

Τ.Ε.Ι ΠΕΙΡΑΙΑ
Τμήμα ΑΥΤΟΜΑΤΙΣΜΟΥ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



ΣΠΟΥΔΑΣΤΗΣ:
ΝΕΛΛΑΣ
ΒΑΣΙΛΕΙΟΣ

ΥΛΟΠΟΙΗΣΗ ΑΣΑΦΟΥΣ ΕΛΕΓΚΤΗ ΣΕ µΡ BASIC
STAMP 2 ΓΙΑ ΚΙΝΗΣΗ ΑΥΤΟΝΟΜΟΥ
ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ



ΕΙΣΗΓΗΤΗΣ:
κ. ΝΤΟΥΝΗΣ ΑΝΑΣΤΑΣΙΟΣ

ΔΙΔΑΚΤΙΚΟ ΕΤΟΣ 2012-2013

Τ.Ε.Ι ΠΕΙΡΑΙΑ

Τμήμα ΑΥΤΟΜΑΤΙΣΜΟΥ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΛΟΠΟΙΗΣΗ ΑΣΑΦΟΥΣ ΕΛΕΓΚΤΗ ΣΕ μP BASIC STAMP 2 ΓΙΑ ΚΙΝΗΣΗ
ΑΥΤΟΝΟΜΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

ΣΠΟΥΔΑΣΤΗΣ:

ΝΕΛΛΑΣ ΒΑΣΙΛΕΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

κ. ΝΤΟΥΝΗΣ ΑΝΑΣΤΑΣΙΟΣ

ΔΙΔΑΚΤΙΚΟ ΕΤΟΣ 2012-2013

ΠΕΡΙΛΗΨΗ

Υλοποίηση ασαφούς ελεγκτή σε μP BASIC STAMP 2 για την κίνηση αυτόνομου ρομποτικού συστήματος.

Κατασκευή πλατφόρμας με τέσσερα πόδια και προγραμματισμός μικροελεγκτή για να περπατάει προς όλες τις κατευθύνσεις. Τα πόδια αυτά είναι κατασκευασμένα με τρόπο ώστε με την χρήση σερβομηχανισμών να κινούνται σε κάθετο και οριζόντιο άξονα. Για την κίνηση και τον έλεγχο τους χρησιμοποιείται ένας μικροεπεξεργαστής BASIC Stamp 2 και ένας ελεγκτής σερβοκινητήρων. Μετά το τέλος της κατασκευής και του προγραμματισμού της κίνησης η πλατφόρμα θα είναι έτοιμη τόσο σε μηχανική κατάσταση όσο σε βασικό προγραμματισμό ώστε να μπορούν να τοποθετηθούν παντός τύπου αισθητήρια πάνω της και μέσω επεξεργασίας των σημάτων τους να εκτελούνται οι επιθυμητές κινήσεις ή εργασίες.

Ο μικροεπεξεργαστής BASIC Stamp 2 είναι κατασκευασμένος ώστε να μπορεί να ελέγχει και να παρακολουθεί συσκευές όπως χρονοδιακόπτες, πληκτρολόγια, αισθητήρες, κινητήρες, διακόπτες, ρελέ, φώτα και πολλά άλλα. Ο προγραμματισμός του γίνεται σε γλώσσα PBASIC. Για την ανάπτυξη του ασαφούς ελεγκτή θα χρησιμοποιηθεί το Fuzzy Toolbox από την εργαλειοθήκη του MATLAB.

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ευκαιρία της κατάθεσης της πτυχιακής μου εργασίας θα ήθελα να ευχαριστήσω τα παρακάτω άτομα:

Όλους όσους βοήθησαν, μέλη στο Forum <http://forums.parallax.com> που ασχολείται με τον προγραμματισμό του μP BASIC STAMP.

Τον υπεύθυνο επιβλέπων καθηγητή κ. Αναστάσιο Ντούνη για την καθοδήγηση του σε αυτή την εργασία, αλλά και για την ανάπτυξη του ενδιαφέροντος μου για ενασχόληση με το MATLAB και τα FUZZY LOGIC συστήματα.

Τους φίλους μου για τις ώρες που αφιέρωσαν μαζί μου κατά την κατασκευή και την συγγραφή της εργασίας.

Τον Παναγιώτη Κοφινά βοηθό του εργαστηρίου Ευφυούς Ελέγχου για την βοήθεια στον προγραμματισμό του Fuzzy Logic ελεγκτή και την διόρθωση του κειμένου.

Ο σπουδαστής

Νέλλας Βασίλειος

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ:

ΚΕΦΑΛΑΙΟ 1ο

1.1. Σκοπός Δημιουργίας της πτυχιακής εργασίας.....	7
---	---

ΚΕΦΑΛΑΙΟ 2ο

2.1. Μηχανικό Μέρος.....	8
2.1.1. Υλικά μηχανικού μέρους.....	8
2.1.2. Επιλογή υλικών.....	9
2.2. Ηλεκτρονικό Μέρος.....	9
2.2.1. Υλικά Ηλεκτρονικού Μέρους.....	9
2.2.2. Μικροελεγκτής Basic Stamp 2.....	10
2.2.2.1. Εισαγωγή στον BASIC Stamp 2.....	10
2.2.2.2. Οδηγίες και Προφυλάξεις.....	10
2.2.2.3. Γρήγορη Εισαγωγή στον BASIC Stamp.....	12
2.2.3. Μητρική Πλακέτα B.O.E. (Board Of Education® - USB (#28850)).....	14
2.2.4. Ελεγκτής Σερβομοτέρ της Parallax (#28023) – Rev B (Parallax Servo Controller (#28023)).....	14
2.2.5. Σερβομοτέρ.....	15
2.2.6. Αισθητήρας απόστασης Sharp IR GP2D12 και Αναλογικός σε Ψηφιακό Μετατροπέα (Analog to Digital Converter) ADC0831.....	16
2.2.6.1. Αισθητήρας απόστασης Sharp IR GP2D12.....	16
2.2.6.2. Αναλογικός σε Ψηφιακό Μετατροπέα (Analog to Digital Converter) ADC0831.....	17
2.2.7. PUSH BUTTON και κύκλωμα LED.....	17
2.2.8. Πηγές Τάσης.....	19
2.2.9. Λόγοι επιλογής.....	19
2.2.10. Κόστος Κατασκευής.....	21
2.3. Κατασκευή και Συναρμολόγηση.....	22
2.3.1. Μηχανικό Μέρος.....	22
2.3.2. Συναρμολόγηση Ηλεκτρονικού Μέρους.....	26

ΚΕΦΑΛΑΙΟ 3^ο

3. Προγραμματισμός.....	27
3.1. Γλώσσα προγραμματισμού.....	27
3.2. Περιβάλλον Προγραμματισμού.....	27
3.3. Προεργασία πριν το Ξεκίνημα του Προγραμματισμού	31
3.3.1. Ρύθμιση των ποδιών και ρύθμιση του εύρους κίνησής τους.....	31
3.3.2. Προγραμματισμός περπατήματος.....	33
3.3.2.1. Περπάτημα Ευθεία.....	34
3.3.2.2. Περπάτημα Ευθεία Πίσω.....	34
3.3.2.3. Τρόπος περπατήματος με στροφή.....	35
3.3.2.4. Ταχύτητα κίνησης και συνδυασμός κινήσεων.....	35
3.3.2.5. Ανάλυση Λογικής Προγραμματισμού του κώδικα Περπατήματος...36	
3.3.3. Υπολογισμός της μετρούμενης απόστασης από τον Sharp IR.....	37
3.4. Σύνδεση του BASIC Stamp με το MATLAB.....	40
3.5. Ελεγκτής FUZZY LOGIC.....	41
3.6. Διάγραμμα Ροής του Προγράμματος.....	48

ΚΕΦΑΛΑΙΟ 4^ο

4.1. Συμπεράσματα και προτάσεις περαιτέρω μελέτης ή ανάπτυξης.....	51
--	----

ΚΕΦΑΛΑΙΟ 5^ο

Βιβλιογραφία – Αναφορές.....	53
------------------------------	----

ΠΑΡΑΡΤΗΜΑΤΑ

A. Συντομογραφίες.....	54
B. Συναρμολόγηση Του Ρομποτικού.....	55
Γ. Χαρακτηριστικά Εξαρτημάτων.....	61
Γ.1. Μικροελεγκτής Basic Stamp 2.....	61
Γ.2. Μητρική Πλακέτα B.O.E. (Board Of Education® - USB (#28850))	64
Γ.3. Ελεγκτής Σερβομοτέρ της Parallax (#28023) – Rev B (Parallax Servo Controller (#28023) – Rev B)	66

Γ.4. Σερβομοτέρ.....	67
Γ.5. Αισθητήρας απόστασης Sharp IR GP2D12.....	68
Γ.6. Αναλογικός σε Ψηφιακό Μετατροπέα (Analog to Digital Converter) ADC0831..	69
Δ. Αναλυτικό Διάγραμμα Ροής του Προγράμματος.....	71
Ε. Πρόγραμμα Και Επεξηγήσεις Εντολών.....	82
ΣΤ. Εντολές και σύνταξη Εντολών.....	100

ΚΕΦΑΛΑΙΟ 1ο

1.1. Σκοπός Δημιουργίας της πτυχιακής εργασίας

Η πτυχιακή αυτή εργασία δημιουργήθηκε με σκοπό να κατασκευαστεί ένα ρομποτικό με αυτόνομο χαρακτήρα, πολλές δυνατότητες και μοναδικότητα. Η επιλογή της κατασκευής ενός τροχήλατου οχήματος απορρίφτηκε αμέσως, καθώς αποτελεί άσκηση πολλών άλλων φοιτητών και θα έχανε την μοναδικότητά της. Επίσης οι δυνατότητες τροχήλατων ρομποτικών σταματάνε εκεί που μπορούν να κυλήσουν οι τροχοί του. Η λύση σε αυτό που ζητούσαμε ήταν μία κατασκευή με αρθρωτά πόδια που θα αντιγράφει την κίνηση περπατήματος ζώων. Με τον τρόπο αυτό θα είναι δυνατή μελλοντικά η βελτίωση της κατασκευής για να μπορεί να κινείται και σε άλλες επιφάνειες εκτός του επίπεδου πατώματος. Με αυτή την σκέψη αναγκαζόμαστε να μελετήσουμε και άλλους τρόπους κίνησης εκτός από την κύλιση τροχών και να κατασκευάσουμε μηχανήματα που να μπορούν να φτάσουν σε σημεία με μεγαλύτερο βαθμό δυσκολίας στην προσέγγισή τους.

Η βασική ιδέα ήταν να σχεδιαστεί ένα ρομποτικό με αρθρωτά πόδια που θα αντιγράφουν τα πόδια των ζώων. Η μελέτη και η κατασκευή αυτού του ρομποτικού είχε σαν αποτέλεσμα μια κατασκευή με πολλές δυνατότητες και προοπτικές βελτίωσης. Το βασικό μέρος της άσκησης, αυτή τη στιγμή είναι να μελετηθούν πιθανοί τρόποι περπατήματος και να υλοποιηθούν κατασκευάζοντας ένα κατάλληλο ρομποτικό. Στη συνέχεια θέλουμε με τη χρήση κάποιου ελεγκτή να περπατάει αυτόνομα και να κάνει κάποιες εργασίες που του ορίζουμε.

Το άλλο μέρος της άσκησης αποτελεί την πρόκληση για να δημιουργηθεί επικοινωνία μεταξύ του επεξεργαστή και Η/Υ σε περιβάλλον MATLAB. Η επικοινωνία αυτή θα μπορεί να δώσει στους χρήστες και μελετητές πολύ περισσότερες δυνατότητες από αυτές που θα μπορεί να έχει το ρομποτικό μόνο με τον μικροελεγκτή, αφού θα είναι δυνατός ο έλεγχος από απομακρυσμένο Η/Υ και ο ελεγκτής BASIC Stamp θα είναι υπεύθυνος μόνο για τις βασικές λειτουργίες και την λήψη και αποστολή δεδομένων.

ΚΕΦΑΛΑΙΟ 2ο

2.1.Μηχανικό Μέρος

2.1.1 Υλικά μηχανικού μέρους

Τα υλικά που χρησιμοποιήθηκαν για την κατασκευή χωρίζονται στα υλικά του μηχανικού και του ηλεκτρονικού μέρους.

Κομμάτια μηχανικού μέρους.

Τα μηχανικά μέρη είναι:

- (1)το πάνω πάτωμα
- (1)το κάτω πάτωμα
- (2)μπροστά και πίσω κάθετες στηρίξεις των πατωμάτων
- (2)αριστερές βάσεις σερβομοτέρ
- (2)δεξιές βάσεις σερβομοτέρ
- (4)κάθετα τμήματα ποδιών
- (12)μικρά οριζόντια τμήματα ποδιών
- (4)μεγάλα οριζόντια τμήματα ποδιών
- (4)σύνδεσμοι ποδιών με σερβομοτέρ
- (4)άκρα στα πόδια

- (20)4/.70*22mm βίδες
- (28)4/.70*10mm βίδες
- (24)4/.70 παξιμάδια ασφαλείας
- (32)4/.70 παξιμάδια
- (128)4*0.8mm ροδέλες τεφλόν
- (12)2.5*10mm λαμαρινόβιδες
- (8)2.5/.50*7mm βίδες
- (8)2.5/.50*15mm βίδες
- (4)2.5/.50*10mm βίδες φρεζάτες
- (12)2.5/.50 παξιμάδια
- (12)2.5 ροδέλες ασφαλείς
- (6)μονωτικά κομμάτια περμανίτη

2.1.2 Επιλογή υλικών.

Το υλικό που επιλέχτηκε για την κατασκευή του σώματος του ρομποτικού, των βάσεων των σερβομοτέρ και των ποδιών είναι αλουμίνιο 5052 σε φύλο 2mm. Ο τύπος αυτός είναι ένας από τα πιο σκληρά αλουμίνια τα οποία δεν έχουν υποστεί θερμική επεξεργασία σκλήρυνσης. Επίσης αυτός ο βαθμός παρουσιάζει αρκετά καλή αντίσταση για εφαρμογές στην ναυτιλία και περιβάλλον με άλατα. Έχει πολύ καλά χαρακτηριστικά για επεξεργασία και μπορεί εύκολα να διαμορφωθεί σε πολύπλοκα σχήματα με ελάχιστη δημιουργία ρωγμών που παρουσιάζουν άλλοι τύποι αλουμινίου. Χρησιμοποιείται σε ένα ευρύ φάσμα κατασκευών στη ναυτιλία, σε εξαρτήματα αεροσκαφών, εξαρτήματα βιομηχανίας, σε οικιακές συσκευές, βαριά μαγειρικά σκεύη και εξοπλισμό μαζικής επεξεργασίας τροφίμων. [1]

Επιλέχτηκε γιατί τόσο οι μηχανικές αντοχές του όσο και τα χαρακτηριστικά του μπορούν να παρέχουν την αντοχή και την αξιοπιστία που χρειάζεται η κατασκευή με μικρό βάρος. Τέλος είναι αρκετά οικονομικό σαν υλικό.

Οι βίδες είναι ανοξείδωτες, για να μην σκουριάζουν στην πάροδο του χρόνου και για καλύτερη εμφάνιση. Οι ροδέλες από τεφλόν επιλέχτηκαν για δύο λόγους. Ο πρώτος είναι η χρήση τους σαν ροδέλες στις βίδες και στα παξιμάδια για να υπάρχει ένα παρέμβυσμα μεταξύ του σκληρού υλικού που είναι κατασκευασμένες οι βίδες και του αλουμινίου του σώματος και των ποδιών. Ο δεύτερος είναι η χρήση τους σαν αποστάτες ανάμεσα στα κενά των κομματιών των ποδιών. Το γεγονός ότι χρησιμοποιούνται πολλές ροδέλες με γράσο μεταξύ τους, μειώνει τις τριβές.

2.2. Ηλεκτρονικό Μέρος

2.2.1. Υλικά Ηλεκτρονικού Μέρους

Τα ηλεκτρονικά μέρη είναι:

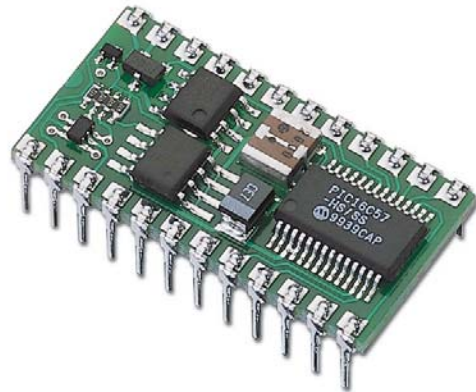
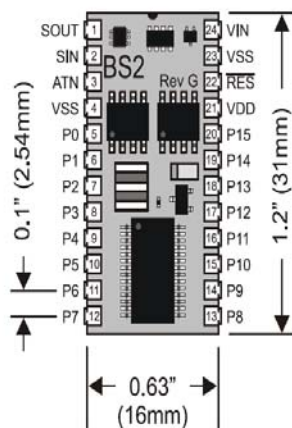
- (1) Ο μικροελεγκτής BASIC STAMP 2
- (1) Η βασική πλακέτα της Parallax, Board of Education USB
- (1) το καλώδιο σύνδεσης με H/Y μέσω USB
- (8) Σερβομοτέρ TowerPro MG955
- (1) Ο ελεγκτής για τους σερβομοτέρ της Parallax, PCS
- (1) ενδεικτική οθόνη 7 στοιχείων LED
- (7) αντιστάσεις 1kΩ
- (2) αντιστάσεις 220Ω
- (2) αντιστάσεις 10kΩ
- (2) pushbuttons

- (1) Ανιχνευτής απόστασης με υπέρυθρες GP2D12
- Καλώδια διαφόρων χρωμάτων
- (1) Μπαταρία 9V
- (1) Τροφοδοτικό 7.5V 2250μΑ

2.2.2. Μικροελεγκτής Basic Stamp 2

2.2.2.1. Εισαγωγή στον BASIC Stamp 2

Ο μικροελεγκτής που χρησιμοποιείται είναι ο BASIC Stamp 2 της εταιρίας PARALLAX. Οι μικροελεγκτές Basic Stamp έχουν σχεδιαστεί για να χρησιμοποιούνται σε ένα ευρύ φάσμα εφαρμογών. Πολλές κατασκευές που απαιτούν ενσωματωμένο σύστημα με κάποιο επίπεδο νοημοσύνης μπορούν να χρησιμοποιήσουν για ελεγκτή κάποιον Basic Stamp. Υπάρχουν έξι διαφορετικοί τύποι Basic Stamp 2. Για την εκτέλεση της συγκεκριμένης εργασίας ο πιο απλός επεξεργαστής της σειράς καλύπτει τις απαιτήσεις και έχει αρκετές δυνατότητες για να καλύψει βελτιώσεις και αναβαθμίσεις τόσο σε προγραμματισμό όσο και σε υποστήριξη επιπλέον εξαρτημάτων και αισθητήρων. (βλ. Παράρτημα Γ.1. Χαρακτηριστικά Εξαρτημάτων). [2]



EIKONA 2.1 BASIC Stamp 2 (Rev. G) (Stock# BS2-IC).

2.2.2.2. Οδηγίες και Προφυλάξεις

Όπως με κάθε IC chip έτσι και με τον BASIC Stamp 2 πρέπει να ακολουθούνται οι παρακάτω οδηγίες.

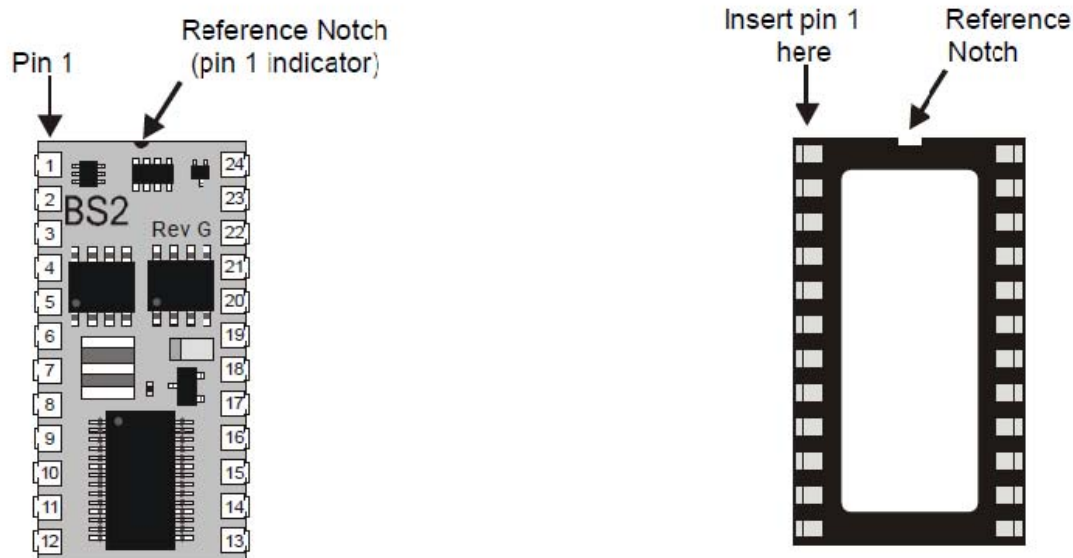
1. Προσοχή σε ευαίσθητες συσκευές στο στατικό ηλεκτρισμό.

Ο BASIC Stamp, όπως και άλλα IC, μπορεί να καταστραφεί από αποφόρτιση στατικού ηλεκτρισμού που συχνά εμφανίζεται αγγίζοντας γειωμένες επιφάνειες ή άλλους αγωγούς. Περιβαλλοντικές συνθήκες (αλλαγές υγρασίας, αέρας, επιφάνειες που εμφανίζουν στατικό ηλεκτρισμό, κλπ) έχουν σημαντικό ρόλο στην παρουσία τυχαίων στατικών φορτίων. Συνιστάται να χρησιμοποιείτε ταινίες γείωσης και αντιστατικά χαλάκια όταν χειρίζεστε συσκευές όπως ο BASIC Stamp. Εάν τα παραπάνω εξαρτήματα δεν είναι διαθέσιμα, να είστε βέβαιος ότι αγγίζετε μία γειωμένη επιφάνεια, πριν έρθετε σε επαφή με την κατασκευή.

2. Επιβεβαίωση ότι όλες οι τροφοδοσίες είναι κλειστές πριν την σύνδεση/αποσύνδεση εξαρτημάτων.

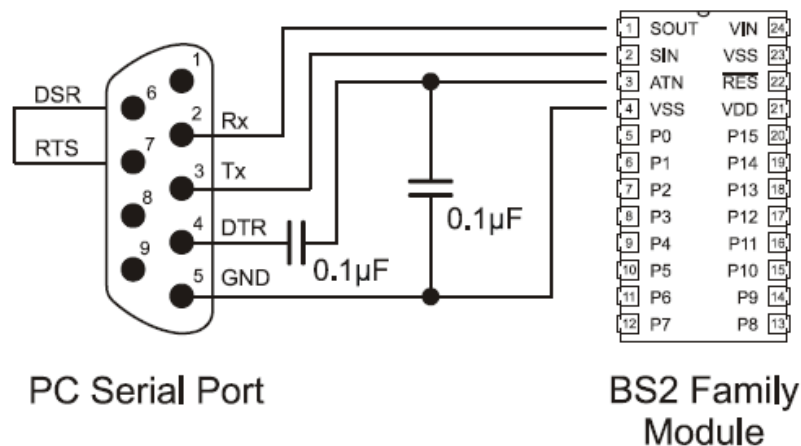
Εάν η τροφοδοσία είναι συνδεδεμένη στον BASIC Stamp οποιαδήποτε άλλο εξάρτημα καθώς το συνδέουμε η αποσυνδέουμε στο κύκλωμα, μπορεί να οδηγήσει στη καταστροφή του.

3. Προσοχή στην σωστή τοποθέτηση του BASIC Stamp πριν τη σύνδεσή του στην βάση σύνδεσης και άλλα κυκλώματα. Όπως όλα τα IC, ο BASIC Stamp θα πρέπει να τοποθετείται σε ένα συγκεκριμένο προσανατολισμό σε σχέση με την βάση του ή το κύκλωμα. Τροφοδοτώντας το κύκλωμα με ένα IC συνδεδεμένο ανάποδα μπορεί να βλάψει το IC και πιθανός άλλα εξαρτήματα στο κύκλωμα. Τα περισσότερα IC έχουν κάποια μορφή ένδειξης για τον πρώτο ακροδέκτη, όπως και στις περισσότερες βάσεις των IC. Ο δείκτης αυτός συνήθως είναι μια τελεία, ένα ημικύκλιο ή το νούμερο 1 γραμμένο κοντά στον ακροδέκτη 1 της συσκευής. Όλοι οι μικροελεγκτές της σειράς BS2 έχουν ένα μισό κύκλο για ένδειξη επάνω μέρος της μονάδας (βλ. Εικόνα 2.2). Αυτό δείχνει ότι ο αριθμός του πρώτου ακροδέκτη είναι ο πρώτος αριστερά από την εγκοπή. Η βάση που τον δέχεται έχει επίσης ένα ημικύκλιο ή εγκοπή στο ένα άκρο, υποδεικνύοντας τη σωστή φορά (Εικόνα 2.2.).



Εικόνα 2.2

Η συνδεσμολογία προγραμματισμού και επικοινωνίας σε πραγματικό χρόνο με θύρα RS-232 φαίνονται παρακάτω (Εικόνα 2.3.). Σημείωση: χρειάζεται επίσης η σύνδεση της τροφοδοσίας VIN και της γείωσης VSS που δεν φαίνονται στην εικόνα.



ΕΙΚΟΝΑ 2.3.

Η σύνδεση για τον προγραμματισμό του BASIC Stamp μέσω ηλεκτρονικού υπολογιστή γίνεται και με θύρα USB κατευθείαν συνδεδεμένη στην μητρική πλακέτα του συστήματος. Υπάρχει ενσωματωμένος μετατροπέας USB σε RS-232.

2.2.2.3. Γρήγορη Εισαγωγή στον BASIC Stamp

Σε αυτό το κεφάλαιο θα γίνει μία σύντομη εισαγωγή για την σύνδεση του BASIC Stamp με Η/Υ και τον προγραμματισμό του. Χωρίς να χρειάζονται ιδιαίτερες γνώσεις είναι εύκολο να επιτευχθεί η σύνδεση του BASIC Stamp με Η/Υ.

Ο εξοπλισμός που χρειάζεται είναι:

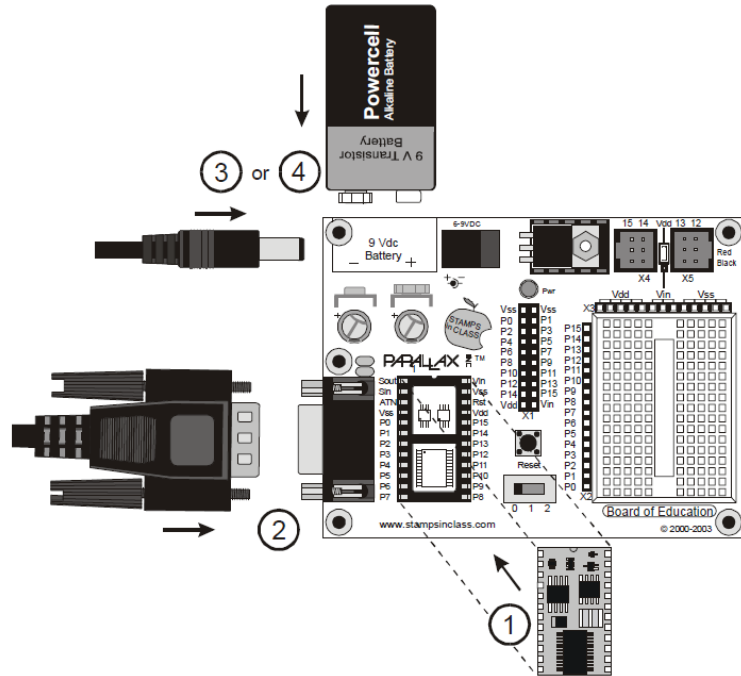
- ο μικροελεγκτής BASIC Stamp
- μια συμβατή πλακέτα σύνδεσης του BASIC Stamp και το καλώδιο σύνδεσης της πλακέτας με τον Η/Υ
- Πηγή ρεύματος ανάλογα με τις απαιτήσεις της πλακέτας σύνδεσης
- ένας Η/Υ με λειτουργικό σύστημα Windows® 2000/XP
- 3 MB χώρος στον σκληρό δίσκο
- Συσκευή CD-ROM ή σύνδεση στο διαδίκτυο
- Μία ελεύθερη θύρα USB στον Η/Υ

Για να τοποθετηθεί ο BASIC Stamp στην πλακέτα αναγνωρίζουμε την σωστή φορά από την εγκοπή και τοποθετούμε το IC στην υποδοχή του ώστε οι εγκοπές να ταιριάζουν. Υπόδειξη 1 στην Εικόνα 2.4. .

Συνδέουμε την πλακέτα με τον Η/Υ μέσω καλωδίου USB σε USB Mini. Υπόδειξη 2

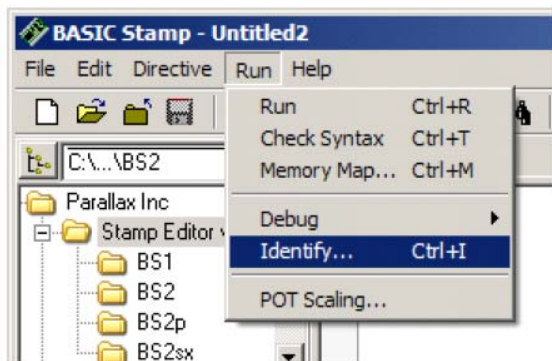
στην Εικόνα 2.4. .

Συνδέουμε τάση τροφοδοσίας 6-9volts στον αντάπτορα. Υπόδειξη 3 στην Εικόνα 2.4. .
 Ή μπορούμε να τροφοδοτήσουμε την πλακέτα με μπαταρία των 9V. Υπόδειξη 1 στην Εικόνα 2.4. .

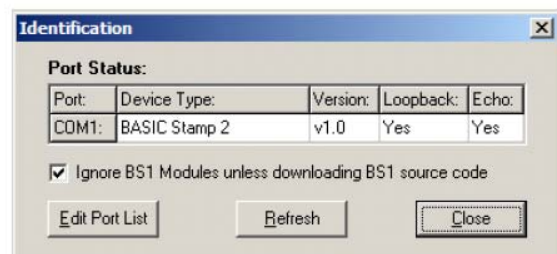


Εικόνα 2.4.

Τελειώνοντας με τη σύνδεση πρέπει να εγκατασταθεί το πρόγραμμα του Editor του BASIC Stamp. Αυτό βρίσκεται σε CD ή μπορούμε να το κατεβάσουμε από το site της Parallax και το εγκαθιστούμε. Μετά την εγκατάσταση τρέχουμε το πρόγραμμά του και στη συνέχεια δοκιμάζουμε την επικοινωνία H/Y με BASIC Stamp. Στο κεντρικό μενού του Editor επιλέγουμε Run → Identify, (Εικόνα 2.5.).



Select Run → Identify

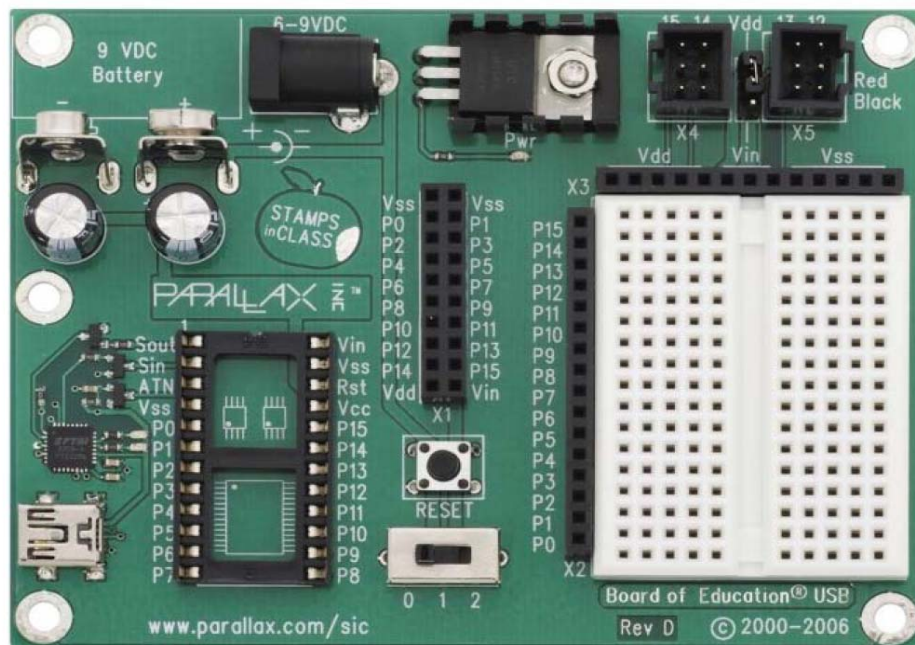


Verify that the BASIC Stamp was detected on one of the COM ports.

Εικόνα 2.5.

2.2.3. Μητρική Πλακέτα B.O.E. (Board Of Education® - USB (#28850))

Η Μητρική Πλακέτα B.O.E. (Board Of Education) (Εικόνα 2.6.) είναι μία ευέλικτη, χαμηλού κόστους πλατφόρμα ανάπτυξης σχεδιασμένη για όσους ενδιαφέρονται να μάθουν και να χρησιμοποιούν οποιοδήποτε μικροελεγκτής της Parallax®, BASIC Stamp των 24-pin. Το συμπαγές μέγεθος, οι βολικές λειτουργίες, και η χαμηλή τιμή του την καθιστούν ιδανικό εργαλείο για το σπουδαστές και εκπαιδευτικούς. Η BOE παρέχει μια εύχρηστη, αποτελεσματική πλατφόρμα ανάπτυξης για επιμέρους κυκλώματα για την λειτουργία αισθητήρων και λοιπών εξαρτημάτων. Είναι διαθέσιμη σε δύο τύπους σύνδεσης με H/Y μια USB την οποία χρησιμοποιούμε και μια σειριακή έκδοση. Η επιλογή της σύνδεσης με USB έγινε με βάση τα δεδομένα του εμπορίου καθώς τις θύρες σύνδεσης USB τις συναντάμε πλέον παντού σε σχέση με την σειριακή θύρα RS-232. (βλ. Παράρτημα Γ.2. Χαρακτηριστικά Εξαρτημάτων). [3]

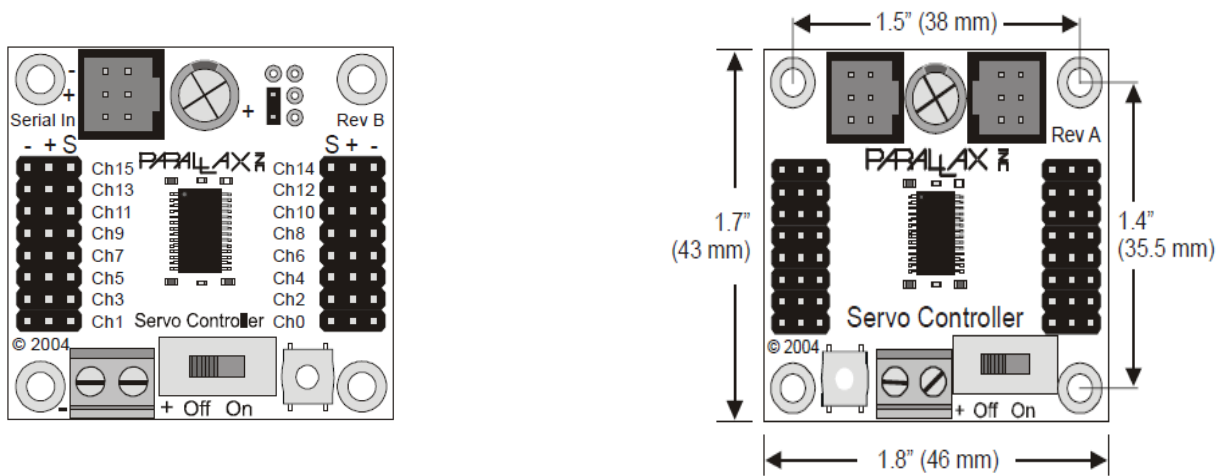


Εικόνα 2.6. Μητρική Πλακέτα B.O.E. (Board Of Education)

2.2.4. Ελεγκτής Σερβομοτέρ της Parallax (#28023) – Rev B (Parallax Servo Controller (#28023) – Rev B)

Ο ελεγκτής της Parallax (#28023) – Rev B ή PSC (Εικόνα 2.7.), έχει σχεδιαστεί για να μπορεί να ελέγχει μέχρι και 16 σερβομοτέρ και σε σύνδεση με ένα δεύτερο PSC μέχρι και 32 χρησιμοποιώντας μόνο μία απλή γραμμή I/O του μικροελεγκτή. Ο PSC διαχειρίζεται τους παλμούς για όλα τα σερβομοτέρ μόνος του και με τον τρόπο αυτό εξοικονομεί πόρους από τον κεντρικό ελεγκτή του συστήματος. Με την λειτουργία ramping επιτρέπει την ξεχωριστή ρύθμιση της ταχύτητας του κάθε σερβομοτέρ, δηλαδή μπορείς να δηλώσεις την θέση που

θέλεις να πάει και το πόσο γρήγορα. Το αποτέλεσμα ελέγχεται για να είναι το προκαθορισμένο, από τη λειτουργία “set-it and forget-it”. (βλ. Παράρτημα Γ.3.



Χαρακτηριστικά Εξαρτημάτων). [4]

Εικόνα 2.7.

Parallax Servo Controller, typical Connections

2.2.5. Σερβομοτέρ

Για την κίνηση των ποδιών απαιτείται η χρήση κινητήρων με ελεγκτή ώστε να είναι δυνατή η παρακολούθηση της θέσης της ταχύτητας και της γωνίας κλίσης που θα κάνει το καθένα από αυτά. Για το σκοπό αυτό χρησιμοποιήθηκαν κοινά σερβομοτέρ από είδη μοντελισμού TowerPro MG995 (Εικόνα 2.8.). Τα σερβομοτέρ αυτά είναι αντίστοιχα με τα γνωστά στο μοντελισμό HiTec HS-322HD heavy duty Servos. Είναι κινέζικης κατασκευής και επιλέχτηκαν λόγω της πολύ χαμηλότερης τιμής τους. (βλ. Γ.4. Χαρακτηριστικά Εξαρτημάτων). [5]



Εικόνα 2.8. Σερβομοτέρ MG-995

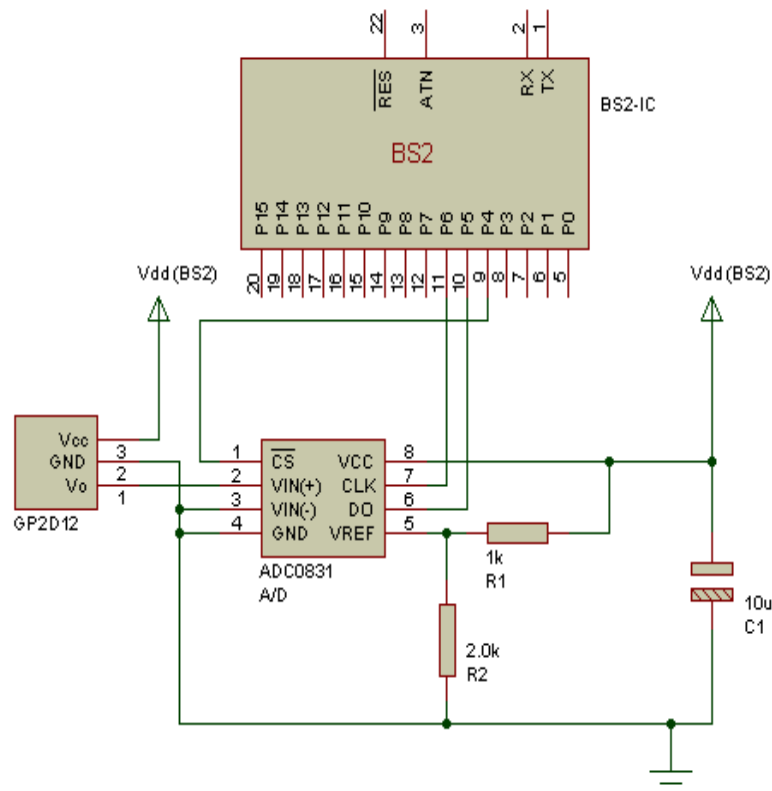
2.2.6. Αισθητήρας απόστασης Sharp IR GP2D12 και Αναλογικός σε Ψηφιακό Μετατροπέα (Analog to Digital Converter) ADC0831

Ο Αισθητήρας απόστασης Sharp IR GP2D12 και ο Αναλογικός σε Ψηφιακό Μετατροπέα (Analog to Digital Converter) ADC0831 είναι δύο στοιχεία που θα χρησιμοποιούνται σαν ένα εξάρτημα, αφού η συνεργασία τους με τον μικροελεγκτή είναι αλληλοεξαρτώμενη.

2.2.6.1. Αισθητήρας απόστασης Sharp IR GP2D12

Τα προϊόντα της SHARP είναι πολύ γνωστά στη ρομποτική και σε εφαρμογές αυτόματης μέτρησης αποστάσεων. Για την μέτρηση της απόστασης των εμποδίων σε αυτή την κατασκευή επιλέχθηκε από τα προϊόντα της SHARP ο ανιχνευτής απόστασης με υπέρυθρες GP2D12. (βλ. Παράρτημα Γ.5. Χαρακτηριστικά Εξαρτημάτων).

Για την μετατροπή της αναλογικής εξόδου του αισθητήρα GP2D12 σε ψηφιακό σήμα, χρησιμοποιήθηκε ένας αναλογικός σε ψηφιακό μετατροπέα ADC0831, ώστε το σήμα που θα καταλήξει στον μικροεπεξεργαστή να είναι αναγνωρίσιμο εύκολα και εύχρηστο στην επεξεργασία και τη χρήση του στο πρόγραμμα. [6], [7].



Σχέδιο 2.1.

Σχέδιο σύνδεσης GP2D12 με ADC0831 στον BS2

2.2.6.2. Αναλογικός σε Ψηφιακό Μετατροπέα (Analog to Digital Converter) ADC0831

Ο ADC0831 είναι ένας 8-bit αναλογικός σε ψηφιακό μετατροπέα διαδοχικής προσέγγισης με σειριακό I/O, σε μορφή ολοκληρωμένου κυκλώματος IC, με εσωτερικά διαμορφωμένους πολυπλέκτες έως και 8 καναλιών. Η σειριακή I/O είναι κατασκευασμένη ώστε να συμμορφώνεται με τα πρότυπα σειριακής ανταλλαγής δεδομένων της NSC MICROWIRE™ για εύκολη σύνδεση με μικροελεγκτές της οικογένειας COPS™ και μπορεί να συνδεθεί με τους πρότυπους καταχωρητές μετατόπισης ψηφίων ή μPs. Οι πολυπλέκτες των 2, 4 ή 8 καναλιών ρυθμίζονται μέσω λογισμικού για μονή έξοδο ή για διαφορικές εισόδους όπως εκχώρηση καναλιών. Η διαφορική αναλογική τάση εισόδου επιτρέπει την αύξηση της απόρριψης της κοινής κατάστασης και μετατοπίζει την αναλογική μηδενική τιμή της τάσης εισόδου. Επιπλέον, η τάση εισόδου αναφοράς μπορεί να προσαρμοστεί για να επιτρέψει της κωδικοποίηση οποιασδήποτε μικρότερης αναλογικής τάσης στα πλήρες 8 bits της ανάλυσης. Τα 8-bit είναι ο δυαδικός αριθμός που χρησιμοποιεί ο ADC0831 για να περιγράψει τα δείγματα της αναλογικής τάσης. 8-bit είναι η ανάλυση του αναλογικού σε ψηφιακό μετατροπέα. Αυτό σημαίνει ότι ο ADC0831 μπορεί να προσεγγίσει την τάση που μετράει και να την εκφράσει σαν ένα από τα 256 επίπεδα. (βλ. Παράρτημα Γ.6. Χαρακτηριστικά Εξαρτημάτων). [8]

Δυαδικό σήμα ελέγχου: Ένα σήμα τάσης με δύο πιθανές καταστάσεις, χαμηλή ή υψηλή τάση, στέλνεται για να πει στην συσκευή πότε και πως να κάνει κάτι. Ο ADC0831 χρειάζεται ένα σήμα ελέγχου για να ενεργοποιηθεί και παλμούς χρονιστή για να συγχρονίσει την αποστολή καθενός από τα bit που θα στείλει η έξοδος. Και τα δύο είναι είσοδοι ελέγχου για το δυαδικό σήμα.

Για να διεγερθεί ο ADC0831 για να πάρει μέτρηση, το /CS pin πρέπει να πάρει ένα σήμα από τον ελεγκτή (BASIC Stamp) που ξεκινάει σε κατάσταση High και μετά γίνεται Low. Αυτό το σήμα πρέπει να είναι σε κατάσταση Low για όσο κρατάει η μετατροπή του αναλογικού σε ψηφιακό σήμα. Μετά στο άκρο CLK πρέπει να σταλεί ένας παλμός του χρονιστή για να δηλώσει ότι η μετατροπή πρέπει να αρχίσει στον επόμενο παλμό. Σε αυτό το IC ο παλμός πρέπει να ξεκινάει σε κατάσταση Low, να γίνεται High και μετά Low ξανά. Για να ολοκληρωθεί η μετατροπή χρειάζονται ακόμα 8 παλμοί. Κάθε φορά που ένας παλμός λαμβάνεται στην είσοδο του CLK από την έξοδο D0 φεύγει και ένα bit του σειριακού ψηφιακού σήματος. Η διαδικασία λειτουργίας φαίνεται στο εγχειρίδιο τεχνικών χαρακτηριστικών του εξαρτήματος, στον χάρτη λειτουργίας και αντίδραση των άκρων στα σήματα των εισόδων του.

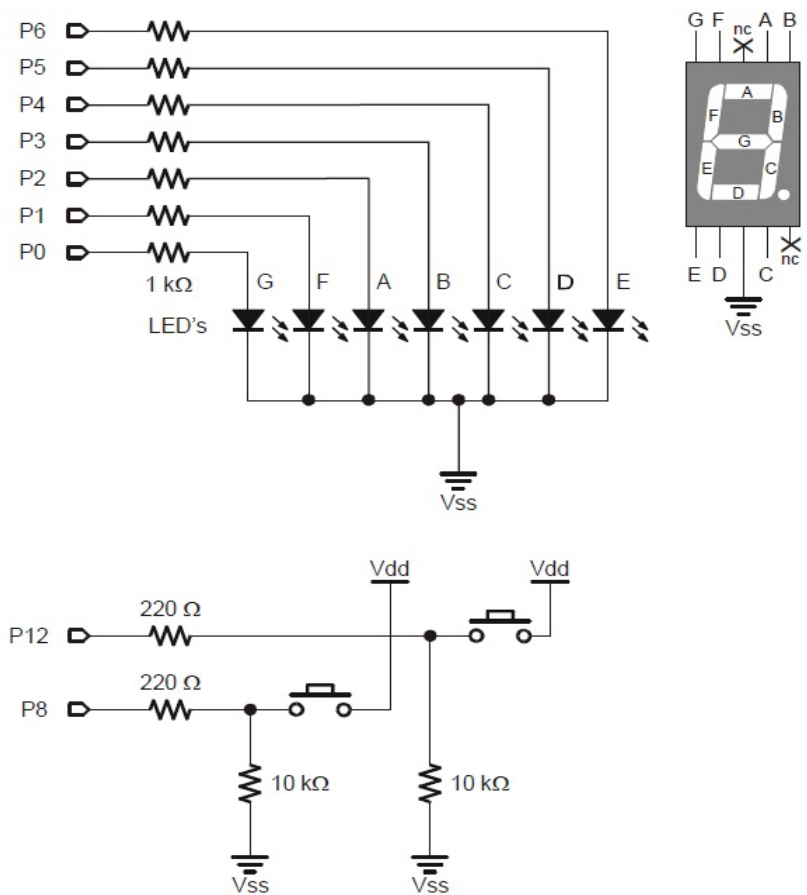
2.2.7. PUSH BUTTON και κύκλωμα LED

Για την επιλογή του τρόπου περπατήματος χρειάζεται να προστεθούν στον μικροελεγκτή δύο button. Το ένα θα επιλέγει την επόμενη κατάσταση, ενώ το άλλο την προηγούμενη. Η σύνδεσή τους θα γίνει σε κάποιο από τα ελεύθερα pin I/O του μικροελεγκτή. Για τη σύνδεσή τους χρησιμοποιούνται 2 αντιστάσεις σε κάθε button, μία 220Ω και μία 10KΩ. Η αντίσταση των 220Ω χρησιμοποιείται για να περιορίζει την τάση Vdd να περνάει ακέραια προς τον μικροελεγκτή όταν πατιέται το button. Η δεύτερη των 10KΩ είναι για την διαφυγή της τάσης προς την γείωση (Σχέδιο 2.2.). [7]

Για να γνωρίζει ο χρήστης την επιλογή προστέθηκε και ένα LED Display που θα χρησιμοποιηθεί για την οπτική αναγνώριση της κάθε κίνησης. Η τάση των Pin όταν είναι σε λειτουργία PinOUT, έξοδοι, είναι 5V ενώ η τάση λειτουργίας των Led στο Display 1.5V. Για αυτόν το λόγο πρέπει να γίνει μία πτώση τάσης με την χρήση αντίστασης 1KΩ. Για τον σχεδιασμό των αριθμών θα γραφτεί μέσα στο πρόγραμμα μια ακολουθία από δεδομένα μεγέθους Nib. Παρακάτω δίνεται ο πίνακας με τις αντιστοιχίες (Πίνακας 2.1.).

Segmet s	No.	.	e	d	c	b	a	f	g	HEX
	0	%	0	1	1	1	1	1	0	\$7E
	1	%	0	0	0	1	1	0	0	\$18
	2	%	0	1	1	0	1	1	0	\$6D
	3	%	0	0	1	1	1	0	1	\$3D
	4	%	0	0	0	1	1	0	1	\$1B
	5	%	0	0	1	1	0	1	1	\$37
	6	%	0	1	1	1	0	1	1	\$77
	7	%	0	0	0	1	1	1	0	\$1C
	8	%	0	1	1	1	1	1	1	\$7F
	9	%	0	0	0	1	1	1	1	\$1F
	A	%	0	1	0	1	1	1	1	\$5F
	B	%	0	1	1	1	0	0	1	\$73
	C	%	0	1	1	0	0	1	0	\$66
	D	%	0	1	1	1	1	0	0	\$79
	E	%	0	1	1	0	0	1	1	\$67
	F	%	0	1	0	0	0	1	1	\$47

Πίνακας 2.1.



Σχέδιο 2.2.

2.2.8. Πηγές Τάσης

Οι πηγές τάσης που χρησιμοποιούνται είναι δύο. Η μία είναι για την τροφοδοσία την πλακέτας BOE και η άλλη για την τροφοδοσία του ελεγκτή των σερβομοτέρ PSC και τους σερβομοτέρ. Η τροφοδοσία της BOE μπορεί να γίνει με δύο τρόπους, μπαταρία 9V ή με την χρήση εξωτερικής πηγής που να δίνει τάση 6-9 V. Για τον PSC μπορούμε να συνδέσουμε μπαταρία 7.2V όπως αυτές που χρησιμοποιούν στο μοντελισμό ή εξωτερική τροφοδοσία 7.2V. Για τους πειραματισμούς και τις δοκιμές επιλέχτηκε η χρήση μπαταρίας στην BOE και ενός τροφοδοτικού με παροχή 2250mA για τον PSC.

2.2.9. Λόγοι επιλογής

Οι λόγοι επιλογής των εξαρτημάτων βασίζονται αρχικά στις απαιτήσεις της κατασκευής, την ευκολία χρήσης και την αξιοπιστία τους σε σχέση με την τιμή που κοστίζουν.

Ο BASIC Stamp 2 είναι ένας πολύ καλός επεξεργαστής με πολλές δυνατότητες. Είναι γνωστός σε μεγάλο μέρος χρηστών μικρών αυτόνομων ελεγκτών κάτι που είναι χρήσιμο για την ολοκλήρωση της κατασκευής. Η επιλογή και η χρήση του αισθητήρα απόστασης και του

αναλογικού σε ψηφιακό μετατροπέα έγινε σαν σύνολο με τον ελεγκτή αυτόν, αφού ήταν εύκολο να βρεθούν πληροφορίες για τον προγραμματισμό και τη χρήση τους. Ένας ακόμα λόγος είναι ότι παρόλο το ότι έχει αρκετά περισσότερες δυνατότητες από αυτές που ζητάει η κατασκευή, μετά το τέλος της πτυχιακής αυτής άσκησης θα μπορεί να αντεπεξέλθει σε διάφορες βελτιώσεις και αναβαθμίσεις της χωρίς να χρειάζεται αντικατάσταση.

Η πλακέτα BOE επιλέχθηκε γιατί είναι ένα προϊόν της εταιρίας που κατασκευάζει τον BASIC Stamp και την έχει δημιουργήσει ειδικά για να χρησιμοποιεί όλους του επεξεργαστές της σειράς BASIC Stamp. Είναι εύχρηστη και παρέχει πολλές θύρες επικοινωνίας με τις I/O του ελεγκτή, καθώς και θύρες με τάσεις τροφοδοσίας. Έχει ενσωματωμένο τον μετατροπέα USB σε RS-232 οπότε μειώνεται το κόστος για αγορά εξωτερικού μετατροπέα. Επίσης είναι κατασκευασμένη για χρήση σε κατασκευές ελέγχου σερβομοτέρ και έχει από μόνη της θύρα επικοινωνίας με ελεγκτές σερβομοτέρ.

Ο ελεγκτής PSC επιλέχθηκε για την ευκολία στην χρήση του και τον προγραμματισμό, καθώς λειτουργεί με λίγες και απλές εντολές. Οι δυνατότητές του είναι πολλές καθώς μπορεί να ελέγχει 16 διαφορετικά σερβομοτέρ ταυτόχρονα με τη χρήση μόνο ενός καλωδίου σύνδεσης για δεδομένα με τον κύριο ελεγκτή. Οι δυνατότητές του για μελλοντικές βελτιώσεις του ρομποτικού είναι μεγάλες αφού μπορεί να συνδέεται παράλληλα και με άλλους PSC και να υποστηρίζονται ακόμα περισσότερα σερβομοτέρ.

Τα σερβομοτέρ της TOWER-Pro είναι αντίστοιχα αυτών της HiTec, HS-322HD heavy duty, αλλά η τιμή τους πολύ χαμηλότερη. Επιλέχθηκαν επίσης γιατί τα εσωτερικά τους γρανάζια είναι μεταλλικά και όχι πλαστικά, κάτι που τους δίνει μεγαλύτερη ανθεκτικότητα και ακρίβεια.

Ο ADC0831 και ο αισθητήρας GP2D12 επιλέχθηκαν λόγω της μεγάλης χρήσης που έχουν σε κατασκευές, το οποίο βοήθησε να βρεθούν πληροφορίες για τη χρήση τους και τον προγραμματισμό τους με τον BASIC Stamp. Σαν εξαρτήματα η τιμή τους είναι η πιο χαμηλή στην κατηγορία τους κάτι που ήταν καθοριστικό για την επιλογή τους.

Η επιλογή του τροφοδοτικού για την παροχή ρεύματος στον PSC έγινε λόγω κόστους αφού μία μπαταρία κοστίζει σχεδόν δέκα φορές παραπάνω.

2.2.10. Κόστος Κατασκευής.

Το κόστος κατασκευής φαίνεται αναλυτικά στον παρακάτω πίνακα (Πίνακας 2.2.).

ΕΞΑΡΤΗΜΑ	Ποσότητα	Τιμή σε Ευρώ
Υλικά αλουμινίου		10
Βίδες, παξιμάδια, παξιμάδια ασφαλείας, ροδέλες, ροδέλες τεφλόν,		10
BASIC STAMP 2	1	50
Board of Education USB, καλώδιο USB σε USB Mini-2	1	70
Ελεγκτής σερβομοτέρ Parallax PCS	1	37
Σερβομοτέρ TowerPro MG955	8 + 8	7*16
Ανιχνευτής απόστασης με υπέρυθρες GP2D12	1	12
ADC0831	1	4
Τροφοδοτικό 7.5V 2250μΑ	1	9
Μπαταρία 9V	1	2
<ul style="list-style-type: none"> • (1) ενδεικτική οθόνη 7 στοιχείων LED Display • (7) αντιστάσεις 1kΩ • (2) αντιστάσεις 220Ω • (2) αντιστάσεις 10kΩ • (2) pushbuttons • Καλώδια διαφόρων χρωμάτων 		10
Σύνολο		326.50Ευρώ

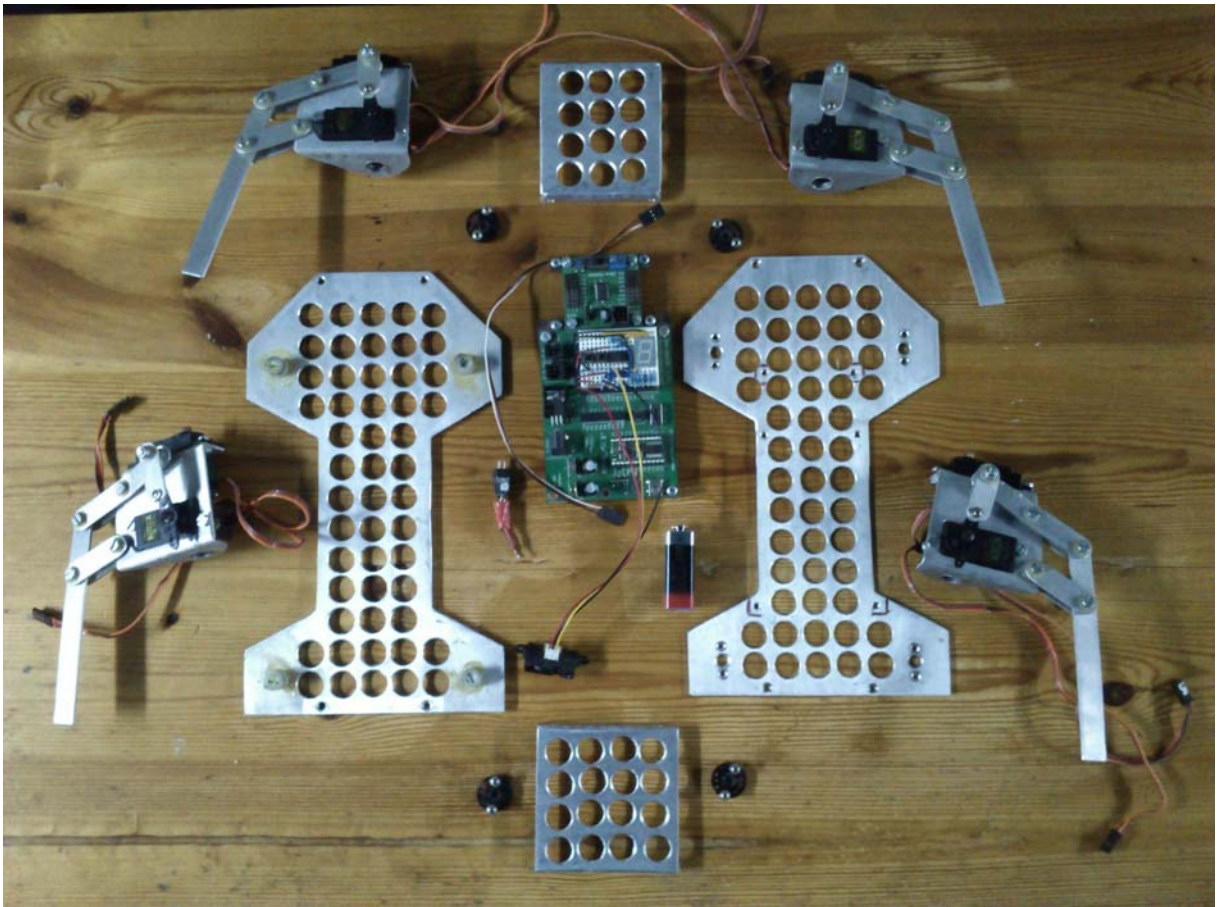
Πίνακας 2.2.

2.3. Κατασκευή και Συναρμολόγηση

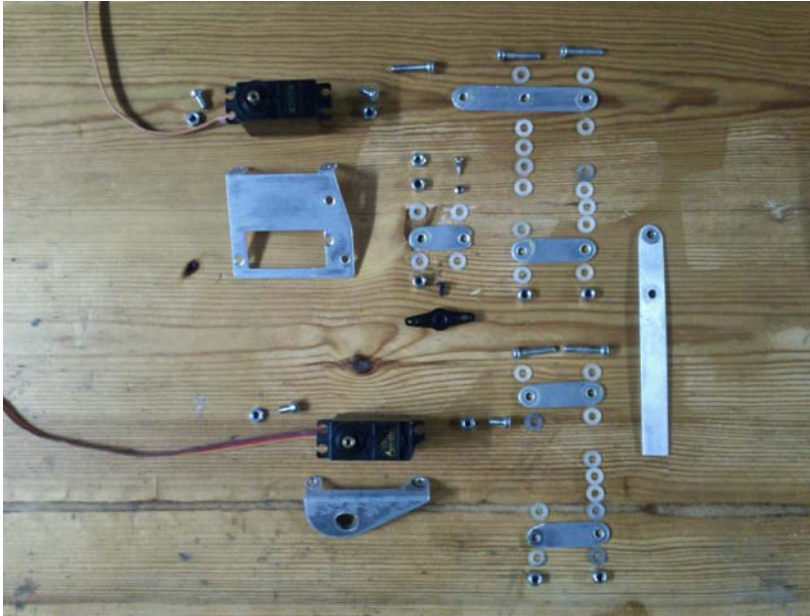
2.3.1. Μηχανικό Μέρος

Για την υλοποίηση της κατασκευής χρειάστηκε η χρήση εργαστηρίου επεξεργασίας μετάλλων καθώς ολόκληρος ο σκελετός του ρομποτικού φτιάχτηκε από ένα κομμάτι λαμαρίνας αλουμινίου. Τα κομμάτια του σώματος και των ποδιών κατασκευάστηκαν από φύλλο αλουμινίου 2mm. Μετά από τον υπολογισμό των διαστάσεων των κομματιών, σηματοδεύτηκαν πάνω στο φύλλο αλουμινίου και κόπηκαν με τη χρήση δισκοπρίονου πάγκου. Τα κομμάτια αφού κόπηκαν, επεξεργάστηκαν στις άκρες τους ώστε να λειανθούν οι αιχμηρές άκρες τους. Τα μέρη του σώματος, πάνω και κάτω πάτωμα, και οι δύο κάθετες στηρίξεις δέχτηκαν ακόμα μια επεξεργασία. Ανοίχτηκαν τρύπες των 15mm σε σημεία που δεν επηρεάζουν τις απαιτούμενες αντοχές στρέβλωσης του ρομποτικού, ώστε να αφαιρεθεί μια αρκετά μεγάλη ποσότητα υλικού που θα οδηγήσει στην μείωση του βάρους του ρομποτικού.

Στην Εικόνα 2.9. και 2.10. φαίνονται τα κομμάτια από τα οποία αποτελείται το ρομποτικό. Στην εικόνα 2.13. φαίνονται αναλυτικά όλα τα κομμάτια που αποτελούν το κάθε πόδι και είναι τοποθετημένα με την σειρά που συναρμολογούνται.

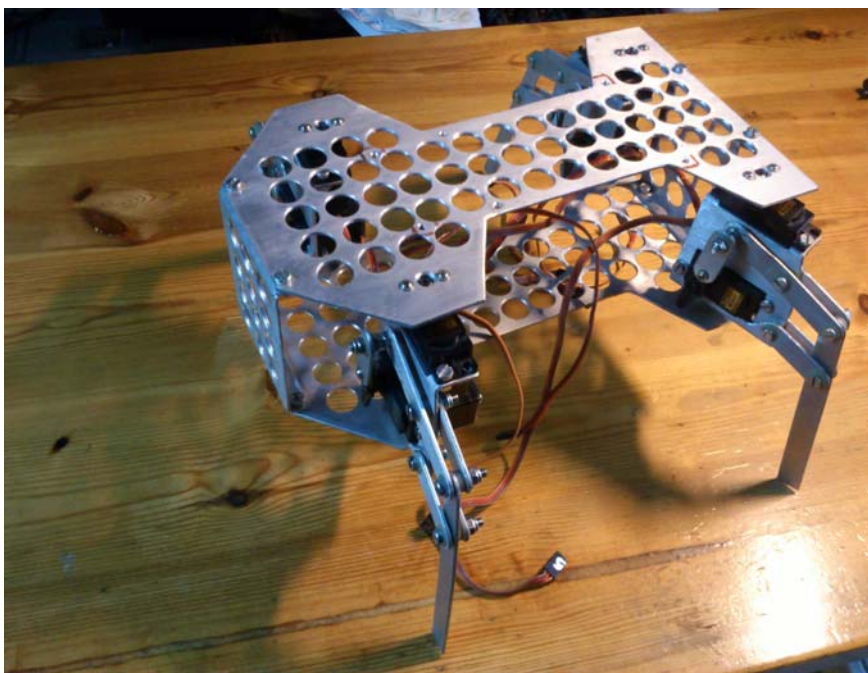


Εικόνα 2.9.



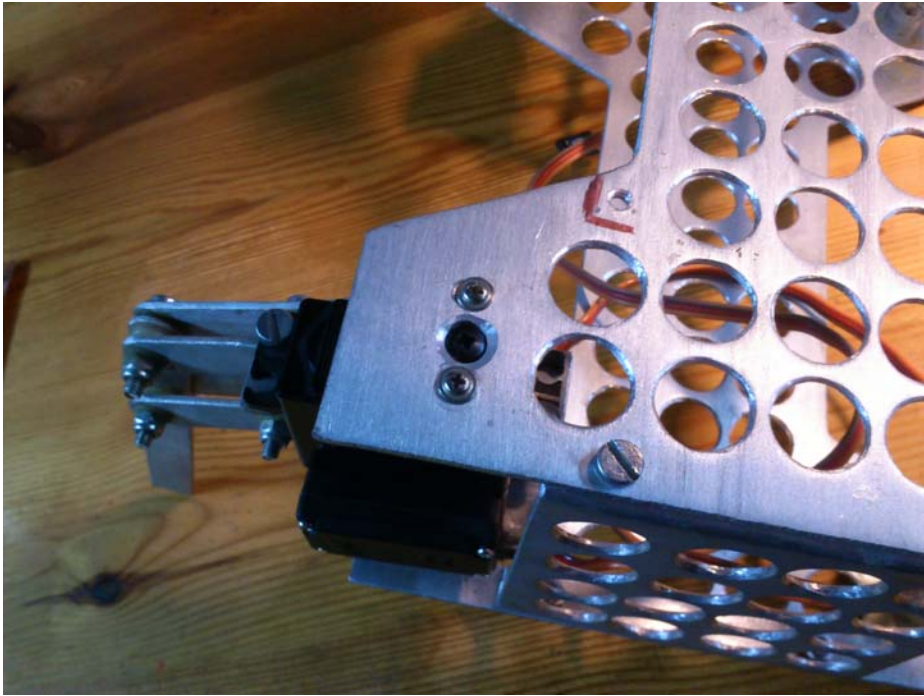
Εικόνα 2.10.

Στην εικόνα 2.11. φαίνεται ο σκελετός του ρομποτικού συναρμολογημένος με τα σερβομοτέρ τοποθετημένα, αφού αποτελούν μέρος στήριξης των ποδιών με το σώμα. Η πάνω και η κάτω πλάκα αποτελούν τα πατώματα του ρομποτικού όπου εκεί τοποθετούνται όλα τα απαραίτητα εξαρτήματα. Ο χώρος που έχουν ανάμεσα και στο πάνω μέρος είναι αρκετός για τις απαιτήσεις που έχει η κατασκευή αυτή τη στιγμή, αλλά και για μελλοντικές αναβαθμίσεις. Εδώ φαίνονται και οι προεξοχές που έχουν γίνει για τις βάσεις των ποδιών, ώστε να μην είναι δυνατόν τα πόδια την στιγμή της κίνησης να ακουμπήσουν στο υπόλοιπο σώμα και να είναι δυνατή η χρήση όλου του εύρους κίνησης. Οι κάθετες στηρίξεις δένονται με βίδες με τις πλάκες των πατωμάτων και ολοκληρώνουν εύκολα το βασικό σώμα.

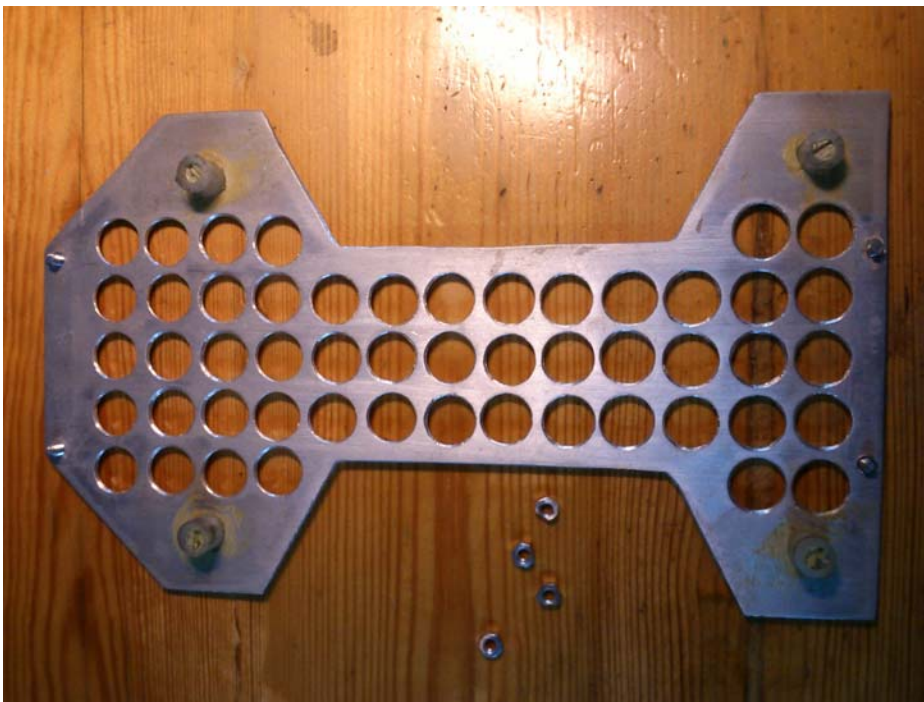


Εικόνα 2.11.

Στην πάνω πλάκα των πατωμάτων υπάρχουν ζευγάρια τρύπες κοντά στις τέσσερις γωνίες της , στις οποίες βιδώνουν οι πλάκες από τα εξαρτήματα των σερβομοτέρ, που θα δεθούν τα σερβομοτέρ της οριζόντιας κίνησης. Ανάμεσά τους υπάρχει μια μεγαλύτερη τρύπα, μέσα από την οποία περνάει η βίδα που σταθεροποιεί το σερβομοτέρ με το σώμα του ρομποτικού ή αλλιώς το πόδι με το σώμα (Εικόνα 2.12.). Στην κάτω πλάκα είναι βιδωμένες κυλινδρικές βάσεις από τεφλόν που αποτελούν το κάτω μέρος στήριξης των ποδιών (Εικόνα 2.13.).



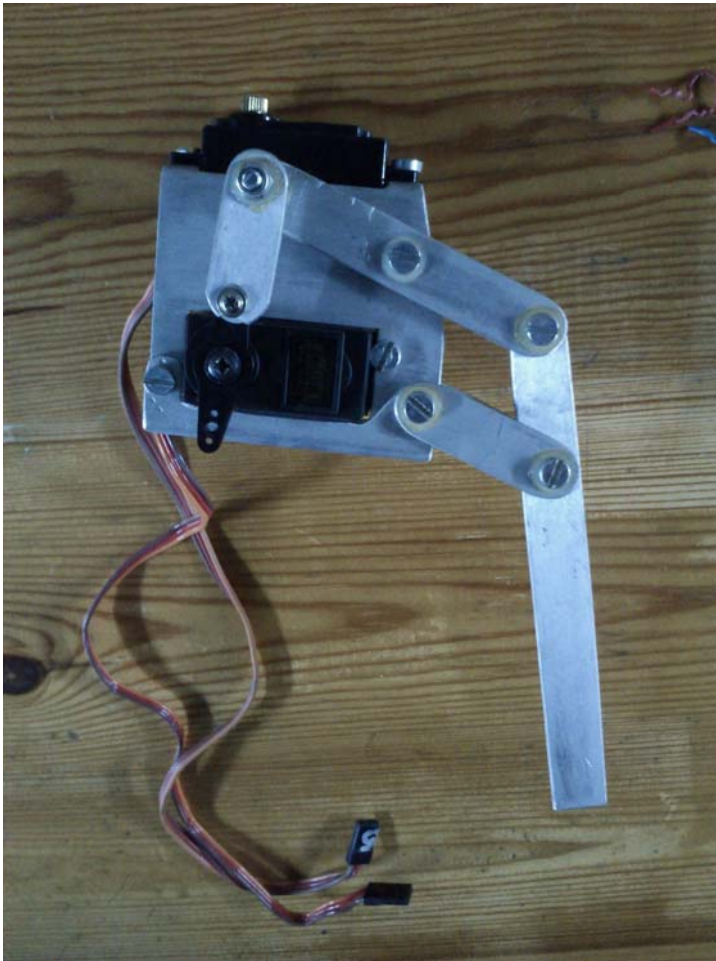
Εικόνα 2.12.



Εικόνα 2.13.

Τα πόδια (Εικόνα2.14) είναι το πιο σύνθετο μέρος. Πάνω τους είναι τοποθετημένα τα σερβομοτέρ, δύο σε κάθε πόδι. Αποτελούνται από δύο βασικά μέρη ενωμένα μεταξύ του με βίδες. Ο λόγος που έγιναν σε δύο κομμάτια ήταν για την ευκολία και την εξοικονόμηση χρόνου στην κατασκευή, γιατί ήταν πολύ πιο εύκολο με τον τρόπο αυτό να γίνουν οι υποδοχές για τα σερβομοτέρ. Τα υπόλοιπα κομμάτια των σερβομοτέρ που αποτελούν τα κινούμενα μέρη του κάθετου άξονα κίνησης. Είναι ενωμένα με βίδες μεταξύ τους και ανάμεσά τους έχουν τοποθετηθεί πολλές ροδέλες από τεφλόν, τόσο για να ρυθμιστούν οι αποστάσεις και τα κενά μεταξύ τους όσο και για να υπάρχει κάποιο υλικό με λιγότερες τριβές που θα βοηθάει και στην συγκράτηση λίπανσης. Για αυτούς τους λόγους χρησιμοποιήθηκαν περισσότερες από δύο ροδέλες όπου ήταν δυνατόν.

Για την τοποθέτηση των ποδιών πάνω στο σώμα χρειάζεται μόνο να μπουν με κλίση από την κάτω πλευρά πρώτα στην κυλινδρική βάση και στην συνέχεια να τραβηχτούν προς τα πάνω και να σφηνώσει ο άξονας του σερβομοτέρ οριζόντιας κίνησης στην βάση του πάνω πατώματος.



Εικόνα 2.14.

Τελειώνοντας η συναρμολόγηση του σώματος σειρά έχουν τα ηλεκτρονικά εξαρτήματα (Εικόνα 2.15.). Στο πάνω πάτωμα με βίδες στηρίζονται η BOE και η PSC. Στο κάτω μέρος τους για μόνωση και για αποστάτες από το μεταλλικό μέρος της κατασκευής υπάρχουν κομμάτια μονωτικού υλικού.

2.3.2. Συναρμολόγηση Ηλεκτρονικού Μέρους

Τα καλώδια των σερβομοτέρ που συνδέονται στον PSC ακολουθούν μία σειρά που επιλέχτηκε από την αρχή για να μην χρειάζονται αλλαγές σε ότι ορίζουμε στο πρόγραμμα.

Οι θέσεις τους είναι:

Ch.0 – Μπροστά Δεξιά οριζόντιο

Ch.2 – Πίσω Δεξιά οριζόντιο

Ch.4 – Μπροστά Αριστερά οριζόντιο

Ch.6 – Πίσω Αριστερά οριζόντιο

Ch.1 – Μπροστά Δεξιά Κάθετο

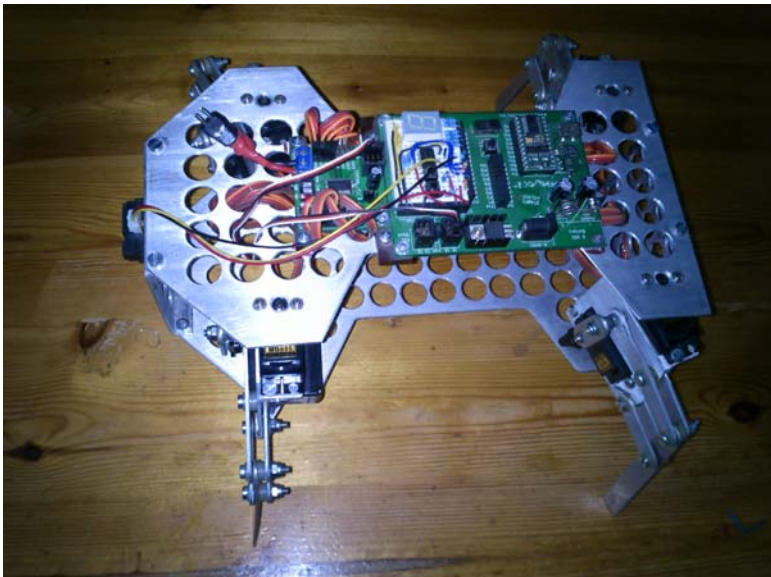
Ch.3 – Πίσω Δεξιά Κάθετο

Ch.5 – Μπροστά Αριστερά Κάθετο

Ch.7 – Πίσω Αριστερά Κάθετο

Από την σειριακή θύρα του PSC συνδέουμε καλώδιο με την σειριακή θύρα X4 – 15 της BOE και το βύσμα για την τροφοδοσία της.

Στην μπροστά πλευρά τοποθετείται ο αισθητήρας απόστασης και το καλώδιο του οδηγείται πάνω στην BOE και συνδέεται στην διάτρητη πλακέτα στις θέσεις όπως φαίνεται στο σχέδιο σύνδεσης GP2D12 με ADC0831 στον BS2 (Σχέδιο 2.2.) για να πάρει τροφοδοσία και να επιστρέψει στον ADC0831 την τάση της μετρούμενης απόστασης.



Εικόνα 2.15.

Στο Παράρτημα Β. Συναρμολόγηση Του Ρομποτικού, υπάρχουν αναλυτικά οι οδηγίες με εικόνες για την συναρμολόγηση του ρομποτικού.

ΚΕΦΑΛΑΙΟ 3^ο

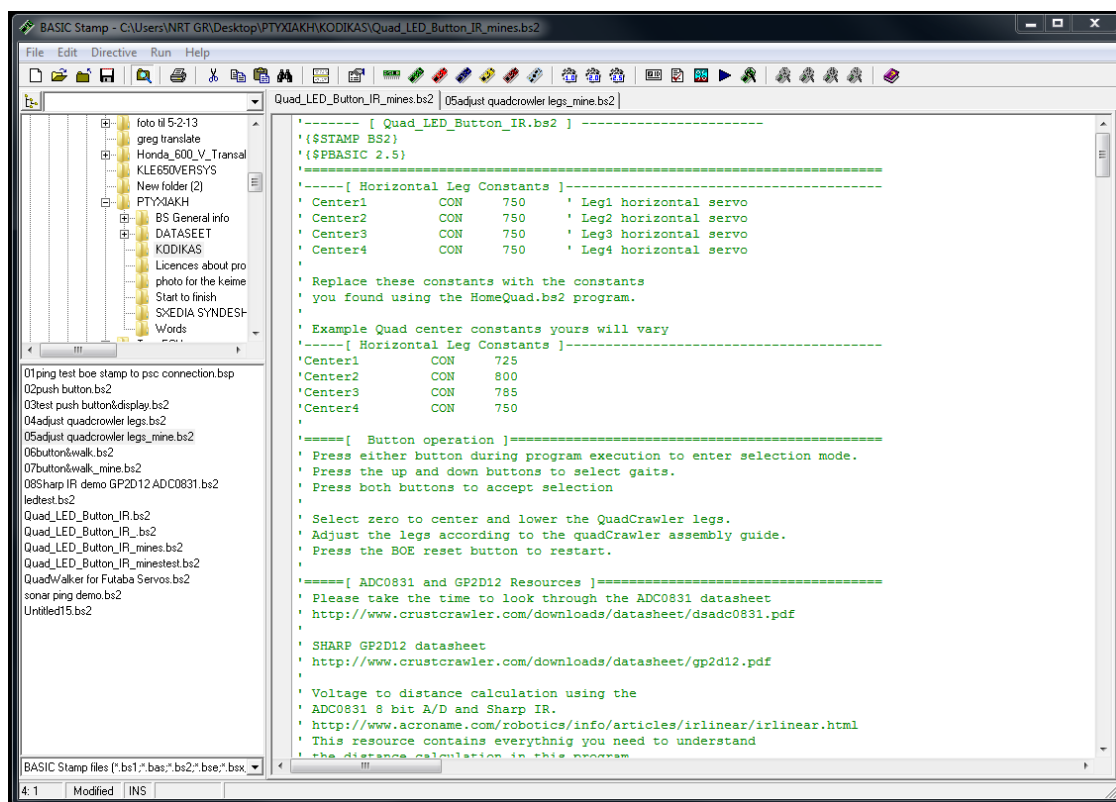
3. Προγραμματισμός

3.1. Γλώσσα προγραμματισμού

Η γλώσσα προγραμματισμού είναι η PBASIC, ειδικά γραμμένη για τους μικροελεγκτές BASIC Stamp. Έχει αναπτυχθεί με τρόπο ώστε να είναι απλή και γρήγορη στην εκμάθηση. Βασικό χαρακτηριστικό της είναι η αρχιτεκτονική της που βοηθάει την χρήση των BASIC Stamp σε αυτόνομα συστήματα. Περιλαμβάνει πολλές λειτουργίες και εντολές που έχουν άλλες γλώσσες προγραμματισμού τύπου BASIC (GOTO, FOR...NEXT, IF...THEN...ELSE) και ειδικές λειτουργίες (SERIN, PWM, BUTTON, COUNT και DTMFOUT). Λειτουργεί σε λογισμικό Windows και LINUX και η τελευταία εκδοσή της υποστηρίζει όλους τους ελεγκτές BASIC Stamp καθώς και τις προηγούμενες της εκδόσεις. [2]

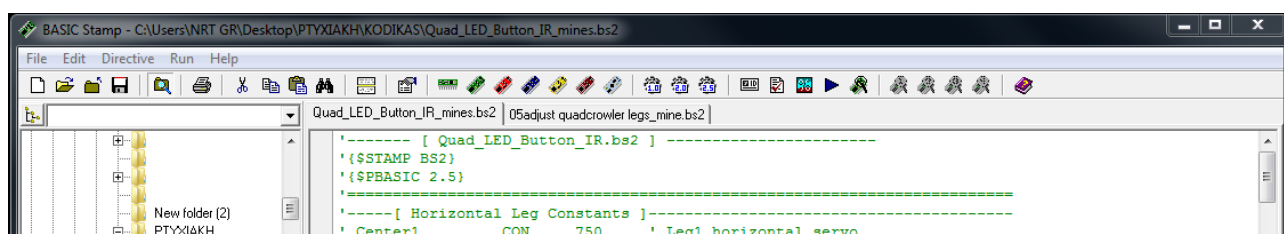
3.2. Περιβάλλον Προγραμματισμού

Το πρόγραμμα σύνταξης (Editor) του BASIC Stamp για τα Windows, φαίνεται στην εικόνα 3.1., έχει σχεδιαστεί για να είναι εύκολο στη χρήση και να παρέχει έξυπνες λειτουργίες. Αυτοί που είναι εξοικειωμένοι με το τυποποιημένο λογισμικό των Windows θα πρέπει να αισθάνονται άνετα με τη χρήση του προγράμματος.



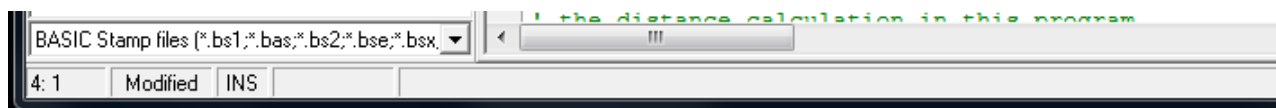
Εικόνα 3.1. BASIC Stamp Editor

Το παράθυρο του προγράμματος σύνταξης BASIC Stamp Editor, αποτελείται από το κύριο παράθυρο επεξεργασίας, με ενσωματωμένο πάνελ explorer στα αριστερά του, όπως φαίνεται παραπάνω. Το κύριο παράθυρο επεξεργασίας μπορεί να χρησιμοποιηθεί για να δείτε και να τροποποιήσετε μέχρι και 16 διαφορετικά αρχεία πηγαίου κώδικα ταυτόχρονα. Κάθε αρχείο πηγαίου κώδικα που φορτώνεται στον επεξεργαστή κειμένου θα έχει τη δική του καρτέλα και στο επάνω μέρος της σελίδας σημαίνεται με το όνομα του αρχείου, όπως φαίνεται στην Εικόνα 3.2.. Η πλήρης ονομασία του αρχείου εμφανίζεται στη γραμμή τίτλου. Οι σελίδες με τον πηγαίο κώδικα που δεν έχει αποθηκευτεί στο δίσκο από προεπιλογή του προγράμματος ονομάζονται ως "Untitled #", όπου "#" είναι ένας αριθμός που παράγεται αυτόματα. Ο χρήστης μπορεί να κάνει εναλλαγές μεταξύ των αρχείων πηγαίου κώδικα, απλά κάνοντας κλικ στην καρτέλα ενός αρχείου ή πατώντας Ctrl + Tab ή Ctrl + Shift + Tab, ενώ το κύριο παράθυρο επεξεργασίας είναι ενεργό.



Εικόνα 3.2.

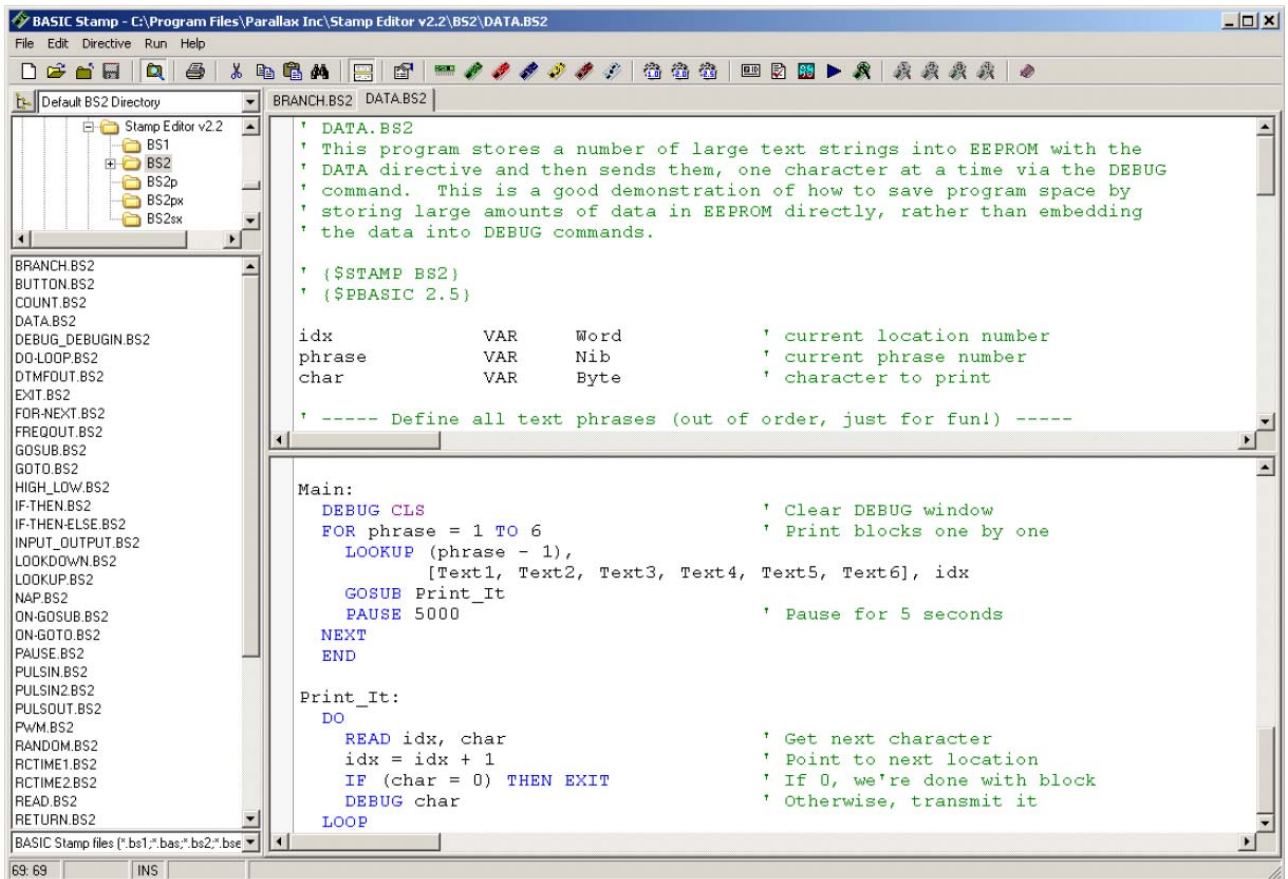
Η κατάσταση του ενεργού κώδικα υποδεικνύεται στη γραμμή κατάστασης κάτω από το κύριο παράθυρο επεξεργασίας και τον πίνακα εξερεύνησης (Εικόνα 3.3.). Η γραμμή κατάστασης περιέχει πληροφορίες όπως τη θέση του κέρσορα, το αρχείο αποθήκευσης, την κατάσταση αποστολής και λήψης δεδομένων, ειδοποιήσεις για συντακτικά σφάλματα και άλλα μηνύματα.



Εικόνα 3.3.

Κάθε παράθυρο του Editor μπορεί μεμονωμένα να χωριστεί σε δύο όψεις του ίδιου κώδικα. Αυτό μπορεί να γίνει μέσω της επιλογής Split (Εικόνα 3.4.) στη γραμμή εργαλείων ή πατώντας Ctrl-L, ή κάνοντας κλικ και σύροντας το επάνω ή το κάτω περίγραμμα του παραθύρου με το ποντίκι. Μόλις χωρίσει το παράθυρο σε δύο μέρη, επιτρέπεται στο χρήστη να βλέπει και να επεξεργάζεται διαφορετικές περιοχές του ίδιου κώδικα. Αυτό μπορεί να είναι βολικό όταν χρειάζεται να κρατήσει δεδομένα όπως δηλώσεις μεταβλητών ή μια

συγκεκριμένη ρουτίνα στην οθόνη και παράλληλα να τροποποιεί ένα άλλο τμήμα του κώδικα. Επισημαίνεται ότι η λειτουργία Split και η συντόμευση Ctrl + L λειτουργούν και αντίστροφα για να κλείσουν τα δύο παράθυρα σε ένα.



Εικόνα 3.4.

Λειτουργία Split. Εμφάνιση δύο τμημάτων του ίδιου προγράμματος.

Στο παράθυρο επεξεργασίας, ο πηγαίος κώδικας του BASIC Stamp εμφανίζεται με επισημάνσεις σύνταξης. Επισήμανση σύνταξης εφαρμόζεται με καθορισμένα χρώματα και χαρακτήρες (άνω, κάτω, κεφαλαία) για να παρακρατούνται οι λέξεις των εντολών και υποδείξεων στην γλώσσα προγραμματισμού Pbasic. Αυτό συμβαίνει αυτόματα καθώς πληκτρολογείτε. Ο πίνακας 3.1 δείχνει την προεπιλεγμένη αντιστοιχία χρωμάτων για κάθε στοιχείο που χρησιμοποιείται στην σύνταξη του προγράμματος. Ο πηγαίος κώδικας μπορεί να τυπωθεί σε χαρτί με τις δραστικές επισήμανση σύνταξης και τα χρώματά τους αν χρησιμοποιείτε έγχρωμο εκτυπωτή.

Οι περισσότερες λειτουργίες του προγράμματος του επεξεργαστή κειμένου είναι ίδιες ή παρόμοιες με αυτές των Windows. Ο χρήστης μπορεί να κινηθεί μέσα στα μενού του Editor και να επεξεργαστεί το πρόγραμμα με λειτουργίες Copy/ Cut/ Paste, Find, Replace και άλλες με συντομεύσεις ή με δεξί πάτημα στο ποντίκι.

Κατά τον προγραμματισμό ο χρήστης πρέπει να γνωρίζει ότι υπάρχει μία λίστα λέξεων οι οποίες είναι δεσμευμένες από το πρόγραμμα. Οι λέξεις αυτές αντιστοιχούν στις εντολές που γνωρίζει ο BASIC Stamp και τις ειδικές λειτουργίες του. Η λίστα αυτή βρίσκεται στο Παράρτημα Β του εγχειριδίου λειτουργίας του BASIC Stamp. [2]

Στοιχείο σύνταξης	Χρώμα κειμένου	Τύπος χαρακτήρων
Εντολή	Μπλε	Κεφαλαία
Σχόλιο	Πράσινο	Χωρίς Αλλαγή
Σταθερά - Δυαδικού	Προκαθορισμένο	Χωρίς Αλλαγή
Σταθερά - Δεκαδικού	Προκαθορισμένο	Χωρίς Αλλαγή
Σταθερά - Δεκαεξαδικού	Προκαθορισμένο	Χωρίς Αλλαγή
Σταθερά - Προκαθορισμένου	Μωβ	Κεφαλαία
Σταθερά – Αλληλουχίας εντολών	Κόκκινο	Χωρίς Αλλαγή
Μαθηματικές Συναρτήσεις	Προκαθορισμένο	Κεφαλαία
Δηλώσεις	Προκαθορισμένο	Κεφαλαία
Υπόθεση, Μεταγλώττιση υποθέσεων	Γκρι	Κεφαλαία
Κατευθυντήρια Οδηγία του επεξεργαστή κειμένου	Γαλαζοπράσινο	Κεφαλαία
Κατευθυντήρια Οδηγία του επεξεργαστή	Γαλαζοπράσινο	Κεφαλαία
Είσοδοι / Έξοδοι	Γαλάζιο	Κεφαλαία
Επιλογές	Άσπρο ή Γαλάζιο	Χωρίς Αλλαγή
Αποτελέσματα λειτουργίας Εύρεσης	Γκριζο ή Μαύρο	Χωρίς Αλλαγή
Σταθερά Τροποποιητή	Προκαθορισμένο	Κεφαλαία
Σταθερά Προκαθορισμένη	Μωβ	Κεφαλαία
Σταθερά Τύπου	Προκαθορισμένο	Κεφαλαιοποίηση

Πίνακας 3.1.

Αντιστοιχία χρωμάτων με τις λέξεις που χρησιμοποιούνται στο πρόγραμμα ανάλογα την χρήση τους

Τα στοιχεία που αναφέρθηκαν παραπάνω είναι βασικές οδηγίες για την περιήγηση του προγράμματος και την κατανόηση χρήσης του για το ξεκίνημα της ενασχόλησης με αυτό. Στο εγχειρίδιο χρήσης του μικροελεγκτή BASIC Stamp υπάρχουν όλες οι οδηγίες και η ανάλυση του προγράμματος σύνταξης. Περιλαμβάνονται οι εντολές και ο τρόπος χρήσης τους, καθώς και παραδείγματα.

3.3. Προεργασία πριν το Ξεκίνημα του Προγραμματισμού

Για να ξεκινήσει η διαδικασία του προγραμματισμού πρέπει να γίνουν πρώτα μερικές ενέργειες υπολογισμοί και πειραματισμοί. Αυτές είναι:

- Να βρεθούν τα όρια λειτουργίας των σερβομοτέρ ώστε οι τιμές που θα δίνουν την κίνηση από το πρόγραμμα να μην υπερβαίνουν τα όρια των σερβομοτέρ και τα όρια για κίνηση των εξαρτημάτων της κατασκευής.
- Να βρεθεί ο τρόπος και η σειρά που θα πρέπει να κινούνται τα πόδια ώστε να περπατάει και να στρίβει το ρομποτικό.
- Να υπολογιστούν και να δοκιμαστούν οι τιμές και που τις λαμβάνει ο μικροελεγκτής από τον αισθητήρα απόστασης.

3.3.1. Ρύθμιση των ποδιών και ρύθμιση του εύρους κίνησής τους

Το ρομποτικό αυτό είναι μια κατασκευή που πρέπει να κινείται με ακρίβεια. Μετά την ρύθμιση των ποδιών με μηχανικό τρόπο μπορεί να μην είναι δυνατόν να ευθυγραμμιστούν όλα τα πόδια. Τα σφάλματα και οι διαφορές θα οδηγήσουν το ρομποτικό να περπατάει στραβά και να χάνει το στόχο του προγράμματός του. Για παράδειγμα μπορεί να στρίβει σταδιακά δεξιά ή αριστερά όταν περπατάει ευθεία ή να μην πατάνε το ίδιο όλα τα πόδια στο έδαφος.

Η κίνηση κάθε σερβομοτέρ και ποδιού σαν σύνολο εξαρτάται τόσο από την μηχανική κατασκευή όσο από την ηλεκτρονική. Κατά την κατασκευή και συναρμολόγηση των εξαρτημάτων υπάρχουν διάφορες μικροδιαφορές με αποτέλεσμα η τελική κίνηση να διαφέρει. Αυτές οι διαφορές έχουν να κάνουν περισσότερο με τα σερβομοτέρ που περιέχουν πολλά εξαρτήματα, όπως γρανάζια και την τοποθέτησή τους καθώς και τον αισθητήρα θέσης που οποιαδήποτε διαφορά προκαλεί το μεγαλύτερο σφάλμα.

Για να βρούμε τα όρια και το κέντρο των σερβομοτέρ στην κατασκευή χρειάζεται ένα πρόγραμμα που να κουνάει το κάθε σερβομοτέρ ξεχωριστά και να τελειώνει την κίνηση στο κέντρο του. Οι τιμές αυτές είναι πολύ σημαντικές και θα χρησιμοποιούνται συνέχεια σε κάθε μορφή προγράμματος για την κίνηση των ποδιών.

Η ρύθμιση ξεκινάει πρώτα μηχανικά τοποθετώντας πάνω στον άξονα κίνησης του σερβομοτέρ το άκρο που τον ενώνει με το αντίστοιχο άκρο του ποδιού. Για να βρεθεί το κέντρο της κίνησης Γυρνώντας με αργή κίνηση τον άξονα δεξιόστροφα και μετά

αριστερόστροφα βρίσκουμε τα όρια του κινητήρα και τέλος γυρνάμε τον άξονα όσο πιο κοντά στο κέντρο της κίνησής του. Μετά τοποθετούνται πάνω στο πόδι και τα σερβομοτέρ που κάνουν την οριζόντια κίνηση έτσι ώστε να γέρνουν το πόδι όσο πιο κάθετα γίνεται με το ρομποτικό. Με παρόμοιο τρόπο τοποθετείται και το σερβομοτέρ που κάνει την κάθετη κίνηση. Αφού είναι όσο πιο κοντά στο κέντρο της κίνησής του τοποθετείται και το κομμάτι του ποδιού έτσι ώστε να το φέρνει και αυτό στο κέντρο την δικιάς του κίνησης.

Στη συνέχεια για την ακριβή ρύθμιση των ποδιών ψάχνουμε πειραματικά μέσω προγραμματισμού τις επιθυμητές τιμές. Από το εγχειρίδιο λειτουργίας γνωρίζουμε ότι ο PSC φέρνει το κάθε σερβομοτέρ στο κέντρο του με την τιμή 750 ή με παλμό 1.5ms. Στις παρακάτω εντολές του προγράμματος:

```
' Center1    CON    750    ' Leg1 horizontal servo
' Center2    CON    750    ' Leg2 horizontal servo
' Center3    CON    750    ' Leg3 horizontal servo
' Center4    CON    750    ' Leg4 horizontal servo
```

Αλλάζοντας τις τιμές σε λίγο μεγαλύτερες ή λίγο μικρότερες μετακινείται το άκρο ανάλογα λίγο μπροστά ή λίγο πίσω. Η κίνηση αυτή είναι αντίστροφη στην δεξιά με την αριστερή πλευρά λόγω της τοποθέτησης των σερβομοτέρ. Κάνοντας τις αλλαγές μεταφέρουμε το πρόγραμμα κάθε φορά στον μικροελεγκτή, το τρέχουμε και οπτικά βλέπουμε την θέση του ποδιού αν είναι σωστή.

```
'    **Right Side**
'  +750-   +750-
' F   |     |
' O  _|_   _|_
' R  / 1 \___/ 2 | R
' W |         | E
' A |   ___   | A
' R  \_3_ /  \_4_ | R
' D   |     |
'     |     |
'  -750+   -750+
'    **Left Side**
```

Αντίστοιχα πρέπει να γίνει για την κάθετη κίνηση του ποδιού. Εδώ η διαδικασία πρέπει να προχωρήσει με περισσότερα βήματα και περισσότερο πειραματισμό για να αποφευχθούν υλικές ζημιές. Μέσα στο πρόγραμμα υπάρχουν οι εντολές για την κίνηση των ποδιών πάνω και κάτω:

'-----[Vertical Leg Constants]-----

RaiseRightF	CON	630
LowerRightF	CON	1100
RaiseRightR	CON	1150
LowerRightR	CON	700
RaiseLeftF	CON	430
LowerLeftF	CON	1000
RaiseLeftR	CON	460
LowerLeftR	CON	1030

Πειραματικά για να βρεθούν τα όρια πόσο εύρος θα έχει στην κίνηση αυτή το πόδι. Ξεκινάμε από νούμερα κοντά στο κέντρο της κίνησης 750 και κάθε φορά ανοίγουμε την διαφορά τους μέχρι να βρεθούν οι τιμές που δεν κινδυνεύει το πόδι να ακουμπήσει στο κατέβασμα σε κάποιο σταθερό σημείο της κατασκευής. Στο ανέβασμα πρέπει να σταματάει χωρίς να κάνει περιττή κίνηση, όπως να γυρίζει περισσότερες μοίρες και να κατεβάζει ξανά το πόδι.

Οι τιμές που καταλήγουμε να επιλέξουμε τις χρησιμοποιούμε σαν σταθερές σε όλο το πρόγραμμα. Αν για κάποιο λόγο χρειαστεί να αλλαχτεί κάποιο σερβομοτέρ η διαδικασία αυτή πρέπει να επαναληφθεί.

3.3.2. Προγραμματισμός περπατήματος

Για να ξεκινήσει ο προγραμματισμός του περπατήματος πρέπει πρώτα να βρεθεί ο τρόπος και η σειρά που θα κινηθούν τα πόδια για να έχει ως αποτέλεσμα την κίνηση του ρομποτικού. Η διαδικασία αυτή περιλαμβάνει διάφορους τρόπους περπατήματος που θα χρειαστούν για την επιτυχία την πτυχιακής αυτής άσκησης. Ο κάθε τρόπος περπατήματος θα πρέπει να γίνεται μέσα σε υπορουτίνα για να είναι δυνατή η επιλογή της καθεμίας αναλόγως το τι θέλουμε να κάνει το ρομποτικό. Ξεκινάμε από το αποτέλεσμα. Το ρομποτικό θέλουμε να αποφεύγει εμπόδια με τη χρήση Fuzzy Logic ελεγκτή που ανάλογα με την απόσταση του εμποδίου θα αποφασίζει για την ταχύτητα και την κίνηση που θα κάνει για να το αποφύγει. Άρα χρειαζόμαστε σίγουρα πρόγραμμα που να κάνει το ρομποτικό να περπατάει:

- ευθεία μπροστά,
- ευθεία πίσω,
- να στρίβει με κίνηση μπροστά ή πίσω και
- να μπορούμε να ρυθμίσουμε την ταχύτητα των κινήσεων.

3.3.2.1 Περπάτημα Ευθεία

Για να πετύχουμε το περπάτημα ευθεία μπροστά πρέπει να γίνει μια ακολουθία κινήσεων που θα φέρνει κάθε φορά τα πόδια σε θέση ώστε να τα μπροστινά να τραβάνε το ρομποτικό και τα πίσω να το σπρώχνουν. Οι κινήσεις αυτές πρέπει να υπολογιστούν με βάση τον κύκλο που θα τρέχει τις εντολές το πρόγραμμα κατά την εκτέλεση της ρουτίνας του περπατήματος. Μετά από πειραματισμούς και διάφορα διαγράμματα ροής για το πώς θα μπορεί να γίνει ο προγραμματισμός πιο εύκολος και το πώς θα σπαταλάει λιγότερη μνήμη από τον ελεγκτή βρέθηκε ένα μοντέλο που στην συγκεκριμένη εφαρμογή είχε το καλύτερο αποτέλεσμα. Η κίνηση των ποδιών στον οριζόντιο άξονα χωρίστηκε σε δύο μέρη, από το κέντρο και προς τα μπροστά και από το κέντρο και προς τα πίσω. Χρησιμοποιήθηκε μια σταθερά που αντί να ορίζεται το σημείο που θα κινηθεί το κάθε σερβομοτέρ προς τα μπροστά ή προς τα πίσω, αλλά προστίθεται ή αφαιρείται από το κέντρο που έχουμε βρει και κινεί τα πόδια μπροστά ή πίσω. Η σταθερά αυτή χρησιμοποιείται και για τις κινήσεις του περπατήματος. Ακόμα για να γίνονται οι κινήσεις με μεγαλύτερη σταθερότητα, χωρίστηκαν σε κινήσεις που θα γίνονται ταυτόχρονα σε δύο πόδια, ζευγάρια διαγώνια. Με τον τρόπο αυτό ήταν εύκολο να οριστούν οι βασικές κινήσεις και να είναι εκμεταλλεύσιμες σε κάθε τρόπο περπατήματος. Ξεκινώντας από την θέση ηρεμίας, δηλαδή όλα τα πόδια στο κέντρο και κατεβασμένα, κάποια από τα πόδια δεν θα κάνουν κάποια κίνηση αφού θα βρίσκονται ήδη στη θέση αυτή. Οι κινήσεις είναι:

- Ξεκινάει σηκώνοντας το 1ο πόδι (μπροστά δεξιά) και ταυτόχρονα το 4ο πόδι (πίσω αριστερά).
- Κίνηση του 1ου ποδιού που είναι σηκωμένο προς τα μπροστά και το 2ο πόδι (πίσω δεξιά) που είναι κατεβασμένο προς τα πίσω, άρα σπρώχνει το ρομποτικό μπροστά.
- Κίνηση του 3ου ποδιού (μπροστά αριστερά) στο κέντρο, αλλά στην πρώτη εκτέλεση του προγράμματος βρίσκεται ήδη και δεν κάνει καμιά κίνηση. Μετά τον πρώτο κύκλο που θα βρίσκεται μπροστά και κατεβασμένο, θα κινηθεί προς τα πίσω και θα τραβήξει το ρομποτικό μπροστά. Κίνηση του 4ου ποδιού από πίσω και σηκωμένο στο κέντρο. Εδώ και πάλι την πρώτη φορά δεν θα κάνει κίνηση αφού βρίσκεται ήδη στο κέντρο.
- Κατεβάζει το 1ο πόδι και το 4ο πόδι.
- Ανεβάζει το 2ο πόδι και το 3ο
- Κίνηση το 1ο πόδι που είναι κατεβασμένο στο κέντρο, άρα τραβάει το ρομποτικό και το 2ο πόδι που είναι σηκωμένο στο κέντρο και το γυρίζει στην αρχική του θέση.
- Κινεί το 3ο πόδι που είναι σηκωμένο προς τα μπροστά και το 4ο πόδι που είναι κατεβασμένο προς τα πίσω και σπρώχνει το ρομποτικό προς τα μπροστά.
- Κατεβάζει το 2ο πόδι και το 3ο. Εδώ τελειώνει και ο κύκλος της ρουτίνας για το περπάτημα μπροστά. Η διαδικασία αυτή για να κινήσει το ρομποτικό θα επαναλαμβάνεται συνεχόμενα.

3.3.2.2. Περπάτημα Ευθεία Πίσω

Το περπάτημα ευθεία προς τα πίσω γίνεται με τον ίδιο τρόπο με το ευθεία μπροστά

μόνο που αντιστρέφονται οι κινήσεις.

- Ξεκινάει σηκώνοντας το 1ο πόδι (μπροστά δεξιά) και ταυτόχρονα το 4ο πόδι (πίσω αριστερά)
- Κίνηση του 1ου ποδιού στο κέντρο, αλλά στην πρώτη εκτέλεση του προγράμματος βρίσκεται ήδη στο κέντρο και δεν κάνει καμιά κίνηση. Μετά τον πρώτο κύκλο που θα βρίσκεται μπροστά και ανεβασμένο. Κίνηση του 2ου ποδιού (πίσω δεξιά) στο κέντρο, ενώ ήταν πίσω και κατεβασμένο, οπότε τραβάει το ρομποτικό προς τα πίσω. Εδώ και πάλι την πρώτη φορά δεν θα κάνει κίνηση αφού θα βρίσκεται ήδη στο κέντρο.
- Κίνηση του 3ου ποδιού (μπροστά αριστερά) που είναι κατεβασμένο προς τα μπροστά, άρα σπρώχνει το ρομποτικό προς τα πίσω και το 4ο πόδι (πίσω δεξιά) που είναι ανεβασμένο κίνηση προς τα πίσω.
- Κίνηση το 3ο πόδι μπροστά άρα σπρώχνει το ρομποτικό προς τα πίσω και το 4ο πόδι που είναι σηκωμένο στο κέντρο και το γυρίζει στην αρχική του θέση προς τα πίσω.
- Κατεβάζει το 1ο πόδι και το 4ο πόδι.
- Ανεβάζει το 2ο πόδι και το 3ο πόδι.
- Κίνηση το 1ο πόδι προς τα μπροστά ενώ είναι κατεβασμένο, άρα σπρώχνει το ρομποτικό προς τα πίσω και το 2ο που είναι σηκωμένο προς τα πίσω.
- Κίνηση το 3ο πόδι που είναι σηκωμένο στο κέντρο και το 4ο πόδι που είναι κατεβασμένο στο κέντρο, άρα τραβάει το ρομποτικό προς τα πίσω.
- Κατεβάζει το 2ο πόδι και το 3ο. Εδώ τελειώνει και ο κύκλος της ρουτίνας για το περπάτημα προς τα πίσω. Η διαδικασία αυτή για να κινήσει το ρομποτικό θα επαναλαμβάνεται συνεχόμενα.

3.3.2.3. Τρόπος περπατήματος με στροφή

Όπως αναφέρθηκε και παραπάνω ο τρόπος προγραμματισμού του περπατήματος έχει γίνει έτσι ώστε να μπορεί να γίνεται εύκολα και με λιγότερη χρήση μνήμης του ελεγκτή. Το περπάτημα με στροφή δεν είναι τίποτα παραπάνω από το περπάτημα σε ευθεία, μόνο που σε κάθε κίνηση του οριζόντιου άξονα του κάθε ποδιού προστίθεται ή αφαιρείται η σταθερά κίνησης. Με τον τρόπο αυτό προστίθεται ή αφαιρείται απόσταση στην κίνηση του κάθε ποδιού. Για να στρίψει το ρομποτικό την στιγμή που περπατάει προς τη μία κατεύθυνση πρέπει να αφαιρεθεί ένα μέρος κίνησης από την πλευρά που θέλουμε να στρίψει και να προστεθεί ένα μέρος κίνησης από την αντίθετη.

3.3.2.4. Ταχύτητα κίνησης και συνδιασμός κινήσεων

Η ταχύτητα κίνησης του ρομποτικού είναι ανάλογη της ταχύτητας κίνησης των σερβομοτέρ. Από το εγχειρίδιο λειτουργίας του ελεγκτή PCS των σερβομοτέρ γνωρίζουμε ότι υπάρχει εντολή που είναι υπεύθυνη για την ταχύτητα της κίνησης αλλά και αναγκαία να

υπάρχει μέσα στο πρόγραμμα. Χρησιμοποιώντας μία σταθερά αντί για τιμή που θα ορίζει την ταχύτητα, μπορούμε να αλλάζουμε και να επιλέγουμε την τιμή της σταθεράς με οποιοδήποτε τρόπο και ταυτόχρονα να αλλάζει παντού η ταχύτητα κίνηση και να υπάρχει ο αντίστοιχος εύκολος και γρήγορος έλεγχος.

3.3.2.5. Ανάλυση Λογικής Προγραμματισμού του κώδικα Περπατήματος

Με βάση όλα τα προηγούμενα, ξεκινώντας τον προγραμματισμό έπρεπε να βρεθεί ένας τρόπος που να χρησιμοποιεί όσο το δυνατόν λιγότερη μνήμη. Η πρώτη και πιο απλή σκέψη, να γράψουμε κώδικα που θα περιελάμβανε την κάθε δυνατή κίνηση ξεχωριστά θα δημιουργούσε πολλά προβλήματα. Όπως ήδη αναφέρθηκε θα χρειαζόταν πολύ χώρο στη μνήμη, οι οποιοσδήποτε αλλαγές για πειραματισμό και ρύθμιση των κινήσεων των ποδιών θα ήταν υπερβολικά χρονοβόρες και τέλος θα περιόριζε σημαντικά την μετέπειτα εξέλιξη του ρομποτικού τόσο λόγω έλλειψης μνήμης όσο του περιορισμού σε σταθερές ταχύτητες και κινήσεις των σερβομοτέρ.

Μετά από διάφορους πειραματισμούς καταλήξαμε τελικά σε μία πολύπλοκη λογική που θα επιλέγει και θα συνθέτει κάθε φορά τον κωδικό που θα αντιστοιχεί σε κάθε κίνηση. Οι κινήσεις του κάθε κύκλου βημάτων αποτελούνται από:

- μία ακολουθία κωδικών που αναγνωρίζει ο PSC,
- την φορά, μπροστά, πίσω ή επιτόπου κίνηση,
- το εύρος κίνησης και
- την ταχύτητα της ράμπας ή κίνησης του κάθε σερβομοτέρ.

Οι επιλογές των κινήσεων όπως θα τις συναντάμε μέσα στον κώδικα του προγράμματος φαίνονται στον Πίνακα 3.2.:

Κίνηση	Κωδικός Κίνησης	Ένδειξη LED Display
Home	\$00	0
Spin Left	\$01	1
Spin Right	\$02	2
Fast Forward	\$10	3
Fast Forward Left	\$11	4
Fast Forward Right	\$12	5
Forward	\$20	6

Forward Left	\$21	7
Forward Right	\$22	8
Backward	\$30	9
Backward Left	\$31	A
Backward Right	\$32	b
Fast Backward	\$40	C
Fast Backward Left	\$41	d
Fast Backward Right	\$42	E

Πίνακας 3.2.

Ο κάθε τρόπος περπατήματος θα γίνει με μία διαδικασία που θα ορίσει στην μνήμη του επεξεργαστή μία σταθερά την οποία θα καλούμε για κάθε τρόπο κίνησης, μπροστά, πίσω, δεξιά αριστερά και adjust. Η κάθε σταθερά θα αντιστοιχεί σε μία σειρά θέσεων την μνήμης στις οποίες αποθηκεύονται οι τιμές της κάθε κίνησης με την σειρά που πρέπει να γίνουν για να ολοκληρωθεί ένας κύκλος βήματος όπως αναφέρθηκαν παραπάνω στους τρόπους περπατήματος. Η σταθερά αυτή που θα επιλέγεται κάθε φορά αποθηκεύεται σε μια μεταβλητή για να χρησιμοποιηθεί αργότερα. Στη συνέχεια επιλέγεται η ταχύτητα των ραμπών που χρειάζεται για τις κάθετες κινήσεις και τέλος η ταχύτητα των ραμπών για την οριζόντια κίνηση. Από την κάθε επιλογή η τιμή που αποθηκεύεται σε κάθε μεταβλητής αντίστοιχα χρησιμοποιείται για να συνθέσουν όλες μαζί την επόμενη κίνηση που θα σταλεί στον PSC.

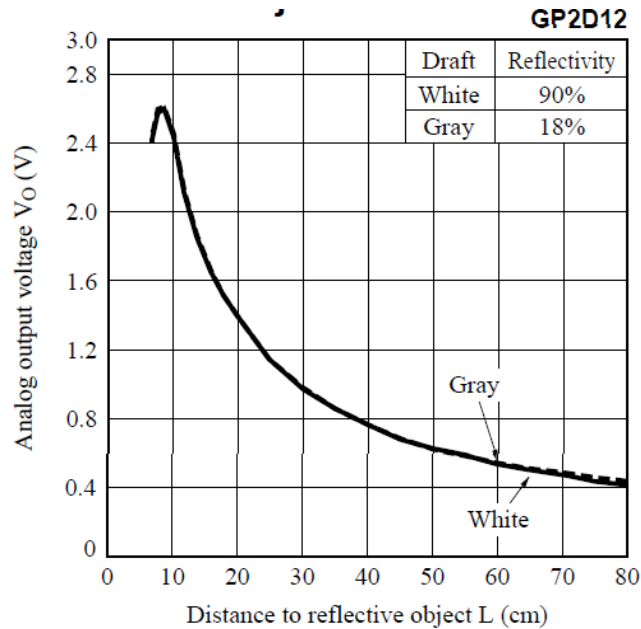
3.3.3. Υπολογισμός της μετρούμενης απόστασης από τον Sharp IR

Τα προϊόντα της SHARP παρόλο που είναι πολύ γνωστά και χρησιμοποιούνται σε πολλές εφαρμογές έχουν ένα βασικό μειονέκτημα. Το μειονέκτημα αυτών των αισθητήρων είναι η μη γραμμικότητα στις αντιδράσεις τους (Γραφική Παράσταση 3.1.). Για να πάρει ο χρήστης το αποτέλεσμα της μέτρησης σε μονάδες μέτρησης απόστασης (εκατοστά, κλπ.) πρέπει να βρει μια συνάρτηση που να μετατρέπει την τιμή της τάσης που παίρνει στην έξοδο του αισθητήρα σε φυσική μονάδα μέτρησης. [9]

Μία προσέγγιση που λειτουργεί καλά είναι με τη χρήση πολύπλοκων μαθηματικών προγραμμάτων για να παραχθεί η καμπύλη. Οι λειτουργίες αυτών των προγραμμάτων συνήθως έχουν την απαίτηση χρήσης δεκαδικών ψηφίων στους αριθμούς, κάτι που πολλές φορές δεν υποστηρίζουν οι επεξεργαστές.

Ένας καλός τρόπος προσέγγισης είναι με τη χρήση ευθύγραμμων τμημάτων για να μετατραπεί η τάση εξόδου σε ένα εύρος τιμών. Με αυτό το τρόπο σπάει η καμπύλη απόκρισης σε πολλά ευθύγραμμα τμήματα και χρησιμοποιείται διαφορετική συνάρτηση για το καθένα. Τα ευθύγραμμα τμήματα τώρα μπορούν να υπολογιστούν με μια καλή προσέγγιση

χρησιμοποιώντας απλά μαθηματικά. Το μειονέκτημα είναι ότι χρειάζεται αρκετός χώρος για τον κώδικα του προγράμματος.



Γραφική Παράσταση 3.1.).

Γραφική Παράσταση Απόστασης – Τάσης αισθητήρα απόστασης Sharp GP2D12

Το ιδανικό είναι να υπάρχει μία μόνο συνάρτηση που να δουλεύει καλά με ακαριαίους αριθμούς.

Ευτυχώς υπάρχουν μερικοί απλοί υπολογισμοί που μπορούν να δώσουν γραμμική μορφή στην καμπύλη απόκρισης των αισθητήρων όπως αυτός της Sharp.

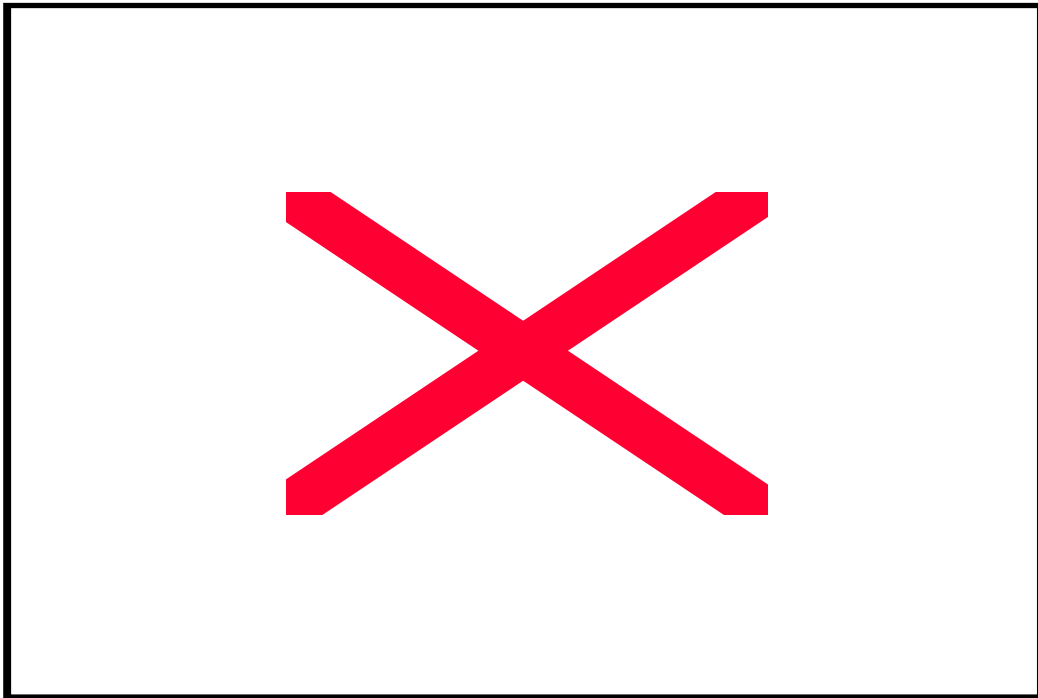
Διαδικασία μετατροπής σε γραμμική μορφή

Σύμφωνα με το τεχνικό βιβλίο του αισθητήρα Sharp GP2D12, η γραφική παράσταση της ακόλουθης εξίσωσης περιγράφεται:

$$V = 1 / (R + k)$$

Όπου V είναι η τάση, R είναι το εύρος και k μία σταθερά. Η εξίσωση δίνει μία πολύ ίσια γραμμή. Η διαίρεση λειτουργεί σαν διαδικασία ευθυγράμμισης που μετατρέπει την καμπύλη σε ευθεία. Αυτός ο τρόπος αποτελεί μια γενική διαδικασία για όλους τους αισθητήρες απόστασης της Sharp.

Η σταθερά k στην διαδικασία ευθυγράμμισης εξαρτάται από τον τύπο του αισθητήρα και τα δεδομένα βαθμονόμησης. Η εύρεση της σταθεράς k αποτελεί το πρώτο βήμα για να πάρουμε μία καλή συνάρτηση. Στο παρακάτω διάγραμμα (Γραφική Παράσταση 3.2.) βλέπουμε πως οι τιμές του GP2D12 ευθυγραμμίζονται όταν η σταθερά k παίρνει την τιμή 4.



Γραφική Παράσταση 3.2.

Το επόμενο βήμα είναι να βρεθεί μια προσέγγιση που να συσχετίζει την τάση με την συνάρτηση ευθυγράμμισης. Για αυτό πρέπει να βρεθούν οι δύο κατάλληλες σταθερές m και b που να ορίζουν την συνάρτηση της ευθείας:

$$y = m * x + b$$

Σε αυτή της περίπτωση το y είναι ίσο με το εύρος της ευθυγράμμισης. Αντικαθιστώντας την εξίσωση ευθυγράμμισης y με $m * x + b$ και την V με $1 / (R + k)$ για τον x άξονα έχουμε:

$$1 / (R + k) = m * V + b$$

για να βρεθεί το εύρος R

$$R = (1 / (m * V + b)) - k$$

$$R = (m' / (V + b')) - k$$

όπου $m' = 1/m$ και $b' = b/m$. Η τελευταία εξίσωση δουλεύει σε γλώσσες προγραμματισμού με ακέραιους αριθμούς.

Για να βρεθούν οι σταθερές χρειάζεται πρώτα να πάρουμε κάποια δεδομένα βαθμονόμησης. Αυτό γίνεται πειραματικά ή με μετρήσεις σημείων από την καμπύλη τάσης – εύρους του αισθητήρα. Σε ένα πίνακα τοποθετούμε τις τιμές τάσης και εύρους. Σε ένα δεύτερο πίνακα τοποθετούμε τις τιμές της τάσης σε μονάδες του ελεγκτή και ευθυγραμμισμένου εύρους. Με λίγο πειραματισμό βρίσκουμε την σταθερά k που μας δίνει το ευθύγραμμο διάγραμμα και τις σταθερές m και b .

Για τον αισθητήρα GP2D12 που χρησιμοποιείται, ισχύουν τα ακόλουθα:

$$R = (6787 / (V - 3)) - 4$$

όπου $m'=6787$, $b'=3$ και $k=4$. Ο χρήστης πρέπει να ελέγχει την είσοδο της τάσης ώστε να αποφευχθεί διαίρεση με το μηδέν και να ελέγχει η τάση V να είναι μεγαλύτερη του b . Εδώ η

τάση είναι μεγαλύτερη του 3. Με 10-bit ακέραια ψηφία οι μετρήσεις της τάσης από τον GP2D12 είναι τυπικά παραπάνω από 80 όταν κάποιο εμπόδιο βρίσκεται μέσα στο εύρος. Αν δεν υπάρχει εμπόδιο οι μετρήσεις μπορεί να είναι χαμηλές έως μηδέν. Λόγω της χρήσης ακέραιων αριθμών, οι μονάδες που παίρνουμε σαν αποτέλεσμα “εκατοστά, cm” δεν είναι πραγματικά, αλλά προσεγγίζουν την απόσταση με μια μονάδα μέτρησης πιο εύκολη στη χρήση από αυτό που διαβάζουμε σαν μονάδα από τον αναλογικό σε ψηφιακό μετατροπέα.

Αυτή η στρογγυλοποίηση δουλεύει καλά σε επεξεργαστές που χρησιμοποιούν ακαιραίους αριθμούς στις μαθηματικές πράξεις. Σε επεξεργαστές που μπορούν να χρησιμοποιούν δεκαδικά ψηφία στις πράξεις αυτή η τεχνική δουλεύει καλύτερα. Η επιτυχία στην υλοποίηση εξαρτάται από την ακρίβεια των δεδομένων βαθμονόμησης και την καλή επιλογή της σταθεράς k.

3.4. Σύνδεση του BASIC Stamp με το MATLAB

Κατά την έρευνα για να επιτευχθεί σύνδεση μεταξύ του BASIC Stamp και του MATLAB συναντήθηκαν πολλές δυσκολίες. Αρχικά από την ιστοσελίδα του MATLAB δεν βρέθηκε κάποια έτοιμη βάση σύνδεσης που να υποστηρίζει την επικοινωνία και την επεξεργασία του κώδικα του BASIC Stamp κατευθείαν από τις γραμμές εντολών του MATLAB. [10] Αυτό που είναι εφικτό να γίνει είναι η σύνδεση από σειριακή θύρα. Με τον τρόπο αυτό χρησιμοποιείται ο BASIC Stamp σαν ο κεντρικός επεξεργαστής και ο H/Y που λειτουργεί το MATLAB σαν ξενιστής για την εκτέλεση κάποιων λειτουργιών. Έτσι μπορεί να επικοινωνεί με την γραμμή εντολών του MATLAB για να λαμβάνει και να δίνει ψηφιακά σήματα, παλμοσειρές, σήματα DTMF καθώς και να τα μετράει. [10]

Μετά από προσπάθειες που έγιναν για επιτευχθεί η σύνδεση από την USB θύρα της B.O.E. δεν ήταν δυνατόν να μπορεί να επέμβει η γραμμή εντολών του MATLAB και τα στοιχεία της εξόδου του FUZZY LOGIC ελεγκτή να μεταβάλουν κατευθείαν τα στοιχεία του κώδικα του προγράμματος στον BASIC Stamp ή έστω να του δώσει από την USB θύρα το αποτέλεσμα και να το δεχθεί ο κώδικας του BASIC Stamp σαν στοιχείο μεταβλητής για να συνεχίσει στην επόμενη κίνηση. Ο μόνος τρόπος που μπορούσε να γίνει ήταν να σταλεί η ανάδραση του Fuzzy Logic σε ένα I/O pin του BASIC Stamp και να το δεχθεί σαν μία τιμή μεταβλητής και ταυτόχρονα σαν την παράμετρο που θα αφήσει το πρόγραμμα να συνεχίσει να τρέχει για να στείλει με την σειρά του τα δεδομένα της επόμενης κίνησης στον PSC. Το αποτέλεσμα αυτό σε συνεργασία με τον επιτηρητή καθηγητή δεν κρίθηκε επιθυμητό καθώς δημιουργούσε επιπλέον πρόβλημα στην ταχύτητα και την ομαλότητα των κινήσεων. Επιπλέον το ίδιο αποτέλεσμα μπορούσε να γίνει μόνο με την χρήση του BASIC Stamp, κάτι που δίνει στο ρομποτικό την άνεση να κινείται αυτόνομα. Αυτός ήταν και ο κύριος λόγος που η εργασία συνεχίστηκε χρησιμοποιώντας έναν Fuzzy Logic ελεγκτή γραμμένο μέσα στον κώδικα του προγράμματος του BASIC Stamp.

Πιο αναλυτικά η έξοδος του MATLAB θα ήταν ένα νούμερο που θα καθόριζε την ταχύτητα της ράμπας κίνησης των σερβομοτέρ ή ο αριθμός της μεταβλητής της gaitCode που καθορίζει την επόμενη κίνηση.

Στην πρώτη περίπτωση το συνολικό αποτέλεσμα θα ήταν πιο σωστό καθώς στον προγραμματισμό του BASIC Stamp θα γινόντουσαν οι ίδιες επιλογές εκτός από την απόφαση για την ταχύτητα ράμπας που θα μπορούσε να πάρει όλες τις τιμές από 0 έως 63 που είναι οι διαφορετικές ταχύτητες ράμπας που υποστηρίζει ο PSC. Το θετικό θα ήταν ότι θα μπορούσε

να υπάρχει μεγάλη γραμμικότητα ανάμεσα στην αλλαγή της ταχύτητας από την μία κίνηση στην άλλη.

Στην δεύτερη περίπτωση η επιλογή της κίνησης από τον Fuzzy Logic ελεγκτή του MATLAB και η αποστολή των δεδομένων της στον BASIC Stamp για να γράψει την μεταβλητή της gaitCode είχε ακριβώς το ίδιο αποτέλεσμα με τον τρόπο που τελικά επιλέχθηκε και να γίνει χωρίς την χρήση του MATLAB. Μειονέκτημα ήταν και η ανάγκη να γραφτεί περισσότερος κώδικας στο πρόγραμμα του BASIC Stamp κάτι που θα χρησιμοποιούσε περισσότερη μνήμη και μελλοντικά θα περιόριζε την εξέλιξη του ρομποτικού.

Κατά τον πειραματισμό για την συνδυαστική λειτουργία του Fuzzy Logic ελεγκτή από το MATLAB και το BASIC Stamp ταυτόχρονα παρατηρήθηκε μεγάλη καθυστέρηση μεταξύ των κινήσεων κάτι που στο τελικό αποτέλεσμα δε έδειχνε ομαλότητα και δεν ήταν ωραίο. Στο τέλος του κύκλου του κάθε βήματος ο χρόνος για να σταλούν τα δεδομένα από τον BASIC Stamp στον H/Y δημιουργούσε την καθυστέρηση αυτή. Έγιναν διάφορες προσπάθειες και πειραματισμοί για να μειωθεί ο χρόνος του κύκλου του προγράμματος και να αυξηθεί η ταχύτητα εκτέλεσης του προγράμματος αλλά το αποτέλεσμα δεν ήταν ορατό παρόλο που ο χρόνος μειώθηκε. Αντιστρόφως ανάλογα ήταν για την αξιοπιστία των μετρήσεων τις απόστασης που παρατηρήθηκε σφάλμα στις μετρήσεις. Προσπάθειες έγιναν και με πειραματισμό στις επιλογές επικοινωνίας του BASIC Stamp και του H/Y αλλά χωρίς θετικό αποτέλεσμα.

Η θετική παρατήρηση που έγινε σε όλη την διαδικασία ήταν στην περίπτωση που το ρομποτικό θα είχε πολλούς αισθητήρες, η ανάλυση και απόφαση του Fuzzy Logic ελεγκτή γραμμένο στον κώδικα του BASIC Stamp θα απαιτούσε πολύ χώρο στην μνήμη του και θα περιοριζόταν. Επίσης η ανάλυση πολλών σημάτων εισόδου θα μπορούσε να γίνει πολύ πιο γρήγορα από έναν H/Y σε σχέση με τον BASIC Stamp, αλλά ταυτόχρονα θα συνεχίζει να υπάρχει το πρόβλημα της καθυστέρησης από την αποστολή δεδομένων προς τον H/Y.

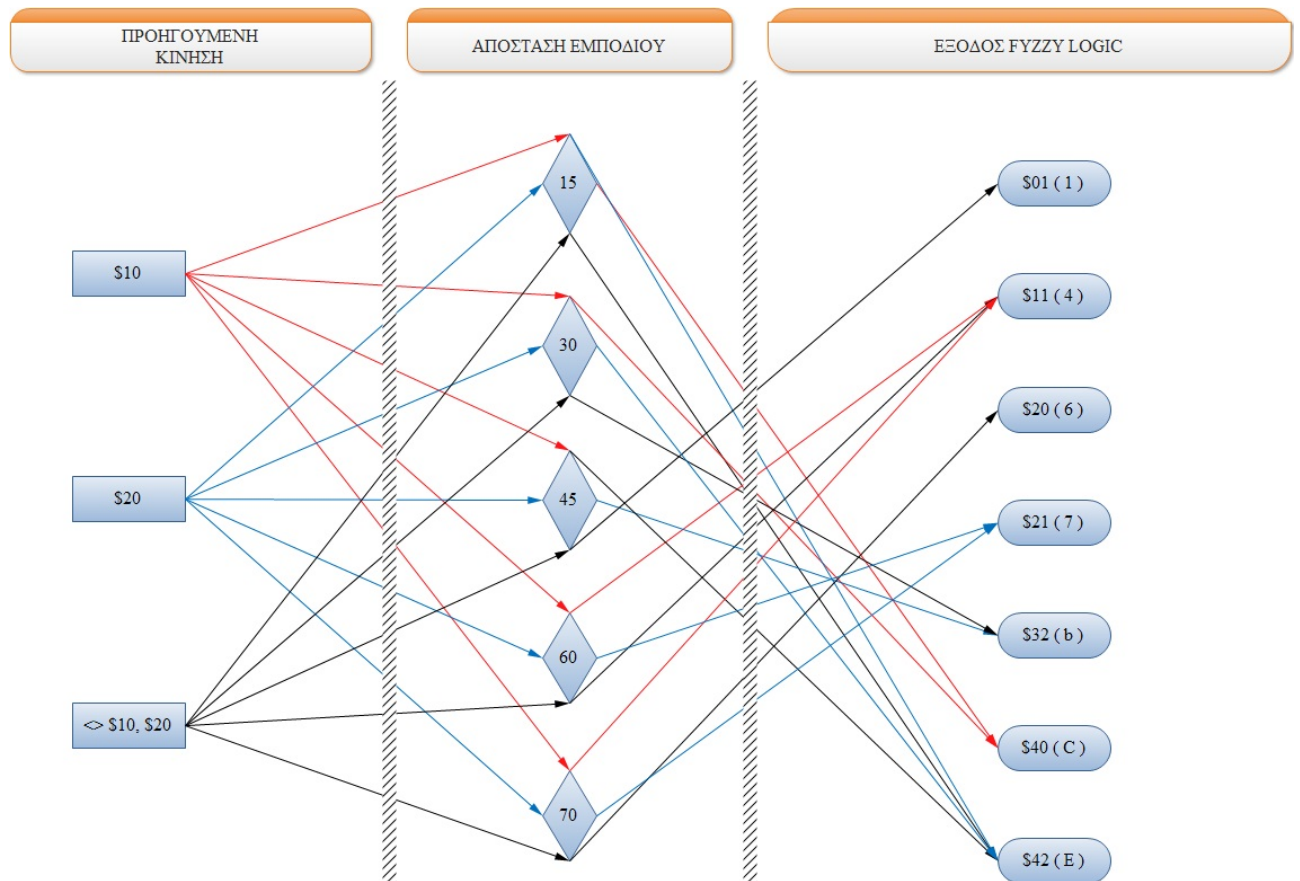
3.5. Ελεγκτής FUZZY LOGIC

Η λειτουργία του Fuzzy Logic αποτελεί μία προσέγγιση βασισμένη στο να υπολογίζει βαθμούς αλήθειας σε ερωτήματα παρά στην συνηθισμένη επιλογή του αληθείς – δεν αληθείς της Boolean λογικής στην οποία βασίζονται οι υπολογιστές. Η Fuzzy Logic είναι μία λογική που προσπαθεί να προσεγγίσει την πραγματικότητα και τα φυσικά μεγέθη. Τα φυσικά μεγέθη όπως και όλα τα πράγματα και δραστηριότητες δεν είναι εύκολο να μεταφραστούν στην γλώσσα των μηχανών που αποτελείται από 0 και 1. Η Fuzzy Logic χρησιμοποιεί τις δύο αυτές τιμές 0 και 1 σαν τις ακραίες επιλογές και περιλαμβάνει και τις ενδιάμεσες καταστάσεις. Σε μία μέτρηση για το περιεχόμενο ενός δοχείου δεν το καταλαβαίνει σαν γεμάτο ή άδειο μόνο αλλά αντιλαμβάνεται και τις ενδιάμεσες καταστάσεις που μπορεί να είναι 0.40 γεμάτο. [11]

Η Fuzzy logic φαίνεται να είναι πιο κοντά στον τρόπο που λειτουργεί ο ανθρώπινος εγκέφαλος. Συγκεντρώνει δεδομένα και αναπτύσσει έναν αριθμό από μερικές αλήθειες για το ερώτημα. Βάση των δεδομένων που λαμβάνει αποφασίζει αν θα διαλέξει μεγαλύτερο ή μικρότερο βαθμό. Ακόμα στην περίπτωση που οι επιλογές υπερβούν τα όρια με μία ανάλογη διαδικασία μπορεί να επιλέξει μια πιο ορθή λύση. Ένας παρόμοιος τρόπος λειτουργίας είναι και αυτός που χρησιμοποιούν στην τεχνητή νοημοσύνη, τα νευρωνικά δίκτυα και τα έξυπνα

συστήματα. [11]

Ο Fuzzy Logic ελεγκτής που χρησιμοποιείται εδώ παίρνει στην είσοδό του τα δεδομένα από τον αισθητήρα απόστασης και ανάλογα ποιά ήταν η προηγούμενη κίνηση επιλέγει την επόμενη. Παρακάτω φαίνονται οι επιλογές που κάνει ανάλογα με τις εισόδους του (Πίνακας 3.3.).



Πίνακας 3.3.

Παρόλο την αποτυχία σύνδεσης του BASIC Stamp με το MATLAB, αναπτύχθηκε Fuzzy Logic ελεγκτής με την χρήση του Fuzzy Toolbox. Η διαδικασία δημιουργίας του ελεγκτή με το MATLAB έγινε για να βοηθηθούμε στην συνέχεια, κατά την δημιουργία του ελεγκτή μέσα στον κώδικα του προγράμματος του BASIC Stamp.[12][13]

Αυτό που θέλουμε από τον ελεγκτή είναι στην έξοδό του, η απόφαση για την επόμενη κίνηση που μας δίνει μία ροή στην κίνηση χωρίς απότομες αλλαγές όταν δεν είναι αναγκαίο.

Οι συναρτήσεις συμμετοχής είναι:

- Αν περπατάει γρήγορα μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 70cm, θέλουμε να συνεχίσει κίνηση γρήγορα μπροστά με στροφή αριστερά.
- Αν περπατάει γρήγορα μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 60cm, θέλουμε να συνεχίσει κίνηση γρήγορα μπροστά με στροφή αριστερά.
- Αν περπατάει γρήγορα μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση

45cm, θέλουμε να συνεχίσει κίνηση γρήγορα πίσω με στροφή δεξιά.

- Αν περπατάει γρήγορα μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 30cm, θέλουμε να συνεχίσει κίνηση γρήγορα πίσω.
- Αν περπατάει γρήγορα μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 15cm, θέλουμε να συνεχίσει κίνηση γρήγορα πίσω.
- Αν περπατάει μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 70cm, θέλουμε να συνεχίσει κίνηση μπροστά με στροφή αριστερά.
- Αν περπατάει μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 60cm, θέλουμε να συνεχίσει κίνηση μπροστά με στροφή αριστερά.
- Αν περπατάει μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 45cm, θέλουμε να συνεχίσει κίνηση πίσω με στροφή δεξιά.
- Αν περπατάει μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 30cm, θέλουμε να συνεχίσει κίνηση γρήγορα πίσω με στροφή δεξιά.
- Αν περπατάει μπροστά και εμφανιστεί ένα εμπόδιο σε απόσταση 15cm, θέλουμε να συνεχίσει κίνηση γρήγορα πίσω με στροφή δεξιά.

Για οποιαδήποτε άλλη κίνηση εκτός των μπροστά και γρήγορα μπροστά θέλουμε:

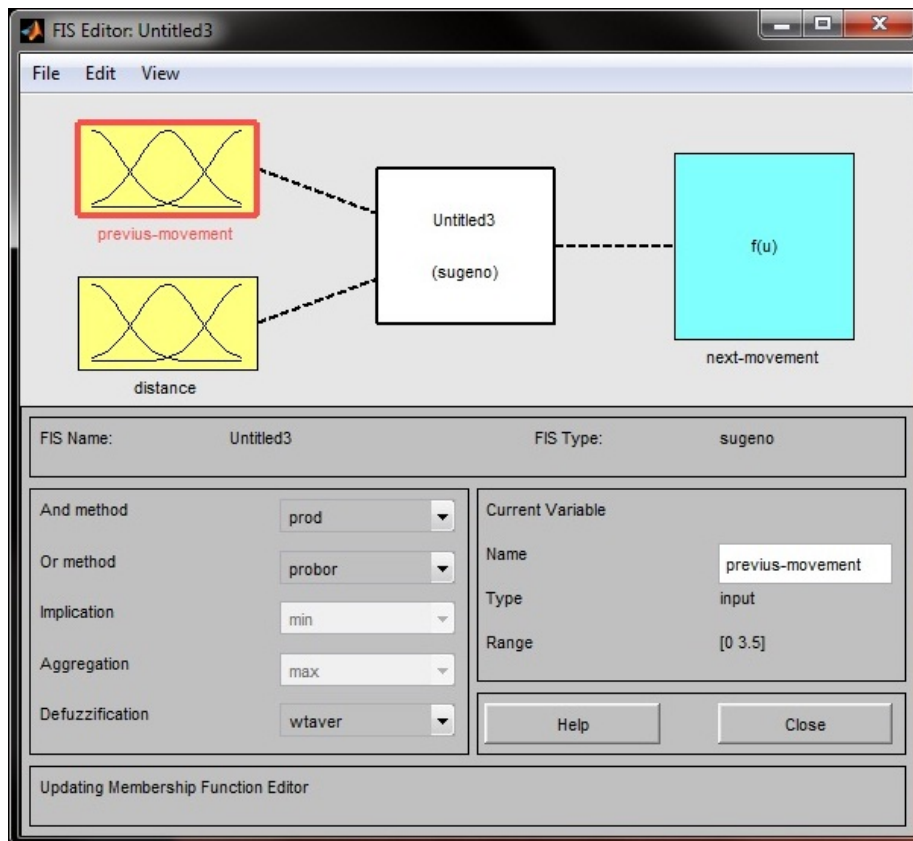
- Αν το εμπόδιο είναι στα 70cm να συνεχίσει κίνηση μπροστά.
- Αν το εμπόδιο είναι στα 60cm να συνεχίσει κίνηση γρήγορα μπροστά και αριστερά.
- Αν το εμπόδιο είναι στα 45cm να συνεχίσει με στροφή αριστερά.
- Αν το εμπόδιο είναι στα 30cm να συνεχίσει κίνηση πίσω δεξιά.
- Αν το εμπόδιο είναι στα 15cm να συνεχίσει κίνηση γρήγορα πίσω δεξιά.

Για την δημιουργία του ελεγκτή χρησιμοποιήθηκε ένας Fuzzy Logic ελεγκτής με δύο εισόδους και μία έξοδο.

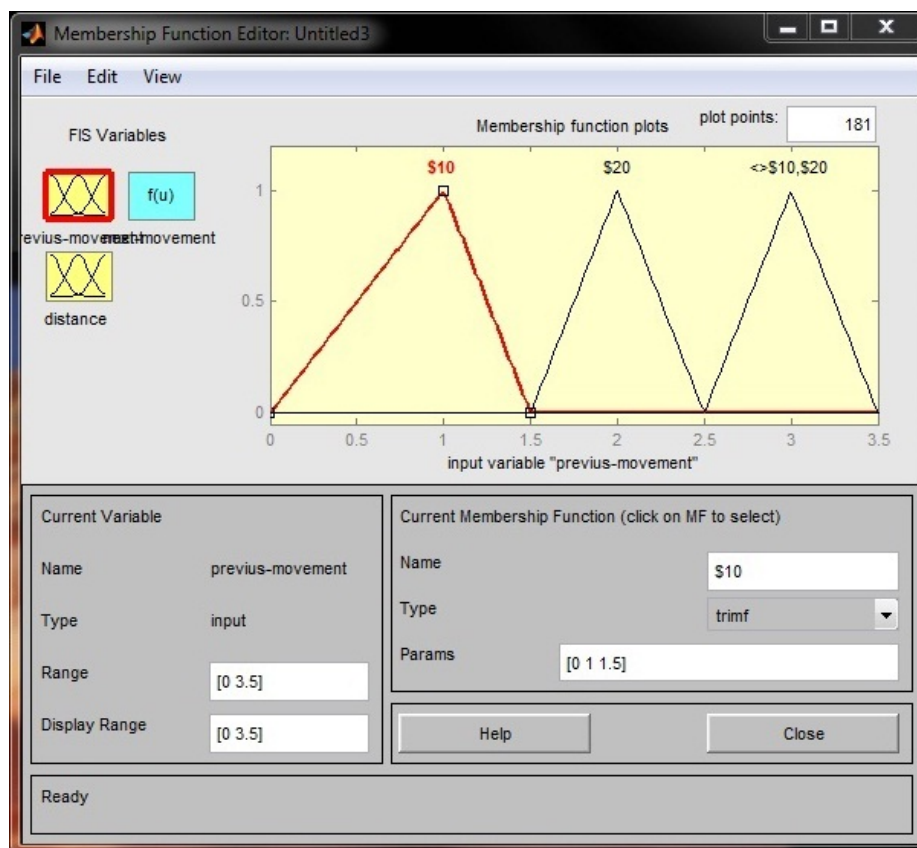
Οι είσοδοι είναι $X_1 = 3\text{mfs}$ και $X_2 = 6\text{mfs}$. Το X_1 αντιστοιχεί στην προηγούμενη κίνηση. Εδώ έχουμε διαλέξει την γρήγορα μπροστά, την μπροστά και το σύνολο των κινήσεων εκτός από τις δύο προηγούμενες. Το X_2 αντιστοιχεί στην απόσταση που μετράει ο αισθητήρας, χωρισμένη σε έξι ομάδες. Η βασική παράμετρος στην είσοδο είναι η απόσταση του εμποδίου αλλά χρησιμοποιούμε και την προηγούμενη κίνηση. Με τον τρόπο αυτό κατά την εμφάνιση κάποιου εμποδίου αναγνωρίζει το ρομποτικό αν το πλησιάζει ενώ βρίσκεται μακριά, απόσταση 80cm, ή αν εμφανίστηκε ξαφνικά μπροστά του, απόσταση μικρότερη των 60cm. Έτσι όσο πιο κοντά εμφανιστεί απότομα κάποιο εμπόδιο τόσο πιο απότομη θα είναι η αντίδραση του ρομποτικού για να το αποφύγει.

Η έξοδος είναι $Y = \text{Sugeno}$. Στην έξοδο μπορεί να δώσει επτά διαφορετικές τιμές που αντιστοιχούν στην μεταβλητή που θα οριστεί ως η επόμενη κίνηση. Το νούμερο που δεχόμαστε είναι δεκαδικός αριθμός. Επειδή το ζητούμενο αποτέλεσμα της εξόδου είναι μία αντιστοιχία σε κίνηση ή μεταβλητή, κρατάμε σαν επιλογή την στρογγυλοποίηση στο πρώτο δεκαδικό ψηφίο, το οποίο γνωρίζουμε σε ποια κίνηση αντιστοιχεί.

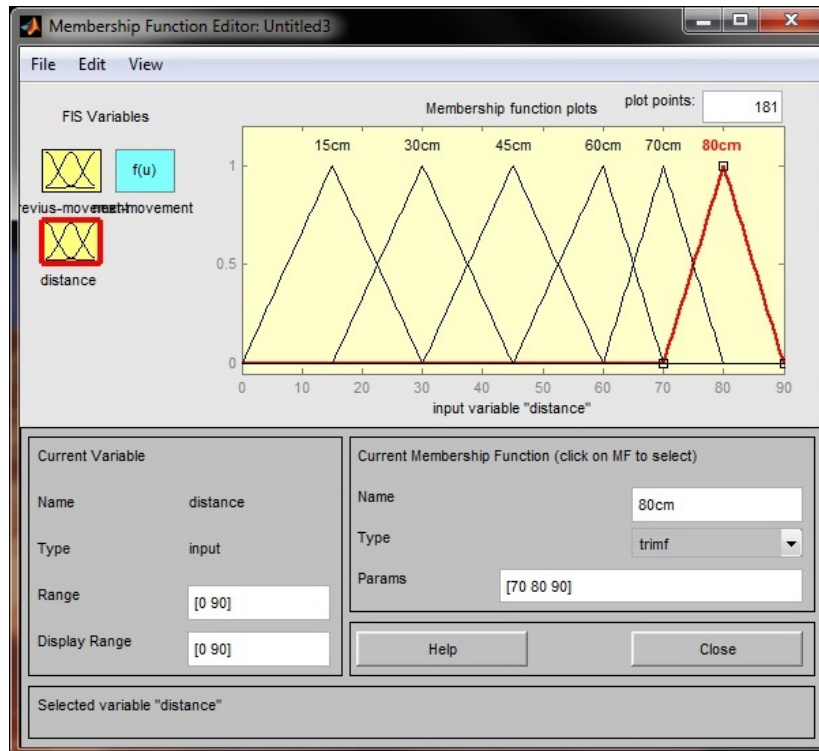
Οι ασαφείς κανόνες που αναφέρθηκαν και προηγουμένως, φαίνονται όπως υλοποιήθηκαν στο MATLAB στην Εικόνα 3.9.



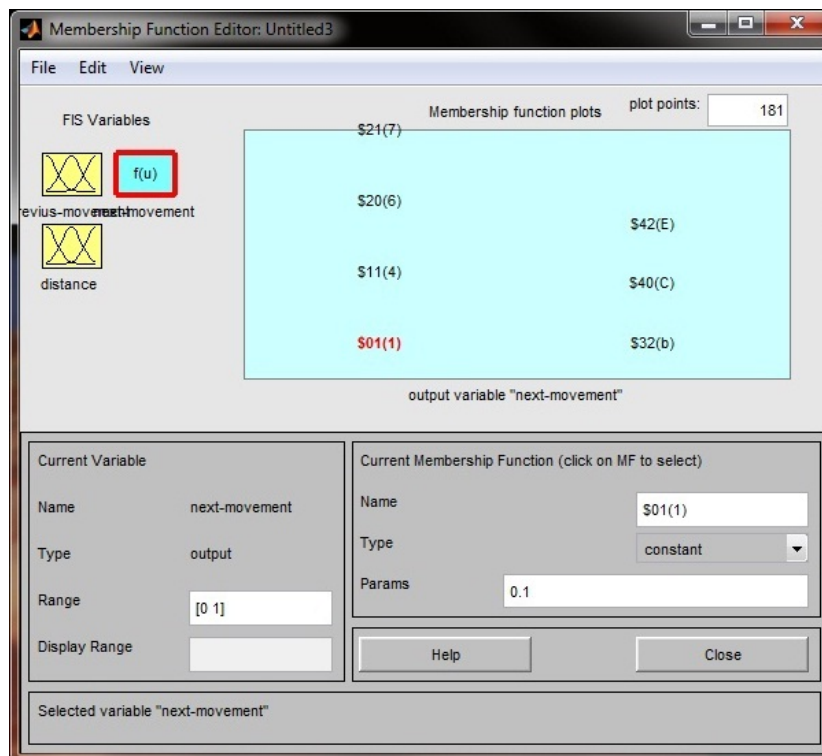
Εικόνα 3.5. Γραφικό περιβάλλον FIS Editor του MATLAB.



Εικόνα 3.6. Συναρτήσεις Συμμετοχής Εισόδου 1



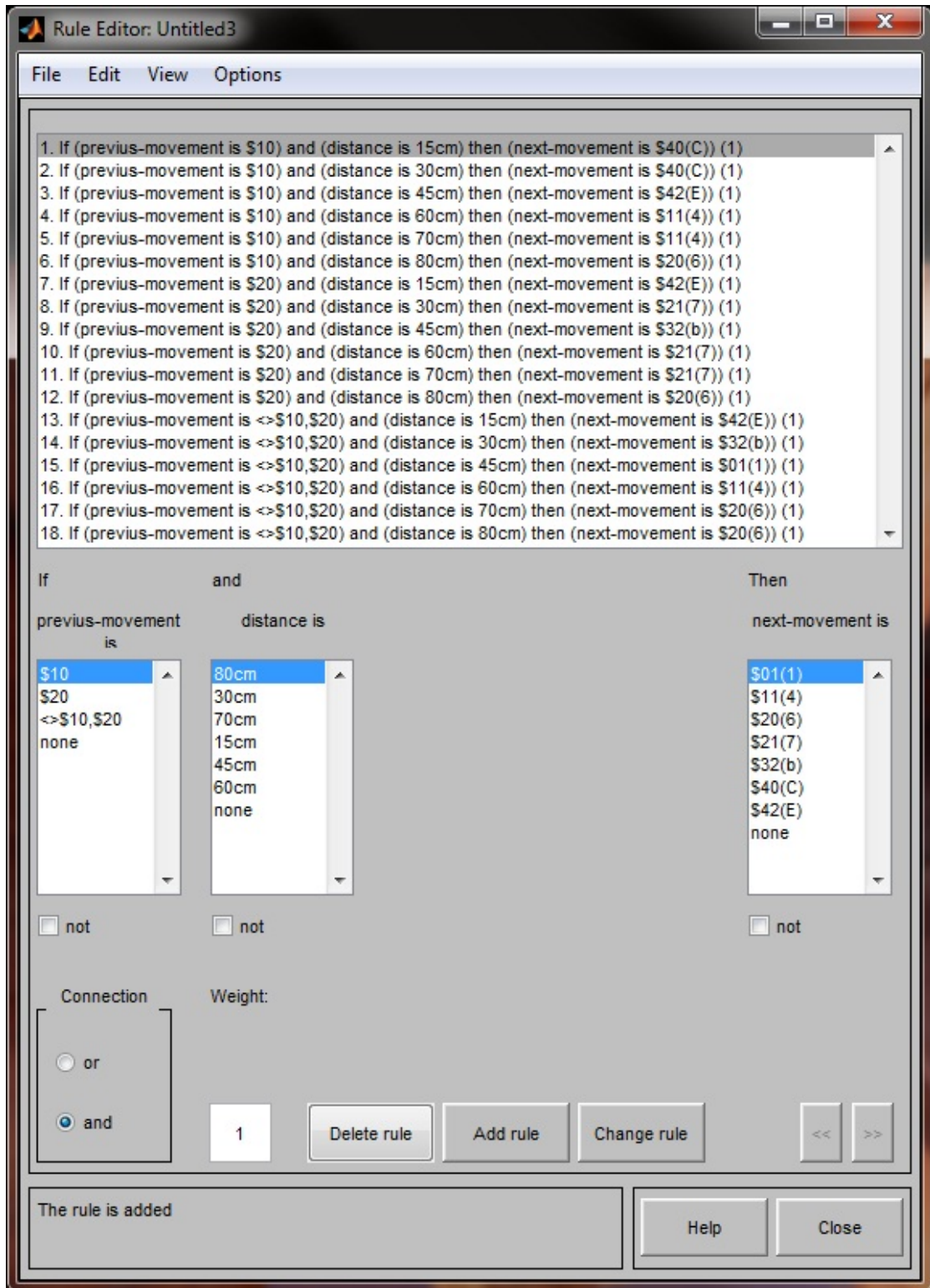
Εικόνα 3.7. Συναρτήσεις Συμμετοχής Εισόδου 2



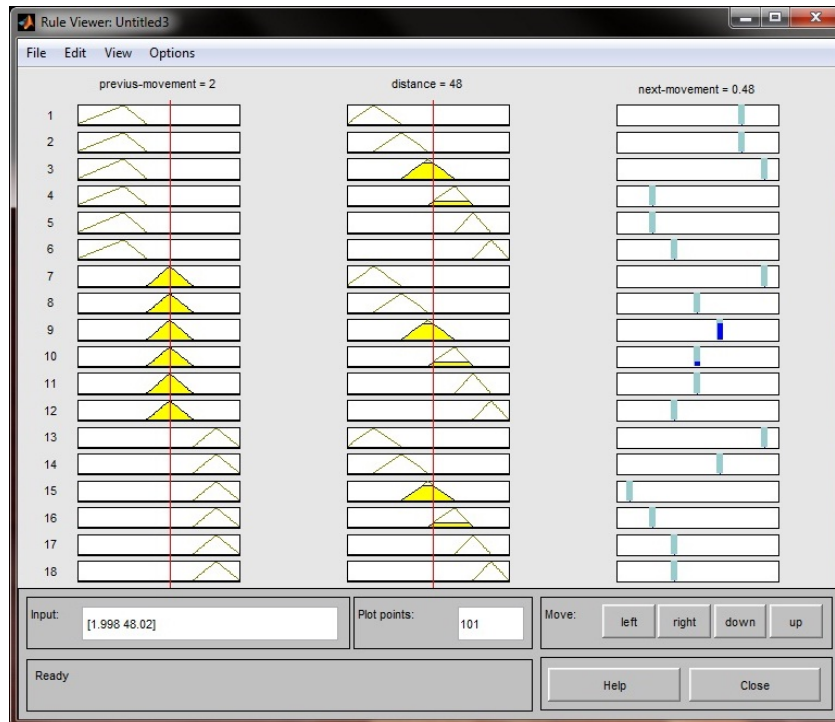
Εικόνα 3.8. Συναρτήσεις Συμμετοχής Εξόδου

Στις συναρτήσεις συμμετοχής παρατηρούμε ότι έχει γίνει η επιλογή τιμών τέτοια ώστε να μην πλέκονται μεταξύ τους οι συναρτήσεις. Η επιλογή αυτή έγινε λόγω ότι η μία είσοδος και η έξοδος αποτελούν μία ακέραια απόφαση και όχι μία δεκαδική τιμή.

Εδώ στην περίπτωση που ήταν εφικτός ο έλεγχος του ρομποτικού από το MATLAB, θα μπορούσε με την χρήση δύο ή και περισσότερων Fuzzy Logic ελεγκτών να πραγματοποιηθούν οι κινήσεις με πολύ πιο φυσικό τρόπο καθώς θα υπήρχε η δυνατότητα για την επιλογή πάρα πολλών διαφορετικών ταχυτήτων ράμπας και εύρους κίνησης των σερβομοτέρ.

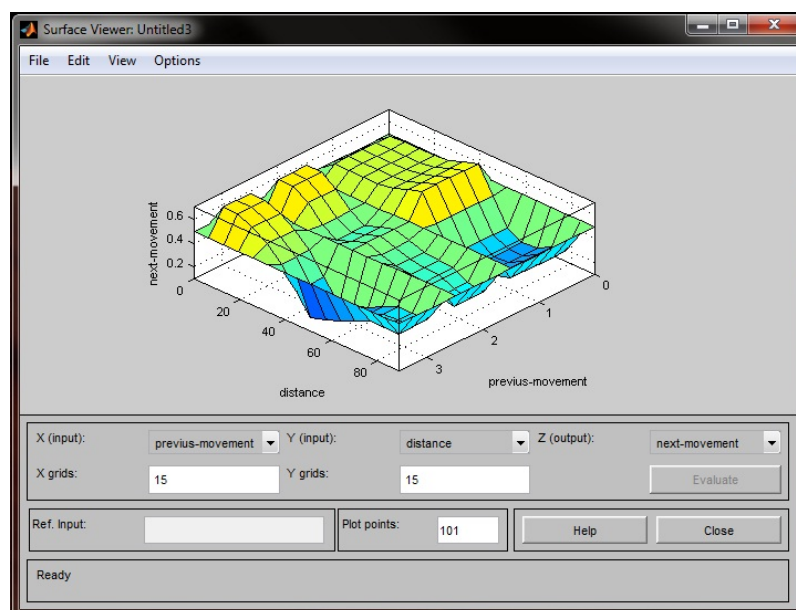


Εικόνα 3.9. Ασαφείς Κανόνες στο MATLAB



Εικόνα 3.10. Rule Viewer του Fuzzy Logic στο MATLAB

Στο Rule Viewer του Fuzzy Logic βλέπουμε ανάλογα με την επιλογή τις τιμές εισόδου πως επιλέγεται η έξοδος. Μετατοπίζοντας τον άξονα των τιμών βλέπουμε συνεχόμενα προς τα που κινείται η επιλογή της εξόδου. Ταυτόχρονα οι επιλογές και τα αποτελέσματα μπορούν να φαίνονται και σε διάγραμμα κάτι που βοηθάει οπτικά να γίνεται γρήγορα και εύκολα έλεγχος για λάθη στους Ασαφείς Κανόνες. Στο παράδειγμα που φαίνεται στην Εικόνα 3.10. στην είσοδο 1 έχουμε 2, δηλαδή την κίνηση \$20 μπροστά. Στην είσοδο 2 την απόσταση 48.02cm και ο ελεγκτής δίνει έξοδο 0.48, με στρογγυλοποίηση στο 0.5, δηλαδή η κίνηση 5, \$32 πίσω δεξιά.



Εικόνα 3.11. Surface Viewer του Fuzzy Logic στο MATLAB

3.6. Διάγραμμα Ροής του Προγράμματος

Παρακάτω στο Διάγραμμα Ροής περιγράφεται η γενική λογική που χρησιμοποιήθηκε για να γραφτεί ο κώδικας του προγράμματος. Στο Παράρτημα Γ. βρίσκεται το αναλυτικό διάγραμμα ροής και στο Παράρτημα Δ. που περιέχει τον κώδικα του προγράμματος υπάρχει αναλυτικά η εξήγηση των εντολών, ρουτινών και υπορουτινών.

Ο κώδικας ξεκινάει ορίζοντας όλες τις μεταβλητές, σταθερές και εκφράσεις που θα χρησιμοποιηθούν. Τις ορίζουμε στην αρχή του προγράμματος ώστε να είναι διαθέσιμες κάθε στιγμή τόσο στο χρήστη για βοήθεια όσο και για να υπάρχει εύκολη πρόσβαση για αλλαγές και πειραματισμούς.

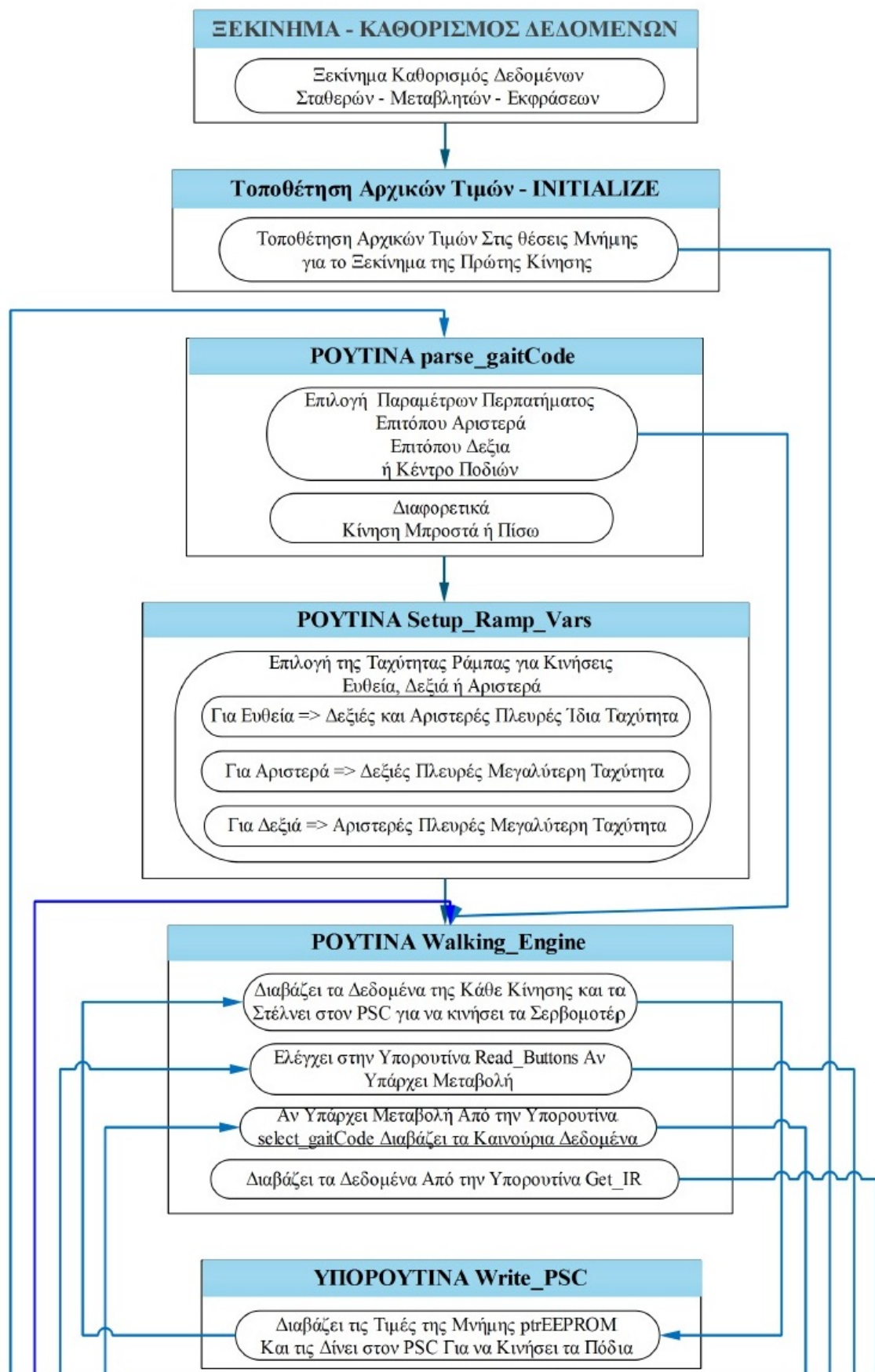
Στην συνέχεια δίνουμε σε κάποιες τιμές στις μεταβλητές που χρειάζονται για να μπορεί να ξεκινήσει να τρέχει ο κώδικας και πηγαίνει στην υπορουτίνα `Select_gaitCode`. Στην υπορουτίνα `Select_gaitCode` ελέγχει τότε θα πατηθούν ταυτόχρονα τα δύο button για να αρχίσει να τρέχει ο κώδικας με την κίνηση που αντιστοιχεί στην ένδειξη του LED Display και συνεχίζει στην ρουτίνα `Parse_gaitCode`.

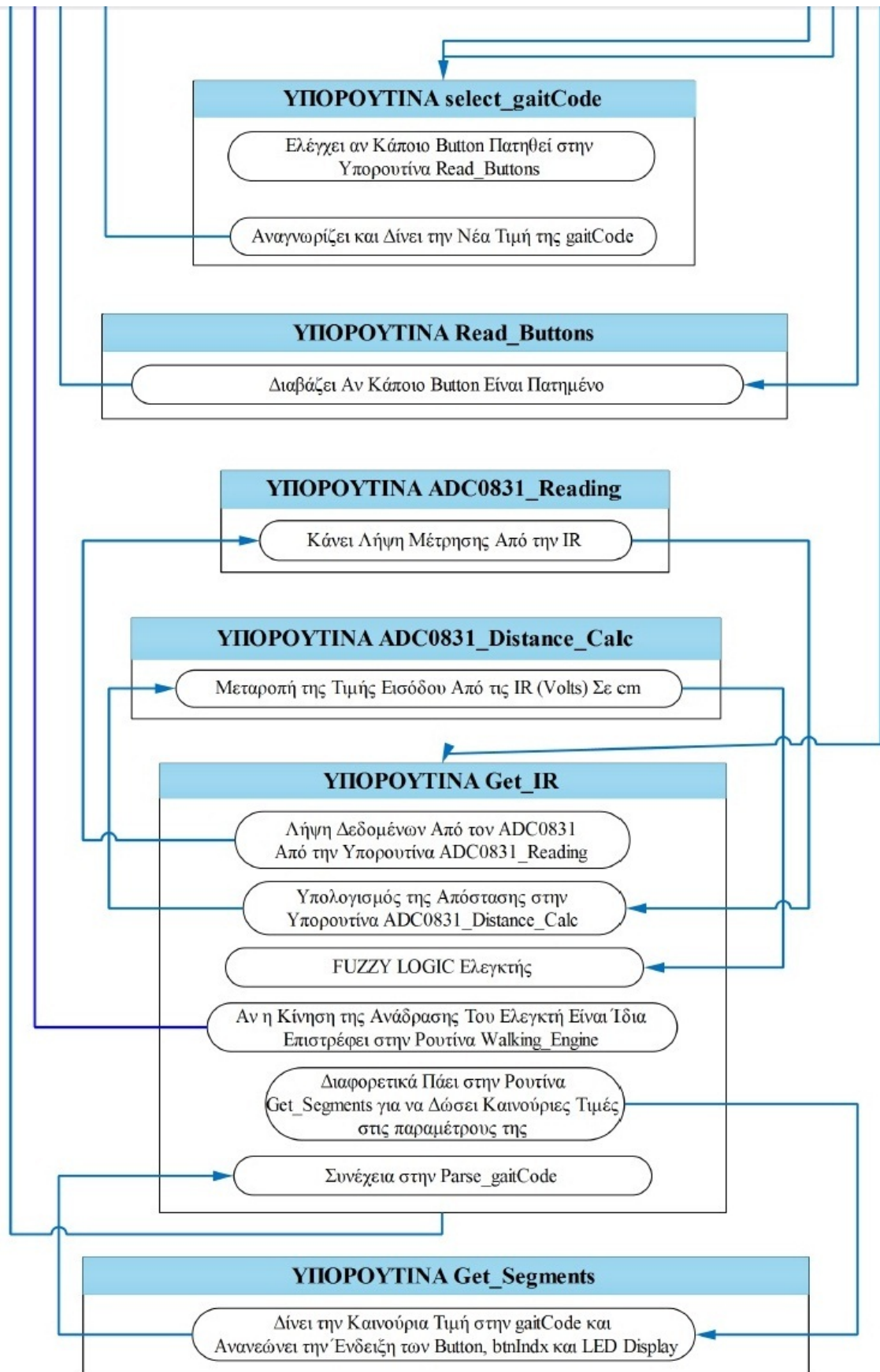
Στην ρουτίνα `Parse_gaitCode` ξεκινάει να επιλέγει τις μεταβλητές που στο τέλος θα συνθέσουν την απόφαση της επόμενης κίνησης. Στην περίπτωση που είναι η πρώτη κίνηση ή η επόμενη κίνηση από αλλαγή απόφασης από τον χρήστη μέσω των button, επιλέγει την ταχύτητα κίνησης των ραμπών των ποδιών ή τοποθετεί τα πόδια στην αρχική τους θέση `Adjust` και συνεχίζει στην ρουτίνα `Walking_Engine`. Διαφορετικά διαλέγει την φορά κίνησης, μπροστά ή πίσω και συνεχίζει στην ρουτίνα `Setup_Ramp_Vars`.

Στην ρουτίνα `Setup_Ramp_Vars` ανάλογα με την απόφαση που έχει δώσει προηγουμένως, ο Fuzzy Logic ελεγκτής στην υπορουτίνα `Get_IR`, επιλέγει την ταχύτητα των ραμπών για την επόμενη κίνηση. Αν η κίνηση είναι ευθεία, δίνει την ίδια ταχύτητα στις ράμπες των ποδιών και των δύο πλευρών, δεξιά και αριστερές. Για αριστερή στροφή δίνει στην αριστερή πλευρά την κανονική ταχύτητα και στην δεξιά γρηγορότερη. Για δεξιά στροφή δίνει στην δεξιά πλευρά την κανονική ταχύτητα και στην αριστερή γρηγορότερη.

Συνεχίζει στην ρουτίνα `Walking_Engine`. Εδώ διαβάζει τα δεδομένα την κίνησης που συνέθεσε προηγουμένως και τα στέλνει στον PSC για να εκτελέσει την κίνηση. Αυτό επαναλαμβάνεται μέχρι να ολοκληρωθεί ένας κύκλος βήματος. Στην συνέχεια ελέγχει αν η προηγούμενη κίνηση ήταν η `Adjust` και σταματάει την ροή του προγράμματος μέχρι να ξαναξεκινήσει πατώντας τα δύο button ταυτόχρονα. Διαφορετικά ελέγχει τυχόν μεταβολές στην μεταβλητή των button. Αν δεν υπάρχει συνεχίζει στην ρουτίνα `Walking_Engine`. Αν μεταβάλλεται διαβάζει την καινούρια τιμή και συνεχίζει στην υπορουτίνα `Select_gaitCode` εκτελεί της εντολές της και επιστρέφει για να συνεχίσει στην υπορουτίνα `Get_IR` και να πάρει καινούριες μετρήσεις.

Στην υπορουτίνα `Get_IR` παίρνει τις καινούριες μετρήσεις από τις υπέρυθρες μέσω της υπορουτίνας `ADC0831_readings`. Μετατρέπει την μέτρηση σε cm μέσω της υπορουτίνας `ADC0831_distance_Calc` και δίνει το αποτέλεσμα στον Fuzzy Logic ελεγκτή. Αν δεν υπάρχει αλλαγή της καινούριας τιμής με την προηγούμενη συνεχίζει στην ρουτίνα `Walking_Engine` και κάνει πάλι τον ίδιο βηματισμό. Αν υπάρχει αλλαγή γράφει τα καινούρια δεδομένα στις αντίστοιχες μεταβλητές και συνεχίζει στην υπορουτίνα `Get_Segments` για να αλλάξει την ένδειξη στο LED Display και πηγαίνει στην ρουτίνα `Parse_gaitCode`





ΚΕΦΑΛΑΙΟ 4ο

4.1. Συμπεράσματα και προτάσεις περαιτέρω μελέτης ή ανάπτυξης

Ο τρόπος κατασκευής του ρομποτικού δίνει την δυνατότητα στο χρήστη να μπορεί να έχει πολλές δυνατότητες βελτίωσης τόσο στην μηχανική κατασκευή όσο στην ηλεκτρονική και στον προγραμματισμό του.

Σαν μηχανική κατασκευή βασικές αλλαγές που μπορούν να γίνουν είναι η αντικατάσταση των τμημάτων αλουμινίου με όμοια κατασκευασμένα από Carbon, ένα υλικό πολύ πιο ελαφρύ από το αλουμίνιο και ανάλογα με την ποιότητά του πολύ πιο ανθεκτικό. Με τον τρόπο αυτό μπορεί να μειωθεί σημαντικά το βάρος του ρομποτικού, καθώς ταυτόχρονα να δημιουργηθεί μία πολύ πιο ανθεκτική και στιβαρή κατασκευή.

Μηχανικά, βελτιώσεις μπορούν να γίνουν και στα πόδια του ρομποτικού. Ο τρόπος κίνησης των ποδιών στο οριζόντιο επίπεδο γίνεται κυκλικά ως προς τον άξονα στήριξης του ποδιού. Αυτό δημιουργεί ένα μεγάλο πρόβλημα. Η άκρη του ποδιού κατά την κίνηση εμπρός ή πίσω διαγράφει ένα τόξο. Σαν κυκλικό τμήμα βλέπουμε ότι κατά την κίνηση το άκρο του ποδιού έρχεται πιο κοντά με τον οριζόντιο άξονα του ρομποτικού κάθε φορά που κινείται είτε προς τα μπροστά είτε προς τα πίσω και να βρίσκεται στο πιο απομακρυσμένο σημείο όταν είναι στο κέντρο. Αυτός ο τρόπος κίνησης δημιουργεί επιπλέον τριβές και απώλειες ενέργειας. Το πρόβλημα με τις αυξημένες τριβές που δημιουργούνται μεταξύ ποδιού και επιφάνειας που περπατάει το ρομποτικό μπορούν να μειωθούν με τον επανασχεδιασμό των άκρων των ποδιών. Το άκρο θα μπορούσε να κατασκευαστεί με τρόπο ώστε να δίνει μία ελευθερία κίνησης μέσα – έξω, κάθετη προς τον άξονα της βάσης του ποδιού. Έτσι θα είναι εφικτό να μετατρέπει τις τριβές που δημιουργούν μεγάλες απώλειες, σε μία παθητική κίνηση μιας επιπλέον άρθρωσης του ποδιού με μικρότερες απώλειες. Τέλος στα άκρα των ποδιών μπορούν να τοποθετηθούν ρουλεμάν κάθετου φορτίου που θα μειώσουν αρκετά τις κάθετες τριβές.

Στο ηλεκτρονικό μέρος οι βελτιώσεις που μπορούν να γίνουν είναι πολλές. Οι μεγάλες δυνατότητες του μικροελεγκτή επιτρέπουν την προθήκη επιπλέον αισθητηρίων και ηλεκτρονικών για την αναγνώριση των εμποδίων και της τοποθεσίας του ρομποτικού.

Μία βασική βελτίωση του ρομποτικού είναι η καλύτερη αναγνώριση του χώρου που βρίσκεται. Η προσθήκη τριών επιπλέον αισθητήρων απόστασης δεξιά αριστερά και πίσω μπορεί να δώσει την άνεση στον προγραμματιστή να γράψει ένα κώδικα στον οποίο το ρομποτικό θα μπορεί με ένα πιο πολύπλοκο ελεγκτή να αποφεύγει πιο γρήγορα πολλαπλά εμπόδια ή να διαλέγει να κινείται κοντά σε αυτά. Αντίστοιχη λειτουργία μπορεί να γίνει χρησιμοποιώντας ένα σερβομοτέρ σαν βάση για της υπέρυθρες που με την κίνηση αριστερά και δεξιά θα μπορεί να συλλέγει πληροφορίες για την απόσταση των αντικειμένων.

Εκτός από την αποφυγή εμποδίων μπορεί να γίνει χρήση άλλων αισθητηρίων και προγραμματισμός για να ακολουθάει κάποιο άλλο κινούμενο αντικείμενο, καθώς και να διαλέγει ποιο αντικείμενο πρέπει να ακολουθήσει. Το ίδιο εύκολη είναι και η χρήση αισθητηρίου sonar αντί για υπέρυθρες, αλλά ανεβάζει το κόστος. Τέλος πολύ ενδιαφέρουσα και με πολλές προοπτικές θα ήταν η χρήση GPS για την κίνηση και την αναγνώριση του χώρου.

Πειραματισμός μπορεί να γίνει και με τον τρόπο περπατήματος. Αντί να υπάρχουν δύο προκαθορισμένες ταχύτητες κίνησης, μπορεί να γραφτεί μία ρουτίνα που ανάλογα με την

απόφαση ενός ελεγκτή για την ταχύτητα, να μεταβάλει την ταχύτητα των ποδιών αλλάζοντας την τιμή της μεταβλητής της ταχύτητας της ράμπας κίνησης. Λειτουργώντας με αυτό τον τρόπο η κίνηση θα γίνεται πιο ομαλά και η λειτουργία του Fuzzy Logic θα πλησιάζει περισσότερο την τέλεια απόφαση.

Μία σημαντική παρατήρηση για τα σερβομοτέρ πρέπει να αναφερθεί στην ποιότητά τους. Λόγω της φθηνής ποιότητας κατά τον προγραμματισμό παρατηρήθηκε ότι κάποια σερβομοτέρ λειτουργούσαν αντίστροφα από άλλα. Εκεί χρειάστηκε να γίνουν πρακτικές αλλαγές στο πρόγραμμα και να αντιστραφούν τα νούμερα της κίνησης. Επίσης λόγω της χαμηλής ποιότητας πολλά σερβομοτέρ δεν άντεχαν να λειτουργούν για παραπάνω από 2-3 λεπτά, ανέβαζαν θερμοκρασία και καίγονταν. Μία ενδιαφέρουσα ιδέα για πειραματισμό έχει να κάνει με την βελτίωση της αντοχής των σερβομοτέρ μέσω του προγράμματος κίνησης. Αν μπορέσει να γραφτεί κώδικας που θα κινεί το πόδι με χαμηλή αρχικά ταχύτητα για να έχει ένα πιο ομαλό ξεκίνημα στην κίνηση και στη συνέχεια να την αυξάνει σταδιακά στην επιθυμητή, θα μπορέσει να αποφορτίσει τον ηλεκτροκινητήρα από το μεγάλο ρεύμα που δημιουργείται την στιγμή του ξεκινήματος της κίνησης και τα γρανάζια από την απότομη ροπή που δέχονται. Με τον τρόπο αυτό θα αυξηθεί το όριο ζωής τους και θα χαμηλώσει η κατανάλωση ενέργειας λόγω απωλειών.

ΚΕΦΑΛΑΙΟ 5ο

Βιβλιογραφία – Αναφορές

1. <http://www.aircraftspruce.com/catalog/mepages/aluminfo.php>, προσπέλαση στις 25-11-2012
http://www.alcoa.com/mill_products/catalog/pdf/Alcoa_Insert_5052and6061_FINAL.pdf, προσπέλαση στις 25-11-2012
<http://www.asecoltd.com/userfiles/cliparts/Aluminum%205052.pdf>, προσπέλαση στις 25-11-2012
2. PARALLAX Inc.: Jeff Martin, Jon Williams, Ken Gracey, Aristides Alvarez, and Stephanie Lindsay, “BASIC Stamp Syntax and Reference Manual”, αρ. έκδοσης 2.2
3. PARALLAX Inc., “Board of Education - USB (#28850)” αρ. Έκδοσης 2.0, 5/14/2010 Σχέδιο BASIC Stamp
4. Parallax, Inc. “Parallax Servo Controller USB – Rev B”, αρ. Έκδοσης 3.4 (#28823), 8/2006
5. <http://www.servodatabase.com/servo/towerpro/mg995>, προσπέλαση στις 15-01-2013
6. http://www.sharpsma.com/webfm_send/1203, “SHARP IR gp2y0a21yk_e.pdf”, προσπέλαση στις 15-01-2013
7. Parallax, Inc. “Basic Analog and Digital Student Guide”, αρ. Έκδοσης 1.4
8. Texas Instruments, “ADC0831/ADC0832/ADC0834/ADC0838 8-Bit Serial I/O A/D Converters with Multiplexer Options”, ημ. Έκδοσης Αύγουστος 1999
9. <http://www.acroname.com/robotics/info/articles/irlinear/irlinear.html>, προσπέλαση στις 17-01-2013
10. <http://www.mathworks.com/matlabcentral/fileexchange/32329-matlab-support-package-for-parallax-basic-stamp>, <http://forums.parallax.com>, προσπέλαση στις 04-07-2013
11. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.html, προσπέλαση στις 04-07-2013
12. S. N. Sivanandam, S. Sumathi and S. N. Deepa, “Introduction to Fuzzy Logic using MATLAB”, αρ. Έκδοσης © Springer-Verlag Berlin Heidelberg 2007
13. The MathWorks, Inc., “Fuzzy Logic Toolbox™ User’s Guide”, αρ. Έκδοσης September 2013 Online only Revised for Version 2.2.18 (Release 2013b)
14. PARALLAX Inc., “BASIC Stamp BS2-IC Module Schematic”, αρ. Έκδοσης 10, 19-10-2012

Παραρτήματα

A. Συντομογραφίες

H/Y = Ηλεκτρονικός Υπολογιστής

CD-ROM = Compact Disk

USB = Universal Serial Bus

RS-232 = Recommended Standard – 232 serial port

GPS = Global Positioning System

BS2 = BasicStamp

PSC = Parallax Servo Controller

B.O.E. = Board Of Education

μPs = Microprocessor

IC = **Integrated Circuit**

EEPROM = **E**lectrically **E**rasable **P**rogrammable **R**OM

A/D = Analogue To Digital

ADC = Analogue To Digital Converter

IR = Infrared

I/O = Input / Output

Hz = hertz

A = Ampere

V = Volt

VDC =

R = Resistance

Ω = Ohm

W = Watt

C = Capacitor

f = frequency

μ = Mikro

m = Milli

K = Kilo

sec = Second

cm = Centimeter

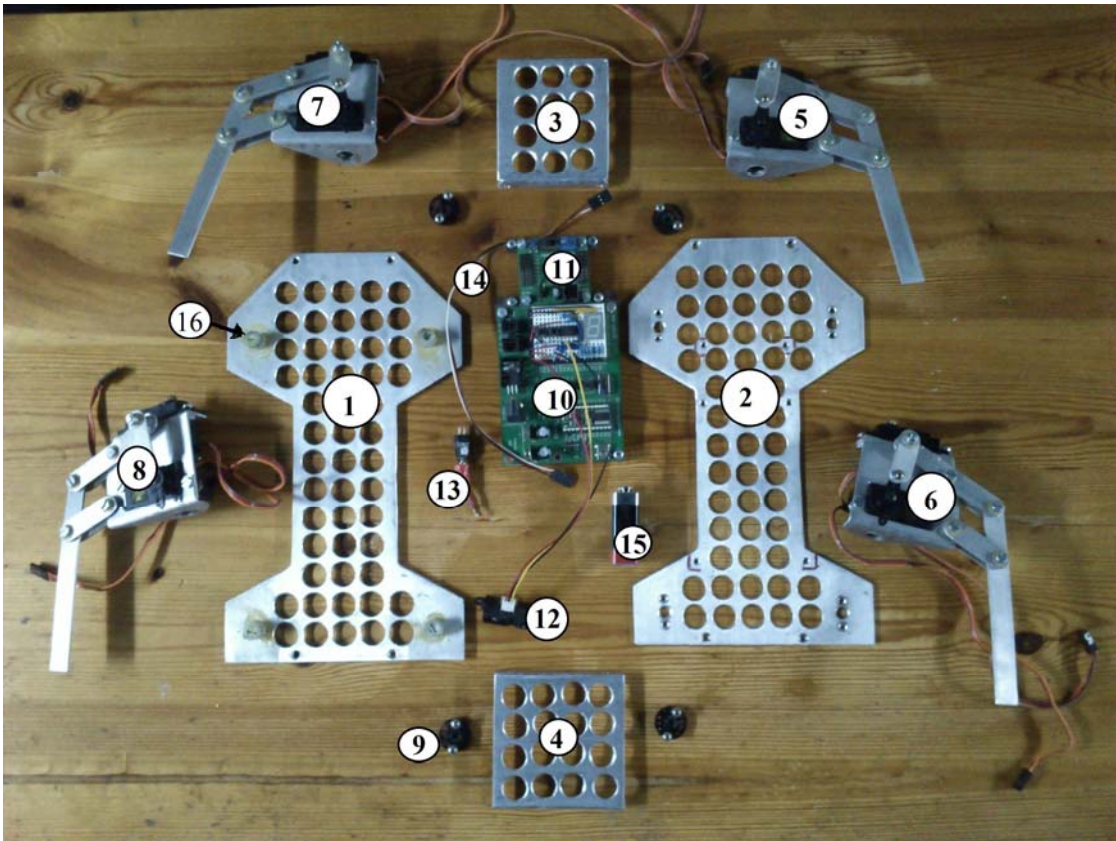
mm = Millimeter

kg = Kilogram

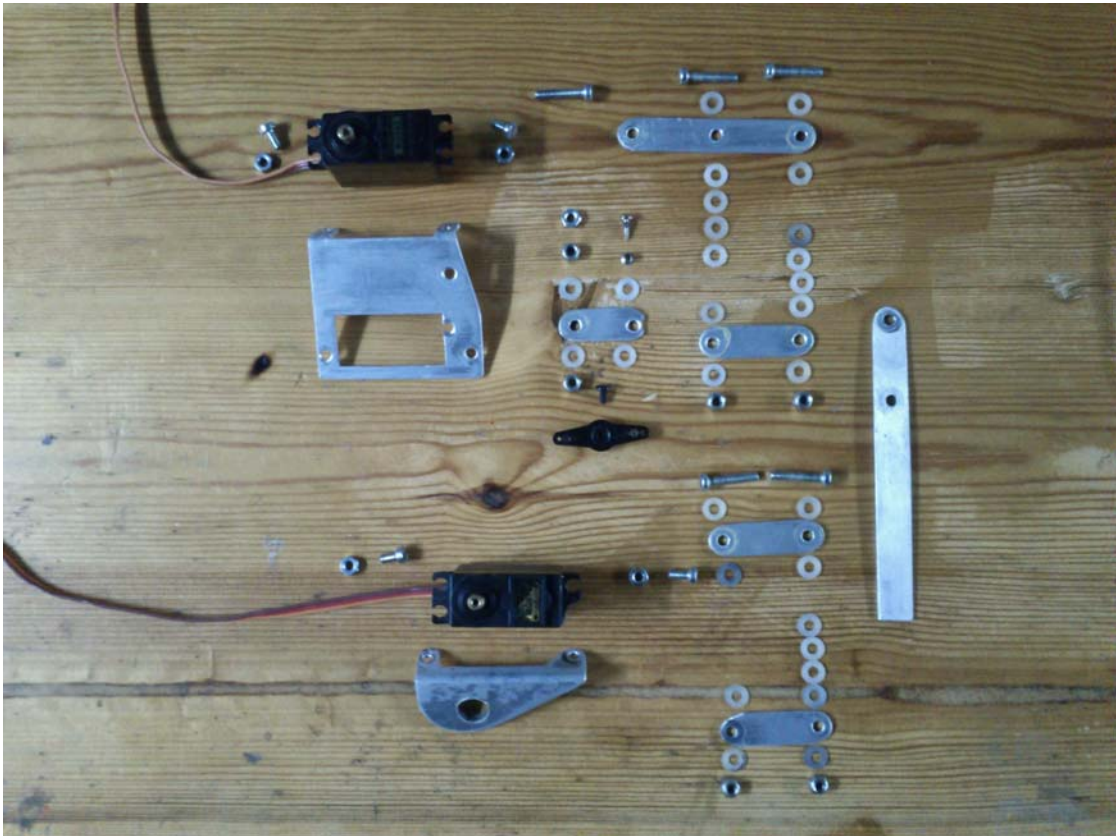
gr = Grammar
C = Celsius
F = Fahrenheit
No. = Number
HEX = hexademical
MB = Megabyte
Temp = Temperature
Topr = Operation Temperature
Tstg = Storage Temperature
Sout = Serial Output
Sin = Serial Input
ATN = Attention
Vin = Intake Voltage
Vo = Output Voltage
Vss = Voltage Common Source
VDD = Voltage Common Drain
P0 – P15 = pin0 –pin15
RES = Reset
Vref = Reference Voltage
Rx = Reciver
Tx = Transmeter
CS = Chip select (
CLK = Clock
Ch. = Channel
NC = Not Connected
LSB = Less Significan Byte
Σχ. =Σχήμα
Εικ. = Εικόνα

B. Συναρμολόγηση Του Ρομποτικού

Η συναρμολόγηση του ρομποτικού ξεκινάει από την αναγνώριση των τμημάτων του. Στην εικόνα B.1. φαίνονται τα τμήματα και στην εικόνα B.2. είναι τα τμήματα ενός ποδιού με την σειρά που χρειάζεται να ακολουθηθεί για να δεθεί.



Εικόνα Β.1.



Εικόνα Β.2.

Αναγνωρίζουμε τα βασικά μέρη του σκελετού, τα δύο πατώματα, 1 το κάτω και 2 το πάνω, και τα δύο κάθετα τμήματα 3 και 4. Αυτά ενώνονται με 8 βίδες με παξιμάδι από την μέσα πλευρά και συναρμολογείται το σώμα του ρομποτικού. Μετά τοποθετούμε τις τέσσερις βάσεις των σερβομοτέρ, εξαρτήματα 16 και 9.



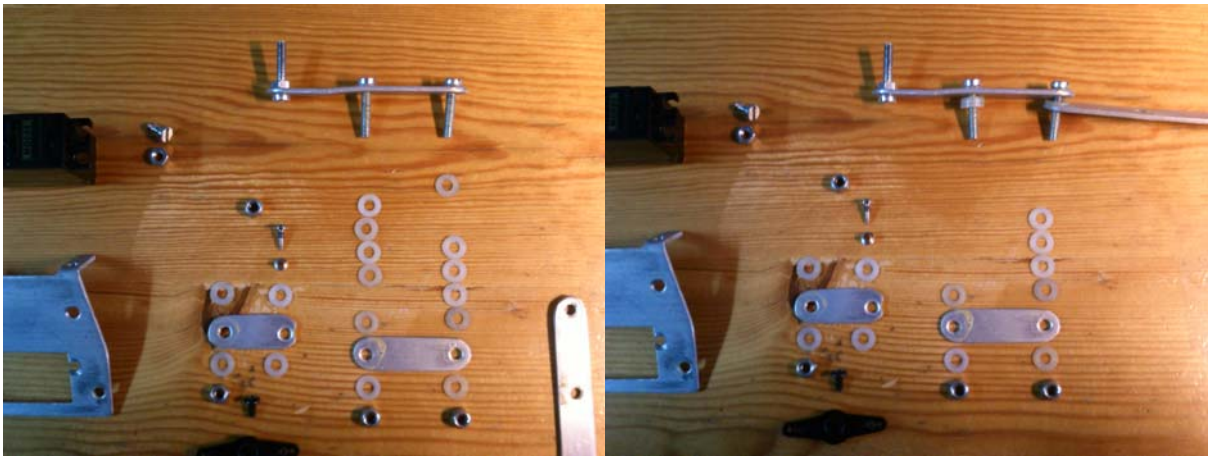
Εικόνες B.3., B.4.



Εικόνες B.5., B.6.

Συνέχεια έχει το πιο πολύπλοκο μέρος, τα πόδια. Αναγνωρίζουμε το κομμάτι που ενώνει το κάθετο άκρο του ποδιού με το κομμάτι που ενώνεται στο σερβομοτέρ (Εικόνα B.7.). Έχει τρεις τρύπες, άκρες και κέντρο. Στην μία άκρη βιδώνουμε βίδα με παξιμάδι και σφίγγουμε κανονικά. Στο κέντρο και στην άλλη άκρη βίδα και παρεμβάλουμε παντού ροδέλες τεφλόν. Στο κέντρο από την πλευρά του σώματος μπαίνουν τέσσερις ροδέλες για να καλύψουν την διαφορά του κενού. Στην άλλη άκρη μία ροδέλα, η άκρη του κάθετου κομματιού και μετά τέσσερις ροδέλες για να καλυφθεί το κενό. Στη συνέχεια τοποθετείται το διπλανό κομμάτι στήριξης αφού η μεσαία βίδα περάσει από την πάνω εξωτερική βίδα του σώματος του ποδιού. Τέλος ροδέλες και τα παξιμάδια ασφαλείας. Χρησιμοποιούνται παξιμάδια τύπου ασφαλείας για να μπορούμε να τα σταθεροποιούμε σε μία θέση οριακά πριν αρχίσει να σφίγγει η άρθρωση και τόσο ώστε να μην παρουσιάζει αστάθεια. Με τον ίδιο τρόπο συνεχίζουμε και την κάτω οριζόντια στήριξη ακολουθώντας τη σειρά όπως φαίνεται και

στις εικόνες. Η βίδα στην μέσα πλευρά κρατάει μαζί και την μία μεριά του κάτω μέρους του σώματος του ποδιού.



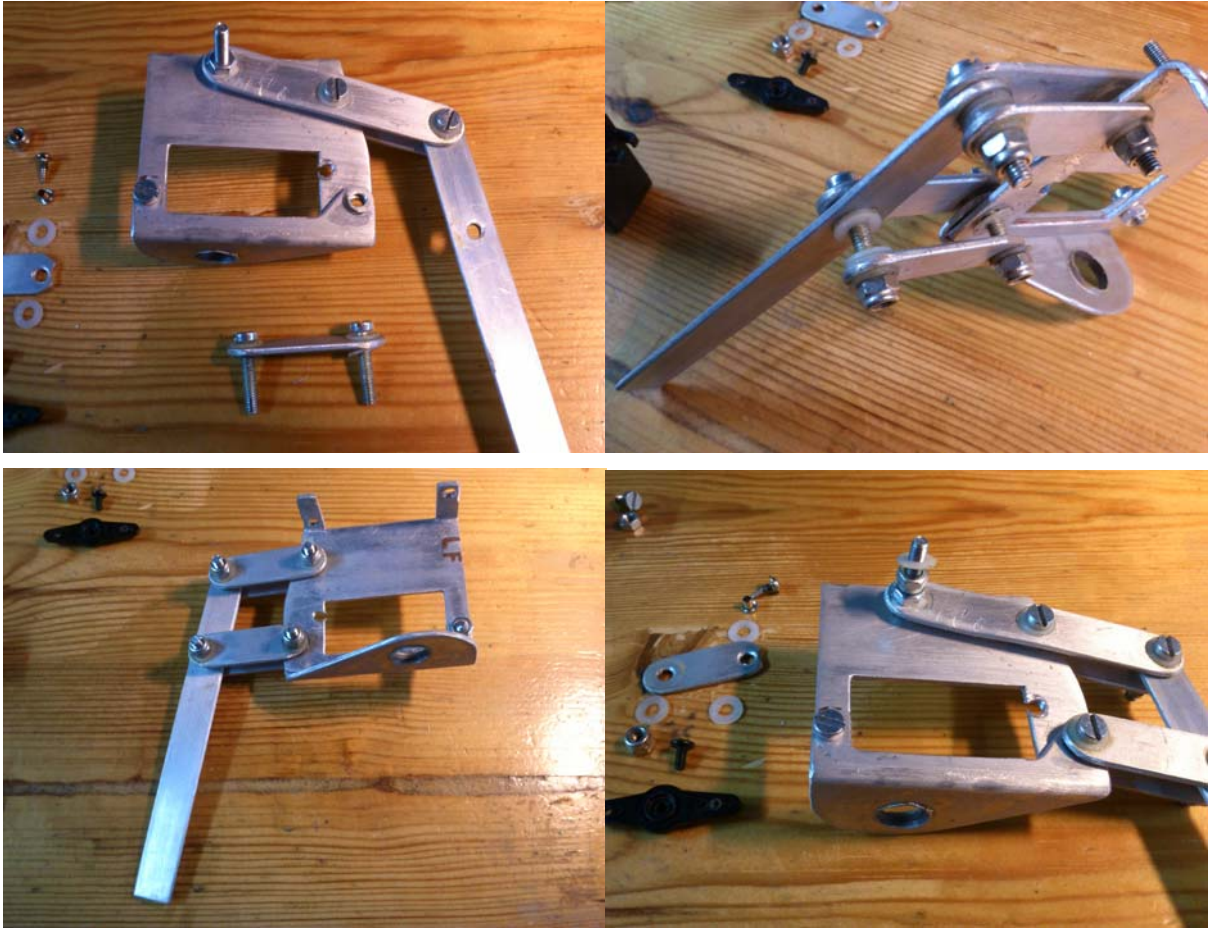
Εικόνες B.7., B.8.



Εικόνες B.9., B.10.

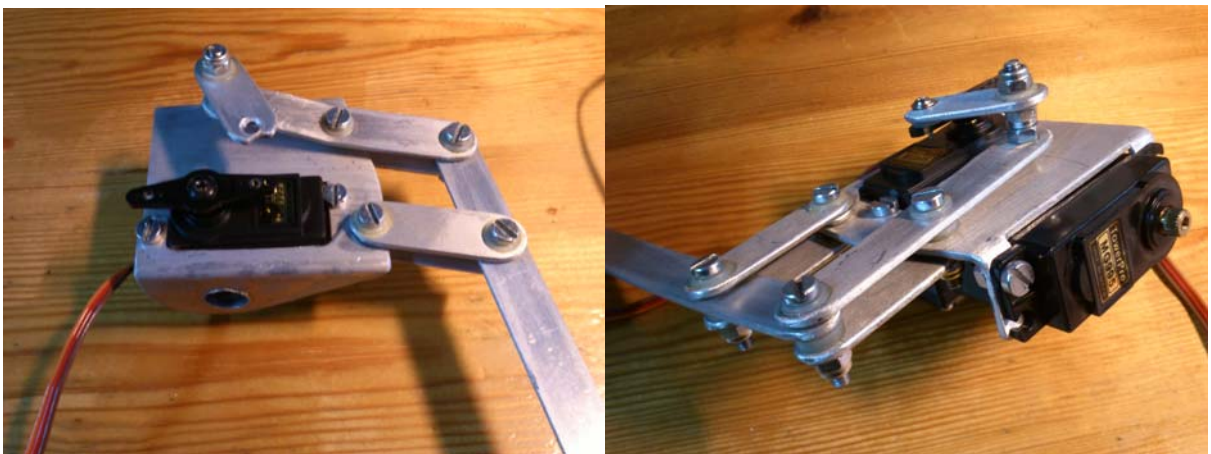


Εικόνες B.11., B.12.



Εικόνες B.13., B.14., B.15., B.16.

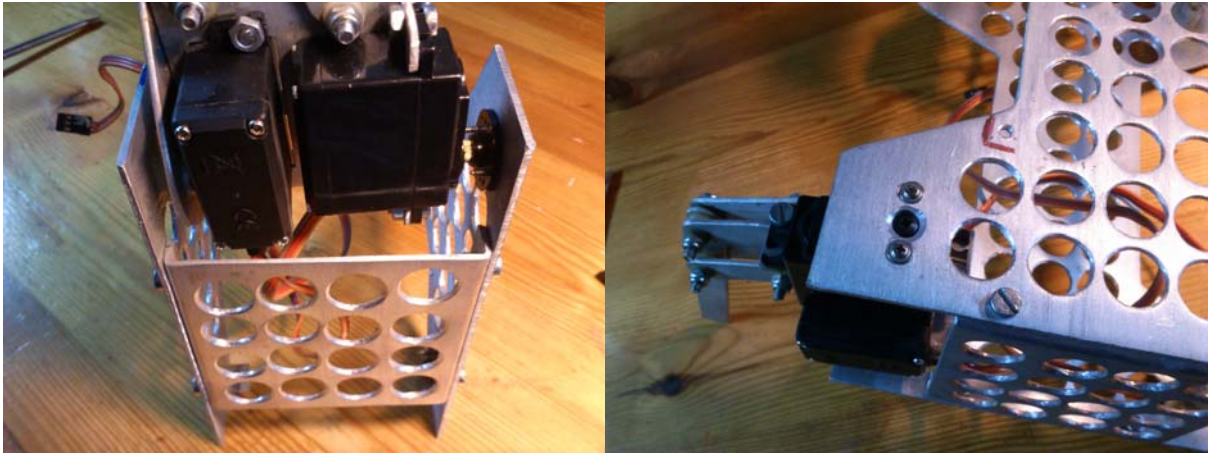
Στην συνέχεια τοποθετούνται τα σερβομοτέρ και ενώνεται ο άξονας του σερβομοτέρ της κάθετης κίνησης με την άρθρωση του ποδιού. Στο σημείο αυτό μεταξύ του άξονα και του τμήματος του ποδιού βάζουμε 2 ροδέλες και στο κέντρο τους τοποθετείται αποστάτης για να μειώνονται οι τριβές, Εικόνες B.17., B.18..



Εικόνες B.17., B.18.

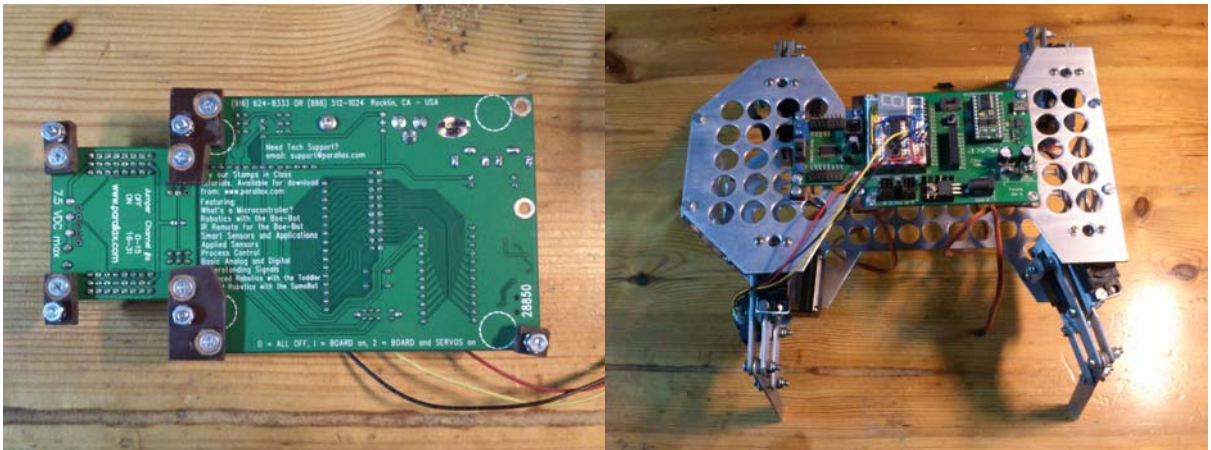
Μετά την συνομολόγηση των ποδιών τα τοποθετούμε στο σώμα του ρομποτικού. Κρατώντας το πόδι πλάγια με την βάση του κάτω πατώματος περνάμε την τρύπα που έχει

στην κάτω πλευρά του ποδιού στην κυλινδρική βάση. Στην συνέχεια ευθυγραμμίζουμε τον άξονα του σερβομοτέρ με την βάση στο πάνω πάτωμα και βιδώνουμε τον άξονα με την βάση, Εικόνες B.19., B.20..

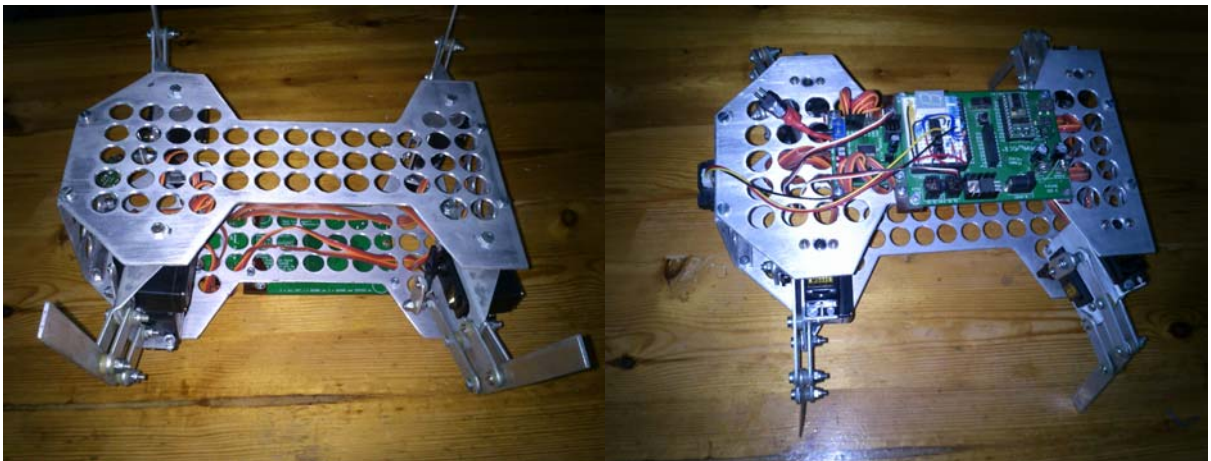


Εικόνες B.19., B.20.

Τέλος τοποθετούνται τα ηλεκτρονικά εξαρτήματα και οι καλωδιώσεις τους. Η συνομολόγηση των κυκλωμάτων εξηγείται μέσα στο κείμενο στο δεύτερο κεφάλαιο, 2.2. Ηλεκτρονικό Μέρος



Εικόνες B.21., B.22.



Εικόνες B.23., B.24.

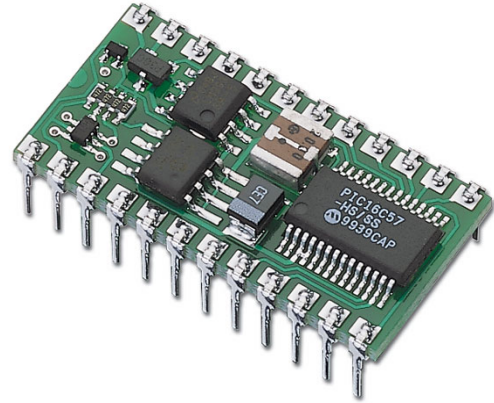
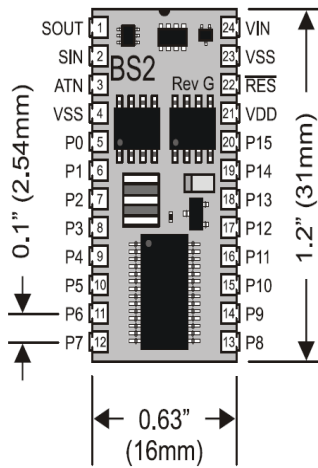
Γ. Χαρακτηριστικά Εξαρτημάτων

Γ.1. Μικροελεγκτής Basic Stamp 2

Τα χαρακτηριστικά του BASIC Stamp BS2 φαίνονται στον παρακάτω πίνακα, (Πίνακας Γ.1.), η μορφή του στην εικόνα, (Εικόνα Γ.1.), και στο διάγραμμά του (Διάγραμμα Γ.1.) [2].

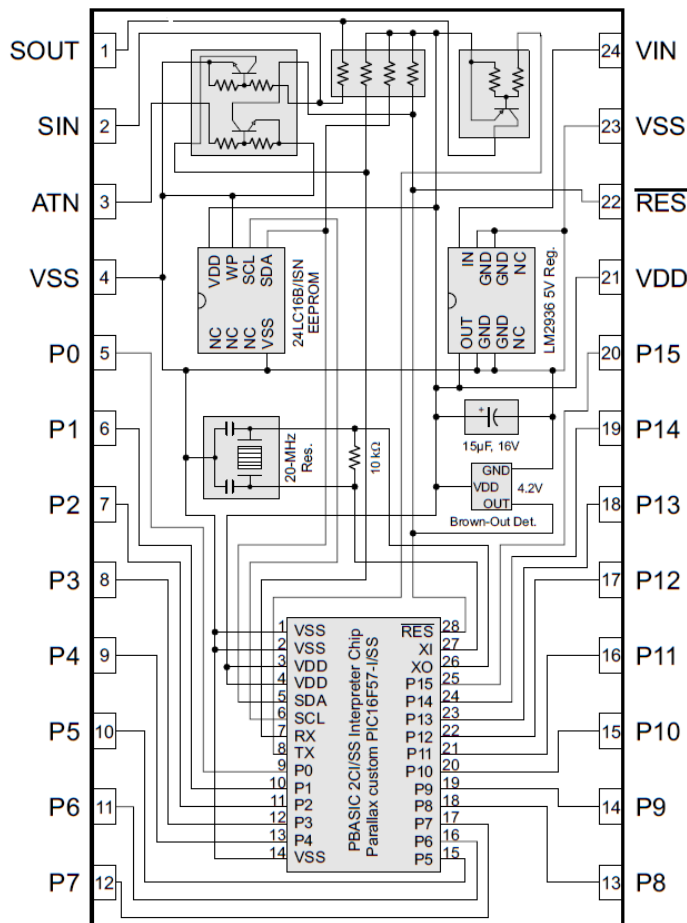
Θερμοκρασία Λειτουργίας Περιβάλλοντος	0° - 70° C (32° - 158° F)
Μικροελεγκτής	Microchip PIC16C57c
Ταχύτητα Επεξεργαστή	20 MHz
Ταχύτητα Εκτέλεσης Προγράμματος	~4,000 instructions/sec
Μνήμη	32 Bytes(6 I/O, 26 Variable)
Χωρητικότητα EEPROM (προγράμματος)	2K Bytes,~500 instructions
Αριθμός I/O Pins	16 + 2 Dedicated Serial
Τάση Λειτουργίας	5 - 15 vdc
Κατανάλωση Ρεύματος στα 5Volts	3 mA σε λειτουργία, 50 μΑ σε αναστολή
Ρεύμα Εξόδου/Εισόδου ανά I/O	20 mA / 25 mA
Συνολικό Ρεύμα Εξόδου/Εισόδου I/O	40 mA / 50 mA ανά 8 I/O pins
Εντολές PBASIC	42
Σύνδεση με H/Y	Σειριακή (9600 baud)
Έκδοση Επεξεργαστή Κειμένου Προγράμματος	Stampw.exe (v1.04 και νεότερο)

ΠΙΝΑΚΑΣ Γ.1.



ΕΙΚΟΝΑ Γ.1

BASIC Stamp[®] BS2-IC Module Schematic



BASIC Stamp 2
(Rev. G) (Stock# BS2-IC).

Διάγραμμα Γ.1.

Στο διάγραμμα του BASIC Stamp 2 βλέπουμε την αρίθμηση και την λειτουργία του κάθε ακροδέκτη του. [14].

Στον παρακάτω πίνακα, (Πίνακας Γ.2.) υπάρχει η επεξήγηση και πληροφορίες της κάθε λειτουργίας.

Αρ. Pin	Όνομα	Περιγραφή
1	SOUT	Σειριακή Έξοδος: συνδέεται στην σειριακή θύρα του H/Y στο Pin RX (DB9 pin2 / DB25 pin3) για τον προγραμματισμό
2	SIN	Σειριακή Είσοδος: συνδέεται στην σειριακή θύρα του H/Y στο Pin TX (DB9 pin3 / DB25 pin2) για τον προγραμματισμό
3	ATN	Attention: συνδέεται στην σειριακή θύρα του H/Y στο Pin DTR (DB9 pin4 / DB25 pin20) για τον προγραμματισμό
4	VSS	Γείωση Συστήματος: (ίδια με το pin23) συνδέεται στην σειριακή θύρα του H/Y στο Pin GND (DB9 pin5 / DB25 pin7) για τον προγραμματισμό
5 έως 20	P0-P15	Γενικά – Σκοπός I/O Pins: Το καθένα μπορεί να δεχτεί ρεύμα εισόδου 25mA και να δώσει 20mA. Το σύνολο του ρεύματος όλων των I/O Pins δε μπορεί να υπερβαίνει τα 50mA στην είσοδο και τα 40mA αν χρησιμοποιείται ο εσωτερικός ανορθωτής τάσης των 5volt. Το σύνολο του ρεύματος ανά 8 I/O Pins (pin0 έως pin7 και pin 8 έως pin15) δε μπορεί να υπερβαίνει τα 50mA στην είσοδο και τα 40mA αν χρησιμοποιείται ο εσωτερικός ανορθωτής τάσης των 5volt.
21	VDD	5volt DC είσοδος/ έξοδος: Αν δίνεται μη σταθεροποιημένη τάση στην είσοδο VIN, τότε στην έξοδο του pin VDD θα πάρουμε 5volt. Αν δεν δίνεται τάση στην είσοδο VIN τότε στην VDD πρέπει να δίνεται σταθεροποιημένη τάση 4.5v – 5.5v
22	RES	Reset είσοδος/έξοδος: όταν η τάση πέφτει κάτω από τα 4.2volts πηγαίνει σε κατάσταση LOW και οδηγεί τον BASIC Stamp σε επανεκκίνηση. Μπορεί να οδηγηθεί σε κατάσταση LOW για να κάνουμε επανεκκίνηση. Το pin αυτό είναι σε κατάσταση HIGH και αν δεν χρειάζεται μπορεί να μείνει ασύνδετο.
23	VSS	Γείωση Συστήματος: (το ίδιο με το pin4) συνδέεται με την γείωση της πηγής τροφοδοσίας (GND)
24	VIN	Μη σταθεροποιημένη είσοδος τάσης: Δέχεται τάση 5.5-15VDC η οποία σταθεροποιείται εσωτερικά στα 5Volts. Πρέπει να μένει ασύνδετη αν δίνεται τάση 5volts στο pin VDD.

Πίνακας Γ.2.

BASIC Stamp 2 Περιγραφή pin.

Γ.2. Μητρική Πλακέτα B.O.E. (Board Of Education® - USB (#28850)) [3]

Τα χαρακτηριστικά της είναι:

- Η βάση για τον μικροελεγκτή δέχεται όλους τους BASIC Stamp με 24-pin .
- Οι συνδέσεις των εισόδων/εξόδων I/O του BASIC Stamp και η τροφοδοσία βρίσκονται και δίπλα σε μία πλακέτα ανάπτυξης (breadboard) 2" x 1 3/8" για μεγαλύτερη ευκολία και χρηστικότητα.
- Ο διακόπτης τροφοδοσίας είναι τριών θέσεων ώστε να είναι δυνατός ο προγραμματισμός του ελεγκτή χωρίς να ενεργοποιούνται οι θύρες επικοινωνίας των σερβομοτέρ, κάτι πολύ χρήσιμο για χρήση του συστήματος σε ρομποτικά.
- Γέφυρα για επιλογή της τάσης λειτουργίας των σερβομοτέρ, ρυθμισμένη ($V_{dd} = 5\text{ V}$) ή αρρυθμιστή (V_{in}).
- Υποδοχή για βύσμα τροφοδοσίας τύπου Jack 2.1 mm με θετική τάση στον κεντρικό αγωγό και υποδοχή για μπαταρία 9V, μηχανικά τοποθετημένα ώστε να μην είναι δυνατή η χρήση και των δυο ταυτόχρονα.
- Τάση τροφοδοσίας 6-9V DC.
- USB Mini-B βύσμα με ενσωματωμένο μετατροπέα από USB σε σειριακή σύνδεση για τον προγραμματισμό του BASIC Stamp και την επικοινωνία σε πραγματικό χρόνο με H/Y κατά την εκτέλεση του προγράμματος.
- Ενσωματωμένη παροχή τάσης 5V ως και 1Amp ρεύματος για να μπορεί να καλύψει απαιτήσεις μεγαλύτερων κατασκευών.
- 2 x 10 θηλυκή βάση σύνδεσης με τους ακροδέκτες του ελεγκτή.
- Διαστάσεις 7.75x10.16cm.
- Γειωμένες τρύπες στις γωνίες γι διατομής 3.18mm για στήριξη με κέντρα τρυπών τους 9.5x7cm.
- Θερμοκρασία λειτουργίας 0–70 °C.

Ανάλυση των εξαρτημάτων της Μητρικής Πλακέτας B.O.E. (Εικόνα Γ.2.):

1. Υποδοχή μπαταρίας 9V: Μπορεί να χρησιμοποιηθεί αλκαλική ή επαναφορτιζόμενη μπαταρία 9V.
2. Είσοδος βύσματος τροφοδοσίας: Δέχεται βύσμα των 2.1mm με τη θετική τάση στο κέντρο του με τάση λειτουργίας 6-9V. Τα δύο βύσματα εισόδου τάσης λειτουργίας φαίνονται στην εικόνα Γ2 πώς είναι τοποθετημένα ώστε να μην είναι δυνατή η σύνδεση και των δύο ταυτόχρονα.
3. Ρυθμιστής τάσης: Παρέχει τάση 5V και μέχρι 1A ρεύματος σε οποιαδήποτε επαφή και σύνδεση έχει την ένδειξη V_{dd} .
4. LED ένδειξη λειτουργίας: Αυτό το LED ανάβει όταν υπάρχει τροφοδοσία ρεύματος και ο διακόπτης βρίσκεται στη θέση 1 ή 2.
5. Βύσματα σύνδεσης με Σερβομοτέρ (X4 και X5) και γέφυρα επιλογής τάσης: Το καθένα από αυτά έχει 2 σειρές των τριών pin συνδέσμων που δίνουν τάση, γείωση και I/O,

τοποθετημένα μαζί για να μπορεί να γίνει εύκολα η σύνδεση με σερβομοτέρ. Η τάση που δίνουν είναι Vdd (+5 V) αλλά μπορεί και να επιλεγεί και η τάση εισόδου λειτουργίας Vin βραχυκυκλώνοντας την γέφυρα που βρίσκεται ανάμεσά τους. Κάθε σειρά των τριών pin είναι αριθμημένη με τα νούμερα 12, 13, 14, και 15. Οι I/O γραμμές σήματος για τα σερβομοτέρ μπορούν να είναι προσβάσιμες και από τα pin P12, P13, P14, και P15 I/O των βυσμάτων X1 και X2.

6. Βύσμα τροφοδοσίας (X3): Τα βύσματα με την ένδειξη Vdd είναι συνδεδεμένα με +5VDC, τα Vin συνδεδεμένα με την τάση λειτουργίας και τα Vss με την γείωση.

7. Πλακέτα ανάπτυξης (breadboard): Η πλακέτα ανάπτυξης έχει μεταλλικές συνδέσεις από την κάτω πλευρά της σε οριζόντια σειρά και συνδέει 5 υποδοχές μεταξύ τους, με δύο συμπλέγματα ανά σειρά χωρισμένα από μία κάθετη εγκοπή. Καλώδια ή άκρα από εξαρτήματα μπορούν να συνδεθούν έτσι μεταξύ τους. Εξαρτήματα όπως ολοκληρωμένα κυκλώματα IC ή διακόπτες τοποθετούνται στο κέντρο της πλακέτας ανάπτυξης για να χωρίζονται οι επαφές τους σε συνδέσεις από την αριστερή και την δεξιά πλευρά

8. Βάση πρόσβασης Pin I/O (X2): Τα 16 I/O pin της βάσης για τους μικροελεγκτές BASIC Stamp, αριθμημένα από το 0 ως το 15, είναι συνδεδεμένα σε αυτήν τη βάση για να μπορεί να υπάρχει εύκολα πρόσβαση και σύνδεση με εξωτερικά κυκλώματα πάνω στην πλακέτα ανάπτυξης.

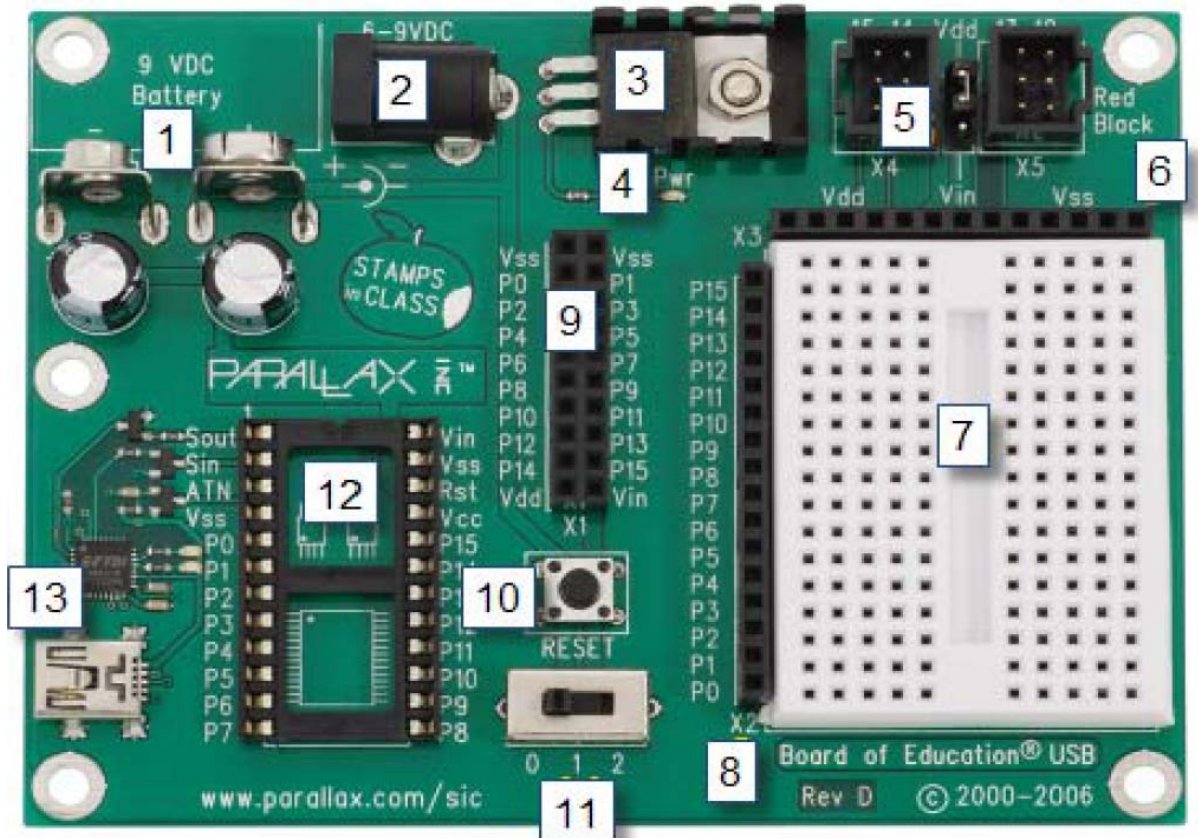
9. Βάση έτοιμων εφαρμογών (X1): Η βάση έτοιμων εφαρμογών παρέχει σύνδεση με τα I/O pin του μικροελεγκτή, τάση λειτουργίας Vdd, Vin και Vss, για συσκευές που είναι κατασκευασμένες να λειτουργούν με αυτή την υποδοχή όπως Bluetooth και LSD οθόνες.

10. Μπουτόν επανεκκίνησης (Reset Button): Το μπουτόν επανεκκίνησης χρησιμοποιείται για να επανεκκινηθεί ο BASIC Stamp χωρίς να χρειαστεί να διακοπεί η τάση λειτουργίας. Με αυτό τον τρόπο ελαττώνονται φθορές στο διακόπτη λειτουργίας για απλές επανεκκινήσεις του προγράμματος. Μερικά προγράμματα χρησιμοποιούν τεχνικές ώστε κατά το πάτημα του μπουτόν να αλλάζει η λειτουργία του προγράμματος.

11. Διακόπτης λειτουργίας τριών θέσεων: Όταν βρίσκεται στην αριστερή θέση (0) όλες οι τάσεις είναι αποσυνδεδεμένες. Στη μεσαία θέση (1) παρέχεται τροφοδοσία στην υποδοχή του μικροελεγκτή και σε όλες τις συνδέσεις με το σήμα Vin. Ακόμα ενεργοποιείται η τάση Vdd(5Volts) στην πλακέτα ανάπτυξης και στην βάση εφαρμογών X1. Στην δεξιά θέση (2) ενεργοποιήσαμε και η τάση στις συνδέσεις των σερβομοτέρ X4 και X5.

12. Υποδοχή για μικροελεγκτή BASIC Stamp: Η υποδοχή αυτή είναι συμβατή με όλους τους μικροελεγκτές BASIC Stamp των 24-pin. Συνδέει τον BASIC Stamp με στο μέσο προγραμματισμού, στην ένδειξη λειτουργίας, στο μπουτόν επανεκκίνησης και σε όλες τις I/O επαφές.

13. Βύσμα προγραμματισμού USB: Αυτό είναι ένα βύσμα USB mini-B και μετατροπέας USB σε σειριακό (RS-232) για προγραμματισμό και αμφίδρομη σύνδεση και μεταφορά δεδομένων του BASIC Stamp με τον H/Y.



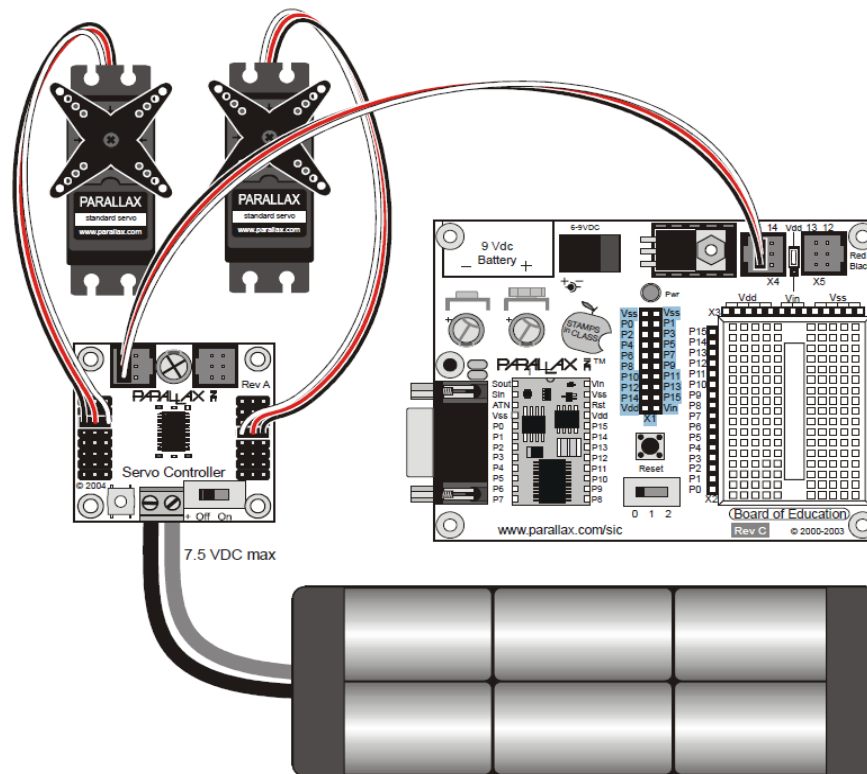
Εικόνα Γ.2.

Γ.3. Ελεγκτής Σερβομοτέρ της Parallax (#28023) – Rev B (Parallax Servo Controller (#28023) – Rev B) [4]

Τα χαρακτηριστικά του είναι:

- Δυνατότητα οδήγησης έως και 16 σερβομοτέρ.
- Δυνατότητα σύνδεσης με άλλο PSC και οδήγηση 32 σερβομοτέρ από μία μόνο έξοδο του ελεγκτή.
- Ταυτόχρονη οδήγηση όλων των σερβομοτέρ, συνεχόμενα και με 0-180° γωνία περιστροφής.
- Λειτουργία Ramping. Δυνατότητα επιλογής 63ων διαφορετικών ταχυτήτων κίνησης για κάθε σερβομοτέρ.
- Αναφορά Θέσης. Ο χρήστης μπορεί να γνωρίζει την ακριβή θέση που βρίσκεται ο κάθε σερβομοτέρ.
- Τάση λειτουργίας του ελεγκτή του PSC 5VDC.
- Τάση λειτουργίας για τους σερβομοτέρ 4-7.5 VDC.
- Διαστάσεις 38mm X 33mm.

Παρακάτω φαίνεται μία τυπική σύνδεση του PSC με την B.O.E, δύο σερβομοτέρ και την πηγή τάσης (Εικόνα Γ.3.)



Εικόνα Γ.3.

Γ.4. Σερβομοτέρ [5]

Τα χαρακτηριστικά τους είναι τα ακόλουθα:

- Τάση λειτουργίας: 4.8 - 7.2Volts
- Τροφοδοσία Τάσης: εξωτερική μέσω ελεγκτή.
- Ταχύτητα λειτουργίας: 0.17sec / 60degrees (4.8V χωρίς φορτίο).
- Ταχύτητα λειτουργίας: 0.13sec / 60degrees (6.0V χωρίς φορτίο).
- Ροπή στρέψης: 9kg-cm (180.5oz-in) στα 4.8V.
- Ροπή στρέψη: 12kg-cm (208.3oz-in) στα 6V.
- Τύπος γραναζιών: Μεταλλικά.
- Καλωδίωση σύνδεσης 300mm.
- Σταθερότητα και αντοχή στις δονήσεις.
- Διαστάσεις: 40 * 19 * 43mm.
- Βάρος: 55gr



Εικόνα Γ.4. Σερβομοτέρ MG-995

Γ.5. Αισθητήρας απόστασης Sharp IR GP2D12 [6]

Βασικά από τα χαρακτηριστικά του είναι:

- Η απόσταση δίνεται από τάση σε αναλογική έξοδο.
- Μετράει απόσταση από 10-80cm
- Δεν χρειάζεται πρόσθετο κύκλωμα για να λειτουργήσει
- Η μικρή επιρροή από τα χρώματα των αντικειμένων που αντανακλούν οι υπέρυθρες .
- Η γραμμικότητα στον τρόπο που μπορεί ο χρήστης να πάρει στην έξοδο του συστήματος ανίχνευσης
- Χαμηλό κόστος

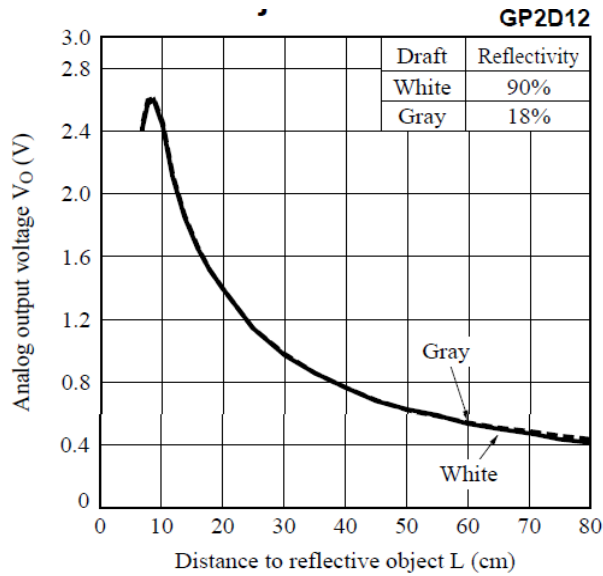
Τα όρια λειτουργίας του φαίνονται στον παρακάτω πίνακα (Πίνακας Γ.3.) και η αναλογική έξοδος τάσης με απόσταση αντικείμενου στο διάγραμμα (Γραφική Παράσταση Γ.1.)

■ Absolute Maximum Ratings

($T_a=25^{\circ}\text{C}$, $V_{cc}=5\text{V}$)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{cc}	-0.3 to +7	V
Output terminal voltage	V_o	-0.3 to $V_{cc}+0.3$	V
Operating temperature	T_{opr}	-10 to +60	$^{\circ}\text{C}$
Storage temperature	T_{stg}	-40 to +70	$^{\circ}\text{C}$

Πίνακας Γ.3.



\Γραφική Παράσταση Γ.1.

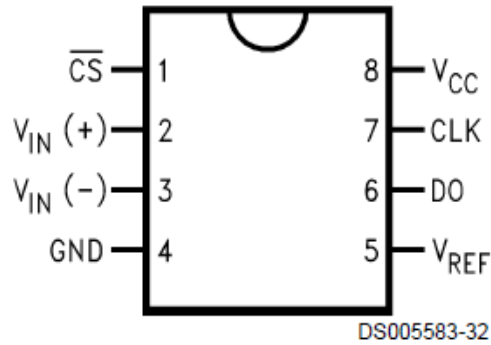
Γ.6. Αναλογικός σε Ψηφιακό Μετατροπέα (Analog to Digital Converter) ADC0831 [7]

Χαρακτηριστικά του ADC0831:

- Συμβατός με τα πρωτόκολλα επικοινωνίας της NSC MICROWIRE™ και άμεση σύνδεση με μικροεπεξεργαστές της οικογένειας COPS™.
- Εύκολη επικοινωνία με όλους τους μικροελεγκτές ή λειτουργεί αυτόνομα.
- Λειτουργεί αναλογομετρικά ή με τάση αναφοράς τα 5VDC .
- Με ρυθμιστή τάσης μπορεί να λειτουργήσει με μεγάλες πηγές τάσης.
- 0 έως 5V εύρος τάσης εισόδου με μονή πηγή τάσης 5VDC.
- Δυνατότητα απομακρυσμένης λειτουργίας μέσω σειριακής ψηφιακής σύνδεσης.
- Συμβατός με TTL/MOS εισόδους/ εξόδους.
- Ανάλυση ψηφιακής εξόδου 8-bit.
- Συνολικό μη ρυθμισμένο λάθος $\pm 1/2$ LSB and ± 1 LSB.
- Τάση εισόδου 5 VDC.
- Χαμηλή κατανάλωση ισχύος 15 mW.
- Χρόνος αντίδρασης 32 μ s.

Στην εικόνα Γ.5. φαίνεται η ανάλυση των άκρων του ADC0831. Κάθε άκρο έχει αρίθμηση και ονομασία. Η αρίθμηση είναι για να μπορεί να γίνεται η σύνδεση του σωστά κατά την κατασκευή του κυκλώματος και η ονομασία δείχνει την λειτουργία του κάθε άκρου.

**ADC0831 Single
Differential Input
Dual-In-Line Package (N)**



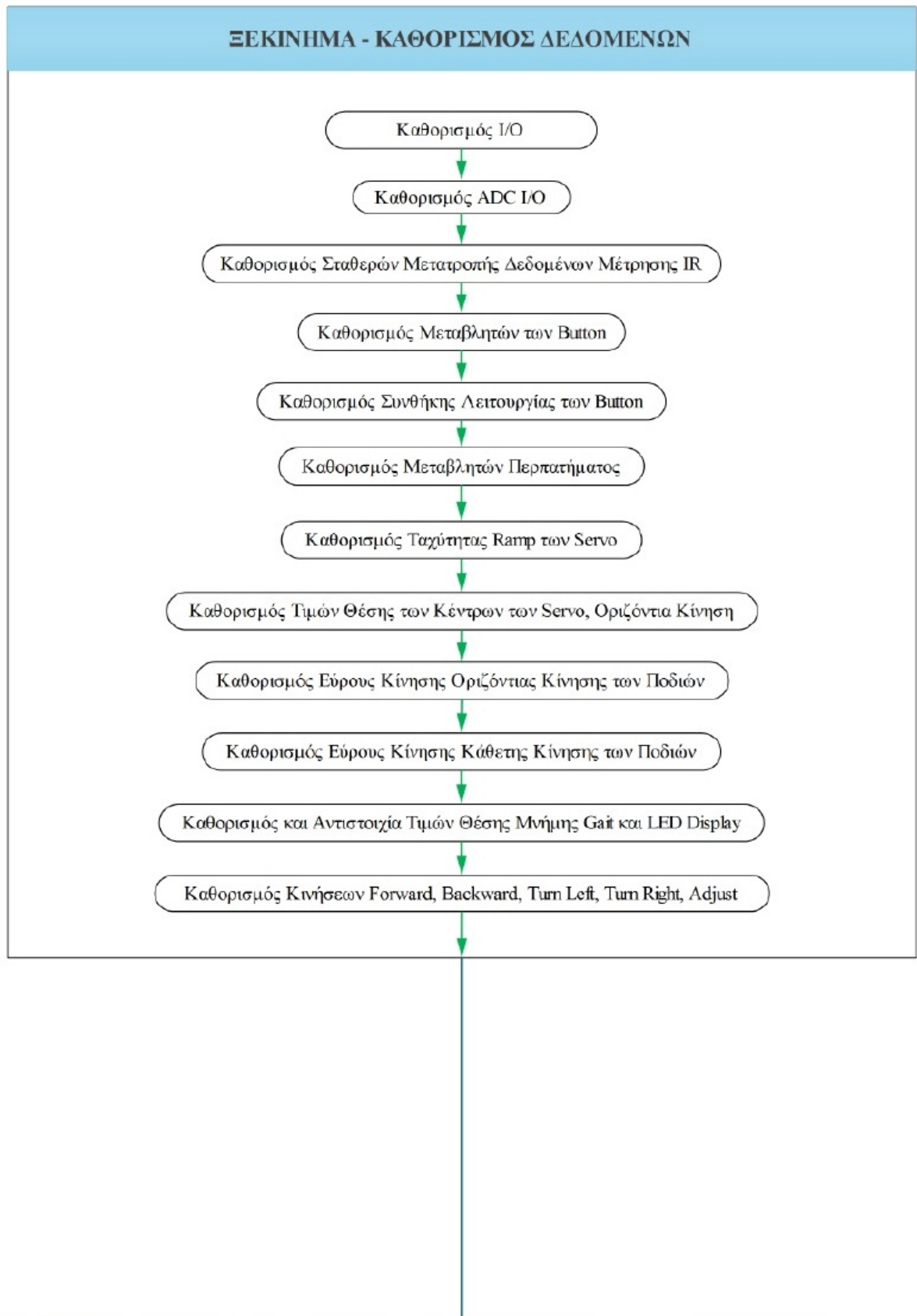
Top View

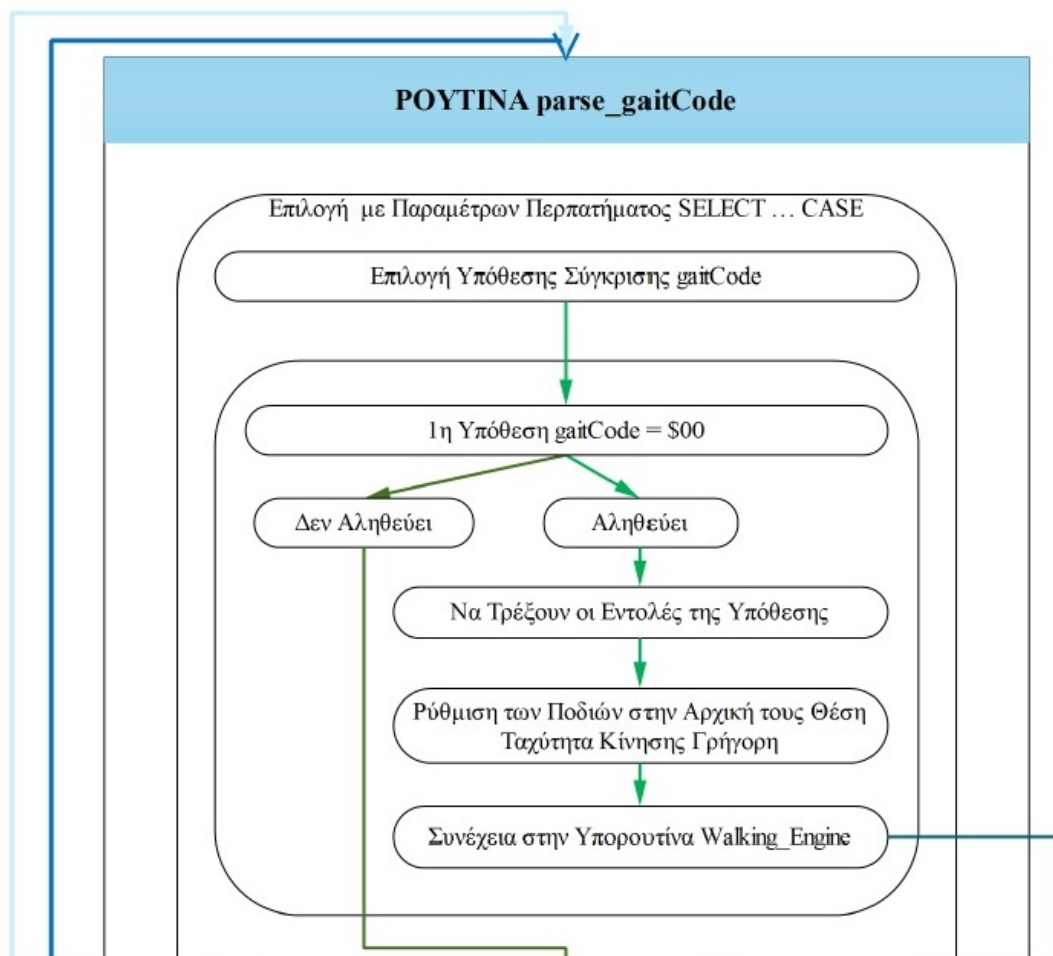
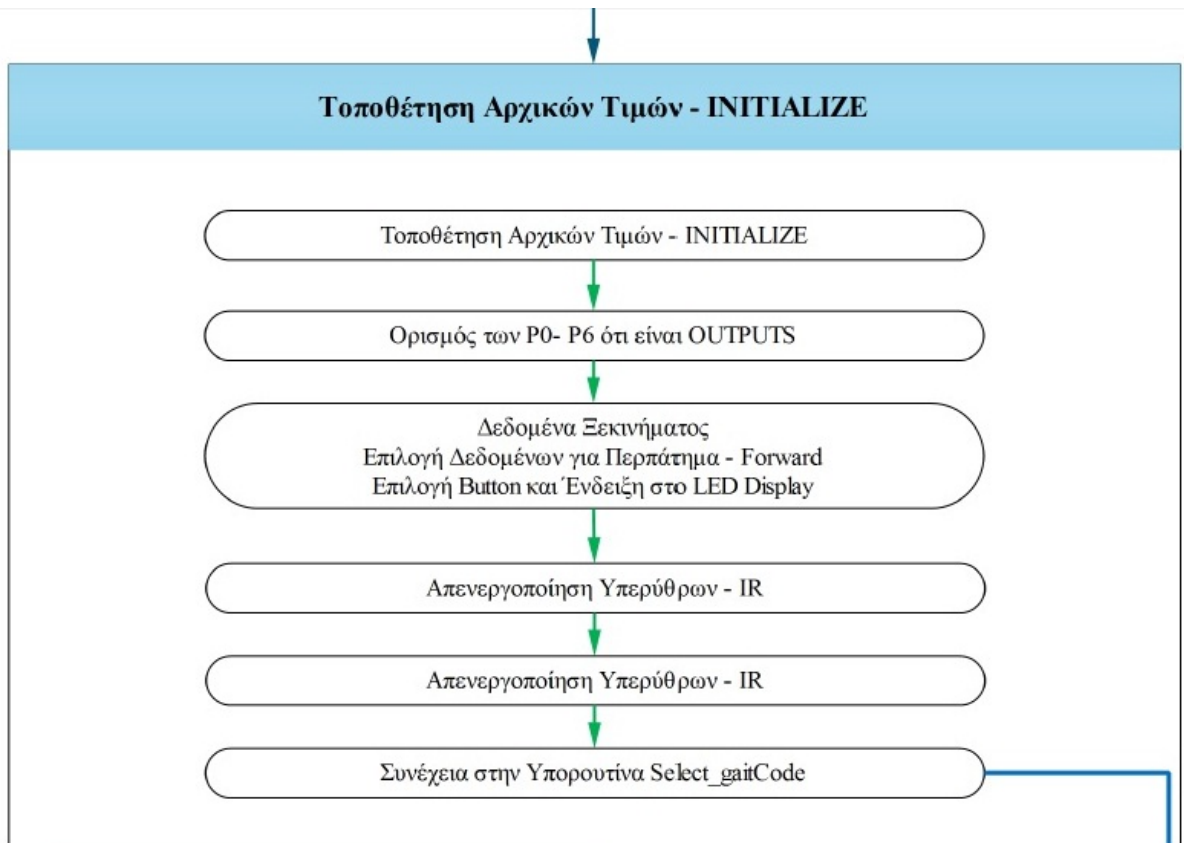
Εικόνα Γ.5. ADC0831

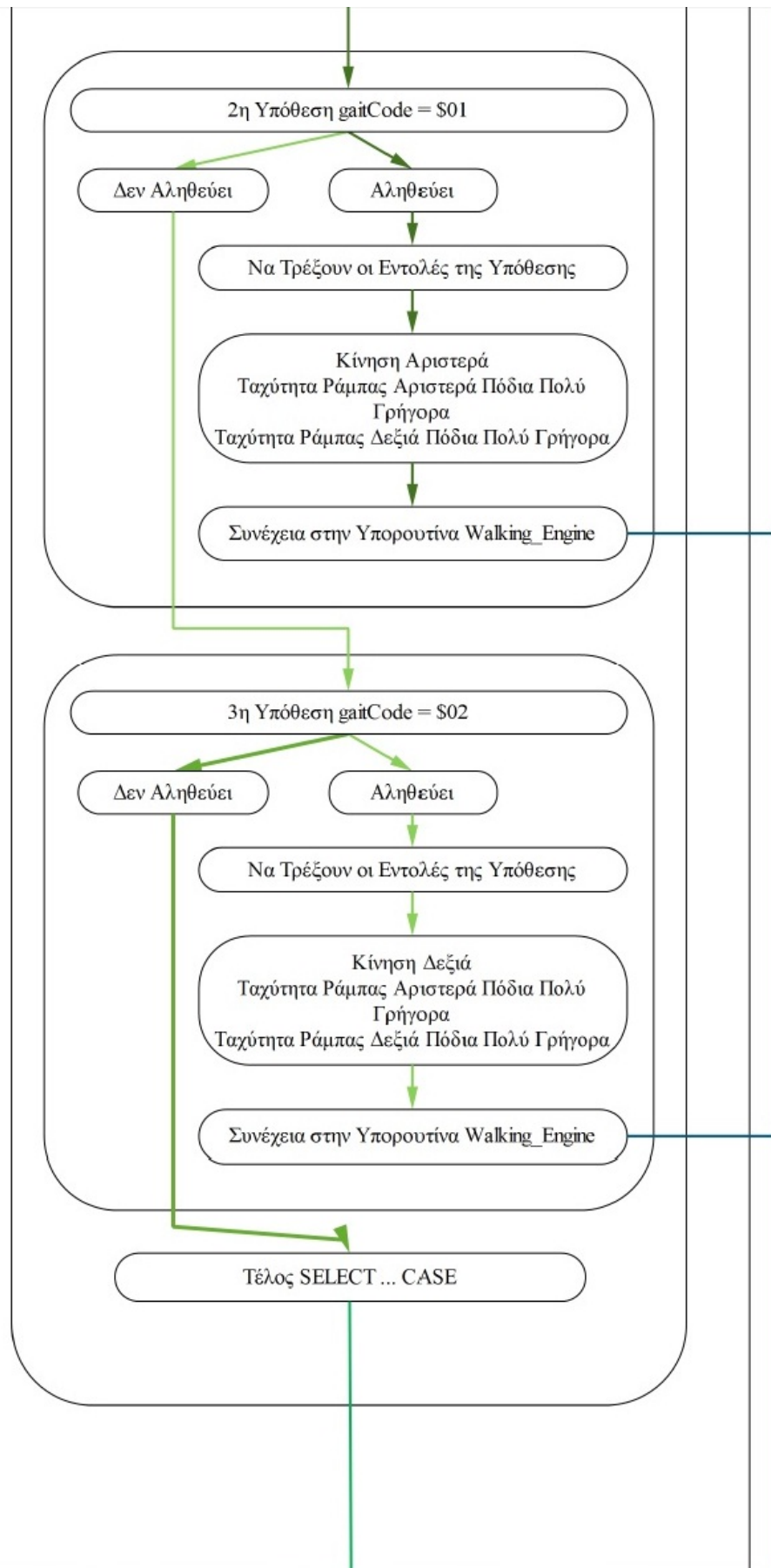
Οι ακροδέκτες του ADC0831 λειτουργούν ως εξής: $V_{in}(+)$ είναι η αναλογική είσοδος και D0 είναι η ψηφιακή έξοδος. V_{REF} και $V_{in}(-)$ χρησιμοποιούνται για να δοθεί μια συγκεκριμένη τάση σε σημεία του κυκλώματος και να ρυθμιστεί ή να βαθμονομηθεί το IC. Το άκρο V_{CC} και GND χρησιμοποιούνται για την τροφοδοσία του IC. Το \overline{CS} χρησιμοποιείται για θέτει το IC σε κατάσταση low και το CLK για να δέχεται την είσοδο του παλμού του χρονιστή.

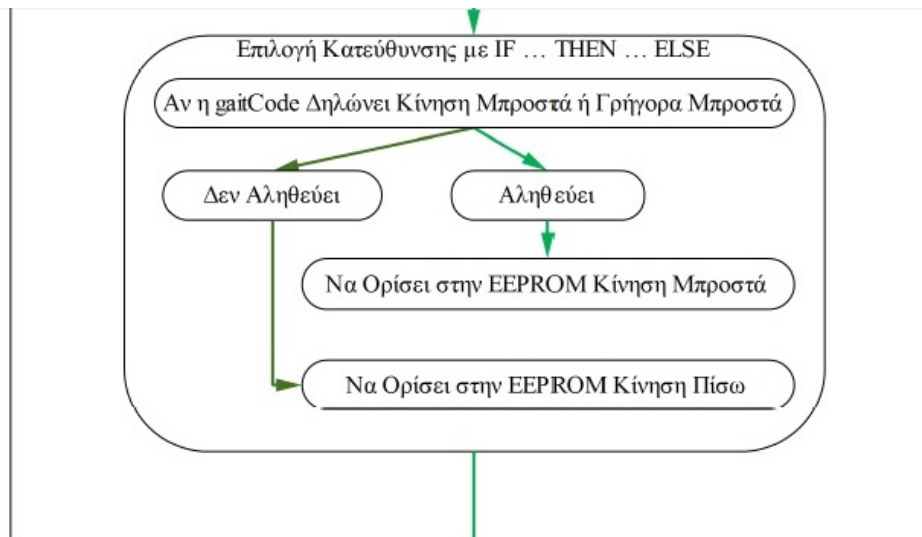
Όλα τα τεχνικά χαρακτηριστικά βρίσκονται αναλυτικά στο διαδίκτυο στις σελίδες του κατασκευαστή: <http://www.ti.com/sitesearch/docs/advancedsearch.jsp?searchTerm=adc0831>

Δ. Αναλυτικό Διάγραμμα Ροής του Προγράμματος

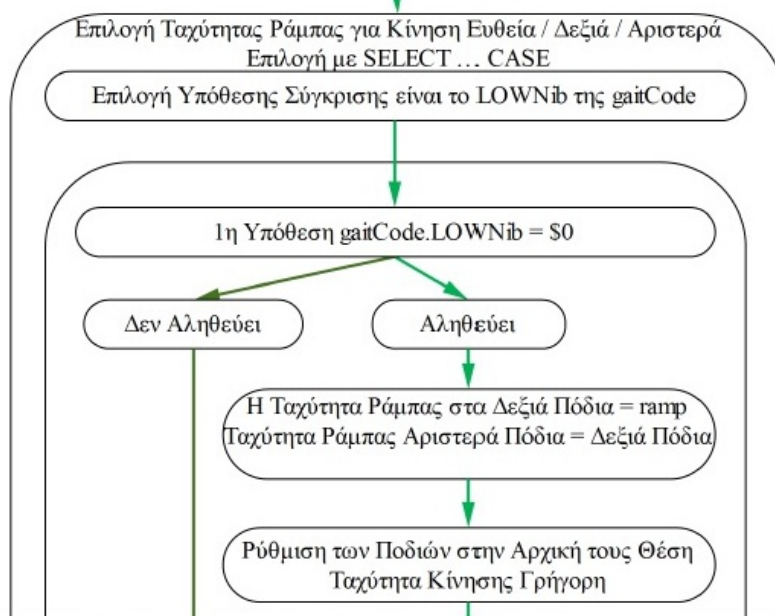
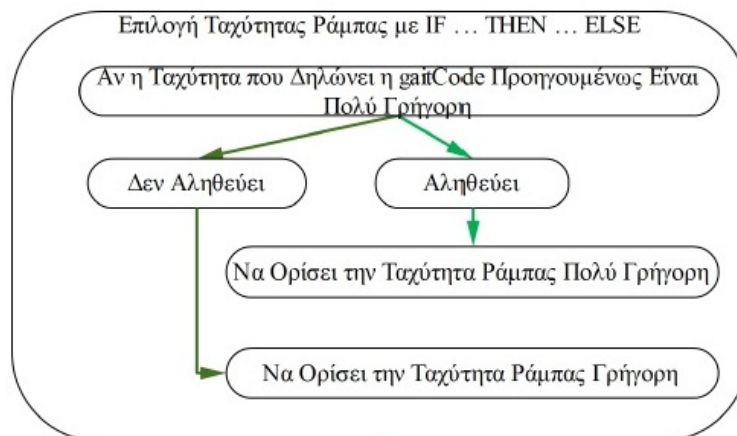


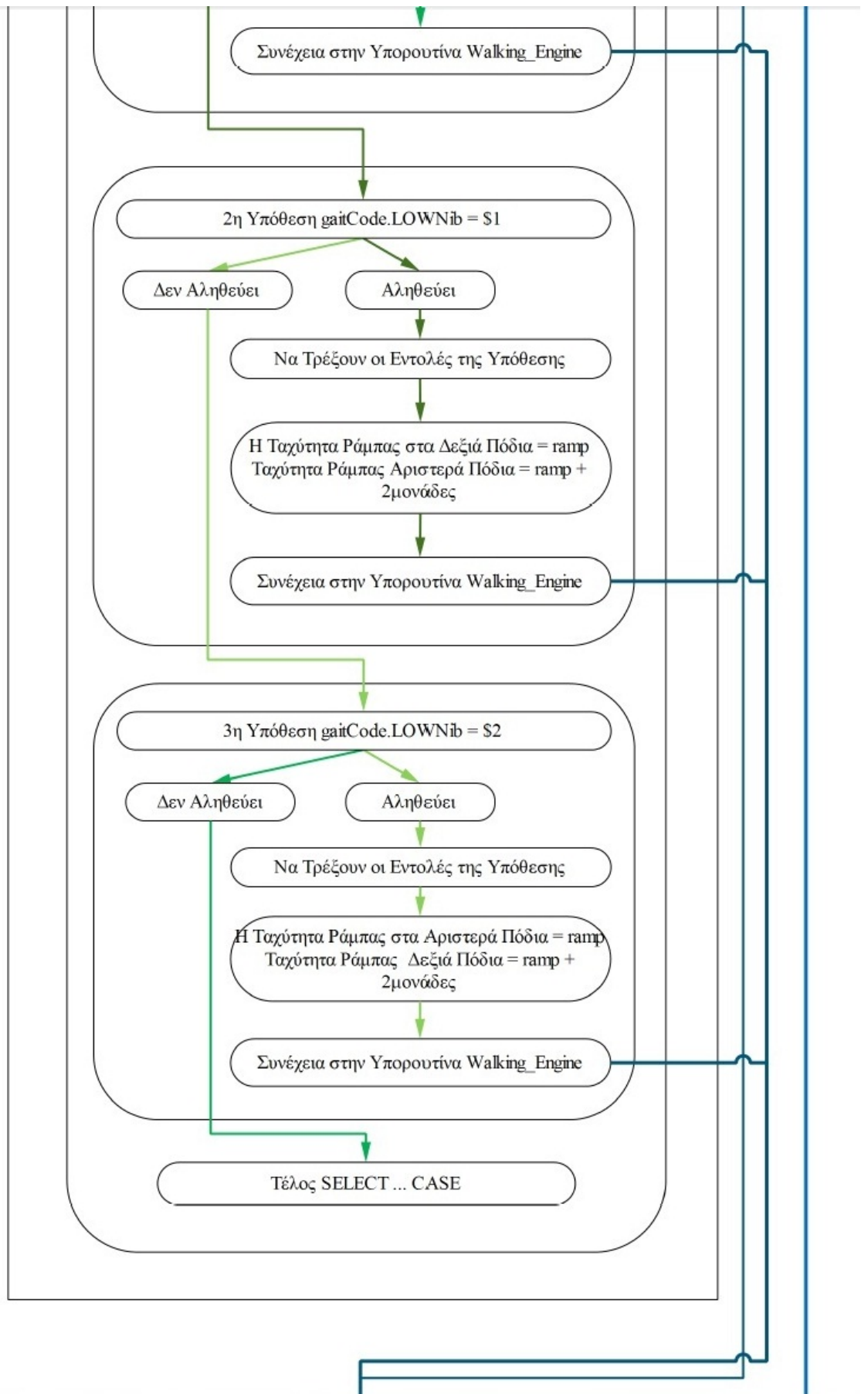






ΡΟΥΤΙΝΑ Setup_Ramp_Vars





POYTINA Walking_Engine

Διαβάζει και Στέλνει τα Δεδομένα των Κινήσεων στον PSC

Διαβάζει τα Δεδομένα Από τη Θέση Μνήμης ptrEEPROM και Δίνει την Τιμή της στην Μεταβλητή ServoAddr

Με την Χρήση της DO WHILE Στέλνει τα Δεδομένα της Κίνησης Ένα Ένα στον PSC, Για Όσο δεν Ισχύει η Συνθήκη

Η Τιμή του ServoAddr να Είναι Διαφορετική Από (\neq) \$FF

Δεν Αληθεύει

Αληθεύει

Χρήση της IF ... THEN ... ELSE για να Ξεχωρίσει σε Ποια Κίνηση Είναι

Αν η Κίνηση Είναι του Κάθετου Άξονα

Τότε

Η Ταχύτητα της Ράμπας να Είναι Η LiftRamp που Ορίστηκε προηγουμένως

Διαφορετικά

Αν είναι του Οριζόντιου Άξονα της Δεξιάς Πλευράς

Τότε

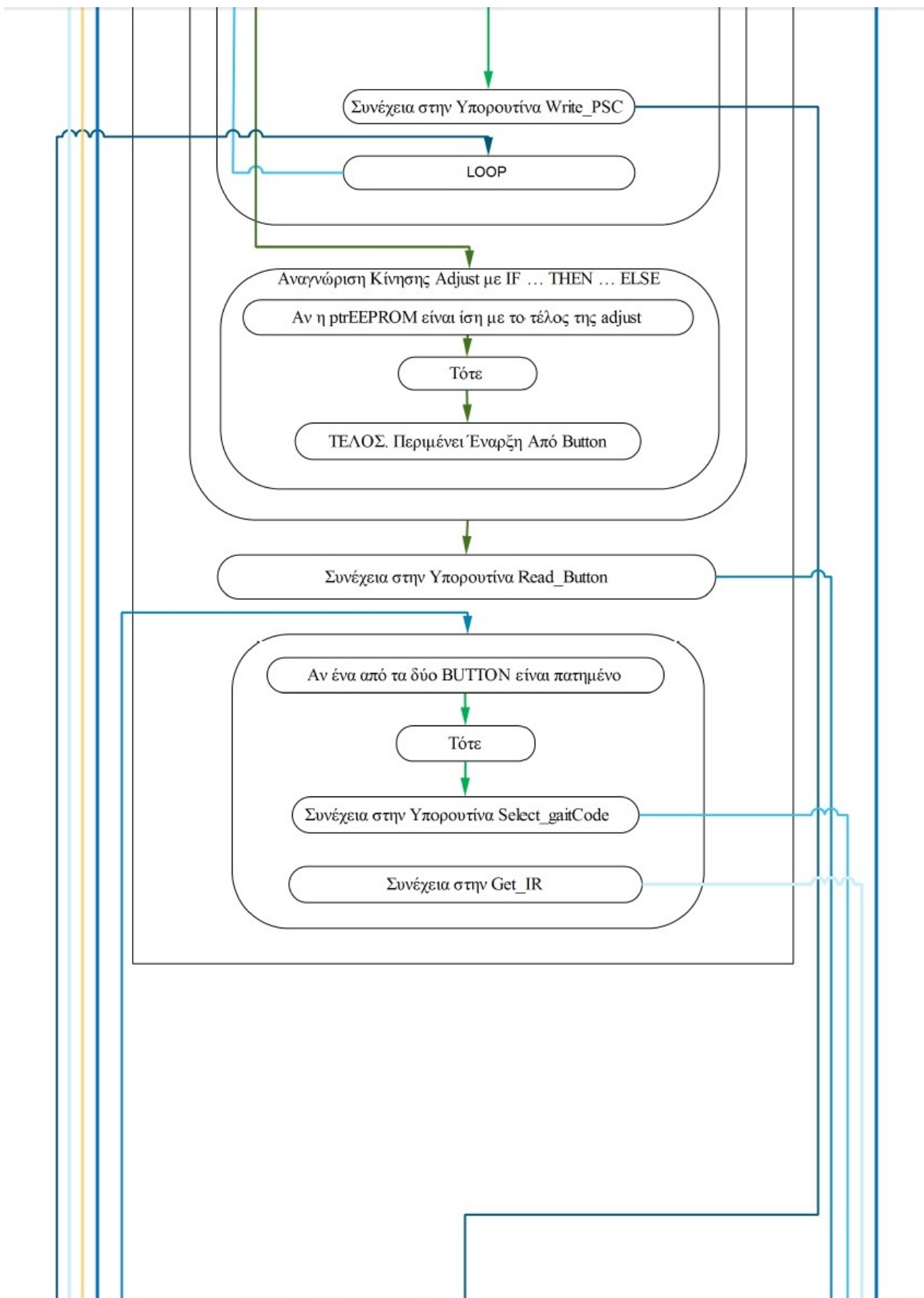
Η Ταχύτητα της Ράμπας να Είναι Η RightRamp

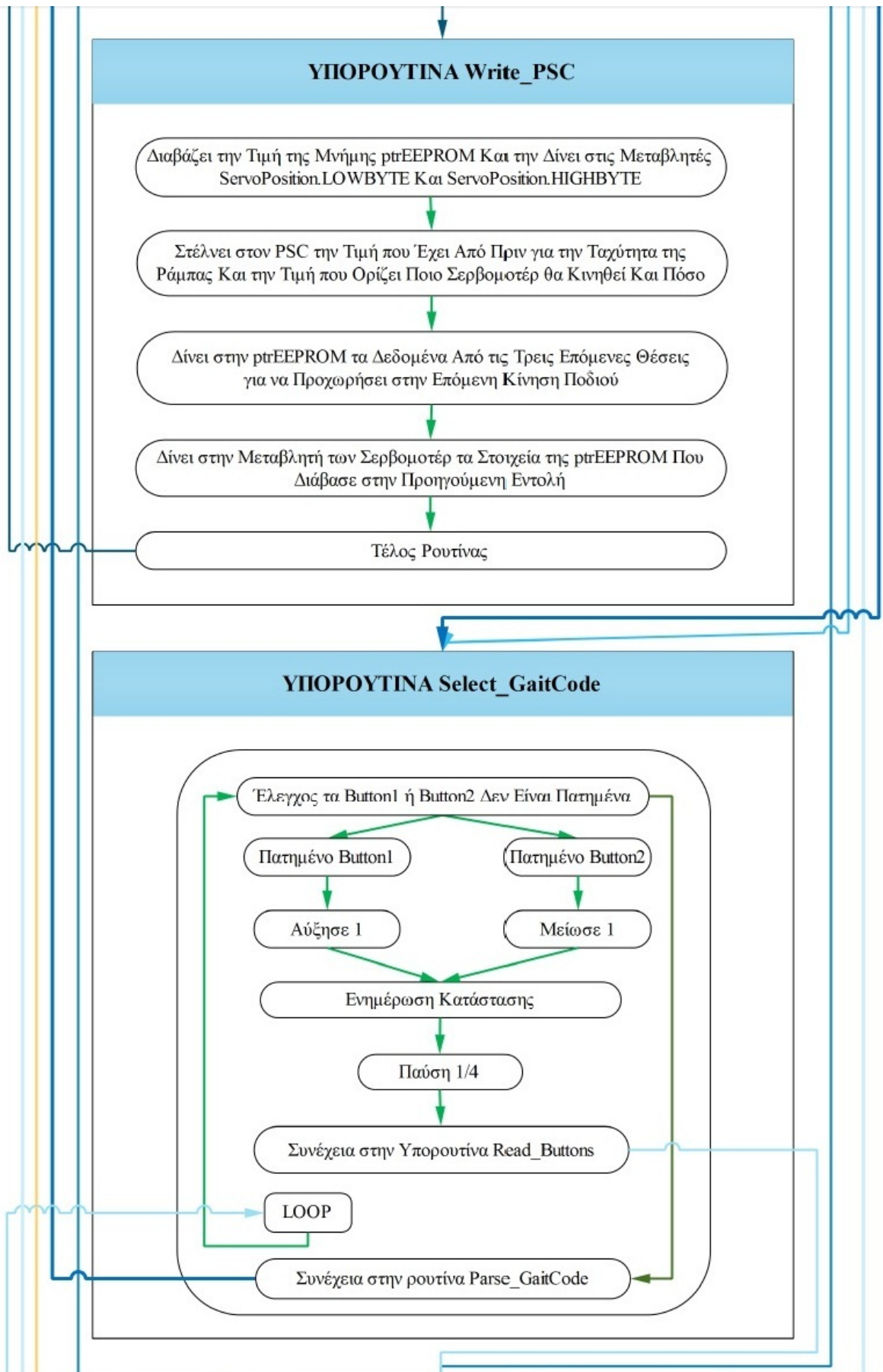
Διαφορετικά

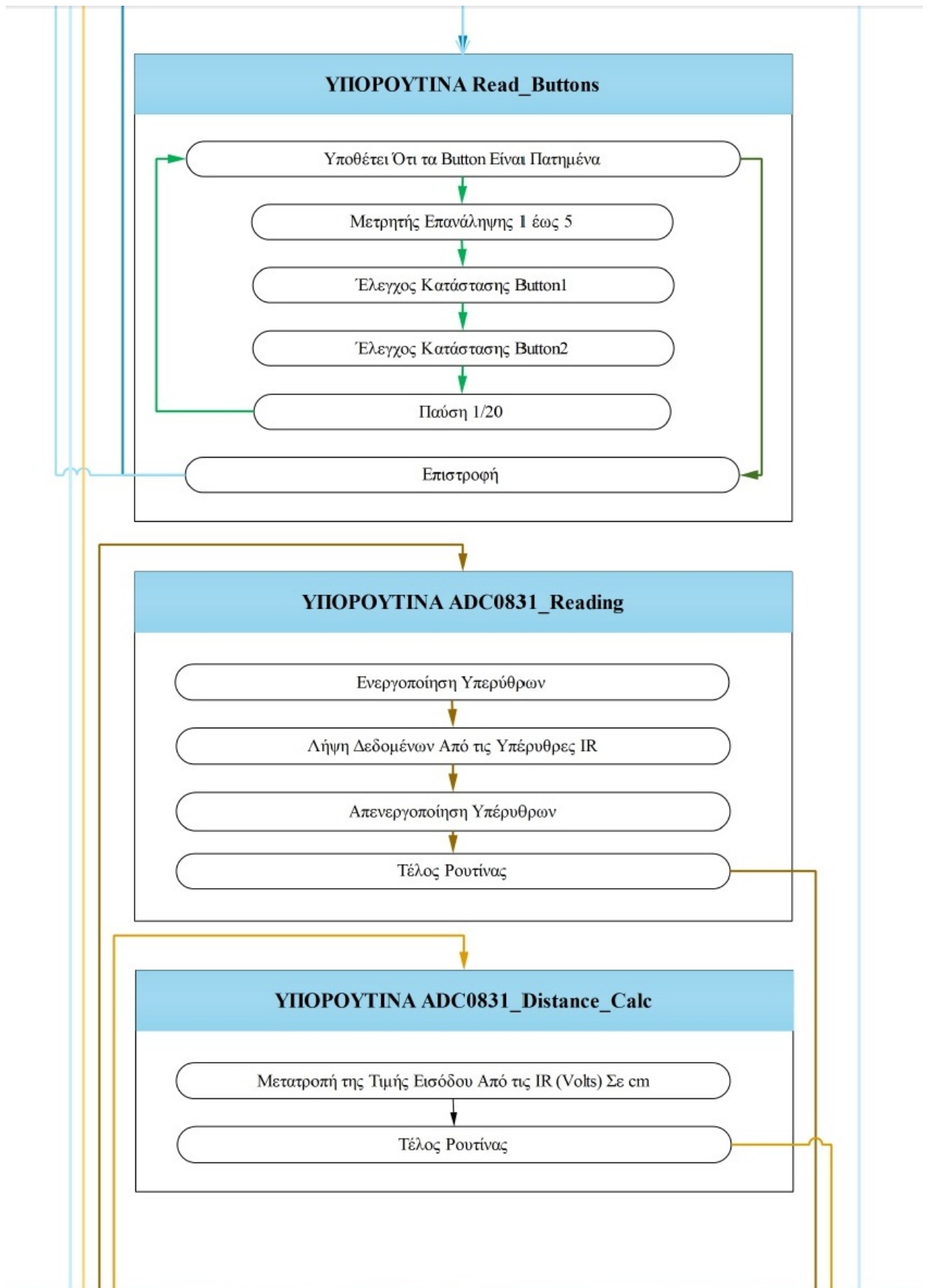
Αν είναι του Οριζόντιου Άξονα της Δεξιάς Πλευράς

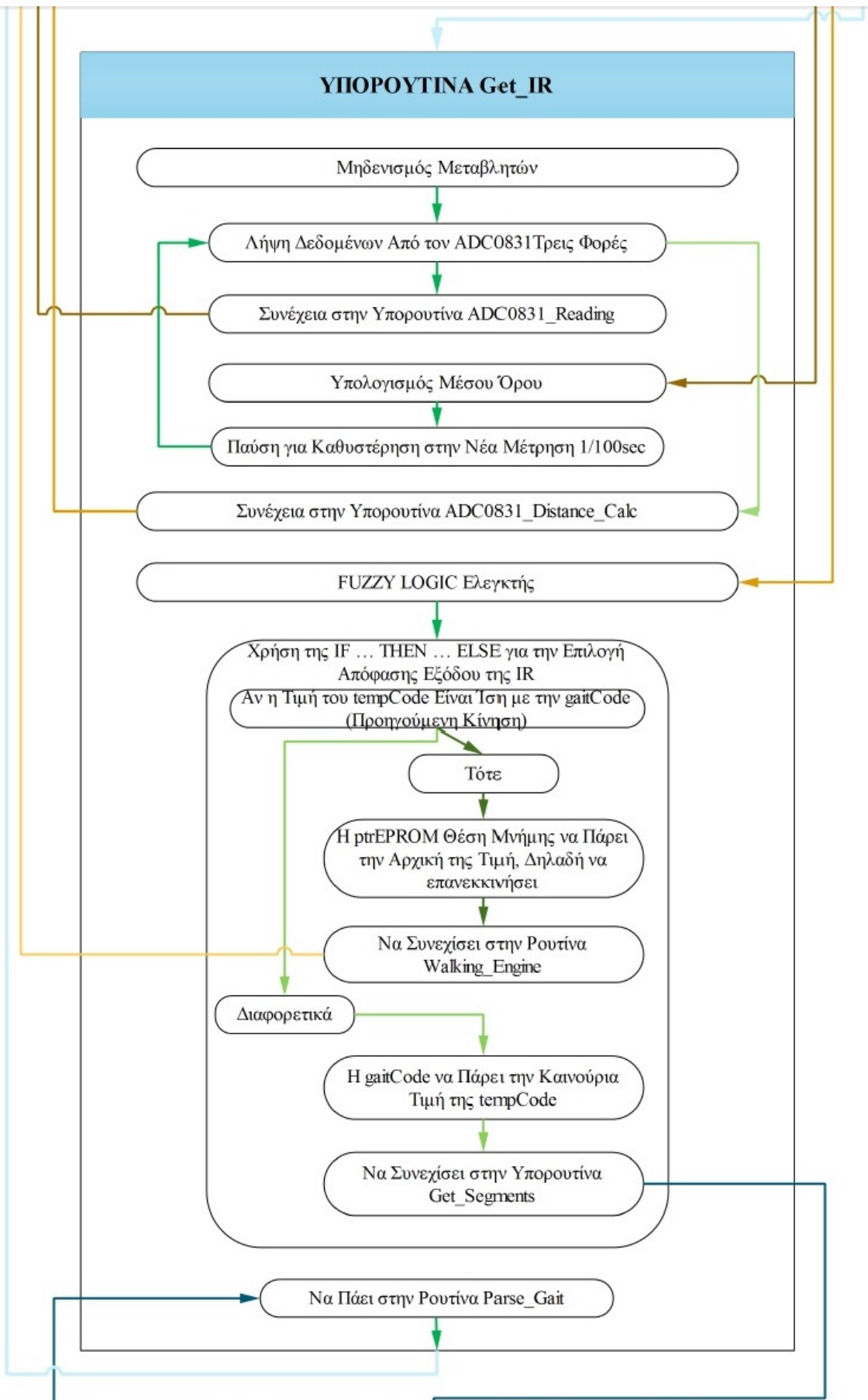
Τότε

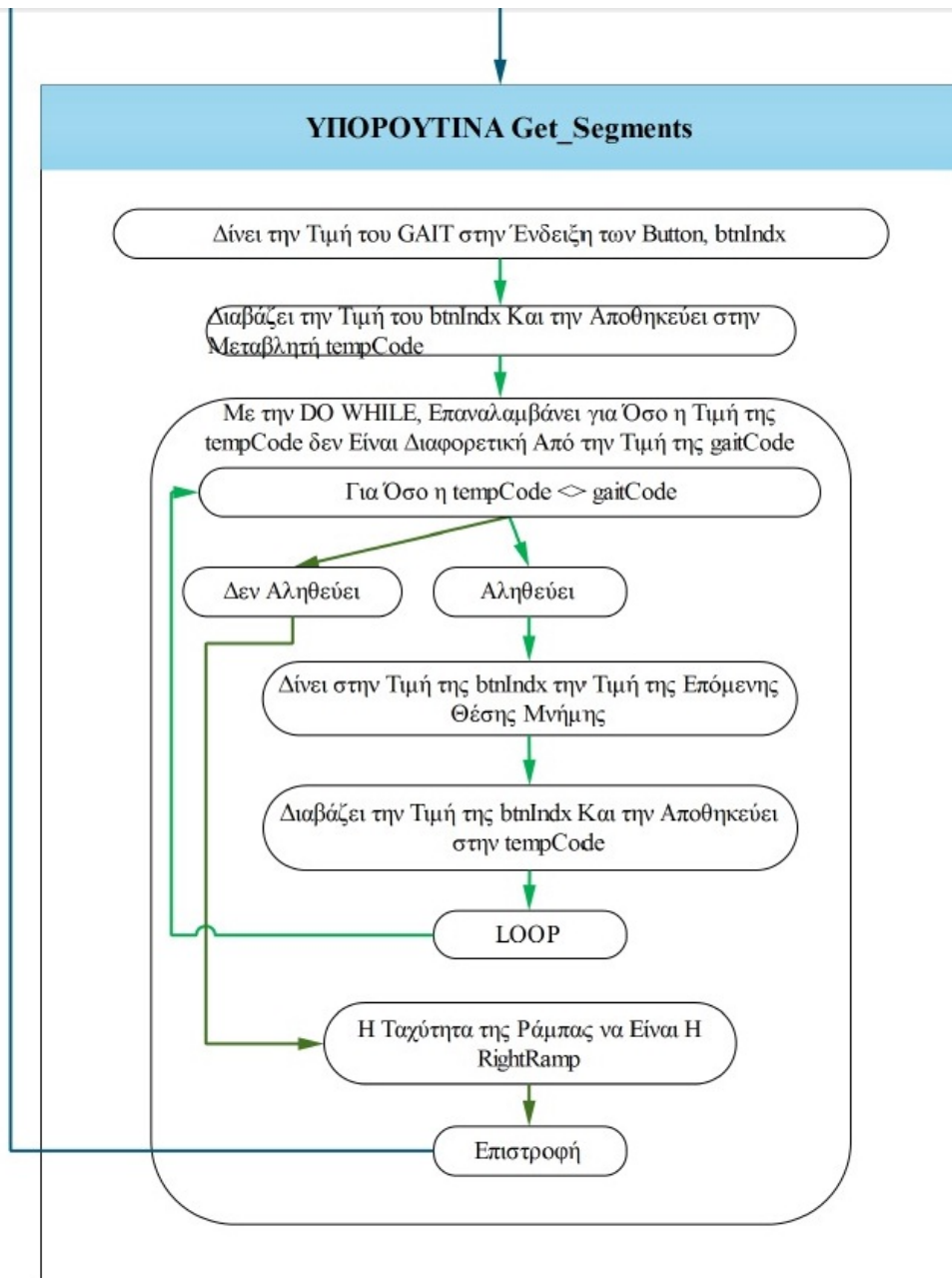
Η Ταχύτητα της Ράμπας να Είναι Η RightRamp











E. Πρόγραμμα Και Επεξηγήσεις Εντολών

Στο κεφάλαιο αυτό δίνεται ο κώδικας του προγράμματος του ρομποτικού με την επεξήγηση της χρήσης της κάθε εντολής μέσα στο πρόγραμμα.

'{\$\$STAMP BS2} ^^Δηλώνουμε στο πρόγραμμα ότι ο μικροελεγκτής είναι Basic Stamp 2

'{\$\$PBASIC 2.5} ^^Δηλώνουμε στο πρόγραμμα ότι ο Editor για το πρόγραμμα είναι η έκδοση v.2.5

^^Με την χρήση του συμβόλου (') δηλώνουμε ότι το κείμενο που ακολουθεί είναι σχόλια μέσα στο πρόγραμμα για την διευκόλυνση του χρήστη και όχι εντολές, έτσι ο μικροελεγκτής αγνοεί τις λέξεις που ακολουθούν μέχρι το τέλος της σειράς. Εδώ είναι γραμμένη η αντιστοιχία του αριθμού που φαίνεται στο πάνελ με τον αριθμό που είναι δηλωμένη η κίνηση.

' LED vs gaitCode table

' LED gaitCode

' (0) \$00 - Home

' (1) \$01 - Spin Left

' (2) \$02 - Spin Right

'

' LED gaitCode

LED gaitCode

' (3) \$10 - Forward Fast

(6) \$20 - Forward

' (4) \$11 - Fast Forward Left

(7) \$21 - Forward Left

' (5) \$12 - Fast Forward Right

(8) \$22 - Forward Right

'

' LED gaitCode

LED gaitCode

' (9) \$30 - Backward

(C) \$40 - Fast Backward

' (A) \$31 - Backward Left

(d) \$41 - Fast Backward Left

' (b) \$32 - Backward Right

(E) \$42 - Fast Backward Right

'

(F) Open

^^Στο ξεκίνημα του προγράμματος δηλώνουμε τις σταθερές, μεταβλητές και διάφορα στοιχεία που θα

χρησιμοποιηθούν στη συνέχεια του προγράμματος σε κάποιες θέσεις μνήμης, καθώς και την λειτουργία των I/O του μικροελεγκτή.

'-----[I/O Definitions]-----			
PSC	PIN	15	^^Δηλώνει ότι το PIN15 είναι η σειριακή έξοδος για το PSC module
ModeBtn	PIN	8	^^Δηλώνει ότι το pin8 θα το ονομάσει στο πρόγραμμα ModeBtn και θα το χρησιμοποιεί για να επιλέγει την επόμενη κίνηση, select robot mode up
StartBtn	PIN	12	^^Δηλώνει ότι το pin12 θα το ονομάσει στο πρόγραμμα StartBtn και θα το χρησιμοποιεί για να επιλέγει την προηγούμενη κίνηση, robot mode down
segments	VAR	OUTL	^^Δηλώνει εξόδους τα pins 0 – 7 . Η μεταβλητή “OUTL” δείχνει ότι επιλέγουμε τα pin “Low” δηλαδή από τα 16 τα πρώτα 8, και τα θέτει ταυτόχρονα ως εξόδους
Baud	CON	33164	^^Δηλώνει το baud επικοινωνίας είναι στα 2400

'-----[ADC0831 I/O]-----			
FrontIR	CON	9	^^Δηλώνει ότι το pin10 θα το ονομάζει FrontIR και θα το χρησιμοποιούμε για την ενεργοποίηση/απενεργοποίηση του ADC0831
DIO_10	CON	10	^^Δηλώνει ότι το pin 10 θα χρησιμοποιείται για την εισαγωγή δεδομένων από τις υπέρυθρες
CLK	CON	11	^^Δηλώνει ότι το pin 11 θα χρησιμοποιείται για σαν για να γεννήτρια παλμών Clock, και θα χρησιμοποιηθεί χρονίζει τον ADC0831 κατά την λήψη δεδομένων από τον ελεγκτή. Αυτόματα θέτει το pin 11 ως έξοδο.

'-----[Transform Function Constants]-----

^^Εδώ ορίζουμε τις σταθερές που θα χρησιμοποιηθούν για την μετατροπή της τάσης από τις υπέρυθρες σε φυσικό μέγεθος δηλαδή εκατοστά. Τα νούμερα έχουν βρεθεί πειραματικά από οδηγίες από την ιστοσελίδα:

<http://www.acroname.com/robotics/info/articles/irlinear/irlinear.html>

και υπάρχει η ανάλυση παραπάνω

Slope	CON	3200
y_Intercept	CON	3
k	CON	4

'----- [Button Variables]-----

			^^Δηλώνει τις μεταβλητές που θα χρησιμοποιηθούν για τις ρουτίνες των Button
btns	VAR	Nib	^^Δηλώνει την μεταβλητή του button holder με μέγεθος 0-15
btn1	VAR	btns.BIT0	^^Δηλώνει την μεταβλητή του debounced button ότι θα είναι ίση με το πρώτο ψηφίο της btns
btn2	VAR	btns.BIT1	^^Δηλώνει την μεταβλητή του debounced button ότι θα είναι ίση με το δεύτερο ψηφίο της btns
idx	VAR	Nib	^^Δηλώνει το digit index θα έχει μέγεθος 0-15
btnIndx	VAR	Byte	^^Δηλώνει ότι η θέση μνήμης btnIndx που δείχνει το LED panel θα έχει μέγεθος 0-255
'---- [Button states] ----- -----			
Pressed	CON	1	^^Δηλώνει τις καταστάσεις που θα χρησιμοποιηθούν για τα Button
NotPressed	CON	0	
'---- [Walking Variables] ----- -----			
			^^Δηλώνει τις μεταβλητές που θα χρησιμοποιηθούν για τις ρουτίνες του περπατήματος
counter	VAR	Nib	^^Δηλώνει τον μετρητή για το Loop που δίνει τον μέσο όρο των μετρήσεων του ADC0831 ότι θα είναι μεγέθους Nib 0-15
servoAddr	VAR	Byte	^^Δηλώνει την μεταβλητή Servo addresses που οδηγεί τα σερβομοτέρ μεγέθους Byte 0-255
gaitCode	VAR	Byte	^^Δηλώνει την μεταβλητή gaitCode που δείχνει τον τρόπο περπατήματος την κάθε στιγμή με μέγεθος Byte 0-255
tempCode	VAR	Byte	^^Δηλώνει την μεταβλητή Temp gaitcode που την χρησιμοποιεί προσωρινά μέχρι για γίνουν υπολογισμοί για την επόμενη κίνηση, με μέγεθος Byte 0-255
ramp	VAR	Byte	^^Δηλώνει την Ramp που χρησιμοποιείται από την εντολή SEROUT για την ταχύτητα κίνησης των σερβομοτέρ με μέγεθος Byte 0-255
rightRamp	VAR	Byte	^^Δηλώνει την rightRamp που θα χρησιμοποιηθεί για να ορίζεται η ταχύτητα των δεξιών σερβομοτέρ με μέγεθος Byte 0-255
leftRamp	VAR	Byte	^^Δηλώνει την leftRamp που θα χρησιμοποιηθεί για να ορίζεται η ταχύτητα των δεξιών σερβομοτέρ με μέγεθος Byte 0-255
distTemp	VAR	Byte	^^Δηλώνει την μεταβλητή distTemp που θα χρησιμοποιηθεί για τους υπολογισμούς της απόστασης

				από τον ADC0831, με μέγεθος Byte 0-255
distance	VAR	Byte		^^Δηλώνει την μεταβλητή distance που θα χρησιμοποιηθεί για τους υπολογισμούς της απόστασης από τον ADC0831, με μέγεθος Byte 0-255
ptrEEPROM	VAR	Word		^^Δηλώνει την μεταβλητή που θα χρησιμοποιείται για την επιλογή του περπατήματος, με μέγεθος Word 0-65535
servoPosition	VAR	Word		^^Δηλώνει την μεταβλητή που θα χρησιμοποιείται για την θέση των σερβομοτέρ, με μέγεθος Word 0-65535
'----- [Adjustable Ramp Values] -----				

				^^Η ταχύτητα της ράμπας (ramp) των σερβομοτέρ καθορίζεται από ένα εύρος τιμών από \$1(1) την πιο γρήγορη έως \$3F(63) την πιο αργή.
LiftRamp	CON	\$1		^^Δηλώνει την σταθερά που ορίζει την ταχύτητα της ράμπας στην κάθετη κίνηση
VeryFast	CON	\$8		^^Δηλώνει την σταθερά που ορίζει την ταχύτητα της ράμπας στην οριζόντια κίνηση για πολύ γρήγορο περπάτημα
Fast	CON	\$A		^^Δηλώνει την σταθερά που ορίζει την ταχύτητα της ράμπας στην οριζόντια κίνηση για γρήγορο περπάτημα
'-----[Horizontal Leg Constants]-----				

				^^Δηλώνει την θέση των κέντρων των σερβομοτέρ στην οριζόντια κίνηση. Το νούμερο το βρίσκουμε με πειραματικό τρόπο και είναι διαφορετικό για κάθε σερβομοτέρ. Τα κέντρα τους, δηλώνονται ξεχωριστά σε αυτό εδώ το τμήμα του προγράμματος, για να είναι πιο εύκολη η ρύθμισή τους και όχι παρακάτω όπως θα μπορούσε να γίνει.
'Servo center				
Center1	CON	760		^^Δηλώνει την σταθερά του κέντρου του μπροστά δεξιού ποδιού
Center2	CON	815		^^Δηλώνει την σταθερά του κέντρου του πίσω δεξιού ποδιού
Center3	CON	755		^^Δηλώνει την σταθερά του κέντρου του μπροστά αριστερού ποδιού
Center4	CON	770		^^Δηλώνει την σταθερά του κέντρου του πίσω αριστερού ποδιού
'-----[Stride Calculations]-----				

			^^Υπολογισμός και δήλωση των σταθερών των βημάτων των ποδιών που θα χρησιμοποιηθούν κατά την εκτέλεση του βασικού προγράμματος
Stride	CON	150	^^Δηλώνει την σταθερά που θα ορίζει το εύρος της κίνησης των σερβομοτέρ, δηλαδή το βήμα του ποδιού
delay	CON	Stride/5	^^Δηλώνει την σταθερά Delay που θα χρησιμοποιηθεί σαν μία καθυστέρηση στην κίνηση
Leg1Center	CON	Center1	^^Δηλώνει την σταθερά του κέντρου του ποδιού 1
Leg1Forward	CON	Center1-Stride	^^Δηλώνει την σταθερά της κίνησης του ποδιού 1 μπροστά
Leg1Back	CON	Center1+Stride	^^Δηλώνει την σταθερά της κίνησης του ποδιού 1 πίσω
Leg2Center	CON	Center2	^^Αντίστοιχα για το πόδι 2
Leg2Forward	CON	Center2+Stride	
Leg2Back	CON	Center2-Stride	
Leg3Center	CON	Center3	^^Αντίστοιχα για το πόδι 3
Leg3Forward	CON	Center3-Stride	
Leg3Back	CON	Center3+Stride	
Leg4Center	CON	Center4	^^Αντίστοιχα για το πόδι 4
Leg4Forward	CON	Center4-Stride	
Leg4Back	CON	Center4+Stride	
'----- [Adjustable vertical servo positions] -----			
-----			^^Δηλώνει την θέση της κίνησης των σερβομοτέρ στην κάθετη κίνηση. Μία σταθερά για την κίνηση που ανεβάζει το πόδι και μία για το κατέβασμα. Το νούμερο το βρίσκουμε με πειραματικό τρόπο και είναι διαφορετικό για κάθε σερβομοτέρ. Την διαδικασία αυτή για την οριζόντια και την κάθετη κίνηση την κάνουμε για να υπάρχει μία οικονομία εντολών στο πρόγραμμα. Οι σταθερές θα χρησιμοποιηθούν παρακάτω στον προγραμματισμό των βημάτων.
RaiseRightF	CON	725	
LowerRightF	CON	1220	
RaiseRightR	CON	1150	
LowerRightR	CON	700	
RaiseLeftF	CON	430	
LowerLeftF	CON	1000	
RaiseLeftR	CON	460	
LowerLeftR	CON	1030	

'---- [7 Segment LED and gaitCode EEPROM Data] -----

^^Δηλώνει στην μνήμη την αντιστοιχία των βημάτων με την ένδειξη στο LED panel

' Hex #	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
LED DATA	\$7E,\$18,\$6D,\$3D,\$1B,\$37,\$77,\$1C,\$7F,\$1F,\$5F,\$73,\$66,\$79,\$67														
Gait DATA	\$00,\$01,\$02,\$10,\$11,\$12,\$20,\$21,\$22,\$30,\$31,\$32,\$40,\$41,\$42														

'---- [Servo Address and Position EEPROM data] -----

^^Δηλώνει την ακολουθία των κινήσεων που θα γίνονται κάθε φορά για να ολοκληρωθεί ένα βήμα. Τα ονόματα των κινήσεων έχουν οριστεί παραπάνω στο πρόγραμμα. Στο κεφάλαιο 3.2.2. Προγραμματισμός περπατήματος, υπάρχει η ανάλυση και η σειρά των βημάτων.

Forward DATA	\$01, Word RaiseRightF,	\$07, Word RaiseLeftR,	^^Περπάτημα μπροστά
	\$00, Word Leg1Forward,	\$02, Word Leg2Back,	
	\$04, Word Leg3Center,	\$06, Word Leg4Center,	
	\$01, Word LowerRightF,	\$07, Word LowerLeftR,	
	\$03, Word RaiseRightR,	\$05, Word RaiseLeftF,	
	\$00, Word Leg1Center,	\$02, Word Leg2Center,	
	\$04, Word Leg3Forward,	\$06, Word Leg4Back,	
	\$03, Word LowerRightR,	\$05, Word LowerLeftF,	
	\$FF 'end of forward		

' Walk Backward

Back DATA	\$01, Word RaiseRightF,	\$07, Word RaiseLeftR,	^^Περπάτημα Πίσω
	\$00, Word Leg1Center,	\$02, Word Leg2Center,	
	\$04, Word Leg3Forward	\$06, Word Leg4Back,	
	\$01, Word LowerRightF,	\$07, Word LowerLeftR,	
	\$03, Word RaiseRightR,	\$05, Word RaiseLeftF,	
	\$00, Word Leg1Forward,	\$02, Word Leg2Back,	
	\$04, Word Leg3Center,	\$06, Word Leg4Center,	
	\$03, Word LowerRightR,	\$05, Word LowerLeftF,	
	\$FF 'end of back		

' Left Turn EEPROM values

LTurn DATA	\$01, Word RaiseRightF,	\$07, Word RaiseLeftR,	^^Στροφή αριστερά
------------	-------------------------	------------------------	-------------------

```

$00,Word Leg1Forward+Stride, $06,Word Leg4Back+Stride,
$04,Word Leg3Forward, $02,Word Leg2Back,
$01,Word LowerRightF, $07,Word LowerLeftR,
$03,Word RaiseRightR, $05,Word RaiseLeftF,
$00,Word Leg1Center+Stride, $06,Word Leg4Center+Stride,
$04,Word Leg3Forward, $02,Word Leg2Back,
$03,Word LowerRightR, $05,Word LowerLeftF,
$FF ' end of left turn

```

' Right turn EEPROM Values

```

RTurn DATA $01,Word RaiseRightF, $07,Word RaiseLeftR, ^^Στροφή δεξιά
$00,Word Leg1Forward, $02,Word Leg2Center-Stride,
$04,Word Leg3Center-Stride, $06,Word Leg4Back,
$01,Word LowerRightF, $07,Word LowerLeftR,
$03,Word RaiseRightR, $05,Word RaiseLeftF,
$00,Word Leg1Forward, $02,Word Leg2Back-Stride,
$04,Word Leg3Forward-Stride, $06,Word Leg4Back,
$03,Word LowerRightR, $05,Word LowerLeftF,
$FF ' end of right turn

```

^^Τοποθέτηση των ποδιών στο κέντρο

```

'
          Raise          Center          Lower
Adjust DATA $01,Word RaiseRightF, $00,Word Center1, $01,Word LowerRightF
          DATA $03,Word RaiseRightR, $02,Word Center2, $03,Word LowerRightR
          DATA $05,Word RaiseLeftF, $04,Word Center3, $05,Word LowerLeftF
          DATA $07,Word RaiseLeftR, $06,Word Center4, $07,Word LowerLeftR,
$FF ' end of adjust legs

```

'---- [End EEPROM Data] -----

'---- [Initialize] -----

DIRL = %01111111

^^Ξεκίνημα βασικού προγράμματος

^^Ορίζει τα PIN P0 - P6 εξόδους

<code>gaitCode = \$10</code>	^^Δηλώνει στην μεταβλητή <code>gaitCode</code> την τιμή \$10 που αντιστοιχεί σε περπάτημα ευθεία μπροστά
<code>btnIndx = \$03</code>	^^Βάζει αρχική τιμή στο <code>button</code> την \$03 για να ξεκινήσει με περπάτημα ευθεία μπροστά
<code>segments = \$3D</code>	^^Βάζει αρχική τιμή στο LED πάνελ την \$3D για να εμφανίσει το 3 δηλαδή περπάτημα ευθεία μπροστά
<code>HIGH FrontIR</code>	^^Απενεργοποιεί τον ADC0831 μετατροπέα
<code>GOTO Select_GaitCode</code>	^^Πηγαίνει στην ρουτίνα <code>Select_GaitCode</code>

'---- [Gait Selection] -----

<code>Parse_GaitCode:</code>	^^Ρουτίνα περπατήματος
	^^Προσδιορισμός παραμέτρων περπατήματος
<code>SELECT gaitCode</code>	^^Χρήση της εντολής <code>SELECT...CASE</code> . Επιλογή της έκφρασης <code>gaitCode</code> για να την συγκρίνει με τις υποθέσεις <code>CASE</code> . Αν η υπόθεση είναι ίδια με την επιλεγμένη παράμετρο το πρόγραμμα εκτελεί τις συνθήκες που περιέχονται μέσα στην υπόθεση και αγνοεί τις υπόλοιπες. Αν δεν είναι συνεχίζει στην επόμενη υπόθεση.
<code>CASE \$00</code>	^^Αν <code>gaitCode = \$00</code> το πρόγραμμα θα συνεχίσει στις επόμενες συνθήκες, διαφορετικά θα συνεχίσει στην επόμενη υπόθεση <code>CASE</code>
<code>ptrEEPROM = Adjust</code>	^^ ορίζει το περιεχόμενο της θέσης μνήμης <code>ptrEEPROM</code> ίσο με <code>Adjust</code> , δηλαδή τρέχει το περιεχόμενο <code>adjust</code> που είδαμε παραπάνω και βάζει τα πόδια στην αρχική τους θέση
<code>rightRamp = Fast</code>	^^Δίνει την τιμή της θέσης μνήμης <code>fast</code> στη κίνηση των δεξιών ποδιών για να ρυθμίσει την ταχύτητα κίνησης τους
<code>leftRamp = Fast</code>	^^Δίνει την τιμή της θέσης μνήμης <code>fast</code> στη κίνηση των αριστερών ποδιών για να ρυθμίσει την ταχύτητα κίνησης τους
<code>GOTO Walking_Engine</code>	^^Το πρόγραμμα να συνεχιστεί στην υπορουτίνα <code>Walking_Engine</code>
<code>CASE \$01</code>	^^Αν δεν ισχύει η προηγούμενη υπόθεση, και <code>gaitCode = \$01</code> το πρόγραμμα θα συνεχίσει στις επόμενες συνθήκες, διαφορετικά θα συνεχίσει στην επόμενη υπόθεση <code>CASE</code> να επιλεγθεί η υπόθεση \$01 αν ισχύουν υποθέσεις που περιλαμβάνει
<code>ptrEEPROM = LTurn</code>	^^ορίζει ως περπάτημα την αριστερή στροφή

`rightRamp = VeryFast` ^^Δίνει την τιμή της θέσης μνήμης `fast` στη κίνηση των δεξιών ποδιών για να ρυθμίσει την ταχύτητα κίνησης τους

`leftRamp = VeryFast` ^^Δίνει την τιμή της θέσης μνήμης `fast` στη κίνηση των αριστερών ποδιών για να ρυθμίσει την ταχύτητα κίνησης τους

`GOTO Walking_Engine` ^^Το πρόγραμμα να συνεχιστεί στην υπορουτίνα `Walking_Engine`

`CASE $02` ^^Αν δεν ισχύει η προηγούμενη υπόθεση, να επιλεγεί η υπόθεση `$02` αν ισχύουν υποθέσεις που περιλαμβάνει

`ptrEEPROM = Rturn` ^^ορίζει ως περπάτημα την δεξιά στροφή

`rightRamp = VeryFast` ^^Όμοια με παραπάνω

`leftRamp = VeryFast` ^^Όμοια με παραπάνω

`GOTO Walking_Engine` ^^Το πρόγραμμα να συνεχιστεί στην υπορουτίνα `Walking_Engine`

`ENDSELECT` ^^Τέλος της εντολής `SELECT...CASE`

^^Στη συνέχεια συγκρίνει αν το `HIGHNIB` ψηφίο της `gaitCode`, επιλογής περπατήματος, είναι διαφορετικό από το 2, δηλαδή κάνει κίνηση διαφορετική από κάποια κίνηση προς τα μπροστά, `$20` – Forward, `$21` - Forward Left, `$22` - Forward Right, να επιλέξει την κίνηση προς τα μπροστά forward διαφορετικά να επιλέξει την κίνηση προς τα πίσω backward.

`IF gaitCode.HIGHNIB <= $2 THEN`

`ptrEEPROM = Forward`

`ELSE`

`ptrEEPROM = Back`

`ENDIF`

`Setup_Ramp_Vars:`

^^Τέλος της IF...

^^Ρουτίνα `Setup_Ramp_Vars`

^^Σε αυτή την ρουτίνα ρυθμίζεται η ταχύτητα της ράμπας (ramp) των ποδιών, ώστε να επιτευχθεί η διαφορά στην ταχύτητα του περπατήματος και η σταδιακή στροφή κατά το περπάτημα.

^^Για να ξεχωρίσουμε την διαφορά σε ποιες κινήσεις πηγαίνει γρήγορα χρησιμοποιείται ξανά η εντολή `IF... THEN` και η σύγκριση με το πρώτο ψηφίο της `gaitCode` που δηλώνει την ταχύτητα και την κίνηση. Αν η κίνηση είναι μία από τις γρήγορα μπροστά (`$10` - Forward Fast, `$11` - Fast Forward Left, `$12` - Fast Forward Right) ή

γρήγορα πίσω (\$40 - Fast Backward, 41 - Fast Backward Left, \$42 - Fast Backward Right) τότε η ταχύτητα της ράμπας (ramp) να είναι πολύ γρήγορη, διαφορετικά για όλες τις άλλες κινήσεις γρήγορα.

```
IF (gaitCode.HIGHNIB = $1) OR (gaitCode.HIGHNIB = $4) THEN
```

```
    ramp = VeryFast
```

```
ELSE
```

```
    ramp = Fast
```

```
ENDIF
```

^^Τέλος της IF...

^^Με την χρήση της εντολής SELECT ... CASE γίνεται η επιλογή της ταχύτητας στις ράμπες των δεξιών και αριστερών ποδιών. Η διαφορά στην ταχύτητα των δεξιών με τα αριστερά πόδια δημιουργεί την σταδιακή κίνηση με ταυτόχρονη στροφή.

^^Το δεύτερο ψηφίο της gaitCode δηλώνει την φορά της κίνησης, (0) για ευθεία, (1) για αριστερά, (2) για δεξιά. Χρησιμοποιώντας την εντολή SELECT ... CASE στην επιλογή της έκφρασης σύγκρισης, με την δεύτερη εντολή LOWNIB, διαλέγει το τελευταίο στοιχείο της gaitCode και το συγκρίνει με τις υποθέσεις. Αν είναι (0) στην πρώτη επιλογή δίνει την ίδια ταχύτητα αριστερά και δεξιά στις ράμπες, αν είναι (1) δίνει στις ράμπες της αριστερής πλευράς λίγο παραπάνω ταχύτητα διαφορετικά αν είναι δεξιά (2) στην δεξιά.

```
SELECT gaitCode.LOWNIB
```

```
CASE $0
```

```
    rightRamp = ramp
```

```
    leftRamp = rightRamp
```

```
CASE $1
```

```
    rightRamp = ramp
```

```
    leftRamp = (ramp+$2)
```

```
CASE $2
```

```
    leftRamp = ramp
```

```
    rightRamp = (ramp+$2)
```

```
ENDSELECT
```

^^τέλος της SELECT ... CASE

```
'-----[ Main Walking Routine ]-----  
-----
```

```
Walking_Engine:
```

^^Υπορουτίνα Walking_Engine

^^διαβάζει τα τρία πρώτα τμήματα των bytes που είναι αποθηκευμένα στην ptrEEPROM και προσδιορίζει την ταχύτητα των σερβομοτέρ (ramp).

' If Walking_Engine encounters an \$FF it has reached the
' end of the Forward EEPROM data. ptrEEPROM is reset
' and the process repeats.

READ ptrEEPROM,ServoAddr ^^Διαβάζει την τιμή της θέσης μνήμης ptrEEPROM και την αποθηκεύει στην μεταβλητή ServoAddr

DO WHILE servoAddr <> \$FF ^^Επαναλαμβάνει το περιεχόμενο της DO WHILE για όσο ισχύει η συνθήκη. Αν διαβαστεί η τιμή \$FF φτάνει το τέλος της κίνησης, η ptrEEPROM επανέρχεται στις αρχικές τιμές της και η διαδικασία επαναλαμβάνεται

IF (servoAddr // 2) = 1 THEN ^^Για να ξεχωρίσουμε διάφορες κινήσεις κατά τη διαδικασία του βήματος χρησιμοποιούνται διάφορες μαθηματικές πράξεις. Η πράξη (//) δίνει σαν αποτέλεσμα το υπόλοιπο από την διαίρεση. Αν το υπόλοιπο είναι (1) τότε σημαίνει ότι η κίνηση είναι του κάθετου άξονα και σηκώνει το πόδι

ramp = LiftRamp

ELSE ^^Διαφορετικά αν υπάρχει κίνηση από πριν, προσδιορίζει τις τιμές της ράμπας την ανάλογα με θέση του ποδιού, δεξιά ή αριστερά, και συνεχίζει περνώντας τον έλεγχο και την μεταβλητή της ptrEEPROM στην υπορουτίνα Write_PSC

^^Αν η τιμή είναι δείχνει ότι η επόμενη κίνηση είναι του μπροστά δεξιού ποδιού ή του πίσω δεξιά ποδιού δίνει στην τιμή της ράμπας στην τιμή της δεξιάς ράμπας

IF (servoAddr = \$00) OR (servoAddr = \$02) THEN

ramp = rightRamp

ENDIF

^^Τέλος της IF...

^^Αν η τιμή είναι δείχνει ότι η επόμενη κίνηση είναι του μπροστά αριστερού ποδιού ή του πίσω αριστερού ποδιού δίνει στην τιμή της ράμπας στην τιμή της αριστερής ράμπας

```

IF(servoAddr = $04) OR (servoAddr = $06) THEN
ramp = leftRamp
ENDIF          ^^Τέλος της IF...
ENDIF          ^^Τέλος της IF...
GOSUB Write_PSC          ^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα
                        Write_PSC
LOOP           ^^Τέλος της DO WHILE

' If adjustment mode end

IF ptrEEPROM = Adjust + 36 THEN          ^^αν έχει επιλεχθεί να γίνει η ρύθμιση
                                         των ποδιών "Adjust legs" τότε
                                         τελειώνει εδώ η ρουτίνα

END
ENDIF          ^^Τέλος της IF...

'Check for a button press

GOSUB Read_Buttons          ^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα
                        Read_Buttons
IF (btns.BIT0 = Pressed) OR (btns.BIT1 = Pressed) THEN
GOTO Select_GaitCode          ^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα
                        Select_GaitCode
GOTO Get_IR          ^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα Get_IR

ENDIF          ^^Τέλος της IF...

'---- [Sub routines] -----
                        ^^ΥΠΟΡΟΥΤΙΝΕΣ
'----- [Serial Out EEPROM values to PSC] -----
Write_PSC:          ^^Υπορουτίνα αποστολής δεδομένων στον PSC
                        ^^Παίρνει τα δεδομένα της κίνησης και ξεχωρίζει το
                        LOWBYTE που δηλώνει την θέση του ποδιού στην
                        οριζόντια κίνηση και το HIGHBYTE που δηλώνει την
                        κάθετη κίνηση

READ ptrEEPROM+1, servoPosition.LOWBYTE, servoPosition.HIGHBYTE
SEROUT PSC,Baud,["!SC",ServoAddr,          ^^Με την εντολή SEROUT στέλνει τα
                                         δεδομένα στον PSC

```

Ramp,servoPosition.LOWBYTE, servoPosition.HIGHBYTE, CR]	
ptrEEPROM = ptrEEPROM + 3	^^Η ptrEEPROM ανανεώνεται και πηγαίνει στα επόμενα 3 BYTE της EEPROM
READ ptrEEPROM,servoAddr	
PAUSE delay	^^Παύση ίση με την τιμή της delay
RETURN	^^Επιστροφή στην επόμενη εντολή μετά το GOTO SUBROUTINE

' ptrEEPROM is updated (+3) to point to the next 3 byte chunk
' of EEPROM and control is returned to Walking_Engine.

'---- [Button Code Section] -----

Select_GaitCode:

^^Η εντολή DO WHILE επαναλαμβάνει τις εντολές που βρίσκονται ενδιάμεσα της και της εντολής LOOP μέχρι να βρεθεί αληθής η συνθήκη που έχει οριστεί

DO WHILE (btns.BIT0 = NotPressed) OR (btns.BIT1 = NotPressed)	^^Αν ή το ένα Button ή το άλλο είναι πατημένο τότε:
IF (btns > %00) AND (btns < %11) THEN ' one or the other pressed?	
btnIndx = btnIndx + btn1 // 15	^^Να αυξήσει την τιμή της btnIndx κατά ένα αριθμό αν το Button1 είναι πατημένο
btnIndx = btnIndx + (14 * btn2) // 15	^^Να μειώσει την τιμή της btnIndx κατά ένα αριθμό αν το Button2 είναι πατημένο
READ btnIndx, segments	^^Ανανέωση της ένδειξης LED
READ gait + btnIndx, gaitCode	^^Ανανέωση της επιλογής περπατήματος gaitCode
PAUSE 250	^^Παύση 1/4 sec μεταξύ των αλλαγών
ENDIF	^^Τέλος της IF...
GOSUB Read_Buttons	^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα Read_Buttons
LOOP	^^Τέλος της DO WHILE
GOTO Parse_GaitCode	^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα Parse_GaitCode

'----- [Read Buttons] -----

Read_Buttons:

^^Υπορουτίνα Αναγνώρισης κατάστασης Button
^^Αναγνωρίζει ότι είναι πατημένα και τα δύο button μέσω μίας επανάληψης που περιμένει να βρει τα button πατημένα κάθε φορά που εκτελείται ο κύκλος εντολών μέσα στην FOR. Στο τέλος υπάρχει μία παύση PAUSE για να δίνει καθυστέρηση και να περιμένει να είναι πατημένα τα button για κάποιο χρόνο.

```
btns = %0011
FOR idx = 1 TO 5
btns.BIT0 = btns.BIT0 & ModeBtn
btns.BIT1 = btns.BIT1 & StartBtn
PAUSE 5
NEXT
```

RETURN

^^Επιστροφή στην επόμενη εντολή μετά το GOTO SUBROUTINE

'-----[IR Routines]-----

Get_IR:

^^Υπορουτίνα Υπέρυθρων
^^μηδενισμός της μεταβλητής distTemp
^^μηδενισμός της μεταβλητής distance
^^Δημιουργία επανάληψης για να πάρουμε τον μέσο όρο μέτρησης από τρεις μετρήσεις των υπερέρυθρων
^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα ADC0831_Reading για να γίνει η λήψη των δεδομένων (8bit) από τον ADC '
^^Υπολογισμός μέσου όρου μέτρησης
^^Παύση 1/20 του δευτερολέπτου
^^Τέλος της επανάληψης αφού ολοκληρωθούν οι τρεις κύκλοι του μετρητή
^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα ADC0831_Distance_Calc για να πάρουμε τον υπολογισμό της αποστάσεως

```
distTemp = 0
distance = 0
FOR counter = 1 TO 3
GOSUB ADC0831_Reading
distance = distance + (distTemp/3)
PAUSE 50
NEXT
GOSUB ADC0831_Distance_Calc
```

^^FUZZY LOGIC ΕΛΕΓΚΤΗΣ για την επιλογή της επόμενης κίνησης. Η έξοδος του εξαρτάται από την απόφαση που θα πάρει βάση των δεδομένων εισόδου που έχει, απόσταση εμποδίου, και προηγούμενη κίνηση.

SELECT distance

^^Εντολή SELECT ... CASE. Γίνεται σύγκριση της απόστασης που διάβασαν οι υπέρυθρες με την υπόθεση και ανάλογα αν αληθεύει ή όχι εκτελείται το περιεχόμενο της ή προχωράει στην επομένη υπόθεση.

CASE < 15

^^Αν η απόσταση είναι μικρότερη από 15cm συνεχίζει

IF gaitCode = \$10 THEN

^^Αν η gaitCode είναι \$10 περπάτημα μπροστά γρήγορα τότε η tempCode να γίνει \$40 περπάτημα γρήγορα πίσω

tempCode = \$40

ELSEIF gaitcode = \$20 THEN

^^διαφορετικά αν η gaitCode είναι \$20 περπάτημα μπροστά τότε η tempCode να γίνει \$20 περπάτημα γρήγορα μπροστά

tempCode = \$40

ELSEIF gaitcode <> \$10 AND gaitCode <> \$20 THEN

^^Διαφορετικά αν η gaitCode είναι διαφορετική από την \$10 και \$40 τότε η tempCode να γίνει \$42 περπάτημα γρήγορα πίσω και δεξιά

tempCode = \$42

ENDIF

^^Τέλος της IF...

CASE < 30

^^Αν η απόσταση είναι μικρότερη από 30cm συνεχίζει

IF gaitCode = \$10 THEN

^^Αν η gaitCode είναι \$10 περπάτημα μπροστά γρήγορα τότε η tempCode να γίνει \$40 περπάτημα γρήγορα πίσω

tempCode = \$40

ELSEIF gaitcode = \$20 THEN

^^διαφορετικά αν η gaitCode είναι \$20 περπάτημα μπροστά τότε η tempCode να γίνει \$42 περπάτημα γρήγορα πίσω δεξιά

tempCode = \$42

ELSEIF gaitcode <> \$10 AND gaitCode <> \$20 THEN

^^Διαφορετικά αν η gaitCode είναι διαφορετική από την \$10 και \$40 τότε η tempCode να γίνει \$32 περπάτημα πίσω και δεξιά

tempCode = \$32

ENDIF	^^Τέλος της IF...
CASE < 45	^^Αν η απόσταση είναι μικρότερη από 45cm συνεχίζει
IF gaitCode = \$10 THEN	^^Αν η gaitCode είναι \$10 περπάτημα μπροστά γρήγορα τότε η tempCode να γίνει \$42 περπάτημα γρήγορα πίσω και δεξιά
tempCode = \$42	
ELSEIF gaitcode = \$20 THEN	^^διαφορετικά αν η gaitCode είναι \$20 περπάτημα μπροστά τότε η tempCode να γίνει \$32 περπάτημα πίσω και δεξιά
tempCode = \$32	
ELSEIF gaitcode <> \$10 AND gaitCode <> \$20 THEN	
tempCode = \$01	^^Διαφορετικά αν η gaitCode είναι διαφορετική από την \$10 και \$20 τότε η tempCode να γίνει \$01 στροφή αριστερά
ENDIF	^^Τέλος της IF...
CASE < 60	^^Αν η απόσταση είναι μικρότερη από 60cm συνεχίζει
IF gaitCode = \$10 THEN	^^Αν η gaitCode είναι \$10 περπάτημα μπροστά γρήγορα τότε η tempCode να γίνει \$11 περπάτημα γρήγορα μπροστά αριστερά
tempCode = \$11	
ELSEIF gaitcode = \$20 THEN	^^διαφορετικά αν η gaitCode είναι \$20 περπάτημα μπροστά τότε η tempCode να γίνει \$21 περπάτημα μπροστά αριστερά
tempCode = \$21	
ELSEIF gaitcode <> \$10 AND gaitCode <> \$20 THEN	
tempCode = \$01	^^Διαφορετικά αν η gaitCode είναι διαφορετική από την \$10 και \$20 τότε η tempCode να γίνει \$01 στροφή αριστερά
ENDIF	^^Τέλος της IF...
CASE < 70	^^Αν η απόσταση είναι μικρότερη από 70cm συνεχίζει
IF gaitCode = \$10 THEN	^^Αν η gaitCode είναι \$10 περπάτημα μπροστά γρήγορα τότε η tempCode να γίνει \$21 περπάτημα μπροστά αριστερά
tempCode = \$21	
ELSEIF gaitcode = \$20 THEN	^^διαφορετικά αν η gaitCode είναι \$20 περπάτημα μπροστά τότε η tempCode να γίνει

\$21 περπάτημα γρήγορο μπροστά αριστερά

tempCode = \$21

ELSEIF gaitcode <> \$10 AND gaitCode <> \$20 THEN

tempCode = \$20

^^Διαφορετικά αν η gaitCode είναι διαφορετική από την \$10 και \$20 τότε η tempCode να γίνει \$20 περπάτημα μπροστά

ENDIF

^^Τέλος της IF...

CASE ELSE

tempCode = \$10

ENDSELECT

^^τέλος της SELECT ... CASE

IF tempCode = gaitCode THEN

^^Αν η tempCode είναι ίδια με την gaitCode Δηλαδή καμιά αλλαγή στην gaitCode

ptrEEPROM = ptrEEPROM - 48

^^Τότε να κάνει επανεκκίνηση του δείκτη της μνήμης EEPROM

GOTO Walking_Engine

^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα Walking_Engine

ELSE

^^διαφορετικά ορίζει την gaitCode ίση με την tempCode

gaitCode = tempCode

και ξαναδοκιμάζει να πάρει καινούρια τιμή

GOSUB Get_Segments

^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα Get_Segments

GOTO Parse_GaitCode

^^Το πρόγραμμα συνεχίζεται στην υπορουτίνα Parse_GaitCode για να ορίσει τις παραμέτρους του περπατήματος

ENDIF

^^Τέλος της IF...

'Get ADC0831 8-bit data

ADC0831_Reading:

^^Υπορουτίνα ADC0831_Reading, δίνει τα δεδομένα που διαβάζει από τις υπέρυθρες στον μικροελεγκτή

LOW FrontIR

^^Ενεργοποίηση του ADC

SHIFTIN DIO_10,CLK,

^^Για τον συγχρονισμό του ADC0831 και του μικροελεγκτή πρέπει να σταλεί από τον μικροελεγκτή ένας παλμός χρονισμού ώστε να ξεχωρίζει πότε ξεκινάει η μετάδοση και πιο bit θα είναι αυτό που θα περιμένει να λάβει. Για το λόγο αυτό χρησιμοποιείτε ο χρονιστής DIO_10 και CLK, Περισσότερες πληροφορίες υπάρχουν παρακάτω στην ανάλυση των εντολών.

MSBPOST,[distTemp\9]	^^Λήψη δεδομένων από ADC0831
HIGH FrontIR	^^Απενεργοποίηση του ADC.
RETURN	^^Επιστροφή στην επόμενη εντολή μετά το GOTO SUBROUTINE
'Linearizing function to calculate 'GP2D12 distance reading	
ADC0831_Distance_Calc:	^^Υπορουτίνα υπολογισμού της απόστασης από την τιμή που διαβάστηκε
distance = (Slope /(distance + y_Intercept)) - k	
RETURN	^^Επιστροφή στην επόμενη εντολή μετά το GOTO SUBROUTINE
'Update LED display	
Get_Segments:	^^Υπορουτίνα Get_Segments: δίνει την ανανέωση του αριθμού που δείχνει το LED display
btnIndx = Gait	^^Δίνει στην btnIndx την τιμή του περπατήματος
READ btnIndx, tempCode	^^διαβάζει την τιμή της btnIndx και την γράφει στην μεταβλητή tempCode
DO WHILE tempCode <> gaitCode	^^Επαναλαμβάνει για όσο ισχύει η συνθήκη
btnIndx = btnIndx + 1	^^Δίνει στην btnIndx την τιμή της επόμενης θέσης μνήμης
READ btnIndx, tempCode	^^Διαβάζει την τιμή της btnIndx και την γράφει στην μεταβλητή tempCode
LOOP	^^Τέλος της DO WHILE
READ btnIndx-Gait, segments	^^Όταν το tempCode είναι ίσο με το gaitCode τότε σταματάει η επανάληψη και συνεχίζει στην επόμενη εντολή και δίνει το νούμερο της αντίστοιχης κίνησης στην ένδειξη του LED Display
RETURN	^^Επιστροφή στην επόμενη εντολή μετά το GOTO SUBROUTINE

ΣΤ. Εντολές και σύνταξη Εντολών

Όλη η ανάλυση για την λειτουργία και την χρήση των εντολών βρίσκεται στο εγχειρίδιο λειτουργίας του μικροελεγκτή Syntax and reference manual web-BSM-v2.2. Στις σελίδες 124-128 υπάρχει μια σύντομη αναφορά στη λειτουργία της κάθε εντολής. [2]

Μια αναφορά στις εντολές και τα στοιχεία που χρησιμοποιούνται στο πρόγραμμα αυτής της εργασίας υπάρχει παρακάτω για την διευκόλυνση του αναγνώστη.

Κατά το ξεκίνημα του προγραμματισμού πρέπει να οριστούν στην μνήμη του μικροελεγκτή τα διάφορα στοιχεία που θα χρησιμοποιούνται κατά την εκτέλεση του προγράμματος, όπως σταθερές(Constants) και μεταβλητές(Variables) τιμές, είσοδοι και έξοδοι από τα PIN επικοινωνίας του, κ.α.

Οι μαθηματικές πράξεις στο πρόγραμμα γίνονται με μια σειρά που είναι γραμμένες και για να ξεχωρίσουν μεταξύ τους είναι αναγκαία η χρήση παρενθέσεων. Κατά την εκτέλεση πράξεων προηγούνται οι μοναδιαίες (Unary) από τις δυαδικές (Binary). Βασική πληροφορία είναι ότι δεν μπορούμε να αποθηκεύουμε αρνητικές τιμές χρησιμοποιώντας το πρόσημο (-) μπροστά από δυαδικούς αριθμούς γιατί τους μετατρέπει στον αντίστροφό τους και μας εμφανίζει στο τέλος ένα άσχετο θετικό αριθμό. Σελίδες 102-121.

VAR (Variables) Μεταβλητές. Είναι μεταβλητές τιμές με τις οποίες ορίζονται διάφορα στοιχεία όπως μεγέθη, ψευδώνυμα, πίνακες, ορισμοί και σταθερές σε μία θέση της μνήμης του μικροελεγκτή. Οι τιμές που μπορούν να πάρουν ορίζονται σε:

BIT τιμές 0 ή 1.

NIB τιμές 0 έως 15.

BYTE τιμές 0 έως 255.

WORD τιμές 0 έως 65535.

Σελίδες 81-89

“Όνομα” VAR “Μέγεθος”

CON (Constants) Σταθερές. Είναι σταθερές τιμές που χρησιμοποιούνται με παρόμοιο τρόπο με τις VAR.

Σελίδες 94-99

DATA Γράφει δεδομένα στην EEPROM κατά την διάρκεια του φορτώματος του προγράμματος.

{ Symbol } DATA DataItem { , DataItem... }

Symbol, είναι ένα μοναδικό σύμβολο όνομα που αυτόματα ορίζεται σαν σταθερά ίση με αριθμό θέσης που βρίσκεται το πρώτο στοιχείο των δεδομένων.

DataItem, είναι μία σταθερά ή έκφραση που υποδεικνύει μία τιμή (0 – 65535) και

προαιρετικά πως θα αποθηκευτεί.

Σελίδες 153-158

PIN Χρησιμοποιείται για ορίσει ο χρήστης την λειτουργία των I/O του μικροελεγκτή 0-15. Ορίζονται σαν εισοδοι ή έξοδοι και ορίζεται ποια θα είναι λειτουργία τους στην περίπτωση που θα χρησιμοποιηθούν σαν γεννήτριες παλμών κ.λ.π. Σελίδες 81-84 και 99-102

“Όνομα” PIN “Αρ. Pin”

DIRS Οι εντολές DIRs είναι υπεύθυνες για την χρήση των PINs και καθορίζουν την λειτουργία τους σαν εισόδους ή εξόδους. Σελίδα 83.

HIGH και LOW Οι εντολές αυτές φέρνουν αντίστοιχα τις εξόδους των PIN σε κατάσταση High και Low.

HIGH Pin

Pin είναι μια σταθερά/μεταβλητή/έκφραση (0 – 15) που καθορίζει ποιο I/O pin θα μπει σε κατάσταση High ή Low. Το Pin αυτό αυτόματα θα οριστεί και σαν έξοδος. Σελίδες 281 - 282, 215 - 216

GOTO Οδηγεί το πρόγραμμα να συνεχίσει στο συγκεκριμένο σημείο που ορίζει διεύθυνση Address.

GOTO Address

GOSUB Address

RETURN Η εντολή GOSUB οδηγεί το πρόγραμμα να συνεχίσει στην διεύθυνση που εμφανίζεται αμέσως μετά την εντολή. Εκτελεί το περιεχόμενο της υπορουτίνας μέχρι να συναντήσει την εντολή RETURN και γυρίζει στη σειρά που ήταν, για να συνεχίσει το πρόγραμμα. Η υπορουτίνα μπορεί να χρησιμοποιηθεί σε διάφορα σημεία του προγράμματος για οικονομία χώρου εντολών. Στις υπορουτίνες βάζουμε το σημάδι (:) αμέσως μετά το όνομα και στο τελείωμα κλείνει το κομμάτι του προγράμματος με την εντολή END. Χρειάζεται προσοχή για να μην πέφτει το κυρίως πρόγραμμα σε υπορουτίνες και τις κάνει να επαναλαμβάνονται από τη ροή του προγράμματος. Σελίδες 209-212

DO...LOOP Δημιουργεί μία διαδικασία επανάληψης των εντολών που βρίσκονται ανάμεσα στο DO και το LOOP και η επανάληψη αυτή συνεχίζεται για όσο οι συνθήκες που του ορίζει ο χρήστης συνεχίζουν να είναι ανάλογα, αληθείς ή όχι. Σελίδες 175–77, 189

DO { WHILE | UNTIL Condition(s) }

Statement(s)

LOOP { WHILE | UNTIL Condition(s) }

Condition Είναι μια προαιρετική σταθερά/μεταβλητή/έκφραση (0 – 65535) που καθορίζει αν

η επανάληψη θα συνεχιστεί ή θα τερματιστεί, και ρέπει να ακολουθείται από WHILE ή UNTIL.

Statement είναι μια οποιαδήποτε πραγματική εντολή της PBASIC.

SELECT...CASE Ορίζεται μία έκφραση στο ξεκίνημα της ακολουθίας των εντολών που ακολουθούν μέχρι το ENDSELECT. Σε κάθε υπόθεση που ακολουθεί γίνεται μια σύγκριση με την έκφραση μέχρι την στιγμή που θα βρεθεί η πρώτη σύγκριση που θα αληθεύει. Τότε εκτελούνται οι εντολές που ακολουθούν και στην συνέχεια αγνοούνται οι υπόλοιπες υποθέσεις και το πρόγραμμα συνεχίζεται μετά το ENDSELECT. Σελίδες 387 – 390

SELECT Expression

CASE Condition(s)

Statement(s)

{ **CASE Condition(s)**

Statement(s)... }

{ **CASE ELSE**

Statement(s) }

ENDSELECT

Expression Είναι μία σταθερά ή μεταβλητή ή έκφραση που όταν συγκρίνεται με την έκφραση εκτιμάται αν είναι αληθής η όχι. Πολλαπλές συνθήκες μέσα στην ίδια υπόθεση χωρίζονται με κόμματα (,).

• **Statement** είναι η οποιαδήποτε ισχύουσα εντολή της PBASIC.

IF...THEN Εκτιμάτε η συνθήκη και αν είναι αληθής συνεχίζει στην διεύθυνση η εκτελείται η εντολή μετά το THEN, διαφορετικά συνεχίζεται στις επιλογές του ELSEIF/ELSE. Η χρήση του ELSEIF είναι προαιρετική και χρησιμοποιείται όπως το IF αλλά υπολογίζεται μόνο όταν η συνθήκη του IF δεν είναι αληθής. Η ακολουθία του ELSE είναι προαιρετική και εκτελείται στην περίπτωση που τα IF και ELSEIF δεν είναι αληθή. Το πρόγραμμα συνεχίζεται στην επόμενη σειρά του κώδικα κάθε φορά μέχρι να βρεθεί μια αληθής συνθήκη και να παρακάμψει τις επόμενες σειρές μέχρι το ENDIF. Σελίδες 231-240

IF Condition(s) THEN Address

IF Condition(s) THEN Statement(s) { ELSE Statement(s) }

IF Condition(s) THEN

Statement(s)

{ **ELSEIF Condition(s) THEN**

Statement(s)... }

{ **ELSE**

Statement(s) }

ENDIF

Condition Είναι μία κατάσταση όπως το “x = 7” που μπορεί να εκτιμηθεί σαν αληθής η όχι. Μπορεί να είναι πολύ απλή ή πολύπλοκη όπως περιγράφεται στο παραπάνω.

- **Address** είναι η διεύθυνση που θα συνεχιστεί το πρόγραμμα στην περίπτωση που η συνθήκη βρεθεί αληθής
- **Statement** Είναι η οποιαδήποτε ισχύουσα εντολή της PBASIC.

READ Διαβάζει την τιμή από μία θέση της μνήμης EEPROM και αποθηκεύει το αποτέλεσμα σε μία μεταβλητή (*Variable*). Σελίδες **369–73**

READ Location, Variable

READ Location, { WORD } Variable {, { WORD } Variable... }

Location είναι μία μεταβλητή/σταθερά/έκφραση (0 – 2047) που ορίζει από ποια θέση της μνήμης διαβάσει την τιμή.

Variable είναι η μεταβλητή που θα αποθηκευτεί η τιμή.

FOR...NEXT Δημιουργεί μία συνεχόμενη επανάληψη των εντολών που βρίσκονται ανάμεσα στο FOR και το NEXT. Η επανάληψη γίνεται τόσες φορές όσες επιλεγούν στον μετρητή μέχρι η τιμή του να ξεπεράσει την τιμή του EndValue. Σελίδες 191-197

FOR Counter = StartValue TO EndValue { STEP {-} StepValue }

Statement(s)

NEXT { Counter }

FOR Counter = StartValue TO EndValue { STEP StepValue }

Statement(s)

NEXT { Counter }

Counter Είναι μία μεταβλητή που χρησιμοποιείται σαν μετρητής.

StartValue Είναι μια σταθερά/μεταβλητή/έκφραση (0 – 65535) που ορίζει την αρχική τιμή της μεταβλητής του μετρητή.

EndValue Είναι μια σταθερά/μεταβλητή/έκφραση (0 – 65535) που ορίζει την τελική τιμή της μεταβλητής του μετρητή. Όταν η τιμή ξεπεράσει τα όρια της EndValue τότε σταματάει η επανάληψη και το πρόγραμμα συνεχίζεται στην επόμενη σειρά μετά το NEXT.

StepValue Είναι μια προαιρετική σταθερά/μεταβλητή/έκφραση (0 – 65535) που ορίζει το κάθε βήμα αύξησης ή μείωσης του μετρητή σε κάθε επανάληψη.

Statement Είναι η οποιαδήποτε ισχύουσα εντολή της PBASIC.

SHIFTIN Χρησιμοποιείται για να διαβάσει ο μικροελεγκτής δεδομένα από σύγχρονες σειριακές συσκευές. Σελίδες **431–34**

SHIFTIN *Dpin, Cpin, Mode, [Variable { \Bits } {, Variable { \Bits }...}]*

• ***Dpin*** Είναι μια σταθερά/μεταβλητή/έκφραση (0 – 15) που θα ορίσει ποιο PIN θα συνδεθεί με την έξοδο της σύγχρονης σειριακής συσκευής. Το PIN αυτό ορίζεται ταυτόχρονα ως είσοδος.

Cpin Είναι μια σταθερά/μεταβλητή/έκφραση (0 – 15) που ορίζει ποιο PIN θα συνδεθεί με την είσοδο CLOCK της σύγχρονης σειριακής συσκευής. Το PIN αυτό ορίζεται ταυτόχρονα ως έξοδος.

Mode Είναι μια σταθερά/μεταβλητή/έκφραση (0 – 3), ή ένα από τα τέσσερα προκαθορισμένα σύμβολα που ορίζει στην SHIFTIN ποια bits των δεδομένων θα διαβάσει και την σχέση των παλμών του CLOCK με τα δεδομένα αυτά.

Variable Είναι η μεταβλητή που θα αποθηκευτούν τα δεδομένα της εισόδου.

Bits Είναι μια προαιρετική σταθερά/μεταβλητή/έκφραση (1 – 16) που ορίζει πόσα bits θα εισαχθούν στην είσοδο της SHIFTIN. Αν δεν δοθεί κάποια εντολή η προκαθορισμένη είσοδος είναι τα 8 bits.