρακτηριστικές λειτουργίες των λειτουργικών συστημάτων καταμερισμού χρόνου. Δηλαδή να παρέχει την δυνατότητα υποστήριξης driven-events και time-sharing σχεδιασμού. Στην ουσία είναι ένα πρότυπο παράδειγμα λειτουργίας ενός τέτοιου συστήματος σε Cortex M4 επεξεργαστή.



Εικόνα 51 : Καταμερισμός Χρόνου

Στον μικροεπεξεργαστή υπάρχει ένα ειδικό κύκλωμα, ο χρονιστής (timer), το οποίο σε τακτά χρονικά διαστήματα δημιουργεί ένα σήμα διακοπής (interrupt).

Χρησιμοποιούνται 2 χρονιστές για την κάλυψη των αναγκών :

- Ο βασικός χρονιστής εφαρμογών, ο οποίος είναι μία συνάρτηση ελέγχου και ανανέωσης των διαφόρων events κλπ.
- Ο χρονιστής περιφερειακών ,χρησιμοποιείται για την ανανέωση των τιμών που δεχόμαστε από τις διάφορες συσκευές που είναι συνδεδεμένες στο κύκλωμα.

### 5.1.1.1 Διεργασίες

Μία Διεργασία μπορεί να έχει μία από τις ακόλουθες καταστάσεις:

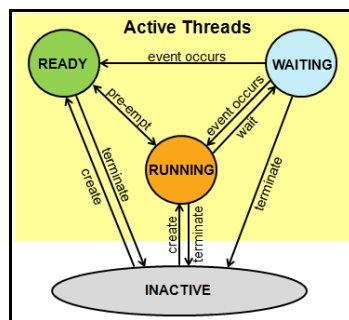
**RUNNING:** Η διεργασία που εκτελείται αυτή τη στιγμή βρίσκεται σε κατάσταση RUNNING. Μόνο ένα νήμα κάθε στιγμή μπορεί να είναι σε αυτή την κατάσταση.

**READY:** Διεργασίες οι οποίες είναι έτοιμες για να εκτελεστούν είναι στην κατάσταση READY. Μόλις το νήμα που εκτελείται λήξει η επόμενη σε σειρά διεργασία (αναλόγως και την προτεραιότητα) εκτελείται.

**WAITING:** Διεργασίες που περιμένουν να συμβεί ένα γεγονός είναι σε κατάσταση αναμονής.

**INACTIVE:** Διεργασίες που δεν έχουν δημιουργηθεί ή ολοκληρωθεί βρίσκονται σε ανενεργή κατάσταση. Συνήθως δεν καταναλώνουν πόρους του συστήματος.

Το επόμενο σχεδιάγραμμα παρουσιάζει τον τρόπο λειτουργίας των διεργασιών.



**Πρότυπο Διεργασίας :**

```

namespace Software
{
    namespace Sensors
    {
        //-----;

        uint8_t getName(void);

        //-----;

        void Initialize_();

        //-----;

        void Main_(uint8_t);

        //-----;

        char* Version(void);

        //-----;

        void OtherFunctions(void);

    }
}

```

**Επεξήγηση :**

Κάθε διεργασία αποτελείται από τέσσερις βασικές συναρτήσεις ,την συνάρτηση επιστροφής ονόματος, την συνάρτηση επιστροφής της έκδοσης, την συνάρτηση αρχικοποίησης και τέλος την κεντρική συνάρτηση.

Το όνομα είναι ένα αναγνωριστικό το οποίο δίνεται κάθε φορά από τον χρήστη (στην προκειμένη περίπτωση όλες οι διεργασίες έχουν αρχικοποιημένα ονόματα με την σειρά κατασκευής τους). Μας παρέχει την δυνατότητα εκτέλεσης της διεργασίας σε ανύποπτο χρόνο ή ακόμα το αποκλεισμό της.

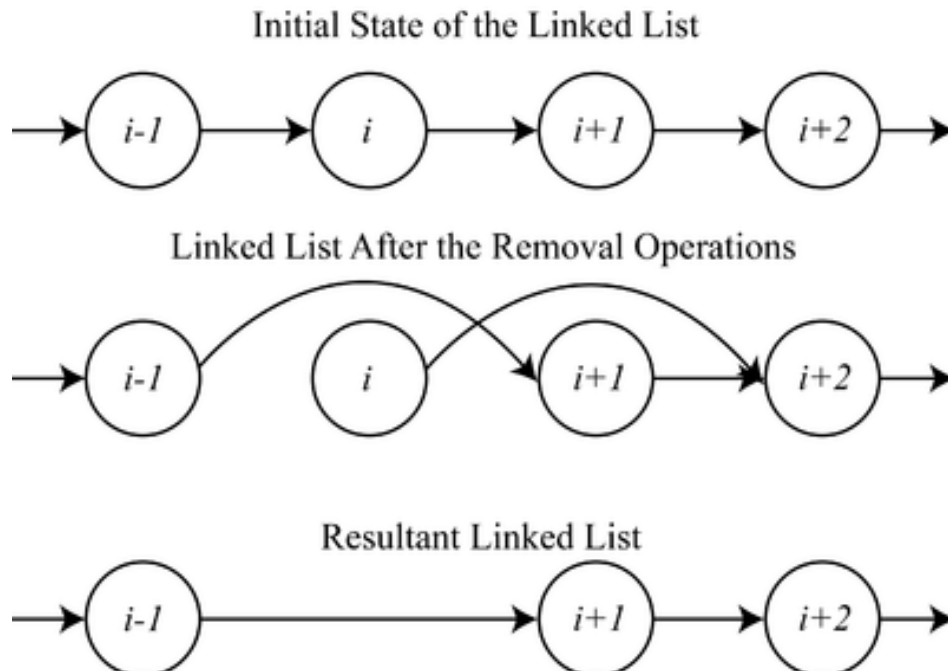
Η συνάρτηση αρχικοποίησης καλείται από το λειτουργικό σύστημα κάθε φορά που ξεκινάει ώστε να δημιουργηθεί ο απαραίτητος διαμοιρασμός πόρων του συστήματος και να υπάρξει δήλωση της εκάστοτε διεργασίας ώστε να μπορεί να κληθεί από άλλες.

Η κεντρική συνάρτηση είναι ο βασικός κώδικας που αποτελείται η κάθε διεργασία. Ένα πολύ βασικό χαρακτηριστικό είναι η μέθοδος γραφής του κώδικα όπου απαιτείτε να μην δημιουργεί ατέρμονους βρόγχους. Σε περίπτωση που συμβεί κάτι τέτοιο το λειτουργικό σύστημα διακόπτει άμεσα την λειτουργία της και την αφαιρεί από την λίστα των εφαρμογών.

### 5.1.1.2 Αμοιβαίος αποκλεισμός (Mutual exclusion)

Με τον όρο αμοιβαίο αποκλεισμό αναφερόμαστε στην διασφάλιση παράλληλων διεργασιών να αποκτήσουν αποκλειστική πρόσβαση όταν φτάσουν σε μια κρίσιμη περιοχή (όπου υπάρχει διαμοιρασμός πόρων όπως για παράδειγμα μια κοινή θέση μνήμης). Η ιδέα είναι να υπάρξει ένας μηχανισμός ώστε όταν δύο παράλληλες διεργασίες θέλουν να χρησιμοποιήσουν ένα κοινό πόρο να μην υπάρχει σύγκρουση επεξεργασίας αλλά να διασφαλίζεται ένας αμοιβαίος αποκλεισμός.

Ένα απλό παράδειγμα γιατί ο αμοιβαίος αποκλεισμός είναι σημαντικός στην πράξη μπορεί να φανεί στο παράδειγμα μιας απλής συνδεδεμένης λίστας. Σε αυτή την λίστα η αφαίρεση ενός κόμβου γίνεται αλλάζοντας τον δείκτη που δείχνει στον επόμενο κόμβο. Για παράδειγμα εάν ο κόμβος  $i$  θέλουμε να αφαιρεθεί τότε το επόμενο δείκτης του  $i-1$  θα αλλαχθεί ώστε να δείχνει τον κόμβο  $i+1$ . Στην περίπτωση που η συνδεδεμένη λίστα είναι προσβάσιμη από δύο παράλληλες διεργασίες και στην περίπτωση που και οι δύο διεργασίες προσπαθήσουν να αφαιρέσουν δύο διαφορετικούς κόμβους παράλληλα ενδέχεται να αντιμετωπίσουμε το παρακάτω πρόβλημα: έστω οι κόμβοι  $i$  και  $i+1$  θέλουμε να αφαιρεθούν. Μετά την αφαίρεση ο κόμβος  $i-1$  θα δείχνει τον  $i+1$  ενώ ο κόμβος  $i$  θα δείχνει στον  $i+2$ . Παρόλο που οι δύο διεργασίες θα ολοκληρώσουν παράλληλα και με επιτυχία τις αφαιρέσεις των δύο κόμβων στην πράξη ο κόμβος  $i+1$  δεν αφαιρείται. Για να λυθεί σωστά το πρόβλημα θα πρέπει να απαιτείται ο αμοιβαίος αποκλεισμών των διεργασιών όταν οι διεργασίες φτάνουν στην κρίσιμη περιοχή αφαίρεσης του κόμβου. Όταν η μια διεργασία αμοιβαία αποκλείει την άλλη κατά την κρίσιμη περιοχή η αφαίρεση και των δύο κόμβων συνδεδεμένη λίστα μπορεί να γίνει με επιτυχία.



Εικόνα 52 : Παράδειγμα συνδεδεμένης λίστας



Για την λύση ενός τέτοιου προβλήματος χρησιμοποιήθηκε ο αλγόριθμος Raymond. Ο αλγόριθμος αυτός στηρίζεται στην δημιουργία ενός επικαλυπτικού δέντρου πριν γίνει οτιδήποτε άλλο, γι' αυτό και κληρονομεί τα πλεονεκτήματα που έχει η κατασκευή αυτή.

| <b>Αλγόριθμος Raymond</b>   |
|---|
| Υπάρχει μία φάση αρχικοποίησης κατά την οποία όλες οι διεργασίες κατασκευάζουν ένα επικαλυπτικό δέντρο. Η ρίζα μπορεί να επιλεγεί τυχαία στην αρχή. Το δέντρο που θα δημιουργηθεί θα έχει κατευθυνόμενες ακμές με κατεύθυνση από τους γονείς προς τα παιδιά.  |
| Η ρίζα αναλαμβάνει να δημιουργήσει ένα μήνυμα – κουπόνι.  |
| Όταν κάποια διεργασία θέλει να κάνει αίτηση για χρησιμοποίηση του πόρου πρέπει να το δηλώσει στην ρίζα. Αυτή με την σειρά της στέλνει στην διεργασία το κουπόνι. Το κουπόνι για να πάει στην διεργασία που το ζήτησε διατρέχει κάποιες ακμές των οποίων αλλάζει την κατεύθυνση. Με αυτό τον τρόπο μετά από την παράδοση του κουπονιού η διεργασία που έκανε την αίτηση θα είναι πλέον η ρίζα του δέντρου. |
| Κάθε διεργασία διατηρεί ένα δείκτη προς το μονοπάτι που οδηγεί στην ρίζα πρακτικά δηλαδή προς τον γονέα της την δεδομένη στιγμή που τρέχει ο αλγόριθμος, ενώ κρατάει και μία ουρά για το ποιες διεργασίες (συμπεριλαμβανομένου και του εαυτού της) έχουν κάνει αίτηση για να πάρουν το κουπόνι.   |

*Αλγόριθμος 4 : Λογικά βήματα αλγορίθμου Raymond*

Όπως είπαμε και στην αρχή λόγω της ύπαρξης του επικαλυπτικού δέντρου τα μηνύματα που ανταλλάσσονται μέχρι να εξυπηρετηθεί μία αίτηση είναι  $O(\log n)$ .

Παρακάτω υπάρχει ένα παράδειγμα κώδικα με τις βασικές μεθόδους που χρησιμοποιεί ο αμοιβαίος αποκλεισμός.

```

osStatus Mutex::lock(uint32_t millisec) {
    return osMutexWait(_osMutexId, millisec);
}

bool Mutex::trylock() {
    return (osMutexWait(_osMutexId, 0) == osOK);
}

osStatus Mutex::unlock() {
    return osMutexRelease(_osMutexId);
}

Mutex::~Mutex() {
    osMutexDelete(_osMutexId);
}

```

## 5.1.2 Οδηγοί Υλικού

Όλες οι συσκευές έχουν έναν μοναδικό τρόπο διασύνδεσης ο οποίος συνεπάγεται με μοναδικό τρόπο υλοποίησης των «οδηγών» τους. Παρόλα αυτά η φιλοσοφία που ακολουθείται χωρίζεται σε 2 μεγάλες κατηγορίες : Η πρώτη κατηγορία αναφέρεται στον παθητικό τρόπο απόκτησης δεδομένων και η δεύτερη στον ενεργητικό τρόπο.

### Παθητικός οδηγός συσκευών

Οι συσκευές που χρησιμοποιούν αυτό τον τρόπο λειτουργίας είναι οι εξής : Επιταχυνσιόμετρο , Μαγνητόμετρο , Αισθητήρας Αέρα , Αισθητήρας μέτρησης αποστάσεων.

### Ενεργητικός οδηγός συσκευών

Σε αυτό τον τρόπο λειτουργίας του οδηγού συσκευών ανήκουν οι : GPS, XBee , Servocontroller

### Παράδειγμα κώδικα Οδηγού Υλικών [ XBEE ]

```
#ifndef _HARDWARE_XBEE_H
#define _HARDWARE_XBEE_H
namespace Hardware
{
namespace XBee
{
struct SMessage
{
char*msg;
struct SMessage *Next;
};
void Initialize_(void);
void DeleteInMessage(void);
void Receive(void);
int CheckCommunication(void);
void SendMessage(char*);
void SendMessageInt(int);
void SendMessageFloat(float);
void AddToInMessages(char*);
char* GetFromInMessages();
void PrintAllInMessages();
void Send(uint8_t);
void CheckPendingMessage(void);
}
}
#endif
```

## 5.1.3 Εφαρμογές

Εκτός από το λειτουργικό σύστημα ,ένα σημαντικό μέρος της λειτουργίας του τετράποδου ρομπότ είναι υλοποίηση μερικών εφαρμογών για την δημιουργία διαδραστικότητας με το περιβάλλον. Συγκεκριμένα υπάρχουν 5 εφαρμογές για την διαχείριση διαφόρων μηχανισμών και αντιδράσεων του ρομπότ.

### Πρότυπο Εφαρμογής

Για να κατανοήσουμε τον τρόπο λειτουργίας των εφαρμογών θα παρουσιάσουμε ένα πρότυπο κώδικα για μία απλή εφαρμογή.

```
//-----;
//  FileName      : _SOFTWARE_BLINK_H
//-----;
#ifndef : _SOFTWARE_BLINK_H
#define : _SOFTWARE_BLINK_H
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

namespace Software
{
namespace Blink
{
void Initialize_(void)

uint8_t getName(void)

void Main_(uint8_t)

char* Version(void)
}
}

#endif
```

## Αναλυτικά οι Εφαρμογές

Η πρώτη εφαρμογή που υλοποιήθηκε ήταν η συλλογή δεδομένων από τους αισθητήρες. Όπως έχουν παρουσιαστεί τα κριτήρια και οι αναλύσεις στα προηγούμενα κεφάλαια η χρήση των αισθητήρων είναι ένα μέρος της εργασίας με μεγάλη βαρύτητα. Για αυτό το λόγο υπάρχει αυτή η εφαρμογή η οποία είναι υπεύθυνη για την επικοινωνία με κάθε αισθητήρα ανάλογα με τις απαιτήσεις του συστήματος. Υπάρχουν δύο τρόποι συλλογής δεδομένων :

- Ανά καθορισμένο χρόνο όπου η επικοινωνία και η αποθήκευση των δεδομένων γίνεται ανεξάρτητα από τις απαιτήσεις των υπολοίπων εφαρμογών
- Με αίτηση : η επικοινωνία και η αποθήκευση των δεδομένων γίνεται μόνο με αίτηση από κάποια άλλη εφαρμογή. Αυτοί οι δύο τρόποι μας δίνουν την δυνατότητα επεξεργασίας των δεδομένων με κάποια φίλτρα πριν μεταβούν στους εκάστοτε αλγόριθμους απόφασης.

Για την αποθήκευση μερικών ρυθμίσεων αλλά και γενικά για την χρήση της SD Card συσκευής ως μέσω αποθήκευσης δημιουργήθηκε μία εφαρμογή διαχείρισης μόνο αυτής της συσκευής. Ο λόγος ήταν κυρίως για την ομαλή λειτουργία του ρομπότ, χωρίς να δημιουργούνται μεγάλα φορτία επικοινωνίας είτε να υπάρξει buffer overflow στις σειριακές θύρες. Η επικοινωνία είναι ιδιαίτερα απλή ,δημιουργώντας ένα event με την εντολή που θέλουμε και το μέρος μνήμης που θέλουμε να αποθηκευτούν τα αποτελέσματα η λειτουργία γίνεται ανεξάρτητα.

Μια επίσης σημαντική εφαρμογή είναι η διαχείριση των κινήσεων του ρομπότ. Όλες οι κινήσεις είναι αποθηκευμένες στην SD Card ως αλληλουχίες με αποτέλεσμα κάθε φορά που χρειάζεται να εκτελεστεί μία κίνηση αρκεί να φορτωθεί το εκάστοτε αρχείο και να αρχίσει η εκτέλεση του. Ένα βασικό χαρακτηριστικό είναι η εύκολη επεξεργασία τους από τον χρήστη.

Η επόμενη αναφέρεται στην επικοινωνία με τον υπολογιστή. Κάθε εισερχόμενο μήνυμα από την συσκευή επικοινωνίας (XBEE Pro 900) δημιουργεί ένα event προς την συγκεκριμένη εφαρμογή και αυτή είναι υπεύθυνη να καθορίσει τον τρόπο επεξεργασίας του μηνύματος και την εκτέλεση των εντολών.

Η τελευταία εφαρμογή είναι μια απλή μορφή ενός interpreter, παρόμοιος με τον τρόπο λειτουργίας του bash interpreter των Linux. Κάθε εισερχόμενο – εξερχόμενο μήνυμα όπως επίσης και κάθε ανταλλαγή εσωτερικών μηνυμάτων επιβάλλεται είτε να διέρχεται μέσω αυτής της εφαρμογής είτε να ακολουθεί τα πρότυπα εντολών της.

## 5.2 Λογισμικό Επικοινωνίας

Ένα εξίσου σημαντικό μέρος της πτυχιακής εργασίας είναι η υλοποίηση μίας εφαρμογής επικοινωνίας της ρομποτικής πλατφόρμας με τον υπολογιστή. Με αυτό τον τρόπο ο χειριστής έχει πρόσβαση στα δεδομένα που συλλέγει το ρομπότ όπως επίσης και την δυνατότητα να στέλνει εντολές. Η επικοινωνία γίνεται ασύρματα με την χρήση ενός ζεύγους XBEE Pro 900 (πληροφορίες για τα XBEE μπορείτε να βρείτε στο κεφάλαιο 3.6.8 *Συσκευή Επικοινωνίας*) τα οποία μας προσφέρουν γρήγορη μετάδοση δεδομένων και αρκετά μεγάλη απόσταση.

Η εφαρμογή **i026 Control Panel** δημιουργήθηκε με την γλώσσα προγραμματισμού C#.

### Τι είναι η C#

Η C# είναι μια σχετικά νέα αντικειμενοστραφής γλώσσα προγραμματισμού η οποία δημιουργήθηκε από την **Microsoft**. Δανείζεται πολλά στοιχεία, και έχει παρόμοια σύνταξη, με την C++ και την Java, κάνοντας την εκμάθηση της σχετικά εύκολη. Είναι γλώσσα ειδικά σχεδιασμένη για να υποστηρίζει το **.NET framework** της ίδιας εταιρείας. Βασικό χαρακτηριστικό της είναι ότι δεν παράγει απευθείας κώδικα μηχανής όπως η C++, αλλά ένα ενδιάμεσο κώδικα που στοχεύει το .NET.

### Απαιτήσεις Εφαρμογής

Τα κύρια χαρακτηριστικά που αποτέλεσαν καθοριστικό παράγοντα στον τρόπο υλοποίησης της εφαρμογής είναι η αξιοπιστία μετάδοσης μηνυμάτων – εντολών και η ευχρηστία.

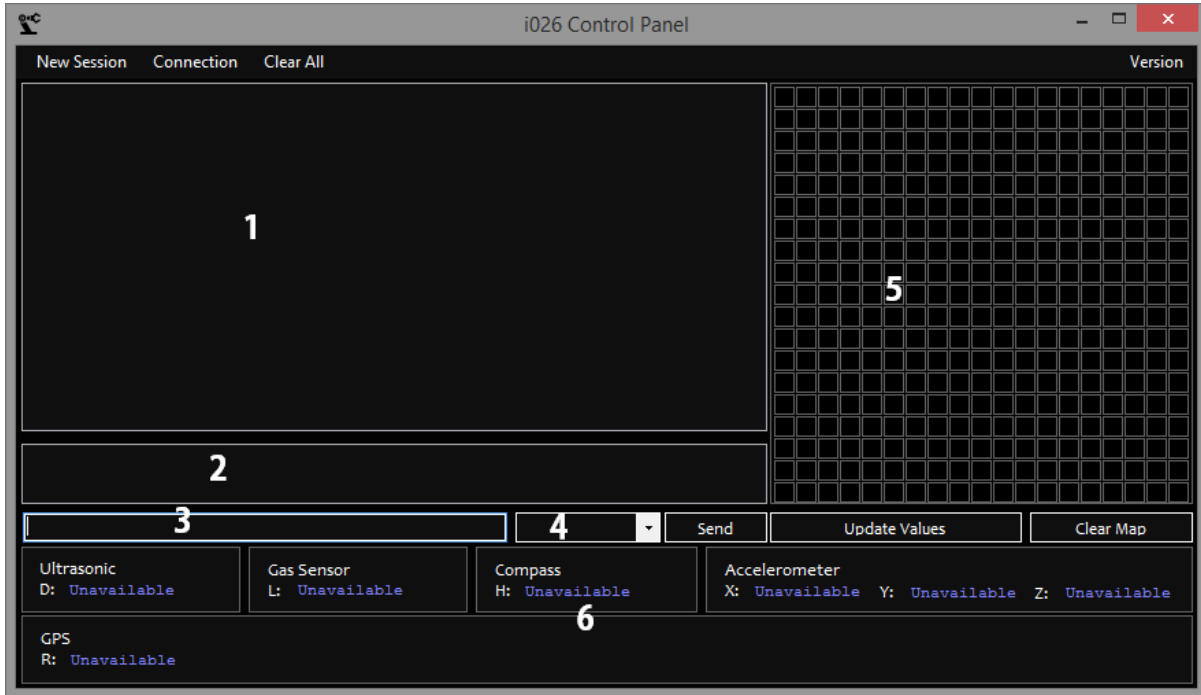
### 5.2.1 Περισσότερες Πληροφορίες

Ο τρόπος λειτουργίας της εφαρμογής είναι πολύ απλός. Η συσκευή XBee Pro 900 δημιουργεί μια virtual serial port από την οποία μπορούμε να επικοινωνήσουμε μαζί της. Η εφαρμογή με την σειρά της συνδέεται στην Serial Port και ταυτόχρονα επιτυγχάνεται η ανταλλαγή μηνυμάτων με την ρομποτική πλατφόρμα.

Αν και υπάρχουν πολλές ρυθμίσεις που μπορούν να αλλάξουν τον τρόπο επικοινωνίας με την σειριακή θύρα, οι συσκευές έχουν καθορισμένη ρύθμιση με αποτέλεσμα η εφαρμογή να μην χρειάζεται κάτι περισσότερο από την επιλογή της θύρας και τον ρυθμό μετάδοσης.

## 5.2.2 Επεξήγηση Διεπαφής της εφαρμογής

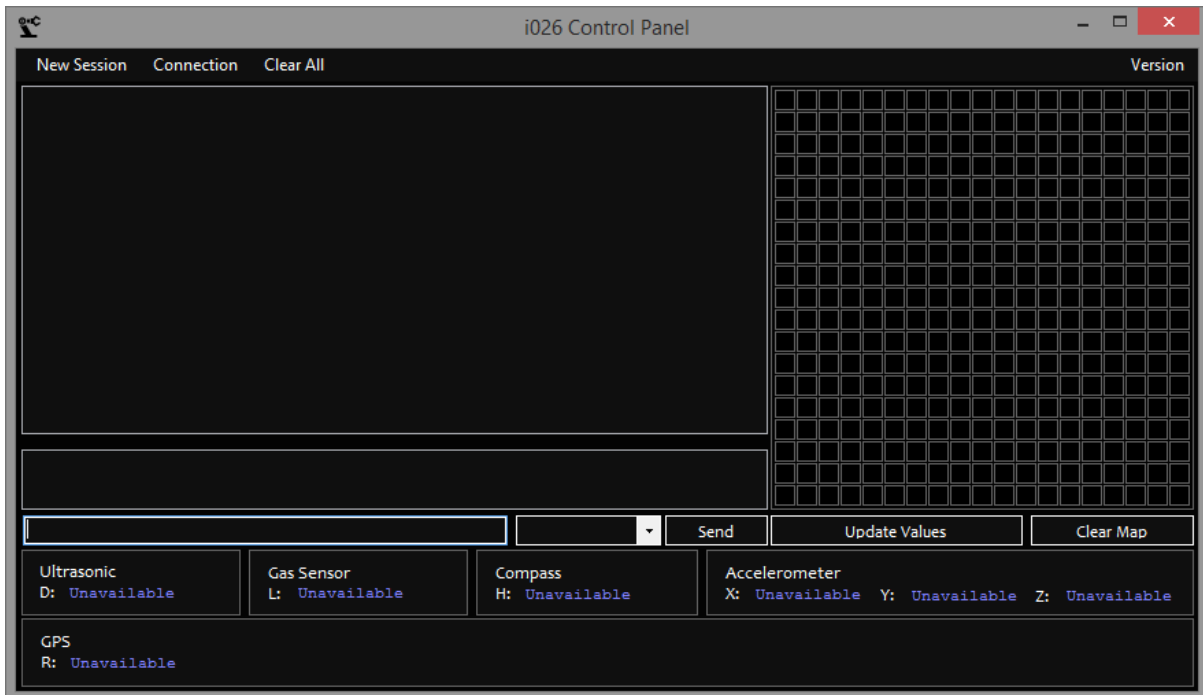
Παρακάτω παρατηρούμε την διεπαφή της εφαρμογής και αναλυτικά τα μέρη που αποτελείται.



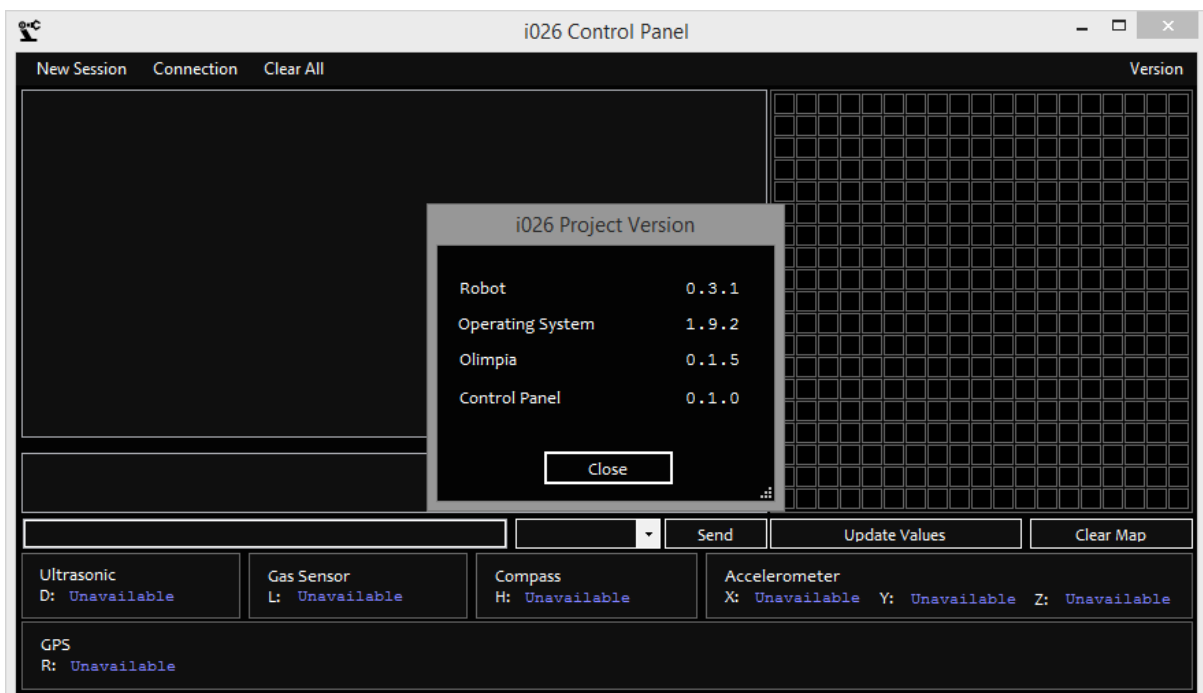
Εικόνα 53 : i026 Control Panel

| Πίνακας Επεξήγησης Εφαρμογής |                               |
|------------------------------|-------------------------------|
| Αριθμός                      | Επεξήγηση                     |
| 1                            | Εισερχόμενα μηνύματα          |
| 2                            | Εξερχόμενα μηνύματα           |
| 3                            | Δημιουργία μηνύματος          |
| 4                            | Έτοιμες εντολές               |
| 5                            | Χάρτης παρουσίασης αποστάσεων |
| 6                            | Δεδομένα αισθητήρων           |

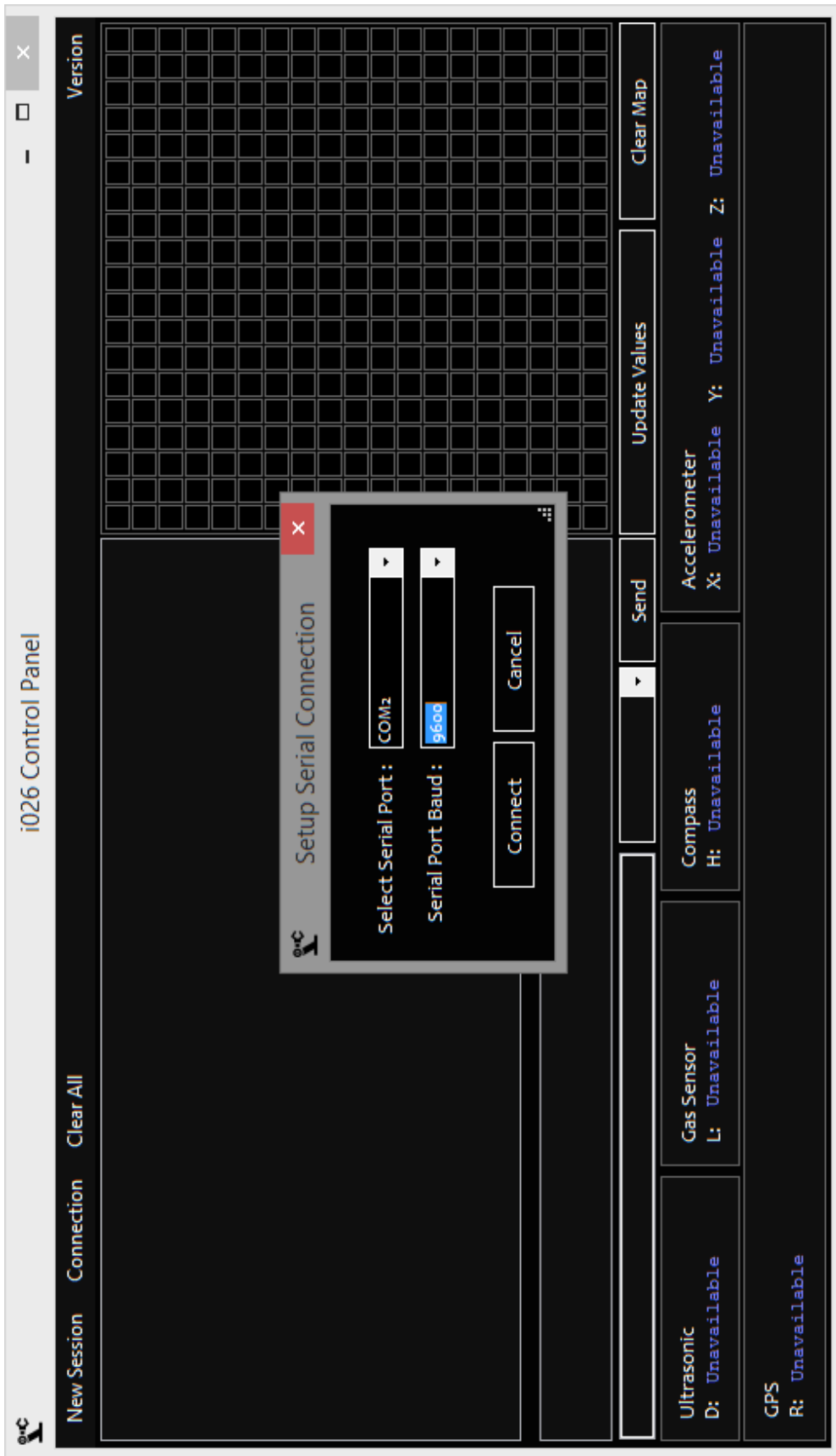
## 5.2.3 Χρήση της Εφαρμογής i026 Control Panel



Εικόνα 54 : i026 Control Panel Δείγμα Χρήσης (1)

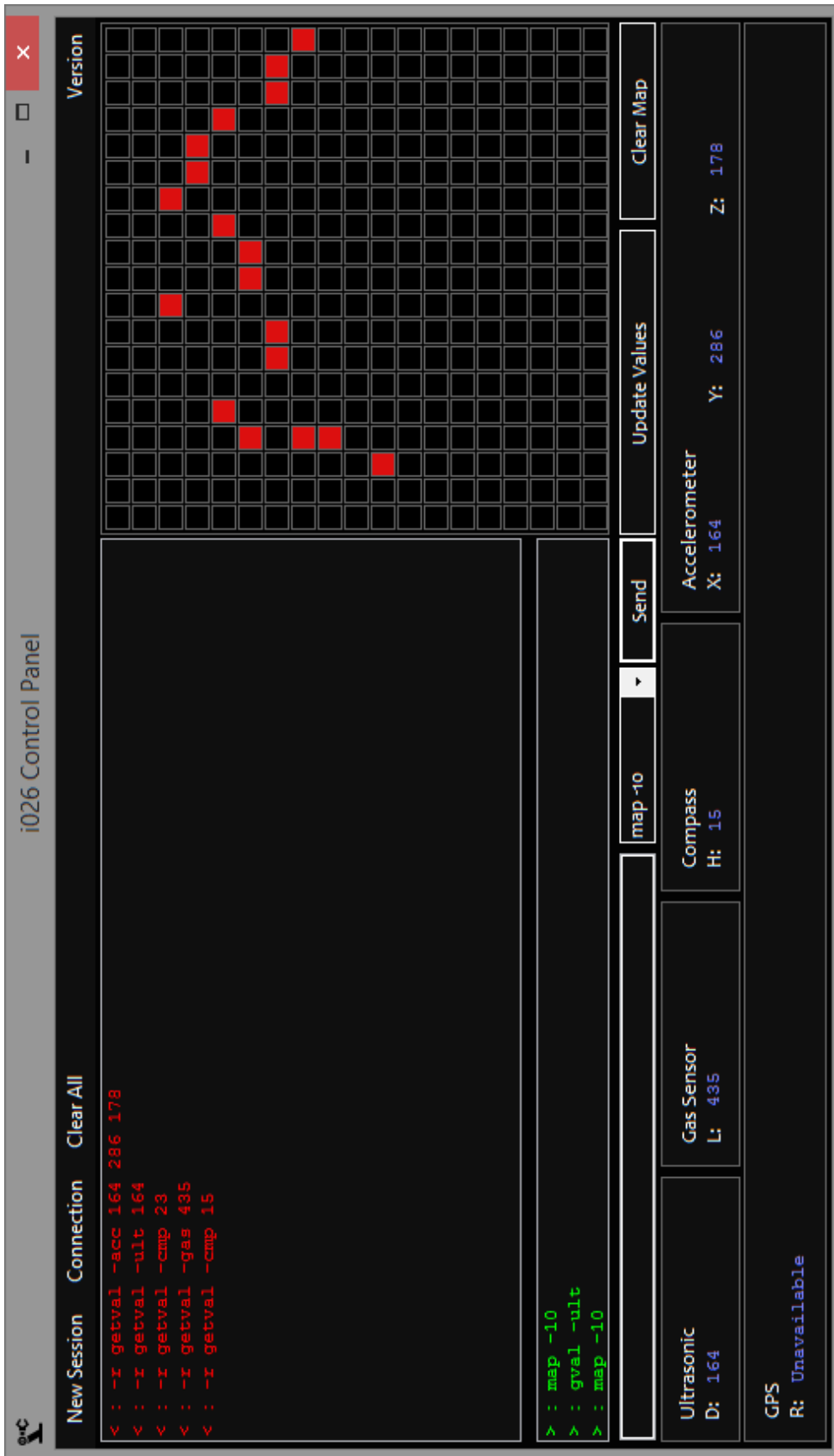


Εικόνα 55 : i026 Control Panel Δείγμα Χρήσης (2)



Εικόνα 56: i026 Control Panel Δείγμα Χρήσης (3)



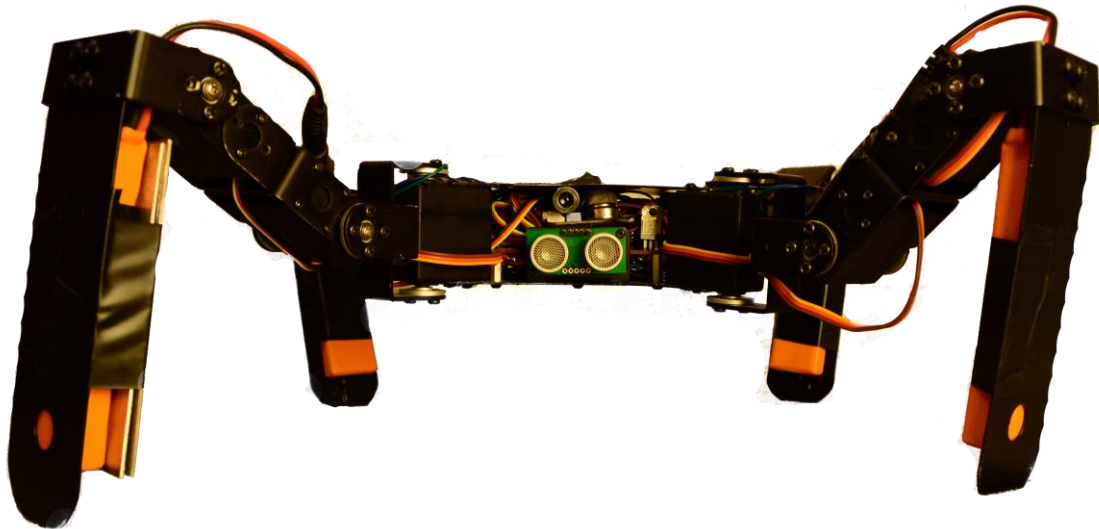


Εικόνα 57 : i026 Control Panel Δείγμα Χρήσης (4)

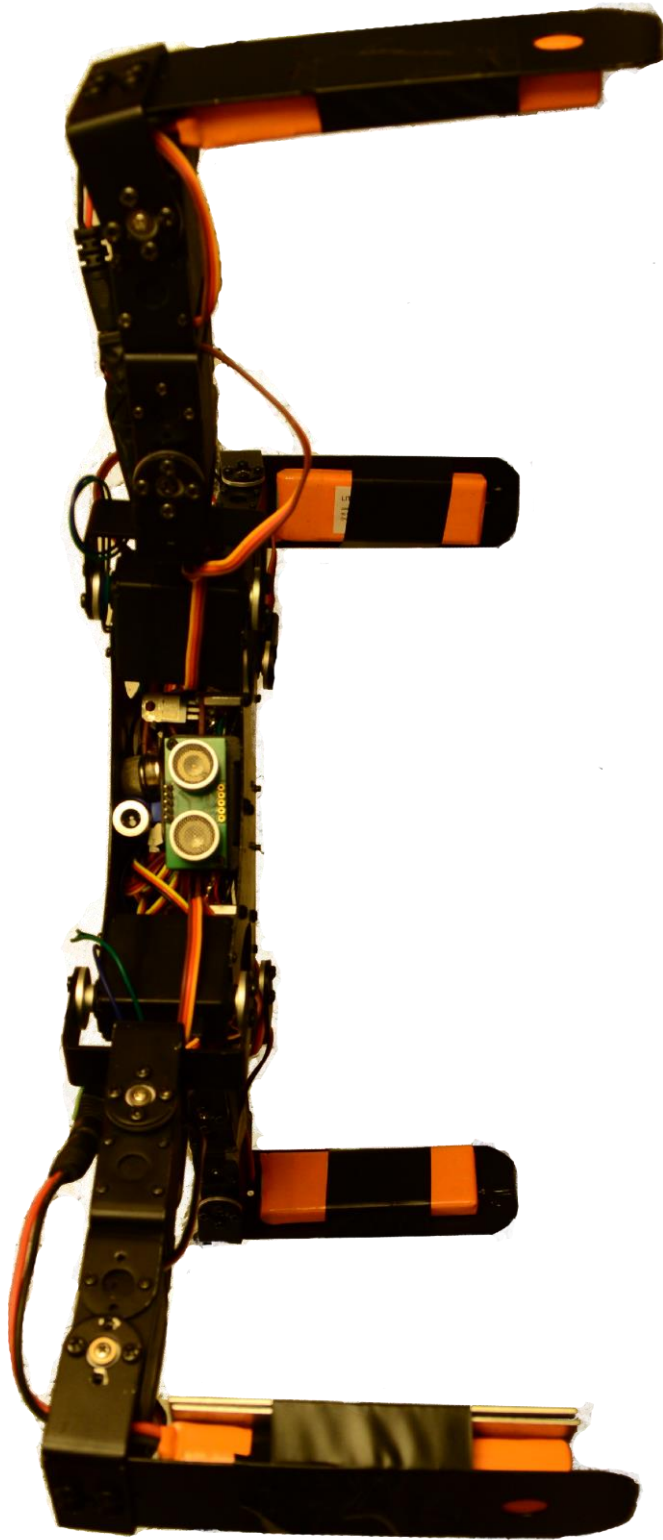
## 6. Αποτελέσματα

Στο παρόν κεφάλαιο θα αναλύσουμε τα αποτελέσματα της πτυχιακής εργασίας και θα παρουσιαστούν διάφορες μελλοντικές αναβαθμίσεις που θα μπορούσαν να γίνουν ώστε η συγκεκριμένη ρομποτική πλατφόρμα να εφαρμοστεί σε διάφορες πραγματικές καταστάσεις.

### 6.1 Παρουσίαση Δείγματος Λειτουργίας



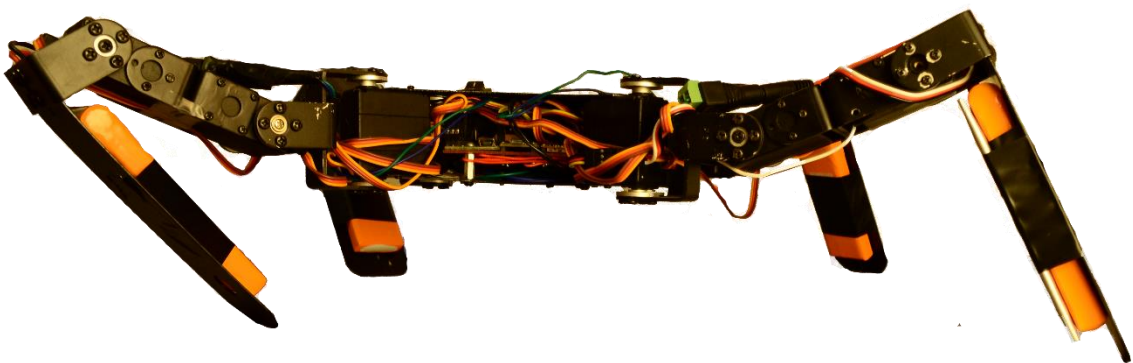
Εικόνα 58 : Φωτογραφία Ρομποτ (1)



Εικόνα 59 : Φωτογραφία Ρομπोट (2)



Εικόνα 60 : Φωτογραφία Ρομπότ (3)



Εικόνα 61 : Φωτογραφία Ρομπότ (4)



Εικόνα 632 : Φωτογραφία Ρομπότ (5)

## 6.2 Συμπεράσματα

Το σύνολο της πτυχιακής εργασίας και ο σκοπός της υλοποιήθηκαν επιτυχώς, από τη μηχανική μελέτη μέχρι την δημιουργία της ρομποτικής πλατφόρμας όπως επίσης και την κατασκευή της εφαρμογής επικοινωνίας του υπολογιστή.

Σε αυτό το στάδιο το ρομπότ έχει τις εξής δυνατότητες:

Μέσω του λειτουργικού του συστήματος, όλες οι συνδεδεμένες συσκευές (αισθητήρες κ.λπ.) ελέγχονται από τον επεξεργαστή του συστήματος, όπως επίσης μας παρέχεται η δυνατότητα ασύγχρονης συλλογής των δεδομένων τους.

Επιπλέον, παρέχει τη δυνατότητα να διαμοιράσει τους πόρους του και το χρόνο επεξεργασίας στις διάφορες εφαρμογές και να κινηθεί στο χώρο με βάση τους αλγόριθμους κίνησης.

Τέλος, η επικοινωνία με τη ρομποτική πλατφόρμα πραγματοποιείται ασύρματα με τον υπολογιστή χρησιμοποιώντας την συσκευή xbee pro 900, που δημιουργήθηκε ειδικά για αυτό το σκοπό, για την εκτέλεση και την συλλογή δεδομένων, δημιουργήθηκε η διεπαφή του control panel..

Κατά τη διάρκεια της υλοποίησης κάθε μέρος χρειάστηκε ειδική αντιμετώπιση καθώς οι απαιτήσεις συνεχώς άλλαζαν, με αποτέλεσμα τον συνεχή επαναπροσδιορισμό των στόχων τους.

## 6.3 Μελλοντικές Εργασίες και Επεκτάσεις

Υπάρχουν πολλές μελλοντικές εξελίξεις της συγκεκριμένης πλατφόρμας που μπορούν να υλοποιηθούν στους εκάστοτε τομείς. Για τα μηχανικά μέρη του ρομπότ θα μπορούσε να μελετηθεί περαιτέρω ο σχηματισμός του, να χρησιμοποιηθεί κάποιο καλύτερο υλικό όπως το ανθρακόνημα με αποτέλεσμα να μειωθεί το βάρος και να αυξηθεί η ανθεκτικότητα. Επιπλέον, οι servo κινητήρες θα μπορούσαν να είναι μεγαλύτερης ισχύος. Στα ηλεκτρονικά μέρη θα μπορούσαμε να έχουμε ένα μικρότερο κύκλωμα με μεγαλύτερη επεξεργαστική ισχύ για τη διεκπεραίωση πολυπλοκότερων αλγόριθμων και σαφώς το ενεργειακό πρόβλημα το οποίο βρίσκεται σε συνάρτηση με τα μηχανικά μέρη της πλατφόρμας.

Στο λογισμικό μέρος μια μελλοντική επέκταση θα μπορούσε να είναι η ανάπτυξη ενός ευφυούς συστήματος με τη χρήση νευρωνικών δικτύων και γενετικών αλγορίθμων για τη λήψη αποφάσεων αλλά και τη καλύτερη εκμάθηση κίνησης.

Εφαρμόζοντας τις μελλοντικές επεκτάσεις εξέλιξης που αναφέρθηκαν παραπάνω μια τέτοια ρομποτική πλατφόρμα θα μπορούσε να χρησιμοποιηθεί σε περιοχές δυσπρόσιτες για τον άνθρωπο με σκοπό την εξερεύνηση ή την επίλυση προβλημάτων.

## 7. Βιβλιογραφία

- DFRobot. (2009). *Product User Manual : Gas Sensor*.
- GlobalSat. (2010). *Product User Manual :GPS Receiver Engine Board EM-406A* .
- Julier, J. S., & Uhlmann, J. K. (2000). Building a million beacon map. *Proceedings of the SPIE Sensor Fusion and Decentralized Control in Robotic Systems IV*, 4571.
- Kalman, E. R. (1960). *A new approach to linear filtering and prediction problems*.
- Khatib, O. (n.d.). Real-time obstacle avoidance for robot manipulator and mobile robots. *The International Journal of Robotics Research*, 5, σσ. 90-98.
- Kleinberg, J. (1994). The localization problem for mobile robots. *35th IEEE Symposium on Foundations of Computer Science*.
- Kwot, T. C., Fox, D., & Meila, M. (2004). Real-time particle filters. *Proceedings of the IEEE. Special Issue on Sequential State Estimation*, 92, σσ. 469–484.
- Labrosse, J. (2009). *MicroC/OS-III: The Real Time Kernel*.
- Michael, B. (2009). *Embedded C Coding Standard*.
- Michael, B., & Massa, A. (2006). *Programming Embedded Systems: With C and GNU Development Tools, 2nd Edition*.
- Prasad, A. (2003). *Embedded Real Time Systems : Concepts, Design Prog Bb*.
- PRJC. (2013). *Product User Manual : Teensy 3.0*.
- Thrun, S., Wolfram, B., & Fox, D. (2006). *Πιθανοτική Ρομποτική*.
- Valvano, J. (2012). *Embedded Systems: Real-Time Operating Systems for Arm Cortex M Microcontrollers*.
- Κουμπουλής, Β., & Μέρτζιος, Φ. (2002). *Εισαγωγή στην Ρομποτική*.