

BASKETBALL ROBOT

Τελική Αναφορά

Αντώνιος Γεώργιος Πίτσης
2013030141

Θεόφιλος Ζαχείλας
2013030126

ΣΚΟΠΟΣ ΤΟΥ PROJECT

Υλοποίηση ενός ρομπότ που θα πετάει μια μπάλα lego σε ένα στόχο μέχρι ένα συγκεκριμένο εύρος αποστάσεων και γωνιών, που θα καθοριστεί στο 2^ο milestone. Ο στόχος αυτός, αρχικά έχει ληφθεί υπόψιν, ως ένα πλαστικό ποτήρι, αλλά ενδέχεται να βελτιστοποιηθεί μέχρι και σε μια μπάσκέτα με καλάθι, σε μεγέθη που αναλογούν.

ΣΧΕΔΙΑΣΜΟΣ ΥΛΟΠΟΙΗΣΗΣ

Υλοποίηση ενός ρομπότ που θα πετάει μια μπάλα σε ένα κουτί μέχρι ένα συγκεκριμένο εύρος αποστάσεων και γωνιών, που θα καθοριστεί στο 2^ο milestone. Η τεχνολογία που θα χρησιμοποιηθεί είναι η lego mindstorms. Θα ξεκινήσουμε με τα Lego υλικά να χτίζουμε σταδιακά το ρομπότ. Επίσης θα προσθέσουμε έναν Large Servo Motor για να μπορέσουμε να περιστρέψουμε το ρομπότ προς την σωστή κατεύθυνση (δηλαδή να κοιτάει προς το κουτί). Την εύρεση της σωστής κατεύθυνσης και της απόστασης (του ρομπότ απ' το κουτί), θα την πετύχουμε μέσω ενός Ultrasonic Sensor. Ακόμη θα προσθέσουμε έναν Large Servo Motor για να επιτύχουμε την εκτίναξη της μπάλας. Το όλο σύστημα θα έχει ενδεχομένως έναν Touch Sensor, ο οποίος θα ξεκινάει τη διαδικασία. Ως πλατφόρμα ανάπτυξης θα χρησιμοποιήσουμε την NXC. Η πλατφόρμα υλοποίησης θα είναι το NXT Intelligent Brick.

ΔΙΑΔΙΚΑΣΙΑ ΥΛΟΠΟΙΗΣΗΣ

Στην παρούσα φάση υλοποίησης επικεντρωθήκαμε στην πλήρη κατασκευή του ρομπότ (την οποία φέραμε εις πέρας), καθώς και στην πλήρη επίτευξη της ευστοχίας του.

Αφού ολοκληρώσαμε το κατασκευαστικό κομμάτι του ρομπότ, ενσωματώσαμε τον **Ultrasonic Sensor** και προσπαθήσαμε να ανιχνεύσουμε το ποτήρι (στόχο), τόσο σε κοντινές αποστάσεις (7cm) όσο και σε μακρινές αποστάσεις (20cm). Στη συνέχεια προσομοιώσαμε τον πρώτο **Large Servo Motor**, ώστε να περιστρέφει το κινητό κομμάτι της κατασκευής μας. Σκοπός αυτής της κίνησης, είναι η δυνατότητα ανίχνευσης του στόχου, όταν αυτός δεν βρίσκεται εξαρχής μπροστά από τον Ultrasonic Sensor. Σειρά είχε ο δεύτερος **Large Servo Motor**, ο οποίος ενεργοποιείται μόνο όταν βρεθεί ο στόχος. Με συγκεκριμένη δύναμη, ανάλογα με την απόστασή του από τον στόχο, εκτοξεύει την Lego μπάλα. Τελευταίος ενσωματώθηκε ο **Touch Sensor**, που ξεκινάει όλη την παραπάνω διαδικασία.

ΠΡΟΔΙΑΓΡΑΦΕΣ

- Το εύρος αποστάσεων εκτείνεται από 0-30cm.
- Η γωνία ανίχνευσης είναι μεταξύ 0-180 μοίρες.
- Η αρχική θέση του ρομπότ είναι κάθετη και προς τα δεξιά της νοητής γραμμής προέκτασης, της μεγαλύτερης πλευράς της κατασκευής.
- Το ποτήρι (στόχος) βρίσκεται σε ευθεία με τον Ultrasonic Sensor, όταν αυτός το ανιχνεύσει.
- Στόχος θεωρείται το τετράγωνο χαρτόνι, το οποίο τοποθετήσαμε σε μία πλευρά του ποτηριού.

ΠΡΟΒΛΗΜΑΤΑ

- Ο Ultrasonic Sensor δεν ανιχνεύει πάντα την σωστή απόσταση. Αυτή η δυσλειτουργία, οφείλεται στην θέση που αποκτά τόσο ο συγκεκριμένος Sensor όσο και το ποτήρι. Κάποιες φορές, επειδή ο US Sensor <κοιτάει> λίγο προς τα κάτω (πχ λόγω ταλάντωσής του από κάποια προηγούμενη ρίψη) ή το ποτήρι δεν βρεθεί ακριβώς σε ευθεία με αυτόν (όταν το ανιχνεύσει), η μέτρηση λαμβάνεται λάθος. Ως συνέπεια των παραπάνω, ο US Sensor καταλήγει να έχει απόκλιση απ'το ποτήρι και η δύναμη που υπολογίζει το πρόγραμμα να είναι λανθασμένη.
- Η παρούσα κατασκευή δεν μας επιτρέπει να υποστηρίξουμε περιστροφή 360 μοιρών, εξαιτίας των καλωδίων που συνδέουν τους Sensors με το Brick και εμποδίζουν την περιστροφή.

ΠΕΡΙΓΡΑΦΗ ΝΧC

Είναι μια απλή γλώσσα για τον προγραμματισμό του προϊόντος LEGO MINDSTORMS NXT. Το NXT έχει έναν διερμηνέα bytecodes (που παρέχεται από το LEGO), ο οποίος μπορεί να χρησιμοποιηθεί για την εκτέλεση προγραμμάτων. Ο μεταγλωττιστής NXC μεταφράζει ένα πρόγραμμα προέλευσης σε bytecodes NXT, τα οποία στη συνέχεια μπορούν να εκτελεστούν στον Brick. Αν και οι δομές προεπεξεργαστή και ελέγχου του NXC είναι πολύ παρόμοιες με τη C, η NXC δεν είναι γενική γλώσσα προγραμματισμού - υπάρχουν πολλοί περιορισμοί που απορρέουν από τους περιορισμούς του διερμηνέα bytecodes NXT.

NXT MULTITHREADING – NXC MULTITASKING

Η NXT υποστηρίζει multi-threading , έτσι κάθε task της NXC απευθύνεται σε ένα NXT thread. Αυτό σημαίνει ότι μπορούμε να τρέχουμε παράλληλα threads , κάτι που είναι εξαιρετικά χρήσιμο καθώς σε ένα thread μπορούμε να ελέγχουμε τους αισθητήρες σε ένα άλλο να κινούμε το ρομπότ και σε ένα τρίτο να παίζουμε ένα ήχο. Όμως αυτό μπορεί να οδηγήσει σε προβλήματα καθώς όταν τρέχουν παράλληλα threads μπορεί να κάνουν αλλαγές πάνω στα objects και να οδηγηθούμε σε race conditions . Έτσι η NXC υποστηρίζει την λειτουργία Semaphores και Mutex. Έτσι μπορούμε να διασφαλίσουμε ότι ευαίσθητα κομμάτια κωδικα θα εκτελεστούν με την σωστή σειρά όταν πρέπει αποφεύγοντας έτσι πιθανά race conditions.

NXT INTERRUPTS

Παρατηρήσαμε πως η γλώσσα NXC δεν υποστηρίζει interrupts, οπότε επικεντρωθήκαμε στο να τρέξουμε τις διεργασίες παράλληλα, ώστε να αποφύγουμε την απλότητα μιας απλής διεργασίας (main) που τα έτρεχε όλα μαζί. Έτσι καταφέραμε να αποφύγουμε στο πλείστο των περιπτώσεων την αλληλοεξάρτηση των tasks ώστε να εξασφαλίσουμε πως αν κάτι πάει στραβά στο πρόγραμμά μας σε κάποιο task, να μην επηρεαστεί όλο το υπόλοιπο πρόγραμμα.

<http://www.tau.ac.il/~stoledo/lego/nxt-native/> <http://www.tau.ac.il/~stoledo/lego/nxt-native/>

TUTORIALS

Τα Manuals μας βοήθησαν να καταλάβουμε τη λειτουργία του NXT Intelligent Brick και της NXC. Συγκεκριμένα μπορέσαμε να υλοποιήσουμε κάποια αρχεία NXC (δικά μας tutorials), για να προσαρμόσουμε στις λειτουργίες των επιμέρους κομματιών (πχ κινητήρας), καθώς και στον τρόπο σύνδεσής τους με το Brick. Αυτά τα NXC αρχεία δημιουργήθηκαν ώστε να:

Εμφανίζουμε μηνύματα στη οθόνη του και να παράγουμε ήχους, όπως φαίνεται στο πρώτο tutorial.

Θέτουμε σε λειτουργία τον κινητήρα, όσο κρατάμε πατημένο τον Touch_Sensor, καθώς και να τον ακινητοποιούμε, όταν αφηίνουμε τον Touch_Sensor, όπως φαίνεται στο δεύτερο tutorial.

Να περιστρέφουμε τον κινητήρα, υπό συγκεκριμένη γωνία προς μία κατεύθυνση (με Touch_Sensor=ON), αλλά και προς την αντίθετη κατεύθυνση (με Touch_Sensor=OFF), όπως φαίνεται στο τρίτο tutorial.

Να περιστρέφουμε τον κινητήρα σταδιακά, με την ενεργοποίηση του Touch_Sensor, όπως φαίνεται στο τέταρτο tutorial.

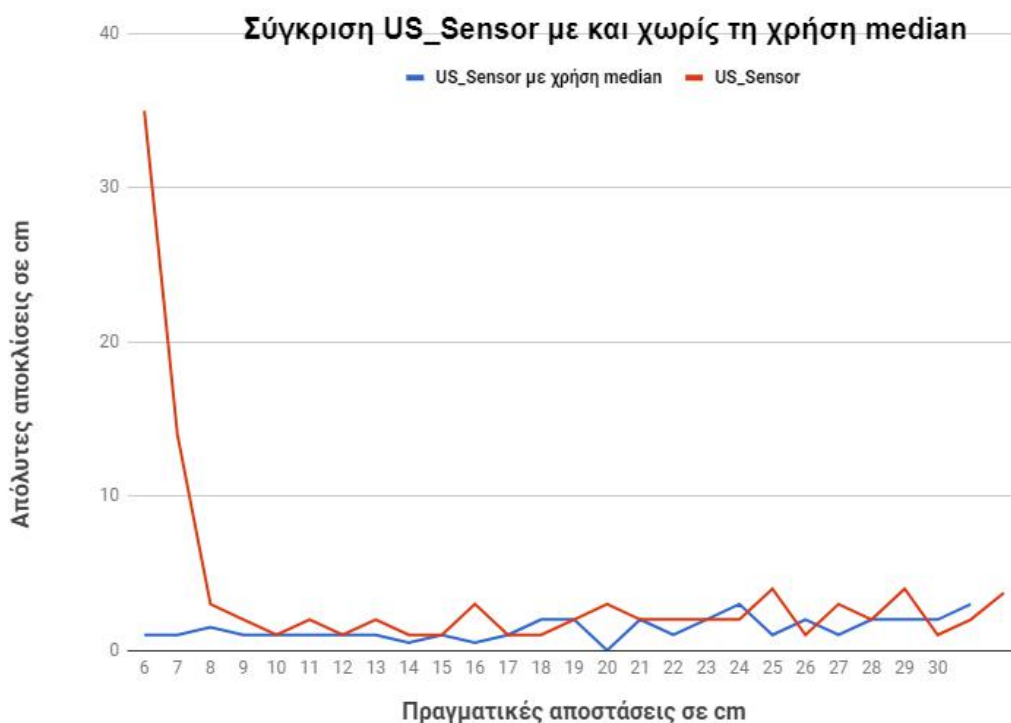
Ανιχνεύει την απόσταση ενός αντικείμενου και να την εμφανίζει στην οθόνη του Brick . Όταν ένα αντικείμενο βρίσκεται σε απόσταση μικρότερη από ένα threshold (20 cm) παίζει έναν ήχο.

ΜΕΤΡΗΣΕΙΣ

Βελτιστοποίηση με χρήση της Median τιμής

Οι μετρήσεις των αποστάσεων έγιναν με τη μέθοδο της **Median** τιμής. Πιο συγκεκριμένα σε κάθε <step> του Ultrasonic Sensor, δειγματοληπτούμε 3 τιμές του και επιλέγουμε την μεσαία, καθώς παρατηρήσαμε πως αυτή ήταν η πιο ακριβής μέτρηση. Δηλαδή, αντί να δειγματοληπτούμε κάθε 35 ms, που είναι το άνω όριο για να ληφθούν δεδομένα από τον Ultrasonic Sensor, αυξήσαμε το όριο στα 105 ms. Με αυτό τον τρόπο εξαλείψαμε σε μεγάλο βαθμό τυχόν λανθασμένες μετρήσεις, όπως την μέτρηση 255 που εμφανιζόταν όταν το ποτήρι βρισκόταν σε σχετικά μακρινή απόσταση (πχ 20 cm), οι οποίες οφείλονται στους λόγους που αναφέραμε παραπάνω.

Παρακάτω παρουσιάζουμε τις αποκλίσεις των μετρήσεων με και χωρίς τη χρήση της μεθόδου της median τιμής, καθώς και το ποσοστό βελτίωσης των μετρήσεων. Συγκεκριμένα αφού καταγράψαμε τις μετρήσεις με και χωρίς τη χρήση median, υπολογίσαμε τις αποκλίσεις από τις αντίστοιχες πραγματικές αποστάσεις. Βλέπουμε πώς οι αποκλίσεις με χρήση της median είναι πιο κοντά στο 0 σε σχέση με αυτές χωρίς τη χρήση της median.



Στη συνέχεια υπολογίσαμε το μέσο όρο των αποκλίσεων. Χωρίς τη χρήση median ο μ.ο προέκυψε 3,73, πράγμα το οποίο επιβεβαιώνει και την αναφορά του NXT manual, το οποίο θέλει τον US Sensor να έχει απόκλιση 2-3 cm στις μετρήσεις που λαμβάνει σε σχέση με τις πραγματικές τιμές των αποστάσεων των αντικειμένων από αυτόν. Με τη χρήση median ο μ.ο προέκυψε 1,40. Επομένως πετύχαμε μια διόρθωση των μετρήσεων σε ποσοστό $(1,4 - 3,73) / 3,73 * 100\% = 62,46\%$.

Μοντελοποίηση της σχέσης απόσταση - δύναμη

Παρακάτω φαίνονται οι μετρήσεις που πήραμε για τον υπολογισμό της δύναμης ανάλογα με την απόσταση.

Αποστάσεις/Δυνάμεις

5	31
6	35
7	42
8	45
9	47
10	52
11	53
12	56
13	57
14	62
15	65
16	67
17	68
18	72
19	73
20	77
21	78
22	79
23	80
24	83
25	88
26	95
27	96
28	97
29	98
30	99

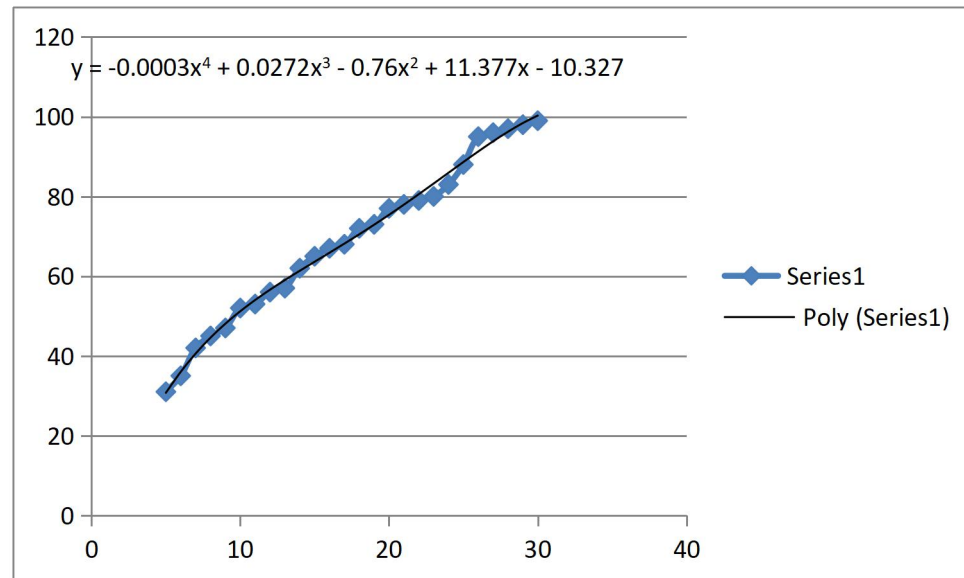
Με βάση τις παραπάνω τιμές, προέκυψε το αντίστοιχο διάγραμμα, από το οποίο εξήχθη η συνάρτηση για τον υπολογισμό της δύναμης. **Η μέθοδος που ακολουθήθηκε είναι η fit curving, με την οποία προέκυψε η πολυωνμική συνάρτηση 4^{ου} βαθμού από το Excel (μαύρη γραμμή).**

Κάθετος άξονας: Δυνάμεις

Οριζόντιος άξονας: Αποστάσεις

Μπλέ γραμμή: Οι δικές μας μετρήσεις

Μαύρη γραμμή: Η πολυωνυμική συνάρτηση 4^{ου} βαθμού.



ΤΟ ΠΡΟΓΡΑΜΜΑ

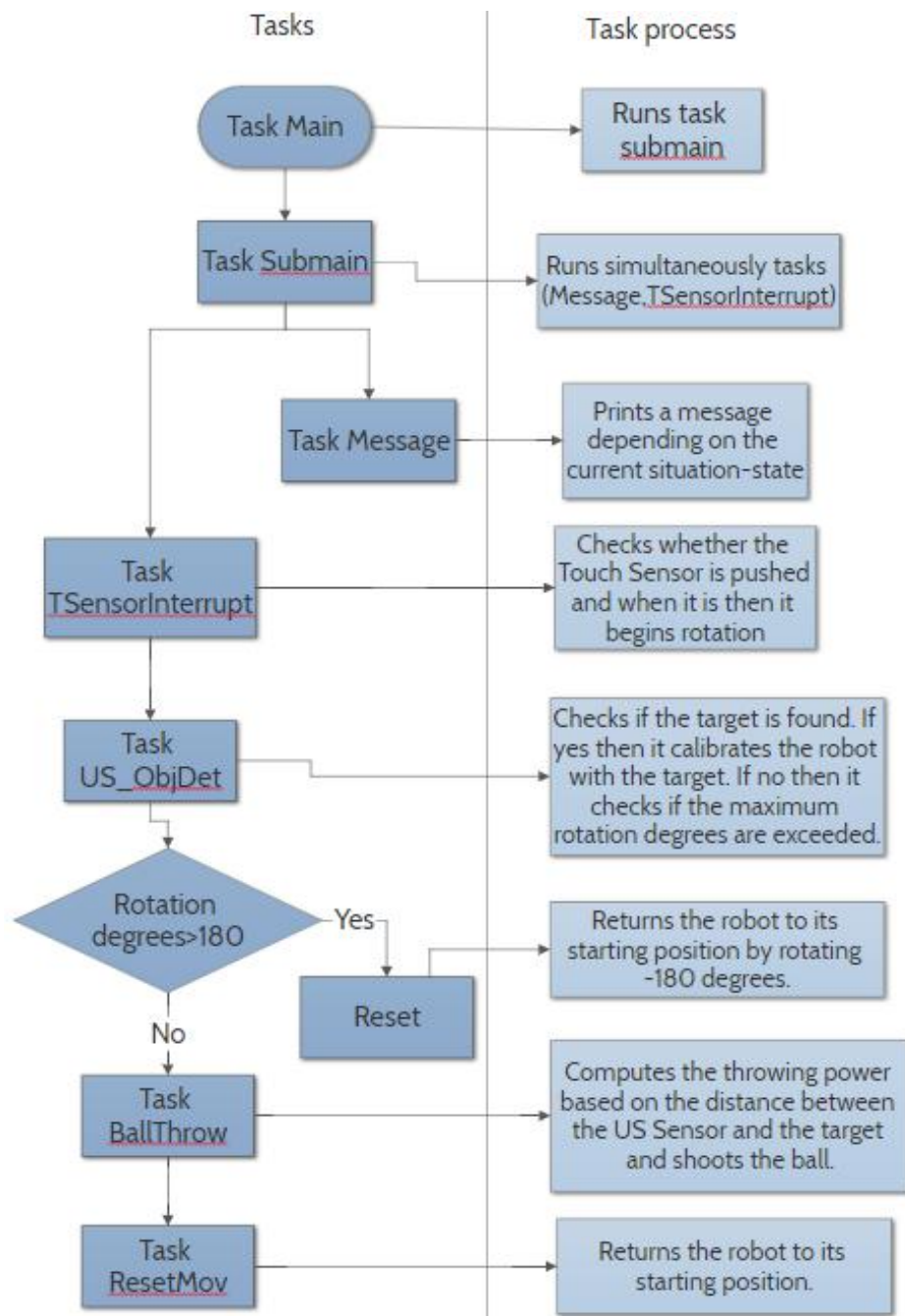
Στο προγραμματιστικό κομμάτι, εργαστήκαμε με tasks και mutexes, τα οποία εξασφάλισαν μια λογική συνέχεια στην εκτέλεση των εντολών μας και διασφάλισαν την ιδιωτική χρήση των επιμέρους compartments (πχ motors) σε κάθε χρονική στιγμή (time slot). Πιο συγκεκριμένα στην main (sub_main) συνάρτησή μας, τρέχουμε παράλληλα 2 συναρτήσεις.

Η μία εκτυπώνει στο brick το κατάλληλο μήνυμα, ανάλογα με την κατάσταση στην οποία βρίσκεται το ρομπότ. Οι καταστάσεις είναι 7:

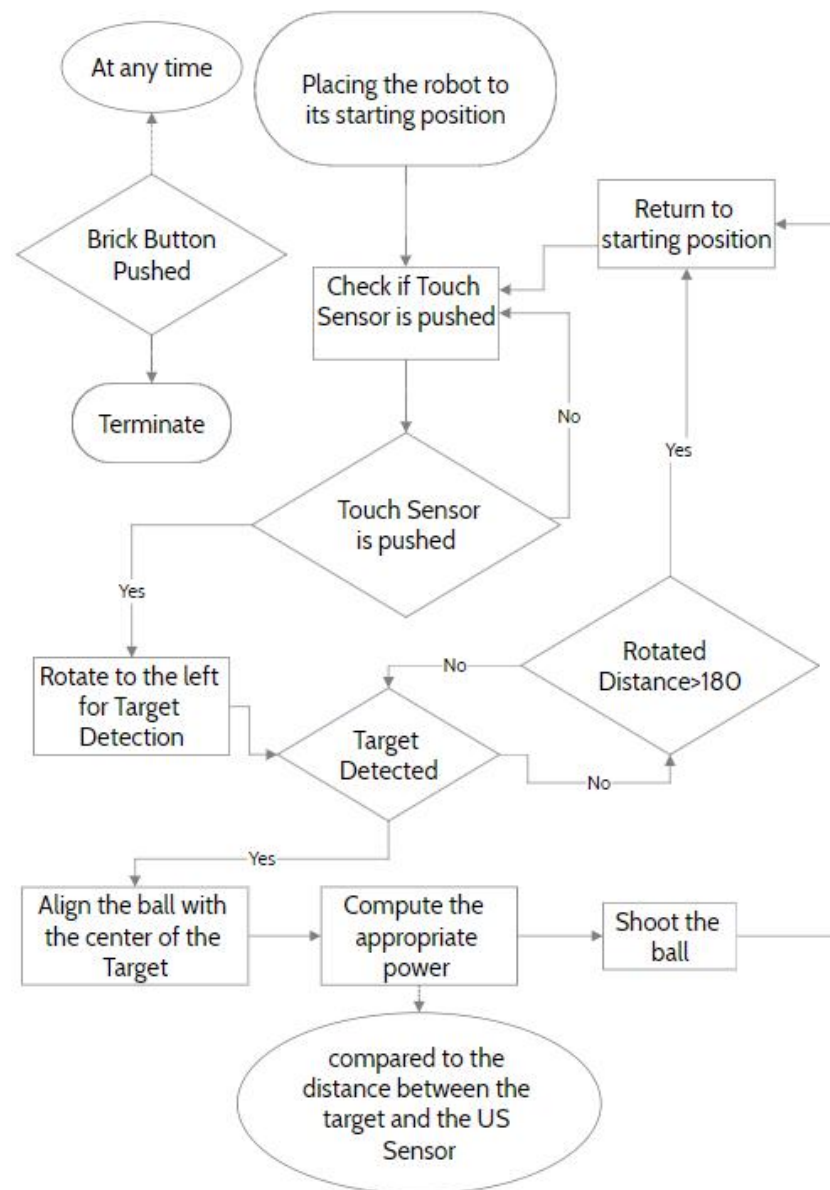
1. Αναμονή ενεργοποίησης του Touch Sensor για να ξεκινήσει το ρομπότ την αναζήτηση του στόχου.
2. Αναζήτηση του στόχου.
- [
3. Εύρεση του στόχου.
4. Ευθυγράμμιση του κινητού μέρους του ρομπότ, με το κέντρο του στόχου.
5. Υπολογισμός της απόστασης του στόχου και της δύναμης που απαιτείται να εφαρμοστεί στην μπάλα για να ευστοχήσει το ρομπότ.
6. Επαναφορά του ρομπότ στην αρχική του θέση.
-]
- Ή
7. Αδυναμία εύρεσης κάποιου στόχου και επαναφορά του ρομπότ στην αρχική του θέση.

Η κάθε μία από τις παραπάνω 7 καταστάσεις υλοποιήθηκε και σε ένα διαφορετικό task.

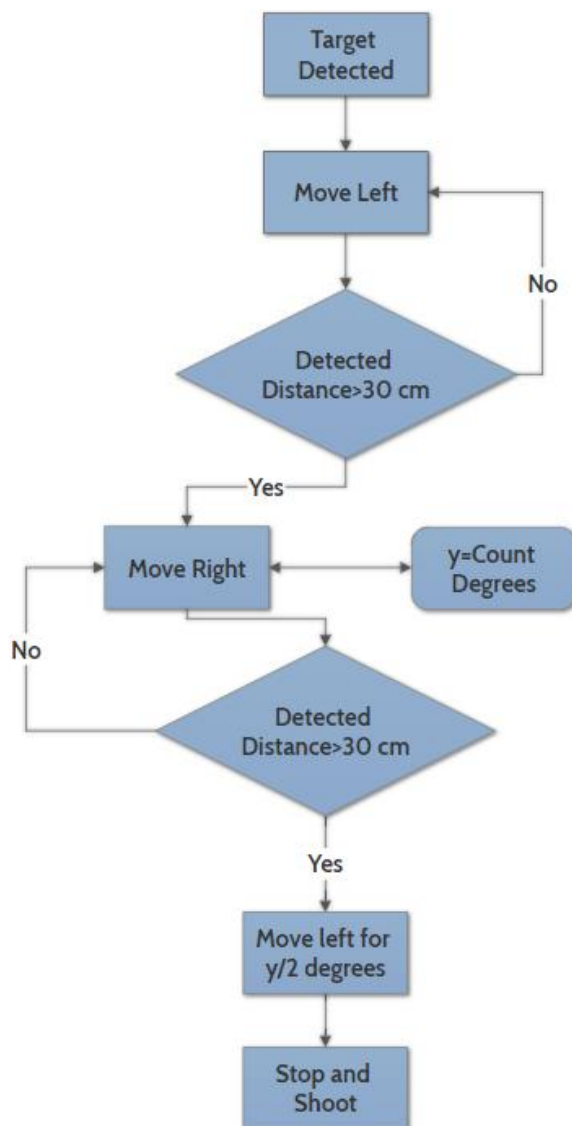
Η δεύτερη συνάρτηση αναμένει την ενεργοποίηση του Touch Sensor. Στη συνέχεια μεταβαίνουμε σε μια αλληλουχία task, τα οποία αντιστοιχίζονται με time slots, στα οποία εκτελείται η κατάλληλη ενέργεια ανάλογα με την προηγούμενη κατάσταση και τα δεδομένα εισόδου του ρομπότ. Αυτό φαίνεται πιο αναλυτικά στο παρακάτω σχήμα.



Παρακάτω φαίνεται το διάγραμμα ροής του όλου προγράμματος, το οποίο επεξηγεί τα παραπάνω βήματα.



Εκτενής αναφορά αξίζει να γίνει στον τρόπο ευθυγράμμισης του κινητού μέρους του ρομπότ, με το κέντρο του στόχου (*Align the ball with the center of the target, διεργασία του προηγούμενου διαγράμματος*). Η προσέγγιση αυτή παρουσιάζεται στο παρακάτω διάγραμμα ροής.



ΒΙΒΛΙΟΓΡΑΦΙΑ

Προκειμένου να εξοικειωθούμε με τη γλώσσα NXC, ανατρέξαμε στα Manual, τα οποία βρίσκονται στους παρακάτω συνδέσμους.

<http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html>

http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf

Το Manual για την πλατφόρμα υλοποίησης NXT Intelligent Brick.

<https://www.generationrobots.com/media/Lego-Mindstorms-NXT-Education-Kit.pdf>

Το παρακάτω link παρουσιάζει το αντίστοιχο ρομπότ του EV3 Lego Mindstorms, από το οποίο πήραμε την ιδέα.

<https://www.youtube.com/watch?v=cVMBcq6UY28>

Για τα tasks:

<http://www.robotc.net/forums/viewtopic.php?t=6897&p=26512>

<http://www.mindstorms.rwth-aachen.de/trac/browser/branches/atorf/NXC/StopRotateMotor.nxc>

To github link: <https://github.com/pitsisg/BasketballRobot/tree/master>