



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ  
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ  
ΘΕΩΡΗΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ

**Προγραμματισμός Αυτόματου Ρομποτικού Συστήματος  
για την εύρεση κατάλληλου χώρου και πραγματοποίηση  
της ευθείας και αντίστροφης διαδικασίας  
στάθμευσης**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΒΑΣΙΛΕΙΑΔΗΣ ΧΡΗΣΤΟΣ**

**Επιβλέπων:** Αν. Ι. Βαρδουλάκης  
Καθηγητής Α.Π.Θ.

Θεσσαλονίκη, Ιούλιος 2010





**Προγραμματισμός Αυτόματου Ρομποτικού Συστήματος  
για την εύρεση κατάλληλου χώρου και πραγματοποίηση  
της ευθείας και αντίστροφης διαδικασίας  
στάθμευσης**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΒΑΣΙΛΕΙΑΔΗΣ ΧΡΗΣΤΟΣ**

Επιβλέπων: Αν. Ι. Βαρδουλάκης  
Καθηγητής Α.Π.Θ.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 2η Ιουλίου 2010.

.....  
Αν. Ι. Βαρδουλάκης  
Καθηγητής Α.Π.Θ.

.....  
Ν. Καραμπετάκης  
Αν. Καθηγητής Α.Π.Θ.

.....  
Ε. Αντωνίου  
Επ. Καθηγητής Α.Τ.Ε.Ι.  
Θεσσαλονίκης

Θεσσαλονίκη, Ιούλιος 2010

---

.....  
Βασιλειάδης Χρήστος  
Πτυχιούχος Μαθηματικός Α.Π.Θ.

Copyright © Βασιλειάδης Χρήστος, 2010.  
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευτεί ότι εκφράζουν τις επίσημες θέσεις του Α.Π.Θ.

---

## ΠΕΡΙΛΗΨΗ

Η εκτέλεση πολλών εργασιών την σημερινή εποχή πραγματοποιείται με την βοήθεια «Αυτόματων Ρομποτικών συστημάτων» που είναι προικισμένα με ποικίλες ικανότητες. Όλο και περισσότερες εργασίες που πριν κάποια χρόνια φάνταζαν ως ακατόρθωτες σήμερα πραγματοποιούνται σε ελάχιστο χρόνο και με μικρό κόστος με τη βοήθεια αυτών των διατάξεων και της «τεχνητής νοημοσύνης» που κατέχουν.

Η εργασία αυτή βασίζεται σε ένα «Αυτόματο Ρομποτικό Σύστημα» το οποίο θα έχει την ικανότητα να εποπτεύει τον χώρο και με την «τεχνητή νοημοσύνη» που διαθέτει θα μπορεί να αναγνωρίζει τον χώρο αυτό ως κατάλληλο ή μη για στάθμευση. Εφόσον ο χώρος κριθεί ως κατάλληλος, μετά τις απαραίτητες μετρήσεις, το παραπάνω ρομποτικό σύστημα θα μπορεί να εκτελέσει με ακρίβεια την διαδικασία στάθμευσης. Επιπλέον μετά την ολοκλήρωση της παραπάνω ενέργειας και εφόσον έχει σταθμεύσει θα μπορεί με την χρήση φωνητικής εντολής να εκτελέσει την αντίστροφη διαδικασία αυτόματα και με ακρίβεια.

Στο 1<sup>ο</sup> κεφάλαιο γίνεται μια σύντομη ιστορική αναδρομή για τα ρομποτικά συστήματα καθώς επίσης και μια αναφορά για την τεχνολογία γύρω από την οποία υλοποιούνται. Επιπρόσθετα πραγματοποιείται και μια εισαγωγή στη γλώσσα προγραμματισμού NXT-G που θα χρησιμοποιηθεί για την κατασκευή του αλγόριθμου.

Στη συνέχεια στο 2<sup>ο</sup> κεφάλαιο γίνεται η παρουσίαση της κατασκευής βήμα προς βήμα του «Αυτόματου Ρομποτικού Συστήματος» που παρουσιάστηκε παραπάνω.

Τέλος στο 3<sup>ο</sup> κεφάλαιο παρουσιάζεται αναλυτικά η διαδικασία κατασκευής του αλγόριθμου με τον οποίο υλοποιείται το project με χρήση της γλώσσας προγραμματισμού LEGO MINDSTORMS NXT-G.

## ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Ρομπότ, LEGO MINDSTORMS NXT, Τουβλάκια, Μπλόκ, Μπλόκ επανάληψης, Μότορας, Αισθητήρας Φωτός, Αισθητήρας Ήχου, Υπερηχητικός αισθητήρας, Αισθητήρας Επαφής, Αισθητήρας Περιστροφής.

---

## ABSTRACT

The implementation of many tasks in the latest years is being accomplished with the aid of “Automatic Robotic Systems” that are dowered with various faculties. Projects that used to be unachievable in the past years can now be accomplished in minimal time and with small cost with the aid of these provisions and their “artificial intelligence”

This task is based on an “Automatic Robotic System” that has the ability to oversee the space and with the “artificial intelligence” that it allocates it shall recognize this space as suitable or not for quartering. With the assumption the space is tagged as suitable, and after the essential measurements the robotic system will be able to execute with precision the process of quartering. Moreover, after the completion of this task and provided that it has parked, the robotic system with the use of a sound signal will have the ability to execute the reverse process automatically and with precision

In the 1<sup>o</sup> chapter a short historical retrospection for the robotic systems is made as well as a report on the technology with which they are materialised. Moreover a first introduction in Mindstorms NXT-G is made.

Afterwards in the 2<sup>o</sup> chapter is being accomplished the presentation of the building instructions of such a robotic system step by step.

Finally in the 3<sup>o</sup> chapter is presented analytically the process of manufacturing the algorithm that the robotic provision is making use to accomplish its tasks. This algorithm is written in the LEGO MINDSTORMS NXT-G language.

## KEY WORDS

Robot, LEGO MINDSTORMS NXT, Bricks, Block, Loop, Motor, Light Sensor, Sound Sensor, Ultrasonic Sensor, Touch Sensor, Rotation Sensor.

---

**Ευχαριστίες προς  
τον επιβλέποντα καθηγητή μου, κ. Αντώνη Βαρδουλάκη  
και την οικογένεια μου**

---

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	5
ABSTRACT.....	6
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	8
ΠΡΟΛΟΓΟΣ.....	9
ΚΕΦΑΛΑΙΟ 1.....	11
1.1 Ρομποτική.....	11
1.2 Είδη Ρομπότ.....	20
1.3 Οργάνωση και λειτουργία.....	24
1.4 Ρομποτική στην καθημερινή ζωή του ανθρώπου.....	26
1.5 LEGO Mindstorms.....	33
ΚΕΦΑΛΑΙΟ 2.....	41
Παρουσίαση κατασκευής αυτόματου ρομποτικού συστήματος για την εύρεση κατάλληλου χώρου και πραγματοποίηση της ευθείας και αντίστροφης διαδικασίας στάθμευσης.....	41
ΚΕΦΑΛΑΙΟ 3.....	69
Κατασκευή αλγόριθμου.....	69
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	95



---

## ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος σπουδών “ΘΕΩΡΗΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ ” κατά το ακαδημαϊκό έτος 2009-2010.

Θα ήθελα να ευχαριστώ τον επιβλέποντα καθηγητή μου κ. Βαρδουλάκη για τις γνώσεις που μου μετέδωσε κατά τη διάρκεια του μεταπτυχιακού καθώς επίσης και τον κ. Καραμπετάκη για την βοήθεια που μου προσέφερε με τις γνώσεις του πάνω στο συγκεκριμένο κλάδο της ρομποτικής. Επιπλέον ένα μεγάλο ευχαριστώ στην κ.Γουσίδου για την ένθερμη υποστήριξη της ιδέας για την συγκεκριμένη υλοποίηση.

Τέλος θα ήθελα να ευχαριστήσω τους καθηγητές μου για την άρτια συνεργασία που είχαμε κατά τη διάρκεια των μεταπτυχιακών σπουδών μου στο τμήμα μαθηματικών του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης καθώς επίσης και για τα απαραίτητα εφόδια και γνώσεις με τα οποία με εφοδίασαν.



---

## ΚΕΦΑΛΑΙΟ 1

### 1.1 Ρομποτική

#### ΕΙΣΑΓΩΓΗ

Από την απαρχή της σύλληψης ενός μηχανοκίνητου, αυτόνομου όντος, ο άνθρωπος είχε παράξενη σχέση με το τεχνητό δημιούργημα του, άλλοτε γεμάτη θαυμασμό, άλλοτε γεμάτη δέος, ίσως επειδή το συνέδεε εκ προοιμίου με την προσφορά υπηρεσιών στον ίδιο. Γι' αυτό και το ονόμασε έτσι, από ένα τσέχικο ουσιαστικό που δηλώνει υπηρετή (robota). Παρά τον υποβόσκοντα φόβο ότι το δημιούργημά του θα αποκτήσει δικό του συναίσθημα και θα τον καθυποτάξει, ο άνθρωπος έχει εντάξει τα σημερινά ρομπότ στη βιομηχανία και τελικά στο οικιακό περιβάλλον του.

Τα πιο προηγμένα σημερινά ρομπότ αποτελούν ευφυή αυτόνομα συστήματα δηλαδή, διαθέτουν ένα είδος σκέψης, ώστε να μην επαναλαμβάνουν μηχανικά μια κίνηση, να μπορούν να αντιληφθούν το περιβάλλον τους και να λάβουν αποφάσεις χωρίς την παρέμβαση ενός χρήστη. Πολλά ρομπότ αξιοποιούνται σε εφαρμογές στις οποίες ο ανθρώπινος παράγοντας δυσκολεύεται, είτε λόγω δυσχερών συνθηκών εργασίας είτε λόγω επικινδυνότητας (στρατιωτικές εφαρμογές) είτε λόγω κρισιμότητας (ιατρικές εφαρμογές). Ακόμα και τα οικιακά ρομπότ μοιράζονται τις ίδιες ιδιότητες με ένα βιομηχανικό ρομπότ. Κινούνται, δηλαδή, χρησιμοποιούν κάποιο όργανο δράσης για να αλληλεπιδρούν με το περιβάλλον, αντιλαμβάνονται, δηλαδή, εξομοιώνουν ανθρώπινες αισθήσεις μέσω των δικών τους αισθητήρων, διαθέτουν μια μονάδα ελέγχου και μπορούν να επικοινωνούν, μετατρέποντας κείμενο σε φωνή, αναγνωρίζοντας πρότυπα ήχου και εικόνας.

Εκτός ελαχίστων εξαιρέσεων, τα οικιακά ρομπότ αλλά κυρίως και τα βιομηχανικά, στερούνται τεχνητής νοημοσύνης. Αποτελούν συστήματα κλειστού βρόχου, περικλείουν δηλαδή μηχανισμούς ανάδρασης. Το περιβάλλον τους θεωρείται κλειστό και αν συναντήσουν αλλαγή, διαμορφώνουν τη συμπεριφορά τους μέσω ήδη προγραμματισμένων οδηγιών.

Επιδιωκόμενο από όλους τους ενδιαφερόμενους με την ρομποτική είναι το να δώσουν στο ρομποτικό τους σύστημα όσο το δυνατόν πληρέστερη αίσθηση του περιβάλλοντος του. Αισθητήρες ήχου, επαφής, φωτός χρώματος κ.α. είναι αρωγοί στην ανάπτυξη τεχνητής νοημοσύνης από τέτοιες ρομποτικές διατάξεις.

---

## Ορισμοί και Ιστορικά Στοιχεία

Η Ρομποτική είναι εκείνος ο κλάδος της επιστήμης του μηχανικού που ασχολείται με τη σύλληψη, το σχεδιασμό, την κατασκευή και τη λειτουργία ρομπότ. Τα ρομπότ είναι μηχανές, η χρήση των οποίων αποσκοπεί στην αντικατάσταση του ανθρώπου στην εκτέλεση έργου. Η αντικατάσταση αυτή αφορά τόσο στο φυσικό επίπεδο του έργου όσο και στο επίπεδο λήψης απόφασης.

Αναζητώντας κανείς τις ρίζες της ρομποτικής θα οδηγηθεί αρκετά πίσω στην ιστορία της ανθρωπότητας. Πράγματι, η φιλοδοξία του ανθρώπου να δημιουργήσει μηχανές που θα του μοιάζουν τόσο στη μορφή όσο και τη λειτουργία πρωτοσυναντάται στην ελληνική μυθολογία. Σύμφωνα με την τελευταία ο τιτάνας Προμηθέας έπλασε την ανθρωπότητα από πηλό. Επιπλέον ο Τάλος, ο μυθικός χάλκινος γίγαντας που κατασκεύασε ο Ήφαιστος για να προστατεύει την Κρήτη από τους εισβολείς, αποτελεί το πρώτο «αυτόματο σύστημα» στην ανθρώπινη ιστορία.

Στη σύγχρονη εποχή, η εισαγωγή της έννοιας των ρομπότ έγινε το 1921 από τον Τσέχο θεατρικό συγγραφέα Karel Capek με το θεατρικό έργο “Rossum’s Universal Robots”. Στο τελευταίο ο συγγραφέας φαντάζεται ένα μηχανικό κατασκεύασμα, το οποίο και ονομάζει robot από την τσέχικη λέξη robota για την καταναγκαστική εργασία.

Λίγα χρόνια αργότερα, κατά τη δεκαετία του 1940, ο ρώσος συγγραφέας επιστημονικής φαντασίας Isaac Asimov συνέλαβε το robot ως ένα «αυτόματο σύστημα» με εμφάνιση ανθρώπου, αλλά απαλλαγμένο από συναισθήματα. Η συμπεριφορά του υπαγορευόταν από ένα «μυαλό» προγραμματισμένο από τον άνθρωπο κατά τέτοιο τρόπο ώστε να ανταποκρίνεται σε συγκεκριμένες αρχές ηθικής συμπεριφοράς. Ο όρος ρομποτική χρησιμοποιήθηκε από τον Asimov ως το σύμβολο της επιστήμης που είναι αφιερωμένη στη μελέτη των ρομπότ και διέπονται από τους παρακάτω τρεις βασικούς νόμους:

- 1. Ένα ρομπότ δεν μπορεί να τραυματίσει ή μέσω της αδράνειάς του να βλάψει ένα ανθρώπινο πλάσμα.**
- 2. Ένα ρομπότ πρέπει να υπακούει στις εντολές που δίνονται από τους ανθρώπους, εκτός και αν αυτό έρχεται σε αντίθεση με τον πρώτο νόμο.**
- 3. Ένα ρομπότ πρέπει να προστατεύει την ίδια του την ύπαρξη, εκτός και αν αυτό έρχεται σε αντίθεση με τον πρώτο ή τον δεύτερο νόμο.**

---

## Ορισμός του Ρομπότ

Σύμφωνα με το Robot Institute of America, ως ρομπότ μπορούμε να ορίσουμε ένα μηχανισμό σχεδιασμένο ώστε, μέσω προγραμματιζόμενων κινήσεων, να μεταφέρει υλικά, τεμάχια, εργαλεία ή ειδικευμένες συσκευές με σκοπό την επιτέλεση ποικιλίας εργασιών.

Ένας τέτοιος μηχανισμός περιλαμβάνει συνήθως τις ακόλουθες συνιστώσες:

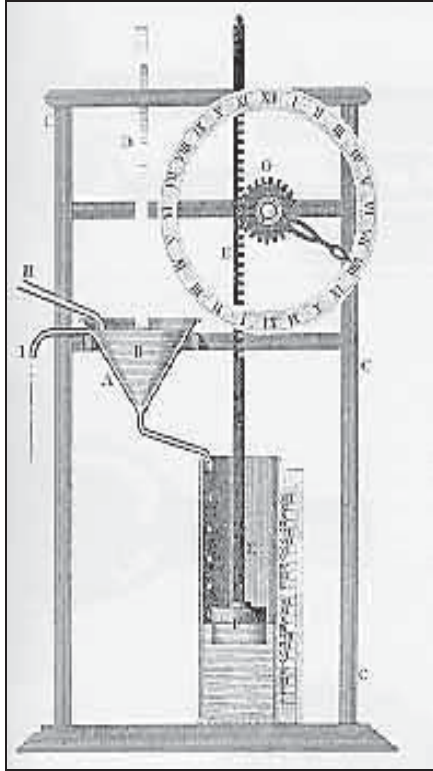
- Ένα μηχανολογικό υποσύστημα, το οποίο ενσωματώνει τη δυνατότητα του ρομπότ για εκτέλεση έργου. Το υποσύστημα αυτό αποτελείται από μηχανισμούς που επιτρέπουν στο ρομπότ να κινείται όπως αρθρώσεις, συστήματα μετάδοσης κίνησης, κινητήρες, οδηγούς κλπ..

- Ένα υποσύστημα αίσθησης, μέσω του οποίου το ρομπότ συγκεντρώνει πληροφορίες για την κατάσταση στην οποία βρίσκονται τόσο το ίδιο όσο και το περιβάλλον. Το υποσύστημα αυτό εκτός των άλλων είναι υπεύθυνο για την αποδοχή των εξωτερικών εντολών, την επεξεργασία τους, τη μετάφρασή τους σε ηλεκτρική ισχύ που θα δοθεί στους κινητήρες του ρομπότ, καθώς επίσης και για την παραγωγή σημάτων εξόδου που θα πληροφορούν για την κατάσταση του συστήματος. Στο υποσύστημα αίσθησης περιλαμβάνονται όργανα μετρήσεως, αισθητήρες, ηλεκτρονικά στοιχεία κλπ..

- Ένα σύστημα ελέγχου, το οποίο συνδυάζει κατάλληλα την αίσθηση με τη δράση, έτσι ώστε το ρομπότ να λειτουργεί αποτελεσματικά και με τον επιθυμητό τρόπο. Ο ελεγκτής του ρομπότ επιβλέπει και συντονίζει ολόκληρο το σύστημα, για τη σχεδίαση και υλοποίησή του δε απαιτείται ο συνδυασμός γνώσεων από πολλές γνωστικές περιοχές, όπως είναι ο αυτόματος έλεγχος, η τεχνητή νοημοσύνη, η επιστήμη των υπολογιστών κλπ..

## Ιστορία των Ρομπότ

Παρακάτω θα πραγματοποιήσουμε μια σύντομη ιστορική αναφορά στα σημαντικότερα ρομποτικά συστήματα που κατασκεύασε ο άνθρωπος.



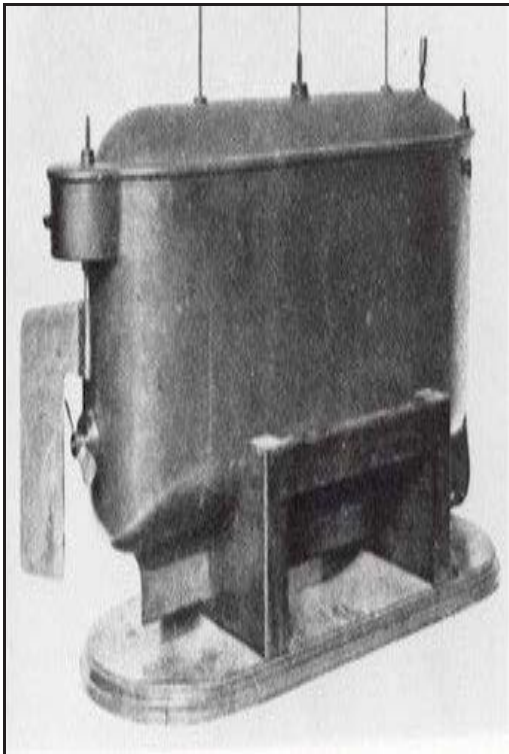
Το 250 π.Χ. ο Κτησίβιος της Αλεξάνδρειας κατασκευάζει ρολόγια με κινητά στοιχεία τα οποία λειτουργούν με την δύναμη του νερού.



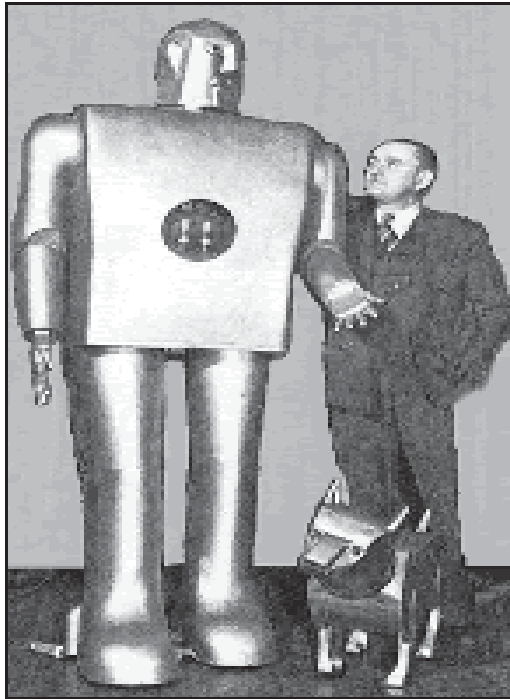
Το 1445 μ.Χ. ο Leonardo da Vinci πιθανολογείται πως σχεδίασε και κατασκεύασε το πρώτο ανθρωποειδές ρομπότ. Το ρομπότ υποτίθεται πως μπορούσε να κινεί τα χέρια του καθώς και να περιστρέφει το κεφάλι του γύρω από τον λαιμό του.



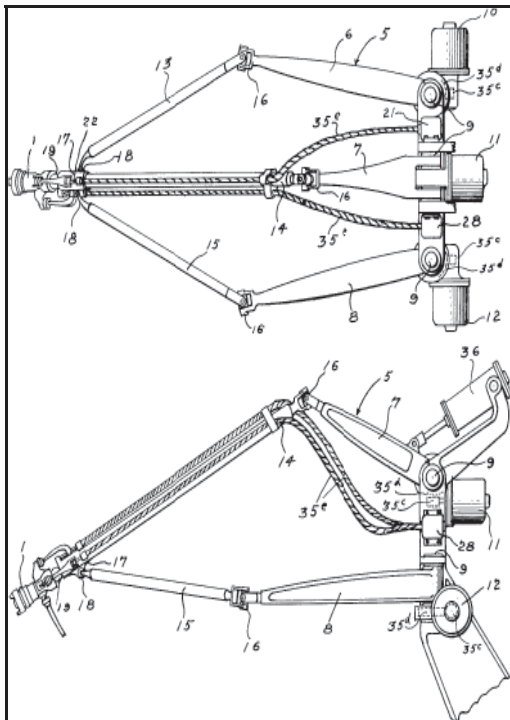
Το 1801 μ.Χ. ο Joseph Jacquard εφευρίσκει την πρώτη υφαντική μηχανή



Το 1897 μ.Χ. ο Nikola Tesla δημιούργησε το πρώτο υποβρύχιο τηλεκατευθυνόμενο σκάφος.

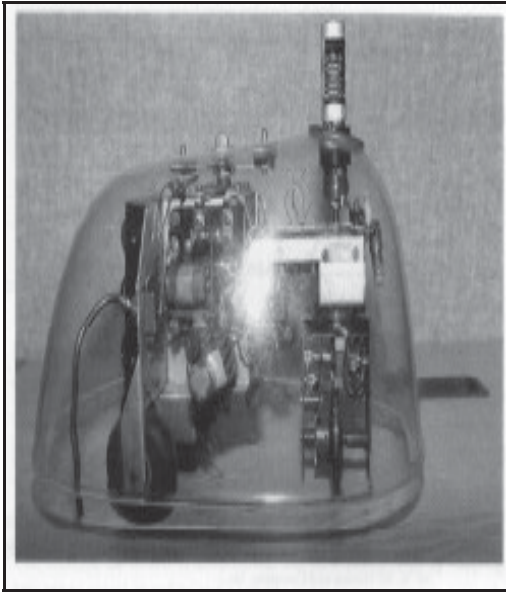


Το 1937 μ.Χ. ο Westinghouse δημιουργεί τον ELEKTRO ένα ρομπότ με ανθρώπινη μορφή που μπορεί να μιλά, να περπατά και να καπνίζει.

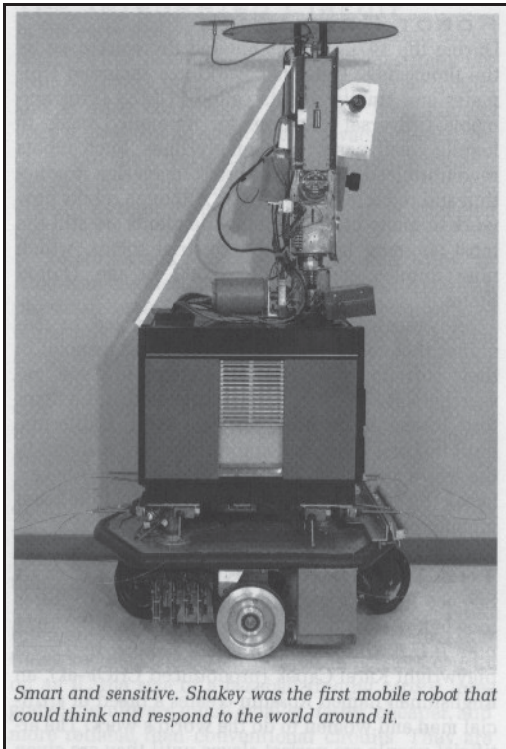


Το 1942 μ.Χ. κατασκευάζεται ο πρώτος προγραμματιζόμενος μηχανισμός ψεκασμού χρωμάτων από τους Willard Pollard και Harold Roselund για την εταιρεία Devilbiss





Το 1949 μ.Χ. ο W. Grey Walter δημιουργεί τα πρώτα του ρομπότ. Τα Elmer και Elsie είναι ικανά να εντοπίσουν σταθμούς ενέργειας όταν τα επίπεδα ενέργειάς τους είναι χαμηλά.



Το 1970 μ.Χ. δημιουργείται το πρώτο ρομπότ που ελέγχεται με τεχνητή νοημοσύνη

*Smart and sensitive. Shakey was the first mobile robot that could think and respond to the world around it.*



Το 1984 μ.Χ. κατασκευάζεται το πρώτο ρομπότ που μπορεί να διαβάσει μουσική και να παίζει ένα μουσικό όργανο.



Το 1996 μ.Χ. κατασκευάστηκε το RoboTuna το οποίο χρησιμοποιήθηκε για να μελετηθεί η κίνηση των ψαριών.



Το 1998 μ.Χ. κατασκευάστηκε το Edinburg Modular το πρώτο βιονικό χέρι.



Το 1999 μ.Χ. η Mitsubishi δημιουργεί ένα ψάρι ρομπότ. Πρόθεση της εταιρίας είναι να δημιουργήσει ρομποτικά ψάρια των ειδών που έχουν εξαφανιστεί.

## 1.2 Είδη Ρομπότ

Κατά την πολυετή εξέλιξη της επιστήμης της ρομποτικής προέκυψαν διάφορα είδη ρομποτικών μηχανισμών, οι οποίοι διαφέρουν σημαντικά στη μορφή, αποτελούνται όμως από αντίστοιχα επιμέρους υποσυστήματα. Αυτά είναι το μηχανολογικό υποσύστημα, το υποσύστημα αίσθησης και το σύστημα ελέγχου.

Τα σπουδαιότερα είδη ρομπότ είναι τα παρακάτω:

- **Ρομπότ Σταθερής Βάσης:** τα ρομπότ αυτά αποτελούνται από διαδοχικά στερεά σώματα (σύνδεσμοι) που συνδέονται μέσω αρθρώσεων σχηματίζοντας μία κινηματική αλυσίδα. Η αλυσίδα αυτή έχει το ένα άκρο της (βάση) σταθερά συνδεδεμένο με κάποιο σημείο του περιβάλλοντος χώρου. Η μορφή αυτή ρομπότ είναι η παραδοσιακή μορφή ενός βιομηχανικού ρομποτικού βραχίονα, και περιλαμβάνει το βραχίονα, τον καρπό και το εργαλείο (Σχήμα 1).

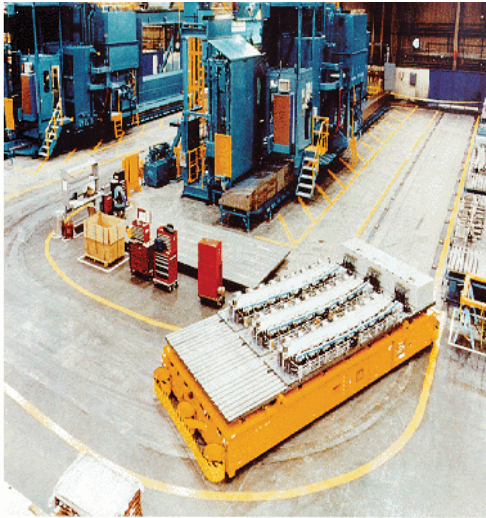


Σχήμα 1 Ο Βιομηχανικός Ρομποτικός Βραχίονας PUMA 500 της Unimation Inc.

---

• **Κινούμενα Ρομπότ:** ως κινητά ρομπότ χαρακτηρίζονται όλα εκείνα τα ρομπότ που έχουν τη δυνατότητα να μετακινήσουν όλα τα σημεία του μηχανισμού τους. Η δυνατότητα αυτή προσφέρεται από ειδικά συστήματα προώθησης, τα οποία μπορεί να είναι είτε απλά (όπως τροχοί) είτε πολύπλοκα (όπως jet, προπέλες, μηχανικά πόδια). Τα κινούμενα ρομπότ διακρίνονται σε επιμέρους κατηγορίες ανάλογα με το βαθμό αυτονομίας τους. Έτσι έχουμε:

• **AGVs:** τα AGVs (Automatic Guided Vehicles) έχουν περιορισμένη αυτονομία κίνησης, δεδομένου ότι η τροχιά τους είναι προκαθορισμένη μέσω καλωδίων στο έδαφος ή πομπών στον περιβάλλοντα χώρο (Σχήμα 2).



Σχήμα 2 AGV σε Βιομηχανικό Περιβάλλον

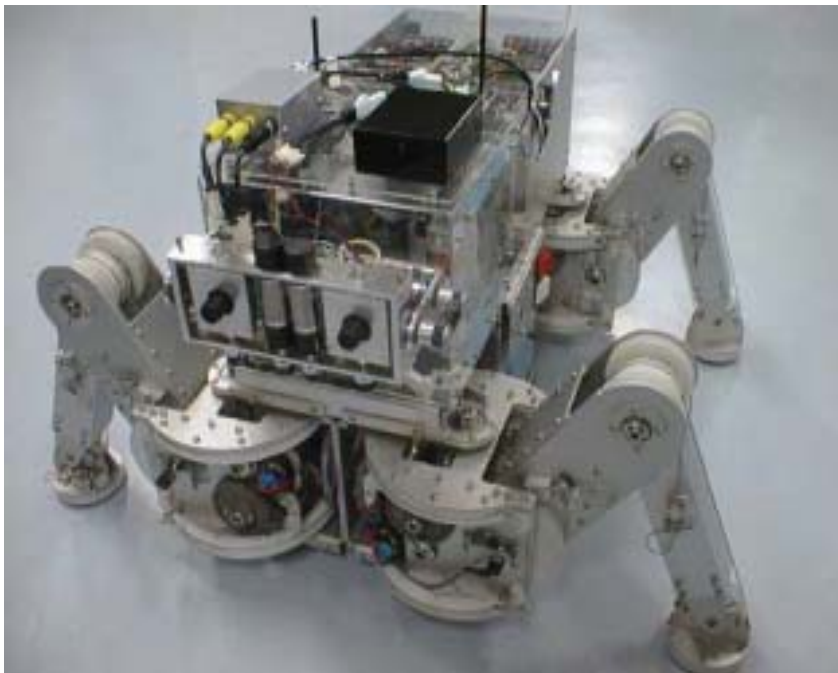
• **Αυτόνομα Έντροχα Ρομπότ:** τα ρομπότ αυτά λειτουργούν με αρκετά υψηλό βαθμό αυτονομίας. Πιο συγκεκριμένα μπορούν και λειτουργούν χωρίς συνεχή εξωτερική επίβλεψη και είναι ικανά να εκτελούν εργασίες αυτόνομα δεχόμενα μόνο ορισμένες υψηλού επιπέδου εντολές (Σχήμα 3).



Σχήμα 3 Αυτόνομο Έντροχο Ρομπότ

---

• **Βαδίζοντα Ρομπότ:** τα ρομπότ αυτά χρησιμοποιούν μηχανικά πόδια για την κίνησή τους και όχι συμβατικούς τροχούς όπως στις προηγούμενες δύο κατηγορίες. Τα κυριότερα πλεονεκτήματα της συγκεκριμένης υλοποίησης είναι η μεγάλη δυνατότητα αποφυγής εμποδίων και η ικανότητα αναρρίχησης σε ανώμαλα εδάφη και μη επίπεδες επιφάνειες . Από τα πιο συνηθισμένα ρομπότ αυτής της κατηγορίας είναι τα δίποδα ενώ δεν αποκλείονται και εφαρμογές με περισσότερα από δύο πόδια (Σχήμα 4)



Σχήμα 4 Ρομπότ αυτόνομο με μηχανικά πόδια

• **Εναέρια ρομπότ:** πρόκειται για μη επανδρωμένα ιπτάμενα ρομπότ, όπως ελικόπτερα και αεροπλάνα. Τα ρομπότ αυτά έχουν διαρκώς αυξανόμενες εφαρμογές, όμως εξαιτίας της μειωμένης ακόμα σταθερότητας και ασφάλειας στη συμπεριφορά τους χρησιμοποιούνται για στρατιωτικούς κυρίως σκοπούς (Σχήμα 5)



Σχήμα 5 Το ρομπότ HELIOS της NASA

• **AUVs:** τα AUVs (Autonomous Underwater Vehicles) είναι πλήρως αυτόνομα και κατά συνέπεια δεν έχουν την ανάγκη καλωδίου. Για τις ανάγκες τροφοδοσίας (ενέργεια) χρησιμοποιούνται ειδικές μπαταρίες, κάτι όμως που θέτει και περιορισμούς στη λειτουργία των ρομπότ αυτών. Τα AUVs έχουν σχήμα τορπιλών και μπορούν να κινούνται με αρκετά μεγάλες ταχύτητες (Σχήμα 6).



Σχήμα 6 Αυτόνομο υποβρύχιο ρομποτικό σύστημα

---

### 1.3 Οργάνωση και λειτουργία

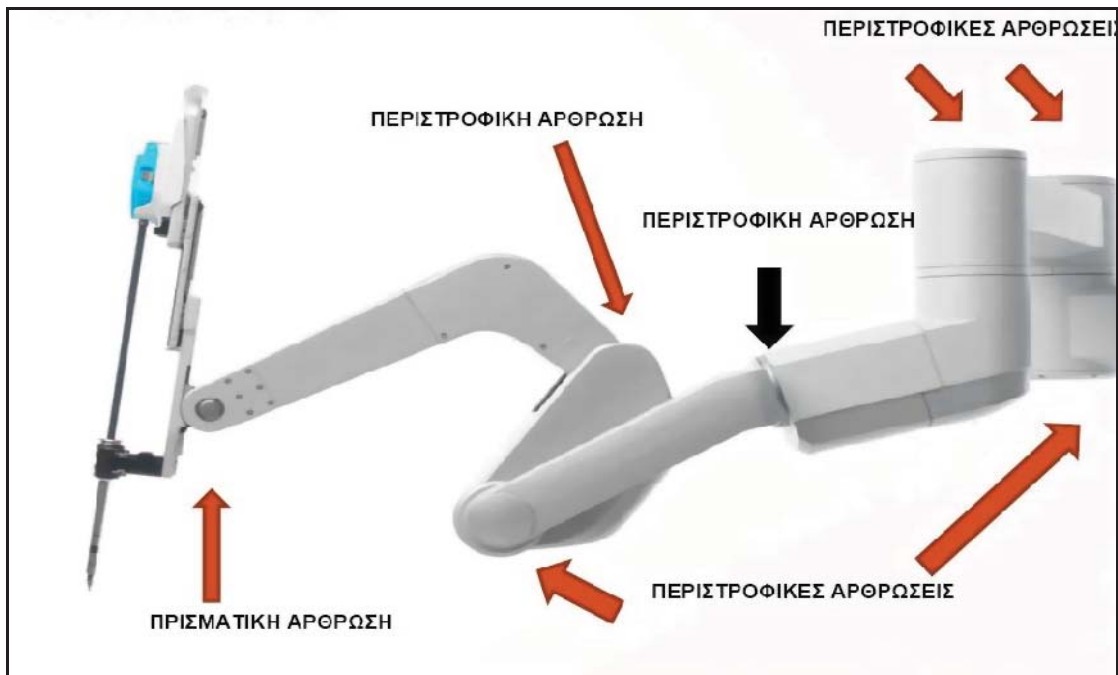
Τα στοιχεία που αποτελούν ένα Ρομποτικό Σύστημα είναι το Μηχανικό μέρος και ο Ελεγκτής.

#### Μηχανικό μέρος

Το μηχανικό μέρος σε ένα ρομποτικό σύστημα αποτελεί ο βραχίονας. Ο βραχίονας αποτελείται από:

- **Βάση:** Η βάση είναι στερεωμένη στο περιβάλλον εργασίας του ρομπότ. Στη βάση είναι συνδεδεμένη αλυσίδα αρθρώσεων – συνδέσμων που καταλήγει στο εργαλείο τελικής δράσης.
- **Σύνδεσμοι:** Οι σύνδεσμοι είναι στερεά σώματα που αποτελούν το σκελετό του ρομπότ.
- **Αρθρώσεις:** Οι αρθρώσεις είναι μηχανισμοί που επιτρέπουν τη σχετική κίνηση μεταξύ των συνδέσμων. Κύριες κατηγορίες αρθρώσεων είναι οι στροφικές και οι πρισματικές. Οι στροφικές αρθρώσεις επιτρέπουν σχετική στροφή μεταξύ δύο συνδέσμων, ενώ οι πρισματικές αρθρώσεις επιτρέπουν σχετική μετατόπιση – σε ευθεία γραμμή – μεταξύ δύο γειτονικών συνδέσμων.
- **Κινητήρες:** Κάθε άρθρωση χρειάζεται και από ένα κινητήρα για να εκτελεί τις λειτουργίες της.
- **Αισθητήρες:** Για να ελέγχουμε τη θέση του ρομπότ χρειαζόμαστε πληροφορίες για την θέση και την ταχύτητα της κάθε άρθρωσης. Έτσι χρησιμοποιούνται διάφοροι τύποι αισθητήρων, όπως αισθητήρες αφής, ήχου κ.α.
- **Εργαλείο τελικής δράσης:** Όλοι οι βραχίονες έχουν προσαρμοσμένο στο άκρο τους ένα μηχανικό εξάρτημα κατάλληλα σχεδιασμένο και επιλεγμένο, προκειμένου να εκτελούν την εργασία για την οποία έχουν προγραμματιστεί.





## Ελεγκτής

Ο ελεγκτής είναι η μονάδα που μας δίνει τη δυνατότητα να προγραμματίσουμε το ρομπότ και ο οποίος ελέγχει την κίνηση του και την εκτέλεση της εργασίας του. Ο ελεγκτής αποτελείται από:

- **Ηλεκτρονικά (Hardware):** Αυτά είναι συνήθως ένας υπολογιστής, όπου αποθηκεύεται το πρόγραμμα που θα εκτελεστεί όπως και τα ηλεκτρονικά επικοινωνίας που χρησιμεύουν στην επικοινωνία του ελεγκτή με το μηχανικό μέρος του ρομπότ και το εξωτερικό περιβάλλον.

- **Λογισμικό (Software):** Το λογισμικό κυρίως ευθύνεται για τη δημιουργία των κατάλληλων σημάτων ελέγχου, σύμφωνα με κάποιον αλγόριθμο, παίρνοντας υπόψη του το φορτίο, την ταχύτητα, την θέση του ρομπότ και άλλες μεταβλητές. Επίσης, στο λογισμικό περιλαμβάνεται και κάποιο βοηθητικό πρόγραμμα που επιτρέπει τον προγραμματισμό του ρομπότ σε μία γλώσσα υψηλού επιπέδου.

---

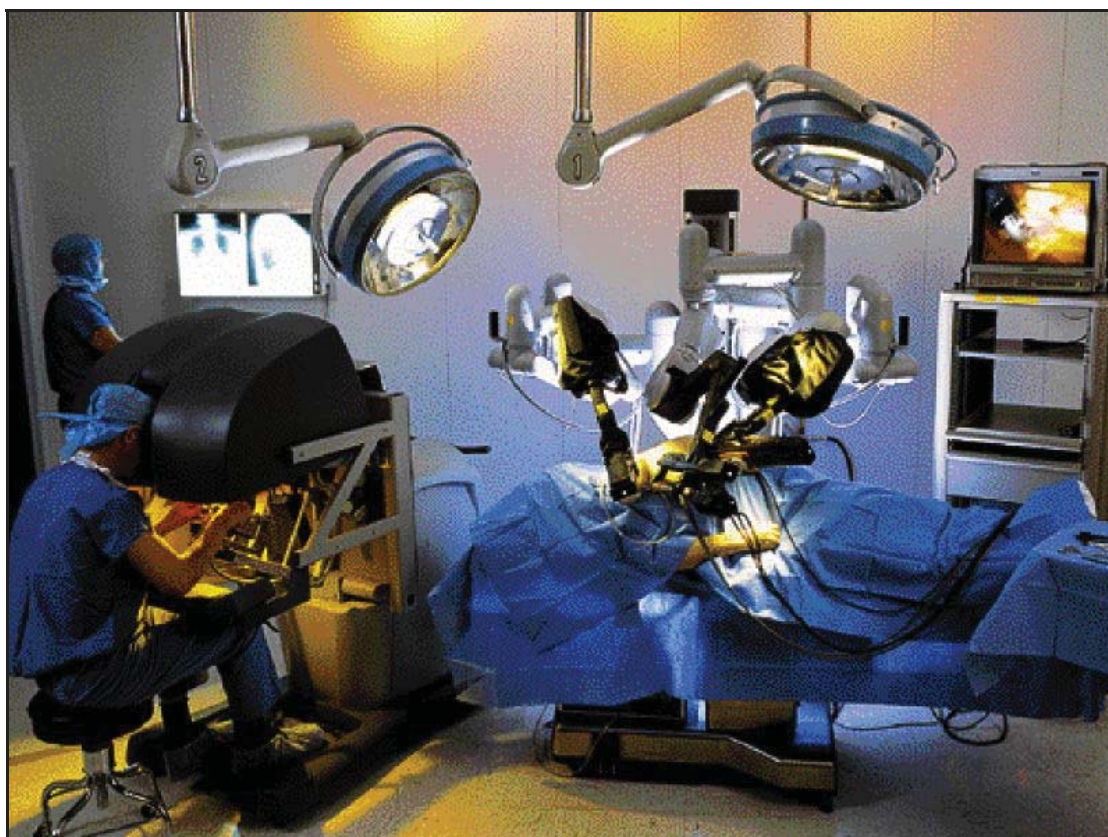
## 1.4 Η Ρομποτική Στην Καθημερινή Ζωή του Ανθρώπου

Η ανάγκη για χρήση αυτόματων ρομποτικών συστημάτων στην καθημερινή ζωή του ανθρώπου οδηγείται πρωτίστως από το χαμηλό κόστος διεκπεραίωσης μιας εργασίας αλλά και από την έλλειψη εξειδικευμένου προσωπικού. Υπάρχουν δηλαδή εφαρμογές άλλοτε απλές και άλλοτε πιο σύνθετες που απαιτούν τη χρήση τέτοιων συστημάτων. Μερικές από τις εφαρμογές αυτές είναι οι παρακάτω:

### Ρομποτική για την δημιουργία εικονικής πραγματικότητας



## Ρομποτική ιατρική



Το διάσημο ρομποτικό θαύμα της ιατρικής Da Vinci

---

Ρομποτικά συστήματα για άτομα με ειδικές ανάγκες



Το σύστημα **HANDY 1**



Η συσκευή **HANDY 1** σε λειτουργία



Το μηχανικό χέρι



Αναπηρικό καροτσάκι για άτομα με ειδικές ανάγκες

---

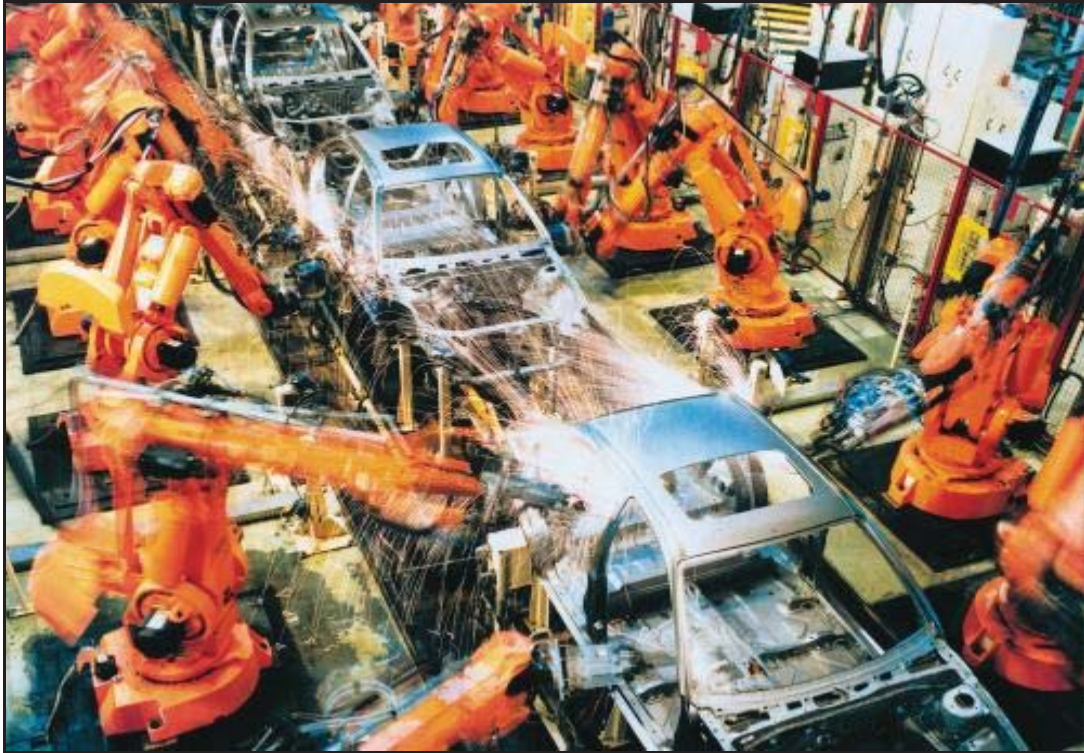
## Ρομποτική στο στρατό



Ρομποτικά συστήματα για παροχή υπηρεσιών στο στρατό

---

## Ρομποτική και Βιομηχανική Παραγωγή



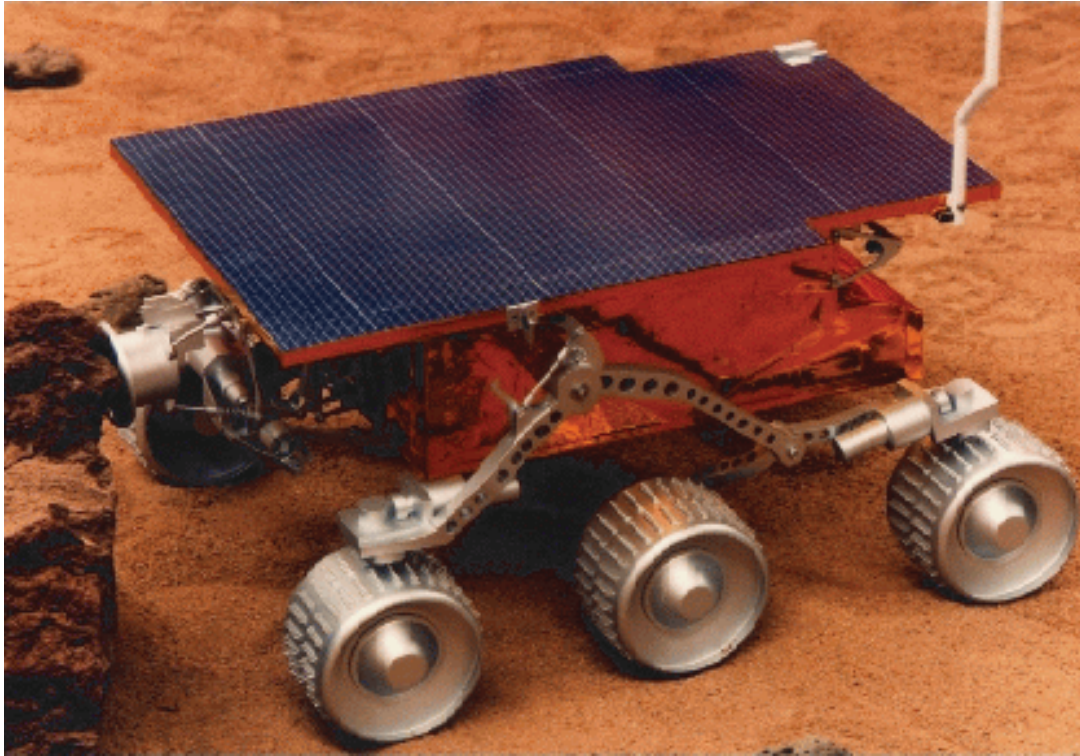
Κατασκευή αυτοκινήτων



Παλετοποίηση κιβωτίων

---

## Ρομποτική Στο Διάστημα



Το περίφημο ρομπότ pathfinder της NASA που στάλθηκε στην πρώτη ανιχνευτική αποστολή στον πλανήτη ΑΡΗ.

## Ρομποτική για την ψυχαγωγία





---

## 1.5 LEGO MINDSTORMS

Τα Mindstorms είναι μια γραμμή παραγωγής της LEGO που δίνει την δυνατότητα στον κάθε ενδιαφερόμενο να προγραμματίσει και να κατασκευάσει πραγματικά ρομπότ που θα έχουν την δυνατότητα να κινούνται και να εκτελούν πλήθος κινήσεων εκμεταλλευόμενοι τους αισθητήρες που παρέχονται.

Τα mindstorms ασπάζονται την εξής βασική αρχή. Μια κεντρική προγραμματιζόμενη μονάδα περιέχει τον επεξεργαστή κι επικοινωνεί ηλεκτρικά με τους αισθητήρες. Η πρώτη γενιά του προγράμματος Mindstorms διατέθηκε περίπου 9 χρόνια πριν, με έναν ελαφρώς πιο περιοριστικό δομικό λίθο. Το RCX όπως λεγόταν, επικοινωνούσε ασύρματα μόνο μέσω υπερύθρων. Η δεύτερη γενιά είναι αυτή που έδωσε περισσότερες δυνατότητες στον κάθε ενδιαφερόμενο με τα νέα χαρακτηριστικά που ενσωμάτωσε. Το mindstorms NXT βασίζεται σε ένα 32bit επεξεργαστή αρχιτεκτονικής ARM7 στα 48MHz και έναν 8bit μικροελεγκτή της Atmel χρονισμένο στα 4MHz. Αντίθετα με το RCX, το NXT επικοινωνεί μέσα από μια θύρα USB 2.0 και έναν προσαρμογέα Bluetooth. Τον προγραμματιστή διευκολύνει μια LCD οθόνη 100x64 χαρακτήρων όπου απεικονίζεται το μενού διαχείρισης του NXT από μονόχρωμα γραφικά έως μαθηματικές παραστάσεις. Η κεντρική μονάδα του NXT ανταλλάσσει δεδομένα με τους αισθητήρες και τους κινητήρες. Από τις 7 διαθέσιμες θύρες, οι 3 αποστέλλουν δεδομένα στα μοτέρ κίνησης και οι υπόλοιπες τέσσερις δέχονται πληροφορίες από αισθητήρες. Η μνήμη του NXT Mindstorms αποτελείται από 256 KB flash RAM προς εγγραφή του firmware και του κώδικα καθώς και 64 KB RAM για την εκτέλεση ενός προγράμματος.



*Το προγραμματιζόμενο «τούβλο» RCX της 1ης γενιάς (1998)  
και το πιο σύγχρονο NXT της 2ης γενιάς (2006)*

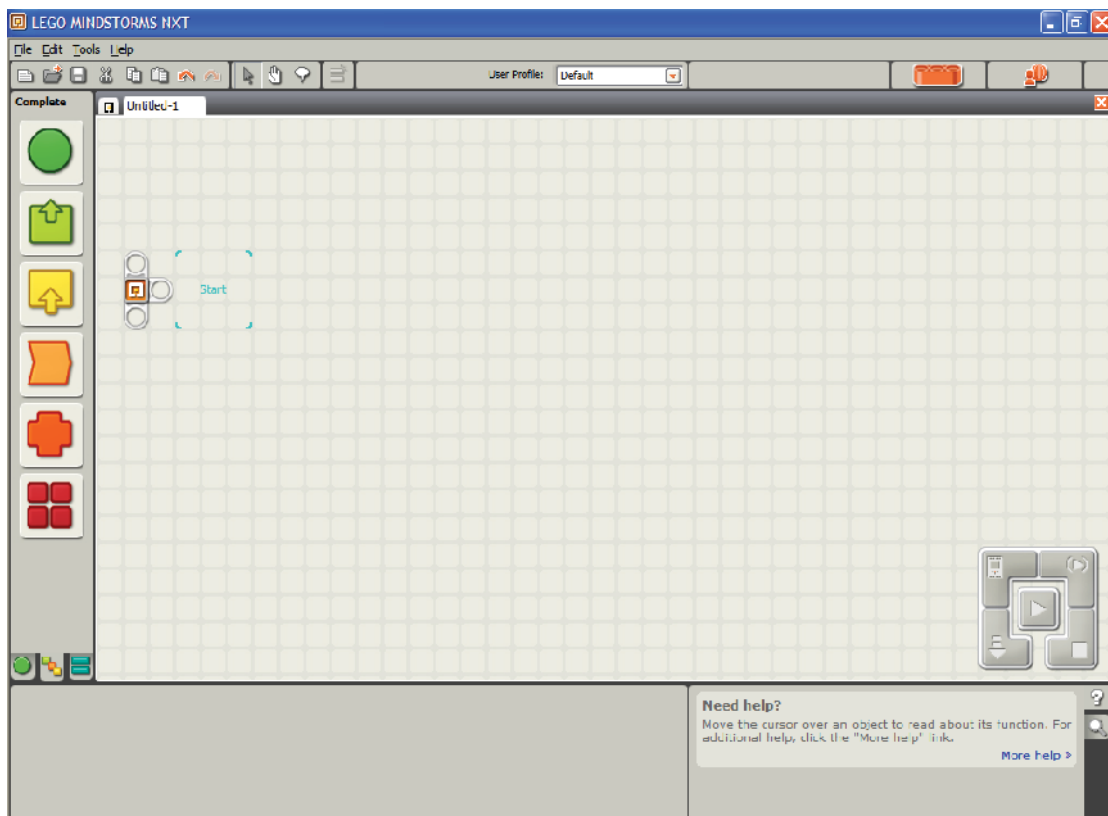
---

Το σύστημα LEGO Mindstorms NXT προσφέρει ένα ευφυές «τούβλο» που προγραμματίζεται μέσω υπολογιστή καθώς επίσης και ένα πλήθος από αντικείμενα που συνεργάζονται μαζί του όπως μοτέρ κίνησης, αισθητήρες ήχου, επαφής ,φωτός .Όλα αυτά υλοποιούνται γύρω από μια κατασκευή που πραγματοποιείται με τη χρήση της μεγάλης συλλογής μικρών αντικειμένων που αποτελούν και τα δομικά υλικά όπως γρανάζια, τροχαλίες και μικρά «τουβλάκια»



**Το σύστημα NXT**

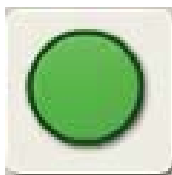
Για την υλοποίηση ενός project η LEGO σε συνεργασία με τη National Instruments δίνει την ευκαιρία στους χρήστες του NXT να σχεδιάσουν ένα πρόγραμμα, χωρίς γνώσεις προγραμματισμού, βασιζόμενοι στο σύστημα των εικονικών οργάνων που έχει αναπτύξει η National Instruments στην σουίτα Labview. Ο χρήστης προγραμματίζει το ρομπότ του με τη μορφή μιας ροής δεδομένων εισάγοντας δομικές μονάδες (blocks) οι οποίες αντιστοιχούν σε λειτουργίες κάποιου κινητήρα ή κάποιου αισθητήρα. Η ροή των δεδομένων ακολουθεί την ιδέα μιας μηχανής πεπερασμένων καταστάσεων (Finite State Machine) και αυτός ο τρόπος οργάνωσης βοηθά στον προγραμματισμό πιο σύνθετων διεργασιών.



Το περιβάλλον σχεδίασης του NXT-G. Αριστερά υπάρχει το σημείο εκκίνησης απ' όπου ξεκινά μια διεργασία ή και περισσότερες.

Αριστερά στο μενού του προγράμματος, εμφανίζεται ολόκληρη η βιβλιοθήκη που αποτελείται από 6 κατηγορίες με blocks. Τα blocks δράσης αναφέρονται στην κίνηση ενός μοτέρ, την αναπαραγωγή ενός ήχου, την αποστολή ενός μηνύματος μέσω Bluetooth ή την εμφάνιση δεδομένων επάνω στην οθόνη του NXT.

Θα παραθέσουμε μία επεξήγηση των βασικών εντολών-blocks του προγράμματος NXT-G χωρίς να προχωρήσουμε σε αναλυτικές λεπτομέρειες καθώς σκοπός δεν είναι η εκμάθηση του λογισμικού αλλά η κατανόηση των βασικών λειτουργιών που επιτελούν οι εντολές αυτές.



Το πρώτο κουμπί (common) αποτελείται από τα blocks-εντολές κίνησης(move), εγγραφής(record), ήχου(sound), επίδειξης(display), αναμονής(wait), επανάληψης(loop) και επιλογής(choose)

---

Οι εντολές-blocks είναι:



Block-εντολή κίνησης(move)



Block-εντολή επίδειξης(display)



Block-εντολή εγγραφής(record)



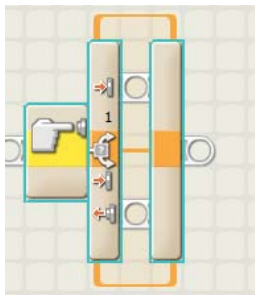
Block-εντολή αναμονής(wait)



Block-εντολή ήχου(sound)



Block-εντολή Επανάληψης(loop)



Block-εντολή επιλογής(switch)



Το δεύτερο κουμπί (action) αποτελείται από τα blocks-εντολές Κίνησης(motor), ήχου(sound), επίδειξης(display), αποστολής μέσω Bluetooth(send message)

---

Οι εντολές-blocks είναι:



Block-εντολή κίνησης(move)



Block-εντολή ήχου(sound)



Block-εντολή επίδειξης(display)



Block-εντολή αποστολής μέσω Bluetooth (send message)



Το τρίτο κουμπί (sensor) αποτελείται από τα blocks-εντολές για τους αισθητήρες.

Τα block-εντολές είναι:



Αισθητήρας επαφής(touch sensor)



Αισθητήρας ήχου (sound sensor)



Αισθητήρας φωτός (light sensor)



Υπερηχητικός αισθητήρας (Ultrasonic sensor)



NXT button sensor



Αισθητήρας περιστροφής (rotation sensor)



Αισθητήρας χρόνου  
(Timer sensor)



Receive message  
sensor



Το τέταρτο κουμπί (flow) αποτελείται κυρίως από τα blocks-εντολές για την παύση κάποιας λειτουργίας του ρομποτικού συστήματος καθώς επίσης και για την πλήρη διακοπή του.

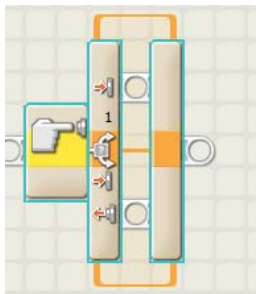
Τα block-εντολές είναι:



Block-εντολή  
αναμονής(wait)



Block-εντολή  
Επανάληψης(loop)



Block-εντολή  
επιλογής(switch)



Block-εντολή  
διακοπής λειτουργίας



Το πέμπτο κουμπί (data) αποτελείται από τα blocks-εντολές όπου μπορούμε να εισάγουμε μεταβλητές όπως λογικές, μαθηματικές, μεταβλητές σύγκρισης και εύρους και τυχαίες μεταβλητές.

Τα block-εντολές είναι:



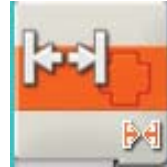
Block-εντολή  
λογικής μεταβλητής



Block-εντολή  
μαθηματικής μεταβλητής



Block-εντολή  
Μεταβλητής σύγκρισης



Block-εντολή  
Μεταβλητής εύρους



Block-εντολή  
τυχαίας μεταβλητής



Block-εντολή  
δημιουργίας μεταβλητής



Τέλος, το έκτο κουμπί (advanced) αποτελείται από τα blocks-εντολές όπου μπορούμε να εισάγουμε εντολές όπως δημιουργίας κειμένου και μετατροπής.

Τα block-εντολές είναι:



Block-εντολή  
δημιουργίας  
κειμένου(text)



Block-εντολή  
μετατροπής αριθμού  
σε κείμενο



Block-εντολή για μη  
είσοδο σε sleep mode



Block-εντολή καταγραφής  
δεδομένων στην μνήμη του  
NXT



Block-εντολή ισοστάθμισης  
για sound και light sensor



Block-εντολή μηδενισμού  
ενός ή περισσότερων  
motors



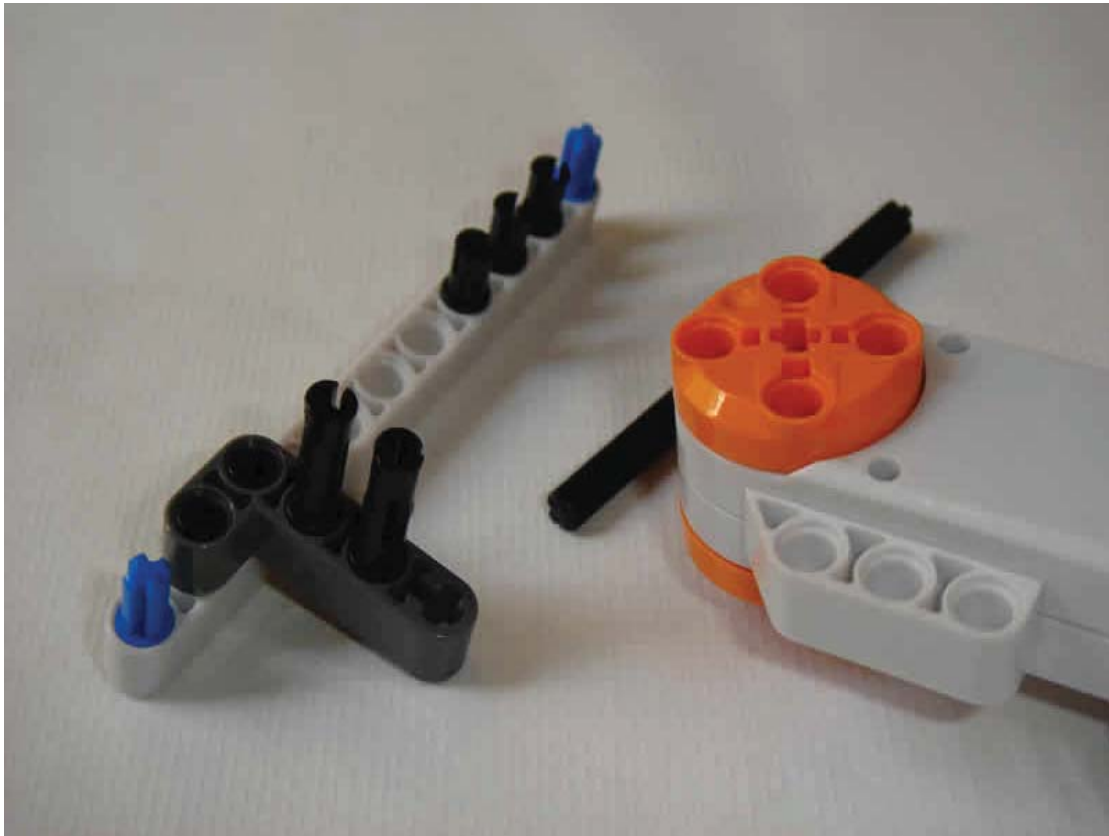
---

## ΚΕΦΑΛΑΙΟ 2

ΠΑΡΟΥΣΙΑΣΗ ΚΑΤΑΣΚΕΥΗΣ ΑΥΤΟΜΑΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ  
ΓΙΑ ΤΗΝ ΕΥΡΕΣΗ ΚΑΤΑΛΛΗΛΟΥ ΧΩΡΟΥ ΚΑΙ ΠΡΑΓΜΑΤΟΠΟΙΗΣΗ ΤΗΣ  
ΕΥΘΕΙΑΣ ΚΑΙ ΑΝΤΙΣΤΡΟΦΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΣΤΑΘΜΕΥΣΗΣ.

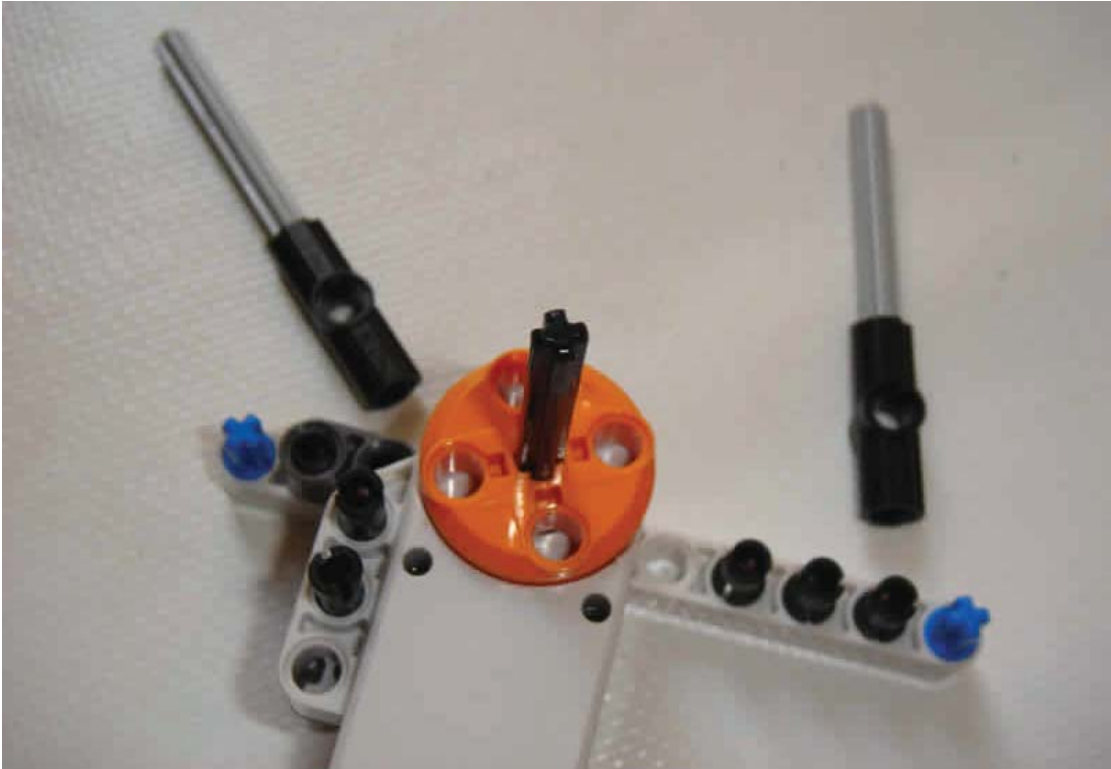
Στο κεφάλαιο αυτό θα παρουσιάσουμε βήμα προς βήμα την κατασκευή του εν  
λόγω ρομποτικού συστήματος.

ΒΗΜΑ1.

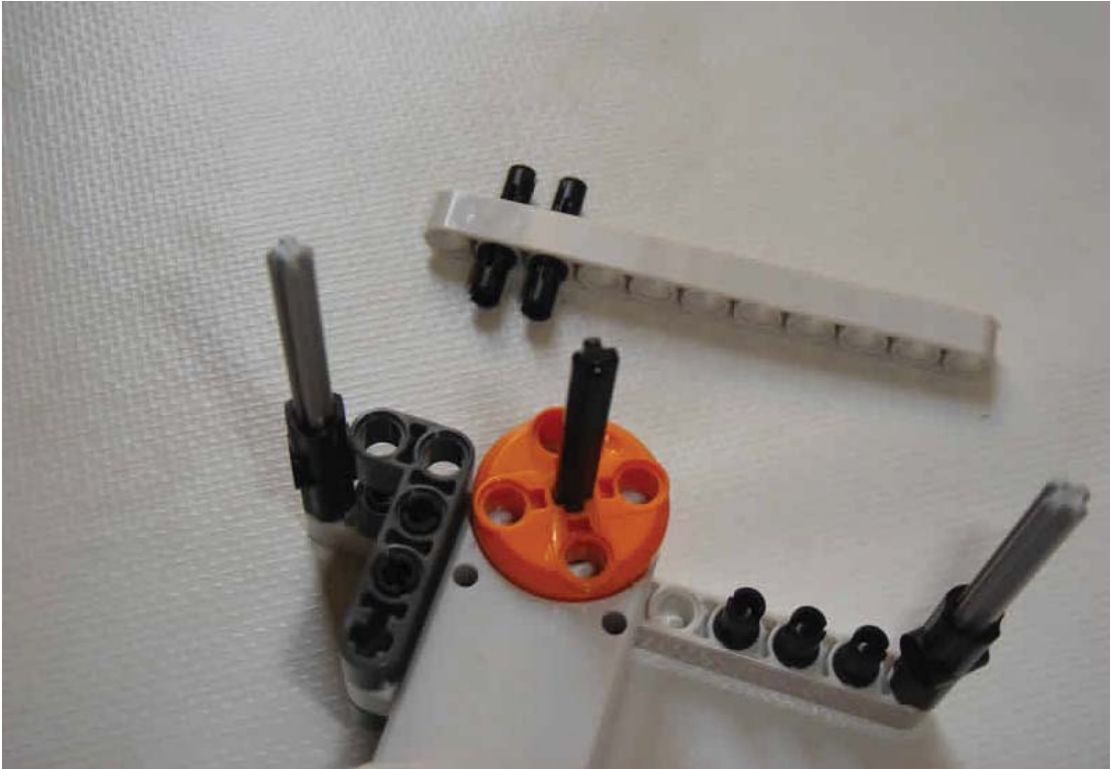


---

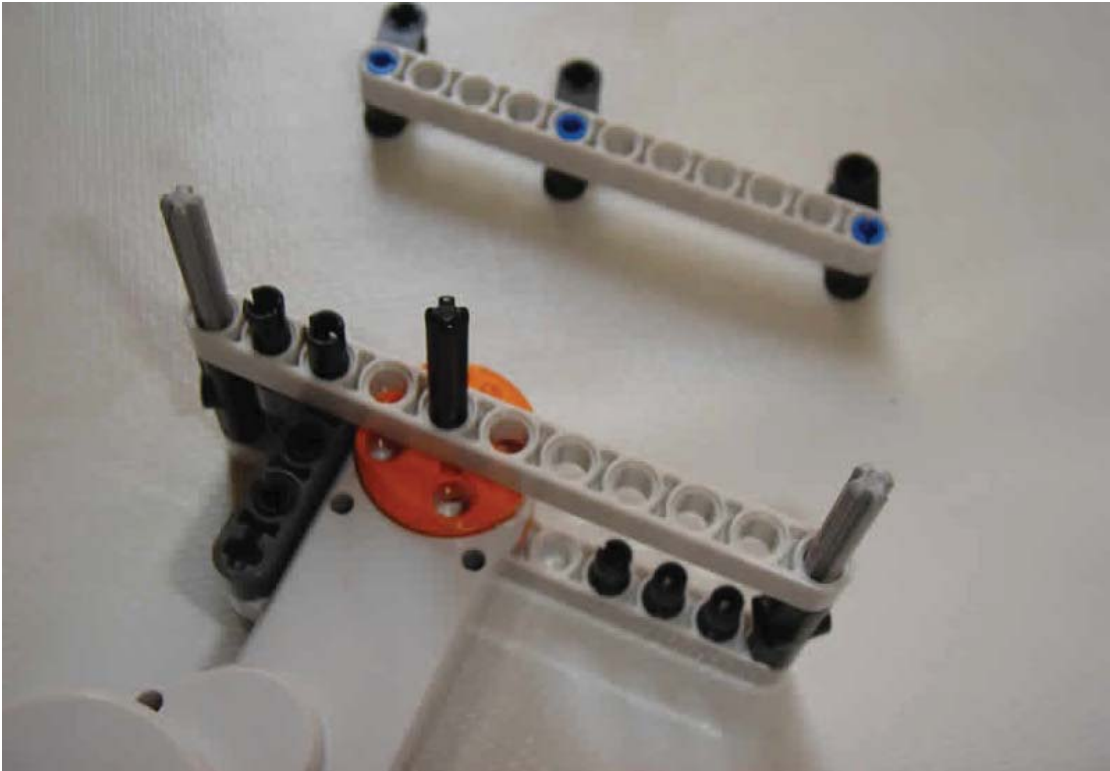
BHMA2.



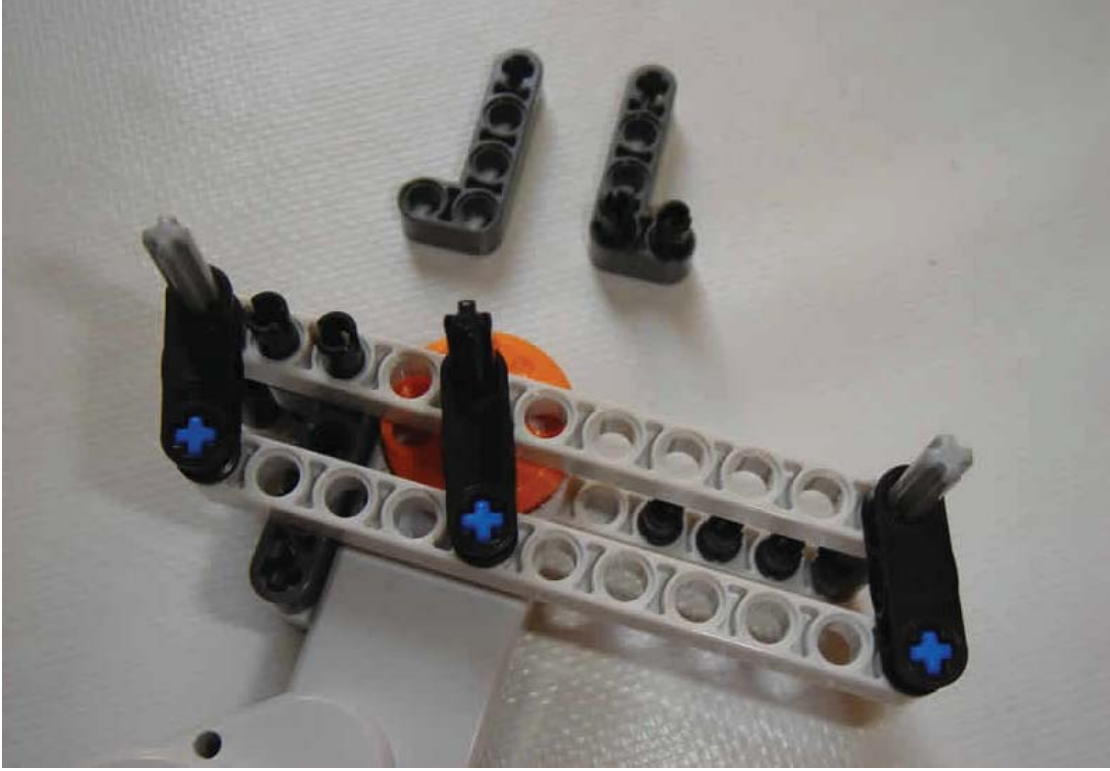
BHMA3.



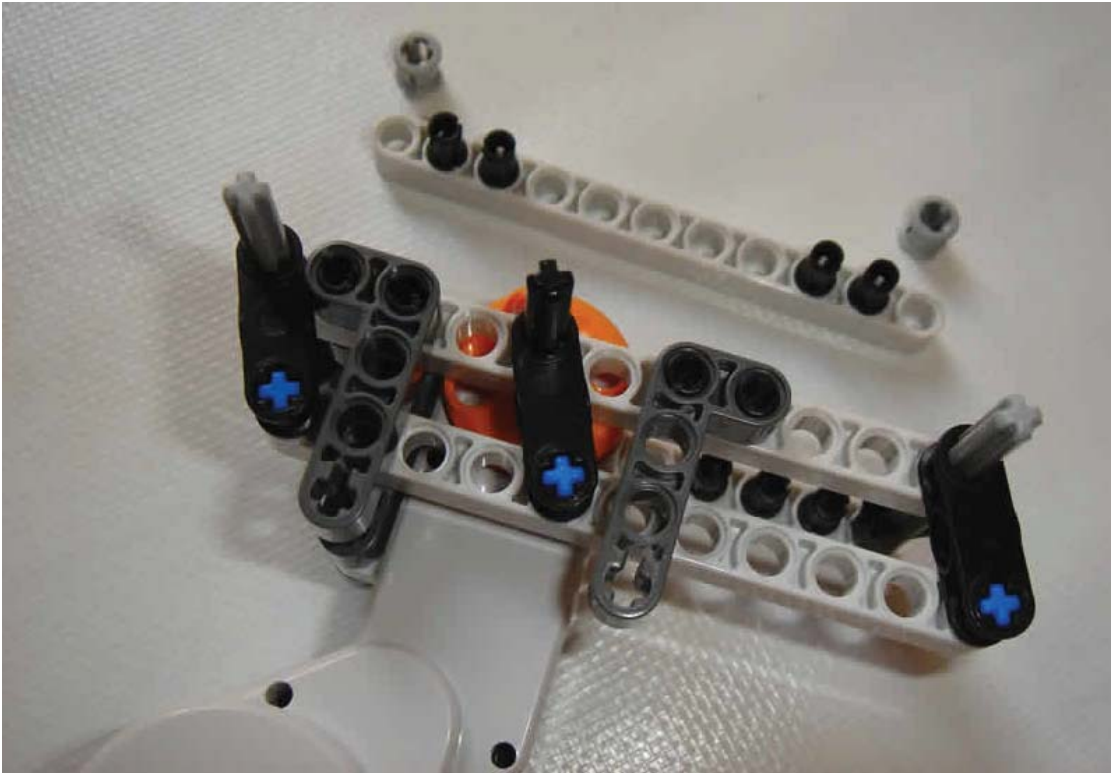
BHMA4.



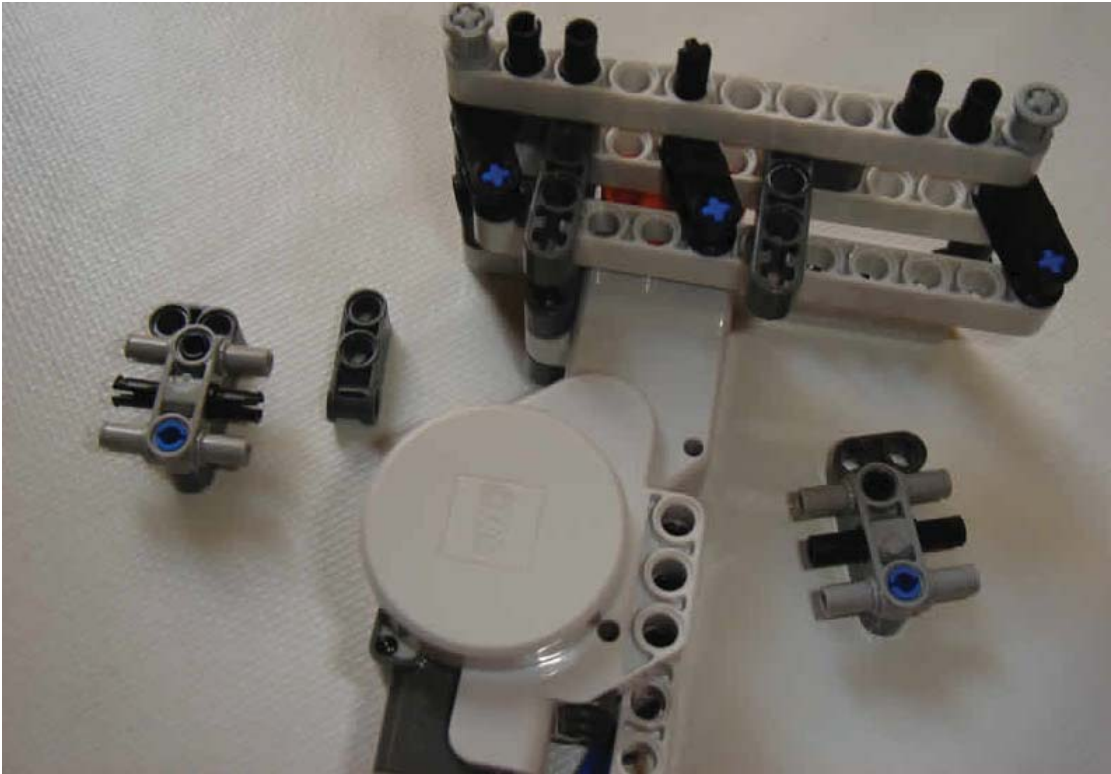
BHMA5.



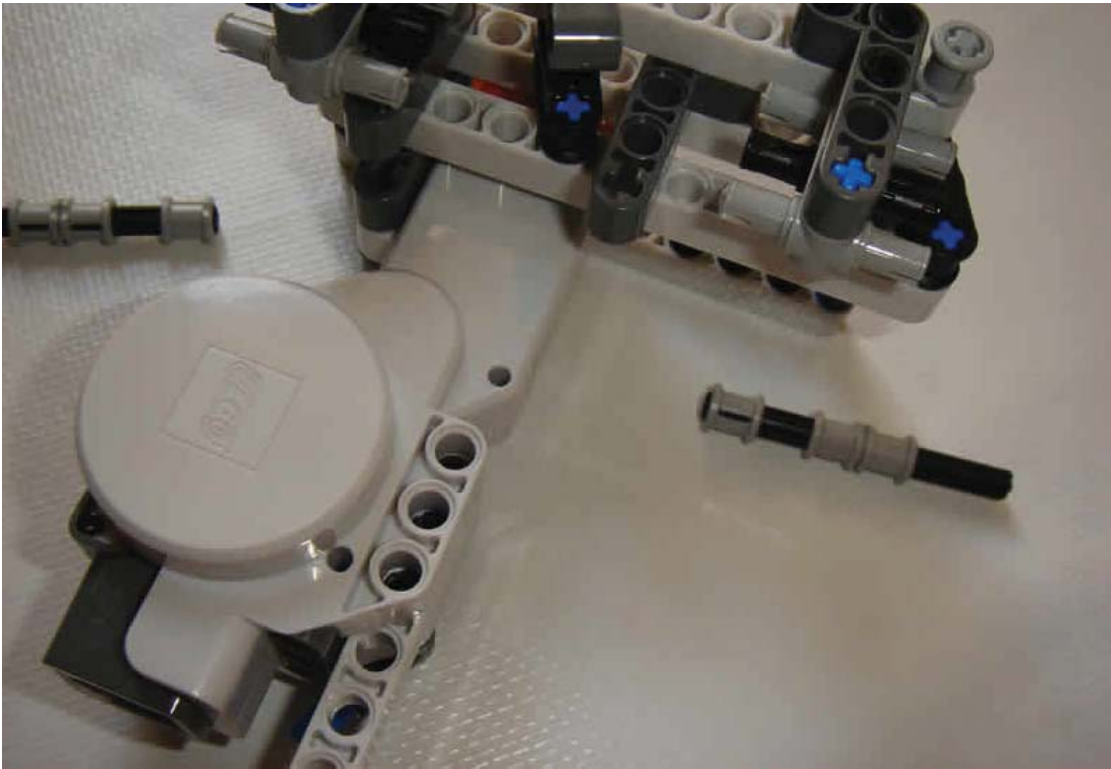
BHMA6.



BHMA7.



BHMA8.



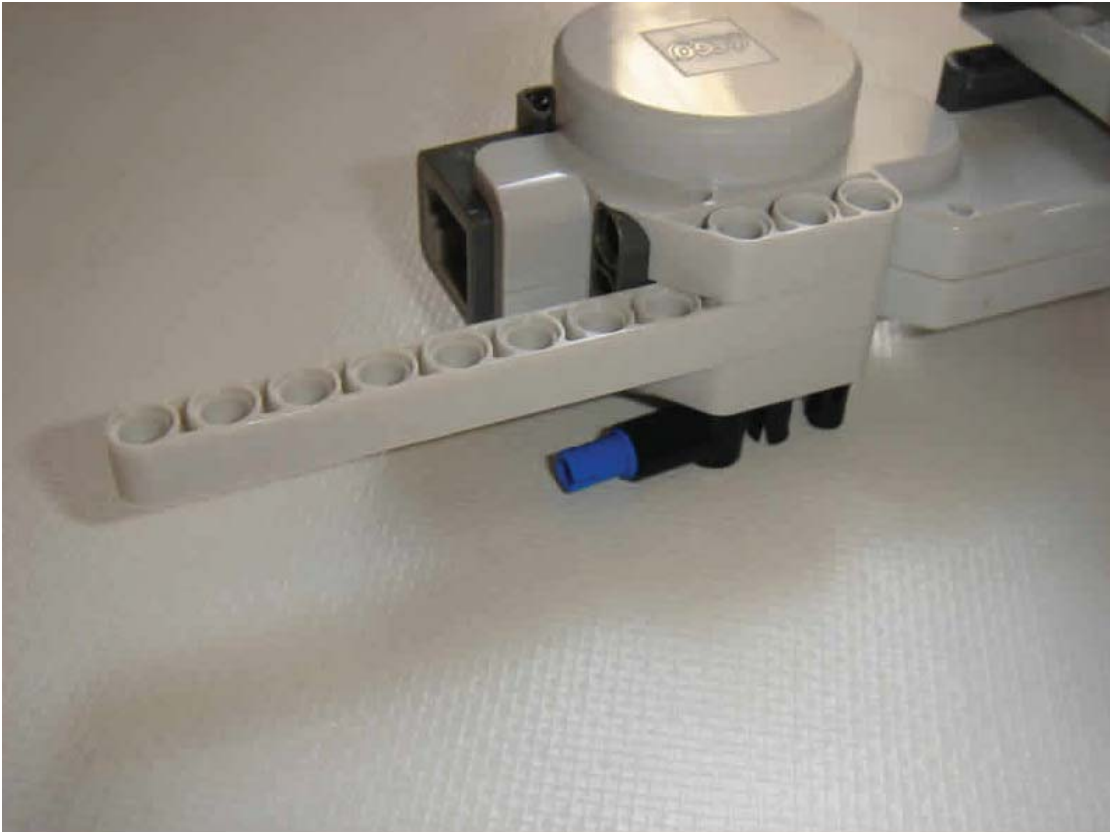
BHMA9.



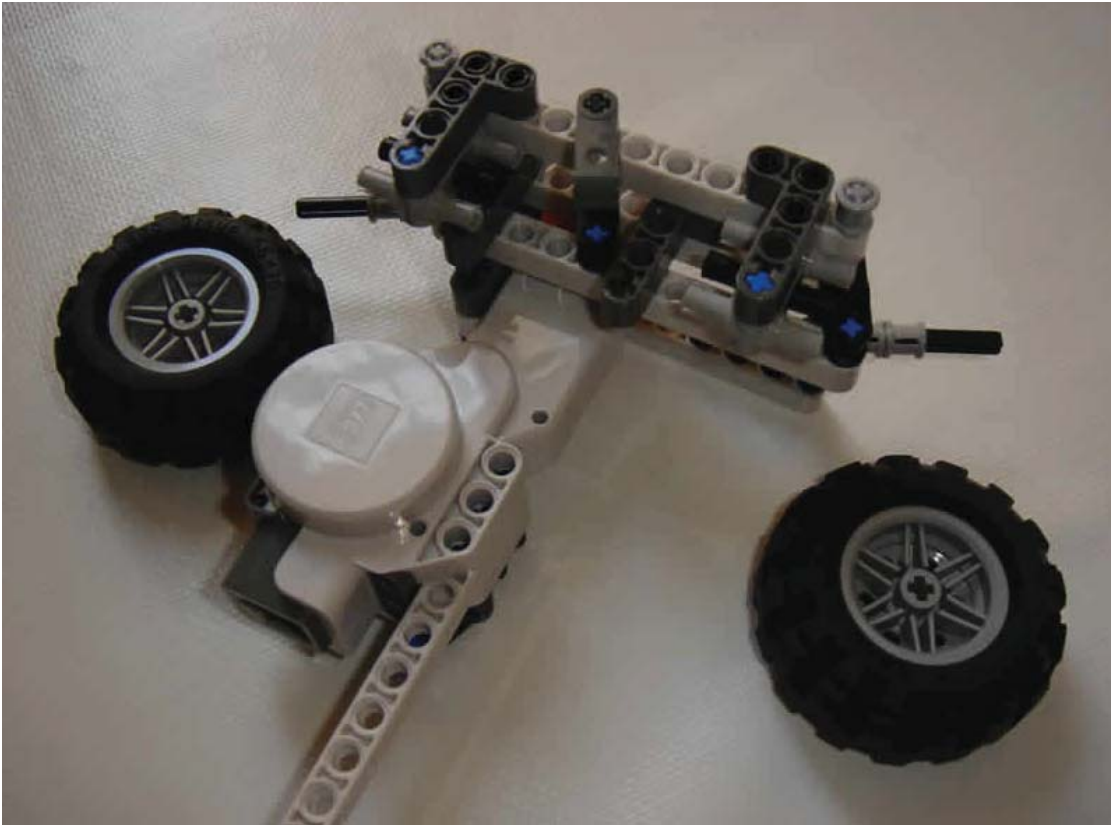
BHMA10.



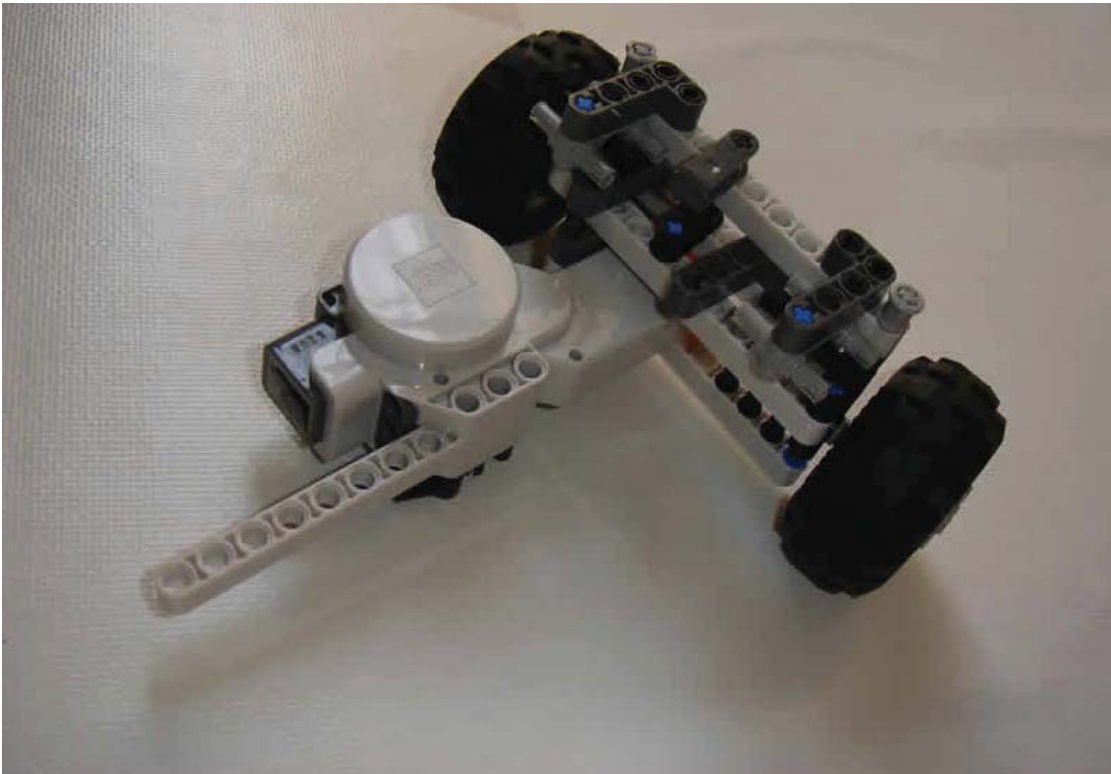
BHMA11.



BHMA12.



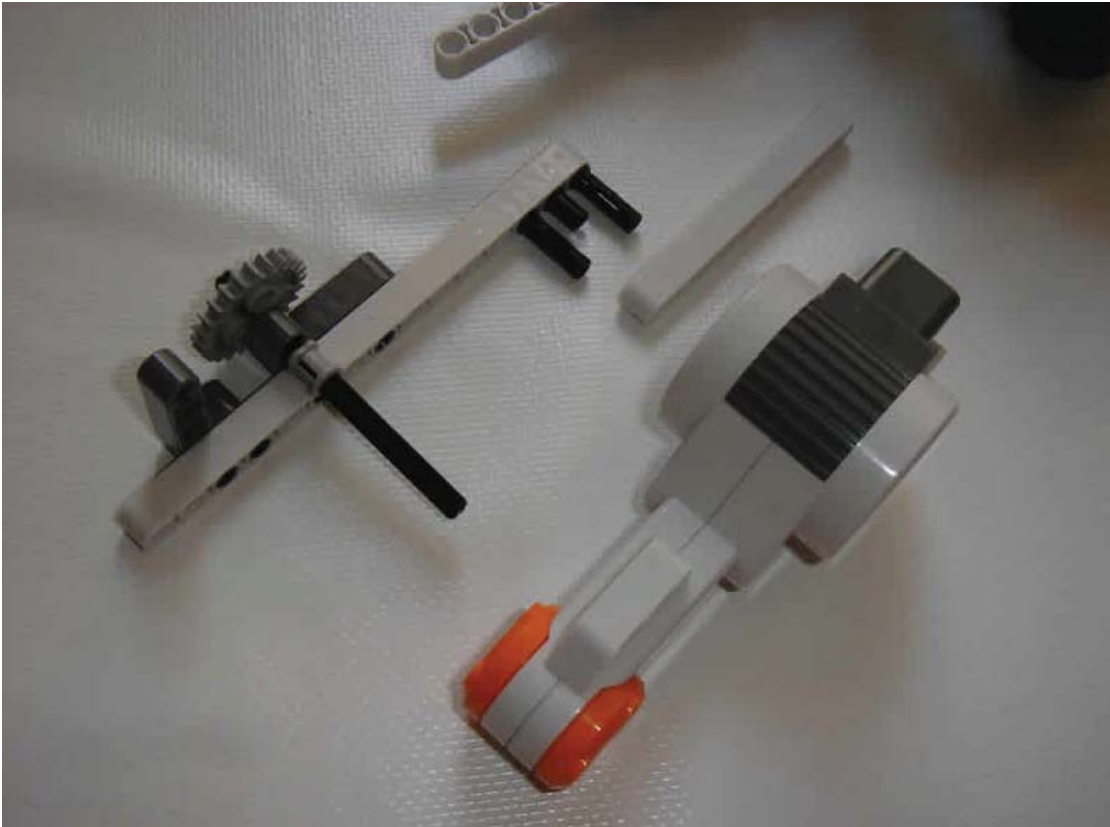
BHMA13.



BHMA14.



BHMA15.

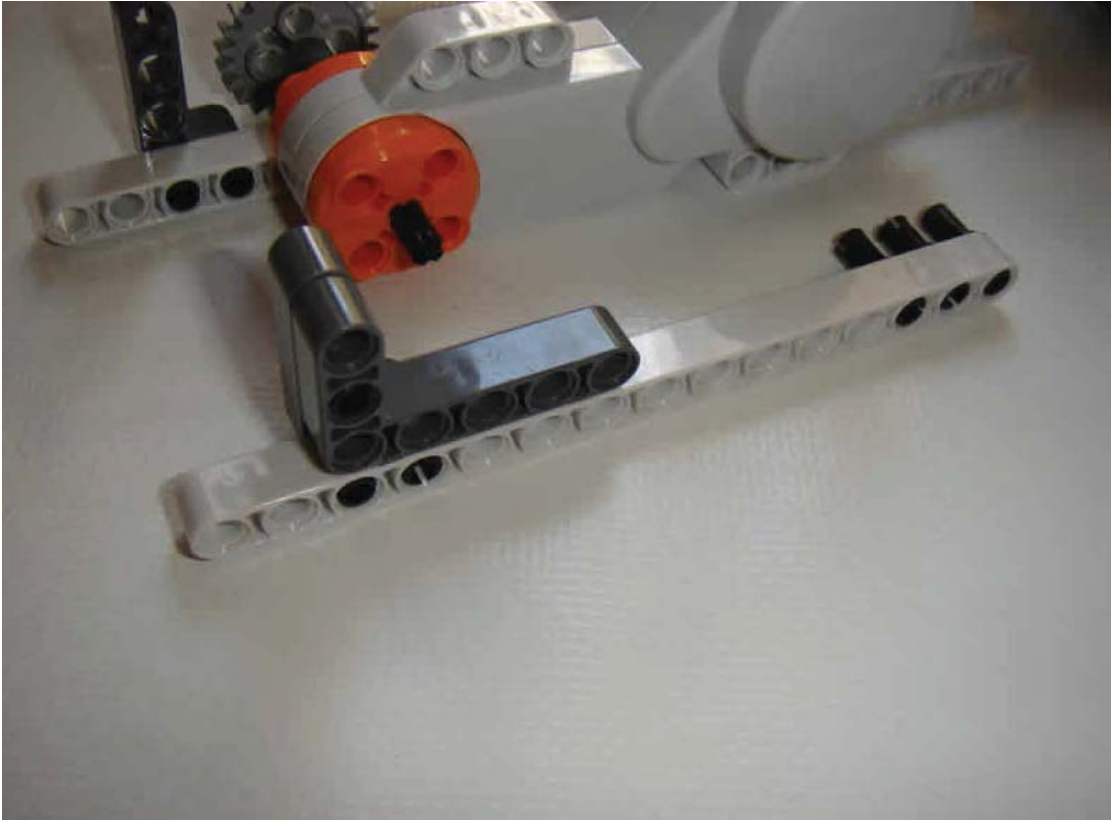




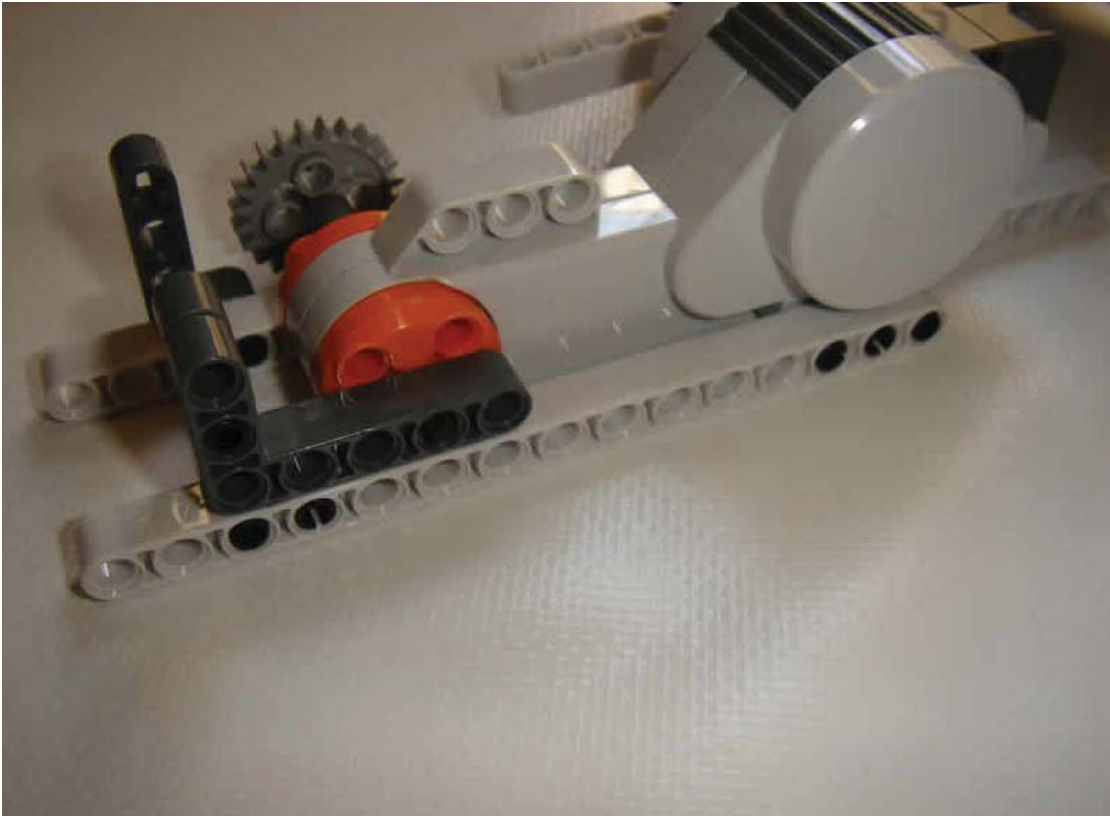
BHMA16.



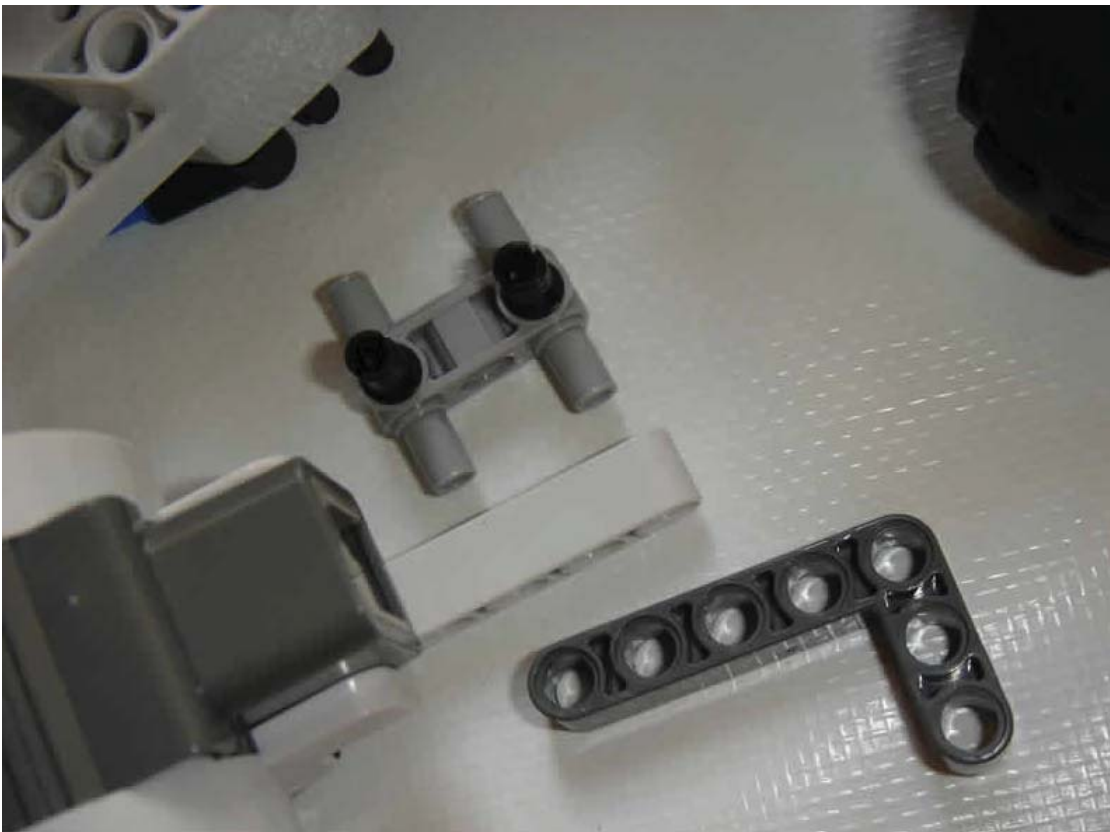
BHMA17.



BHMA18.



BHMA19.

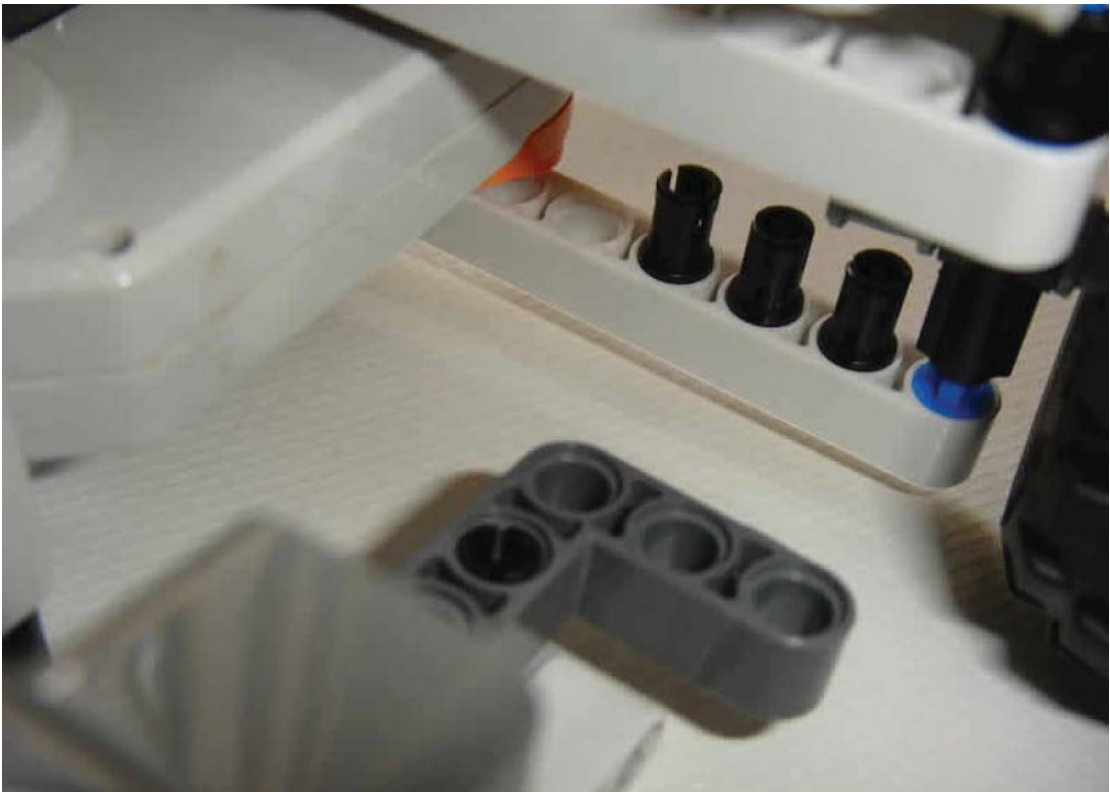


---

BHMA20.

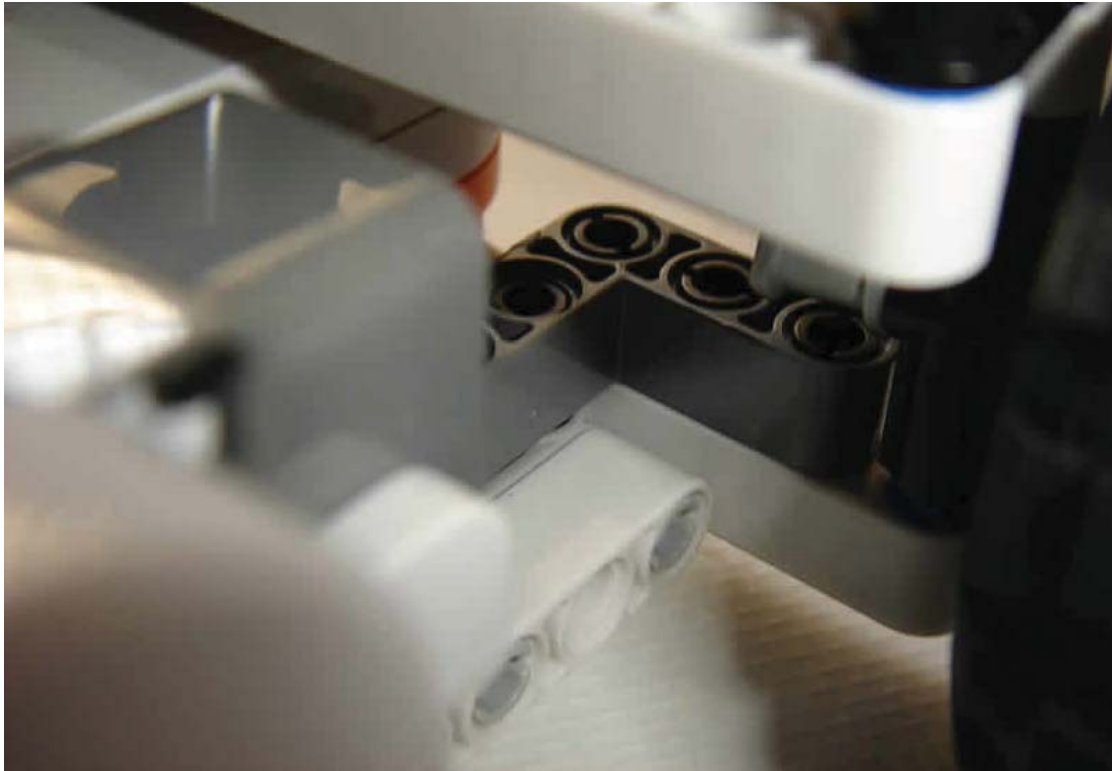


BHMA21.

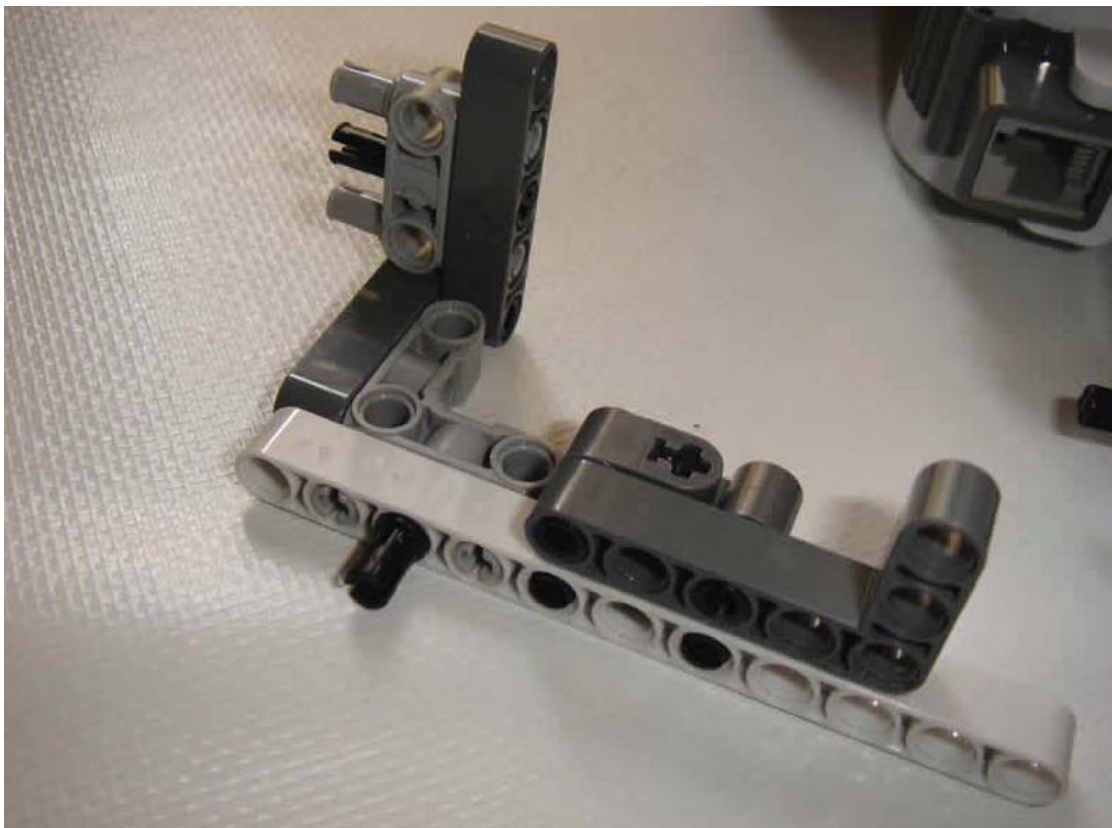


---

BHMA22.

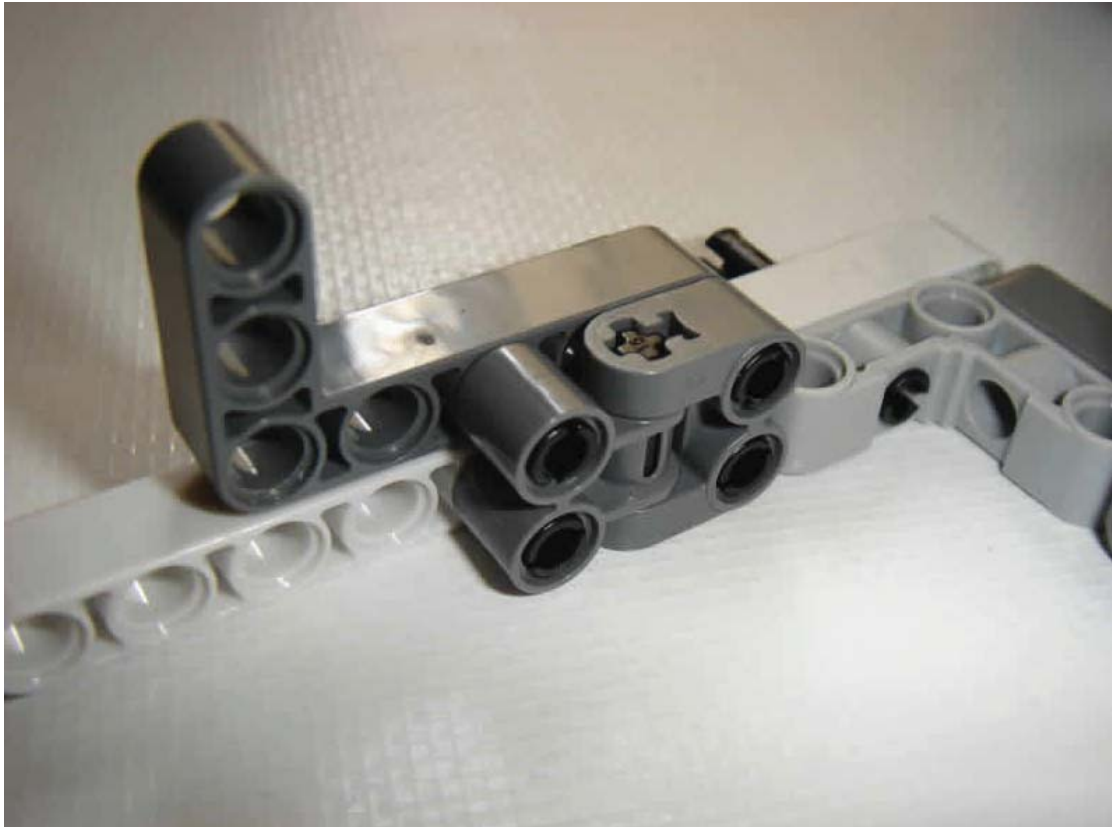


BHMA23.

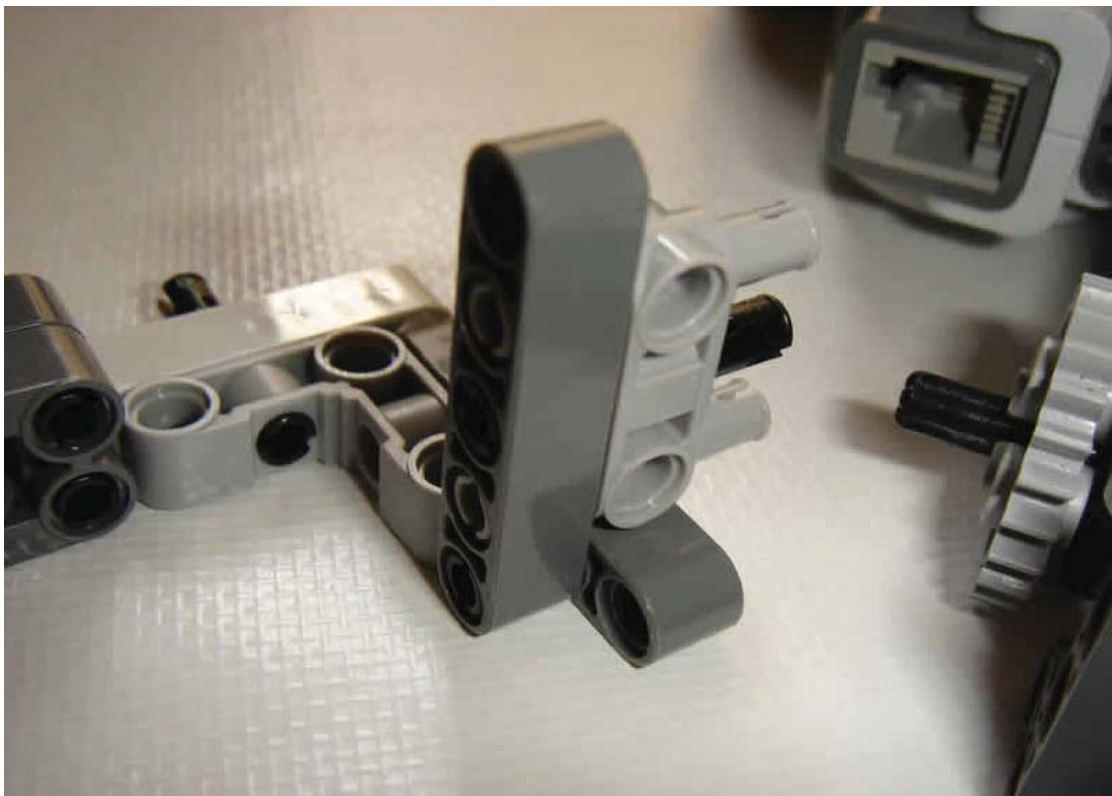


---

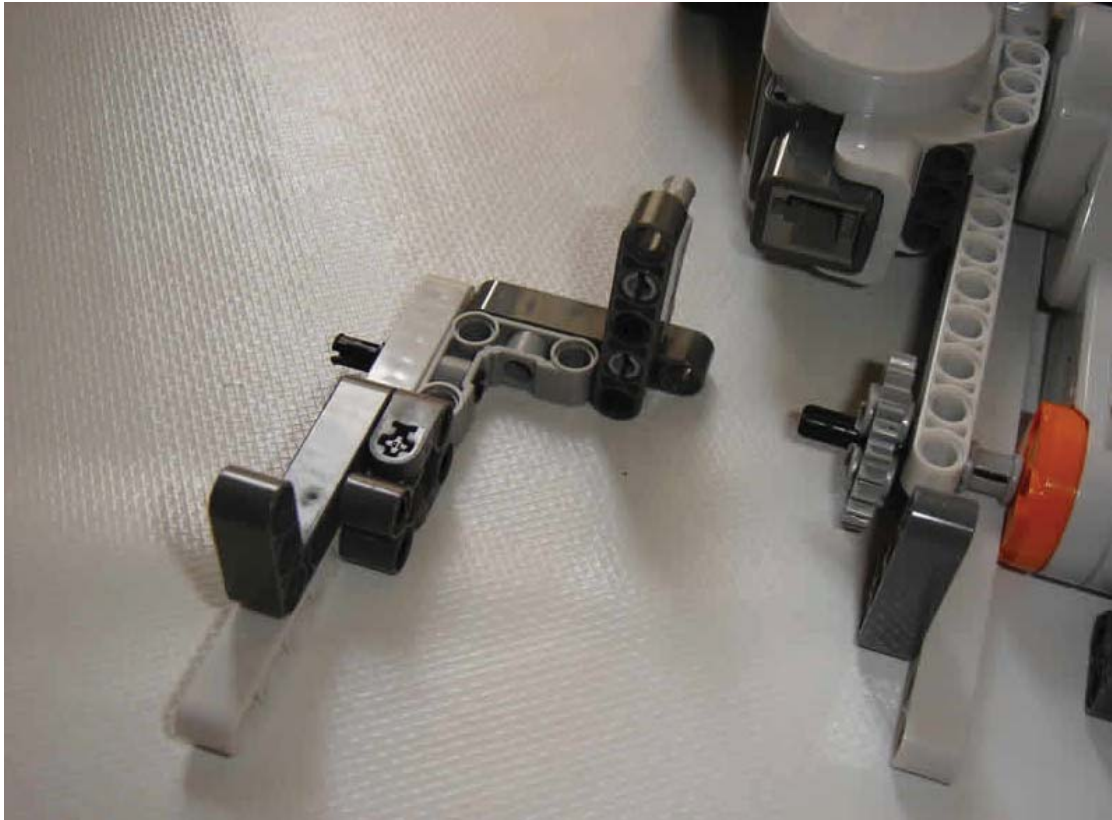
BHMA24.



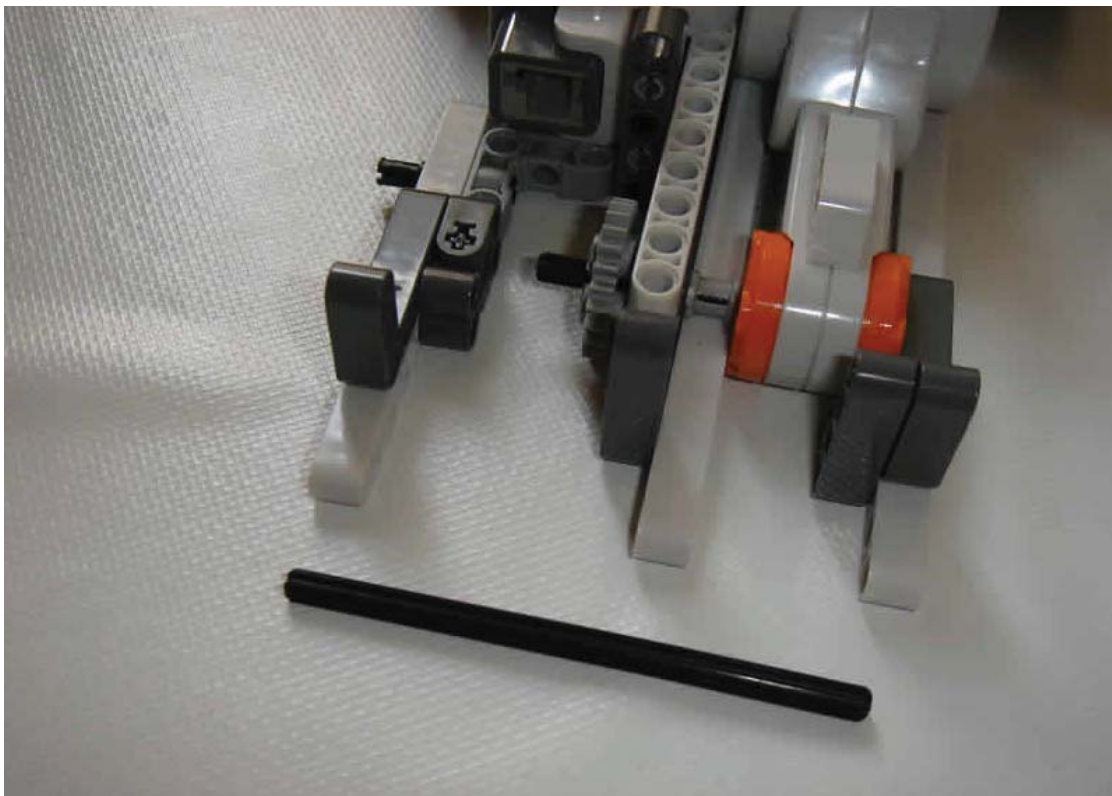
BHMA25.



BHMA26.

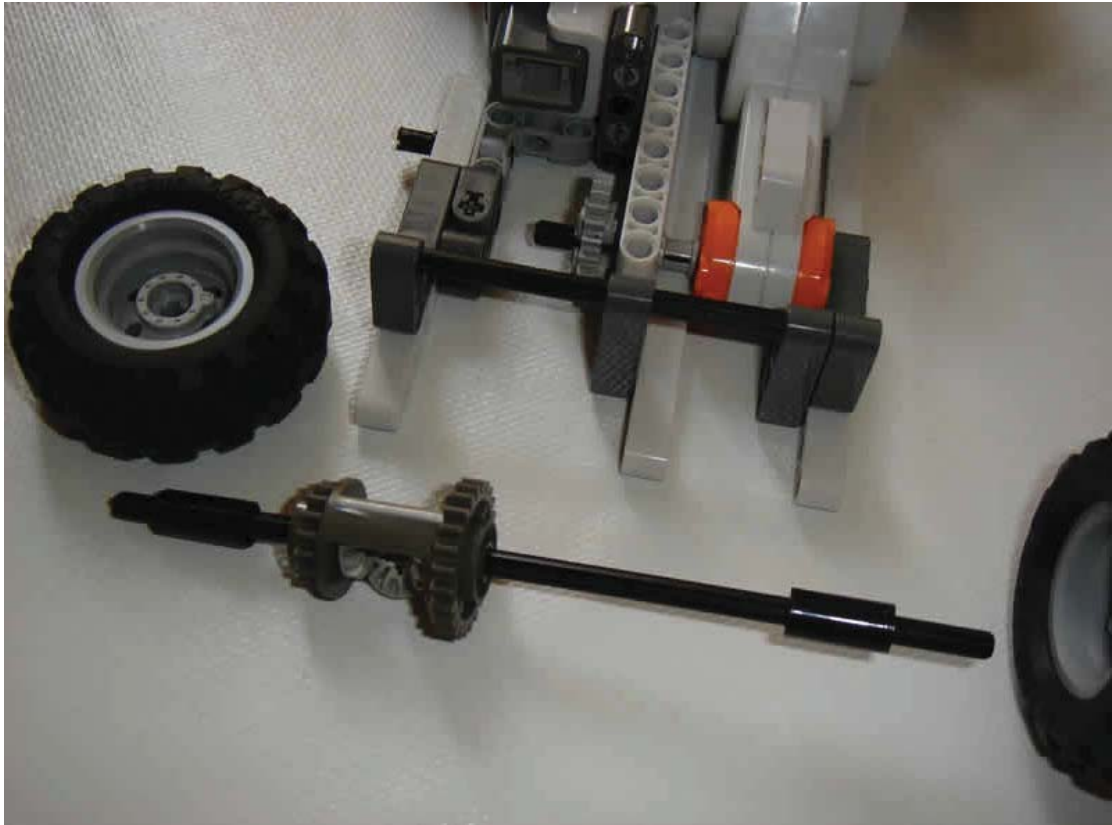


BHMA27.

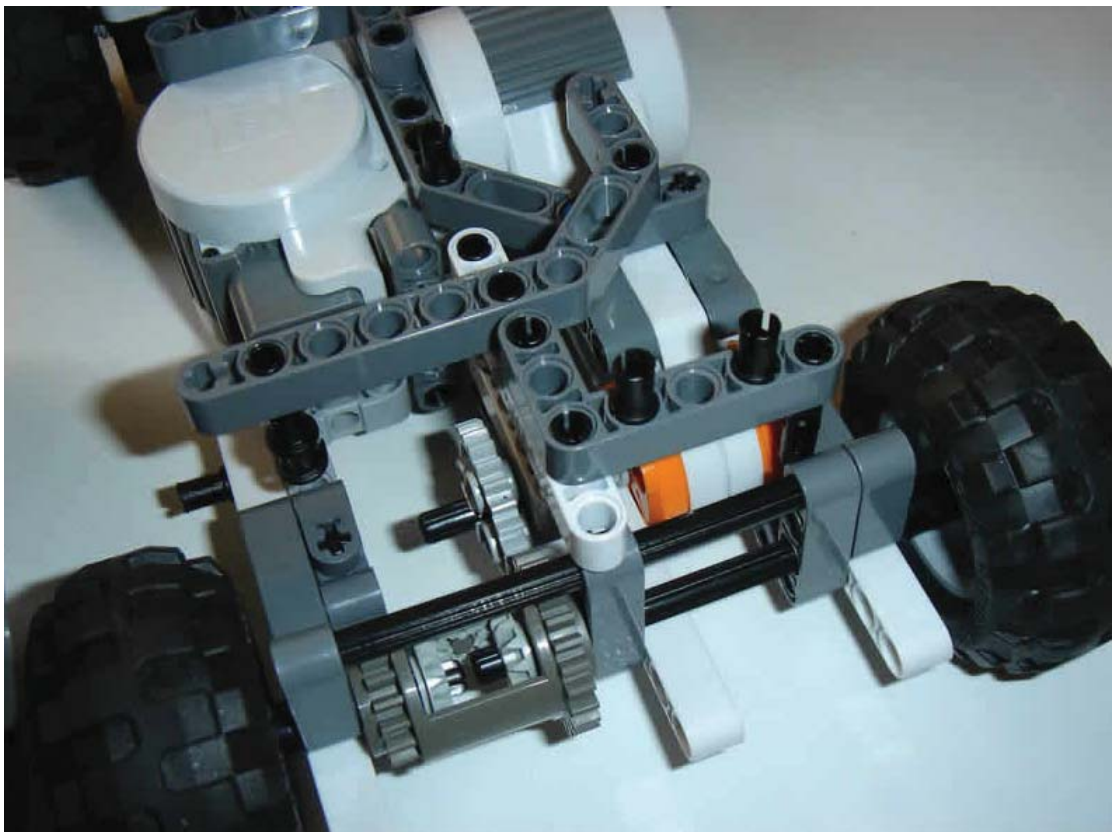


---

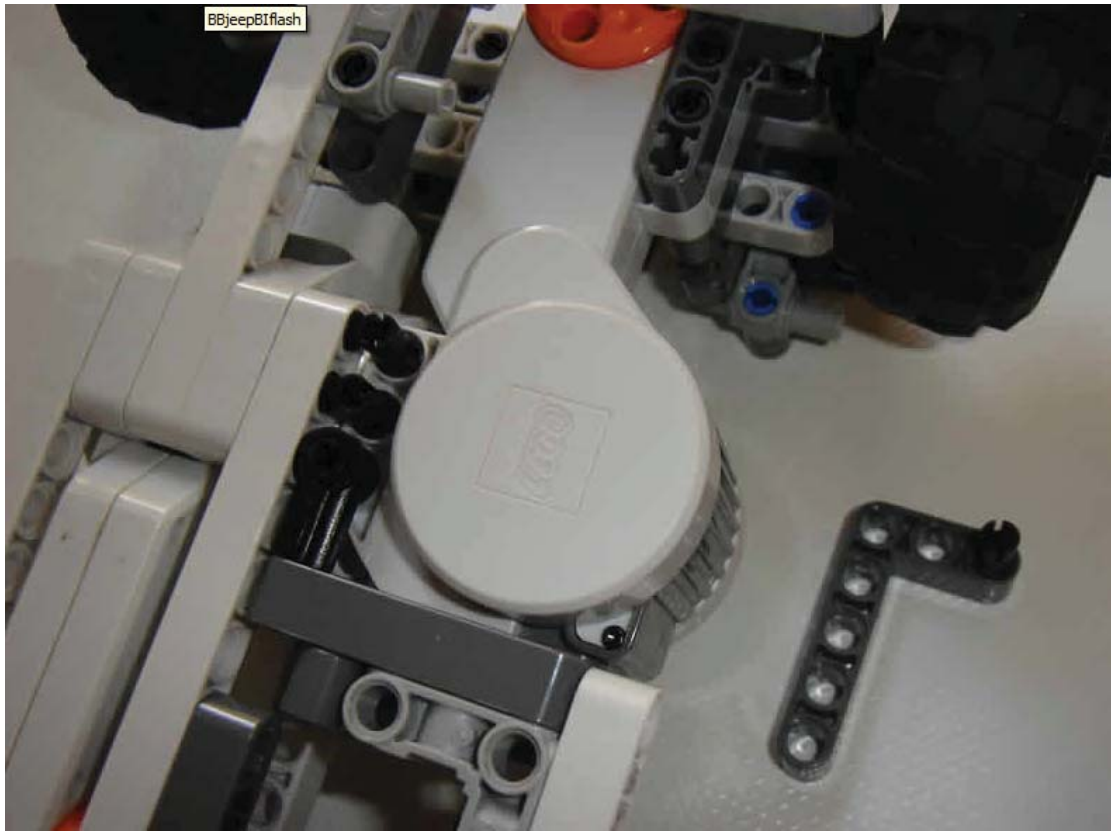
BHMA28.



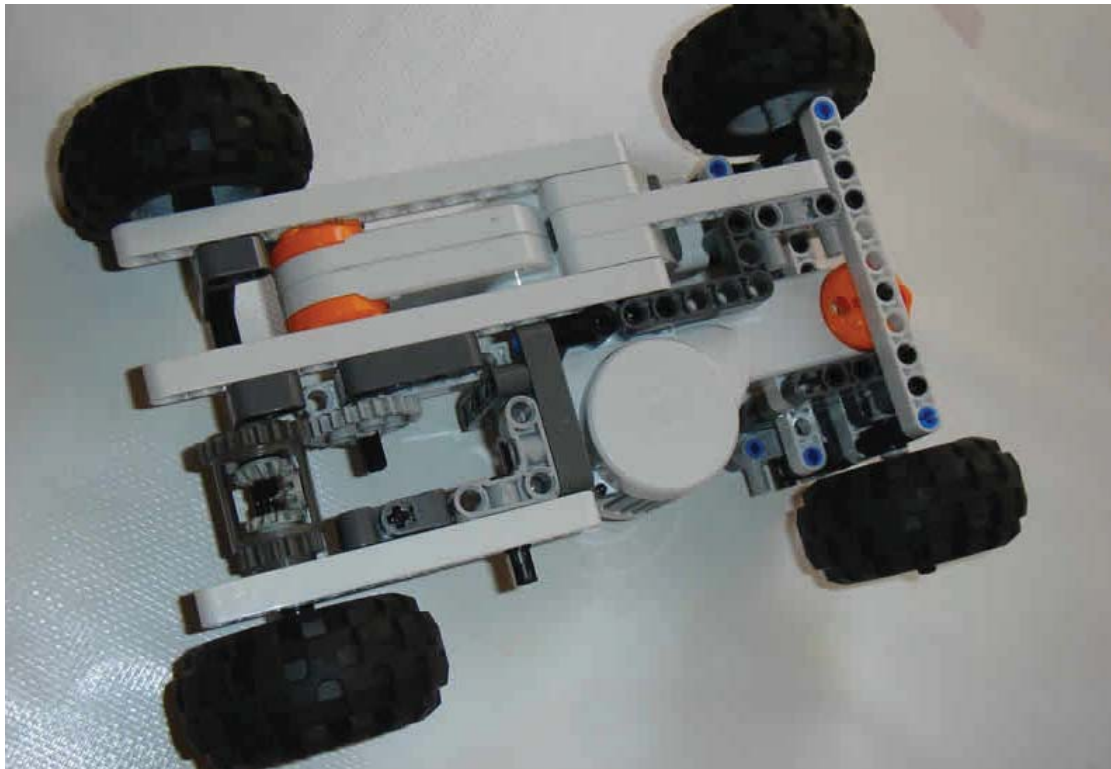
BHMA29.



BHMA30.



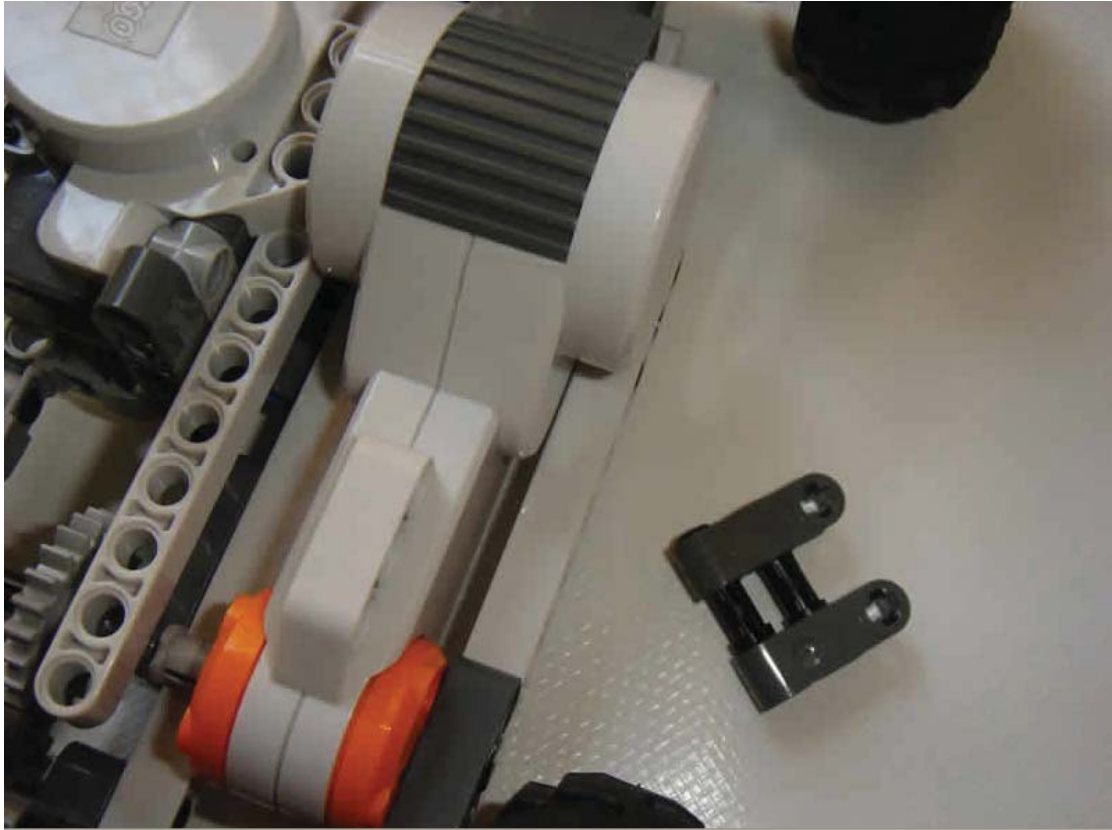
BHMA31.



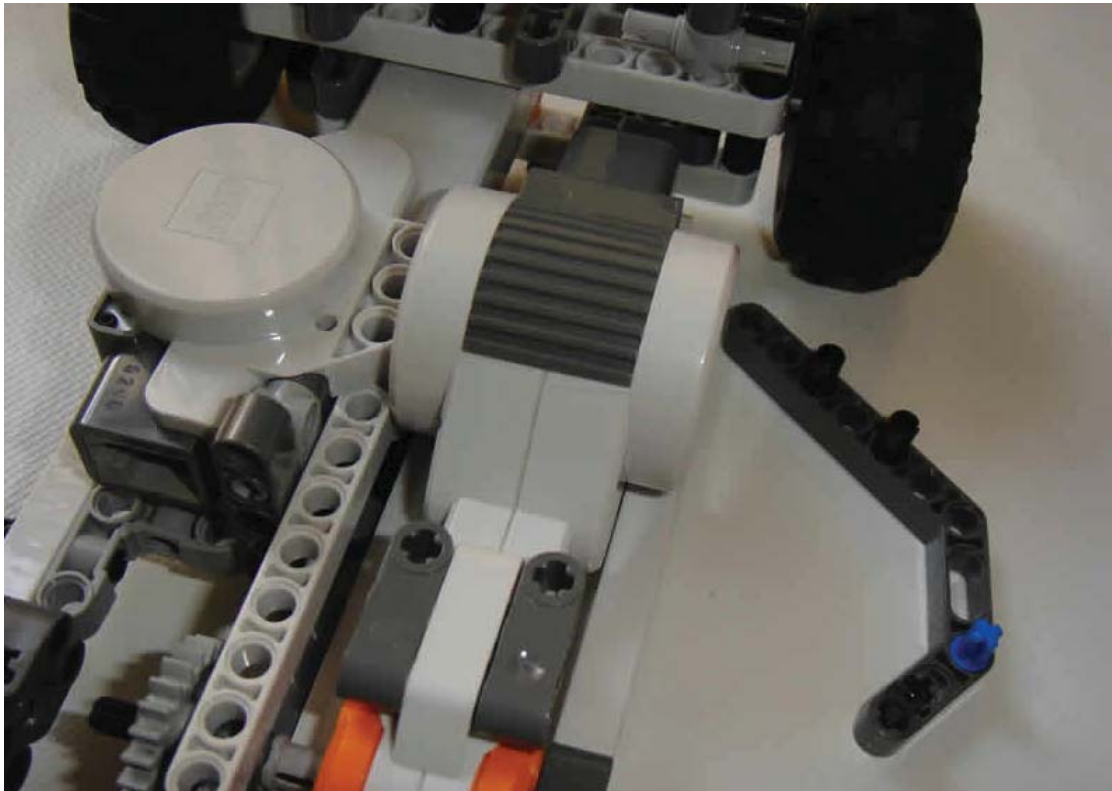


---

BHMA32.

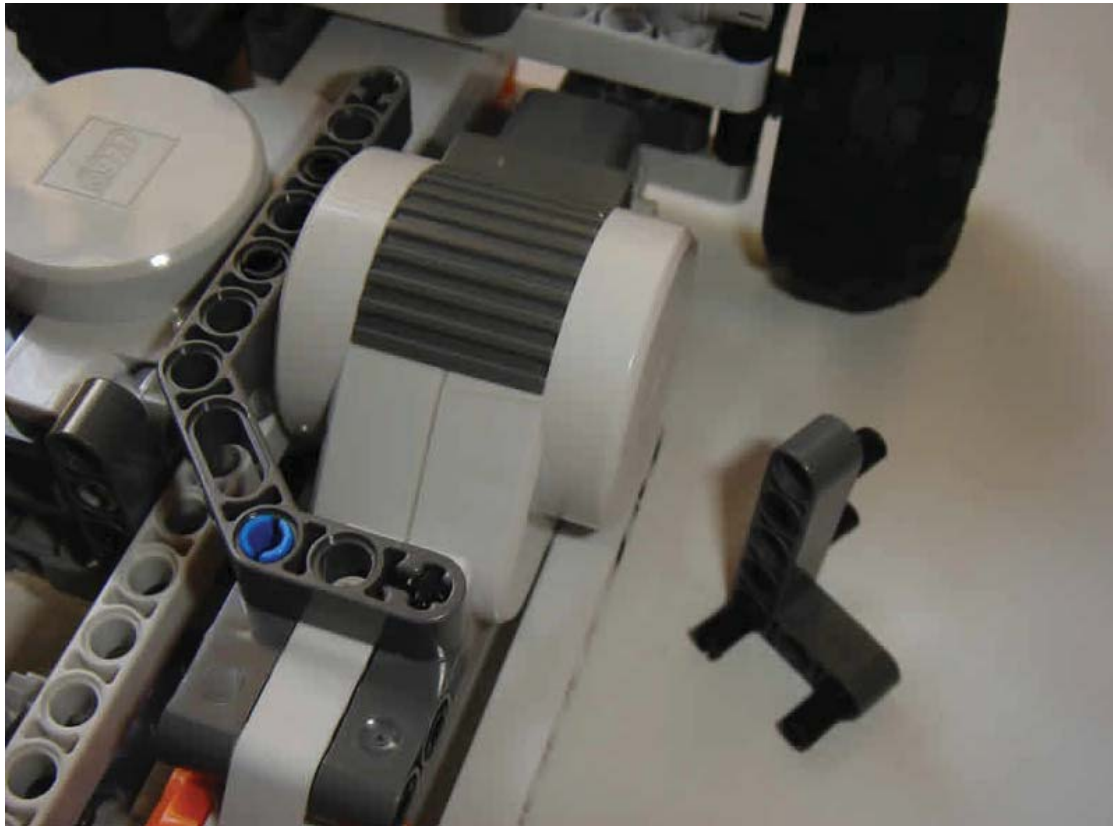


BHMA33.

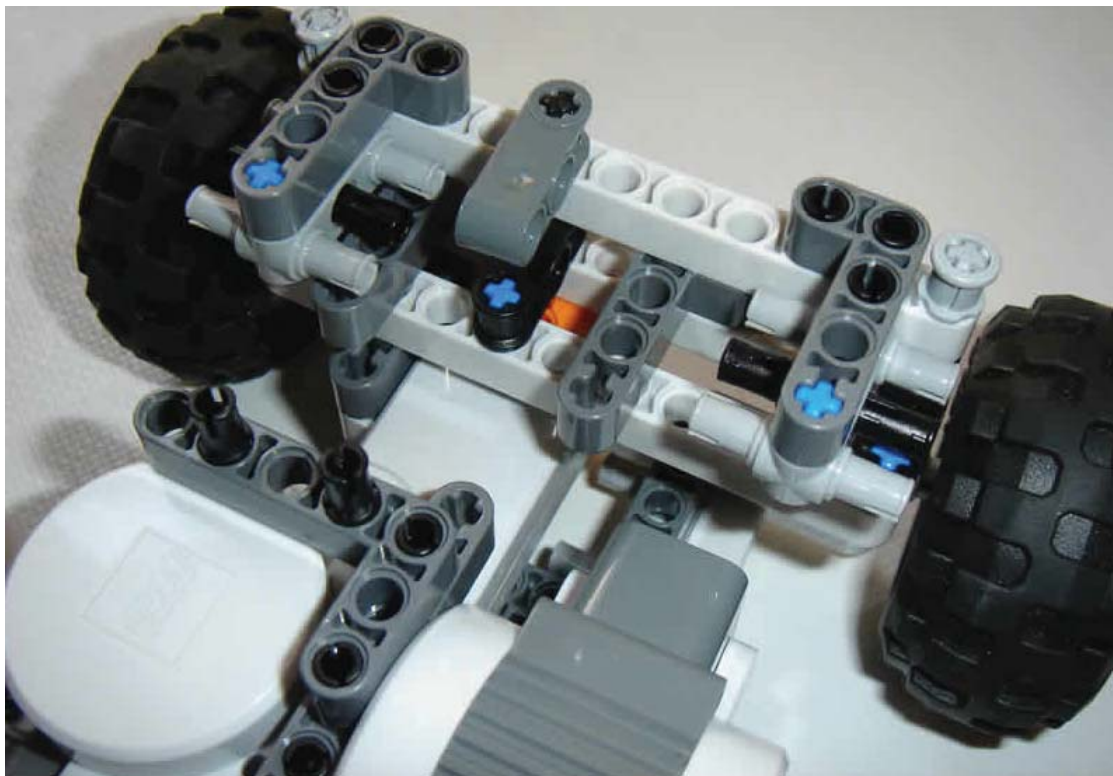


---

BHMA34.

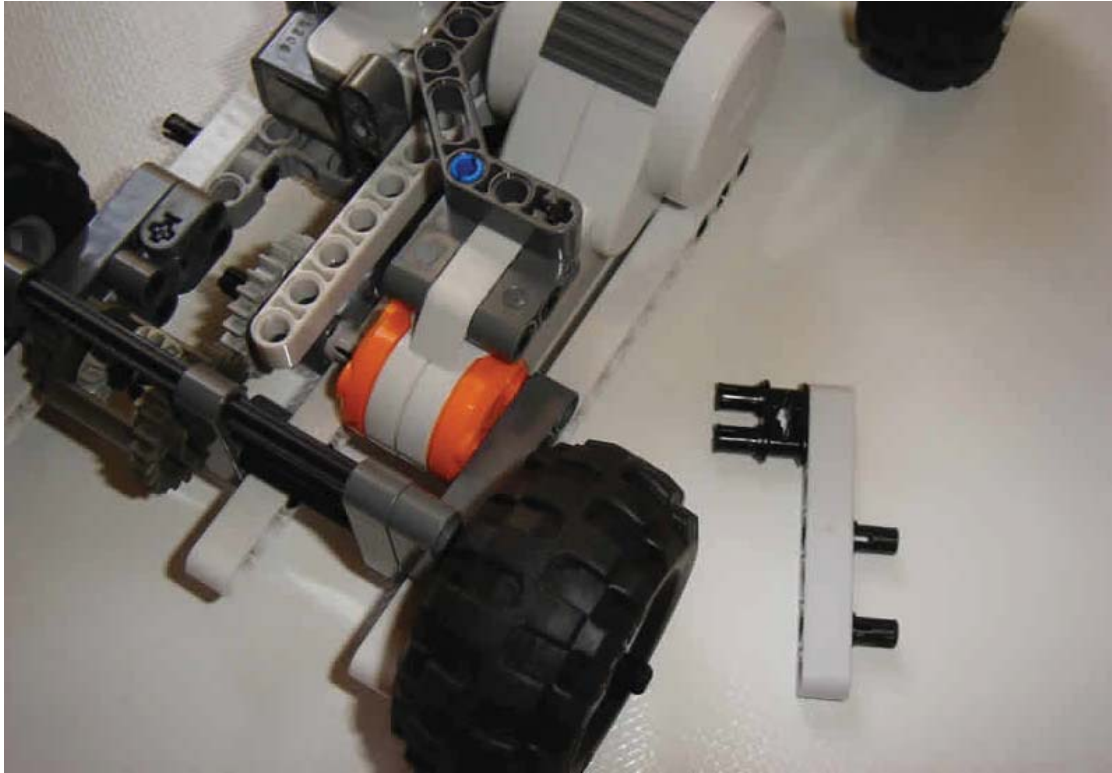


BHMA35.

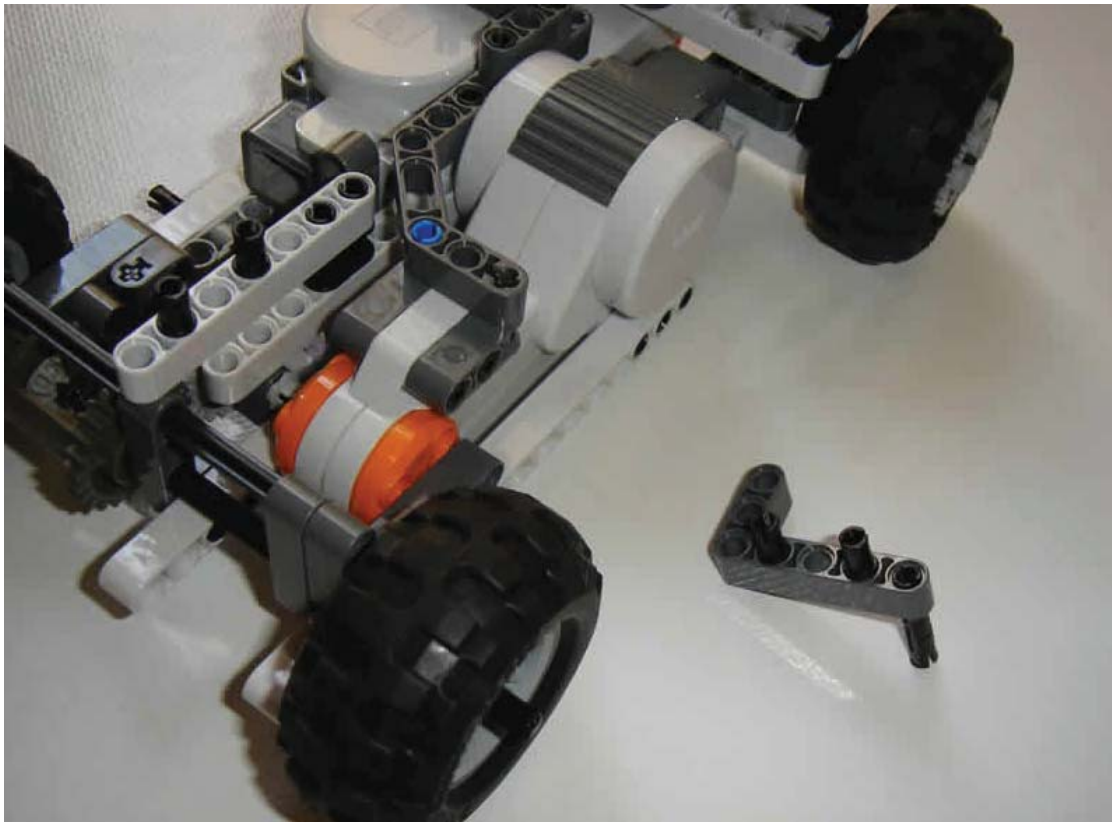


---

BHMA36.

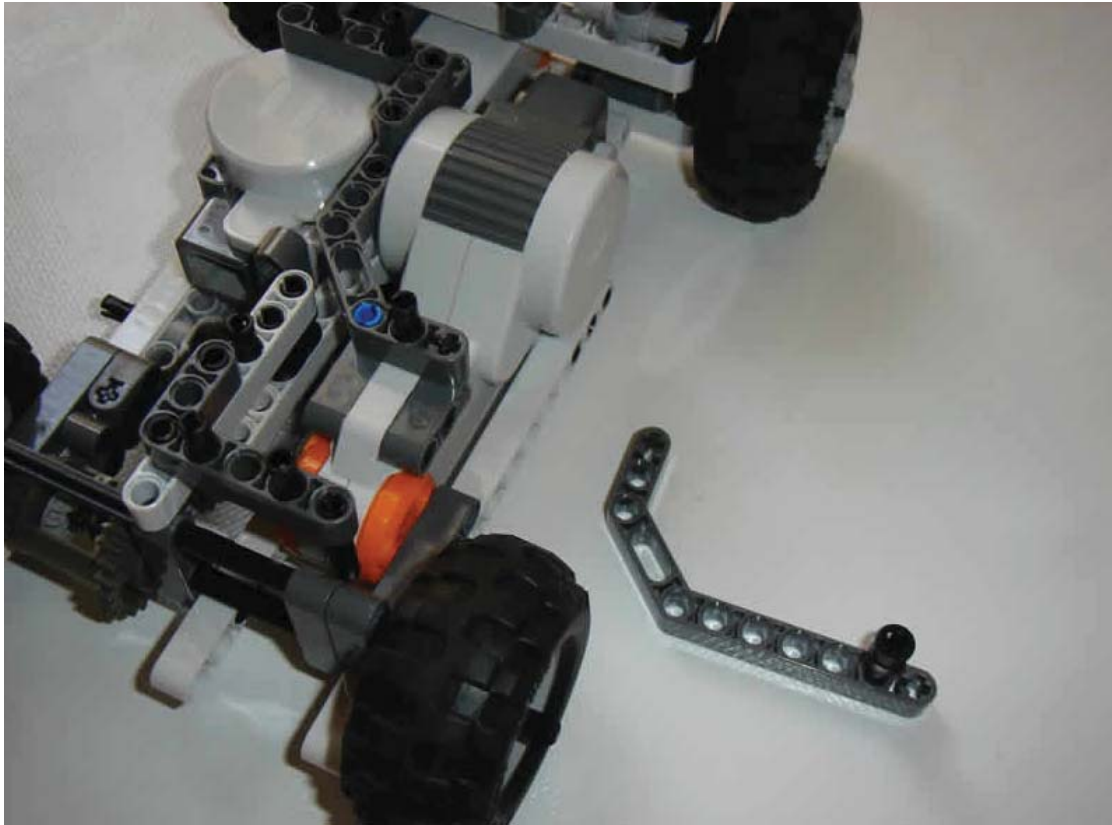


BHMA37.

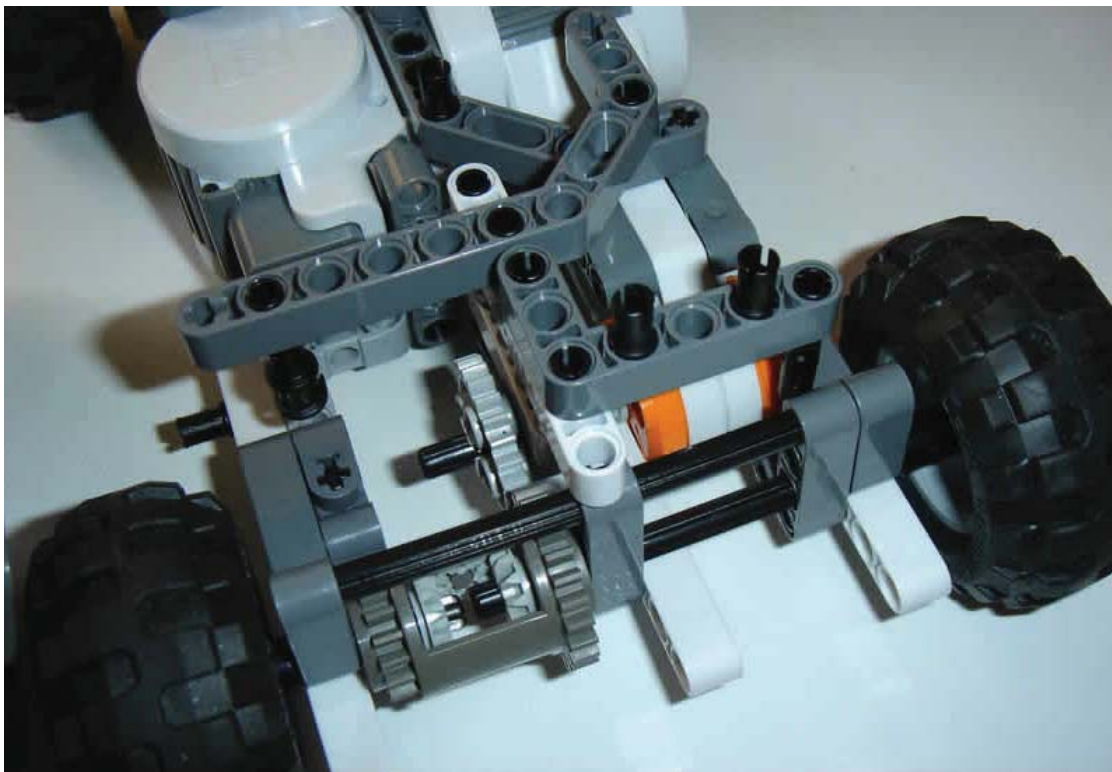


---

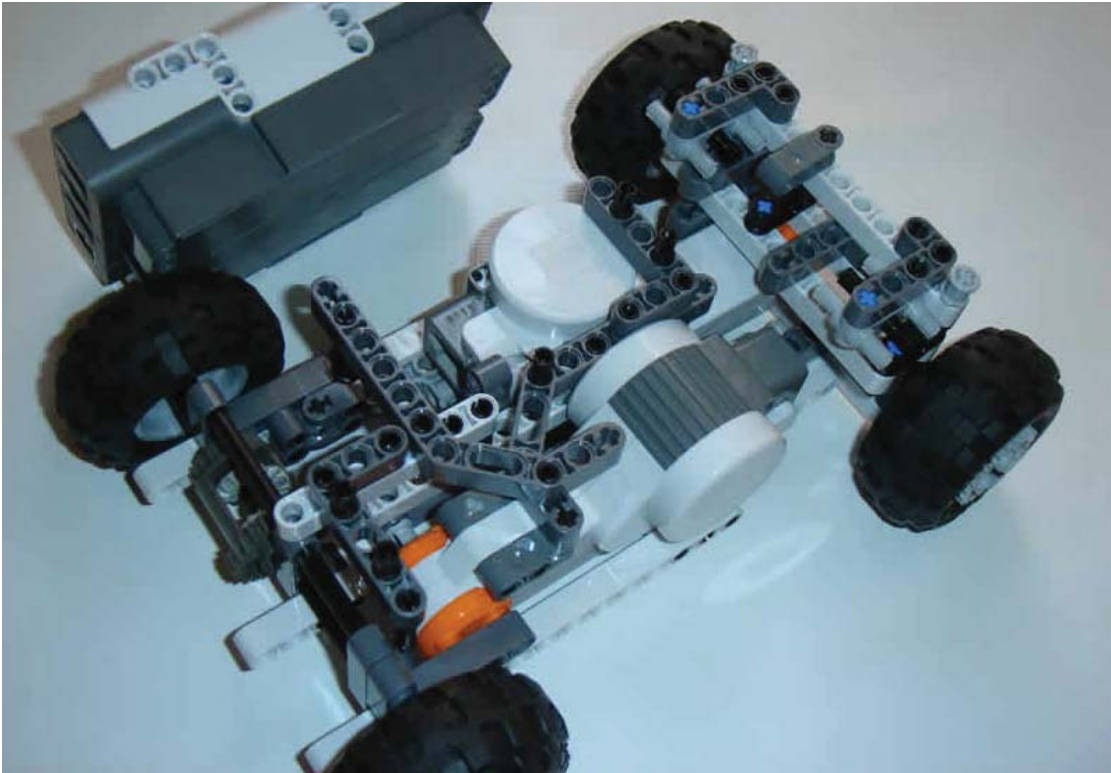
BHMA38.



BHMA39.



BHMA40.



BHMA41.



---

BHMA42.

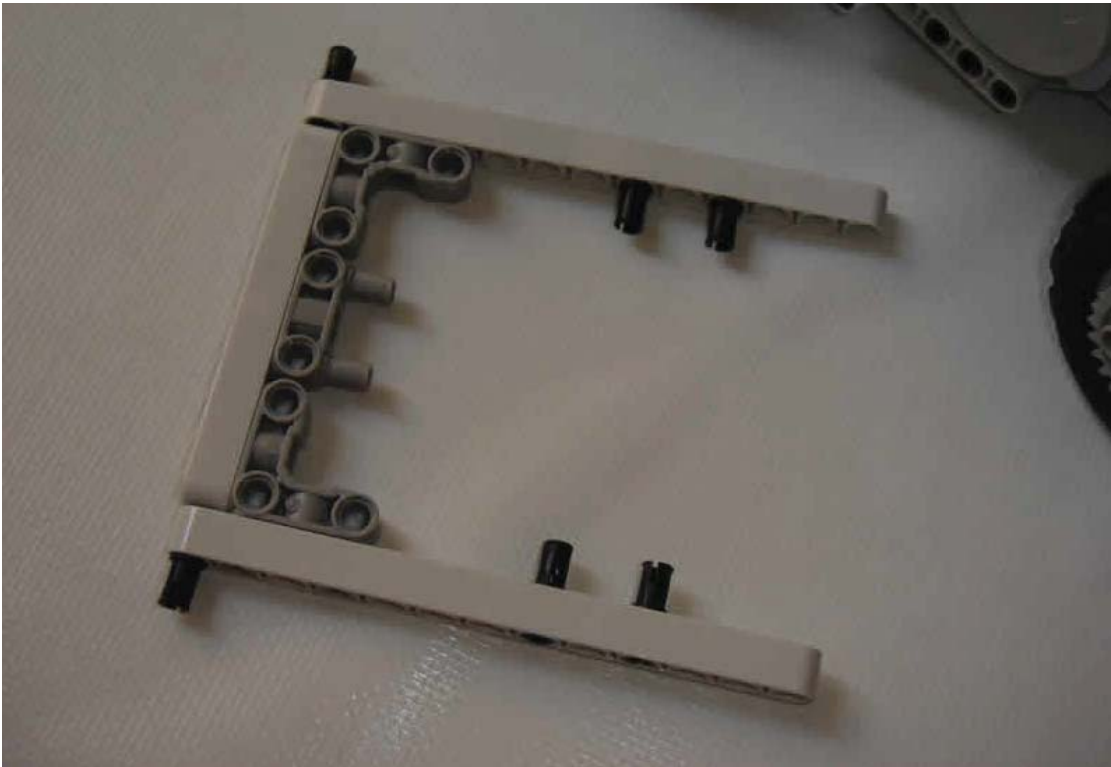


BHMA43.



---

BHMA44.



BHMA45.



---

BHMA46.



BHMA47.



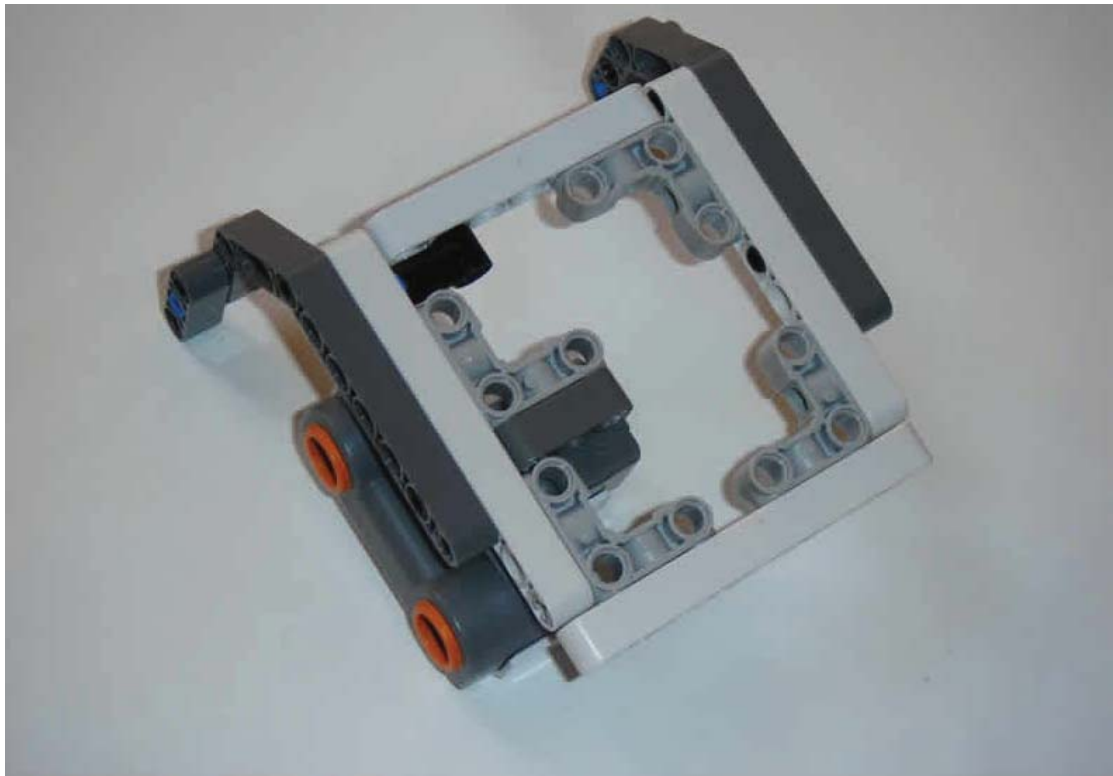


---

BHMA48.



BHMA49.



---

BHMA50.



BHMA51.

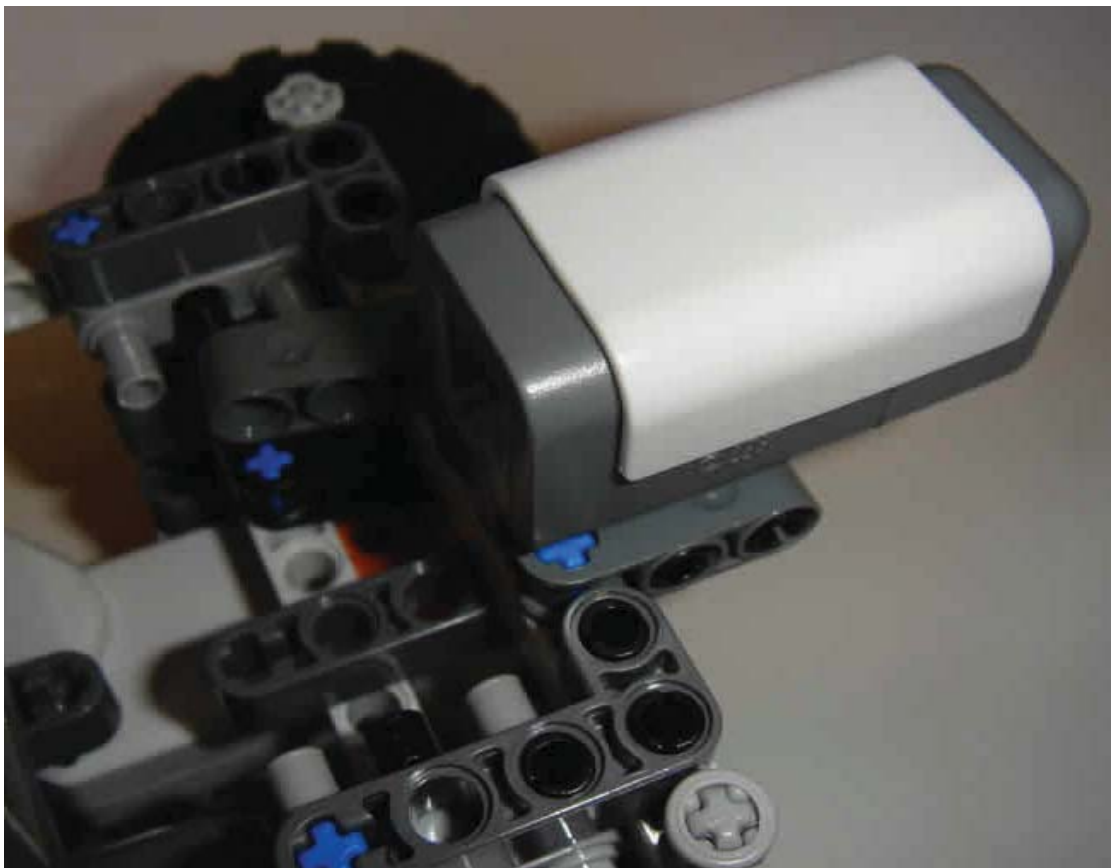


---

BHMA52.



BHMA53.



BHMA54.



---

## ΚΕΦΑΛΑΙΟ 3

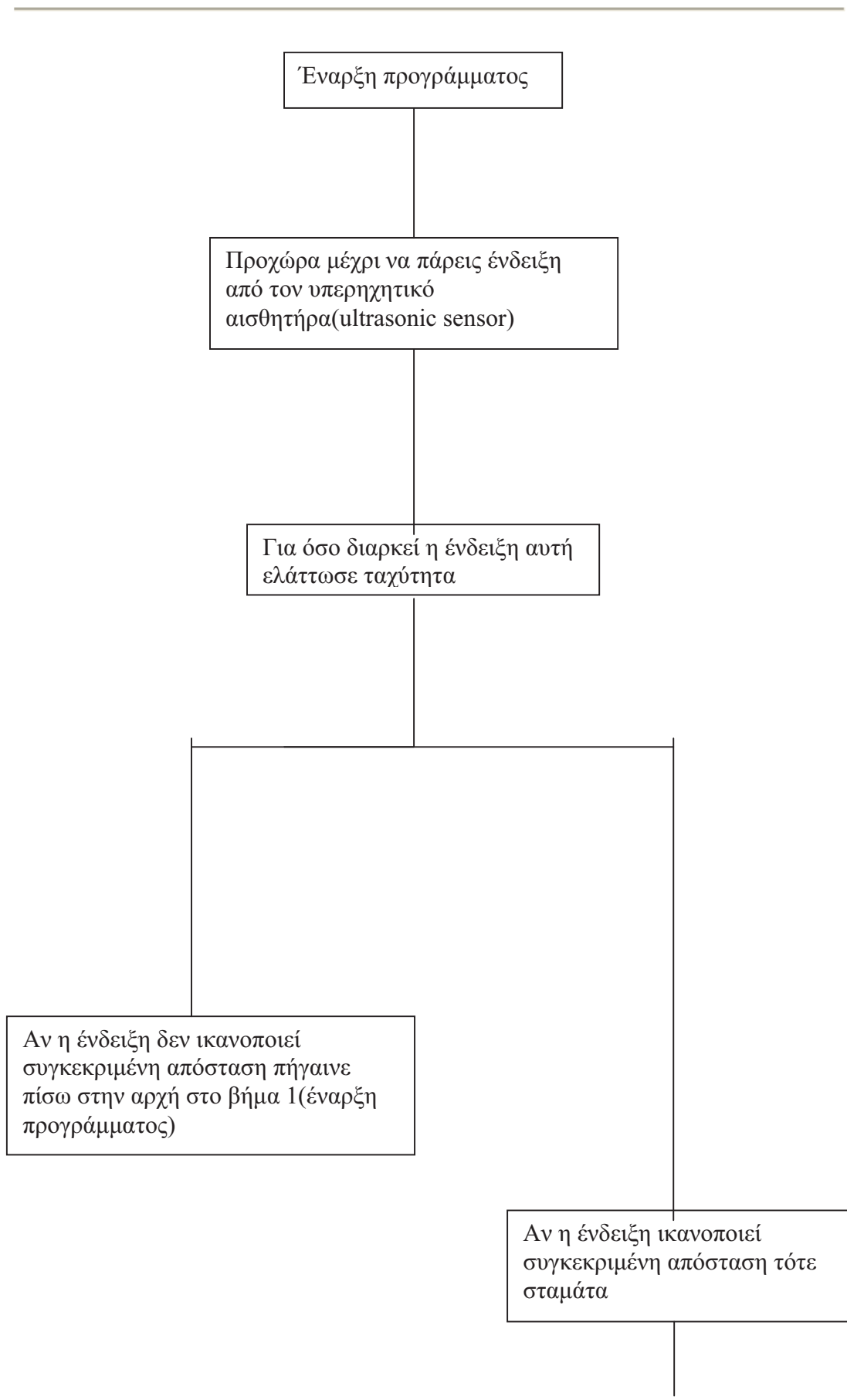
Στο κεφάλαιο αυτό θα αναπτύξουμε αναλυτικά τον αλγόριθμο που χρησιμοποιεί το ρομποτικό μας σύστημα. Ο αλγόριθμος αυτός θα υλοποιηθεί με τη βοήθεια της γλώσσας προγραμματισμού NXT-G την οποία και περιγράψαμε στο προηγούμενο κεφάλαιο.

Πριν αναπτύξουμε τον αλγόριθμο θα προσπαθήσουμε να κατασκευάσουμε πρώτα το λογικό διάγραμμα του προγράμματος ώστε να γίνει πιο κατανοητό το πρόγραμμα και εν συνεχεία θα κατασκευαστεί ο αλγόριθμος βήμα-προς-βήμα.

### ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ

Τα βήματα της στρατηγικής που θα ακολουθήσουμε είναι τα παρακάτω:

- 1) Βρες την κατάλληλη απόσταση χρησιμοποιώντας τον υπερηχητικό αισθητήρα (ultrasonic sensor)
- 2) Προσάρμοσε την κατάλληλη ταχύτητα
- 3) Μετακίνησε κατάλληλα τους μότερες (motors) ώστε να λάβεις θέση για την διαδικασία στάθμευσης.
- 4) Κινήσου σε αντίθετη κατεύθυνση μέχρι να πιεστεί ο αισθητήρας επαφής (touch sensor)
- 5) Κινήσου μπροστά μέχρι να δώσει ένδειξη ο αισθητήρας φωτός και μετά σταμάτα.
- 6) Μόλις λάβεις ένδειξη από τον αισθητήρα ήχου κινήσου προς τα πίσω μέχρι να πιεστεί ο αισθητήρας επαφής (touch sensor)
- 7) Προσάρμοσε κατάλληλα τους μότερες (motors) και κινήσου προς τα εμπρός για συγκεκριμένη απόσταση
- 8) Μετακίνησε προς την αντίθετη κατεύθυνση τους μότερες (motors) μέχρι να επανέλθουν στην αρχική τους κατάσταση και κινήσου προς τα εμπρός για συγκεκριμένη απόσταση.
- 9) Τότε ολοκληρώθηκε η αποστολή σου και σταματάς.



---

Προσάρμοσε τους τροχούς σε κατάλληλη κλίση και κινήσου στην αντίθετη κατεύθυνση

Σταμάτα (STOP)

Επανάφερε τους τροχούς στην προηγούμενη κατάσταση και κινήσου προς τα πίσω.

Μόλις λάβεις ένδειξη από τον αισθητήρα επαφής σταμάτα

Κινήσου προς τα εμπρός μέχρι να λάβεις ένδειξη από τον αισθητήρα φωτός

Σταμάτα (STOP)

---

Μόλις λάβεις ένδειξη από τον αισθητήρα ήχου κινήσου προς τα πίσω μέχρι να λάβεις ένδειξη από τον αισθητήρα επαφής και τότε σταμάτα

Προσάρμοσε τους τροχούς στην κατάλληλη κλίση και κινήσου προς τα εμπρός για συγκεκριμένη απόσταση

Επανάφερε τους τροχούς στην προηγούμενη κατάσταση και κινήσου προς τα εμπρός για συγκεκριμένη απόσταση

Η αποστολή σου τελείωσε και σταμάτα.



---

## ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ NXT-G

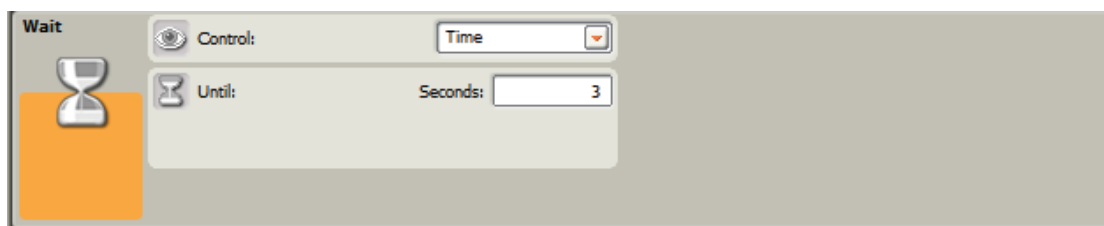
Σε αυτό το βήμα θα προσπαθήσουμε να αναπτύξουμε αναλυτικά τον αλγόριθμο του ρομποτικού μας συστήματος. Θα παραθέσουμε εκτός από την ανάλυση και οπτικό υλικό έτσι ώστε ο κάθε ενδιαφερόμενος να μπορεί να κατασκευάσει εύκολα τον αλγόριθμο από μόνος του.

Παρακάτω κάθε block(κουτάκι) θα αντιστοιχεί σε ένα σύνολο εντολών, δηλαδή κάθε block αποτελεί ένα συγκεκριμένο κομμάτι κώδικα.

Στην αρχή του προγράμματος θα τοποθετήσουμε το block-εντολή wait με χρόνο αναμονής 3 δευτερόλεπτα. Σύμφωνα με το block αυτό το ρομποτικό μας σύστημα θα περιμένει την πάροδο 3 δευτερολέπτων μέχρι να προχωρήσει στο επόμενο block-εντολή. Το block αυτό δεν είναι απαραίτητο για την υλοποίηση του αλγορίθμου αλλά το χρησιμοποιούμε για πρακτικούς σκοπούς έτσι ώστε ο δημιουργός του project να έχει αρκετό χρόνο (3 sec) από την στιγμή που θα εκκινήσει το πρόγραμμα να απομακρυνθεί από τον χώρο που θα υλοποιηθεί το εν λόγω project.

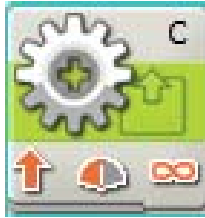


**Σχήμα 1.** Το block-εντολή wait με ρύθμιση χρόνου αναμονής 3 sec



Ξεκινώντας το πρόγραμμά μας θέλουμε ο κινητήρας που δίνει την ώθηση στο ρομποτικό μας σύστημα για να κινηθεί (αυτός που αντιστοιχεί στο port C) να λειτουργήσει με κατεύθυνση προς τα εμπρός με το μέγιστο της δύναμής του (power) στο 60 και με διάρκεια κίνησης απεριόριστη (duration unlimited). Για το λόγο αυτό θα χρησιμοποιήσουμε το block motor.

Το block αυτό είναι:



Σχήμα 2. Το block-εντολή motor με ρυθμίσεις



Χρησιμοποιώντας το προηγούμενο block το ρομπότ μας θα κινείται χωρίς περιορισμούς απεριόριστα. Κάτι τέτοιο όμως πρακτικά δεν θα είχε νόημα. Πρέπει να υπάρχει κάποιος περιορισμός που να υπακούει σε ορισμένες συνθήκες. Στην περίπτωση μας θέλουμε να κινείται το ρομπότ μας μέχρι ο υπερηχητικός αισθητήρας να δώσει ένδειξη ότι η απόσταση που μέτρησε είναι μεγαλύτερη από 15 εκατοστά.

Για το λόγο αυτό θα εισάγουμε το παρακάτω wait block.



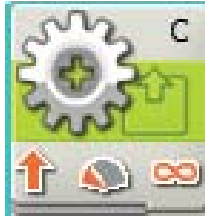
Σχήμα 3. Το block-εντολή wait με ρυθμίσεις



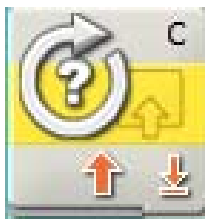
Αφού εκτελέσει το ρομποτικό μας σύστημα αυτές τις κινήσεις το προγραμματίζουμε να κινηθεί με μειωμένη ταχύτητα και πιο συγκεκριμένα με το μέγιστο της δύναμής του στο 40 μέχρι να καλύψει μια συγκεκριμένη απόσταση. Για να μετρήσουμε την απόσταση αυτή θα εισάγουμε ένα block-εντολή το οποίο θα δώσει ένδειξη όταν ο μόντορας (motor) περιστραφεί για συγκεκριμένο αριθμό μοιρών. Το block αυτό ονομάζεται rotation sensor block και στον αλγόριθμό μας το έχουμε

ρυθμίσει να δώσει ένδειξη μετά την ολοκλήρωση μιας περιστροφής, δηλαδή μετά από 360 μοίρες(degrees).

Τα block που χρησιμοποιήσαμε για το συγκεκριμένο κομμάτι του κώδικα είναι τα εξής:



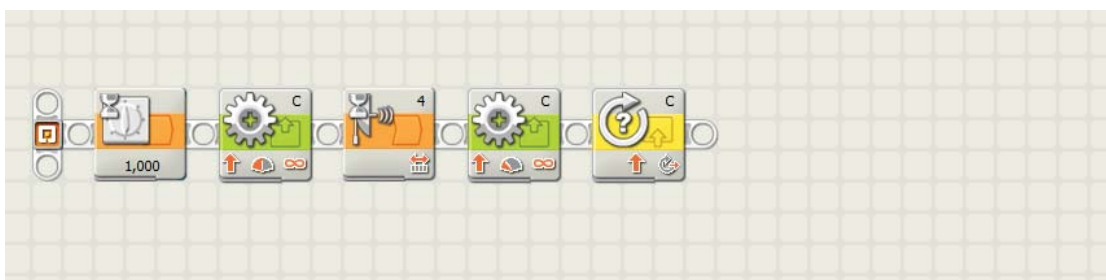
Σχήμα 4. Το block-εντολή motor με ρυθμίσεις



Σχήμα 5. Το block-εντολή rotation sensor με ρυθμίσεις



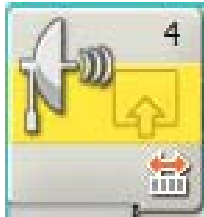
Για να έχουμε μια πιο ολοκληρωμένη μορφή το κομμάτι του κώδικα που αναπτύξαμε μέχρι αυτό το χρονικό σημείο είναι το παρακάτω



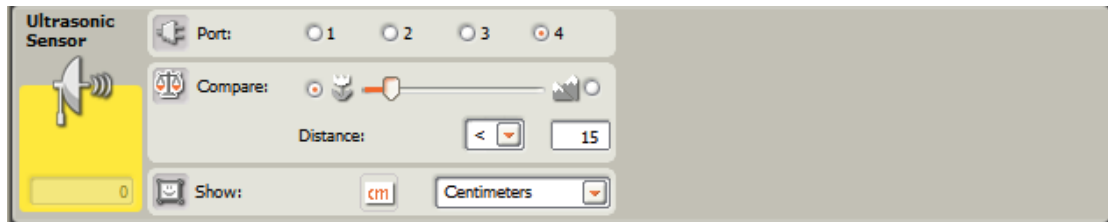
Η επόμενη διαδικασία που απαιτούμε το ρομποτικό μας σύστημα να πραγματοποιήσει είναι πιο σύνθετη καθώς αποτελείται από δυο λειτουργίες που πρέπει να "τρέξουν" ταυτόχρονα.

Προγραμματίζουμε το ρομπότ μας να προχωρήσει στην επόμενη κίνηση όταν ο υπερηχητικός αισθητήρας δίνει ένδειξη για απόσταση μικρότερη από 15 εκατοστά και ταυτόχρονα ο μόντορας έχει περιστραφεί 950 μοίρες.

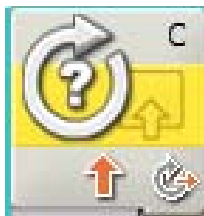
Τα blocks που χρησιμοποιήθηκαν είναι τα εξής



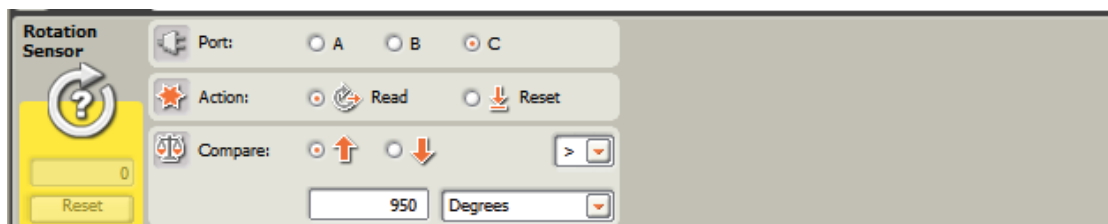
Σχήμα 6 Το block-εντολή ultrasonic sensor με ρυθμίσεις



Το δεύτερο block είναι



Σχήμα 7 Το block-εντολή rotation sensor με ρυθμίσεις



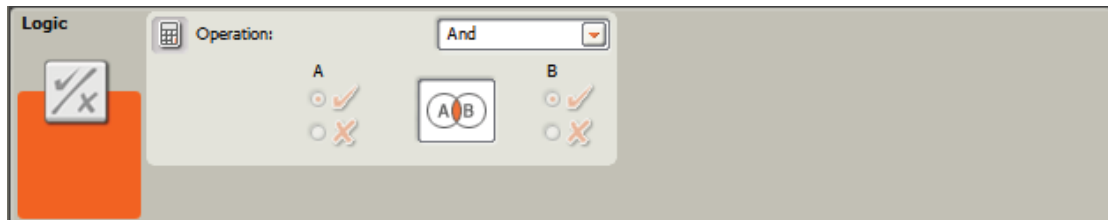
Το πρόβλημα που δημιουργείται καθώς χρησιμοποιούμε τα blocks αυτά είναι πως θα μπορέσουμε να τα συνδέσουμε μεταξύ τους και να μας δώσουν ταυτόχρονη ένδειξη. Για να το ξεπεράσουμε αυτό θα εισάγουμε το block-εντολή logic που υλοποιείται γύρω από την Boole άλγεβρα. Κάθε Block αφού εκτελεστεί εξάγει μια αριθμητική τιμή(1 ή 0) συγκρίνοντας τα δεδομένα που προκύπτουν μετά την ολοκλήρωσή της με αυτά που είχαμε εισάγει ως αρχική τιμή. Αν αυτά συμφωνούν η τιμή που εξάγει είναι 1-αληθές αλλιώς 0-ψευδές. Η τιμή αυτή αποτελεί είσοδο για την εντολή logic που

δέχεται μια αριθμητική τιμή (αριθμό) και την μετατρέπει σε λογική. Το αποτέλεσμα που εξάγει είναι πάλι μια Boolean μεταβλητή 1(true) ή 0(false). Η χρησιμότητα της εντολής αυτής είναι όταν θέλουμε να συνδυάσουμε πολλές εισόδους και να έχουμε μία μόνο έξοδο.

Το logic block είναι το παρακάτω



Σχήμα 8 Το block-εντολή logic με ρυθμίσεις

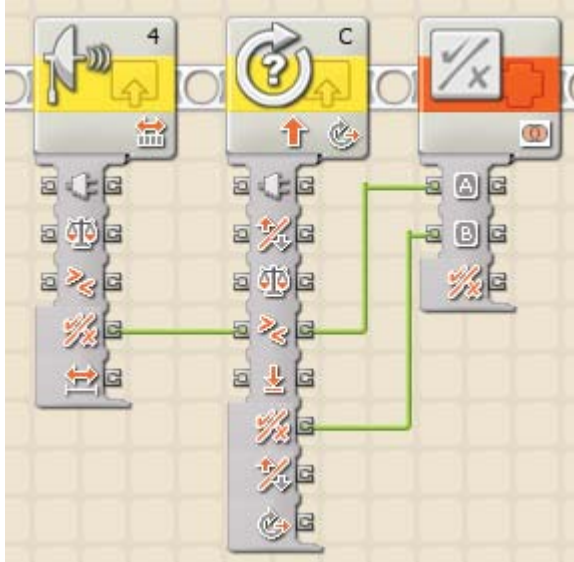


Για να μπορέσουμε τώρα να ενώσουμε αυτά τα τρία Blocks μεταξύ τους για να επικοινωνούν θα χρησιμοποιήσουμε τα data hubs. Τα data hubs μας επιτρέπουν να σχεδιάζουμε data wires από το ένα block στο άλλο χρησιμοποιώντας data plugs. Τα data wires αποτελούν ουσιαστικά τον δίαυλο επικοινωνίας μεταξύ των blocks. Επιτρέπουν δηλαδή στα blocks να ανταλλάσσουν πληροφορίες μεταξύ τους. Οι έξοδοι του ενός αποτελούν τις εισόδους του άλλου. Τα δεδομένα που ανταλλάσσονται μέσω των data hubs πρέπει αν είναι της ίδιας μορφής, δηλαδή οι αριθμητικές έξοδοι ενός block αποτελούν αριθμητικές εισόδους ενός άλλου block.

Για να εμφανιστούν τα data hubs πρέπει στο εκάστοτε block να κάνουμε κλικ κάτω αριστερά. Ένα παράδειγμα ενός τέτοιου data hub αποτελεί το παρακάτω



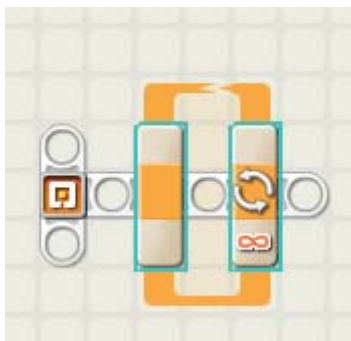
Κάνοντας χρήση των data hubs μπορούμε τώρα να ενώσουμε τα τρία blocks ώστε να έρθουν σε επικοινωνία μεταξύ τους. Το τελικό αποτέλεσμα που προκύπτει μετά από αυτή τη σύνδεση φαίνεται παρακάτω



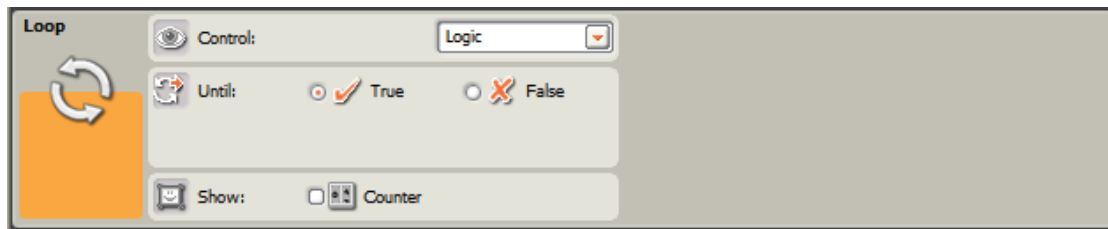
Τα data hubs έχουν ενωθεί με data wires για να μπορούν να επικοινωνούν μεταξύ τους τα blocks

Καθώς ολοκληρώνουμε το κομμάτι αυτό του κώδικα διαπιστώνουμε πως αυτό θα εκτελεστεί μια φορά. Θέλουμε όμως το ρομπότ μας πριν προχωρήσει στην επόμενη κίνησή του να λάβει την τιμή 1(true) από αυτό το σύμπλεγμα των Blocks. Για να το επιτύχει αυτό, το προγραμματίζουμε ώστε να επαναλαμβάνει την διαδικασία αυτή μέχρι να λάβει την τιμή 1(true). Χρειαζόμαστε δηλαδή μια εντολή-block που να δηλώνει επανάληψη. Η εντολή αυτή δίνεται από το block loop. Με άλλα λόγια η εντολή αυτή μας επιτρέπει μια επαναλαμβανόμενη διαδικασία να την προγραμματίζουμε μόλις μια φορά κάνοντας τον αλγόριθμό μας πιο ευσταθή και με μικρότερο υπολογιστικό έργο.

Το loop block είναι το παρακάτω

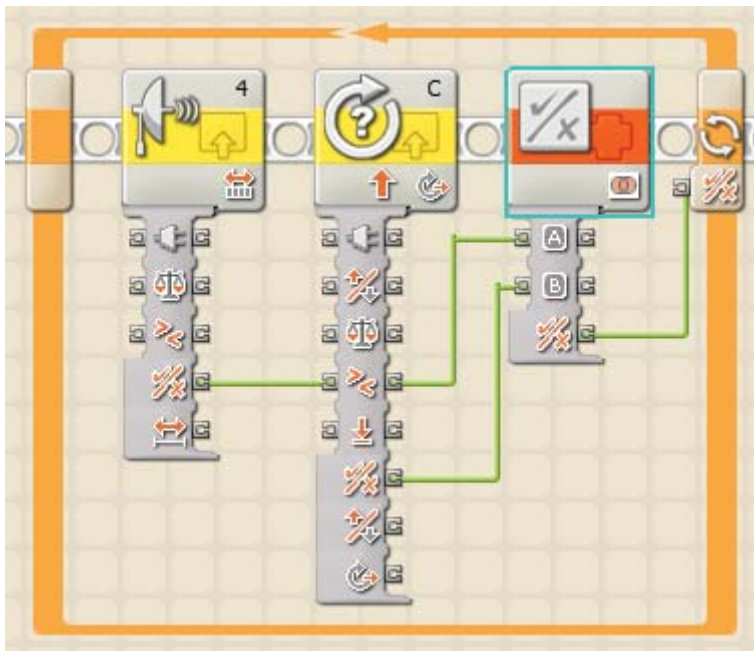


Σχήμα 9 Το block-εντολή loop με ρυθμίσεις



Η συγκεκριμένη loop-εντολή θα δέχεται τις λογικές τιμές true or false. Όσο η τιμή που δέχεται είναι false θα επανέρχεται στην αρχική κατάσταση και θα επαναλαμβάνεται (loop). Μόλις δεχτεί την τιμή true θα διακοπεί η λειτουργία της και το πρόγραμμα θα συνεχίσει την εκτέλεση των επόμενων blocks-εντολών.

Επομένως το κομμάτι του κώδικα με χρήση του loop block θα έχει την παρακάτω μορφή

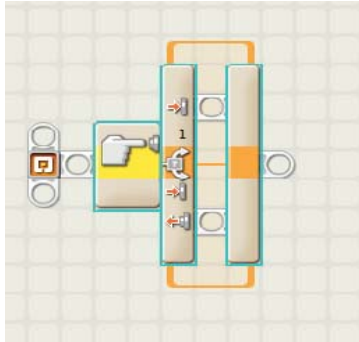


Στη συνέχεια θέλουμε το ρομποτικό μας σύστημα να εκτελέσει μια διαδικασία κάνοντας χρήση της τεχνητής νοημοσύνης( artificial intelligence) με την οποία και είναι εφοδιασμένο. Απαιτούμε δηλαδή η επόμενη κίνηση που θα εκτελέσει να εξαρτάται από την λογική τιμή που θα λάβει ως είσοδο.

Στον συγκεκριμένο αλγόριθμο θέλουμε αν η είσοδος που θα δεχτεί είναι η τιμή 0(false) να μην κάνει τίποτα το ρομπότ και να επιστρέψει στην προηγούμενη κίνηση ,δηλαδή στην loop,ενώ αν η είσοδος είναι η τιμή 1(true) να ακινητοποιηθεί, δηλαδή να σταματήσει ο μόντορας (Motor) και να εξάγει την λογική τιμή 1 που να αντιστοιχεί στην ολοκλήρωση αυτής της διαδικασίας.

Για να επιτύχουμε αυτήν την σύνθετη διαδικασία θα κάνουμε χρήση του block switch. Το block αυτό μας επιτρέπει οι αποφάσεις που θα λάβουμε να εξαρτώνται αποκλειστικά από την υπόθεση που προηγείται.

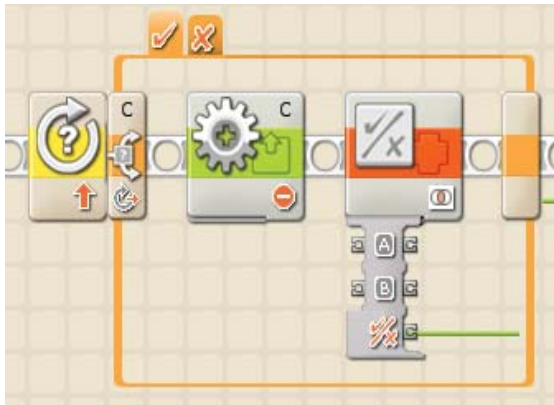
Το switch block που θα χρησιμοποιήσουμε είναι το παρακάτω



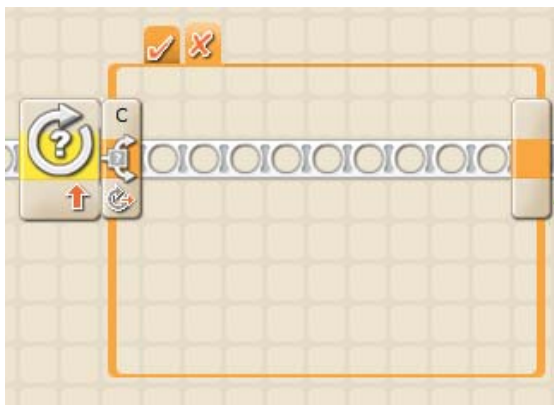
Σχήμα 10. Το block-εντολή switch με ρυθμίσεις



Συμπερασματικά η τελική διπλή μορφή που θα πάρει το switch block που θα χρησιμοποιήσουμε είναι η εξής



Το switch block όταν η είσοδος που δέχεται είναι η λογική τιμή 1(true)



Το switch block όταν η είσοδος που δέχεται είναι η λογική τιμή 0(false)





Αφού εκτελέσουμε το συγκεκριμένο κομμάτι κώδικα συμπεραίνουμε πως είναι αυτό που δίνει την δυνατότητα στο ρομποτικό μας σύστημα μετά από επαναλαμβανόμενες μετρήσεις να καταλήξει στην ελάχιστη απόσταση (min distance) που απαιτείται ώστε να πραγματοποιήσει με επιτυχία τη διαδικασία στάθμευσης.

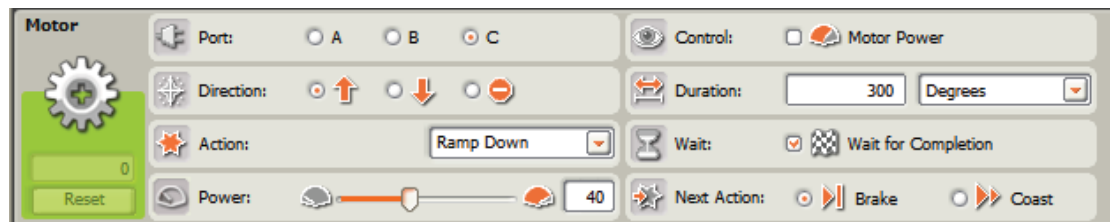
Το επόμενο κομμάτι κώδικα θα πρέπει να περιλαμβάνει τις απαραίτητες κινήσεις ώστε η παραπάνω διαδικασία να πραγματοποιηθεί.

Αφού καταλήξαμε στην ελάχιστη απόσταση θέλουμε το ρομπότ να εκτελέσει μια σειρά από κινήσεις τις οποίες και θα προγραμματίσουμε. Συγκεκριμένα θέλουμε να καλύψει απόσταση το ρομπότ μας που να αντιστοιχεί σε περιστροφή 300 μοιρών των τροχών(μότορας C) με αυξανόμενη δύναμη(ramp up) και με μέγιστό της το 40. Στη συνέχεια αφού ολοκληρώσει αυτή τη διαδικασία (wait for completion) να ακινητοποιηθεί (next action brake).

Το block που χρησιμοποιήθηκε είναι το εξής.



Σχήμα 11. Το block-εντολή motor με ρυθμίσεις

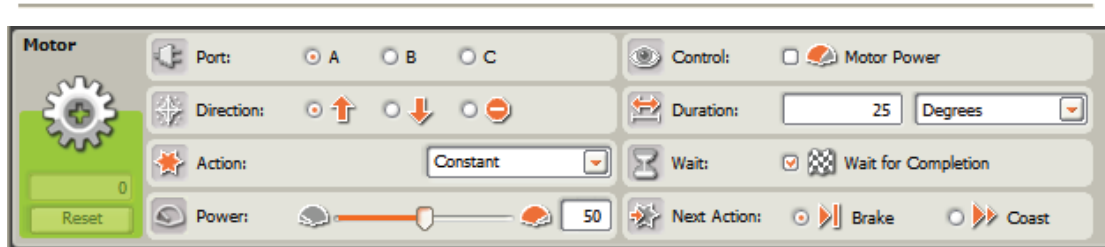


Στη συνέχεια θέλουμε οι μπροστινοί τροχοί(motor A) του ρομπότ να συγκλίνουν προς τα δεξιά κατά 25 μοίρες απότομα με μέγιστο της δύναμης στο 50 και αφού ολοκληρωθεί η διαδικασία αυτή οι πίσω τροχοί (motor C) να δώσουν ώθηση στο ρομποτικό μας σύστημα να κινηθεί προς τα πίσω( αντίθετη πορεία) με αυξανόμενη ένταση που να αγγίζει την μέγιστη τιμή ίση με 40. Αφού διανύσει απόσταση που θα αντιστοιχεί σε περιστροφή των τροχών (motor C) κατά 540 μοίρες να σταματήσει σταδιακά καθώς θα ξεκινάει ταυτόχρονα η επόμενη κίνηση.

Για να πετύχουμε αυτή την κίνηση χρησιμοποιήσαμε την ακόλουθη αλληλουχία από blocks.



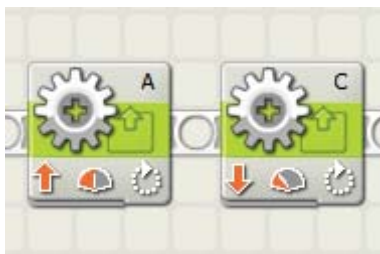
Σχήμα 12. Το block-εντολή motor A με ρυθμίσεις



Σχήμα 13. Το block-εντολή motor C με ρυθμίσεις



Συνολικά δηλαδή χρησιμοποιήθηκε το



Καθώς ολοκληρώνεται η προηγούμενη διαδικασία θέλουμε οι μπροστινοί τροχοί να συγκλίνουν προς τα αριστερά (αντίθετη κατεύθυνση με πριν) κατά 50 μοίρες απότομα (constant) με μέγιστη τιμή της δύναμης στο 50 και αφού ολοκληρωθεί η διαδικασία αυτή, οι πίσω τροχοί(motor C) να κινηθούν με αντίθετη κατεύθυνση με δύναμη 40 μέχρι να πιεστεί ο αισθητήρας επαφής(touch sensor) οπότε και να σταματήσει να κινείται το ρομπότ.

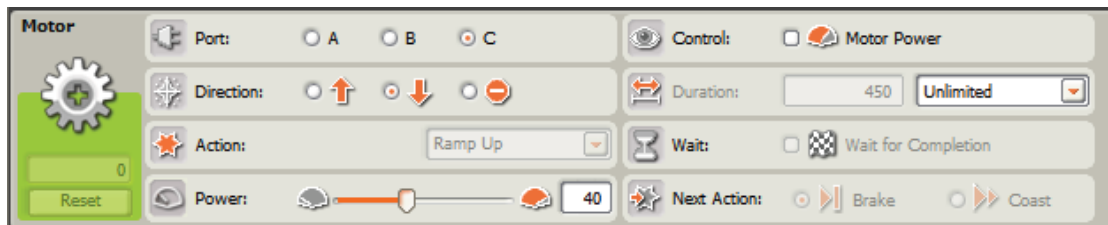
Τα blocks που χρησιμοποιήθηκαν σε αυτό το κομμάτι του κώδικα είναι τα εξής



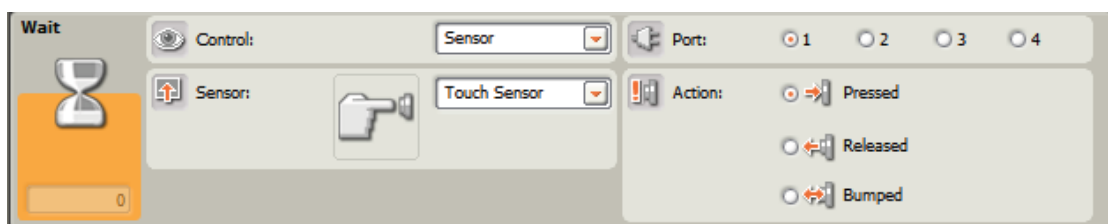
Σχήμα 14. Το block-εντολή motor A με ρυθμίσεις



Σχήμα 15. Το block-εντολή motor C με ρυθμίσεις



Σχήμα 16. Το block-εντολή wait με ρυθμίσεις



Συνοπτικά χρησιμοποιήθηκε το παρακάτω κομμάτι κώδικα



Στη συνέχεια θέλουμε το ρομποτικό μας σύστημα να κινηθεί προς τα εμπρός (motor C) με μέγιστη τιμή της δύναμης στο 40 και ταυτόχρονα οι μπροστινοί τροχοί να συγκλίνουν προς τα δεξιά κατά 25 μοίρες ώστε να επανέλθουν στην αρχική τους κατάσταση. Το ρομπότ να σταματήσει να κινείται μόλις ο αισθητήρας φωτός λάβει ένδειξη μεγαλύτερη από το 30 που έχει παραχθεί από την αυτόνομη γεννήτρια φωτός με την οποία είναι εξοπλισμένος και να ολοκληρώσει την διαδικασία κινούμενο προς την αντίθετη κατεύθυνση με αυξανόμενη δύναμη με μέγιστο 40 μέχρι να διανύσει απόσταση που να αντιστοιχεί σε περιστροφή κατά 200 μοίρες του μότορα C.

Τα blocks που χρησιμοποιήθηκαν είναι τα παρακάτω



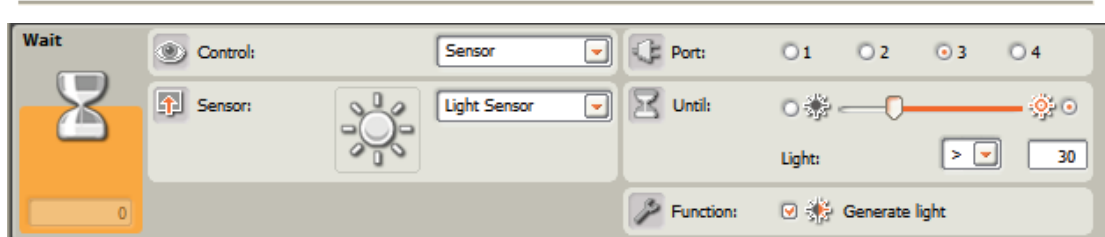
Σχήμα 17. Το block-εντολή motor C με ρυθμίσεις



Σχήμα 18. Το block-εντολή motor A με ρυθμίσεις



Σχήμα 19. Το block-εντολή wait με ρυθμίσεις



Σχήμα 20. Το block-εντολή motor C με ρυθμίσεις



Σχήμα 21. Το block-εντολή motor C με ρυθμίσεις



Η τελική ακολουθία των blocks που χρησιμοποιήθηκε στο συγκεκριμένο κομμάτι του κώδικα είναι η παρακάτω



Στο σημείο αυτό πραγματοποιήθηκε η ευθεία πορεία της διαδικασίας στάθμευσης. Για να κάνουμε τον αλγόριθμο πιο σύνθετο θα προσπαθήσουμε να κατασκευάσουμε και το κομμάτι του κώδικα που να αντιστοιχεί στην αντίστροφη διαδικασία.

Θέλουμε η διαδικασία αυτή να εκκινεί μόλις ο αισθητήρας ήχου λάβει ένδειξη (sound sensor) μεγαλύτερη από το 50. Τότε οι μπροστινοί τροχοί(motor A) να συγκλίνουν προς τα δεξιά κατά 25 μοίρες με μέγιστη τιμή της δύναμης στο 50 και αφού ολοκληρωθεί η συγκεκριμένη διαδικασία οι πίσω τροχοί (motor C) να κινηθούν προς την αντίθετη κατεύθυνση με δύναμη 40 μέχρι να πιεστεί ο αισθητήρας επαφής (touch sensor) οπότε και να σταματήσει το ρομπότ.

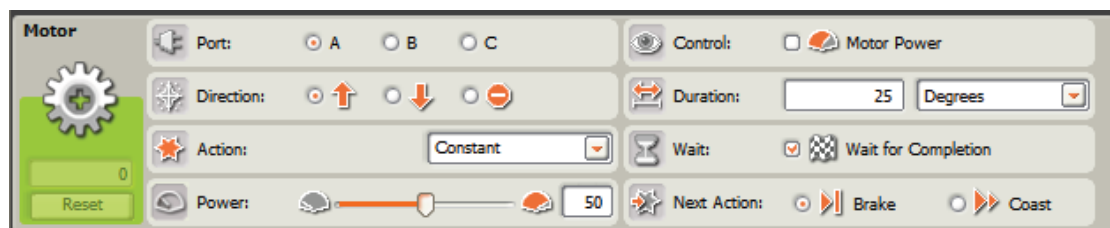
Τα blocks-εντολές που χρησιμοποιήθηκαν είναι τα παρακάτω

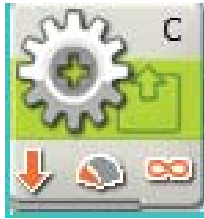


Σχήμα 22. Το block-εντολή wait με ρυθμίσεις



Σχήμα 23. Το block-εντολή motor A με ρυθμίσεις





Σχήμα 24. Το block-εντολή motor C με ρυθμίσεις



Σχήμα 25. Το block-εντολή wait με ρυθμίσεις

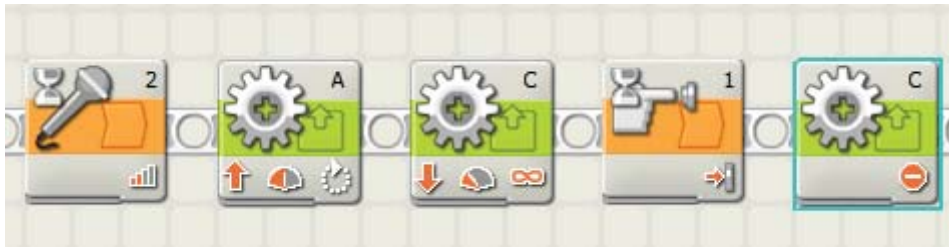


Σχήμα 26. Το block-εντολή motor με ρυθμίσεις





Συνοπτικά το κομμάτι του κώδικα που χρησιμοποιήσαμε είναι το παρακάτω



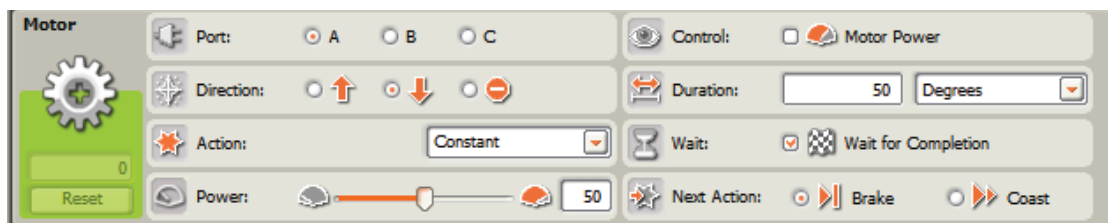
Η αντίστροφη διαδικασία έχει σχεδόν ολοκληρωθεί. Χρειάζεται να δημιουργήσουμε το τελευταίο κομμάτι του κώδικα.

Θέλουμε τώρα οι μπροστινοί τροχοί (motor A) να συγκλίνουν προς τα αριστερά κατά 50 μοίρες με μέγιστη δύναμη στο 50 και αφού ολοκληρωθεί η κίνηση αυτή ο μόντορας C να δώσει ώθηση στους πίσω τροχούς να κινηθούν προς τα εμπρός με δύναμη 40 και για απόσταση που θα αντιστοιχεί σε περιστροφή των τροχών κατά 400 μοίρες. Αμέσως μετά ο μόντορας A να συγκλίνει προς τα δεξιά κατά 25 μοίρες με απότομη κίνηση των τροχών και με δύναμη που να αντιστοιχεί στο 50.

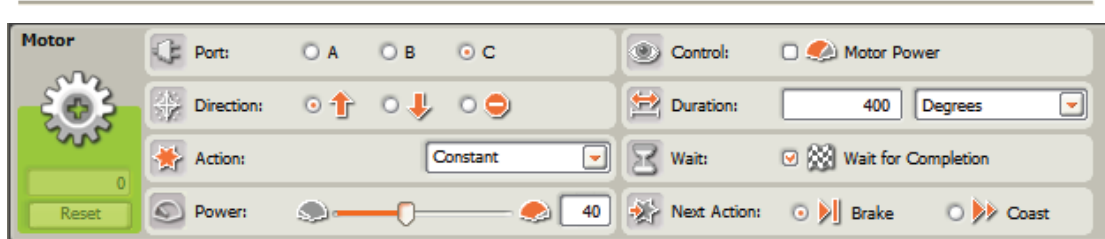
Τα blocks που χρησιμοποιήσαμε μέχρι τώρα είναι τα εξής



Σχήμα 27. Το block-εντολή motor A με ρυθμίσεις



Σχήμα 28. Το block-εντολή motor C με ρυθμίσεις



Σχήμα 29. Το block-εντολή motor A με ρυθμίσεις



Συνολικά το Block μας είναι το παρακάτω



Θέλουμε τέλος το ρομπότ να κινηθεί προς τα εμπρός (motor C) με δύναμη 40 και για απόσταση που να αντιστοιχεί σε περιστροφή των τροχών κατά 150 μοίρες. Μόλις ολοκληρωθεί η διαδικασία αυτή ο μόντορας A (motor A) να συγκλίνει προς τα δεξιά κατά 25 μοίρες με δύναμη στο 50 αναμένοντας και εδώ την ολοκλήρωση αυτής της διαδικασίας. Στη συνέχεια ο μόντορας C (motor C) θα δώσει ώθηση στους πίσω τροχούς να κινηθούν προς τα εμπρός με ταχύτητα 40 και για περιστροφή των τροχών κατά 500 μοίρες. Τέλος οι μπροστινοί τροχοί θα συγκλίνουν προς τα αριστερά κατά 25 μοίρες απότομα με δύναμη στο 50 για να επανέλθουν στην αρχική τους κατάσταση και αφού ολοκληρωθεί η διαδικασία αυτή θα σταματήσει το ρομπότ οπότε και τελειώνει η αντίστροφη διαδικασία στάθμευσης

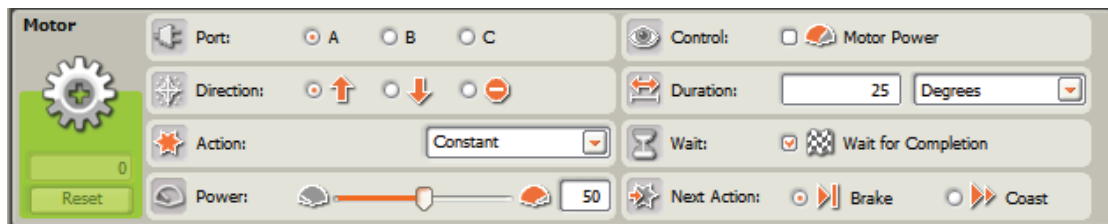
Τα blocks που χρησιμοποιήθηκαν είναι τα παρακάτω



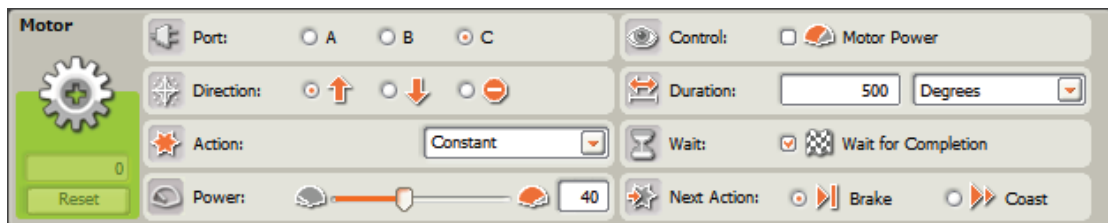
Σχήμα 30. Το block-εντολή motor C με ρυθμίσεις



Σχήμα 31. Το block-εντολή motor A με ρυθμίσεις



Σχήμα 32. Το block-εντολή motor C με ρυθμίσεις

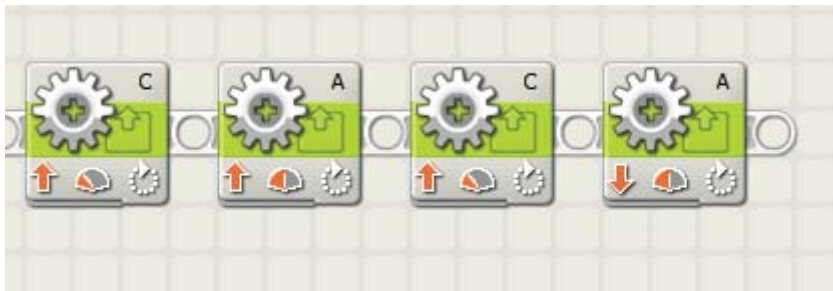




Σχήμα 33. Το block-εντολή motor A με ρυθμίσεις



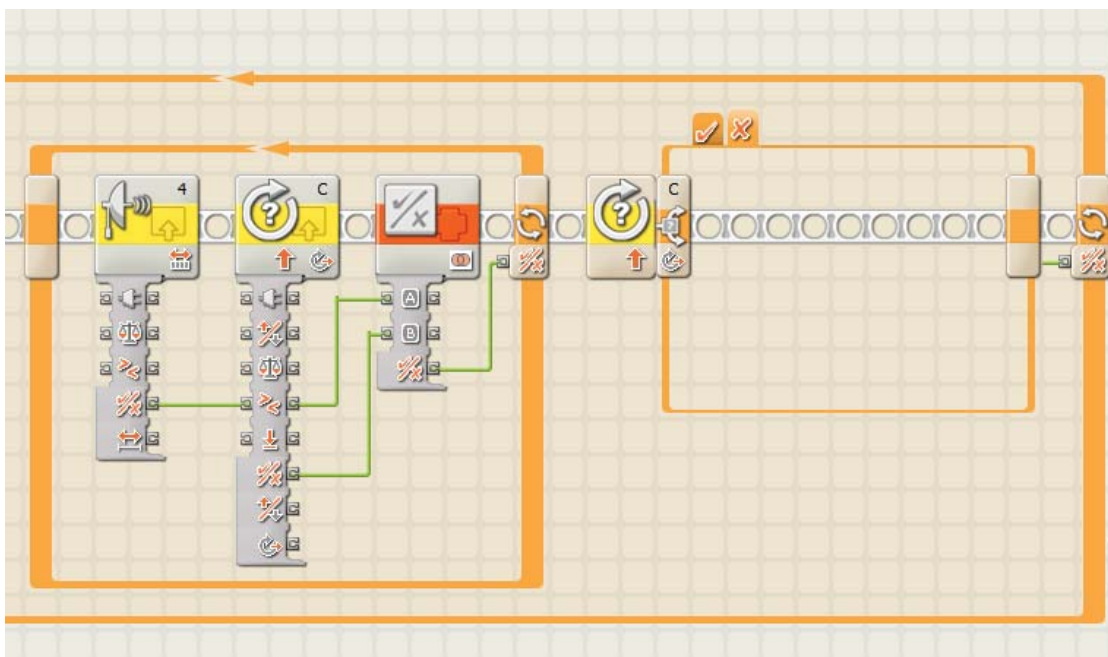
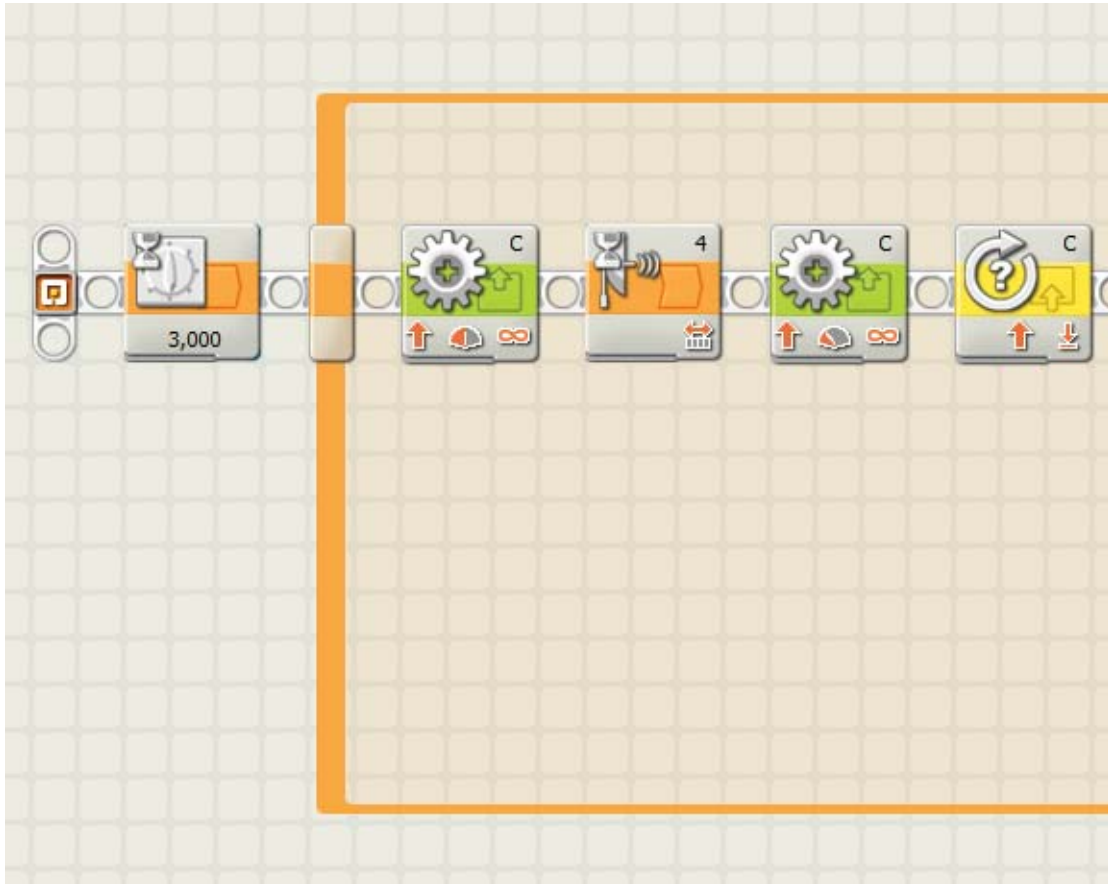
Το συνολικό κομμάτι κώδικα είναι το παρακάτω

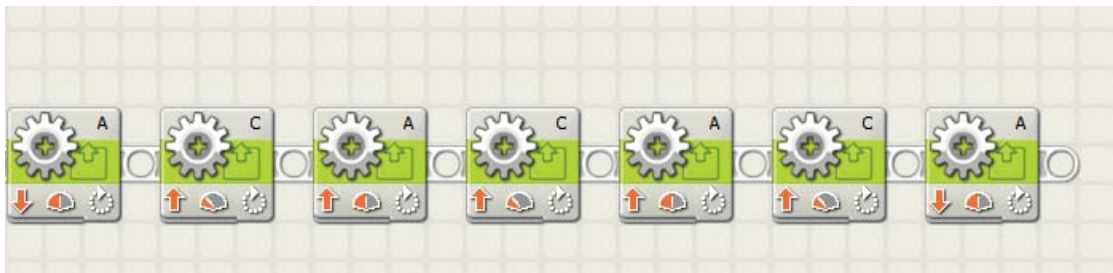
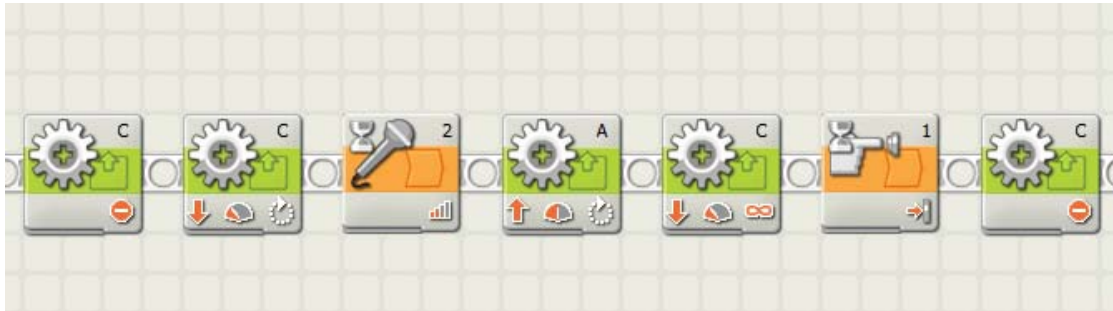
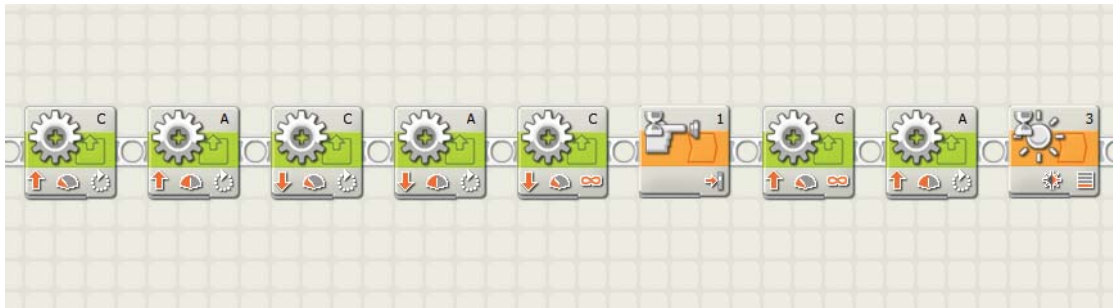


Το ολικό κομμάτι του αλγόριθμου που αντιστοιχεί στην αντίστροφη διαδικασία στάθμευσης δίνεται παρακάτω



Τέλος ο συνολικός αλγόριθμος που αντιστοιχεί στο πρόγραμμά μας είναι ο





---

## BIBΛΙΟΓΡΑΦΙΑ

- 1) LEGO Mindstorms NXT-G Programming Guide-James Floyd Kelly
- 2) Introduction to Control Systems – Eric Salinas
- 3) Introduction to Robotics-Vikram Kapila
- 4) Robotics-Rajat Khare, Ashok Sapra, Ishan Gupta, Vipin Kumar, Anuj Khare, Tarun Wig, and Monika Chawla.
- 5) Σημειώσεις ρομποτικής-Τεχνολογικό εκπαιδευτικό ίδρυμα(Τ.Ε.Ι.) τμήμα μηχανολογίας
- 6) Robotics-Wikipedia
- 7) Ειδικό θέμα ρομποτικού προγραμματισμού-Βασιλειάδης Χρήστος, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, τμήμα Μαθηματικών
- 8) Εισαγωγή στη Ρομποτική-Νίκος Βλάσσης, Τμήμα Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης
- 9) Εισαγωγικές Έννοιες Εργαστηρίου Ρομποτικής-Σημειώσεις του Τμήματος Μηχανολογίας στη Σχολή Τεχνολογικών Εφαρμογών του Α.Τ.Ε.Ι. Κρήτης.
- 10) Robotics-Ashish Dixit
- 11) Robotics-Integrated Manufacturing systems
- 12) Machinevision-Part I
- 13) Machinevision-Part II
- 14) Introductory Workshop NXT BASICS
- 15) LEGO Mindstorms NXT-Hacker's guide
- 16) Programming LEGO Mindstorms NXT