

ΠΟΛΥΤΕΧΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ
ΚΑΙ ΔΙΟΙΚΗΣΗΣ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

*“ΕΛΕΓΧΟΣ ΚΑΙ ΚΑΤΑΣΚΕΥΗ
ΑΥΤΟΪΣΟΡΡΟΠΟΥΜΕΝΟΥ ΔΙΤΡΟΧΟΥ ΡΟΜΠΟΤ ΣΕ
ΠΛΑΤΦΟΡΜΑ LEGO MINDSTORMS NXT/MATLAB
SIMULINK”*



ΜΥΚΟΝΙΑΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΧΑΝΙΑ, 2010

ΕΥΧΑΡΙΣΤΙΕΣ

Καταρχήν θα ήθελα να ευχαριστήσω θερμά τον κ. Αναστάσιο Πουλιέζο για την πολύτιμη βοήθεια του, την καθοδήγησή του και τις πολύτιμες συμβουλές του καθ' όλη την διάρκεια της εκπόνησης της παρούσας εργασίας.

Επίσης ευχαριστώ θερμά των κ. Αρναουτάκη Νεκτάριο για την παραχώρηση του απαραίτητου εργαστηριακού εξοπλισμού αλλά και την βοήθειά του, η οποία ήταν καταλυτική για να ξεπεραστούν διάφορες τεχνικές δυσκολίες που παρουσιάστηκαν.

Στη συνέχεια θα ήθελα να ευχαριστήσω θερμά την εξεταστική επιτροπή, τους κ. Νικόλαο Τσουρβελούδη και τον κ. Γιάννη Μαρινάκη, οι οποίοι δέχτηκαν με πολύ ευχαρίστηση να διορθώσουν την παρούσα εργασία και να μου δώσουν πολύτιμες συμβουλές για την βελτίωσή της.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου, την μητέρα μου Κατερίνα, τον πατέρα μου Παναγιώτη, τον αδελφό μου Νίκο και την συγγάτοικο μου Αναστασία για την ηθική υποστήριξη που μου παρείχαν.

Διπλωματική Εργασία Μυκονιάτη Κωνσταντίνου με θέμα :

**“ΕΛΕΓΧΟΣ ΚΑΙ ΚΑΤΑΣΚΕΥΗ
ΑΥΤΟΪΣΟΡΡΟΠΟΥΜΕΝΟΥ ΔΙΤΡΟΧΟΥ
ΡΟΜΠΟΤ ΣΕ ΠΛΑΤΦΟΡΜΑ LEGO
MINDSTORMS NXT/MATLAB SIMULINK”**

Επιβλέπων Καθηγητής :

Αναστάσιος Πουλιέζος

Εξεταστική Επιτροπή :

Αναστάσιος Πουλιέζος

Νικόλαος Τσουρβελούδης

Γιάννης Μαρινάκης

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ.....	2
ΠΕΡΙΛΗΨΗ.....	11
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	12
ΠΕΡΙΕΧΟΜΕΝΑ ΤΟΥ LEGO MINDSTORMS NXT & EDUCATION RESOURCE SET	13
1.1 Το NXT	14
1.2 Αισθητήρες (Sensors).....	16
1.2.1 Αισθητήρας Αφής (Touch Sensor).....	16
1.2.2 Αισθητήρας Ήχου (Sound Sensor Sensor)	16
1.2.3 Αισθητήρας Φωτός (Light Sensor).....	17
1.2.4 Αισθητήρας Υπέρηχων (Sonar Sensor)	18
1.2.5 Αισθητήρας Χρώματος (Color Sensor)	19
1.2.6 Γυροσκοπικός Αισθητήρας (Gyroscopic Sensor).....	21
1.2.7 Αισθητήρας Επιτάχυνσης (Acceleration Sensor).....	21
1.2.8 Μαγνητικός Αισθητήρας (Compass Sensor)	23
1.2.9 Αισθητήρας εντοπισμού Υπέρυθρων (infrared Seeker Sensor).....	24
1.3 Συσκευές εξόδου (Actuators)	27
1.3.1 Σερβοκινητήρες (Servomotors)	27
1.3.2 Λαμπτήρες (Lamps)	28
1.4 Πρόσθετες συσκευές που χρησιμοποιήθηκαν	28
1.4.1 USB Bluetooth Dongle	28
1.4.2 Xbox360 Analog PC Gamepad	28
ΚΕΦΑΛΑΙΟ 2: ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ	29
2.1 Γλώσσες Προγραμματισμού για το NXT.....	29
2.1.1 Επίσημη γλώσσα προγραμματισμού	29
2.1.2 Ανεπίσημες Γλώσσες Προγραμματισμού.....	30
2.2 Τεχνολογίες Υλοποίησης της παρούσας εργασίας	31

2.2.1 MATLAB v 7.7.0 (R2008b).....	31
2.2.2 Simulink v 7.2.....	32
2.2.3 Simulink Report Generator 3.8.....	33
2.2.4 Simulink Verification and Validation 2.7.....	33
2.2.5 Real Time Workshop v 7.2.....	34
2.2.6 Control System Toolbox.....	34
2.2.7 Fixed -Point Toolbox.....	34
2.2.8 Simulink Fixed-Point.....	34
2.2.9 Target Language Compiler.....	35
2.2.10 Stateflow.....	35
2.2.11 Γραφικά αντικείμενα (Graphical objects)	37
2.2.12 Μη γραφικά αντικείμενα (Non Graphical objects)	43
2.2.13 Stateflow Coder 7.5	44
2.2.14 Simulink Design Verifier 1.6.....	44
2.2.15 BrickOS(Lejos).....	44
2.2.16 BricxCC v.3.3 (Bricx Command Center).....	45
2.2.17 Cygwin v 1.5.25-15.....	45
2.2.18 GCC cross compiler.....	45
2.2.19 Embedded Coder Robot NXT v 3.17.....	46
2.2.20 Real Time Workshop Embedded Coder.....	46
2.2.21 nxtOSEK v 2.09	46
2.2.22 NeXTTool	47
2.2.23 GNU ARM toolchain v 4.0.2.....	47
2.2.24 MINDSTORMS NXT Driver v1.02	47
2.2.25 IVT BlueSoleid EDR Edition v 2.1	47
2.2.26 NXT GamePad v 1.04	48
2.2.27 LEGO Digital Designer	48

2.3 Εγκατάσταση Λογισμικού.....	48
ΚΕΦΑΛΑΙΟ 3 : ΒΙΒΛΙΟΘΗΚΗ EMBEDDED CODER ROBOT NXT.....	53
3.1 Light Sensor blocks	53
3.2 Touch Sensor blocks	54
3.2 Sound Sensor blocks	54
3.3 Revolution Sensor blocks	55
3.4 Servo Motor blocks.....	55
3.5 Battery Voltage blocks	56
3.6 System Clock blocks	57
3.7 Sound Volume Tone blocks.....	57
3.8 Sound Wav blocks	58
3.9 Ultrasonic Sensor blocks.....	58
3.10 Bluetooth Rx blocks	59
3.11 Bluetooth Tx Blocks	59
3.12 NXT GamePad Data Logger block	60
3.13 NXT GamePad ADC Data Logger block	61
3.14 Acceleration Sensor blocks.....	62
3.15 Gyro Sensor Blocks.....	62
3.16 Exported Function-Calls Scheduler block	63
3.17 USB Rx blocks	64
3.18 USB Tx blocks	65
3.19 ENTER button blocks	65
3.20 RUN button blocks	66
3.21 OSEK Resource block	66
3.22 XY Map Graph block	67
ΚΕΦΑΛΑΙΟ 4 : Ο ΕΛΕΓΚΤΗΣ PID (PID CONTROLLER)	68
4.1 Ο όρος “P” στον PID (The "P" in "PID": Proportion(al)).....	69
4.1.1 Τα πραγματικά επίπεδα ισχύος του κινητήρα από το P	71
4.1.2 Ο ψευδοκώδικας για τον P Ελεγκτή	72
4.1.3 Περίληψη του P Ελεγκτή.....	73
4.2 Προσθέτοντας το “I” στον Ελεγκτή: Ο PI Ελεγκτής.....	74
4.2.1 Ο Ψευδοκώδικας για τον ελεγκτή PI.....	75
4.3 Προσθέτοντας το "D" στον Ελεγκτή: Ο ολοκληρωμένος PID ελεγκτής	75

4.3.1 Ο ψευδοκώδικας για τον PID ελεγκτή	76
ΚΕΦΑΛΑΙΟ 5 : ΤΟ ΜΟΝΤΕΛΟ NXTWAY_GS	78
5.1 Σχέδιο Μοντελοποίησης	80
5.1.1 Η σχεδιαστική Μοντελοποίηση(MBD).....	80
5.1.2 V-Διαδικασία.....	81
5.1.3 Πλεονεκτήματα της σχεδιαστικής μοντελοποίησης (MBD)	82
5.2 Το Σύστημα NXTway-GS	83
5.2.1 Δομή	83
5.2.2 Αισθητήρες/Ενεργοποιητές	83
5.3 Μοντελοποίηση του NXTway-GS	84
5.3.1 Μοντέλο διτροχού εκκρεμούς.....	84
5.3.2 Εξισώσεις κίνησης του διτροχού εκκρεμούς.....	86
5.3.3 Καταστατικές εξισώσεις διτροχού εκκρεμούς.....	89
5.4 Σχεδιασμός Ελεγκτή NXTway-GS.....	92
5.4.1 Σύστημα ελέγχου	92
5.4.2 Σχεδιασμός Ελεγκτή.....	93
5.5 Μοντέλο NXTway-GS	95
5.5.1 Το Μοντέλο συνοπτικά	95
5.5.1.1 Περιβάλλον	96
5.5.1.2 Αναφορά γεννήτριας.....	96
5.5.1.3 Ελεγκτής.....	98
5.5.1.4 NXTway-GS.....	99
5.5.1.5 Παρουσιαστής (“viewer”)	99
5.5.2 Αρχεία Παραμέτρων	100
5.6 Μοντέλο Εγκατάστασης (Plant Model)	101

5.6.1 Το Μοντέλο Συνοπτικά.....	101
5.6.2 Ενεργοποιητής	101
5.6.3 Εγκατάσταση(Plant)	103
5.6.4 Αισθητήρας(Sensor)	103
5.7 Μοντέλο Ελεγκτή	105
5.7.1 Συνοπτική περιγραφή ελέγχου του προγράμματος.....	105
5.7.1.1 Διάγραμμα καταστάσεων μηχανής	105
5.7.1.2 Διαμόρφωση εργασιών	106
5.7.1.3 Τύπος Δεδομένων	106
5.7.2 Σύνοψη Μοντέλου	107
5.7.2.1 Διεπαφή συσκευής (Device Interface).....	108
5.7.2.2 Χρονοπρογραμματισμός εργασιών. (Scheduler -Tasks).....	108
5.7.2.3 Προτεραιότητα (Priority	109
5.7.2.4 Δεδομένα διαμοιρασμού (Shared Data)	109
5.8 Αρχικοποίηση Εργασίας : task_init.....	110
5.8.1 4ms Task : task_ts1	110
5.8.1.1 Block διακριτής Παραγωγού και διακριτού Ολοκληρώματος	112
5.8.1.2 Υπολογισμός αναφοράς [(Calculation) -Cal Reference subsystem]	112
5.8.1.3 Υπολογισμός Κατάστασης(Cal x1)	113
5.8.1.4 Υπολογισμός PWM (Cal PWM)	113
5.8.1.5 Φόρτωση Δεδομένων.....	114
5.8.2 20ms task : task_ts2.....	115
5.8.3 100 ms Task : task_ts3	116
5.8.4 Παράμετροι συντονισμού	116
5.9 Εντοπισμός Μάυρης Γραμμής.....	117
5.10 Προσομοίωση	121

5.10.1 Διαδικασία εκτέλεσης της προσομοίωσης.....	121
5.10.2 Αποτελέσματα Προσομοίωσης.....	122
5.10.2.1 Στάσιμη Εξισορρόπηση	122
5.10.2.2 Κίνηση προς τα εμπρός και προς τα πίσω	122
5.10.3 Τρισδιάστατος Παρουσιαστής (3D viewer)	123
5.11 Παραγωγή Κώδικα και Εφαρμογή	124
5.11.1 Υλικό και Λογισμικό (Hardware & Software)	124
5.11.2 Τρόπος Παραγωγής και Φόρτωσης Κώδικα.....	124
5.11.3 Πειραματικά αποτελέσματα	126
5.11.3.1 Στάσιμη Εξισορρόπηση	126
5.11.3.2 Κίνηση προς τα εμπρός και προς τα πίσω	126
5.12 Μοντέλο Ελεγκτή [Αριθμός Σταθερής Υποδιαστολής (Fixed-Point Arithmetic)]	127
5.12.1 Ο Αριθμός Σταθερής Υποδιαστολής(Fixed point Number)	127
5.12.2 Μετατροπή Κινητής Υποδιαστολής σε Σταθερή.	128
5.12.2.1 Τύπος Δεδομένων Σταθερής Υποδιαστολής.....	128
5.12.2.2 Μοντέλο Σταθερής Υποδιαστολής (nxtway_gs_controller_fixpt.mdl)	130
5.12.3 Αποτελέσματα Προσομοίωσης.....	131
5.12.4 Παραγωγή Κώδικα και Πειραματικά Αποτελέσματα.....	132
ΠΑΡΑΡΤΗΜΑ Α : ΣΥΓΧΡΟΝΗ ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ	133
A.1 Σταθερότητα.....	133
A.2 Κατάσταση Ελέγχου Ανατροφοδότησης.....	133
A.3 Σέρβο- Έλεγχος (Servo Control).....	135
ΠΑΡΑΡΤΗΜΑ Β : ΧΩΡΟΣ ΕΙΚΟΝΙΚΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ.....	137
B.1 Σύστημα Συντεταγμένων.....	137
B.2 Δημιουργία ενός Αρχείου Χάρτη (Map File)	138
B.3 Υπολογισμός Απόστασης και Εντοπισμός Πρόσκρουσης σε τοίχο	139
ΠΑΡΑΡΤΗΜΑ Γ. ΠΑΡΑΓΟΜΕΝΟΣ ΚΩΔΙΚΑΣ	141
ΠΑΡΑΡΤΗΜΑ Δ. ΟΔΗΓΙΕΣ ΚΑΤΑΣΚΕΥΗΣ ΤΟΥ NXTWAY-GS.....	144

ΠΑΡΑΡΤΗΜΑ Ε. ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΗ ARM7	171
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	172

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας εργασίας ήταν η κατασκευή αυτοϊσορροπούμενου δίτροχου ρομπότ Lego Mindstorms NXT, καθώς και ο έλεγχός του με χρήση της πλατφόρμας MATLAB/SIMULINK για την ισορροπία του, την αποφυγή εμποδίων, τον εντοπισμό και την κίνησή του πάνω σε μαύρη γραμμή. Επιπλέον υπάρχει δυνατότητα λειτουργίας του ρομπότ σε δύο καταστάσεις, την αυτόνομη οδήγηση και τον απομακρυσμένο έλεγχο μέσω Bluetooth με χρήση Gamepad. Τέλος πραγματοποιήθηκε η προσομοίωση του συστήματος και η εξαγωγή των πειραματικών αποτελεσμάτων.

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή για το NXT και τα μέρη που το αποτελούν καθώς και στις συσκευές που χρησιμοποιήθηκαν.

Στο δεύτερο κεφάλαιο γίνεται μια αναφορά στις διάφορες γλώσσες προγραμματισμού που χρησιμοποιούνται για το NXT και στην συνέχεια περιγράφεται το λογισμικό που θα χρησιμοποιηθεί για την παρούσα εργασία.

Στο τρίτο κεφάλαιο περιγράφεται η βιβλιοθήκη Embedded Coder Robot NXT και στο τέταρτο κεφάλαιο ο αλγόριθμος PID για την εύρεση γραμμής.

Τέλος στο πέμπτο κεφάλαιο περιγράφεται το μοντέλο NXTway_GS, ο έλεγχος και η κατασκευή του δίτροχου αυτοϊσορροπούμενου ρομπότ με την χρήση LEGO MINDSTORMS NXT/MATLAB-SIMULINK. Περισσότερες λεπτομέρειες σχετικά με τα υποσυστήματα των εφαρμογών παρουσιάζονται στα παραρτήματα.

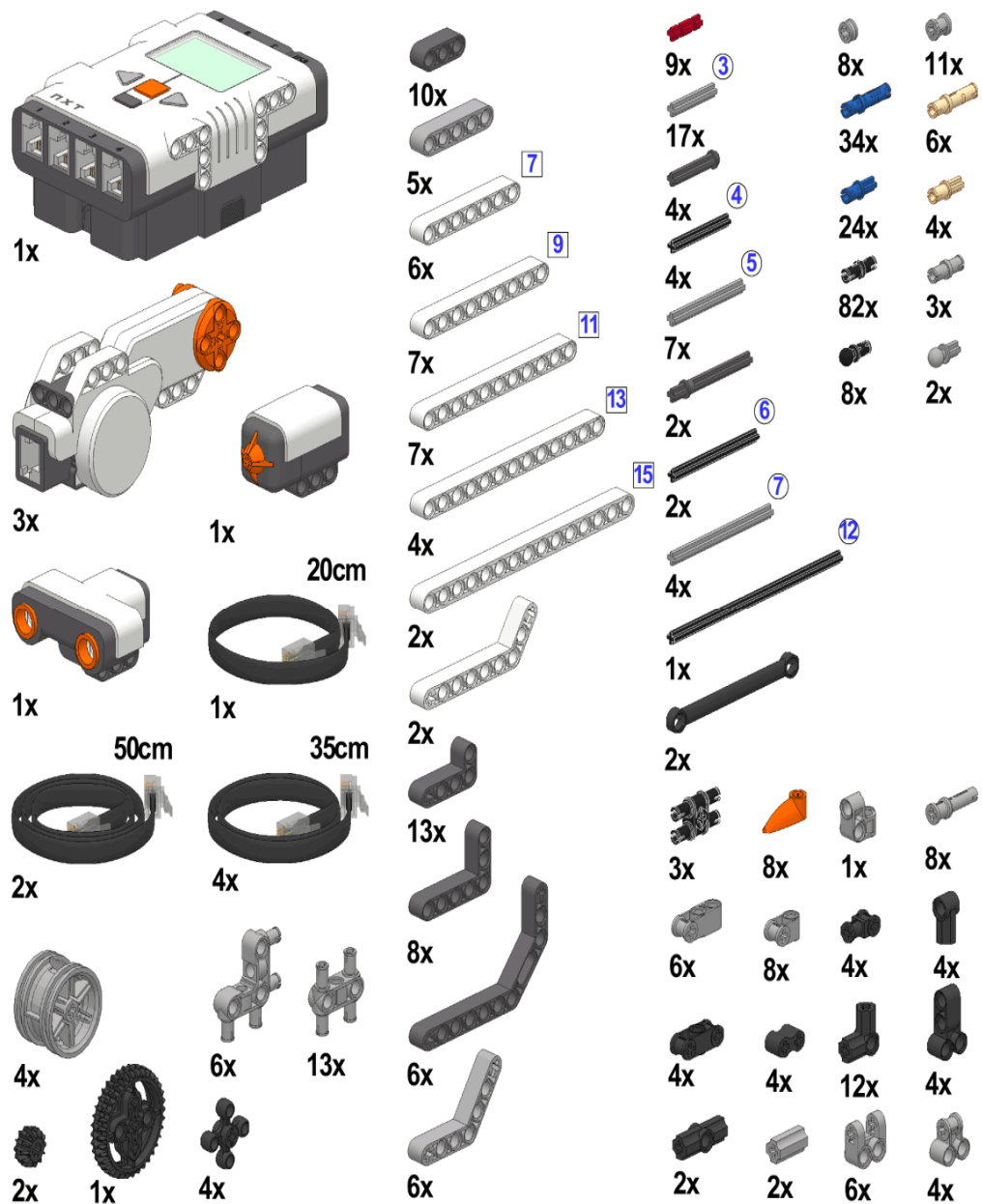
ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ



Η πρώτη επίσημη έκδοση του Lego Mindstorms κυκλοφόρησε στο εμπόριο το 1998 ως Lego Mindstorms Invention Robotic Kit (RIS) από την εταιρία Lego. Η επόμενη έκδοση κυκλοφόρησε το 2006 ως Lego Mindstorms NXT και η νεότερη έκδοση που κυκλοφόρησε τον Αύγουστο του 2009, είναι γνωστή ως Lego Mindstorms NXT 2.0. Το Lego Mindstorms NXT παρέχει τη δυνατότητα κατασκευής και προγραμματισμού διάφορων ρομπότ Lego, χρησιμοποιώντας τα πασίγνωστα “τουβλάκια” Lego. Το σετ Lego Mindstorms NXT αποτελείται από 519 κομμάτια και περιλαμβάνει εκτός από τα κομμάτια Lego Technic, άξονες, γρανάζια, κινητήρες, διάφορους αισθητήρες και το κεντρικό ‘τουβλάκι’ NXT.



Περιεχόμενα του Lego Mindstorms NXT & Education Resource set



1.1 Το NXT



Εικόνα 1.1 Το NXT.

Το NXT είναι ο εγκέφαλος του MINDSTORMS ρομπότ και περιέχει τον μικροελεγκτή 32-bit ARM7 σχεδιασμένο από την Arm Limited. Το NXT είναι ένα έξυπνο ελεγχόμενο κομπιούτερ Lego ‘τουβλάκι’ που επιτρέπει στο MINDSTORMS ρομπότ να ζωντανεύει και να εκτελεί αυτόνομα διάφορες εργασίες.

Τα βασικά χαρακτηριστικά του ολοκληρωμένου κυκλώματος βρίσκονται στο παράρτημα Ε. Επιπλέον το τουβλάκι αυτό περιλαμβάνει εικόνα.

Θύρες εξόδου :

Το NXT έχει τρεις θύρες εξόδου (Outputs) στις οποίες συνδέονται οι ελέγξιμες από το NXT συσκευές που είναι οι κινητήρες και οι λαμπτήρες. Θύρα Α, Β και C.

Θύρες εισόδου :

Το NXT έχει 4 θύρες εισόδου για την τοποθέτηση αισθητήρων. Με τον τρόπο αυτό το NXT αλληλεπιδρά με το εξωτερικό περιβάλλον. Θύρα 1, 2, 3 και 4.

Θύρα USB :

Θύρα USB για την επικοινωνία με τον ηλεκτρονικό υπολογιστή. Δίνει τη δυνατότητα να κατεβάσετε προγράμματα στο NXT (ή να ανεβάστε προγράμματα από το ρομπότ στον υπολογιστή). Μπορείτε επίσης να χρησιμοποιήσετε την ασύρματη σύνδεση Bluetooth για αποστολή και λήψη.

Ηχείο :

Δημιουργήστε ένα πρόγραμμα από πραγματικούς ήχους ώστε να τους ακούτε κατά την εκτέλεση του προγράμματος.

Περιβάλλον διεπαφής με τον χρήστη (User Interface) :

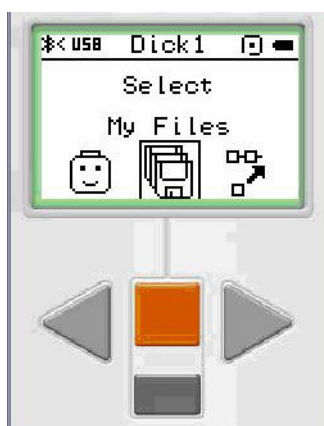
Τα πλήκτρα του NXT είναι τέσσερα[4]:

Το πορτοκαλί πλήκτρο : Έχει τέσσερις λειτουργίες : το [On-off] για ενεργοποίηση και απενεργοποίηση του NXT, το [Enter] για πρόσβαση σε φακέλους του μενού και επιλογή εφαρμογών, το [Run] για την εκτέλεση εφαρμογών που έχουν επιλεγθεί.

Βέλη (χρώματος) ανοιχτού γκρι : Χρησιμοποιούνται για μετακίνηση αριστερά και δεξιά στο μενού εφαρμογών του NXT.

Πλήκτρο (χρώματος) σκούρο γκρι : Έχει 2 λειτουργίες : το [stop] όπου σταματάει μια εφαρμογή που εκτελείται και το [Go back] όπου πηγαίνει πίσω, στην προηγούμενη κατάσταση.

Οθόνη υγρών κρυστάλλων NXT (LCD Display) :



Οθόνη υγρών κρυστάλλων 100×64 pixel (LCD Display). Στην οθόνη αυτή μπορούν να εμφανιστούν εικονίδια διάφορων λειτουργιών του καθώς επίσης και διάφορες ενδείξεις όπως η στάθμη της μπαταρίας, η ένταση του ήχου, φάκελοι με αποθηκευμένα προγράμματα ή ήχους, η ύπαρξη επικοινωνίας με τον ηλεκτρονικό υπολογιστή μέσω USB ή με Bluetooth, η κατάσταση των αισθητήρων και των κινητήρων (δηλαδή αν λειτουργούν ή όχι) και η κατάσταση του προγράμματος (αν εκτελείται το πρόγραμμα ή όχι).

Τεχνικές προδιαγραφές :

- 32 bit ARM7 μικροελεγκτή.
- 256 Kbytes FLASH, 64 Kbytes RAM.
- Bluetooth ασύρματης επικοινωνίας (Bluetooth Class II V2.0 compliant).
- USB θύρα πλήρης ταχύτητας (12 Mbit/sec).
- 4 θύρες εισόδου, 6-ενσώματα καλώδια ψηφιακής πλατφόρμας (Μία θύρα περιλαμβάνει IEC 61158 Τύπου 4/EN 50 170 συμβατή θύρα επέκτασης για μελλοντική χρήση.
- 3 θύρες εξόδου, 6-ενσώματα καλώδια ψηφιακής πλατφόρμας.
- LCD οθόνη γραφικών 100×64 pixel.
- Ηχείο - Ποιότητας ήχου 8 kHz. Κανάλι ήχου με 8-bit ανάλυση και 2-16 ρυθμός δειγματοληψίας kHz.
- Πηγή ενέργειας: 6 μπαταρίες τύπου AA.

1.2 Αισθητήρες (Sensors)

1.2.1 Αισθητήρας Αφής (Touch Sensor)



Ο αισθητήρας αφής ανιχνεύει την πίεση, π.χ. όταν το κουμπί έχει πιεστεί ή όχι. Προγραμματίζεται να εκτελεί εντολές βάσει τριών ενεργειών (on push, on loose, on bumped). Συνδέεται με καλώδιο στο NXT.



Όταν ο αισθητήρας αφής είναι πατημένος δίνει ως έξοδο στην οθόνη ένα [1].

Όταν ο αισθητήρας δεν είναι πατημένος δίνει ως έξοδο στην οθόνη μηδέν [0].

Προτάσεις για χρήση:

Μπορείτε να χρησιμοποιήσετε τον αισθητήρα αφής για να κάνετε το ρομπότ να σηκώσει αντικείμενα : ένας ρομποτικός βραχίονας συνδεδεμένος με έναν αισθητήρα αφής επιτρέπει στο ρομπότ να ξέρει αν υπάρχει ή δεν υπάρχει κάτι στο χέρι του για να το αρπάξει, αλλιώς μπορείτε να χρησιμοποιήσετε τον αισθητήρα αφής για να κάνετε ρομπότ να εκτελεί μία εντολή. Για παράδειγμα πατώντας τον αισθητήρα αφής μπορείτε να κάνετε το ρομπότ να κινείται, να μιλάει, να κλίνει μια πόρτα, ή να ανοίγει την τηλεόραση.

1.2.2 Αισθητήρας Ήχου (Sound Sensor)



Ο αισθητήρας ήχου κάνει το ρομπότ να ακούει!

Ο αισθητήρας ήχου είναι ικανός να μετρήσει τα επίπεδα θορύβου σε dB και dBA. Ακόμα αναγνωρίζει συγκεκριμένους ήχους και διαφορετικούς τόνους. Συνδέεται με καλώδιο στο NXT.

Το ντεσιμπέλ (dB) είναι μια λογαριθμική μονάδα μέτρησης που εκφράζει το μέγεθος της φυσικής ποσότητας (συνήθως ισχύς ή ένταση), σε σχέση με κάποιο συγκεκριμένο ή σιωπηρή επίπεδο αναφοράς.

dBA : Ανίχνευση προσαρμοσμένων ντεσιμπέλ. Η ευαισθησία του αισθητήρα είναι προσαρμοσμένη στην ευαισθησία του ανθρώπινου αυτιού, δηλαδή είναι οι ήχοι που τα ανθρώπινα αυτιά είναι σε θέση να ακούσουν.

dB : Ανίχνευση (μη προσαρμοσμένων) ντεσιμπέλ, όλοι οι ήχοι μετριοούνται με την ίδια ευαισθησία. Συνεπώς μπορεί να είναι ήχοι πολύ δυνατοί ή πολύ χαμηλοί για να τους ακούσει το ανθρώπινο αυτί.

Ο αισθητήρας ήχου μπορεί να μετρήσει τα επίπεδα ακουστικής πίεσης έως 90 dB σε σχέση με το επίπεδο μιας μηχανής. Οι στάθμες πίεσης του ήχου είναι εξαιρετικά περίπλοκες στο να διαβαστούν, έτσι ο αισθητήρας ήχου του mindstorms NXT τις εμφανίζει σε ποσοστό (%). Όσο πιο χαμηλό είναι το ποσοστό τόσο πιο χαμηλή είναι η ένταση του ήχου. Για παράδειγμα από πειραματικές μετρήσεις (το εύρος τιμών έχει υπολογιστεί από απόσταση ενός μέτρου από την πηγή του ήχου έως τον αισθητήρα ήχου) :

- 4 – 5 % σε έναν ήσυχο χώρο
- 5 – 10 % όταν κάποιος μιλάει από κάποια απόσταση
- 10 – 30 % σε φυσιολογική συνομιλία κοντά στον αισθητήρα ή με μουσική που παίζεται σε κανονικό επίπεδο
- 30 – 100 % όταν κάποιος φωνάζει, ή η μουσική παίζεται σε υψηλή ένταση

Προτάσεις για χρήση :

Μπορείτε να προσθέσετε τον αισθητήρα ήχου στον NXT μοντέλο και να το προγραμματίσετε έτσι ώστε να αλλάζει η συμπεριφορά του ρομπότ όταν ενεργοποιείται ο αισθητήρας ήχου.

1.2.3 Αισθητήρας Φωτός (Light Sensor)



Ο αισθητήρας φωτός είναι ένας από τους δύο αισθητήρες που δίνουν όραση στο ρομπότ. (Ο άλλος είναι ο αισθητήρας υπερήχων).

Ο αισθητήρας φωτός δίνει τη δυνατότητα να γίνει διάκριση μεταξύ φωτός και σκοταδιού.

Μπορεί να διαβάσει τις διαφορετικά επίπεδα τιμών της έντασης του φωτός σε ένα χώρο υπό σταθερές συνθήκες φωτισμού (ανίχνευση φωτεινότητας περιβάλλοντος) και αντιλαμβάνεται τα χρώματα χρησιμοποιώντας την κλίμακα του γκρι (μέτρηση της έντασης του φωτός σε χρωματιστές επιφάνειες).



This is what your eyes see



This is what your robot will see, using the light sensor.

Συνδέεται με καλώδιο στο NXT.

Προτάσεις για χρήση:

Μπορείτε να χρησιμοποιήσετε τον αισθητήρα φωτός για να φτιάξετε έναν «ρομπότ - συναγερμό για διαρρήκτη»: όταν ένας εισβολέας μπει στο δωμάτιο και ανάψει τα φώτα, τότε το ρομπότ μπορεί να αντιδράσει για να προστατέψει την περιουσία σας. Επίσης μπορείτε με τον αισθητήρα φωτός να φτιάξετε ένα ρομπότ που να ακολουθεί γραμμές ή ένα ρομπότ που να ταξινομεί τα πράγματα ανάλογα με το χρώμα τους.

1.2.4 Αισθητήρας Υπερήχων (Sonar Sensor)



Ο αισθητήρας υπερήχων είναι ένας από τους δύο αισθητήρες που δίνουν όραση στο ρομπότ. [Ο Αισθητήρας φωτός είναι ο άλλος]. Ο αισθητήρας υπερήχων είναι ικανός να ανιχνεύσει αντικείμενα, να υπολογίσει την απόσταση αυτών καθώς και να ανιχνεύσει κίνηση. Επίσης αναγνωρίζει συγκεκριμένους ήχους (<20KHz) και διαφορετικούς τόνους. Συνδέεται με καλώδιο στο NXT, το οποίο υπάρχει στο βασικό πακέτο 709797.

Ο αισθητήρας υπερήχων μέτρα απόσταση σε εκατοστά και σε ίντσες. Είναι σε θέση να μετρήσει αποστάσεις από 0 έως 255 εκατοστά με ακρίβεια + / - 3 εκατοστά.

Ο αισθητήρας υπέρηχου χρησιμοποιεί την ίδια επιστημονική αρχή, όπως οι νυχτερίδες : μετράει την απόσταση υπολογίζοντας τον χρόνο που χρειάζεται ένα ηχητικό κύμα για να χτυπήσει σε ένα αντικείμενο και να επιστρέψει - ακριβώς όπως η ηχώ.

Μεγάλου μεγέθους αντικείμενα με σκληρές επιφάνειες επιστρέφουν καλύτερες αναγνώσεις. Αντικείμενα κατασκευασμένα από μαλακό ύφασμα, ή από κυρτές επιφάνειες [όπως μια μπάλα], ή πολύ λεπτά ή μικρά μπορεί να είναι δύσκολο για την ανίχνευσή τους από τον αισθητήρα.

* Σημείωση : Δύο ή περισσότεροι αισθητήρες υπέρηχων που λειτουργούν στον ίδιο χώρο μπορεί να διακόψουν ο ένας την ανάγνωση του άλλου.

1.2.5 Αισθητήρας Χρώματος (Color Sensor)



Ο Αισθητήρας χρώματος είναι ικανός να αναγνωρίζει το χρώμα το οποίο στοχεύει ή να ανιχνεύει γραμμές σε μία επιφάνεια.

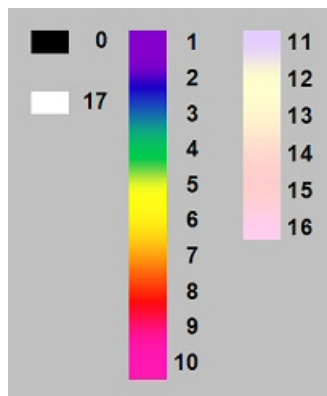
Λειτουργεί με την χρήση ενός λευκού LED (δίοδος εκπομπής φωτός) για να φωτίζει τον στόχο και τις αναλύσεις των στοιχείων του χρώματος του φωτός που αντανακλάται από την επιφάνεια του στόχου στον δέκτη και υπολογίζει χρώμα - αριθμό που επιστρέφει στο πρόγραμμα NXT.

Η ανίχνευση επεκτείνεται σε ένα φάσμα άνω των 15 χρωμάτων. Ο αισθητήρας χρώματος είναι ρυθμισμένος για τα καθιερωμένα χρώματα Lego.

Όταν ο αισθητήρας στοχεύει με την μπροστά όψη του σε διαφορετικές χρωματιστές επιφάνειες παρατηρείται ότι τα δεδομένα θα αλλάζουν από το ένα έως το δεκαεπτά [1-17]. (Το μηδέν [0], το οποίο αντιπροσωπεύει το μαύρο, ή σε περίπτωση που δεν εντοπιστεί επιφάνεια θα εμφανίσει.)

Τοποθέτηση αισθητήρα σε απόσταση περίπου 6 χιλιοστά από την χρωματιστή επιφάνεια.

Ο αισθητήρας χρώματος μπορεί να λειτουργεί με έναν από τους δύο τρόπους, χρώμα – αριθμός ή κόκκινο-πράσινο-μπλέ πρότυπο. Με το χρώμα-αριθμός πρότυπο επιστρέφει αριθμητικές τιμές ανάλογα με το χρώμα που ανιχνεύει, από το ένα έως το δεκαεπτά [1-17]. Τα χρώματα αυτά παρουσιάζονται στην γκάμα χρωμάτων του διαγράμματος από κάτω.



Το διάγραμμα χρώμα-αριθμός δείχνει την σχέση μεταξύ χρώματος-στόχου και του χρώματος-αριθμού που επιστρέφει ο αισθητήρας χρώματος.

Στο κόκκινο – πράσινο - μπλε πρότυπο έχετε την δυνατότητα πρόσβασης στο εύρος τιμών που χρησιμοποιούνται από τον αισθητήρα για να εκτιμήσετε το χρώμα. Αυτό δίνει την δυνατότητα στο να πειραματιστείτε με ένα δικό σας σύστημα ανίχνευσης χρωμάτων.

Ο αισθητήρας χρησιμοποιεί τη δική του πυγή φωτός για να φωτίσει την χρωματιστή επιφάνεια έτσι ώστε να τον καθιστά απρόσβλητο στον περιβάλλοντα φωτισμό, για αυτό το λόγο πρέπει να προσέξετε έτσι ώστε να αποφευχθεί ο φωτισμός της επιφάνειας από πολύ έντονο φωτισμό.

Οι μικροί φακοί στην μπροστά όψη του αισθητήρα χρώματος είναι ευαίσθητοι σαν τους φακούς της κάμερας για αυτό χρειάζεται προσοχή και να αποφευχθούν οι γρατζουνιές.

Sensor Register Layout

The color firmware supports the standard Lego Company's sensor memory model.

Address	Type	Contents
00 – 07H	chars	Sensor version number
08 – 0FH	chars	Manufacturer
10 – 17H	chars	Sensor type
18 – 3DH	bytes	Not used
3E, 3FH	chars	Reserved
40H	byte	Not used
41H	byte	Command
42H	byte	Color number
43H	byte	Red reading
44H	byte	Green reading
45H	byte	Blue reading

*Σημείωση :

- Ο αισθητήρας χρώματος πρέπει να ρυθμιστεί ώστε να ταιριάζει με τη συχνότητα της ηλεκτρικής ενέργειας για το ηλεκτρικό δίκτυο της χώρας σας.
- Ο αισθητήρας χρώματος λειτουργεί καλύτερα όταν βρίσκεται κατ'ευθείαν πάνω από την επιφάνεια του στόχου και δεν είναι υπό γωνία προς την επιφάνεια του στόχου.

1.2.6 Γυροσκοπικός Αισθητήρας (Gyroscopic Sensor)

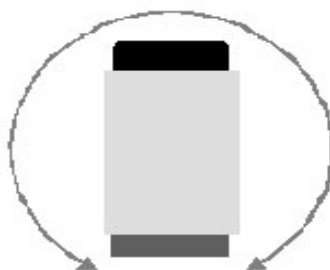


Ο γυροσκοπικός αισθητήρας NXT δίνει τη δυνατότητα στο ρομπότ να ανιχνεύει την εναλλαγή περιστροφικής κίνησης.

Περιλαμβάνει έναν αξονικό γυροσκοπικό αισθητήρα που ανιχνεύει την περιστροφική κίνηση και επιστρέφει μια τιμή που αντιπροσωπεύει τον αριθμό των μοιρών ανά δευτερόλεπτο περιστροφής.

Ο αισθητήρας περιστροφής μπορεί να υπολογίσει περιστροφή $\pm 360^\circ/\text{sec}$. Ο ρυθμός περιστροφής μπορεί να διαβαστεί περίπου ως και 300 φορές ανά δευτερόλεπτο.

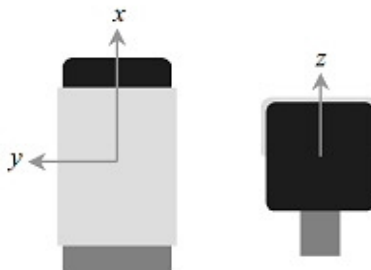
Ο αισθητήρας περιστροφής χρησιμοποιείται για την μέτρηση του μεγέθους των γωνιών σε πολύ σύντομο χρονικό διάστημα (περίπου 100 φορές ανά δευτερόλεπτο). Με αυτό τον τρόπο μπορεί να καθορίζει αν το ρομπότ πέφτει (χάνει την ισορροπία του) από ποια κατεύθυνση και πόσο γρήγορα πέφτει.



1.2.7 Αισθητήρας Επιτάχυνσης (Acceleration Sensor)



Ο αισθητήρας επιτάχυνσης περιέχει έναν accelerometer (μετρητή επιτάχυνσης) τριών αξόνων, που υπολογίζει τις δυνάμεις επιτάχυνσης σε τρεις κατευθύνσεις ως αποτέλεσμα του ποσοστού αλλαγής της ταχύτητας, στους τρεις άξονες x , y και z όπως φαίνεται στο σχήμα.



Η επιτάχυνση μετράται στο εύρος από $-2g$ έως $2g$ με κλίμακα της τάξεως των διακοσίων [200] μετρήσεων ανά g .

Ο αισθητήρας επιτάχυνσης μπορεί επίσης να υπολογίσει και την κλίση κατά μήκος του κάθε άξονα, ώστε να μπορεί να πει αν το ρομπότ κλίνει αριστερά ή δεξιά και προς τα πάνω ή προς τα κάτω. Ο αισθητήρας είναι εργοστασιακά ρυθμισμένος. Ο αισθητήρας επιτάχυνσης συνδέεται σε μία θύρα εισόδου (αισθητήρα) NXT χρησιμοποιώντας ένα πρότυπο καλώδιο NXT και χρησιμοποιεί το πρωτόκολλο I2C ψηφιακών επικοινωνιών. Η μέτρηση της επιτάχυνσης για κάθε άξονα ανανεώνεται περίπου 100 φορές ανά δευτερόλεπτο.

Σε πειραματική δοκιμή χρησιμοποιώντας το περιβάλλον υπέρυθρων (-Ultrasonic cm-) παρατηρείται ότι καθώς δίνεται κλίση στον αισθητήρα πάνω και κάτω, οι αναγνώσεις αλλάζουν από 1-50 για προς τα πάνω κλίση και από 255-205 καθώς κλίνει προς τα κάτω (το [0] θα εμφανιστεί ως ???). Αυτό παρατηρείται για τον x άξονα ο οποίος αναφέρετε συχνά και ως άξονας μεταφοράς (pitch axis). Επίσης υπάρχουν αναγνώσεις και από τον y άξονα που ονομάζεται άξονας περιστροφής και από τον z άξονα.

Τα δεδομένα των y και z αξόνων δεν παρατηρούνται χρησιμοποιώντας το περιβάλλον υπέρυθρων (ultrasonic interface). Για ανάκτηση αναγνώσεων και από τους τρεις άξονες πρέπει είτε να χρησιμοποιήσετε το HiTechnic Acceleration Sensor block είτε κάποιο προγραμματιστικό περιβάλλον συμβατό με το NXT.

Αν χρησιμοποιείται το HiTechnic Acceleration Sensor block θα παίρνετε τρεις ενδεικτικές τιμές για τους x , y και z άξονες σε κλίμακα δεδομένων της τάξεως των 200 κομματιών ανά g .

Χρησιμοποιώντας κάποιο άλλο προγραμματιστικό περιβάλλον, η ανάκτηση των τιμών των x , y και z αξόνων γίνεται με ανάγνωση των τοποθεσιών που παρουσιάζονται στον πίνακα.

Address	Type	Contents
42H	byte	X axis upper 8 bits
43H	byte	Y axis upper 8 bits
44H	byte	Z axis upper 8 bits
45H	byte	X axis lower 2 bits
46H	byte	Y axis lower 2 bits
47H	byte	Z axis lower 2 bits

1.2.8 Μαγνητικός Αισθητήρας (Compass Sensor)



Ο αισθητήρας πυξίδα καθορίζει την κατεύθυνση όπου στοχεύει το ρομπότ. Η κατεύθυνση αυτή αναφέρεται ως μαγνητικό επίπεδο.

Ο αισθητήρας μπορεί να προγραμματιστεί χρησιμοποιώντας το Lego MINDSTORMS NXT Software Compass Block ή με το Ultrasonic Block. Από πειραματικές δοκιμές παρατηρείται ότι καθώς περιστρέφετε τον αισθητήρα οι αναγνώσεις αλλάζουν από 1-179 (με το 0 να εμφανίζεται ως ?????).

Ο μαγνητικός αισθητήρας (compass sensor) είναι μια ψηφιακή πυξίδα που μετράει το μαγνητικό πεδίο της γης και υπολογίζει το επίπεδο στο οποίο βρίσκεται με ακρίβεια 1° και επιστρέφει αριθμητικά δεδομένα από 0-359. Η μαγνητική πυξίδα ενημερώνει για το επίπεδο 100 φορές ανά δευτερόλεπτο.

Ο μαγνητικός αισθητήρας λειτουργεί σε δύο καταστάσεις, ανάγνωσης και ρύθμισης. Στην κατάσταση ανάγνωσης υπολογίζεται το επίπεδο στο οποίο βρίσκεται και επιστρέφει κάθε φορά στο πρόγραμμα NXT που εκτελείται μια εντολή για ανάγνωση. Στην κατάσταση ρύθμισης η πυξίδα μπορεί να ρυθμίζεται ώστε να αντισταθμίζει το μαγνητικό πεδίο από εξωτερικές ανωμαλίες όπως αυτές που περιβάλλουν τους κινητήρες και τις ηλεκτρικές μπαταρίες, διατηρώντας έτσι τη μέγιστη ακρίβεια. Κατά την ρύθμιση είναι άκρως αναγκαίο να τοποθετήσετε την πυξίδα σε απόσταση τουλάχιστο 15 εκατοστά (6

ίντσες) μακριά από τους κινητήρες και 10 εκατοστά (4 ίντσες) μακριά από το τουβλάκι NXT.

*Σημείωση : Η πυξίδα πρέπει πάντα να τοποθετείτε σταθερά και οριζόντια, ώστε η ανάκτηση των δεδομένων από τις αναγνώσεις του αισθητήρα να είναι αξιόπιστες.

Sensor Register Layout

Address	Type	Contents
00 – 07H	chars	Sensor version number
08 – 0FH	chars	Manufacturer
10 – 17H	chars	Sensor type
18 – 3DH	bytes	Not used
3E, 3FH	chars	Reserved
40H	byte	Not used
41H	byte	Mode control
42H	byte	Heading } two degree heading
43H	byte	Heading } one degree adder
44, 45H	word	Heading (low byte, high byte)
46 – 7FH	bytes	Not used

1.2.9 Αισθητήρας εντοπισμού Υπέρουθρων (Infrared Seeker Sensor)



Ο αισθητήρας Infrared Seeker είναι μια συσκευή ανίχνευσης υπέρυθρων φωτεινών πηγών (σημάτων) και καθορισμού της κατεύθυνσής τους. Αυτός ο αισθητήρας αρχικά σχεδιάστηκε για να εντοπίζει υπέρυθρα σήματα από πυγές όπως : οι υπέρυθρες μπάλες ποδοσφαίρου της Hitechnic, υπέρυθρα τηλεχειριστήρια και το φώς του ηλίου. Λόγω του ειδικά

σχεδιασμένου καμπυλωτού φακού του και των πέντε [5] εσωτερικών ανιχνευτών παραταγμένων σε απόσταση ανά 60°, έχει οπτική 240° πλάμα που τον κάνει τέλειο για διαγωνισμούς ποδοσφαίρου ρομπότ.

Ο αισθητήρας IR(Infrared) Seeker μπορεί να λειτουργεί σε δύο καταστάσεις :

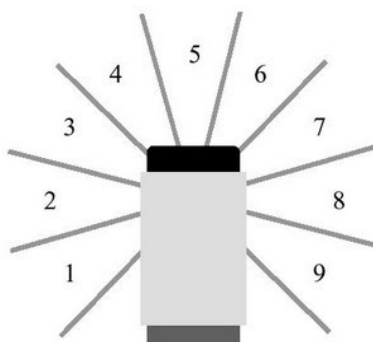
Εναλλασσόμενη κατάσταση (AC) : Ο αισθητήρας εντοπίζει εναλλασσόμενα υπέρυθρα σήματα όπως αυτά που εκπέμπουν οι υπέρυθρες μπάλες της Hitechnic και τα τηλεχειριστήρια. Στην εναλλασσόμενη κατάσταση ο αισθητήρας θα φιλτράρει έξω τα περισσότερα άλλα υπέρυθρα σήματα ώστε να μειώσει τις παρεμβολές, για παράδειγμα, από τα φώτα και το φως του ηλίου. Ο αισθητήρας είναι συντονισμένος σε εναλλασσόμενα κύματα των 1200 Hz.

Συνεχές κατάσταση (DC) : Ο αισθητήρας εντοπίζει συνεχή υπέρυθρα σήματα όπως οι παλαιότερου τύπου υπέρυθρες μπάλες και το φως του ηλίου.

Ο αισθητήρας IRSeeker V2 χρησιμοποιεί προηγμένες τεχνικές ψηφιακής επεξεργασίας σήματος για να φιλτράρονται τα σήματα που λαμβάνει και να επιλέγονται μόνο τα σήματα που απαιτούνται. Ένα κάλυμμα νέας σχεδίασης με σταθερής ακτίνας καμπυλωτό καπάκι αυξάνει τις επιδόσεις κατεύθυνσής του με την ελαχιστοποίηση των στρεβλώσεων από φωτεινά σήματα που εισέρχονται στον αισθητήρα.



Οι έξοδοι των τιμών κατεύθυνσης του αισθητήρα IR Seeker παρουσιάζονται στο σχήμα, όπου το [1] σημαίνει ότι ο υπέρυθρος στόχος είναι αριστερά και πίσω, το [5] όταν ο στόχος είναι άμεσα μπροστά, το [9] εφόσον ο στόχος είναι προς τα δεξιά και πίσω. Η τιμή [0] επιστρέφεται εάν δεν ανιχνεύεται το σήμα (η τιμή [0] εμφανίζεται ως ?????).



Στο NXT-G IR Seeker block λογισμικό θα έχει έξι τιμές εξόδου, για τις τιμές κατεύθυνσης και πέντε τιμές για τα αντικείμενα που εντοπίζει ο αισθητήρας στις κατευθύνσεις [1], [3], [5], [7] και [9].

Σε περίπτωση χρήσης διαφορετικού προγραμματιστικού περιβάλλοντος, η τιμή της κατεύθυνσης καθορίζεται από την τοποθεσία 0×42 και οι πέντε αναγνώσεις του αισθητήρα από τοποθεσίες 0×43 έως 0×47 όπως φαίνεται στο σχήμα.

IRSeeker V2 Sensor Register Layout

Address	Type	Contents
00 – 07H	chars	Sensor version number
08 – 0FH	chars	Manufacturer
10 – 17H	chars	Sensor type
41H	byte	Mode
42H	byte	DC Direction (0 - 9)
43H	byte	Sensor 1 DC signal strength
44H	byte	Sensor 2 DC signal strength
45H	byte	Sensor 3 DC signal strength
46H	byte	Sensor 4 DC signal strength
47H	byte	Sensor 5 DC signal strength
48H	byte	Sensor DC mean
49H	byte	AC Direction Data (0 - 9)
4AH	byte	Sensor 1 AC signal strength
4BH	byte	Sensor 2 AC signal strength
4CH	byte	Sensor 3 AC signal strength
4DH	byte	Sensor 4 AC signal strength
4EH	byte	Sensor 5 AC signal strength

Οι τιμές των εξαρτημάτων του αισθητήρα μπορούν να χρησιμοποιήσουν στον υπολογισμό της απόστασης του υπέρυθρου στόχου εάν έχει προηγηθεί η ρύθμιση του επιπέδου ισχύος της πυγής. Σε περίπτωση που εντοπιστεί η ο στόχος σε μία από τις συμπληρωματικές ζώνες ([2], [4], [6] ή [8]) πρέπει να προσθέσετε κατά μέσο όρο τις μετρήσεις από τα δύο

παρακείμενα εξαρτήματα του αισθητήρα. Για παράδειγμα αν η αναφορά κατεύθυνσης είναι [6], τότε τα εξαρτήματα του αισθητήρα [5] και [7] διαμοιράζονται το σήμα και οι δύο τιμές μπορούν να προστεθούν ώστε να καθορίσουν το συνολικό σήμα που λαμβάνεται.

*Σημείωση :

- Η χρήση υπέρυθρου τηλεχειριστηρίου ως πηγή σήματος : Τα περισσότερα τηλεχειριστήρια θα ανιχνευθούν από το IRSeekerV2 σε λειτουργία AC αλλά μερικά μπορεί να μην είναι ανιχνεύσιμα, αναλόγως το σήμα που παράγεται. Η IRSeekerV2 είναι συντονισμένοι για την ανίχνευση σημάτων κύματος 1200 Hz.
- Η διεύθυνση I2C του αισθητήρα V2 IRSeeker είναι 0×10.

1.3 Συσκευές εξόδου (Actuators)

1.3.1 Σερβοκινητήρες (Servomotors)



Κάθε κινητήρας έχει έναν ενσωματωμένο αισθητήρα περιστροφής. Αυτό δίνει τη δυνατότητα ελέγχου των κινήσεων του ρομπότ με ακρίβεια.

Ο αισθητήρας γωνίας μέτρα τις περιστροφές του κινητήρα σε μοίρες ή πλήρης περιστροφές [ακρίβεια + / - μίας μοίρας]. Μία περιστροφή ισούται με 360 μοίρες, συνεπώς αν ρυθμίσετε τον κινητήρα για στροφή 180 μοιρών, ο άξονας εξόδου του θα κάνει μισή στροφή.

Ο ενσωματωμένος αισθητήρας περιστροφής σε κάθε κινητήρα επιτρέπει την ρύθμιση διαφορετικών ταχυτήτων στους κινητήρες (θέτοντας διαφορετικές παραμέτρους ισχύος στο λογισμικό). Πολλοί κινητήρες μπορούν να συγχρονιστούν στην ίδια ταχύτητα.

Συνδέεται με το πρότυπο καλώδιο στο NXT.

1.3.2 Λαμπτήρες (Lamps)

Οι λαμπτήρες είναι η πιο απλή συσκευή εξόδου και χρησιμοποιούνται στα Lego MINDSTORMS ως ενδεικτικά λειτουργιών. Είναι απλοί λαμπτήρες οι οποίοι έχουν δύο καταστάσεις, σβηστοί ή αναμμένοι με έξοδο [0] ή [1] αντίστοιχα.



1.4 Πρόσθετες συσκευές που χρησιμοποιήθηκαν

1.4.1 USB Bluetooth Dongle



Το USB Bluetooth Dongle χρησιμεύει για την ασύρματη επικοινωνία και μεταφορά δεδομένων μεταξύ του Η/Υ και ψηφιακών συσκευών που διαθέτουν ενεργοποιημένο Bluetooth.

1.4.2 Xbox360 Analog PC Gamepad



Το Xbox360 Gamepad της Microsoft είναι ένα αναλογικό χειριστήριο παιχνιδιών που θα χρησιμοποιηθεί για τον έλεγχο της κατεύθυνσης του ρομπότ.

ΚΕΦΑΛΑΙΟ 2 : ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ

Σε αυτό το κεφάλαιο παρουσιάζονται περιληπτικά αρχικά διάφορες γλώσσες προγραμματισμού που χρησιμοποιούνται για το NXT (από αυτές χρησιμοποιήθηκαν nxtOSEK, MATLAB & SIMULINK) και στην συνέχεια περιγράφεται το λογισμικό που θα χρησιμοποιηθεί για την παρούσα εργασία και για τον προγραμματισμό του NXT μέσω του ηλεκτρονικού υπολογιστή.

2.1 Γλώσσες Προγραμματισμού για το NXT

2.1.1 Επίσημη γλώσσα προγραμματισμού

- **NXT-G code v1.0 & v2.0**

Αυτό το λογισμικό είναι επαρκές για το βασικό προγραμματισμό, όπως η οδήγηση των μηχανών, η ενσωμάτωση των αισθητήρων, υπολογισμούς και εκμάθηση απλουστευμένων δομών προγραμματισμού μέσω διαγραμμάτων ροής. Στην συνέχεια παρουσιάζονται μερικά πλεονεκτήματα και μειονεκτήματα από την χρήση της έκδοσης 1.0 του NXT-G λογισμικού :

Πλεονεκτήματα :

- + Εύκολη εγκατάσταση στα λειτουργικά συστήματα Windows XP, Windows Vista, Windows 7 καθώς υποστηρίζει και Mac OS X.
- + Μπορεί να μεταφέρει δεδομένα μέσω Bluetooth ή μέσω σύνδεσης USB.
- + Ευκολία χρήσης μέσω drag and drop, γραφικό περιβάλλον.
- + Η γραφική αναπαράσταση περιλαμβάνει στοιχεία για τα καλώδια που παρουσιάζονται στο διάγραμμα ροή από block σε block.

Μειονεκτήματα :

- Τα προγράμματα NXT μπορεί να είναι πολύ μεγαλύτερα από τα ίδια προγράμματα σε ανεπίσημη γλώσσα προγραμματισμού τρίτων.
- Τα προγράμματα αυτά χρειάζονται περισσότερο χρόνο φόρτωσης από τα ίδια προγράμματα σε ανεπίσημη γλώσσα προγραμματισμού τρίτων.
- Κατά την δημιουργία μεγάλων προγραμμάτων το πρόγραμμα μπορεί να μπλοκάρει και να χαθούν μη αποθηκευμένα δεδομένα.
- Το λογισμικό NXT-G καθυστερεί ακόμη και σε ισχυρούς υπολογιστές.
- Η NXT-G v.1 δεν υποστηρίζει αριθμητικά δεδομένα κινητής υποδιαστολής και η NXT-G v.2 υποστηρίζει αριθμητικά δεδομένα κινητής υποδιαστολής μέχρι ένα δεκαδικό ψηφίο.

2.1.2 Ανεπίσημες Γλώσσες Προγραμματισμού

- **NXC (Not eXactly C) & NBC (NeXT Byte Code) (και BricxCC)**

Το **BricxCC** ή Κέντρο Εντολών Bricx (Bricx Command Center), αναπτύσσεται συνεχώς και είναι το περιβάλλον προγραμματισμού που χρησιμοποιείται για να γράφει/συντάσσει/εκδίδει NBC και NXC προγράμματα για το NXT. Επίσης χρησιμοποιώντας BricxCC, μπορείτε να αλλάξετε σε διαφορετικά firmwares στο NXT.

Η **Next Byte Codes** (NBC) είναι μια απλή γλώσσα ανοιχτός-πηγής (open source) με μια σύνταξη συμβολικής γλώσσας (Assembly) που μπορεί να χρησιμοποιηθεί για να προγραμματίσει το τούβλο NXT.

Η **Not eXactly C** (NXC) είναι μια υψηλού επιπέδου open source γλώσσα προγραμματισμού, που μοιάζει με την C και χτίζεται πάνω στον μεταγλωττιστή NBC. Μπορεί επίσης να χρησιμοποιηθεί για να προγραμματίσει το τούβλο NXT. Είναι μια ευρύτατα χρησιμοποιημένη γλώσσα προγραμματισμού τρίτων για το NXT. Στην πραγματικότητα, 10% των ανθρώπων στις περισσότερες ψηφοφορίες ψηφίζουν ότι αυτό ήταν η χρησιμοποιημένη γλώσσα προγραμματισμού τους για το NXT.

Πλεονεκτήματα :

- + Το BricxCC μπορεί να μεταφέρει δεδομένα μέσω Bluetooth ή μέσω του USB καλωδίου.
- + Το BricxCC παρέχει ένα προηγμένο, textual-based προγραμματιστικό περιβάλλον.
- + 32 bit Float Types.
- + Ο πυγαιός κώδικας και τα μεταγλωττισμένα αρχεία έχουν μικρό μέγεθος, πολλές φορές πολύ μικρότερο από του NXT-G.
- + Τα προγράμματα φορτώνονται γρήγορα, πολύ πιο γρήγορα από ότι στο NXT.

Μειονεκτήματα :

- Όχι τόσο κατάλληλα για νέους προγραμματιστές και παιδιά λόγω της πολυπλοκότητας τους.

- **RobotC**

Η **RobotC** είναι μια άλλη γλώσσα προγραμματισμού βασισμένη στην C. Απαιτείται κατάλληλα προσαρμοσμένο firmware προκειμένου να εκτελεστεί.

- **LeJos NXJ**

Η **leJOS NXJ** είναι μια υψηλού επιπέδου open source γλώσσα προγραμματισμού βασισμένη στην JAVA και χρησιμοποιεί κατάλληλο firmware που αναπτύσσεται από την ομάδα leJOS.

- **nxtOSEK**

Η **nxtOSEK** μπορεί να χρησιμοποιηθεί για να γράφει C/C++ (γλώσσα προγραμματισμού), είναι αναγκαίο το κατάλληλο firmware.

SELECTED PROGRAMMING TOOLS FOR LEGO MINDSTORMS NXT

	LabVIEW Toolkit	Matlab/ Simulink	NXC	RobotC	leJOS NXJ	nxtOSEK
Programming	Graph	Graph	Text	Text	Text	Text
Syntax	NI blocks	Simulink blocks, C	Like C	C	Java	C/C++
Firmware	Standard	Repl.	Standard	Repl.	Repl.	Repl.
License	LabView	Matlab/Simulink	Freeware	Yes	Open source	Open source
Events	No	Yes	No	Yes	Java events	Yes (OSEK RTOS)
Multithreading	Yes	Yes	Yes	Yes	Yes	Yes
Bluetooth communication: Brick to PC	Yes	Yes	Yes	Yes	Yes	Yes
BluetoothBrick to Brick	Yes	Yes	Yes	Yes	Yes	Not yet
Floating point	No	No	No	Yes	Yes	Yes

- **MATLAB & SIMULINK**

Η **MATLAB** είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού για αριθμητικούς υπολογισμούς, απόκτησης δεδομένων και ανάλυση. Μπορεί να χρησιμοποιηθεί για να ελέγξει τα ρομπότ LEGO NXT μέσω Bluetooth ή μέσω μιας σύνδεσης USB παραδείγματος χάριν χρησιμοποιώντας την εργαλειοθήκη RWTH.

Το **SIMULINK** είναι ένα περιβάλλον βασισμένο στην MATLAB για την μοντελοποίηση και προσομοίωση δυναμικών συστημάτων. Χρησιμοποιώντας το SIMULINK, ο χρήστης μπορεί να σχεδιάσει αλγορίθμους ελέγχου, να παραγάγει αυτόματα κώδικα C για τους αλγορίθμους και να φορτώσει τον μεταγλωττισμένο κώδικα στο LEGO NXT.

Στην συνέχεια γίνεται η αναλυτική περιγραφή του απαραίτητου λογισμικού της Mathworks και ακολουθούν οι πρόσθετες εφαρμογές οι οποίες είναι απαραίτητες για την μεταγλώττιση των κατασκευασμένων μοντέλων Simulink σε γλώσσα C/C++ και στην ακόλουθη αποστολή τους μέσω Bluetooth στο NXT.

2.2 Τεχνολογίες Υλοποίησης της παρούσας εργασίας

2.2.1 MATLAB v 7.7.0 (R2008b)

Η **MATLAB** είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου. Ενοποιεί τον υπολογισμό, την απεικόνιση και τον προγραμματισμό σε ένα περιβάλλον, εύκολο στη χρήση, όπου τα προβλήματα και οι λύσεις, εκφράζονται σε μία μαθηματική σημειολογία. Με τη χρήση της MATLAB είναι δυνατή η επίλυση τεχνικών προγραμματιστικών

προβλημάτων γρηγορότερα από άλλες παραδοσιακές γλώσσες προγραμματισμού όπως η C, η C++ και η Fortran. Τυπικές χρήσεις της MATLAB περιλαμβάνουν :

- Μαθηματικά και υπολογισμούς.
- Ανάπτυξη αλγορίθμων.
- Μοντελοποίηση, προσομοίωση και προτυποποίηση.
- Ανάλυση δεδομένων, εξερεύνηση και οπτική απεικόνιση.
- Επιστημονικά και μηχανολογικά γραφικά.
- Ανάπτυξη εφαρμογών, συμπεριλαμβανομένης της κατασκευής γραφικού περιβάλλοντος διεπαφής.

Η MATLAB έχει αναπτυχθεί κατά τη διάρκεια αριστών ετών μέσω ανατροφοδότησης πολλών χρηστών. Στα πανεπιστήμια, αποτελεί το βασικό εκπαιδευτικό εργαλείο τόσο για εισαγωγικά όσο και για προχωρημένα μαθήματα μαθηματικών και μηχανικής. Στη βιομηχανία, η MATLAB αποτελεί ένα εργαλείο για έρευνα υψηλής παραγωγικότητας, ανάπτυξης και ανάλυσης. Πρόσθετες εργαλειακές επεκτείνουν το περιβάλλον της MATLAB επιτρέποντας την επίλυση συγκεκριμένων κατηγοριών προβλημάτων σε πολλούς τομείς εφαρμογών. Αυτές είναι πολύ σημαντικές στους περισσότερους χρήστες της MATLAB και επιτρέπουν την εκμάθηση και εφαρμογή εξειδικευμένης τεχνολογίας. Περιοχές εφαρμογών στις οποίες είναι διαθέσιμες οι εργαλειακές αυτές περιλαμβάνουν : την επεξεργασία σημάτων, τα συστήματα ελέγχου, τα νευρωνικά δίκτυα, την ασαφής λογική, την προσομοίωση και πολλές άλλες.

2.2.2 Simulink v 7.2

Το **Simulink** είναι ένα πακέτο λογισμικού που επιτρέπει την γραφική μοντελοποίηση, την προσομοίωση και την ανάλυση δυναμικών συστημάτων. Υποστηρίζει γραμμικά και μη γραμμικά συστήματα, μοντελοποιημένα σε συνεχή ή διακριτό χρόνο, ή ακόμη και υβριδικά συστήματα (εν μέρει μοντελοποιημένα σε συνεχή και εν μέρει σε διακριτό χρόνο). Υποστηρίζονται ακόμη συστήματα με τμηματικά διαφορετικούς χρόνους δειγματοληψίας. Χρησιμοποιείται για την εξέταση της συμπεριφοράς μιας μεγάλης έκτασης δυναμικών συστημάτων πραγματικού κόσμου, συμπεριλαμβανομένων ηλεκτρικών κυκλωμάτων, συστημάτων πέδησης, και πολλών άλλων ηλεκτρικών, μηχανικών και θερμοδυναμικών συστημάτων. Για τη μοντελοποίηση, το Simulink παρέχει ένα γραφικό περιβάλλον διεπαφής (GUI) που επιτρέπει την κατασκευή μοντέλων ως δομικών διαγραμμάτων, χρησιμοποιώντας λειτουργίες click-and-drag του ποντικιού. Το Simulink περιλαμβάνει ένα πλήθος βιβλιοθηκών δομικών στοιχείων (blocks), τα βασικότερα από τα οποία είναι οι πηγές (sources), τα στοιχεία «απορρόφησης» (sinks), τα συνεχή γραμμικά στοιχεία και τα μη γραμμικά.

Επίσης είναι δυνατή η τροποποίηση και η δημιουργία νέων δομικών στοιχείων από το χρήστη με τη χρήση των συναρτήσεων συστήματος, S-functions (Systemfunctions). Οι S-functions χρησιμοποιούνται για τη δημιουργία νέων δομικών διαγραμμάτων Simulink. Ο χρήστης μπορεί να χρησιμοποιήσει τις S-functions σε μια ποικιλία εφαρμογών που περιλαμβάνουν :

- ✓ Προσθήκη νέων δομικών διαγραμμάτων γενικού σκοπού (general purpose blocks) στο Simulink.

- ✓ Προσθήκη δομικών διαγραμμάτων που αντιπροσωπεύουν οδηγούς συσκευών υλικού (hardware device drivers).
- ✓ Περιγραφή ενός συστήματος σαν ένα σύνολο μαθηματικών εξισώσεων.

Οι S-functions, επιτρέπουν στον χρήστη να προσθέσει τους δικούς του αλγόριθμους στα μοντέλα του Simulink. Οι αλγόριθμοι αυτοί μπορούν να γραφτούν στη MATLAB ή σε γλώσσες προγραμματισμού όπως η C, η C++, η Fortran και η Ada. Ακολουθώντας ένα σύνολο απλών κανόνων, ο χρήστης μπορεί να εφαρμόσει τους αλγόριθμους αυτούς σε μια συνάρτηση συστήματος (S-function). Μετά την κατασκευή της S-function και την ονομασία του δομικού διαγράμματος της (Sfunction block), που μετά είναι διαθέσιμο στην υπό-βιβλιοθήκη Nonlinear Block sublibrary του Simulink, ο χρήστης μπορεί να την τροποποιεί χρησιμοποιώντας μάσκα (masking). Ένα πλεονέκτημα της χρήσης S-functions είναι η δυνατότητα κατασκευής ενός δομικού διαγράμματος γενικού σκοπού το οποίο στη συνέχεια μπορούμε να το χρησιμοποιήσουμε πολλές φορές μέσα σε ένα μοντέλο, διαφοροποιώντας τις παραμέτρους του κάθε φορά και βλέποντας τα αποτελέσματα. Μετά τη δημιουργία του μοντέλου, είναι δυνατή η προσομοίωση του, χρησιμοποιώντας μια από τις διάφορες μεθόδους ολοκλήρωσης που παρέχει το Simulink. Χρησιμοποιώντας παλμογράφους (scopes) και άλλα μπλοκ απεικόνισης, είναι δυνατή η παρακολούθηση των αποτελεσμάτων της προσομοίωσης καθώς αυτή εξελίσσεται. Επιπλέον, είναι δυνατή η εξαγωγή αποτελεσμάτων της προσομοίωσης στο χώρο εργασίας της MATLAB για περαιτέρω επεξεργασία.

2.2.3 Simulink Report Generator 3.8

Το **Simulink Report Generator** παράγει αυτόματα έγγραφα εφαρμογών τα οποία εκτελούνται στην MATLAB, όπως η ανάλυση και η οπτικοποίηση των δεδομένων και η ανάπτυξη αλγορίθμων. Επιτρέπει την εκτέλεση κώδικα MATLAB και κρατάει τα γραφικά και τα δεδομένα που έχουν παραχθεί. Παρέχει την δυνατότητα χρήσης των προκατασκευασμένων προτύπων ή την δημιουργία προτύπου από τον χρήστη ενσωματώνοντας το δικό του ύφος (styles) και πρότυπα.

2.2.4 Simulink Verification and Validation 2.7

Το λογισμικό **Simulink Verification and Validation** (Επαλήθευσης και επικύρωσης) εκθέτει σχεδιαστικά σφάλματα, ανεπαρκή απαιτήσεις, ελλείψεις ελέγχους, και περιττές σχεδιαστικές κατασκευές στα πρώιμα στάδια της διαδικασίας ανάπτυξης. Παρέχει την δυνατότητα εντοπισμού απαιτήσεων των εγγράφων για τα μοντέλα σχεδιασμού, τον έλεγχο των εξαρτημάτων και του παραγόμενου κώδικα. Επίσης επιτρέπει την επιβεβαίωση των σχεδίων και των δοκιμών μέσω της κάλυψης του μοντέλου και του ελέγχου των προτύπων μοντελοποίησης.

Το Simulink Verification and Validation, προβλέπει τον έλεγχο μοντέλων για τα βιομηχανικά πρότυπα DO-178B και IEC 61508.

2.2.5 Real Time Workshop v 7.2

Χρησιμοποιώντας την εργαλειοθήκη πραγματικού χρόνου (Real-Time Workshop) είναι δυνατή η προσομοίωση αλλά και ο έλεγχος συστημάτων σε πραγματικό χρόνο.

Το **Real-Time Workshop** παράγει κώδικα C απευθείας από τα μοντέλα του Simulink και κατασκευάζει αυτόματα προγράμματα τα οποία μπορούν να τρέξουν, σε πραγματικό χρόνο, σε μία ποικιλία από περιβάλλοντα, συμπεριλαμβανομένων συστημάτων πραγματικού χρόνου (real-time systems) και αυτοδύναμων προσομοιώσεων (stand-alone simulations).

Με το Real-Time Workshop, μπορούμε να τρέξουμε το μοντέλο Simulink σε έναν απομακρυσμένο επεξεργαστή (remote processor) σε πραγματικό χρόνο. Μπορούμε επίσης να τρέξουμε αυτοδύναμες προσομοιώσεις μεγάλης ταχύτητας (high-speed stand-alone simulations) σε έναν κεντρικό ή εξωτερικό υπολογιστή.

2.2.6 Control System Toolbox

Το **Control System Toolbox** λογισμικό παρέχει εργαλεία για τη συστηματική ανάλυση, το σχεδιασμό, και την προσαρμογή γραμμικών συστημάτων ελέγχου. Επιτρέπει τον καθορισμό ενός γραμμικού μοντέλου του συστήματος, τον σχεδιασμό του χρόνου επεξεργασίας του και την απόκριση συχνότητας ώστε να γίνεται ευκολότερη η κατανόηση της συμπεριφοράς του συστήματος. Επίσης καθιστά ικανό τον συντονισμό των δεδομένων του ελεγκτή με τη χρήση αυτοματοποιημένων και διαδραστικών τεχνικών, έτσι ώστε να επαληθεύονται οι αναμενόμενες απαιτήσεις επιδόσεων, όπως ο χρόνος ανόδου και τα όρια κέρδους/φάσης.

2.2.7 Fixed -Point Toolbox

Το **Fixed-Point Toolbox** προσφέρει σταθερού σημείου τύπου δεδομένα και αριθμητικών δεδομένων στην MATLAB. Η εργαλειοθήκη επιτρέπει τον σχεδιασμό αλγορίθμων σταθερού σημείου χρησιμοποιώντας MATLAB σύνταξη και εκτελείται καταρτίζοντας ταχύτατο C-κώδικα. Καθιστά ικανή την επαναχρησιμοποίηση αυτών των αλγορίθμων σε Simulink και μπορεί να περάσει το σταθερό σημείο δεδομένων από και προς τα Simulink μοντέλα, διευκολύνοντας στην “bit true” προσομοίωση, την εφαρμογή, και την ανάλυση. Η εργαλειοθήκη δίνει επίσης τη δυνατότητα να δημιουργούν ακολουθίες δοκιμών για λογισμικό και υλικό επαλήθευσης (software/hardware verification) σταθερού σημείου.

2.2.8 Simulink Fixed-Point

Το **Simulink Fixed Point** επιτρέπει την μηχανογράφηση των δυνατοτήτων του σταθερού σημείου στα προϊόντα της Simulink, επιτρέποντάς την χρήση αυτών των

προϊόντων στον σχεδιασμό των συστημάτων ελέγχου και επεξεργασίας σήματος, που υλοποιούνται μέσω σταθερών αριθμητικών στοιχείων.

2.2.9 Target Language Compiler

Για την παραγωγή κώδικα, το Real-Time Workshop υποστηρίζεται από τον **Target Language Compiler** (TLC). Ο Target Language Compiler μετασχηματίζει μία ενδιάμεση περιγραφή του μοντέλου, που έχει παραχθεί από το Real-Time Workshop, του διαγράμματος Simulink σε κώδικα σχετικό με τον στόχο. Αυτή η ενδιάμεση περιγραφή του μοντέλου αποθηκεύεται σε ένα αρχείο ASCII που ονομάζεται *model.rtw*. Ο Target Language Compiler επιτρέπει την τροποποίηση των περισσότερων κομματιών του παραγόμενου κώδικα. Ο μεταγλωττιστής διαβάζει το αρχείο *model.rtw* και εκτελεί ένα πρόγραμμα TLC που αποτελείται από ένα σύνολο αρχείων στόχων [target files (.tlc files)]. Αυτά είναι αρχεία ASCII γραμμένα για τον Target Language Compiler. Το πρόγραμμα TLC καθορίζει το πώς να μετασχηματιστεί το αρχείο *model.rtw* σε κώδικα.

Ένα πρόγραμμα TLC αποτελείται από :

- Το σημείο εκκίνησης (entry point) ή το κύριο αρχείο (main file).
- Ένα σύνολο από δομικά στοιχεία αρχείων στόχων (block target files). Αυτά καθορίζουν πως θα μεταφραστεί κάθε δομικό στοιχείο (block) του μοντέλου σε κώδικα σχετικό με τον στόχο.
- Μία βιβλιοθήκη συναρτήσεων (function library). Αυτή αποτελεί ένα σύνολο από συναρτήσεις βιβλιοθήκης που χρησιμοποιεί το πρόγραμμα TLC κατά την μετατροπή του αρχείου *model.rtw* σε κώδικα.

2.2.10 Stateflow Coder

Το **Stateflow** είναι ένα εργαλείο ανάπτυξης και γραφικού σχεδιασμού που χρησιμοποιείται σε συνδυασμό με το Simulink. Παρέχει καθαρή και σαφή περιγραφή της συμπεριφοράς σύνθετων συστημάτων χρησιμοποιώντας, στο ίδιο διάγραμμα Stateflow, την θεωρία των πεπερασμένων καταστάσεων μηχανών (finite state machine theory), την σημειολογία των διαγραμμάτων ροής (flow diagram notations) και τα διαγράμματα καταστάσεων - μεταβάσεων (state-transition diagrams).

Με την χρήση του Stateflow Coder, μπορούμε να παράγουμε κώδικα για εφαρμογές που κατασκευάζουμε σε άλλα περιβάλλοντα, όπως σε ένα ενσωματωμένο περιβάλλον (embedded environment). Μπορούμε ακόμα να συμπεριλάβουμε τον δικό μας, κατασκευασμένο κώδικα, ο οποίος χρησιμοποιείται από το Stateflow για την κατασκευή του στόχου.

Με την εργαλειοθήκη του Real-Time Workshop είναι δυνατή η λήψη κώδικα από το Simulink και το Stateflow και η εκτέλεση του σαν εφαρμογή σε ένα άλλο περιβάλλον, για τον έλεγχο μιας διαδικασίας. Επίσης, είναι εφικτή η παραγωγή κώδικα από τα διαγράμματα του Stateflow και η χρησιμοποίησή του με κάθε δυνατό τρόπο. Το Stateflow συνεργάζεται με το Real-Time Workshop με τους παρακάτω τρόπους :

- Το Stateflow επιτρέπει την δημιουργία και προσομοίωση των μοντέλων του Simulink που περιέχουν μηχανές καταστάσεων (state machines).
- Το Stateflow επιτρέπει την δημιουργία αυτοδύναμων εκτελέσιμων (standalone executables) που καθορίζονται από μηχανές καταστάσεων.
- Το Real-Time Workshop παράγει αυτοδύναμα εκτελέσιμα (stand-alone executables) από τα μοντέλα του Simulink, συμπεριλαμβάνοντας αυτές τις μηχανές καταστάσεων που καθορίζονται από το Stateflow.

Η χρήση του Stateflow επιτρέπει :

- Την παραστατική μοντελοποίηση και προσομοίωση σύνθετων συστημάτων που βασίζονται στην θεωρία της μηχανής πεπερασμένων καταστάσεων (finite state machine theory). Σύμφωνα με την θεωρία αυτή, η μετάβαση από μία κατάσταση (state) σε μία άλλη προκαθορισμένη κατάσταση, γίνεται εάν η συνθήκη που καθορίζει την μετάβαση είναι αληθής. Σαν παράδειγμα, η θεωρία της μηχανής πεπερασμένων καταστάσεων μπορεί να εφαρμοστεί στον αλγόριθμο PID, που θα περιγραφεί στην συνέχεια, για την αναπαράσταση της κίνησης του NXT ακολουθώντας μια μαύρη γραμμή. Η κίνηση του NXT έχει έναν αριθμό από λειτουργικές καταστάσεις : την κίνηση προς τα εμπρός (forwards), τη στροφή αριστερά και τη στροφή δεξιά.
- Την σχεδίαση και ανάπτυξη αιτιοκρατικών συστημάτων εποπτικής λογικής (deterministic, supervisory control systems). Τα συστήματα αυτά δημιουργούνται με την βοήθεια διαγραμμάτων. Ένα διάγραμμα Stateflow (Stateflow diagram) είναι μια γραφική αναπαράσταση ενός μηχανισμού πεπερασμένων καταστάσεων όπου οι καταστάσεις και οι μεταβάσεις σχηματίζουν τα βασικά δομικά στοιχεία του συστήματος. Στα διαγράμματα αυτά γίνεται ταυτόχρονη χρήση της σημειολογίας διαγραμμάτων ροής (flow diagram notation) και σημειολογίας μεταβάσεων καταστάσεων (state transition notation). Η σημειολογία των διαγραμμάτων ροής αποτελεί έναν αποτελεσματικό τρόπο αναπαράστασης κοινών δομών όπως οι επαναλήψεις for ή η δομές if-then-else.
- Την εύκολη τροποποίηση του συστήματος, την αξιολόγηση των αποτελεσμάτων και την επιβεβαίωση της συμπεριφοράς του σε οποιοδήποτε στάδιο της σχεδίασης.
- Την αυτόματη παραγωγή κώδικα C απευθείας από το μοντέλο, με χρήση του Stateflow Coder.
- Την μοντελοποίηση, προσομοίωση και ανάλυση του συστήματος με χρήση του Simulink.

- Την αυτόματη κατασκευή προγραμμάτων τα οποία μπορούν να τρέξουν, σε πραγματικό χρόνο (real time), σε μια μεγάλη ποικιλία από περιβάλλοντα, στην περίπτωση μας το για τον μικροελεγκτή του NXT, γίνεται με την χρήση του Real-Time Workshop (RTW) και του Real-Time Workshop Embedded Coder.








Η λειτουργία του Stateflow στηρίζεται στην κατασκευή διαγραμμάτων. Στις παρακάτω παραγράφους γίνεται μία σύντομη περιγραφή των πιο σημαντικών στοιχείων που θα χρησιμοποιηθούν για την κατασκευή των μοντέλων που θα εκτελεστούν για τον PID αλγόριθμο με τον οποίο το ρομπότ βρίσκει και ακολουθεί την μαύρη γραμμή.

Το Stateflow έχει τη δική του σημειολογία. Η σημειολογία αυτή καθορίζει ένα σύνολο αντικειμένων και τους κανόνες που ελέγχουν τις σχέσεις μεταξύ αυτών των αντικειμένων. Η σημειολογία του Stateflow αποτελείται από :

- Ένα σύνολο γραφικών αντικειμένων (Graphical objects).
- Ένα σύνολο μη γραφικών αντικειμένων (Non Graphical objects).
- Καθορισμένες σχέσεις μεταξύ αυτών των αντικειμένων.

2.2.11 Γραφικά αντικείμενα (Graphical objects)

Τα πιο σημαντικά γραφικά αντικείμενα του Stateflow είναι οι καταστάσεις (states), οι μεταβάσεις (transitions), οι προκαθορισμένες μεταβάσεις (Default transitions) και οι συνδετικοί σύνδεσμοι (connective junctions). Οι γραφικές τους αναπαραστάσεις στο περιβάλλον του Stateflow φαίνονται στην εικόνα 2.1.

Name	Notation	Toolbar Icon
State		
Transition		NA
Default Transition		
Connective Junction		

Εικόνα 2.1 : Γραφικά αντικείμενα του Stateflow

Οι **καταστάσεις (states)** περιγράφουν τις δυνατές καταστάσεις που μπορεί να βρεθεί το σύστημα. Στο Stateflow απεικονίζονται όπως στην εικόνα 2.2.



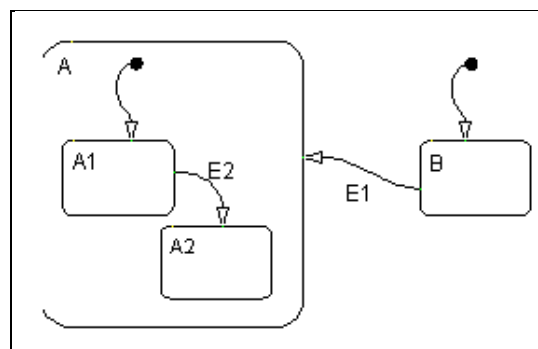
Εικόνα 2.2 : Γραφική απεικόνιση της κατάστασης στο stateflow

Οι καταστάσεις μπορούν να είναι **υπέρ-καταστάσεις (superstates)**, **υπό- καταστάσεις (substates)** ή απλά **καταστάσεις (states)**. Μία κατάσταση καλείται υπέρ-κατάσταση εάν περιέχει στο εσωτερικό της άλλες καταστάσεις, που καλούνται υπό-καταστάσεις. Μία κατάσταση καλείται υπό-κατάσταση εάν περιέχεται σε μία άλλη κατάσταση. Μία κατάσταση η οποία δεν είναι ούτε υπέρ-κατάσταση ούτε υπό-κατάσταση καλείται απλά κατάσταση.

Οι καταστάσεις διακρίνονται σε **ενεργές (active)** και **μη ενεργές (inactive)**. Μία κατάσταση είναι ενεργή όταν το διάγραμμα βρίσκεται στην συγκεκριμένη κατάσταση, ενώ είναι μη ενεργή όταν δεν βρίσκεται σε αυτήν. Η ενεργοποίηση ή όχι μίας κατάστασης αλλάζει δυναμικά και βασίζεται στα γεγονότα (events) και τις συνθήκες (conditions). Η ύπαρξη των γεγονότων οδηγεί στην εκτέλεση του διαγράμματος Stateflow κάνοντας τις καταστάσεις ενεργές ή μη ενεργές. Κατά τη διάρκεια εκτέλεσης ενός διαγράμματος, υπάρχει ένας συνδυασμός ενεργών και μη ενεργών καταστάσεων.

Κάθε κατάσταση (και διάγραμμα) αναλύεται στα στοιχεία που το αποτελούν και που υποδεικνύουν το είδος των υπό-καταστάσεων που μπορεί να περιέχει. Όλες οι υπό-καταστάσεις μίας υπέρ-κατάστασης πρέπει να είναι του ίδιου τύπου με αυτόν της υπέρ-κατάστασης. Μπορούμε να έχουμε δύο ειδών ανάλυση μίας κατάστασης : την παράλληλη (parallel ή AND) ή την αποκλειστική (exclusive ή OR).

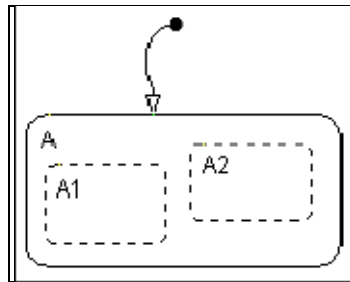
Η αποκλειστική ανάλυση (exclusive ή OR decomposition) μίας υπέρ-κατάστασης χρησιμοποιείται για να περιγράψει καταστάσεις συστημάτων οι οποίες είναι αμοιβαία αποκλεισμένες. Στην περίπτωση αυτή μόνο μία υπό-κατάσταση μπορεί να είναι ενεργή κάθε χρονική στιγμή. Στο παράδειγμα τις εικόνες 2.3 οποιαδήποτε από τις καταστάσεις A ή B μπορεί να είναι ενεργή. Εάν η A είναι ενεργή τότε οποιαδήποτε από τις καταστάσεις A1 ή A2 μπορεί να είναι ενεργή, αλλά όχι και οι δύο ταυτόχρονα.



Εικόνα 2.3 : Παράδειγμα ανάλυσης OR

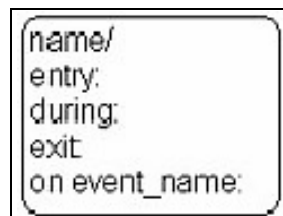
Η παράλληλη ανάλυση μίας υπέρ-κατάστασης χρησιμοποιείται για να περιγράψει καταστάσεις συστημάτων οι οποίες μπορούν να συμβούν ταυτόχρονα. Στο παράδειγμα της

εικόνας 2.4 όταν η κατάσταση A είναι ενεργή, οι καταστάσεις A1 και A2 είναι ενεργές ταυτόχρονα.



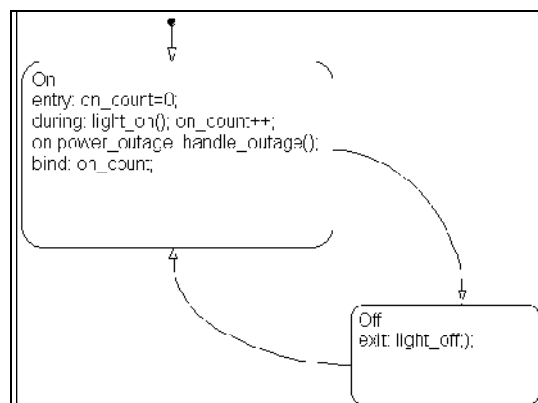
Εικόνα 2.4 : Παράλληλες ενεργές καταστάσεις

Κάθε κατάσταση περιγράφεται από μία **επιγραφή (label)** που εμφανίζεται σχήματος της εικόνα 2.5. Κάθε τμήμα της επιγραφής είναι προαιρετικό. Η επιγραφή έχει την παρακάτω γενική μορφή:



Εικόνα 2.5 : Τα μέρη της επιγραφής μιας κατάστασης

Ένα παράδειγμα φαίνεται παρακάτω :



Εικόνα 2.6 : Παράδειγμα επιγραφής μιας κατάστασης

Η επιγραφή μιας κατάστασης ξεκινάει με το **όνομα της κατάστασης (name)**. Στο παραπάνω παράδειγμα, τα ονόματα των καταστάσεων είναι On και Off.

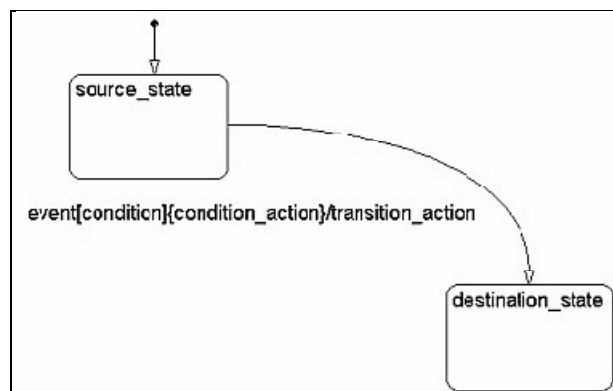
Ακολουθεί η δήλωση των **ενεργειών εισόδου (entry)**. Στο παράδειγμα, η κατάσταση On έχει σαν ενέργεια εισόδου on_count=0. Αυτό σημαίνει ότι η τιμή της μεταβλητής on_count θα γίνεται 0 κάθε φορά που η κατάσταση On θα γίνεται ενεργή.

Στη συνέχεια δηλώνουμε τις ενέργειες που θα συμβαίνουν όταν μία κατάσταση είναι ενεργή και συμβαίνει οποιοδήποτε γεγονός. Αυτές δηλώνονται μετά από το **during**. Στο παράδειγμα, για την κατάσταση On, οι ενέργειες `light_on()` και `on_count++` θα εκτελεστούν όταν η κατάσταση On είναι ενεργή και συμβεί οποιοδήποτε γεγονός.

Κατόπιν δηλώνονται οι **ενέργειες εξόδου (exit)**. Οι ενέργειες αυτές εκτελούνται όταν η κατάσταση απενεργοποιείται. Στο παράδειγμα, για την κατάσταση Off, η ενέργεια εξόδου είναι η `light_off()`.

Τέλος, ακολουθούν οι ενέργειες οι οποίες εκτελούνται αφού συντελεστεί ένα γεγονός (**on όνομα_γεγονότος**). Στο παράδειγμα, στην κατάσταση On δηλώνεται ότι `on power_outage:handle_outage()`. Εάν η κατάσταση On είναι ενεργή και συμβεί το γεγονός `power_outage`, τότε η ενέργεια `handle_outage` εκτελείται.

Οι **μεταβάσεις (transitions)** είναι καμπύλες γραμμές με βέλη που ενώνουν ένα γραφικό αντικείμενο με ένα άλλο. Στις περισσότερες περιπτώσεις, μία μετάβαση αναπαριστά το πέρασμα του συστήματος από μία κατάσταση σε μία άλλη. Μία μετάβαση ξεκινάει από την κατάσταση πηγή (source state) και καταλήγει στην κατάσταση προορισμού (destination state). Στην εικόνα 2.7 βλέπουμε ένα απλό παράδειγμα μετάβασης. Σε αυτό το παράδειγμα, έχουμε μία μετάβαση από την κατάσταση πηγή `source_state` στην κατάσταση προορισμού `destination_state`.



Εικόνα 2.7 : Απεικόνιση απλής μετάβασης

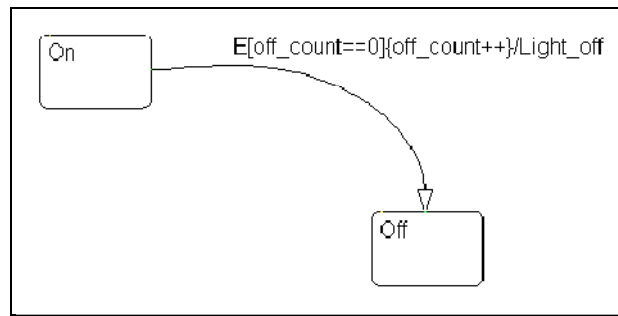
Κάθε μετάβαση περιγράφεται από μία επιγραφή, η οποία έχει την παρακάτω γενική μορφή:

`event[condition]{condition_action}/transition_action`

γεγονός[συνθήκη]{ενέργεια συνθήκης}/ενέργεια μετάβασης

Το κάθε τμήμα της επιγραφής είναι προαιρετικό.

Για να γίνει κατανοητή η έννοια αυτή θα δώσουμε ένα παράδειγμα (εικόνα 2.8).



Εικόνα 2.8 : Παράδειγμα επιγραφής μιας μετάβασης

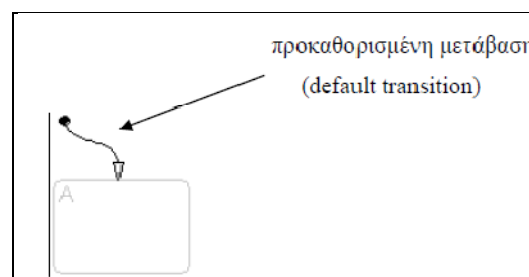
Πρώτα δηλώνεται το **γεγονός (event)**. Το καθορισμένο γεγονός είναι αυτό που προκαλεί την μετάβαση, εάν η συνθήκη, είναι αληθής. Ο καθορισμός του γεγονότος είναι προαιρετικός. Έλλειψη γεγονότος δηλώνει ότι η μετάβαση θα γίνεται με την εμφάνιση οποιουδήποτε γεγονότος. Πολλαπλά γεγονότα καθορίζονται με τη χρήση του λογικού τελεστή OR (|). Στο παραπάνω παράδειγμα, η εμφάνιση του γεγονότος E ενεργοποιεί την μετάβαση από το On στο Off εάν η συνθήκη [off_count==0] είναι αληθής.

Στη συνέχεια δηλώνεται η **συνθήκη (condition)**. Μία συνθήκη είναι μία Boolean έκφραση για να καθορίσουμε ότι η μετάβαση θα συμβεί δεδομένου ότι η καθορισμένη έκφραση είναι αληθής. Η συνθήκη αυτή γράφεται μέσα σε [].

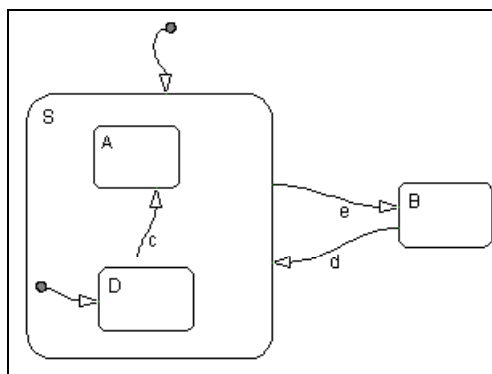
Κατόπιν, δηλώνονται οι **ενέργειες συνθήκης (condition_action)** μέσα σε { }. Οι ενέργειες αυτές εκτελούνται αμέσως μόλις η συνθήκη εκτιμηθεί ως αληθής και πριν καθοριστεί σαν έγκυρη η μετάβαση προς την κατάσταση προορισμού. Εάν δεν καθοριστεί συνθήκη, τότε υπονοείται μία συνθήκη, εκτιμάται ως αληθής και η ενέργεια συνθήκης εκτελείται. Στο παράδειγμα, εάν η συνθήκη [off_count==0] είναι αληθής, η ενέργεια συνθήκης, off_count++, εκτελείται αμέσως.

Τέλος, δηλώνονται οι **ενέργειες μετάβασης (transition action)**. Αυτές εκτελούνται αφού έχει καθοριστεί έγκυρη η μετάβαση προορισμού, και εάν η συνθήκη είναι αληθής. Στο παράδειγμα, εάν η συνθήκη [off_count==0] είναι αληθής και η κατάσταση προορισμού είναι έγκυρη, τότε εκτελείται η ενέργεια μετάβασης Light_off.

Οι **προκαθορισμένες μεταβάσεις (default transitions)** (εικόνα 2.9), χρησιμοποιούνται για να καθορίσουν σε ποια κατάσταση θα εισέλθει το σύστημα εξ' ορισμού. Είναι μία μετάβαση η οποία καταλήγει μόνο σε μία κατάσταση προορισμού. Επίσης, χρησιμοποιείται για να καθορίσει σε ποια κατάσταση OR θα πρέπει να εισέλθει το σύστημα όταν υπάρχουν δύο ή περισσότερες γειτονικές καταστάσεις OR στο ίδιο επίπεδο ιεραρχίας.



Εικόνα 2.9 : Σχηματική απεικόνιση προκαθορισμένης κατάστασης (default transition).



Εικόνα 2.10 : Παράδειγμα χρήσης εικονικών μεταβάσεων.

Στην εικόνα 2.10, δίνεται ένα παράδειγμα για τις χρήσεις των προκαθορισμένων καταστάσεων. Όταν το διάγραμμα ενεργοποιείται για πρώτη φορά, θα πρέπει να αποφασίσει ποια από τις καταστάσεις S ή B πρέπει να ενεργοποιηθεί, αφού είναι καταστάσεις OR. Επειδή έχουμε προκαθορισμένη μετάβαση στην S, αυτή ενεργοποιείται. Η κατάσταση S, που είναι τώρα ενεργή, έχει δύο υπό-καταστάσεις, την A και την D. Η κατάσταση που πρέπει να ενεργοποιηθεί εξ' ορισμού και πάλι δίνεται από την προκαθορισμένη μετάβαση και είναι η κατάσταση D.

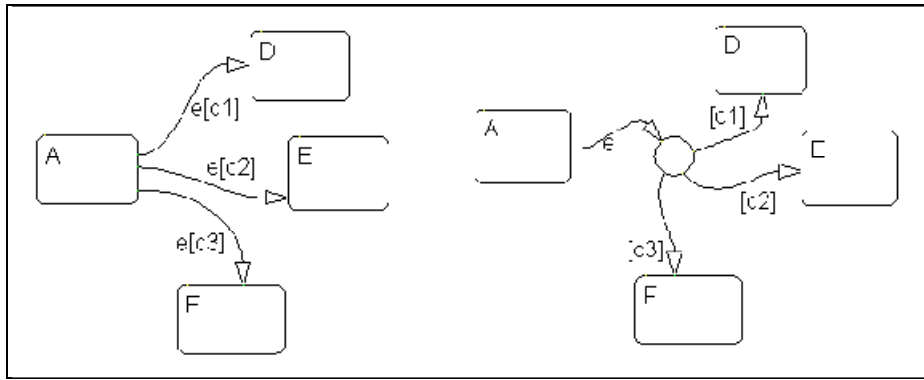
Οι **συνδετικοί σύνδεσμοι (connective junctions)** χρησιμοποιούνται για την αναπαράσταση κοινών δομών όπως οι επαναλήψεις for και οι δομές if-then-else, χωρίς τη χρήση καταστάσεων. Με την μείωση του αριθμού των καταστάσεων τα διαγράμματα Stateflow παράγουν αποδοτικότερο κώδικα που βοηθάει στην βελτιστοποίηση της χρήσης της μνήμης. Η απεικόνιση ενός συνδετικού συνδέσμου δίνεται στην εικόνα 2.11. Υπάρχουν οι ακόλουθοι συνδυασμοί :

- Μεταβάσεις από και προς συνδετικούς συνδέσμους.
- Επαναλήψεις σε συνδετικούς συνδέσμους.
- Εσωτερικές μεταβάσεις σε συνδετικούς συνδέσμους.



Εικόνα 2.11: Γραφική απεικόνιση συνδετικού δεσμού.

Στο παρακάτω παράδειγμα (εικόνα 2.12), στα αριστερά, εάν η κατάσταση A είναι ενεργή όταν το γεγονός e συμβεί, τότε η μετάβαση από την κατάσταση A σε οποιαδήποτε κατάσταση D, E, F θα λάβει χώρα εάν πραγματοποιηθεί μία από τις συνθήκες [c1], [c2] ή [c3]. Στην ισοδύναμη αναπαράσταση στα δεξιά, η μετάβαση από την κατάσταση πηγή (source state) στον συνδετικό σύνδεσμο χαρακτηρίζεται από την ετικέτα του γεγονότος. Οι μεταβάσεις από τον σύνδεσμο προς κάθε κατάσταση προορισμό (destination state) χαρακτηρίζονται από τις συνθήκες. Εάν η κατάσταση A είναι ενεργή όταν συμβεί το γεγονός e, πρώτα συμβαίνει η μετάβαση από την κατάσταση A στον σύνδεσμο. Ακολουθεί η μετάβαση από τον σύνδεσμο σε κάποια κατάσταση προορισμό, βασιζόμενη στο ποια από τις συνθήκες [c1], [c2] ή [c3] είναι αληθής. Εάν καμία από τις συνθήκες δεν είναι αληθής, τότε δεν συμβαίνει μετάβαση και η κατάσταση A παραμένει ενεργή.



Εικόνα 2.12 : Παράδειγμα χρήσης συνδετικών συνδέσμων

2.2.12 Μη γραφικά αντικείμενα (Non Graphical objects)

Τα γεγονότα (events), τα δεδομένα (data) και τα αντικείμενα στόχου (target objects), δεν έχουν γραφική αναπαράσταση σε ένα διάγραμμα Stateflow.

Τα **γεγονότα (events)** οδηγούν στην εκτέλεση των διαγραμμάτων. Η εμφάνιση ενός γεγονότος μπορεί να προκαλέσει την εμφάνιση μίας μετάβασης ή την εκτέλεση μίας ενέργειας. Επειδή τα διαγράμματα του Stateflow ενεργοποιούνται αντιδρώντας στα γεγονότα, πρέπει να καθορίζονται και να προγραμματίζονται όλα τα γεγονότα που επηρεάζουν το διάγραμμα Stateflow. Μπορούμε να ορίσουμε άμεσα (explicit) γεγονότα, τα οποία δηλώνονται αμέσως. Επίσης, μπορούμε να δηλώσουμε έμμεσα (implicit) γεγονότα, τα οποία λαμβάνουν χώρα όταν εκτελούνται συγκεκριμένες ενέργειες.

Ένα διάγραμμα Stateflow αποθηκεύει και ανακτά **δεδομένα (data)** τα οποία τα χρησιμοποιεί για τον έλεγχο της εκτέλεσης του. Τα δεδομένα αυτά ανήκουν στο χώρο εργασίας του Stateflow, αλλά μπορούμε να έχουμε πρόσβαση και σε δεδομένα τα οποία ανήκουν εξωτερικά στο μοντέλο του Simulink ή στην εφαρμογή που ενσωματώνει ο μηχανισμός του Stateflow. Κατά την δημιουργία ενός μοντέλου στο Stateflow, πρέπει να ορίζονται όλα τα εσωτερικά και εξωτερικά δεδομένα (internal and external data) που χρησιμοποιούνται στο διάγραμμα του Stateflow.

Τέλος, υπάρχουν και τα **αντικείμενα στόχου (target objects)**. Κατασκευάζουμε στόχους (targets) στο Stateflow για να εκτελέσουμε τις εφαρμογές που κατασκευάσαμε στα διαγράμματα του Stateflow και στα μοντέλα του Simulink που τα περιέχουν. Ένας στόχος είναι ένα πρόγραμμα που εκτελεί ένα μοντέλο Stateflow ή ένα μοντέλο Simulink που περιέχει τον μηχανισμό του Stateflow. Είναι δυνατή η κατασκευή ενός στόχου προσομοίωσης (simulation target) για την εκτέλεση μίας προσομοίωσης του μοντέλου και ενός στόχου Real-Time Workshop (Real-Time Workshop target) για την εκτέλεση του μοντέλου Simulink σε ένα υποστηριζόμενο περιβάλλον (εδώ στο NXT).

2.2.13 Stateflow Coder 7.5

Το **Stateflow Coder** παράγει ακέραιο, κινητής υποδιαστολής ή σταθερής υποδιαστολής κώδικα C για τα Stateflow γραφήματα. Υποστηρίζει όλα τα Stateflow αντικείμενα και τη σημασιολογία, βοηθώντας στην ανάπτυξη και δοκιμή αλγορίθμων που μπορούν να αναπτυχθούν ως μεμονωμένες εφαρμογές ή να παρεμβάλλονται ως υπό-ρουτίνες στον υπάρχοντα κώδικα.

2.2.14 Simulink Design Verifier 1.6

Το **Simulink Design Verifier** (Επαληθευτής Σχεδίασης) λογισμικού δημιουργεί δοκιμές για τα Simulink και τα Stateflow μοντέλα που ικανοποιούν την κάλυψη μοντέλου και τους καθορισμένους στόχους του χρήστη. Αποδεικνύει επίσης τις ιδιότητες του μοντέλου και παράγει παραδείγματα παραβιάσεων.

2.2.15 BrickOS(Lejos)

Το λειτουργικό σύστημα που υπάρχει στο NXT και διατίθεται από την Lego περιλαμβάνει έναν διερμηνέα δυαδικού κώδικα (bytecode interpreter), που καθιστά το NXT ικανό να καταλαβαίνει τον κώδικα που παράγεται από το γραφικό περιβάλλον του λογισμικού της Lego (ROBOLAB). Επιτρέπει τον προγραμματισμό του NXT σε κώδικα του NXT, είναι όμως ένα πολύ απλό και περιορισμένο εργαλείο. Αντικαθιστώντας το λειτουργικό αυτό σύστημα με ένα άλλο, ο προγραμματιστής αποκτά εξ' ολοκλήρου τον έλεγχο χαμηλού επιπέδου του NXT. Μετά την κυκλοφορία των Lego Mindstorms, έγιναν διαθέσιμα αρκετά ανεξάρτητα εργαλεία ανάπτυξης όπως το NQC (Not Quiet C), το pbForth (Programmable Brick FORTH) και το BrickOS.

Το **BrickOS** είναι το πιο δυναμικό εργαλείο ανάπτυξης που έχει σχεδιαστεί να λειτουργεί στο NXT των Lego Mindstorms που βασίζεται στο μικροελεγκτή ARM7 της Arm Limited. Ξεκίνησε από τον Markus Noga (με την ονομασία LegOS) τον Οκτώβριο του 1998 και συνεχίζει να αναπτύσσεται σαν λογισμικό ανοικτού κώδικα (open source).

Το λειτουργικό αυτό σύστημα αντικαθιστά το σύστημα που είναι εγκατεστημένο στο NXT. Προσφέρει μεγάλη βελτίωση σε σχέση με το κλασσικό λειτουργικό σύστημα της Lego, και στην απόδοση αλλά και στην ευελιξία. Είναι ένα ολοκληρωμένο πακέτο το οποίο δίνει πρόσβαση στους κινητήρες, στους αισθητήρες και στα άλλα μέρη του NXT. Το BrickOS προσφέρει στους προγραμματιστές την δυνατότητα να αναπτύξουν προγράμματα για το NXT σε γλώσσες προγραμματισμού όπως η C και η C++.

Αποτελεί ένα ισχυρό εργαλείο, το οποίο όμως δεν είναι εύκολο στη χρήση. Στην αρχή κατασκευάστηκε για τα Linux και ακόμη και αν έχει γίνει αρκετή δουλειά για να μετασχηματιστεί και να τρέξει σε περιβάλλον Windows, χρειάζεται η εγκατάσταση ενός προσομοιωτή περιβάλλοντος Linux (Linux emulator) όπως το Cygwin, για να λειτουργήσει. Επίσης, απαιτείται ένας δια-μεταγλωττιστής C (C cross-compiler) για την μεταγλώττιση των προγραμμάτων πριν το τρέξιμο τους στο NXT. Ο δια-μεταγλωττιστής τρέχει στον ηλεκτρονικό υπολογιστή αλλά παράγει κώδικα για την πλατφόρμα του μικροελεγκτή της

Arm Limited. Επιτρέπει στον προγραμματιστή να γράφει πηγαίο κώδικα (source code) στη C σε ένα οποιοδήποτε περιβάλλον, ο οποίος στη συνέχεια μεταγλωττίζεται σε εκτελέσιμο πρόγραμμα για έναν διαφορετικό στόχο, δηλαδή τον μικροελεγκτή ARM7.

2.2.16 BricxCC v.3.3 (Bricx Command Center)

Το **Bricx Command Center** (BricxCC) είναι ένα πρόγραμμα των windows (95, 98, ME, NT, W2K, XP, Vista) program κοινός γνωστό και ως ολοκληρωμένο περιβάλλον ανάπτυξης IDE (integrated development environment) για τον προγραμματισμό του RCX (όλων των εκδόσεων), του Scout, του Cybermaster και του Spybot προγραμματιζόμενων τούβλων χρησιμοποιώντας την Not Quite C (NQC) γλώσσα του Dave Baum's. Υποστηρίζει τον προγραμματισμό RCX «τούβλων» σε C, C++, Pascal, Forth, και Java χρησιμοποιώντας τις brickOS, pbForth, and leJOS ως εναλλακτικά firmwares. Το BricxCC υποστηρίζει επίσης τον προγραμματισμό του LEGO Mindstorms NXT «τούβλου» χρησιμοποιώντας Not eXactly C (NXC), Next Byte Codes (NBC) και μια απλή γλώσσα προγραμματισμού τούβλων που καλείται NPG.

2.2.17 Cygwin v 1.5.25-15

Το **Cygwin** είναι μια συλλογή από εργαλεία ανοικτού κώδικα που επιτρέπει σε εφαρμογές Linux να μεταγλωττίζονται και να τρέχουν σε λειτουργικά συστήματα Windows, μέσα από ένα περιβάλλον διεπαφής που ομοιάζει με Linux. Η δυνατότητα αυτή, επιτρέπει στους προγραμματιστές να μεταφέρουν τις εφαρμογές τους από το λειτουργικό σύστημα Linux στο λειτουργικό σύστημα Windows, κάνοντας ευκολότερη την υποστήριξη τους. Περιλαμβάνει τον μεταγλωττιστή GCC.

2.2.18 GCC cross compiler

Ο **GCC cross compiler** είναι ένας μεταγλωττιστής που τρέχει σε έναν υπολογιστή ενώ είναι ικανός να δημιουργήσει εκτελέσιμο κώδικα για έναν διαφορετικό υπολογιστή. Οι μεταγλωττιστές χρησιμοποιούνται για την παραγωγή λογισμικού που μπορεί να τρέχει σε υπολογιστές με μια νέα αρχιτεκτονική ή σε συσκευές ειδικού σκοπού, οι οποίες δεν μπορούν να φιλοξενήσουν τους δικούς τους μεταγλωττιστές. Ένα τέτοιο εργαλείο είναι βολικό όταν χρειαζόμαστε να μεταγλωττίσουμε κώδικα για μία πλατφόρμα στην οποία δεν έχουμε πρόσβαση ή επειδή είναι αδύνατη η μεταγλώττιση για την συγκεκριμένη πλατφόρμα. Ο μεταγλωττιστής που θα χρησιμοποιηθεί στην παρούσα εργασία είναι ο Gcc cross compiler, ο οποίος προσφέρεται δωρεάν στο διαδίκτυο, είναι λογισμικό ανοικτού κώδικα και υποστηρίζει δεκάδες πλατφόρμες και γλώσσες προγραμματισμού.

2.2.19 Embedded Coder Robot NXT v 3.17

Το **Embedded Coder** είναι ένα περιβάλλον ανάπτυξης για τα Lego Mindstorms NXT από τον Takashi Chikamasa βασισμένο στην MATLAB και στο SIMULINK. Το Embedded Coder είναι βασίζεται στην σχεδιαστική μοντελοποίηση (MBD) για την παραγωγή κώδικα. Το Embedded Coder Robot NXT παρέχει δυνατότητες προγραμματισμού και μοντελοποίησης για την στρατηγική ελέγχου του NXT, δυναμικές εγκαταστάσεις, καθώς και δυνατότητα προσομοίωσης και οπτικοποίησης αυτών των συστατικών στοιχείων σε ένα τρισδιάστατο γραφικό περιβάλλον εικονικής πραγματικότητας.

Το Embedded Coder Robot NXT παρέχει επίσης το Real-Time Workshop Embedded Coder που βασίζεται στον στόχο ανάπτυξης του nxtOSEK με το πραγματικό υλικό του NXT. Η MATLAB και το Simulink είναι ιδιόκτητο λογισμικό που παρέχεται από την MathWorks και χρησιμοποιούνται ευρέως κυρίως στην μηχανική ελέγχου (control engineering). Το Simulink παρέχει πολλών τύπων blocks (από βασικές μαθηματικές λειτουργίες σε πολύπλοκες μαθηματικές εξισώσεις, καταστάσεις μηχανής/διαγράμματα ροής και blocks προσαρμόσιμα από τον χρήστη), όπου ο χρήστης μπορεί να καθορίσει την ροή των δεδομένων μέσω της σύνδεσης των blocks με γραμμές.

Μια άλλη σημαντική τεχνική του Embedded Coder Robot NXT είναι ότι τα block diagram (μοντέλα) της Simulink μπορούν να μεταγλωττιστούν σε κώδικα ANSI-C χρησιμοποιώντας το προϊόν Real Time Workshop Embedded Coder. Το Real-Time Workshop Embedded Coder παράγει ANSI-C κώδικα χωρίς να απαιτείται ειδικό “Run Time περιβάλλον”, έτσι ώστε ο χρήστης να μπορεί να εφαρμόσει τον παραγόμενο ANSI-C κώδικα για διάφορους τύπους υλικού (8bit/16bit/32bit μικροελεγκτές, DSP, και H/Y) χρησιμοποιώντας εργαλεία μεταγλωττιστή ANSI-C.

2.2.20 Real Time Workshop Embedded Coder

Το **Real-Time Workshop Embedded Coder** είναι ένα ξεχωριστό, επιπρόσθετο προϊόν για χρήση με το Real-Time Workshop.

Παρέχει ένα πλαίσιο εργασίας για την ανάπτυξη ενός βελτιστοποιημένου κώδικα όσον αφορά την ταχύτητα, την χρήση μνήμης και την απλότητα του. Το Real-Time Workshop Embedded Coder προτείνεται για χρήση με ενσωματωμένα συστήματα (embedded systems). Επίσης, παράγει κώδικα που είναι εύκολο να διαβαστεί, να σχεδιαστεί και να τροποποιηθεί για οποιοδήποτε περιβάλλον παραγωγής (production environment).

2.2.21 nxtOSEK v 2.09

Η πλατφόρμα **nxtOSEK v 2.09** παρέχει κώδικα C/C++ API που καλείται ECRobot. Αρχικά σχεδιάστηκε για την ανάπτυξη MATLAB & Simulink που βασίζεται στο

περιβάλλον σχεδιαστικής μοντελοποίησης (που καλείται Embedded Coder Robot) για LEGO MINDSTORMS NXT.

2.2.22 NeXTTool

Το **NeXTTool** είναι μια κοσόλα εφαρμογών για Η/Υ η οποία ενεργοποιεί την φόρτωση εκτελέσιμων αρχείων .rxe(executable files) και αρχείων .rfw (firmware files) στο NXT. Το NeXTTool έχει σχεδιαστεί από τον John Hansen. Το NeXTTool χρησιμοποιεί το LEGO MINDSTORMS NXT Driver για την διεπαφή του NXT μέσω USB (και Bluetooth, αλλά δεν χρησιμοποιείται για τον Embedded Coder Robot NXT). Κατά την εγκατάσταση είναι αναγκαία μόνο η εξαγωγή των αρχείων.

Αναφορικά : Το NeXTTool δεν εμφανίζει μηνύματα σχετικά με εντολές επιτυχίας/αποτυχίας.

2.2.23 GNU ARM toolchain v 4.0.2

Η **GNU ARM toolchain** χρησιμεύει για την διανομή του GCC (του GNU μεταγλωττιστή) για τον πυρήνα ARM το οποίο υποστηρίζει τον ARM7 CPU στο NXT. Κατά την εγκατάσταση δεν είναι αναγκαία να σημειωθεί η μονάδα κινητής υποδιαστολής

Αναφορικά : Αν χρησιμοποιηθεί άλλη έκδοση GCC είναι πιθανή η εμφάνιση σφαλμάτων.

2.2.24 MINDSTORMS NXT Driver v1.02

Το λογισμικό **MINDSTORMS NXT Driver v1.02** εγκαθιστά και ενημερώνει τον οδηγό (driver) του Lego Mindstorms NXT, ο οποίος ενεργοποιεί τα windows ώστε να αναγνωρίζουν το NXT.

2.2.25 IVT BlueSoleid EDR Edition v 2.1

Το **BlueSoleid** είναι ένα λογισμικό βασισμένο στα Windows από την IVT που επιτρέπει την ενεργοποίηση Bluetooth στην επιφάνεια εργασίας του υπολογιστή ώστε να μπορεί να συνδεθεί ασύρματα με άλλες ενεργές συσκευές Bluetooth. Το BlueSoleil επιτρέπει στους χρήστες των MS Windows να έχουν ασύρματη πρόσβαση σε μια μεγάλη ποικιλία Bluetooth ψηφιακών συσκευών, όπως το NXT, φωτογραφικές μηχανές, κινητά τηλέφωνα, ακουστικά, εκτυπωτές, και δέκτες GPS. Επίσης έχει την δυνατότητα σχηματισμού δικτύου και την ανταλλαγή δεδομένων με άλλους υπολογιστές που υποστηρίζουν τεχνολογία Bluetooth ή PDA. Οι πλατφόρμες που υποστηρίζονται από το BlueSoleil περιλαμβάνουν τα : Windows 98SE, ME, 2000 , XP και VISTA.

2.2.26 NXT GamePad v 1.04

Το NXT GamePad v 1.04 του Tomoki Fakuda είναι μια εφαρμογή που επιτρέπει τον έλεγχο του ρομπότ μέσω Bluetooth με αναλογικό χειριστήριο παιχνιδιών.

2.2.27 LEGO Digital Designer

Το LEGO Digital Designer είναι ένα πρόγραμμα CAD που επιτρέπει την σχεδίαση/κατασκευή με τουβλάκια LEGO στον υπολογιστή.

2.3 Εγκατάσταση Λογισμικού

Μετά την περιγραφή του απαραίτητου λογισμικού, ακολουθεί μία σύντομη περιγραφή της διαδικασίας εγκατάστασής του. Τα βήματα που πρέπει να ακολουθηθούν είναι:

1. **Εγκατάσταση της MATLAB.** Για την δημιουργία, την τροποποίηση και την αποστολή των μοντέλων Simulink, μέσω του Bluetooth στο NXT, είναι απαραίτητη η εγκατάσταση της MATLAB έκδοση R2008b (ή R2006a). Επιλέγουμε τα παρακάτω πακέτα λογισμικού, καθώς είναι απαραίτητα τόσο για την επεξεργασία των μοντέλων στο γραφικό περιβάλλον του Simulink όσο και για την επικοινωνία με το NXT:

Products	Version	Release
MATLAB	7.7.0	R2008b
Simulink	7.2	R2008b
Fixed-Point Toolbox	2.3	R2008b
MATLAB Report Generator	3.4	R2008b
Real-Time Workshop	7.2	R2008b
Real-Time Workshop Embedded Coder	5.2	R2008b
Simulink Design Verifier	1.3	R2008b
Simulink Fixed-Point	6.0	R2008b
Simulink Report Generator	3.4	R2008b
Simulink Verification and Validation	2.4	R2008b
Stateflow	7.2	R2008b
Stateflow Coder	7.2	R2008b
Virtual Reality Toolbox	4.8	R2008b

2. **Εγκατάσταση του Cygwin/GNU Make.** Το Cygwin και το GNU Make είναι αναγκαία προκειμένου να εγκατασταθεί ο κατάλληλος εξομοιωτής περιβάλλοντος για το ATMEL AT91SAM7S256 (ένας ARM7 πυρήνας μικροεπεξεργαστή) στο NXT.

2.1 Κατεβάστε στην Cygwin 1.5.x ή κάποια πιο καινούργια έκδοση (εμείς χρησιμοποίησαμε την 1.5.24) από την Cygwin: <http://www.cygwin.com/>

2.2 Εκτελέστε την εγκατάσταση.



2.3 Εγκαταστήστε το Cygwin σε διεύθυνση που δεν περιέχει κενά ή multi-byte χαρακτήρες (πχ C:/cygwin).

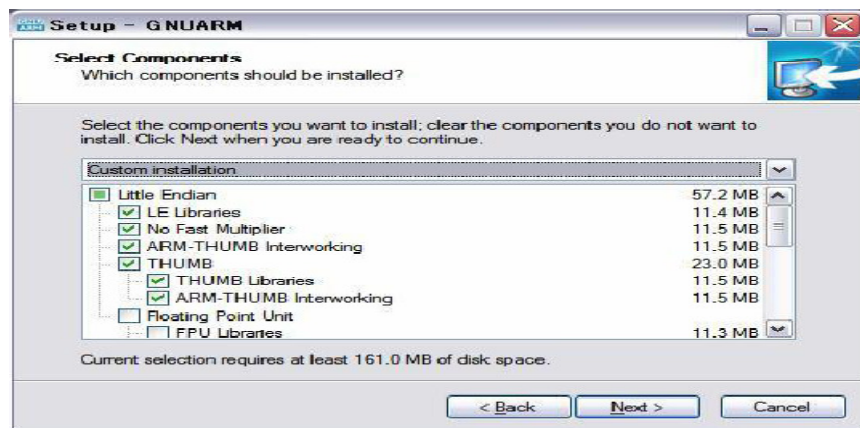


3. Εγκατάσταση του GNU ARM. Το GNU ARM είναι ένας GCC διαμεταγλωτιστής για το ARM και υποστηρίζει τον ATMEL AT91SAM7S256 στο NXT.

3.1 Κατεβάστε την εγκατάσταση GCC-4.0.2 από το site της GNU ARM:
<http://www.gnuarm.com/>
http://www.gnuarm.com/bu-2.16.1_gcc-4.0.2-c-c++_nl-1.14.0_gi-6.4.exe

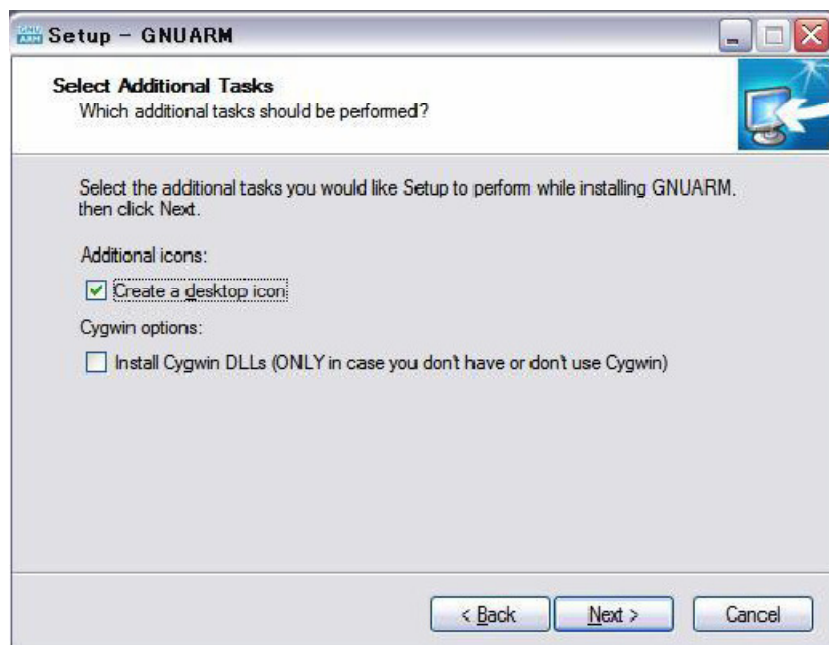
3.2 Εκτελέστε την εγκατάσταση και εγκαταστήστε τον GNU ARM σε μία διεύθυνση η οποία δεν περιέχει κενά ή multy-byte χαρακτήρες (πχ C:/cygwin/GNUARM).

3.3 Ο ATMEL AT91SAM7S256 δεν έχει FPU για αυτό ακολουθήστε την εγκατάσταση όπως σας παρουσιάζετε στα παράθυρα που ακολουθούν.



Εικόνα του GNUARM Installer

3.4 Αφαιρέστε την επιλογή όπως φαίνεται στο σχήμα. Το Cygwin DLLs επειδή το Cygwin έχει ήδη εγκατασταθεί.



3.5 Στο τέλος της εγκατάστασης σας ζητείτε να προσθέσετε ως διεύθυνση εγκατάστασης το Windows Environment Path, αλλά δεν είναι αναγκαίο.

4. **Εγκατάσταση του LEGO standard USB Driver.** Αν το καθιερωμένο LEGO λογισμικό (NXT-G) είναι ήδη εγκατεστημένο στον υπολογιστή μπορείτε να προσπεράσετε την εγκατάσταση του LEGO MINDSTORMS Driver v1.02.

Κατεβάστε από το site <http://mindstorms.lego.com/Support/Updates/>



MINDSTORMS NXT Driver v1.02

Description

This software updates the LEGO MINDSTORMS NXT driver and addresses an issue that prevents the firmware from being downloaded to the NXT on some occasions.

When you have installed the new driver and want to download new firmware to the NXT, please initialize the NXT brick before you start downloading the firmware. **If your NXT is clicking when you insert batteries**, push the hardware reset button for five seconds before you insert the USB cable. This will ensure that the brick initializes correctly for the firmware download. The hardware reset button is located within the LEGO Technic hole below the USB connector on the NXT brick. **If your NXT brick is not clicking when you insert batteries** just go through the normal firmware download process as described within the manual when the new driver is installed.

Version
1.02

Post Date
26/07/2006

PC
7.12MB
MAC
351KB

Instructions

Click on the download to save the patch to your harddrive. Unzip the downloaded archive and run Setup.exe

System Requirements
Mac OS X or Windows XP

MINDSTORMS NXT Driver v1.02

5. **Εγκατάσταση του NeXTTool και του Enhanced NXT firmware.** Το NeXTTool είναι ένα πρόγραμμα πλατφόρμας H/Y το οποίο ενεργοποιεί την φόρτωση των .rxn αρχείων (NXT application program) και των .rfw αρχείων (NXT firmware program) στο NXT.

5.1 Κατεβάστε το NeXTTool από το : <http://bricxss.sourceforge.net/utilities.html> και εξαγωγή του σε διεύθυνση χωρίς κενά και Multi-byte χαρακτήρες.

Να σημειωθεί ότι το NeXTTool παρέχει πολλά χαρακτηριστικά τα οποία επικοινωνούν με το NXT. Παρόλα αυτά δεν δουλεύει με την nxtOSEK επειδή η nxtOSEK δεν χρησιμοποιεί το καθιερωμένο πρωτόκολλο επικοινωνίας LEGO για USB/Bluetooth σύνδεση.

5.2 Κατεβάστε το Enhanced NXT firmware από: http://bricxcc.sourceforge.net/lms_arm_jch.zip και κάντε το εξαγωγή στην διεύθυνση όπου αποθηκεύτηκε το NeXXtool.



6. Φόρτωση του Enhanced NXT firmware στο NXT.

6.1 Πιέστε το “reset button” (βρίσκεται στο πίσω μέρος του NXT στην πάνω αριστερή γωνία του USB connector).

6.2 Συνδέστε τον Η/Υ και το NXT μέσω USB καλωδίου.,

6.3 Εκτελέστε το Cygwin και γράψτε τις ακόλουθες εντολές στην γραμμή εντολών για να αλλάξετε την τωρινή διεύθυνση του NeXTTool στην εξαγόμενη διεύθυνση. (Το NeXTTool υποθέτεται ότι έχει εξαχθεί στο C:/cygwin/nexttool directory)

```
$ cd C:/cygwin/nexttool
```

6.4 Γράψτε την ακόλουθη εντολή για την φόρτωση του Enhanced NXT firmware στο NXT.

```
$ ./nexttool.exe /COM=usb -firmware=lms_arm_nbcnxc_107.rfw
```

Για την φόρτωση του προγράμματος χρειάζεται περίπου μισό λεπτό και μετά εμφανίζονται κάποιες κουκίδες στην LCD οθόνη του NXT.

6.5 Βγάλτε την μπαταρία από το NXT και ξαναβάλτε την και πιέστε το πορτοκαλί κουμπί του NXT για να ανοίξει το Enhanced NXT firmware. Το Enhanced NXT firmware έχει το ίδιο GUI όπως το καθιερωμένο NXT firmware.

Troubleshooting:

Verify that this worked by typing

```
nexttool /COM=usb -versions
```

You should see the following at the prompt, where the firmware version should match what you downloaded:

```
Protocol version = 1.124
```

```
Firmware version = 1.06
```

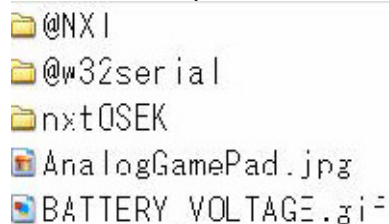
7. Εγκατάσταση του Embedded Coder Robot NXT. Εξάγεται τον φάκελο ecrobotNXT σε διεύθυνση χωρίς κενά και multi-byte χαρακτήρες. Εκτελέστε το αρχείο ecrobotnxtsetup.m στην Matlab και εκτελούμε τις εξής εντολές.

Specified path to cygwin/bin: C:/cygwin/bin

Specified path to GNUARM root: C:/cygwin/GNUARM

Specified path to NeXTTool root : C:/cygwin/nexttool

Set MATLAB path to environment directory



```
@NXT
@w32serial
nxtOSEK
AnalogGamePad.jpg
BATTERY_VOLTAGE.gif
```

Εικόνα: ecrobotNXT/environment direction

8. Εγκατάσταση του NXT Game Pad. Εξαγωγή του φακέλου NXT Game Pad.

9. Εγκατάσταση του BlueSoleid 2.6. Εκτέλεση της εφαρμογής BlueSoleid 2.6.exe.

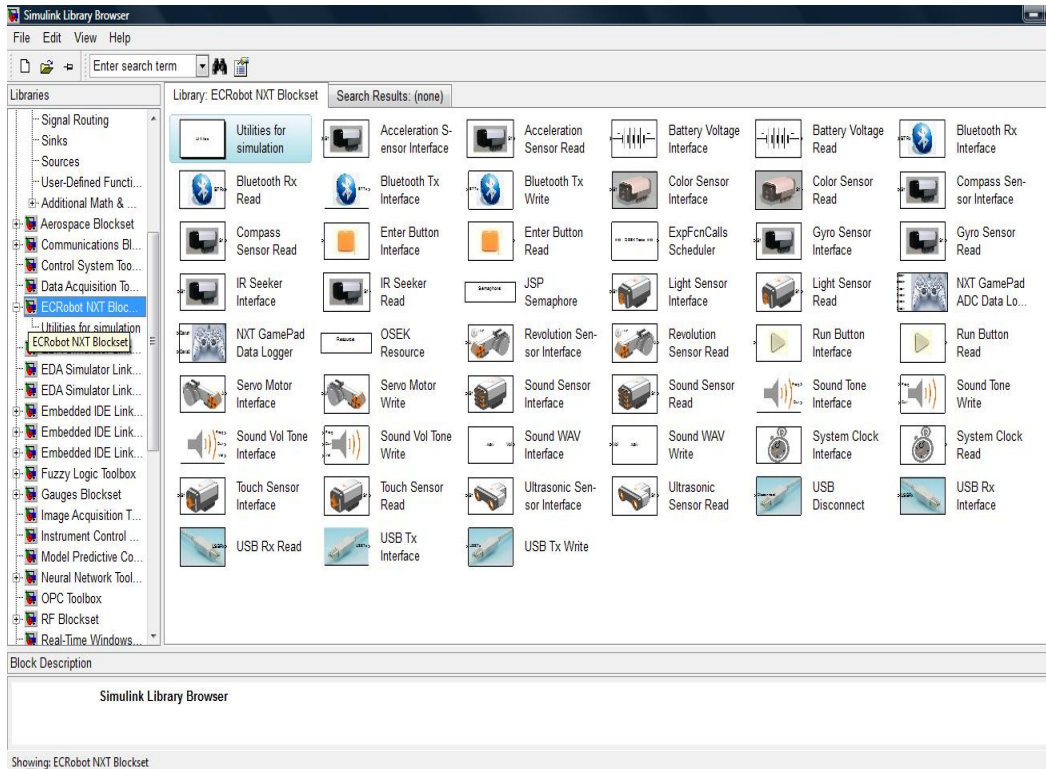
10. Εγκατάσταση του Microsoft Gamepad xbox360.

11. Εγκατάσταση του Lego Digital Designer.

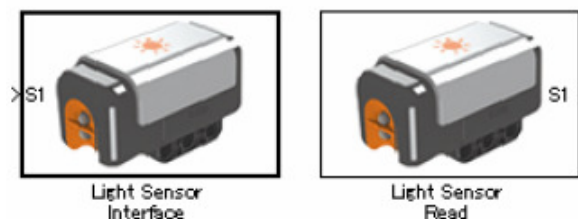
ΚΕΦΑΛΑΙΟ 3 : ΒΙΒΛΙΟΘΗΚΗ EMBEDDED CODER ROBOT NXT

- **ECRobot NXT Blockset**

Εφόσον η βιβλιοθήκη ECRobot NXT έχει εγκατασταθεί επιτυχώς, το ECRobot NXT Blockset εμφανίζεται στον πλοηγό της βιβλιοθήκης Simulink.



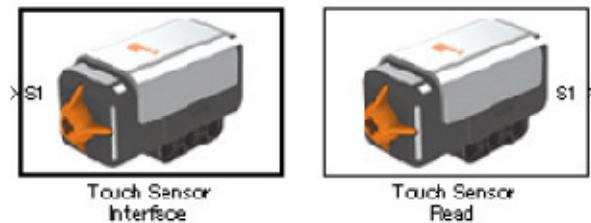
3.1 Light Sensor blocks



Τα **Light Sensor Blocks** χρησιμοποιούνται για την μέτρηση της φωτεινότητας. Η μεγαλύτερες τιμές σημαίνουν σκοτεινότερο(ή οι μικρότερες τιμές την αντανάκλαση). Ο Light Sensor αποτελείται από δύο Blocks. Το light Sensor Interface είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Light Sensor Read block το οποίο χρησιμοποιείται για την ανάγνωση δεδομένων του Light Sensor. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις θα χρησιμοποιηθούν για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	unt16
Dimension	[1 1]
Data Range	0 to 1023 (raw A/D value)
Port ID	S1/S2/S3/S4

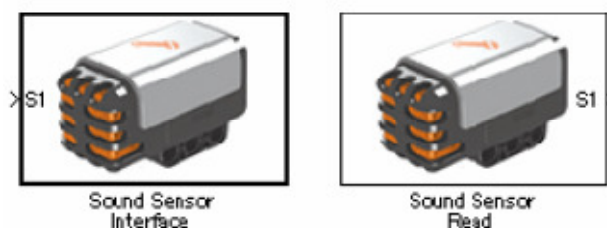
3.2 Touch Sensor blocks



Τα Touch Sensors Blocks χρησιμοποιούνται για τον εντοπισμό επαφής με κάποιο αντικείμενο. Όταν ο Touch Sensor έρθει σε επαφή με κάποιο αντικείμενο ο αισθητήρας επιστρέφει την τιμή 1. Ο Touch Sensor αποτελείται από δύο Blocks. Το Touch Sensor Interface είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Touch Sensor Read block το οποίο χρησιμοποιείται για την ανάγνωση δεδομένων του Touch Sensor. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις θα χρησιμοποιηθούν για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	unt8
Dimension	[1 1]
Data Range	0(not touched), 1(touched)
Port ID	S1/S2/S3/S4

3.2 Sound Sensor blocks

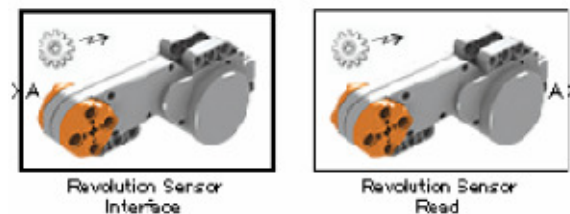


Τα Sound Sensor Blocks χρησιμοποιούνται για την μέτρηση της έντασης του ήχου. Όσο πιο μικρή είναι η τιμή τόσο πιο μεγάλης έντασης είναι ο ήχος. Ο Sound Sensor αποτελείται από δύο blocks. Το Sound Sensor Interface block είναι ένα block διεπαφής

(interface) εξωτερικά του NXT controller model. Το Sound Sensor Read block το οποίο χρησιμοποιείται για την ανάγνωση δεδομένων του Sound Sensor. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις θα χρησιμοποιηθούν για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	unt16
Dimension	[1 1]
Data Range	0 to 1023 (raw A/D value)
Port ID	S1/S2/S3/S4

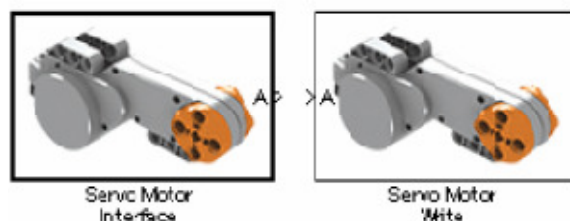
3.3 Revolution Sensor blocks



Τα Revolution Sensor blocks χρησιμοποιούνται για την μέτρηση της περιστροφής του σερβοκινητήρα. Ο Revolution Sensor αποτελείται από δύο blocks. Το Revolution Sensor Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Revolution Sensor Read block το οποίο χρησιμοποιείται για την ανάγνωση των δεδομένων περιστροφής του Revolution Sensor. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις θα χρησιμοποιηθούν για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	int32
Dimension	[1 1]
Data Range	Range of int32 [deg]
Port ID	A/B/C

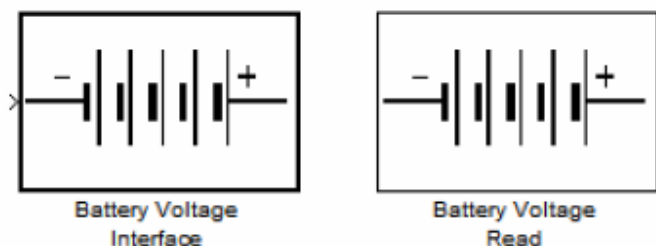
3.4 Servo Motor blocks



Τα Servo Motor Blocks χρησιμοποιούνται για τον έλεγχο των σερβοκινητήρων. Ο Servo motor αποτελείται από δύο blocks. Το Servo Motor Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Servo Motor write block χρησιμοποιείται για να οριστούν τα δεδομένα ελέγχου του σερβοκινητήρα. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις θα χρησιμοποιηθούν για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	int8
Dimension	[1 1]
Data Range	-100 to 100
Port ID	A/B/C
Mode	Brake/Float

3.5 Battery Voltage blocks



Τα Battery Voltage Blocks χρησιμοποιούνται για να ελέγξουν την τάση των μπαταριών του NXT. Το Battery Voltage αποτελείται από δύο blocks. Το Battery Voltage Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Servo Battery Voltage Read block το οποίο χρησιμοποιείται για την ανάγνωση των δεδομένων της μπαταρίας. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data type	unt16
Dimension	[1 1]
Data Range	0 to possible NXT maximum voltage value in mv (i.e. 9000 = 9.000 V)
Port ID	None

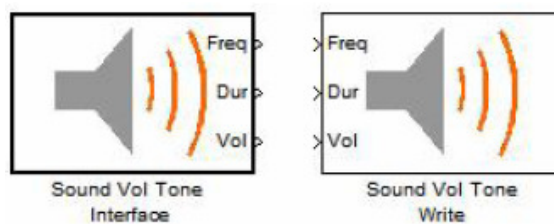
3.6 System Clock blocks



Τα System Clock Blocks χρησιμοποιούνται για την ανάγνωση των «τιμ» του συστήματος του NXT. Το System Clock αποτελείται από δύο blocks. Το System Clock Interface block δεν έχει είσοδο/έξοδο (input/output), εντούτοις παρέχει «τιμ» συστήματος στο System Clock Read block κατά την διάρκεια της προσομοίωσης. Χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα. Στο πραγματικό NXT το «τιμ» του συστήματος ξεκινάει από το 0 όταν το NXT ανοίξει. (Δεν ξεκινάει με την έναρξη της εφαρμογής του προγράμματος του ERobot.)

Data Type	uint32
Dimension	[1 1]
Data Range	0 to maximum of uint32 in mille-second
Port ID	None

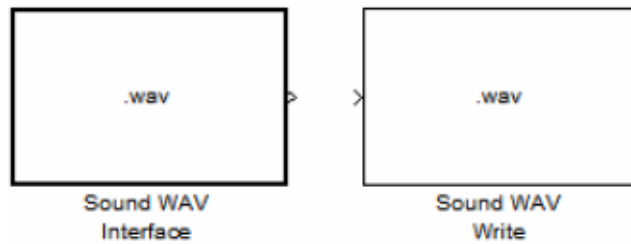
3.7 Sound Volume Tone blocks.



Τα Sound Volume blocks χρησιμοποιούνται για την παραγωγή ήχου ο οποίος καθορίζεται από την συχνότητα και την διάρκεια. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	Freq: uint32, Dur: uint32
Dimension	[1 1]
Data Range	Freq: 31 [Hz] to 14080 [Hz], Dur: 10 to 2560 [msec], Vol: 0 to 100 (0 is mute)
Port ID	None

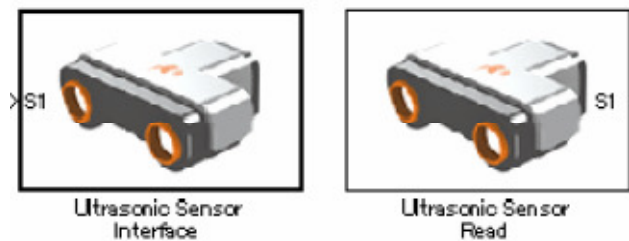
3.8 Sound Wav blocks



Τα Sound Wav blocks χρησιμοποιούνται για την παραγωγή ήχου ο οποίος αποθηκεύεται ως 8bit μονοφωνικό WAV. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	Freq: uint32, Dur: uint32
Dimension	[1 1]
Data Range	-
Port ID	None

3.9 Ultrasonic Sensor blocks



Τα Ultrasonic Sensor blocks χρησιμοποιούνται για τον υπολογισμό της απόστασης από ένα εμπόδιο ή για τον εντοπισμό εμποδίου χωρίς επαφή. Τα Ultrasonic Sensor blocks αποτελούνται από δύο Blocks. Το Ultrasonic Sensor Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Ultrasonic Sensor Read block χρησιμοποιείτε για την ανάγνωση των δεδομένων του Ultrasonic Sensor. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	int32
Dimension	[1 1]
Data Range	0 to 255 [cm], -1 (the sensor is not ready for measurement)
Port ID	S1/S2/S3/S4

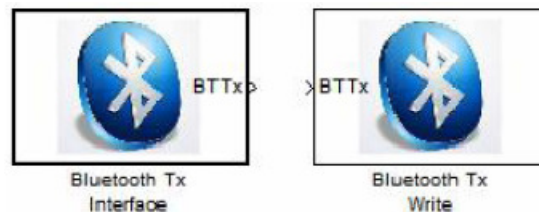
3.10 Bluetooth Rx blocks



Τα Bluetooth Rx blocks αντιπροσωπεύουν την ασύρματη επικοινωνία από τον Η/Υ στο NXT. Τα Bluetooth Rx blocks αποτελούνται από δύο blocks. Το Bluetooth Rx Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Bluetooth Rx Read block χρησιμοποιείτε για την ανάγνωση δεδομένων. Τα Bluetooth Rx blocks επιτρέπονται για δέσμευση μόνο ενός στιγμιότυπου για το μοντέλο. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα. Αυτή η σύνδεση Bluetooth υποστηρίζετε μόνο για Η/Υ-NXT (δεν υποστηρίζετε για NXT-NXT).

Data Type	uint8
Dimension	32
Data Range	0 to 255
Port ID	None

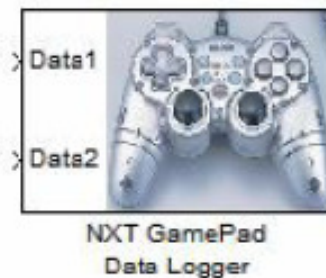
3.11 Bluetooth Tx Blocks



Τα Bluetooth Tx blocks αντιπροσωπεύουν την ασύρματη επικοινωνία από το NXT στον Η/Υ. Το Bluetooth Tx block αποτελείται από δύο blocks. Το Bluetooth Tx Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Bluetooth Tx Write block το οποίο χρησιμοποιείται για την αποστολή δεδομένων. Τα Bluetooth Tx blocks επιτρέπονται για δέσμευση μόνο ενός στιγμιότυπου για το μοντέλο. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα. Η Bluetooth σύνδεση υποστηρίζετε και για τα δύο Η/Υ-NXT και NXT-NXT.

Data Type	uint8
Dimension	32
Data Range	0 to 255
Port ID	None

3.12 NXT GamePad Data Logger block



Από την πλατφόρμα nxtOSEK έχει παραχθεί ένα εργαλείο χρησιμότητας που καλείται NXT GamePad. Το NXT GamePad επιτρέπει στον χρήστη τον απομακρυσμένο έλεγχο του NXT, το οποίο διαθέτει πλατφόρμα nxtOSEK, μέσω Bluetooth με την χρήση του αναλογικού PC GamePad. Από την έκδοση V1.01 ο χρήστης είναι σε θέση να επιλέξει "Analog Stick Control" με και/ή ανάκτηση δεδομένων του NXT ανάλογα με τα χαρακτηριστικά που επιθυμεί. Ακόμη και αν ο χρήστης δεν διαθέτει αναλογικό GamePad, το NXT GamePad μπορεί να χρησιμοποιηθεί για την ανάκτηση δεδομένων από το NXT ('NXT Data Acquisition') και την φόρτωση δεδομένων (data logging). Το NXT GamePad Data logger block δεν επηρεάζει τα αποτελέσματα προσομοίωσης αλλά χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα. Τα Data1 και Data2 μπορούν να χρησιμοποιηθούν για την φόρτωση εισόδων του αναλογικού GamePad. Τα δεδομένα που έχουν φορτωθεί μπορούν να αποθηκευτούν σε ένα CSV αρχείο και μπορεί να αναλυθεί η συμπεριφορά της εφαρμογής του NXT στην Matlab.

Το NXT GamePad Data Logger block δεν πρέπει να χρησιμοποιείται μαζί με το Bluetooth Tx Write block/NXT GamePad ADC Data Logger block στο μοντέλο.

Data Type	int8
Dimension	[1 1]
Data Range	-128 to 127
Port ID	None



	A	B	C	D	E	F	G	H	I	J	K	L
1	Time	Data1	Data2	Battery	Motor Rev A	Motor Rev B	Motor Rev C	ADC S1	ADC S2	ADC S3	ADC S4	I2C
320	3180	0	0	8156	0	0	0	451	984	478	1023	255
321	3190	0	0	8156	0	0	0	451	997	478	1023	255
322	3200	0	0	8156	0	0	0	451	969	478	1023	255
323	3210	38	0	8156	0	0	0	451	1002	478	1023	255
324	3220	38	0	7976	0	0	0	451	981	477	1023	255
325	3230	90	0	7962	0	1	1	450	985	477	1023	255
326	3240	90	0	7546	0	2	2	451	988	478	1023	255
327	3250	90	0	7546	0	4	5	450	1001	478	1023	255
328	3260	90	0	7657	0	8	8	450	1003	478	1023	255
329	3270	90	0	7685	0	12	11	450	992	477	1023	255
330	3280	100	0	7689	0	15	15	451	986	477	1023	255
331	3290	100	0	7588	0	19	19	451	988	477	1023	255
332	3300	100	0	7580	0	23	23	451	1000	477	1023	255
333	3310	100	0	7546	0	27	27	452	1002	478	1023	255
334	3320	100	0	7602	0	31	31	452	982	477	1023	255
335	3330	100	0	7643	0	35	35	450	985	465	1023	255
336	3340	100	0	7671	0	40	40	450	987	468	1023	255
337	3350	100	0	7713	0	45	48	447	1000	484	1023	255
338	3360	100	0	7726	0	50	51	447	1002	478	1023	255
339	3370	100	0	7754	0	55	57	447	981	475	1023	255
340	3380	100	0	7754	0	62	62	448	985	472	1023	255

Τα καταγραφόμενα δεδομένα του NXT GamePad αποθηκευμένα σε ένα CSV αρχείο.

3.13 NXT GamePad ADC Data Logger block



Το NXT GamePad ADC Data Logger block έχει την ίδια λειτουργικότητα με το NXT GamePad Data Logger, αλλά ο χρήστης μπορεί να διαμορφώσει τα τέσσερα σήματα που καταγράφονται. Το NXT GamePad Data Logger block δεν πρέπει να χρησιμοποιείται μαζί με το Bluetooth Tx Write block/NXT GamePad Data Logger block στο μοντέλο. Αυτό το block απαιτεί την πλατφόρμα nxtOSEK στο *Exported Function-Calls Scheduler* block

για να επιδείξει στον χρήστη τις επιλεγμένες ADC τιμές στην οθόνη υγρών κρυστάλλων LCD.

Input name	Data type	Data range	Dimension
Data1/Data2	int8	-128 to 127	1
ADC1/ADC2/ADC3/ADC4	int16	-32768 to 32767	1

3.14 Acceleration Sensor blocks



Τα Acceleration Sensor blocks χρησιμοποιούνται για την μέτρηση της επιτάχυνσης στους τρεις άξονες. Ο Acceleration Sensor δεν υπάρχει στα καθιερωμένα προϊόντα της LEGO αλλά φτιάχτηκε από την Hitechnic. Τα Acceleration Sensor blocks αποτελούνται από δύο blocks. Το Acceleration Sensor Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Acceleration Sensor Read block χρησιμοποιείται για την ανάγνωση των δεδομένων του Acceleration Sensor. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	int16
Dimension	[1 3] (index 1: x axis, index 2: y axis, index 3: z axis)
Data Range	-512 to 511
Port ID	S1/S2/S3/S4

3.15 Gyro Sensor Blocks



Τα Gyro Sensor Blocks χρησιμοποιούνται για την μέτρηση της τιμής που αντιπροσωπεύει τον αριθμό των μοιρών ανά δευτερόλεπτο της περιστροφής. Το Gyro

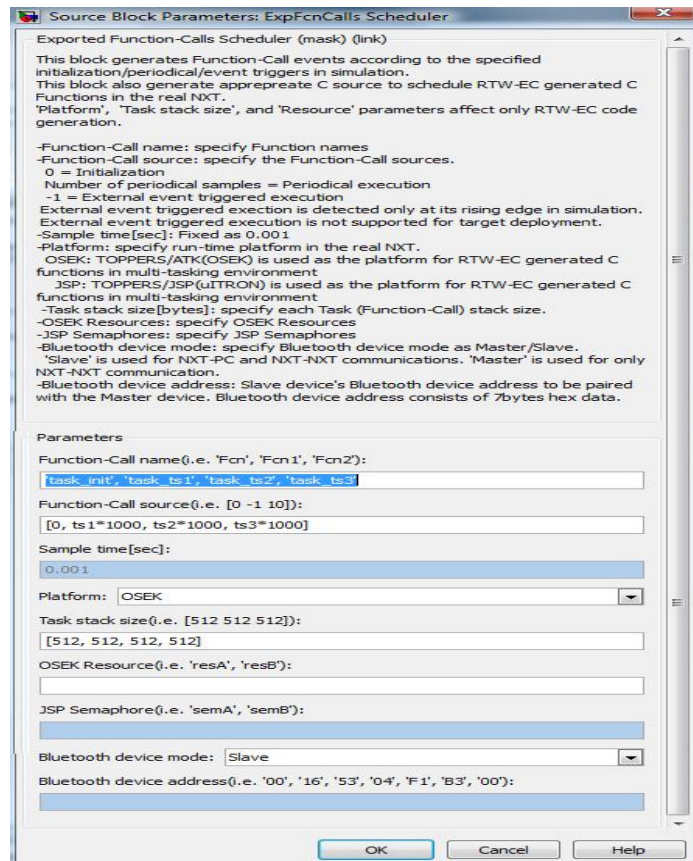
Sensor αποτελείται από δύο blocks. Το Gyro Sensor Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το Gyro Sensor Read block χρησιμοποιείται για την ανάγνωση δεδομένων από τον Gyro Sensor. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	unt16
Dimension	[1 1]
Data Range	0 to 1023 (raw A/D value)
Port ID	S1/S2/S3/S4

3.16 Exported Function-Calls Scheduler block



Το χαρακτηριστικό ενσωματωμένο λογισμικό συστημάτων ελέγχου (embedded control system software) απαιτείται για να εκτελέσει τη στρατηγική ελέγχου του σε αρκετές διαφορετικές χρονικές στιγμές (δηλ. έναρξη, ασύγχρονα, περιοδικά). Το Exported Function-Calls Scheduler block ελέγχει τον συγχρονισμό εκτέλεσης του Function Call subsystem κατά τη διάρκεια της προσομοίωσης σύμφωνα με τις καθορισμένες παραμέτρους στο Block Parameters dialog. Αυτές οι πληροφορίες παραμέτρου θα χρησιμοποιηθούν επίσης για την παραγωγή του αρχείου ecrebot_main.c. Από το λογισμικό της πλατφόρμα Embedded Coder Robot NXT v3.16 μπορεί να διαμορφωθεί είτε από την TOPPERS/OSEK είτε από την TOPPERS/JSP. Η TOPPERS/JSP συμμορφώνεται με το Ιαπωνικό αρχικό open RTOS προδιαγραφής “μ ITRON 4.0”.



3.17 USB Rx blocks



Τα USB Rx blocks αντιπροσωπεύουν την USB επικοινωνία από τον Η/Υ στο NXT. Τα USB Rx blocks αποτελούνται από δύο blocks. Το USB Rx Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το USB Rx Read block χρησιμοποιείται για την ανάγνωση δεδομένων. Τα USB Rx blocks επιτρέπονται για δέσμευση μόνο ενός στιγμιότυπου για το μοντέλο. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	uint8
Dimension	64
Data Range	0 to 255
Port ID	None

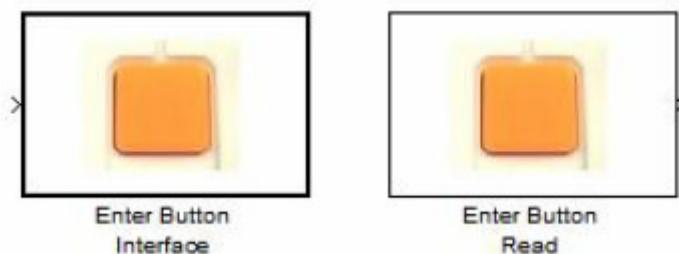
3.18 USB Tx blocks



Τα USB Tx blocks αντιπροσωπεύουν την επικοινωνία από το NXT στον Η/Υ. Τα USB Tx blocks αποτελούνται από δύο blocks. Το USB Tx Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το USB Tx write block χρησιμοποιείται για την αποστολή δεδομένων. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	uint8
Dimension	64
Data Range	0 to 255
Port ID	None

3.19 ENTER button blocks



Τα ENTER button blocks χρησιμοποιούνται για τον εντοπισμό επαφής με κάποιο εμπόδιο. Αν το Enter button πιεστεί επιστρέφει τιμή 1. Τα ENTER button blocks αποτελούνται από δύο blocks. Το ENTER button Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το ENTER button Read block χρησιμοποιείται για την ανάγνωση της κατάστασης του ENTER button. Στην

προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

Data Type	unt8
Dimension	[1 1]
Data Range	0(not touched), 1(touched)
Port ID	S1/S2/S3/S4

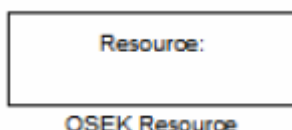
3.20 RUN button blocks



Τα RUN button blocks χρησιμοποιούνται για τον εντοπισμό επαφής με κάποιο εμπόδιο. Αν το RUN button πιεστεί επιστρέφει την τιμή 1. Τα RUN button blocks αποτελούνται από δύο blocks. Το RUN button Interface block είναι ένα block διεπαφής (interface) εξωτερικά του NXT controller model. Το RUN button Read block χρησιμοποιείται για την ανάγνωση της κατάστασης του RUN button. Στην προσομοίωση αυτά τα blocks είναι κάτοχοι θέσεων, εντούτοις χρησιμοποιούνται για να θέσουν σε εφαρμογή την κατάλληλη συσκευή API στον παραγόμενο κώδικα.

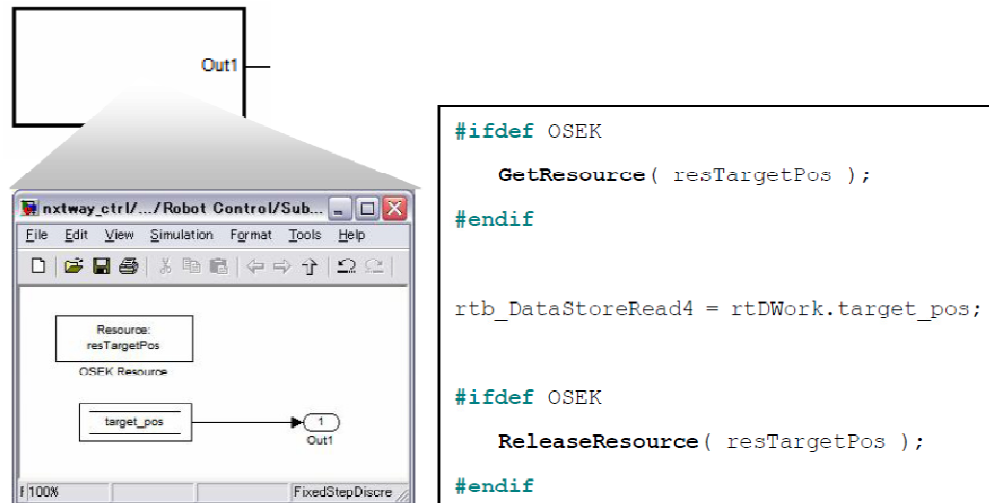
Data Type	unt8
Dimension	[1 1]
Data Range	0(not touched), 1(touched)
Port ID	S1/S2/S3/S4

3.21 OSEK Resource block



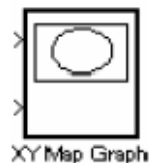
Το OSEK Resource block χρησιμοποιείται για να εφαρμόσει το OSEK GetResource/ReleaseResource API κατά την αρχή/τέλος του κρίσιμου τμήματος. Στην προσομοίωση το κρίσιμο τμήμα θα πρέπει να τοποθετηθεί μέσα σε ένα ατομικό υποσύστημα και το OSEK Resource block θα πρέπει να δεσμεύεται μέσα στο ατομικό

υποσύστημα με καθορισμένο Resource. Για τον καθορισμό ενός Resource identifier στο block πρέπει να καθοριστεί ο Resource identifier στο *Exported Function-Call Scheduler* block. Το OSEK Resource block δεν έχει επιπτώσεις στην προσομοίωση και ισχύει μόνο για την παραγωγή κώδικα.



Παράδειγμα του OSEK Resource block και παραγόμενου κώδικα (nxtway_conrol.mdl).

3.22 XY Map Graph block

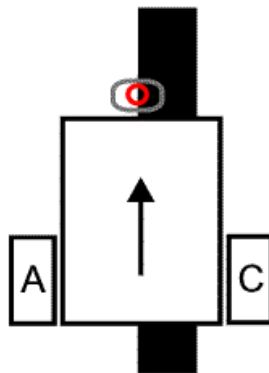


Το XY Map Graph block αποθηκεύεται στις χρησιμότητες (Utilities) για προσομοίωση και χρησιμεύει στο να φαίνεται το ίχνος του ρομπότ στην εικονική διαδρομή κατά την διάρκεια της προσομοίωσης. Η XY Map Graph βασίζεται στο καθιερωμένο Simulink block το οποίο καλείται XY γραφική παράσταση(XY Graph).

ΚΕΦΑΛΑΙΟ 4 : Ο ΕΛΕΓΚΤΗΣ PID (PID CONTROLLER)

Ο PID ελεγκτής είναι μια συνηθισμένη τεχνική που χρησιμοποιείται για τον έλεγχο ευρείας ποικιλίας μηχανημάτων συμπεριλαμβανομένων οχημάτων, ρομπότ, ακόμα και πυραύλων. Η πλήρης μαθηματική περιγραφή του PID ελεγκτή είναι αρκετά πολύπλοκη όμως μια πολύ απλούστερη κατανόηση είναι πραγματικά αυτό που απαιτείται για να χρησιμοποιηθεί ένας PID ελεγκτής αποτελεσματικά.

Παρακάτω περιγράφεται η δημιουργία ενός PID για την ακολουθία γραμμών από ένα NXT. Μόλις δημιουργηθεί, το ίδιο PID μπορεί να χρησιμοποιηθεί, με μόνο ελάχιστες τροποποιήσεις, με οποιαδήποτε άλλη εφαρμογή Mindstorms όπως σε ένα ρομπότ που μπορεί να οδηγηθεί όσο το δυνατόν πιο ευθεία, ή ακόμα και για ένα ρομπότ που μπορεί να αυτοισορροπήσει σε 2 ρόδες όπως το `nxtway_gs`.



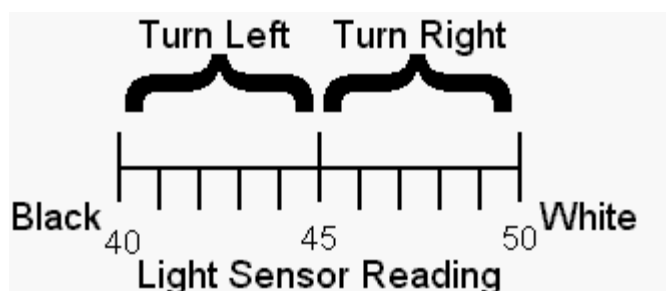
Στο παραπάνω σχήμα, φαίνεται το βασικό σχεδιάγραμμα ενός ρομπότ κατάλληλου για ακολουθία γραμμών. Το ρομπότ αποτελείται από δύο κινητήρες (Servo Motors), καθένας από τους οποίους συνδέεται με τους τροχούς A και C. Το ρομπότ έχει ένα αισθητήρα φωτός (Light Sensor) τοποθετημένο στο μπροστινό μέρος και δείχνει προς τα κάτω ώστε να «βλέπει» μόνο το έδαφος (ή οτιδήποτε πάνω στο οποίο το ρομπότ είναι τοποθετημένο). Ο κόκκινος κύκλος αντιπροσωπεύει το σημείο στο έδαφος που μπορεί να «δει» ο αισθητήρας. Το υπόλοιπο μέρος του ρομπότ αναπαρίσταται από το ορθογώνιο και το βέλος μέσα σ' αυτό δείχνει την κανονική κατεύθυνση κίνησης.

Στόχος του ρομπότ είναι να ακολουθεί μια παχιά μαύρη γραμμή. Αυτό μπορεί να γίνει χρησιμοποιώντας έναν ή περισσότερους αισθητήρες φωτός. Όσο περισσότερους αισθητήρες διαθέτει το ρομπότ, τόσο πιο γρήγορα κινείται καθώς ακολουθεί τη γραμμή. Στη συγκεκριμένη περίπτωση θα χρησιμοποιηθεί ένας αισθητήρας φωτός.

Η στρατηγική που θα ακολουθηθεί είναι ότι το ρομπότ δε θα ακολουθεί ακριβώς την γραμμή αλλά την άκρη της γραμμής. Αυτό επιλέχθηκε γιατί όταν το ρομπότ ξεφύγει από την γραμμή που ακολουθεί, ο αισθητήρας «βλέπει άσπρο» και δεν καταλαβαίνει σε ποια πλευρά της γραμμής βρίσκεται. Αντίθετα, όταν ακολουθεί την άκρη της γραμμής τότε γνωρίζει σε ποια κατεύθυνση βρίσκεται όταν έχει ξεφύγει από τη γραμμή. Αν ο αισθητήρας φωτός «βλέπει άσπρο», το ρομπότ βρίσκεται αριστερά της ακμής γραμμής (και αριστερά της γραμμής) , ενώ αν «βλέπει μαύρο», το ρομπότ βρίσκεται δεξιά της ακμής της γραμμής

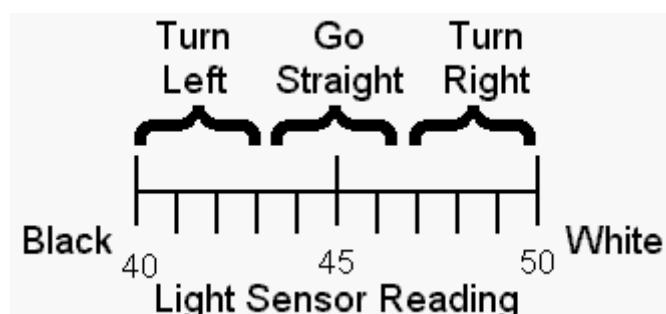
(και πάνω στη γραμμή). Αυτό καλείται «αριστερής πλευράς ακολουθία γραμμής» ("*left hand line follower*") αφού ακολουθεί την αριστερή ακμή της γραμμής.

Αρχικά, είναι αναγκαία η γνώση των τιμών που επιστρέφει ο αισθητήρας φωτός όταν «βλέπει άσπρο» και όταν «βλέπει μαύρο». Ένας τυπικός αισθητήρας δίνει τιμή 50 για το άσπρο και 40 για το μαύρο (σε κλίμακα 0 έως 100). Στο παρακάτω σχήμα φαίνονται οι τιμές του αισθητήρα σε σχέση με την κίνηση του ρομπότ.



Το εύρος διαιρέθηκε σε δύο ίσα τμήματα. Αν η τιμή του αισθητήρα φωτός είναι μεγαλύτερη από 45 τότε το ρομπότ θα στρίψει δεξιά, ενώ αν είναι μικρότερη από 45 θα στρίψει αριστερά. Για γραμμή με καμπύλες, μια απαλή στροφή του ρομπότ θα χρειαστεί το 50% της ισχύος του κινητήρα για τον γρήγορο τροχό και το 20% για τον αργό τροχό, ενώ για μια πιο απότομη στροφή θα χρειαστεί το 30% της ισχύος του κινητήρα για τον γρήγορο τροχό και φρένο για τον αργό τροχό.

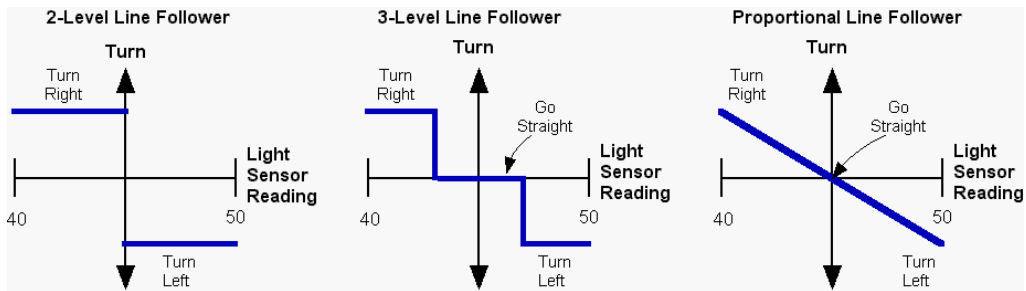
Η παραπάνω προσέγγιση δεν είναι τόσο γρήγορη και ακριβής. Αυτό διορθώνεται διαιρώντας το παραπάνω εύρος σε τρία τμήματα, όπως φαίνεται παρακάτω:



Σε αυτή την περίπτωση, αν το επίπεδο φωτός είναι μικρότερο από 43 το ρομπότ θα στρίψει αριστερά, αν το επίπεδο φωτός είναι ανάμεσα από 44 και 47 θα κινηθεί ευθεία και αν είναι μεγαλύτερο από 47 θα στρίψει δεξιά.

4.1 Ο όρος “P” στον PID (The “P” in “PID”: Proportion(al))

Όπως αναφέρθηκε παραπάνω, στην πρώτη προσέγγιση το ρομπότ μπορούσε να κάνει δύο πράγματα: να στρίψει δεξιά ή αριστερά. Στη δεύτερη προσέγγιση προστέθηκε και η κατεύθυνση «Ευθεία κίνηση». Στην περίπτωση που το εύρος διαιρεθεί σε περισσότερα από τρία τμήματα, χρειάζονται περισσότερα «είδη» στροφών. Παρακάτω παρουσιάζονται διαγράμματα για αυτές τις προσεγγίσεις. Στον x-άξονα είναι οι τιμές που «διαβάζει» ο αισθητήρας φωτός και στον y-άξονα το είδος στροφής.

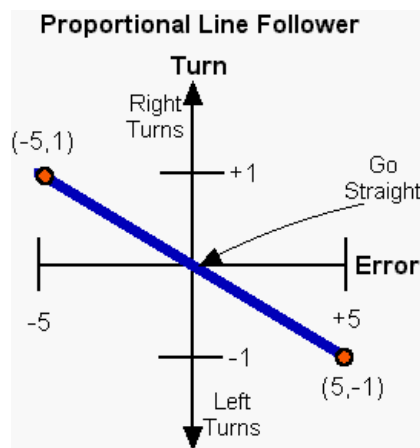


Στο αριστερό διάγραμμα φαίνεται η προσέγγιση δύο επιπέδων ακολουθίας γραμμής. Το ρομπότ, όπως αναφέρθηκε παραπάνω, μπορεί να κάνει μόνο δύο πράγματα (φαίνονται με τις μπλε γραμμές), να στρίψει δεξιά ή αριστερά. Στο κεντρικό διάγραμμα είναι η τριών επιπέδων ακολουθία γραμμής, όπου έχει προστεθεί η ευθεία κίνηση (Turn=0). Τέλος, στο δεξί διάγραμμα είναι η αναλογική (**Proportional**) ακολουθία γραμμής. Σε αυτή την περίπτωση η τιμή ποικίλει ομαλά ανάμεσα σε δύο όρια. Αν η τιμή του αισθητήρα φωτός δείχνει ότι το ρομπότ βρίσκεται κοντά στη γραμμή τότε κάνει μια μικρή στροφή. Αν δείχνει ότι είναι μακριά από τη γραμμή, κάνει μια μεγάλη στροφή. Το Proportional σημαίνει ότι υπάρχει γραμμική σχέση μεταξύ δύο μεταβλητών, δηλαδή το διάγραμμα των μεταβλητών αυτών παράγει μια ευθεία γραμμή.

Η εξίσωση της ευθείας γραμμής είναι:

$y = mx + b$, όπου y είναι η απόσταση στον y -άξονα, x η απόσταση στον x -άξονα, m είναι η κλίση της γραμμής και b είναι το σημείο που η γραμμή περνάει από τον y -άξονα όταν το x είναι μηδέν. Η κλίση ορίζεται ως η αλλαγή της τιμής του y προς την αλλαγή της τιμής του x χρησιμοποιώντας οποιοδήποτε ζεύγος σημείων της γραμμής.

Αρχικά, το κέντρο της γραμμής μετατοπίζεται στο μηδέν (x -άξονα). Για το εύρος των τιμών 40 και 50 αφαιρείται το 45 (που είναι ο μέσος όρος του 40 και 50, $(40+50)/2$) από κάθε ανάγνωση της τιμής του αισθητήρα φωτός. Αυτό το αποτέλεσμα καλείται σφάλμα (**error**). Έτσι, αν η τιμή φωτός είναι 47, μετά από αφαίρεση του 45 προκύπτει σφάλμα ίσο με 2 ($\text{error}=2$). Το σφάλμα δείχνει το πόσο μακριά βρίσκεται το ρομπότ από την ακμή της γραμμής. Αν ο αισθητήρας φωτός βρίσκεται ακριβώς πάνω στην ακμή της γραμμής, το σφάλμα είναι μηδέν ($\text{error}=0$). Αν ο αισθητήρας βρίσκεται μόνο σε άσπρο το σφάλμα είναι +5 ($\text{error}=5$), ενώ αν βρίσκεται μέσα στο μαύρο το σφάλμα είναι -5 ($\text{error}=-5$).



Στο παραπάνω διάγραμμα ο x-άξονας δείχνει το σφάλμα. Επειδή η γραμμή περνάει από το 0, το b είναι επίσης μηδέν και η εξίσωση της γραμμής γίνεται:

$$y=mx \text{ ή } \mathbf{Turn} = m \cdot \mathbf{error} \text{ (όπως φαίνεται στο διάγραμμα).}$$

Ο άξονας y (ή Turn) έχει εύρος από -1(απότομη στροφή αριστερά) έως 1(απότομη στροφή δεξιά) και το 0 σημαίνει ευθεία κίνηση. Η κλίση μπορεί να υπολογιστεί από τα δύο σημεία που έχουν σημειωθεί στο παραπάνω διάγραμμα:

$$\text{Slope} = m = (\text{μεταβολή του } y) / (\text{μεταβολή του } x) = (1 - (-1)) / (-5 - 5) = 2/10 = 0.2$$

Η κλίση (slope) είναι μια αναλογική σταθερά (**proportionality constant**) και είναι ο παράγοντας με τον οποίο πολλαπλασιάζεται το σφάλμα (error) για να μετατραπεί σε τιμή y (Turn). Η κλίση μπορεί επίσης να συμβολιστεί και με K, οπότε η εξίσωση γράφεται:

$$\mathbf{Turn} = \mathbf{K} \cdot (\mathbf{error})$$

Η τιμή της στροφής **Turn** είναι η έξοδος του P ελεγκτή και καλείται «όρος P» (**P term**).

Εκτός του εύρους -5 έως +5 δεν υπάρχει γνώση του πόσο μακριά βρίσκεται ο αισθητήρας από τη γραμμή. Όλα τα «άσπρα» φαίνονται ίδια καθώς ο αισθητήρας δεν μπορεί να «δει» καθόλου μαύρο. Μόλις ο αισθητήρας απομακρυνθεί αρκετά από την ακμή της γραμμής δίνει μια σταθερή τιμή που σημαίνει ότι δεν είναι πια αναλογικός(proportional) στο σφάλμα. Μόνο όταν ο αισθητήρας είναι σχετικά κοντά στη γραμμή μπορεί να υπάρχει γνώση του πόσο μακριά είναι από αυτή. Γι' αυτό ο αισθητήρας είναι αναλογικός (proportional) στην απόσταση. Εκτός του εύρους, λοιπόν, δίνεται η σωστή κατεύθυνση αλλά λάθος μέγεθος πλάτους (magnitude size). Το σφάλμα είναι μικρότερο απ' ότι θα έπρεπε και δίνει λάθος προσέγγιση για τη στροφή ώστε να διορθωθεί το σφάλμα.

Το εύρος στο οποίο ο αισθητήρας δίνει αναλογική (proportional) απόκριση (response) καλείται αναλογικό εύρος (**proportional range**). Το αναλογικό εύρος για τον αισθητήρα φωτός είναι 40 έως 50, για το σφάλμα -5 έως +5 και για τους κινητήρες -100 (κίνηση προς τα πίσω με μέγιστη ισχύ) έως +100 (κίνηση προς τα μπρος με μέγιστη ισχύ).

Η Σημαντικότητα του αναλογικού εύρους (proportional range):

(1) Το αναλογικό εύρος (proportional range) πρέπει να είναι όσο το δυνατόν πιο ευρύ και αυτό εξαρτάται από το ύψος τοποθέτησης του αισθητήρα φωτός. Όμως, μια πολύ υψηλή τοποθέτηση του αισθητήρα αντιδρά πιο πολύ στο φως του δωματίου και είναι λιγότερο αποτελεσματικός στη διάκριση των χρωμάτων.

(2) Εκτός του αναλογικού εύρους (proportional range) ο ελεγκτής θα ακολουθεί την σωστή κατεύθυνση αλλά θα μειονεκτεί στη διόρθωση του σφάλματος.

4.1.1 Τα πραγματικά επίπεδα ισχύος του κινητήρα από το P

Ένας τρόπος για τον υπολογισμό των στροφών είναι ο ορισμός του Tr ("Target power level"), το οποίο είναι η ισχύς και των δύο κινητήρων όταν το ρομπότ κινείται ευθεία (error=0). Όταν το σφάλμα δεν είναι μηδέν χρησιμοποιείται η εξίσωση **Turn = K*(error)**

για τον υπολογισμό της νέας ισχύος των κινητήρων. Ο ένας κινητήρας (έστω ο αριστερός – Α) θα έχει ισχύ **Tr+Turn** και ο άλλος (έστω ο δεξιός – C) **Tr-Turn**. Αν το **error** είναι θετικό τότε το **Turn** είναι θετικό και το **Tr+Turn** είναι μεγαλύτερο του **Tr** και ο αριστερός τροχός επιταχύνει ενώ ο δεξιός επιβραδύνει. Αν το **error** είναι αρνητικό τότε το **Tr-Turn** είναι μεγαλύτερο του **Tr** και ο δεξιός τροχός επιταχύνει ενώ ο αριστερός επιβραδύνει.

4.1.2 Ο ψευδοκώδικας για τον P Ελεγκτή

Αρχικά, χρειάζονται οι τιμές που επιστρέφει ο αισθητήρας φωτός για το άσπρο και το μαύρο. Από αυτές θα υπολογιστεί το **offset**, το οποίο είναι η τιμή που πρέπει να αφαιρεθεί από την τιμή που διαβάζει ο αισθητήρας ώστε να μετατραπεί σε τιμή σφάλματος. Το **offset** είναι ο μέσος όρος των τιμών του άσπρου και του μαύρου.

Επίσης, θα χρειαστεί μια τοποθεσία αποθήκευσης της σταθεράς **K**, έστω η **Kp**. Η αρχική τιμή της **Kp** μπορεί να υπολογιστεί είτε με τη μέθοδο δοκιμής σφάλματος είτε με βάση τα χαρακτηριστικά του αισθητήρα και του ρομπότ. Ως **Tr** (target power) ορίζεται η τιμή 50, δηλαδή όταν το σφάλμα είναι 0 και οι δύο κινητήρες θα έχουν ισχύ 50. Όταν το σφάλμα μεταβάλλεται από 0 σε -5, η ισχύς θα μεταβάλλεται από 50 σε 0. Αυτό σημαίνει ότι το **Kp** είναι :

$$Kp = (0 - 50)/(-5 - 0) = 10.$$

Επομένως, θα χρησιμοποιηθεί η τιμή **Kp** = 10 για τη μετατροπή του σφάλματος σε τιμή για τη στροφή (**turn**).

Ο ψευδοκώδικας παρουσιάζεται παρακάτω :

```

Kp = 10                                ! Αρχικοποίηση Μεταβλητών
offset = 45
Tr = 50

Loop                                     forever
  LightValue = read light sensor         ! Τρέχουσα Τιμή Αισθητήρα φωτός
  error = LightValue - offset             ! Υπολογισμός σφάλματος
                                          αφαιρώντας το offset
  Turn = Kp * error                       ! Ο "όρος P", επιθυμητή αλλαγή
                                          στην ισχύ των κινητήρων
  powerA = Tr + Turn                      ! Ισχύς κινητήρα Α
  powerC = Tr - Turn                      ! Ισχύς κινητήρα C
  MOTOR A direction=forward power=powerA ! προώθηση εντολής με τη
                                          νέα ισχύ σε ένα MOTOR
                                          block
  MOTOR C direction=forward power=powerC ! ομοίως για τον άλλο
                                          κινητήρα αλλά με την
                                          άλλη ισχύ
end loop forever                         ! τέλος loop, εκτέλεση από την
                                          αρχή

```

Αν το ρομπότ αποφεύγει τη γραμμή αντί να προσπαθεί να τη βρει, πρέπει να γίνει αλλαγή στις κατευθύνσεις.

Συγκεκριμένα, πρέπει:

powerA = Tp - Turn
powerC = Tp + Turn

Η τιμή **Kp** και η ισχύς **Tp** είναι ρυθμιζόμενες παράμετροι, δηλαδή υπολογίζονται με τη μέθοδο δοκιμής σφάλματος. Το **Kp** ελέγχει πόσο γρήγορα ο ελεγκτής προσπαθεί να επιστρέψει στην ακμή της γραμμής όταν έχει απομακρυνθεί από αυτή και το **Tp** ελέγχει πόσο γρήγορα το ρομπότ κινείται κατά μήκος της γραμμής.

Αν η γραμμή είναι αρκετά ευθεία γίνεται χρήση μεγάλης τιμής **Tp** ώστε το ρομπότ να κινείται με μεγάλη ταχύτητα και μικρό **Kd** ώστε οι στροφές να είναι απαλές.

Αν η γραμμή έχει καμπύλες θα υπάρχει μια μέγιστη τιμή **Tp** που θα λειτουργεί ο αλγόριθμος. Αν το **Tp** είναι μεγαλύτερο από αυτή τη μέγιστη τιμή, ανεξάρτητα από την τιμή του **Kp**, το ρομπότ θα χάσει την γραμμή λόγω της κίνησης με μεγάλη ταχύτητα. Αν το **Tp** είναι πολύ μικρό τότε κάθε τιμή **Kp** θα λειτουργεί καθώς το ρομπότ θα κινείται πολύ αργά. Με αρκετά μικρή ταχύτητα πάντως, κάθε γραμμή μπορεί να ακολουθηθεί. Στόχος είναι το ρομπότ να κινείται όσο πιο γρήγορα γίνεται ενώ θα μπορεί να ακολουθεί και τη γραμμή.

Παραπάνω, χρησιμοποιήθηκε ως αρχική τιμή για το **Kp** το 10. Για το **Tp** μπορεί να χρησιμοποιηθεί πιο χαμηλή από την παραπάνω (τιμή 50), όπως 15. Αν το ρομπότ χάσει τη γραμμή επειδή κινείται γλιστρώντας χρειάζεται αύξηση του **Kp**. Αν χάσει τη γραμμή επειδή το ρομπότ είναι υπερκίνητο και κυνηγάει τη γραμμή μπρος και πίσω τότε χρειάζεται μείωση του **Kp**. Αν ακολουθεί τη γραμμή κανονικά μπορεί να γίνει αύξηση **Tp** για την εύρεση μεγαλύτερης ταχύτητας. Για κάθε νέο **Tp** χρειάζεται και νέα εκτίμηση του **Kp**, παρόλο που το **Kp** συνήθως δε μεταβάλλεται πολύ.

Είναι πιθανόν ο καλύτερος P ελεγκτής να είναι διαφορετικός για κάθε είδος γραμμής (πάχος γραμμής, απότομες καμπύλες κλπ.) και για διαφορετικά ρομπότ. Με άλλα λόγια, ο P ελεγκτής ρυθμίζεται για συγκεκριμένο είδος γραμμής και ρομπότ και δεν λειτουργεί απαραίτητα και σε άλλες γραμμές ή ρομπότ. Ο κώδικας θα λειτουργεί για άλλες εφαρμογές και ρομπότ, αλλά πρέπει να ρυθμιστούν οι παράμετροι **Kp**, **Tp** και **offset**.

4.1.3 Περίληψη του P Ελεγκτή

Για τον P ελεγκτή η γενική διαδικασία δημιουργίας είναι η εξής:

- Χρήση ενός αισθητήρα για μέτρηση αυτού που πρόκειται να ελεγχθεί.
- Μετατροπή αυτής της μέτρησης σε σφάλμα. Στη συγκεκριμένη περίπτωση αυτό έγινε με αφαίρεση του μέσου όρου μαύρου και άσπρου.
- Πολλαπλασιασμός του σφάλματος με μια σταθερά **Kp**. Το αποτέλεσμα είναι μια διόρθωση για το σύστημα. Στη συγκεκριμένη περίπτωση η διόρθωση εφαρμόζεται ως αύξηση/μείωση της ισχύος των κινητήρων.
- Το **Kp** υπολογίζεται με τη μέθοδο δοκιμής σφάλματος.

Γενικά, οι P ελεγκτές λειτουργούν πολύ καλά κάτω από κάποιες συνθήκες :

1. Ο αισθητήρας πρέπει να έχει μεγάλο εύρος (το οποίο δυστυχώς δεν ισχύει στον παρόν ελεγκτή).
2. Αυτό που πρέπει να ελεγχθεί (οι κινητήρες σε αυτή την περίπτωση) πρέπει επίσης να έχουν μεγάλο εύρος (αυτό ισχύει στους κινητήρες του NXT).
3. Και ο αισθητήρας και αυτό που πρέπει να ελεγχθεί (οι κινητήρες) πρέπει να ανταποκρίνονται γρήγορα. "Γρήγορα" σημαίνει να είναι ταχύτερα από οτιδήποτε άλλο συμβαίνει στο σύστημα.

4.2 Προσθέτοντας το "I" στον Ελεγκτή: Ο PI Ελεγκτής

Για τη βελτίωση της απόκρισης του P ελεγκτή θα προστεθεί ένας νέος όρος στην εξίσωση. Αυτός ο όρος καλείται **integral (ολοκληρωτής)**, το "I" στο PID. Το integral είναι το τρέχον άθροισμα του σφάλματος. Κάθε φορά που διαβάζεται η τιμή του αισθητήρα φωτός και υπολογίζεται το σφάλμα, θα προστίθεται το σφάλμα σε μια μεταβλητή με όνομα **integral** :

$$\text{integral} = \text{integral} + \text{error}$$

Στη συνέχεια, όπως και στον όρο P, πολλαπλασιάζεται το **integral** με μια σταθερά **Ki**, ώστε να προκύψει η διόρθωση. Έτσι η μεταβλητή **Turn** γίνεται:

$$\text{Turn} = Kp * (\text{error}) + Ki * (\text{integral})$$

Αυτή είναι η βασική εξίσωση του PI ελεγκτή. Το **Turn** είναι η διόρθωση για τους κινητήρες. Ο proportional όρος είναι **Kp*(error)** και ο integral όρος είναι **Ki*(integral)**.

Το integral κάνει δύο πράγματα :

1. Αν το σφάλμα διατηρεί το ίδιο πρόσημο για αρκετούς βρόχους επανάληψης, το **integral** μεγαλώνει συνεχώς. Για παράδειγμα, αν το **error** είναι 1 και ελέγξουμε ξανά άλλες δύο φορές και το σφάλμα είναι και τις δύο φορές 2, το **integral** θα είναι $1+2+2=5$, αλλά το σφάλμα στο τελευταίο βήμα είναι 2. Το integral είναι παράγοντας διόρθωσης αλλά καθυστερεί λίγο να βρεθεί το σημείο το σημείο στο οποίο αρχίζει να συνεισφέρει.
2. Αφαιρεί μικρά σφάλματα. Αφού το **integral** προσθέτει τα σφάλματα, τα μικρά σφάλματα σταδιακά μεγαλώνουν ώστε να γίνονται αισθητά για την διόρθωση.
3. Ο όρος **integral** είναι η "μνήμη" του ελεγκτή. Το **integral** είναι το ιστορικό του σφάλματος και παρέχει στον ελεγκτή μια μέθοδο διόρθωσης σφαλμάτων που διαρκούν μεγάλο χρονικό διάστημα.

4.2.1 Ο Ψευδοκώδικας για τον ελεγκτή PI

Για τον PI ελεγκτή αρκεί να προστεθεί μια νέα μεταβλητή για το **Ki** και μια για το **integral**:

```
Kp          =          10          !          Kp=10
Ki          =          1          !          Ki=1
offset      =          45
Tp          =          50
integral    =    0          ! μέρος αποθήκευσης integral
Loop
  LightValue = read light sensor ! Τρέχουσα Τιμή Αισθητήρα φωτός
  error = LightValue - offset    ! Υπολογισμός σφάλματος
                                αφαιρώντας το offset
  integral = integral + error    ! ο νέος integral όρος
  Turn = Kp * error + Ki*integral ! Ο "όρος P" και "όρος I"
  powerA = Tp + Turn             ! Ισχύς κινητήρα A
  powerC = Tp - Turn             ! Ισχύς κινητήρα C
  MOTOR A direction=forward power=powerA ! προώθηση εντολής με τη
                                         νέα ισχύ σε ένα MOTOR
                                         block
  MOTOR C direction=forward power=powerC ! ομοίως για τον άλλο
                                         κινητήρα αλλά με την
                                         άλλη ισχύ
end loop forever                ! τέλος loop, εκτέλεση από την
                                αρχή
```

4.3 Προσθέτοντας το "D" στον Ελεγκτή: Ο ολοκληρωμένος PID ελεγκτής

Ο ελεγκτής σε αυτή την περίπτωση περιλαμβάνει και τον αναλογικό όρο "P"(Proportional), που προσπαθεί να διορθώσει το τρέχον σφάλμα και τον ολοκληρωτικό όρο "I" (Integral), που προσπαθεί να διορθώσει τα σφάλματα του παρελθόντος. Ένας τρόπος ώστε ο ελεγκτής να προβλέψει κάποιο μελλοντικό σφάλμα και να προσπαθήσει να το διορθώσει πριν καν αυτό συμβεί δίνεται από τα προηγμένα μαθηματικά και ονομάζεται παράγωγος (derivative). Αυτό είναι το "D"(derivative) στον PID. Όπως ο ολοκληρωτής έτσι και η παράγωγος αντιπροσωπεύει κάποια αρκετά σοβαρά μαθηματικά. Για την περίπτωση των αναγκών του παρόντος PID τα πράγματα είναι πιο απλά.

Υποθέτουμε ότι η επόμενη μελλοντική μεταβολή του **σφάλματος** θα είναι η ίδια με την τελευταία μεταβολή του **σφάλματος**.

Αυτό σημαίνει ότι το επόμενο σφάλμα αναμένεται να είναι το τρέχον σφάλμα συν τη μεταβολή του σφάλματος μεταξύ των δύο προηγούμενων δειγμάτων του αισθητήρα. Η αλλαγή του σφάλματος μεταξύ δύο διαδοχικών σημείων καλείται παράγωγος(derivative). Η παράγωγος ισούται με την κλίση της γραμμής.

Ένα παράδειγμα από ένα ζεύγος δείγματος δεδομένων θα μας βοηθήσει για το πώς λειτουργεί. Υποθέσουμε ότι το τρέχον σφάλμα είναι 2 και το σφάλμα που έγινε πριν ήταν 5. Η αλλαγή στο σφάλμα (error) είναι η παράγωγος η οποία ισούται με :

$$(\text{το τωρινό error}) - (\text{το προηγούμενο error})$$

Το οποίο στο συγκεκριμένο παράδειγμα είναι $2 - 5 = -3$. Άρα η τωρινή **παράγωγος** ισούται με -3. Χρησιμοποιώντας την παράγωγο για την πρόβλεψη του επόμενου σφάλματος (error) θα χρησιμοποιήσουμε :

$$(\text{next error}) = (\text{the current error}) + (\text{the current derivative})$$

$$(\text{το επόμενο error}) = (\text{τωρινό error}) + (\text{τωρινή παράγωγος})$$

Όπου για την περίπτωση των αριθμών του παραδείγματός μας είναι $2 + (-3) = -1$. Έτσι προβλέφτηκε ότι το επόμενο σφάλμα θα είναι -1. Στην πράξη, η πρόβλεψη του επόμενου σφάλματος δεν γίνεται κατά αυτόν τον τρόπο αλλά χρησιμοποιούμε μόνο την παράγωγο απευθείας στην εξίσωση του ελεγκτή.

Τώρα μπορεί να γραφτεί η ολοκληρωμένη εξίσωση για τον PID ελεγκτή.

$$\text{Turn} = K_p * (\text{error}) + K_i * (\text{integral}) + K_d * (\text{derivative})$$

$$\text{Στροφή} = K_p * (\text{σφάλμα}) + K_i * (\text{ολοκληρωτικό}) + K_d * (\text{παράγωγο})$$

Για το πόσο ακριβής είναι η πρόβλεψη και πώς μας βοηθάει περιγράφεται στην συνέχεια.

Εάν το τρέχον σφάλμα είναι χειρότερο από το προηγούμενο σφάλμα τότε ο όρος “D” προσπαθεί να διορθώσει το σφάλμα. Αν το τρέχον σφάλμα είναι καλύτερο από το προηγούμενο σφάλμα τότε ο όρος “D” προσπαθεί να σταματήσει τον ελεγκτή από τη διόρθωση του σφάλματος. Η δεύτερη περίπτωση που περιγράφεται στη συνέχεια είναι ιδιαίτερα χρήσιμη. Εάν το σφάλμα είναι να πάρει τιμές κοντά στο μηδέν, τότε προσεγγίζουμε το σημείο στο οποίο θέλουμε να σταματήσει η διόρθωση. Δεδομένου ότι το σύστημα θέλει κάποιο χρόνο για να ανταποκριθεί στην αλλαγή της ισχύς των κινητήρων «θέλουμε να αρχίσει ο περιορισμός της ισχύς του κινητήρα πριν το σφάλμα φτάσει στο μηδέν, διαφορετικά θα υπάρξει υπέρβαση των ορίων. Κατά αυτό τον τρόπο θα μπορούσε να υποτεθεί ότι η εξίσωση για τον όρο “D” θα έπρεπε να είναι πιο πολύπλοκη από ό, τι είναι, πράγμα το οποίο δεν ισχύει. Σημαντικό είναι να γίνει η αφαίρεση με τη σωστή σειρά. Η σωστή σειρά για τέτοιου είδους πράγματα είναι το “τρέχουσα” μείον το “προηγούμενο”. Έτσι για τον υπολογισμό της παραγώγου παίρνουμε το τρέχον σφάλμα και αφαιρούμε το προηγούμενο σφάλμα.

4.3.1 Ο ψευδοκώδικας για τον PID ελεγκτή

Για να προστεθεί ο όρος της παραγώγου για να τον ελεγκτή πρέπει να προστεθεί μια νέα μεταβλητή για το **Kd** και μία μεταβλητή να θυμόμαστε το τελευταίο σφάλμα.

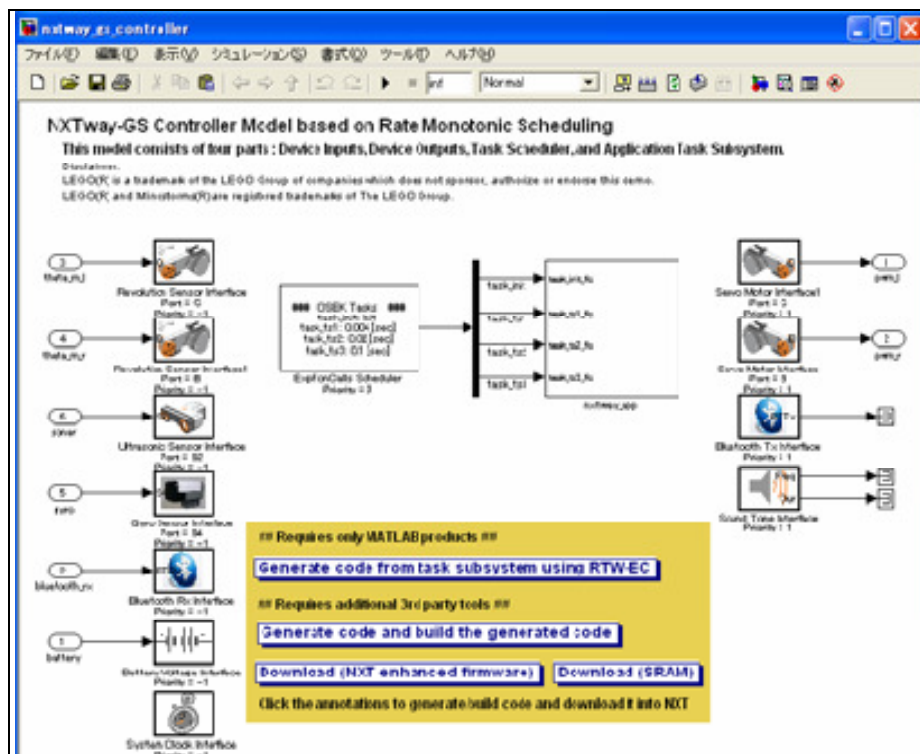
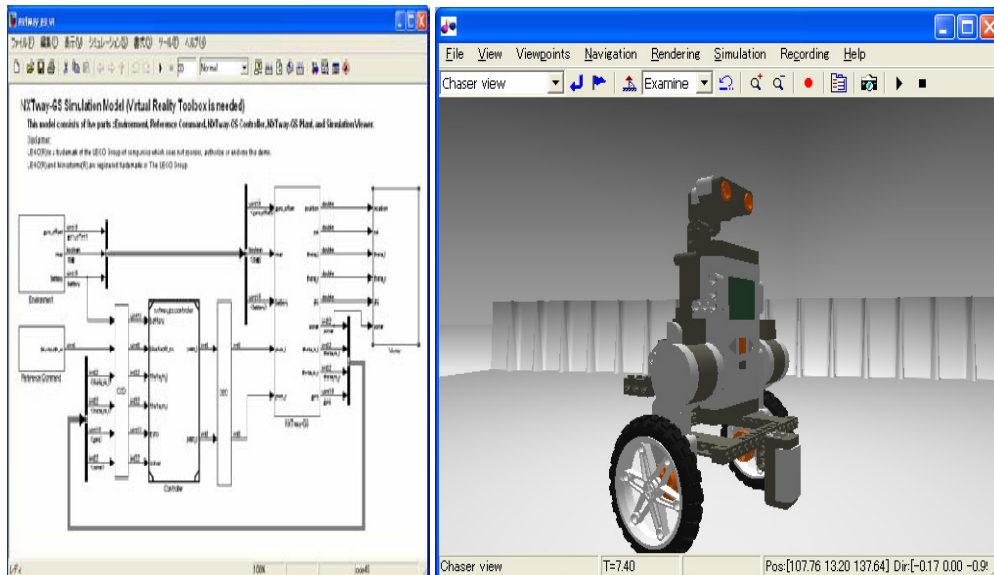

```

Kp          =          10          !          Kp=10
Ki = 1          ! Ki=1
Kd          =          100          !          Kd=100
offset      =          45
Tp          =          50
integral = 0          ! μέρος αποθήκευσης integral
lastError = 0          ! μέρος αποθήκευσης τελευταίου
                σφάλματος
derivative=0          ! μέρος αποθήκευσης derivative
Loop
  LightValue = read light sensor ! Τρέχουσα Τιμή Αισθητήρα φωτός
  error = LightValue - offset    ! Υπολογισμός σφάλματος
                                αφαιρώντας το offset
  integral = integral + error    ! ο νέος integral όρος
  derivative = error - lastError ! υπολογισμός derivative
  Turn = Kp * error + Ki*integral + Kd*derivative ! Ο "όρος P" και
                                                ο "όρος I" και
                                                ο "όρος D"
  powerA = Tp + Turn            ! Ισχύς κινητήρα A
  powerC = Tp - Turn            ! Ισχύς κινητήρα C
  MOTOR A direction=forward power=powerA ! προώθηση εντολής με τη
                                          νέα ισχύ σε ένα MOTOR
                                          block
  MOTOR C direction=forward power=powerC ! ομοίως για τον άλλο
                                          κινητήρα αλλά με την
                                          άλλη ισχύ
  lastError = error          ! αποθήκευση τρέχοντος
error
                                ώστε να είναι lastError την
                                επόμενη φορά
end loop forever              ! τέλος loop, εκτέλεση από την
                                αρχή

```

ΚΕΦΑΛΑΙΟ 5 : ΤΟ ΜΟΝΤΕΛΟ NXTway_GS

ΕΛΕΓΧΟΣ & ΚΑΤΑΣΚΕΥΗ ΑΥΤΟΪΣΟΡΡΟΠΟΥΜΕΝΟΥ ΡΟΜΠΟΤ ΔΥΟ ΤΡΟΧΩΝ ΜΕ ΧΡΗΣΗ ΠΛΑΤΦΟΡΜΑΑΣ LEGO MINDSTORMS NXT/MATLAB/SIMULINK



Εισαγωγή

Το NXTway GS είναι ένα αυτοϊσορροπούμενο ρομπότ δύο τροχών κατασκευασμένο από Lego Mindstorms NXT. Στην συνέχεια παρουσιάζεται η βασική σχεδίαση του μοντέλου σχετικά με την ισορροπία και ο οδικός έλεγχος με τη χρήση MATLAB / SIMULINK.

Στην συνέχεια παρουσιάζονται τα κύρια περιεχόμενα:

- Μαθηματικά μοντέλα
- Σχεδιασμός Ελέγχου
- Εικονική σχεδίαση μοντέλου
- Προσομοίωση και πειραματικά αποτελέσματα

Προετοιμασία

Για την κατασκευή του NXTway-GS διαβάστε τις σχεδιαστικές οδηγίες στο Παράρτημα.

Απαιτήσεις προϊόντων:

Προϊόν	Έκδοση	εμφάνιση
MATLAB [®]	7.5.0	R2007b
Control System Toolbox	8.0.1	R2007b
Simulink [®]	7.0	R2007b
Real-Time Workshop [®]	7.0	R2007b
Real-Time Workshop [®] Embedded Coder	5.0	R2007b
Fixed-Point Toolbox (N1)	2.1	R2007b
Simulink [®] Fixed Point (N1)	5.5	R2007b
Virtual Reality Toolbox (N2)	4.6	R2007b

Η προσομοίωση του μοντέλου NXTway-GS και η παραγωγή κώδικα μπορεί να πραγματοποιηθεί και χωρίς τα (N1), (N2).

(N1) : Απαιτείται για την εκτέλεση του αριθμού σταθερής υποδιαστολής (fixed-point number)(*) για τον έλεγχο του μοντέλου (nxtway_gs_controller_fixpt.mdl).

(N2) : Απαιτείται για την εκτέλεση τρισδιάστατης απεικόνισης (nxtway_gs_vr.mdl).

(*)αριθμός σταθερής υποδιαστολής: είναι ένας πραγματικός τύπος δεδομένων για αριθμούς που έχουν ένα σταθερό αριθμό ψηφίων μετά την υποδιαστολή. Η αναπαράσταση του αριθμού σταθερής υποδιαστολής μπορεί να συγκριθεί με την πιο πολύπλοκη (και πιο απαιτητική υπολογιστικά) αναπαράσταση αριθμού κινητής υποδιαστολής .

Απαιτούμενες εγκαταστάσεις Λογισμικού :

Λογισμικό	Έκδοση
Embedded Coder Robot NXT	3.14
nxtOSEK (previous name is LEJOS OSEK)	2.03
Cygwin	1.5.24
GNU ARM	4.0.2

Λίστα αρχείων:

Αρχείο	Περιγραφή
iswall.m	M-function for detecting wall in map
mywritevrtrack.m	M-function for generating map file (track.wrl)
nxtway_gs.mdl	NXTway-GS model (It does not require Virtual Reality Toolbox)
nxtway_gs_controller.mdl	NXTway-GS controller model (single precision floating-point)
nxtway_gs_controller_fixpt.mdl	NXTway-GS controller model (fixed-point)
nxtway_gs_plant.mdl	NXTway-GS plant model
nxtway_gs_vr.mdl	NXTway-GS model (It requires Virtual Reality Toolbox)
param_controller.m	M-script for controller parameters
param_controller_fixpt.m	M-script for fixed-point settings (Simulink.NumericType)
param_nxtway_gs.m	M-script for NXTway-GS parameters (It calls param_*.m)
param_plant.m	M-script for plant parameters
param_sim.m	M-script for simulation parameters
track.bmp	map image file
track.wrl	map VRML file
vrnxtwaytrack.wrl	map & NXTway-GS VRML file

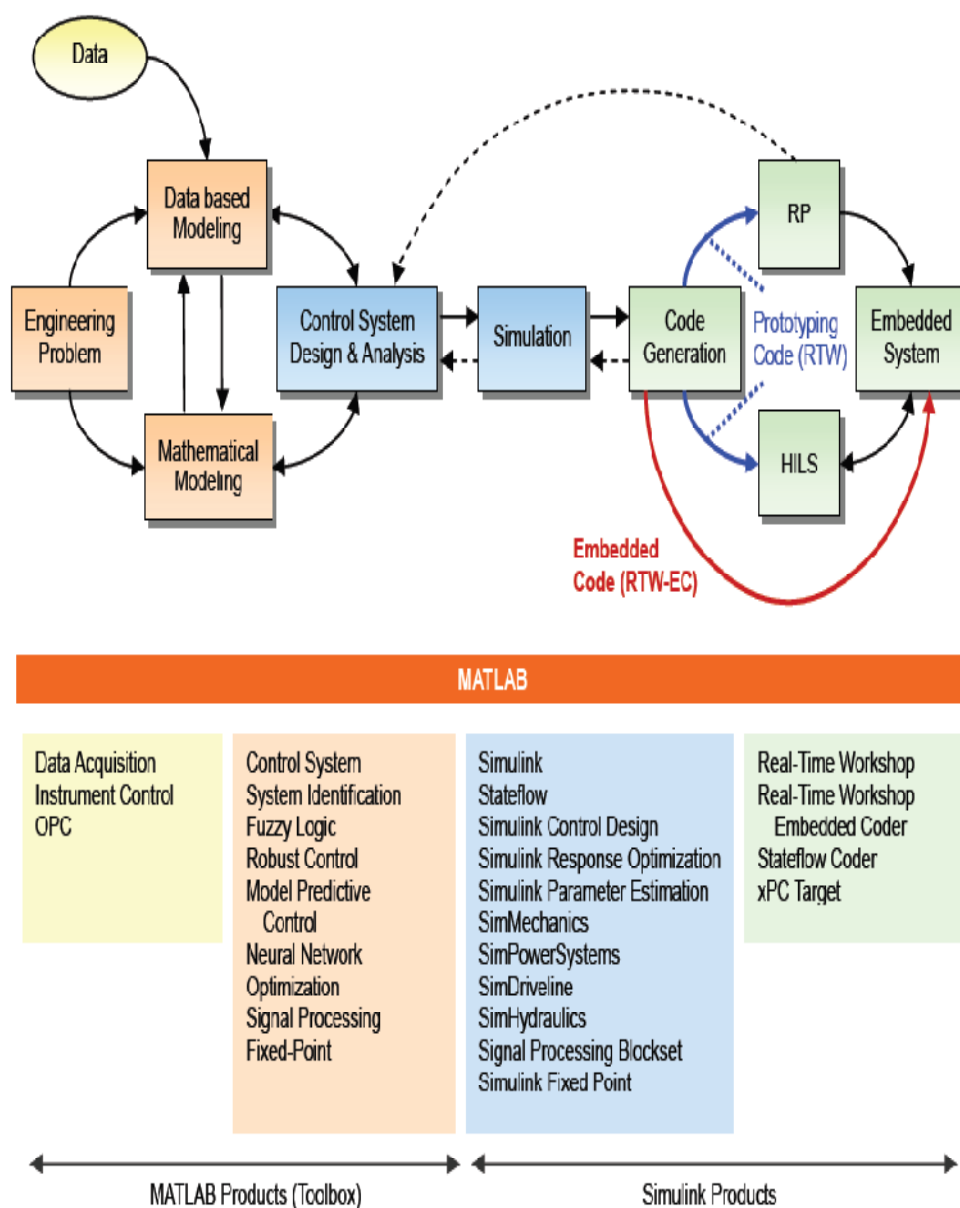
5.1 Σχέδιο Μοντελοποίησης

Αυτό το κεφάλαιο περιγράφει περιληπτικά την Βασική σχεδίαση του μοντέλου.

5.1.1 Η σχεδιαστική Μοντελοποίηση (MBD)

Η σχεδιαστική μοντελοποίηση είναι η τεχνική ανάπτυξης λογισμικού που χρησιμοποιεί μοντέλα προσομοίωσης. Γενικά για συντομογραφία αναφέρεται και ως MBD (Model-Based Design). Για τα συστήματα ελέγχου, ένας σχεδιαστής διαμορφώνει τις εγκαταστάσεις και τους ελεγκτές ή ένα μέρος τους, και εξετάζει τον αλγόριθμο ελεγκτών βασισμένο σε μια προσομοίωση Η/Υ ή σε πραγματικού χρόνου προσομοίωση. Η προσομοίωση πραγματικού χρόνου επιτρέπει τον έλεγχο και την επικύρωση του αλγορίθμου σε πραγματικό χρόνο, με τη χρησιμοποίηση του κώδικα που παράγεται από το μοντέλο. Είναι ένα γρήγορο πρωτότυπο [Rapid Prototyping (RP)] στο οποίο ένας ελεγκτής (controller) αντικαθίσταται από έναν πραγματικού χρόνου προσομοιωτή (real-time simulator) και το υλικό βρόχων προσομοίωσης [Hardware In the Loop Simulation (HILS)] είναι μια έκδοση του γρήγορου πρωτοτύπου. Επιπλέον, τα προϊόντα αυτόματης παραγωγής κώδικα όπως η RTW-EC επιτρέπουν την παραγωγή κώδικα C/C++ για τους ενσωματωμένους ελεγκτές (μικροεπεξεργαστής, DSP, κ.λπ.) από το πρότυπο ελεγκτών. Στο

σχήμα παρουσιάζεται η έννοια MBD για τα συστήματα ελέγχου βασισμένα στα προϊόντα της MATLAB.

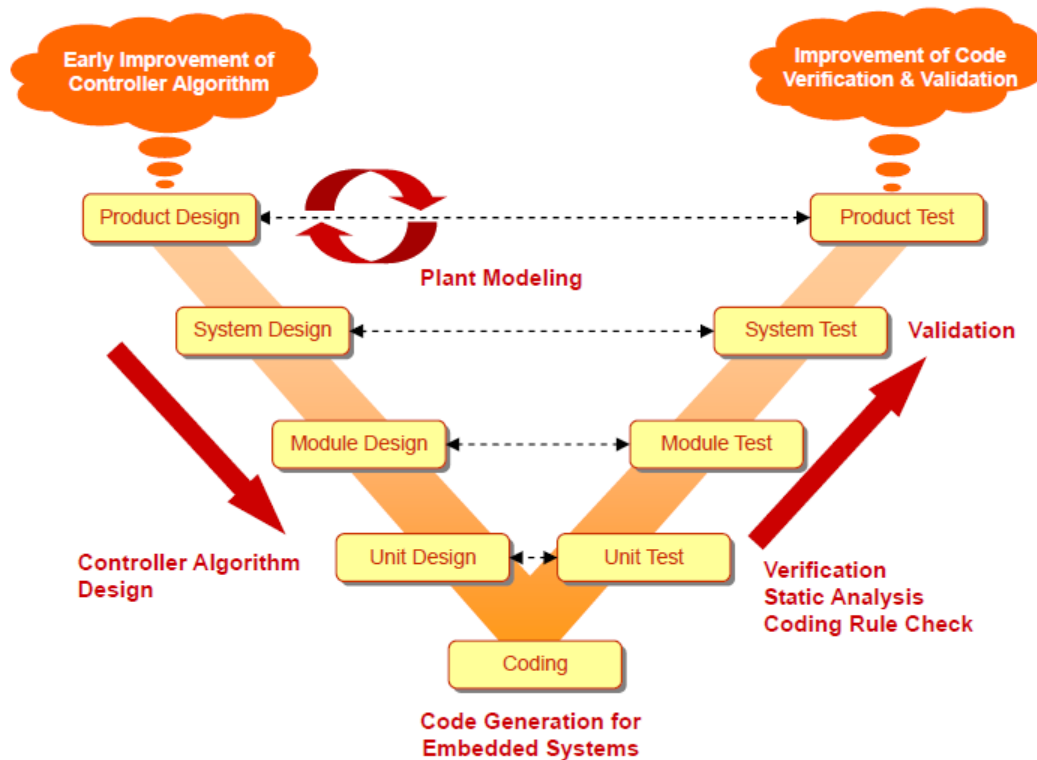


Σχήμα : MBD για τα συστήματα ελέγχου βασισμένα στα προϊόντα της MATLAB

5.1.2 V-Διαδικασία

Η V-διαδικασία που παρουσιάζεται στο σχήμα περιγράφει την MBD αναπτυξιακή διαδικασία της σχεδιαστικής μοντελοποίησης για τα συστήματα ελέγχου. Η V-διαδικασία αποτελείται από το σχεδιασμό, την κωδικοποίηση, και το στάδιο δοκιμής. Τα στάδια κάθε δοκιμής αντιστοιχούν στα κατάλληλα στάδια σχεδιασμού. Ο υπεύθυνος για την ανάπτυξη φτιάχνει τα πρότυπα εγκαταστάσεων/ελεγκτών στη αριστερή πλευρά της V-διαδικασίας για

την πρόωρη βελτίωση του αλγορίθμου του ελεγκτή, και τα επαναχρησιμοποιεί στη δεξιά πλευρά για τη βελτίωση της επαλήθευσης και της επικύρωσης του κώδικα.



Σχήμα: V-διαδικασία για συστήματα ελέγχου

5.1.3 Πλεονεκτήματα της σχεδιαστικής μοντελοποίησης (MBD)

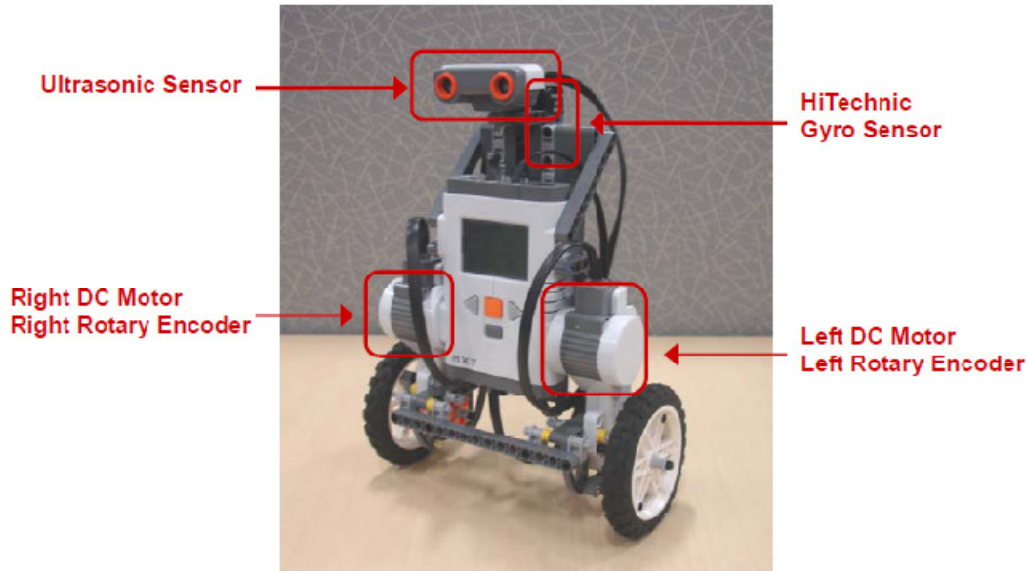
Η σχεδιαστική μοντελοποίηση (MBD) έχει τα ακόλουθα πλεονεκτήματα:

- Ανίχνευση για τα λάθη προδιαγραφών στα πρώτα στάδια της ανάπτυξης.
- Μείωση των πρωτοτύπων υλικού και ασφαλής επαλήθευση με προσομοίωση σε πραγματικό χρόνο.
- Αποδοτική δοκιμή από την επαλήθευση του μοντέλου.
- Αποτελεσματική επικοινωνία από την χρήση του μοντέλου
- Μείωση του χρόνου κωδικοποίησης και λαθών από αυτόματη παραγωγή κώδικα.

5.2 Το Σύστημα NXTway-GS

5.2.1 Δομή

Αυτό το κεφάλαιο περιγράφει τη δομή και τους αισθητήρες/ενεργοποιητές του NXTway-GS.



Σχήμα: NXTway-GS

5.2.2 Αισθητήρες/Ενεργοποιητές

Ο πίνακας 5.2.1 και 5.2.2 παρουσιάζουν τις ιδιότητες των αισθητήρων και των ενεργοποιητών.

Πίνακας 5.2.1 Ιδιότητες αισθητήρων

Αισθητήρας	Έξοδος	Μονάδα	Τύπος Δεδομένων	Μέγιστο Δείγμα [1/sec]
Περιστροφικός Κωδικοποιητής- (Rotary Encoder)	Γωνία	deg	int32	1000
Αισθητήρας Υπέρηχων- (Ultrasonic Sensor)	Απόσταση	cm	int32	50 (N1)
Γυροσκοπικός αισθητήρας- (Gyro Sensor)	Γωνιακή ταχύτητα	deg/sec	uint16	300

Πίνακας 5.2.2 Ιδιότητες ενεργοποιητών

Ενεργοποιητής	Είσοδος	Μονάδα	Τύπος Δεδομένων	Μέγιστο Δείγμα [1/sec]
DC Motor	PWM	%	int8	500

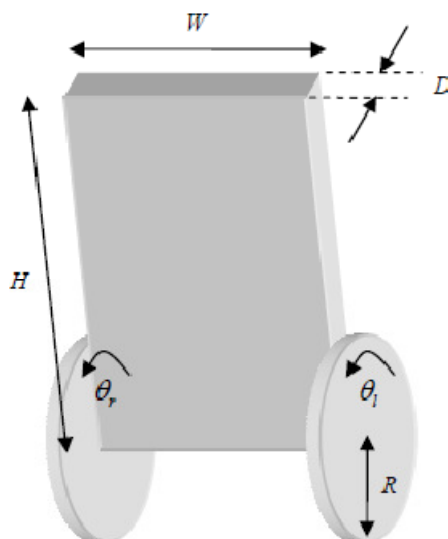
$N(1)$:Το μέγιστο ποσοστό δειγμάτων για την ακριβή απόσταση

Η αναφορά [1] επεξηγεί πολλές ιδιότητες για τον (DC) κινητήρα. Γενικά οι αισθητήρες και οι ενεργοποιητές διαφέρουν. Ειδικά, πρέπει να σημειωθεί ότι το offset του γυροσκοπικού αισθητήρα και η κλίση του ασκεί μεγάλη επίδραση στον έλεγχο της ισορροπίας. Το offset του γυροσκοπικού αισθητήρα είναι η έξοδος όταν δεν περιστρέφεται, και η κλίση του γυροσκοπικού αισθητήρα είναι η χρονική μεταβολή του offset.

5.3 Μοντελοποίηση του NXTway-GS

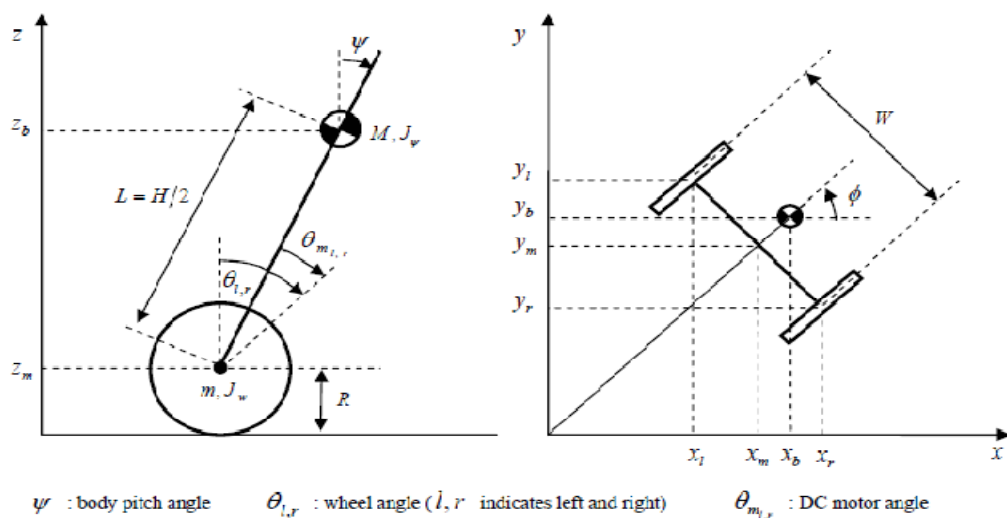
5.3.1 Μοντέλο δίτροχου εκκρεμούς

Αυτό το κεφάλαιο περιγράφει την μαθηματική μοντελοποίηση και τις εξισώσεις κίνησης του NXTway-GS.



Σχήμα 5.1 :Δίτροχο εκκρεμές

Το Σχήμα 5.2 παρουσιάζει την πλάγια όψη και την κάτοψη του δίτροχου εκκρεμούς. Το σύστημα συντεταγμένων που χρησιμοποιείτε στις εξισώσεις κίνησης 5.2 του δίτροχου εκκρεμούς περιγράφεται στο σχήμα 5.2



Σχήμα 5.2 : Πλάγια όψη και κάτοψη του δίτροχου εκκρεμούς.

Ακολουθούν οι φυσικές παράμετροι του NXTway-GS.

Δεδομένα	Μονάδα α	Ονομασία
$g=9.81$	[m/sec ²]	Επιτάχυνση βαρύτητας
$m=0.03$	[kg]	Βάρος τροχού
$R=0.04$	[m]	Ακτίνα τροχού
$J_w=mR^2/2$	[kgm ²]	Στιγμιαία αδράνεια τροχού
$M=0.6$	[kg]	Βάρος σώματος
$W=0.14$	[m]	Πλάτος σώματος
$D=0.04$	[m]	Πάχος σώματος
$H=0.144$	[m]	Υψος σώματος
$L=H/2$	[m]	Απόσταση του κέντρου βάρους από τον άξονα του τροχού
$J_\psi=M(W^2+D^2)/12$	[kgm ²]	Στιγμιαία αδράνεια εκτροπής από την πορεία του σώματος Κατά άξονα (z), Yaw=z
$J_m=1 \times 10^{-5}$	[kgm ²]	Στιγμιαία αδράνεια (DC) κινητήρα
$J_\psi=ML^2/3$	[kgm ²]	Στιγμιαία αδράνεια βήματος σώματος Κατά άξονα (y), Pitch=y
$R_m=6.69$	[Ω]	Αντίσταση κινητήρα
$K_b=0.468$	[V sec/rad]	Ηλεκτρεγερτική δύναμη
$K_t=0.317$	[Nm/A]	Σταθερή ροπή κινητήρα(DC)
$n=1$		Σχέση μετάδοσης (γραναζιού)
$f_m=0.0022$		Συντελεστής τριβής μεταξύ του σώματος και του κινητήρα(DC)
$f_w=0$		Συντελεστής τριβής μεταξύ της ρόδας και του εδάφους

- Οι τιμές που χρησιμοποιούνται περιγράφονται στην αναφορά [2] για τα R_m , K_b , K_t .
- Οι τιμές που χρησιμοποιούνται είναι προσεγγιστικά κατάλληλες για τα J_m , n , f_m , f_w λόγω της δυσκολίας τους στο να υπολογιστούν.

5.3.2 Εξισώσεις κίνησης του δίδροχου εκκρεμούς

Οι εξισώσεις κίνησης του δίδροχου εκκρεμούς μπορούν να παραχθούν τις με την μέθοδο Lagrange βασισμένη στο σύστημα συντεταγμένων στο σχήμα 5.2. Εάν η κατεύθυνση του δίδροχου εκκρεμούς είναι στο θετικό Χ-άξονα για $t=0$, κάθε μια από τις συντεταγμένες δίνεται όπως περιγράφεται από τις παρακάτω σχέσεις.

$$(\theta, \phi) = \left(\frac{1}{2}(\theta_i + \theta_r), \frac{R}{W}(\theta_r - \theta_i) \right) \quad (5.1)$$

$$(x_m, y_m, z_m) = \left(\int \dot{x}_m dt, \int \dot{y}_m dt, R \right), (\dot{x}_m, \dot{y}_m) = (R\dot{\theta} \cos \phi, R\dot{\theta} \sin \phi) \quad (5.2)$$

$$(x_l, y_l, z_l) = \left(x_m - \frac{W}{2} \sin \phi, y_m + \frac{W}{2} \cos \phi, z_m \right) \quad (5.3)$$

$$(x_r, y_r, z_r) = \left(x_m + \frac{W}{2} \sin \phi, y_m - \frac{W}{2} \cos \phi, z_m \right) \quad (5.4)$$

$$(x_b, y_b, z_b) = (x_m + L \sin \psi \cos \phi, y_m + L \sin \psi \sin \phi, z_m + L \cos \psi) \quad (5.5)$$

Η Κινητική ενέργεια λόγω μεταφοράς T_1 , η Κινητική ενέργεια λόγω περιστροφής T_2 και η Δυναμική ενέργεια U είναι:

$$T_1 = \frac{1}{2} m (\dot{x}_l^2 + \dot{y}_l^2 + \dot{z}_l^2) + \frac{1}{2} m (\dot{x}_r^2 + \dot{y}_r^2 + \dot{z}_r^2) + \frac{1}{2} M (\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2) \quad (5.6)$$

$$T_2 = \frac{1}{2} J_w \dot{\theta}_l^2 + \frac{1}{2} J_w \dot{\theta}_r^2 + \frac{1}{2} J_\psi \dot{\psi}^2 + \frac{1}{2} J_\phi \dot{\phi}^2 + \frac{1}{2} n^2 J_m (\dot{\theta}_l - \dot{\psi})^2 + \frac{1}{2} n^2 J_m (\dot{\theta}_r - \dot{\psi})^2 \quad (5.7)$$

$$U = mgz_l + mgz_r + Mgz_b \quad (5.8)$$

Ο πέμπτος και έκτος όρος στην T_2 οφείλονται στην κινητική ενέργειας λόγω περιστροφής του περιβλήματος αριστερά και δεξιά από τον (DC) συνεχή κινητήρα.

Η Λανγκρανσιανή εξίσωση L εκφράζεται ως εξής :

$$L = T_1 + T_2 - U \quad (5.9)$$

Χρησιμοποιούμε τις ακόλουθες μεταβλητές ως γενικευμένες συντεταγμένες.

Όπου:

θ : η μέση τιμή της γωνίας από τον αριστερό και τον δεξιό τροχό

ψ : γωνία βήματος του σώματος κατά άξονα y .

ϕ : γωνία εκτροπής του σώματος από την πορεία κατά άξονα z .

Σημείωση*

- Ο προσανατολισμός ενός συστήματος $\{B\}$ ως προς ένα σύστημα $\{A\}$ μπορεί να περιγραφεί και ως γνόμενο τριών περιστροφών κατά άξονα.
- Μια τέτοια περιγραφή είναι η Roll-Pitch-Yaw (Roll= x , Pitch= y , Yaw= z):
- ${}^A R_B = \text{Rot}(z_A, \alpha) \text{Rot}(y_A, \beta) \text{Rot}(x_A, \gamma)$
- Αντίστοιχες περιγραφές (π.χ γωνίες Euler) μπορούν να προκύψουν με κατάλληλη επιλογή των πινάκων περιστροφής και της σειράς τους στον πολλαπλασιασμό.

Στην συνέχεια ακολουθούν εξισώσεις Λαγκρανζ των γωνιών θ , ψ , ϕ αντίστοιχα:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = F_\theta \quad (5.10)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \frac{\partial L}{\partial \psi} = F_\psi \quad (5.11)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \phi} = F_\phi \quad (5.12)$$

Παραγωγίζοντας τις εξισώσεις (5.10)-(5.12) προκύπτουν οι εξισώσεις:

$$\left[(2m+M)R^2 + 2J_w + 2n^2 J_m \right] \ddot{\theta} + (MLR \cos \psi - 2n^2 J_m) \ddot{\psi} - MLR \dot{\psi}^2 \sin \psi = F_\theta \quad (5.13)$$

$$(MLR \cos \psi - 2n^2 J_m) \ddot{\theta} + (ML^2 + J_\psi + 2n^2 J_m) \ddot{\psi} - MgL \sin \psi - ML^2 \dot{\phi}^2 \sin \psi \cos \psi = F_\psi \quad (5.14)$$

$$\left[\frac{1}{2} m W^2 + J_\phi + \frac{W^2}{2R^2} (J_w + n^2 J_m) + ML^2 \sin^2 \psi \right] \ddot{\phi} + 2ML^2 \dot{\psi} \dot{\phi} \sin \psi \cos \psi = F_\phi \quad (5.15)$$

Λαμβάνοντας υπόψη τη ροπή των (DC) κινητήρων και την ιξώδη τριβή, οι γενικευμένες δυνάμεις δίνονται ως εξής:

$$(F_\theta, F_\psi, F_\phi) = \left(F_l + F_r, F_\psi, \frac{W}{2R}(F_r - F_l) \right) \quad (5.16)$$

$$F_l = nK_t i_l + f_m(\dot{\psi} - \dot{\theta}_l) - f_w \dot{\theta}_l \quad (5.17)$$

$$F_r = nK_t i_r + f_m(\dot{\psi} - \dot{\theta}_r) - f_w \dot{\theta}_r \quad (5.18)$$

$$F_\psi = -nK_t i_l - nK_t i_r - f_m(\dot{\psi} - \dot{\theta}_l) - f_m(\dot{\psi} - \dot{\theta}_r) \quad (5.19)$$

Όπου $i_{l,r}$ είναι το ρεύμα των (DC motor) κινητήρων.

Το ρεύμα των κινητήρων (DC motor) δεν μπορεί να χρησιμοποιηθεί προκειμένου να το ελέγξουμε άμεσα, επειδή είναι βασισμένο στον έλεγχο PWM(*) (τάσης). Επομένως, αξιολογούμε τη σχέση μεταξύ του ρεύματος $i_{l,r}$ και της τάσης $v_{l,r}$ χρησιμοποιώντας την εξίσωση κινητήρα (DC motor). Εάν θεωρήσουμε ότι η τριβή εντός του κινητήρα είναι αμελητέα η εξίσωση του κινητήρα συνεπάγεται ως εξής:

Σημείωση(*):

Pulse-width modulation (PWM): Η Διαμόρφωση παλμού-πλάτους είναι ένας πολύ αποτελεσματικός τρόπος παροχής τμηματικών ποσοτήτων ηλεκτρικής ενέργειας μεταξύ on/off κυκλώματος. Ένας απλός διακόπτης ισχύος με κανονική πυγή ενέργειας παρέχει πλήρη ισχύ μόνο όταν ο διακόπτης είναι ανοιχτός (on).

$$L_m \dot{i}_{l,r} = v_{l,r} + K_b(\dot{\psi} - \dot{\theta}_{l,r}) - R_m i_{l,r} \quad (5.20)$$

Αν θεωρήσουμε ότι η επαγωγική αντίσταση του κινητήρα είναι αμελητέα τότε συνεπάγεται ότι το ρεύμα είναι:

$$i_{l,r} = \frac{v_{l,r} + K_b(\dot{\psi} - \dot{\theta}_{l,r})}{R_m} \quad (5.21)$$

Από την εξίσωση (5.21), η γενικευμένη δύναμη μπορεί να εκφραστεί χρησιμοποιώντας την τάση των κινητήρων.

$$F_{\theta} = \alpha(v_i + v_r) - 2(\beta + f_w)\dot{\theta} + 2\beta\dot{\psi} \quad (5.22)$$

$$F_{\psi} = -\alpha(v_i + v_r) + 2\beta\dot{\theta} - 2\beta\dot{\psi} \quad (5.23)$$

$$F_{\phi} = \frac{W}{2R} \alpha(v_r - v_i) - \frac{W^2}{2R^2} (\beta + f_w)\dot{\phi} \quad (5.24)$$

$$\alpha = \frac{nK_t}{R_m}, \quad \beta = \frac{nK_t K_{\phi}}{R_m} + f_m \quad (5.25)$$

5.3.3 Καταστατικές εξισώσεις δίτροχου εγκιρμους

Στην συνέχεια ακολουθεί η παραγωγή των καταστατικών εξισώσεων βασισμένη στη σύγχρονη θεωρία ελέγχου με τις γραμμικές εξισώσεις κίνησης σε ένα σημείο ισορροπίας του ρομπότ NXTway-GS. Αυτό συνεπάγεται ότι το όριο του $\psi \rightarrow 0$ ($\sin\psi \rightarrow \psi$, $\cos\psi \rightarrow 1$) και την παραγωγή του δεύτερου όρου της σχέσης $\dot{\psi}^2$. Συνεπώς οι εξισώσεις κίνησης (5.13 - 5.15) γίνονται ως εξής:

$$\left[(2m + M)R^2 + 2J_w + 2n^2 J_m \right] \ddot{\theta} + (MLR - 2n^2 J_m) \ddot{\psi} = F_{\theta} \quad (5.26)$$

$$(MLR - 2n^2 J_m) \ddot{\theta} + (ML^2 + J_{\psi} + 2n^2 J_m) \ddot{\psi} - MgL\psi = F_{\psi} \quad (5.27)$$

$$\left[\frac{1}{2} mW^2 + J_{\phi} + \frac{W^2}{2R^2} (J_w + n^2 J_m) \right] \ddot{\phi} = F_{\phi} \quad (5.28)$$

Οι εξισώσεις (5.26) και (5.27) έχουν και θ και ψ , Η εξίσωση (5.28) έχει μόνο ϕ και εκφράζονται παρακάτω ως :

$$E \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + F \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} + G \begin{bmatrix} \theta \\ \psi \end{bmatrix} = H \begin{bmatrix} v_i \\ v_r \end{bmatrix} \quad (5.29)$$

$$E = \begin{bmatrix} (2m + M)R^2 + 2J_w + 2n^2 J_m & MLR - 2n^2 J_m \\ MLR - 2n^2 J_m & ML^2 + J_\psi + 2n^2 J_m \end{bmatrix}$$

$$F = 2 \begin{bmatrix} \beta + f_w & -\beta \\ -\beta & \beta \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & 0 \\ 0 & -MgL \end{bmatrix}$$

$$H = \begin{bmatrix} \alpha & \alpha \\ -\alpha & -\alpha \end{bmatrix}$$

$$I\ddot{\phi} + J\dot{\phi} = K(v_r - v_l) \quad (5.30)$$

$$I = \frac{1}{2} m W^2 + J_\phi + \frac{W^2}{2R^2} (J_w + n^2 J_m)$$

$$J = \frac{W^2}{2R^2} (\beta + f_w)$$

$$K = \frac{W}{2R} \alpha$$

Στη συνέχεια θεωρούμε τις ακόλουθες μεταβλητές κατάστασης x_1, x_2 και u ως είσοδο. Το x^T συμβολίζει την μετάθεση του x .

$$\mathbf{x}_1 = [\theta, \quad \psi, \quad \dot{\theta}, \quad \dot{\psi}]^T, \quad \mathbf{x}_2 = [\phi, \quad \dot{\phi}]^T, \quad \mathbf{u} = [v_l, \quad v_r]^T \quad (5.31)$$

Συνεπώς παραγωγίζουμε τις καταστατικές εξισώσεις του δίτροχου ενκινημούς από τις εξισώσεις (5.29) και (5.30).

$$\dot{\mathbf{x}}_1 = A_1 \mathbf{x}_1 + B_1 \mathbf{u} \quad (5.32)$$

$$\dot{\mathbf{x}}_2 = A_2 \mathbf{x}_2 + B_2 \mathbf{u} \quad (5.33)$$

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_1(3,2) & A_1(3,3) & A_1(3,4) \\ 0 & A_1(4,2) & A_1(4,3) & A_1(4,4) \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_1(3) & B_1(3) \\ B_1(4) & B_1(4) \end{bmatrix} \quad (5.34)$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ 0 & -J/I \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 0 \\ -K/I & K/I \end{bmatrix} \quad (5.35)$$

$$\begin{aligned} A_1(3,2) &= -gMLE(1,2)/\det(E) \\ A_1(4,2) &= gMLE(1,1)/\det(E) \\ A_1(3,3) &= -2[(\beta + f_w)E(2,2) + \beta E(1,2)]/\det(E) \\ A_1(4,3) &= 2[(\beta + f_w)E(1,2) + \beta E(1,1)]/\det(E) \\ A_1(3,4) &= 2\beta[E(2,2) + E(1,2)]/\det(E) \\ A_1(4,4) &= -2\beta[E(1,1) + E(1,2)]/\det(E) \\ B_1(3) &= \alpha[E(2,2) + E(1,2)]/\det(E) \\ B_1(4) &= -\alpha[E(1,1) + E(1,2)]/\det(E) \\ \det(E) &= E(1,1)E(2,2) - E(1,2)^2 \end{aligned}$$

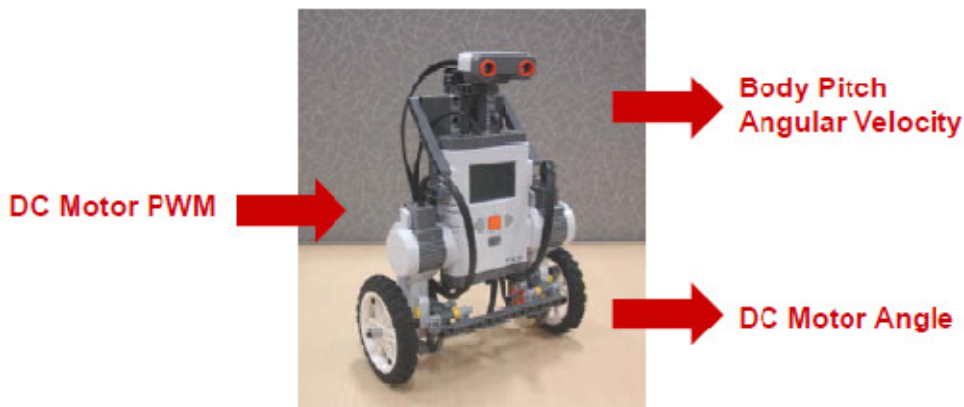
5.4 Σχεδιασμός Ελεγκτή NXTway-GS

Αυτό το κεφάλαιο περιγράφει τον σχεδιασμό του ελεγκτή του NXTway-GS (δίτροχου εκκρεμούς) βάση της σύγχρονης θεωρίας των συστημάτων αυτόματου ελέγχου.

5.4.1 Σύστημα ελέγχου

Είσοδοι και Έξοδοι

Η είσοδος του ενεργοποιητή είναι η ισχύς PWM του αριστερού και του δεξιού κινητήρα (DC) , επίσης η είσοδος της εξίσωσης (5.31) είναι τάση. Οι έξοδοι από τους αισθητήρες είναι η γωνία κινητήρα $\theta_{ml,r}$ και η γωνιακή ταχύτητα βήματος του σώματος ως προς άξονα y είναι $\dot{\psi}$.



Σχήμα 5.3 : Είσοδοι και Έξοδοι

Για τον υπολογισμό των θ , φ χρησιμοποιείται η $\theta_{ml,r}$ και για τον υπολογισμό της ψ υπάρχουν δύο μέθοδοι χρησιμοποιώντας την $\dot{\psi}$.

1^η . Παραγωγή της ψ και αντικατάσταση της γωνιακής ταχύτητα αριθμητικά.

2^η . Εκτίμηση της ψ με την χρήση ενός παρατηρητή βασισμένο στη σύγχρονη θεωρία αυτόματου ελέγχου.

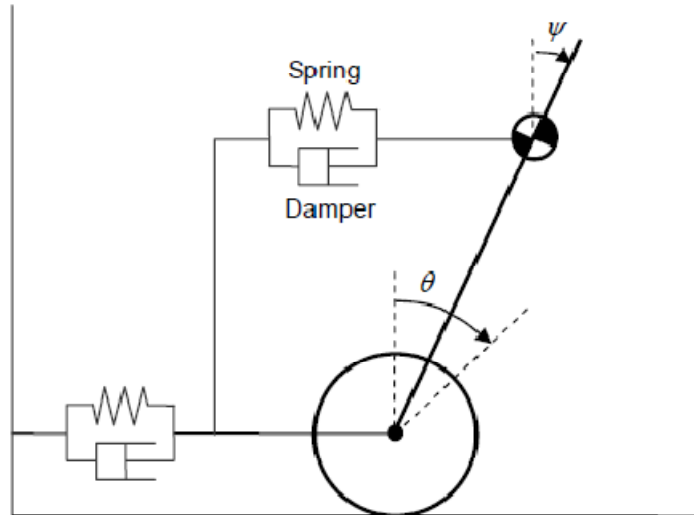
Στην συνέχεια ακολουθείτε η χρήση της 1^{ης} μεθόδου για τον σχεδιασμό ελεγκτή.

Ευστάθεια

Είναι προφανές ότι η θέση ισορροπίας του NXTway-GS δεν είναι σταθερή. Το NXTway-GS θα πρέπει να κινείται στην ίδια κατεύθυνση με την γωνία βήματος του σώματος για να διατηρεί την ισορροπία του. Η σύγχρονη θεωρία συστημάτων αυτόματου ελέγχου δίνει πολλές τεχνικές για την σταθεροποίηση ενός ασταθούς συστήματος (*κοίτα Παράρτημα A1*).

5.4.2 Σχεδιασμός Ελεγκτή

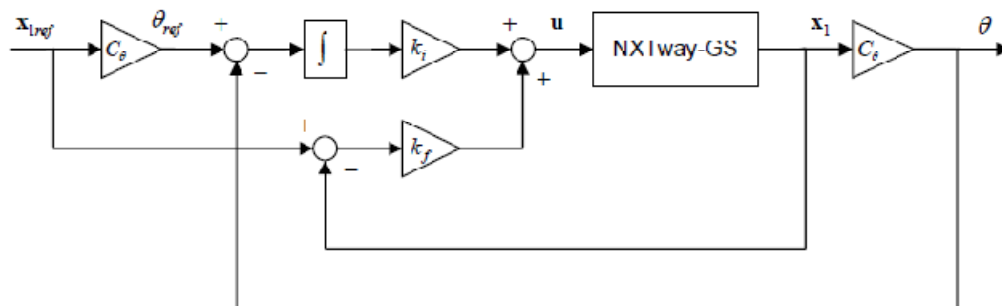
Η εξίσωση (5.29) είναι παρόμοια με την εξίσωση μάζας-ελατηρίου ενός συστήματος αποσβεστήρα ταλάντωσης. Στο σχήμα 5.4 παρουσιάζεται το ισοδύναμο σύστημα του δίτροχου εκκρεμούς που ερμηνεύεται ως ένα σύστημα αποσβεστήρα ταλάντωσης μάζας-ελατηρίου.



Σχήμα 5.4 Ισοδύναμο σύστημα αποσβεστήρα ταλάντωσης μάζας-ελατηρίου.

Για να μπορέσει να γίνει το δίτροχο εκκρεμές σταθερό είναι αναγκαία η ρύθμιση των σταθερών του ελατηρίου και των σταθερών της τριβής του αποσβεστήρα ταλάντωσης (σχήμα 5.4). Οι τεχνικές ελέγχου που περιγράφονται στο παράρτημα Α παρέχουν τη θεωρητική μέθοδο για τον υπολογισμό των σταθερών που προαναφέρθηκαν.

Στην συνέχεια χρησιμοποιείται ένας σέρβο/ελεγκτής που περιγράφεται στο παράρτημα Α.3 ως ελεγκτής NXTway-GS και επιλέγεται η θ ως αναφορά του ελέγχου του σερβοκινητήρα. Σημαντικό είναι ότι δεν μπορούν να χρησιμοποιηθούν μεταβλητές εκτός από την θ ως αναφορά λόγω του ότι το σύστημα τείνει να γίνει ανεξέλεγκτο. Στο σχήμα 5.5 παρουσιάζεται το block diagram του σέρβο/ελεγκτή NXTway-GS.



Σχήμα 5.5 Διάγραμμα ροής σερβοελεγκτή NXTway-GS

Όπου c_θ δίνει ως έξοδο έναν πίνακα από την παραγώγιση του θ από το x_1 .

Υπολογίζεται το κέρδος ανάδρασης και το εσωτερικό κέρδος με την γραμμική τετραγωνική μέθοδο ρυθμιστή. Επιλέγεται ο ακόλουθος πίνακας βαρών Q και R με την μέθοδο δοκιμής και σφάλματος.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 \times 10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^{-2} \end{bmatrix}, R = \begin{bmatrix} 1 \times 10^3 & 0 \\ 0 & 1 \times 10^3 \end{bmatrix}$$

Το $Q(2,2)$ είναι το βάρος για την γωνία βήματος του σώματος και το $Q(5,5)$ είναι το βάρος για τη χρονική ολοκλήρωση της διαφοράς μεταξύ της μέσης υπολογισμένης γωνίας και της αναφοράς αυτής.

Το **param_controller.m** υπολογίζει το γραμμικό τετραγωνικό ρυθμιστή και καθορίζει άλλες παραμέτρους. Ο υπολογισμός του κέρδους από αυτό εξάγεται ως εξής.

```
param_controller.m
% Controller Parameters
% Servo Gain Calculation using Optimal Regulator
A_BAR = [A1, zeros(4, 1); C1(1, :), 0];
B_BAR = [B1; 0, 0];
QQ = [
    1, 0, 0, 0, 0
    0, 6e5, 0, 0, 0
    0, 0, 1, 0, 0
    0, 0, 0, 1, 0
    0, 0, 0, 0, 4e2
];
RR = 1e3 * eye(2);
KK = lqr(A_BAR, B_BAR, QQ, RR);
k_f = KK(1, 1:4); % feedback gain
k_i = KK(1, 5); % integral gain
% suppress velocity gain because it fluctuates NXTway-GS
k_f(3) = k_f(3) * 0.85;
```

Το αποτέλεσμα είναι :

$$k_f = [-0.8351, -34.1896, -1.0995, -2.8141]$$

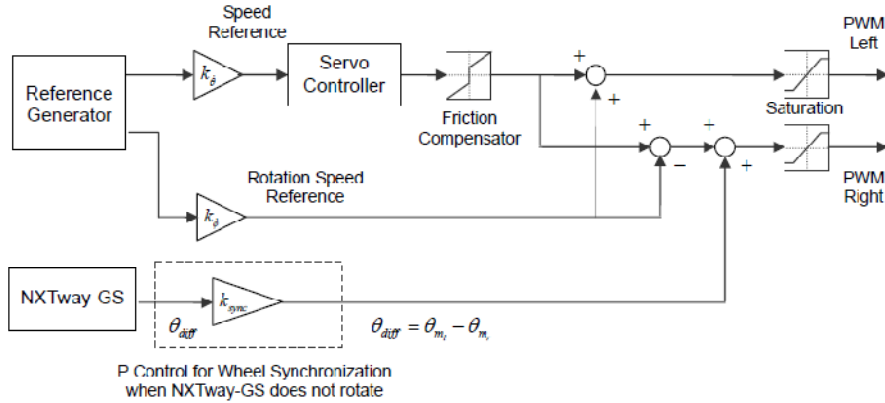
$$k_i = -0.4472$$

Ρυθμίζουμε την τιμή της ταχύτητας του κέρδους $k_f(3)$ μετά από τον γραμμικό τετραγωνικό υπολογισμό ρυθμιστών επειδή το NXTway-GS κυμαίνεται υπερβολικά.

Επιπλέον προστίθεται ο ακόλουθος έλεγχος.

- Περιστροφή του NXTway-GS δίνοντας διαφορετικές τιμές στον αριστερό και στον δεξιό κινητήρα.
- Χρήση P ελέγχου για τον συγχρονισμό των τροχών όταν η NXTway-GS τρέχει ευθεία επειδή η γωνία περιστροφής των κινητήρων δεν είναι ίδια ακόμη και αν εφαρμόζεται το ίδιο PWM.

Συνεπώς παράγουμε τον NXTway-GS ελεγκτή όπως στο σχήμα 5.6. Εδώ οι k_θ , k_ϕ , k_{sync} ρυθμίζουν τις παραμέτρους.



Σχήμα 5.6 : Διάγραμμα μοντέλου ελεγκτή NXTway-GS.

5.5 Μοντέλο NXTway-GS

Αυτό το κεφάλαιο περιγράφει συνοπτικά το μοντέλο NXTway-GS και τα αρχεία παραμέτρων.

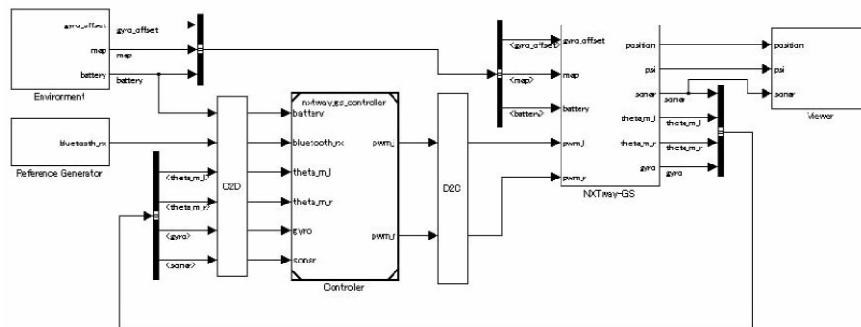
5.5.1 Το Μοντέλο συνοπτικά

Το `nxtway_gs.mdl` και το `nxtway_gs_vr.mdl` είναι μοντέλα του συστήματος ελέγχου του NXTway-GS. Τα δύο μοντέλα είναι παρόμοια αλλά διαφέρουν ως προς το σημείο της τρισδιάστατης αναπαράστασης που παρέχεται από την εργαλειοθήκη εικονικής πραγματικότητας (Virtual Reality Toolbox). Αυτό το μοντέλο αποτελείται από 5 τμήματα : Environment, Reference Generator, NXTway-GS Controller, NXTway-GS Plant, και Simulation Viewer.

NXTway-GS Simulation Model (Virtual Reality Toolbox is not needed)

This model consists of five parts: Environment, Reference Generator, NXTway-GS Controller, NXTway-GS Plant, and Simulation Viewer.

Disclaimer:
LEGO®(R) is a trademark of the LEGO Group of companies which does not sponsor, authorize or endorse this demo.
LEGO®(R) and Mindstorms®(R) are registered trademarks of The LEGO Group.

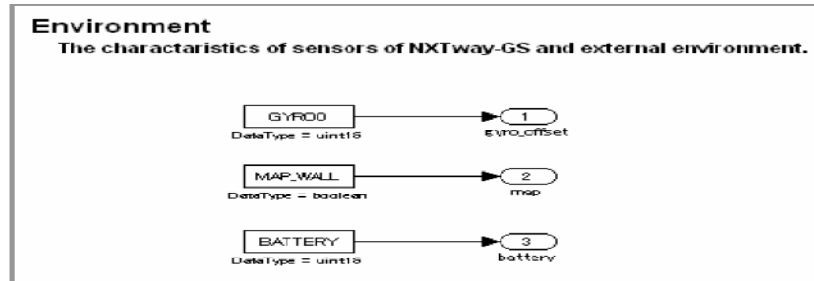


Σχήμα 5.7 `nxtway_gs.mdl`

Τα κύρια μέρη του `nxtway_gs.mdl` και του `nxtway_gs_vr.mdl` παρουσιάζονται στην συνέχεια.

5.5.1.1 Περιβάλλον

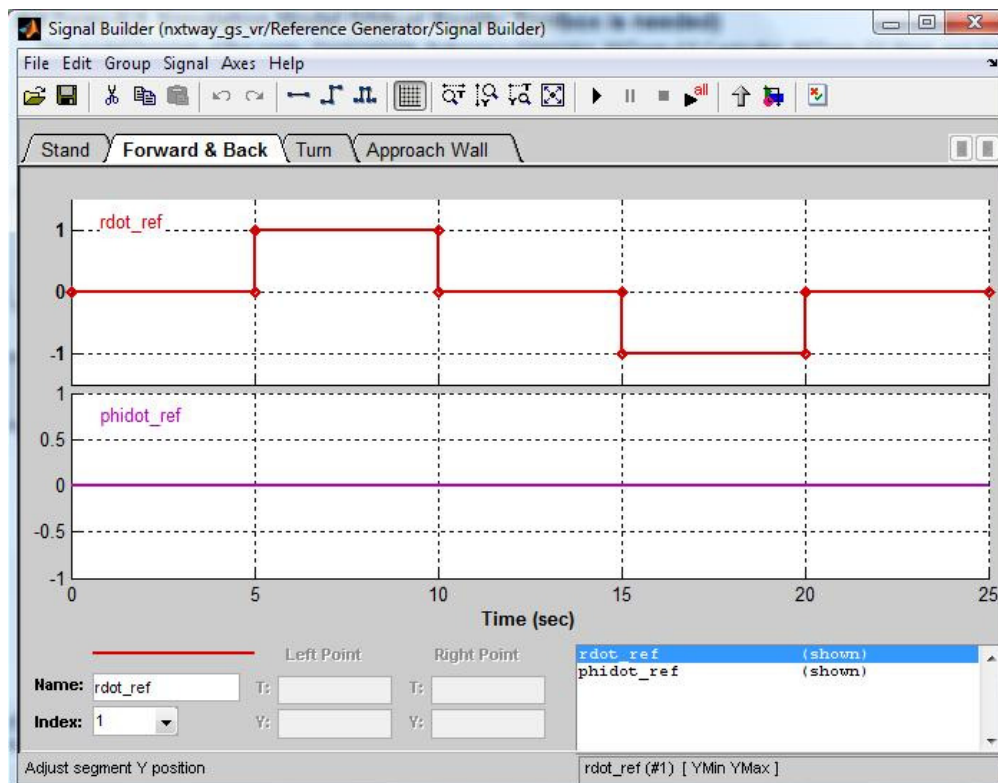
Το υποσύστημα που παρουσιάζεται από κάτω καθορίζει τις παραμέτρους του περιβάλλοντος. Για παράδειγμα δεδομένα χάρτη(map data), τιμή του gyro offset και δεδομένα μπαταρίας.



Σχήμα 5.8 Περιβάλλον υποσυστήματος

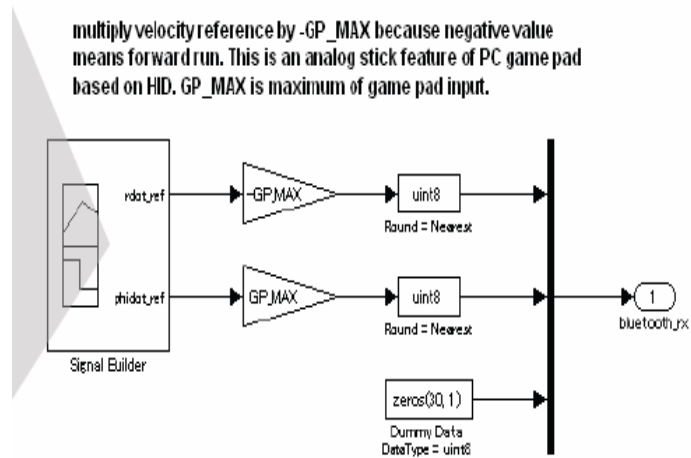
5.5.1.2 Αναφορά γεννήτριας

Αυτό το υποσύστημα αναφέρεται στο σήμα της γεννήτριας του NXTway-GS. Η ταχύτητα αναφοράς και η ταχύτητα περιστροφής μπορεί να αλλάξει χρησιμοποιώντας ένα μοντέλο κατασκευής σήματος(Signal Builder block). Το σήμα εξόδου είναι δεδομένο 32byte το οποίο έχει απλά δεδομένα προκειμένου να ικανοποιούνται οι προδιαγραφές χρήσης του NXT GamePad.



Reference Generator

Reference signal generator.



Σχήμα 5.9 Αναφορά υποσυστήματος γεννήτριας

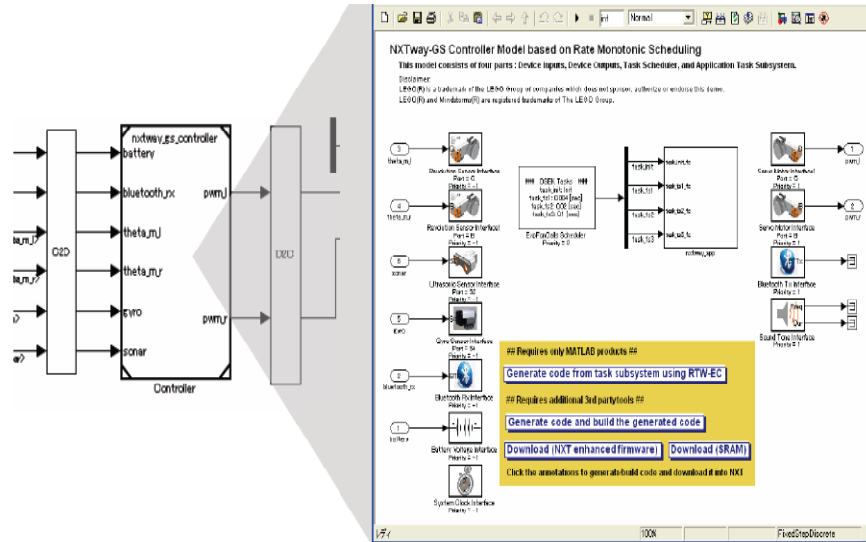
Το σχήμα 5.9 παρουσιάζει την σχέση μεταξύ του υπολογισμού της ταχύτητας και της ταχύτητας περιστροφής για ένα αναλογικό χειριστήριο παιχνιδιών υπολογιστή (pc gamepad).



Σχήμα 5.10 Είσοδοι από αναλογικό pc gamepad

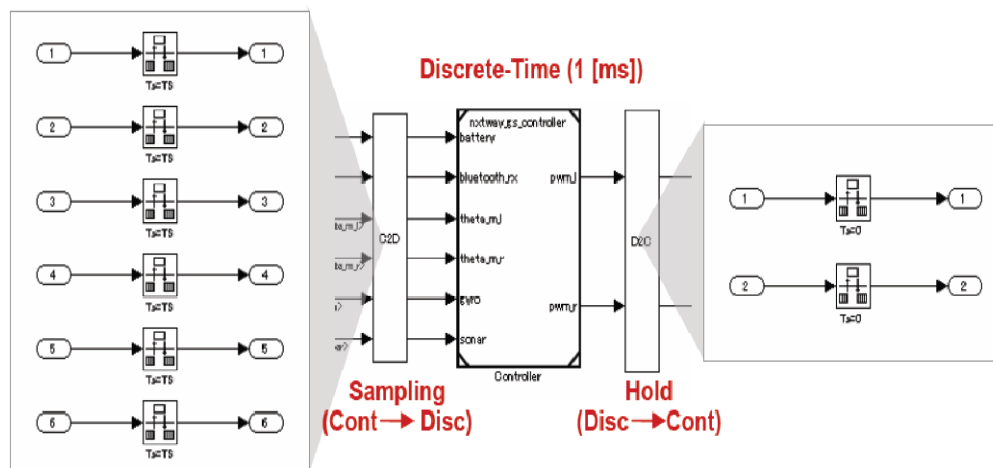
5.5.1.3 Ελεγκτής

Αυτό το diagram block είναι ο ψηφιακός ελεγκτής του NXTway-GS και αναφέρεται ως **nxtway_gs_controller.mdl**. Για περισσότερες λεπτομέρειες η αναφορά για το μοντέλο του ελεγκτή γίνεται στο κεφάλαιο 5.12 (απλή ακρίβεια αριθμητικής κινήτης υποδιαστολής- “Single Precision Floating-Point Arithmetic”).



Σχήμα 5.11 Διάγραμμα μοντέλου(block) ψηφιακού ελεγκτή (nxtway_gs_controller.mdl)

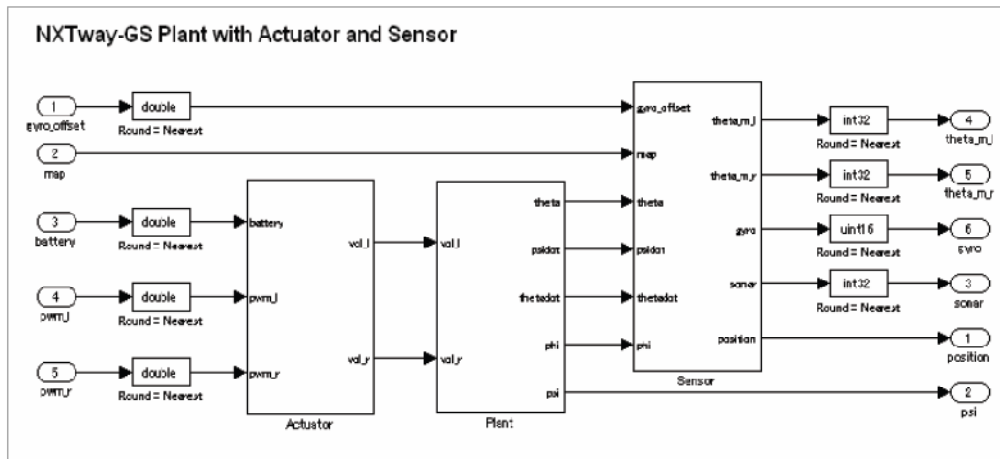
Το block του ελεγκτή εκτελείται σε διακεκριμένες χρονικές στιγμές (χρονικά δείγματα βάσης – “base sample time” = 1 [ms]) και η εγκατάσταση (plant) (το υποσύστημα NXTway-GS) εκτελείται σε συνεχή χρόνο (χρόνος δείγματος= 0[ms]). Συνεπώς είναι απαραίτητο να γίνεται η μετατροπή του συνεχή-χρόνου σε διακεκριμένο(discrete time) και το αντίστροφο με την καταχώρηση μετατροπών εύρους, “Rate Transition blocks”, κατάλληλα συνδεδεμένων μεταξύ τους. Το Rate Transition block χειρίζεται την μεταφορά δεδομένων μεταξύ blocks που λειτουργούν σε διαφορετικό εύρος στην συγκεκριμένη περίπτωση μετατρέπει το δείγμα από συνεχή σε διακριτό και το αντίστροφο (C2D/D2C).



Σχήμα 5.12 Μετατροπές εύρους ανάμεσα στον ελεγκτή(“controller”) και την εγκατάσταση(“plant”)

5.5.1.4 NXTway-GS

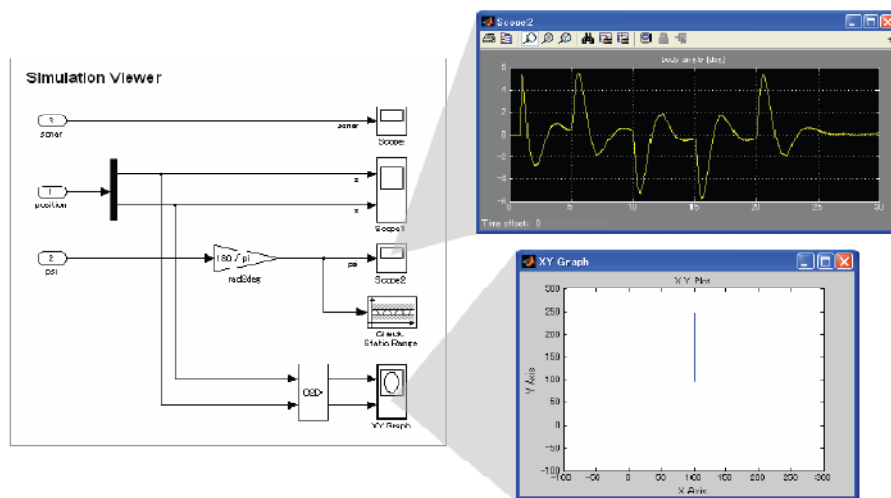
Αυτό το υποσύστημα είναι το μαθηματικό μοντέλο του NXTway-GS. Αποτελείται από τον αισθητήρα, τον ενεργοποιητή, και το γραμμικό μοντέλο εγκατάστασης (“linear plant model”) και αναφέρεται ως `nxtway_gs_plant.mdl`. Περισσότερες λεπτομέρειες αναφέρονται στο κεφάλαιο 5.6.3 μοντέλο εγκατάσταση (“plant model”).



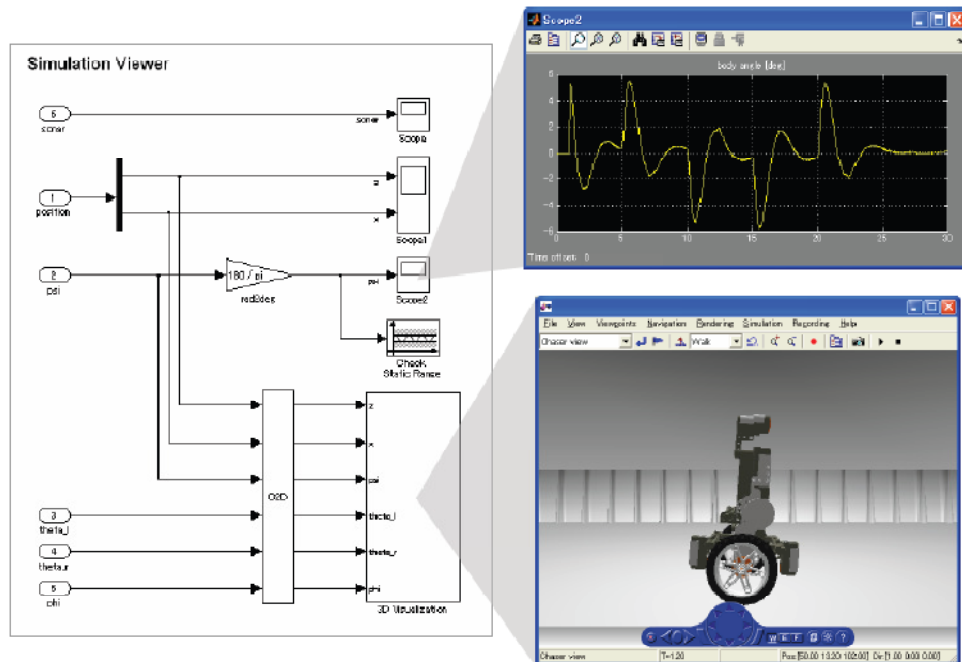
Σχήμα 5.13 Υποσύστημα NXTway-GS

5.5.1.5 Παρουσιαστής (“viewer”)

Αυτό το υποσύστημα περιλαμβάνει τον παρουσιαστή προσομοίωσης (“simulation viewer”). Το `nxtway_gs.mdl` περιλαμβάνει έναν παρουσιαστή θέσης με ένα διάγραμμα X,Y “block” και το `nxtway_gs_vr.mdl` παράγει έναν τρισδιάστατο παρουσιαστή (“3D viewer”) που παρέχεται από την εργαλειοθήκη εικονικής πραγματικότητας (“Virtual Reality Toolbox”).



Σχήμα 5.14 Παρουσιαστής υποσυστήματος (`nxtway_gs.mdl`)



Σχήμα 5.15 Παρουσιαστής υποσυστήματος

5.5.2 Αρχεία Παραμέτρων

Ο Πίνακας 7.1 παρουσιάζει τα αρχεία προσομοίωσης και παραγωγής κώδικα.

Όνομα αρχείου	Περιγραφή
param_controller.m	M-script for controller parameters
param_controller_fixpt.m	M-script for fixed-point settings (Simulink.NumericType)
param_nxtway_gs.m	M-script for NXTway-GS parameters (It calls param_*.m)
param_plant.m	M-script for plant parameters
param_sim.m	M-script for simulation parameters

Πίνακας 7.1 Αρχεία Παραμέτρων

param_nxtway_gs.m calls **param_*.m** (τα ******* υποδηλώνουν controller, plant, sim) και δημιουργεί όλες τις παραμέτρους στον βασικό χώρο εργασίας (“base workspace”). Η συνάρτηση ανάκλησης του μοντέλου χρησιμοποιείται για να εκτελεστεί το **param_nxtway_gs.m** αυτόματα όταν φορτώνεται το μοντέλο.

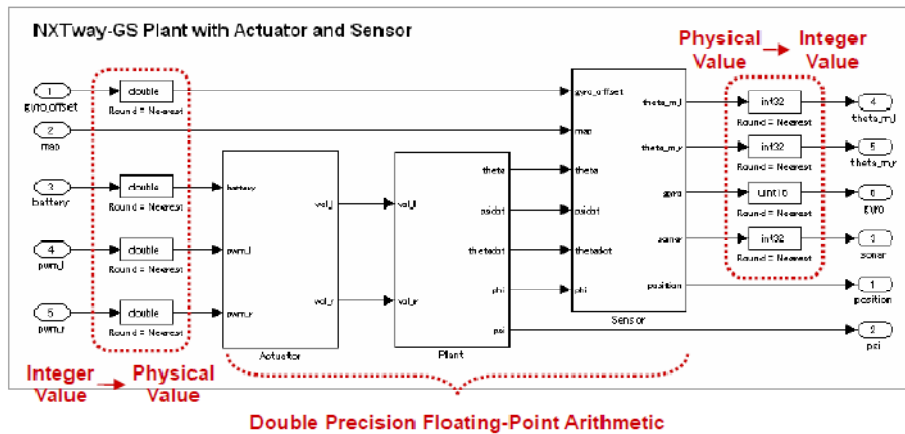
Για να εμφανίσετε την συνάρτηση ανάκλησης του μοντέλου, επιλέξτε ιδιότητες μοντέλου [“model properties”] από τις επιλογές αρχεία [“files”] του Simulink.

5.6 Μοντέλο Εγκατάστασης (Plant Model)

Σε αυτό το κεφάλαιο περιγράφεται το υποσύστημα NXTway-GS στο `nxtway_gs.mdl` και `nxtway_gs_vr.mdl`

5.6.1 Το Μοντέλο Συνοπτικά

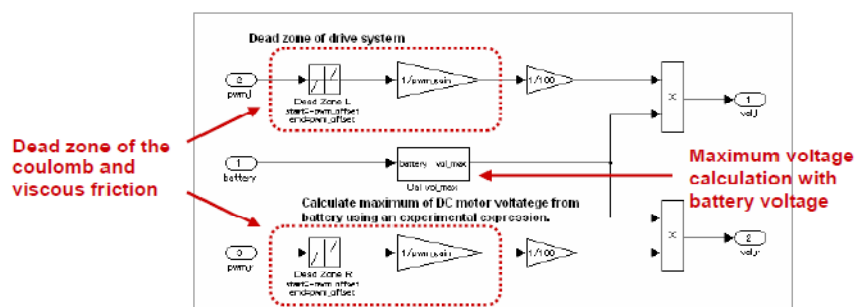
Το υποσύστημα NXTway-GS αποτελείται από τους αισθητήρες, τους ενεργοποιητές, και το γραμμικό μοντέλο εγκατάστασης (“plant model”). Μετατρέπει τον τύπο δεδομένων των σημάτων εισόδου σε “double” τύπου δεδομένα, υπολογίζει τη δυναμική εγκατάστασης χρησιμοποιώντας την διπλής αριθμητικής ακρίβειας κινητή υποδιαστολή (“double precision floating-point”), και εξάγει τα αποτελέσματα μετά την κβάντιση.



Σχήμα 5.16 Το υποσύστημα NXTway-GS

5.6.2 Ενεργοποιητής

Το υποσύστημα του ενεργοποιητή υπολογίζει την τάση των κινητήρων με χρήση του PWM που παράγεται από τον ελεγκτή. Εξετάζοντας την μονάδα ηλεκτρικού φορτίου (Coulomb) και την ιξώδη τριβή στη μετάδοση, το μοντελοποιούμε ως νεκρή ζώνη προτού υπολογιστεί η τάση.



Σχήμα 5.17 Το υποσύστημα του ενεργοποιητή

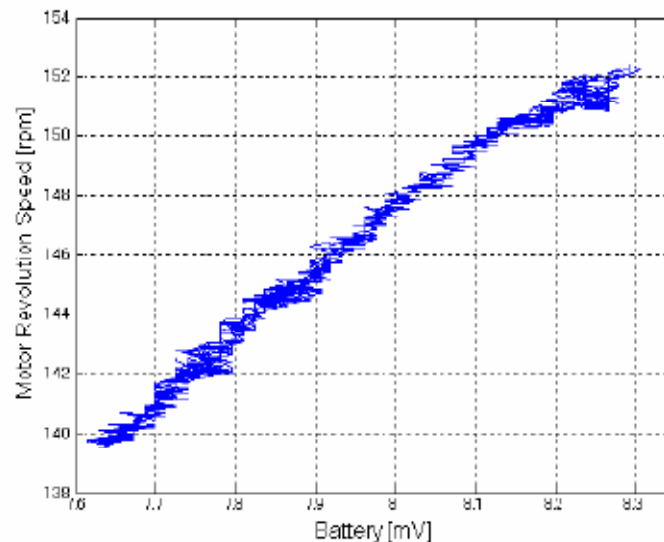
Το dead zone block παρέχει μηδενικά σε μια περιοχή(μεταξύ κάποιων ορίων),πχ αν η είσοδος είναι \geq του άνω ορίου τότε η έξοδος = είσοδος –άνω όριο ενώ αν η είσοδος \leq του κάτω ορίου τότε η έξοδος= είσοδος – κάτω όριο.

Η μέγιστη τάση των κινητήρων είναι απαραίτητη για να υπολογιστεί το PWM. Στο υποσύστημα **Cal vol_max subsystem**, χρησιμοποιούμε την ακόλουθη πειραματική εξίσωση που προκύπτει από έναν κανόνα μετατροπής μεταξύ της τάσης της μπαταρίας V_B και την μέγιστη τάση του κινητήρα V_{max} .

$$V_{max} = 0.001089 \times V_B - 0.625 \quad (8.1)$$

Η Εξίσωση (8.1) έχει προκύψει από το σχήμα 5.18 και από τα δεδομένα στο παράρτημα [1].

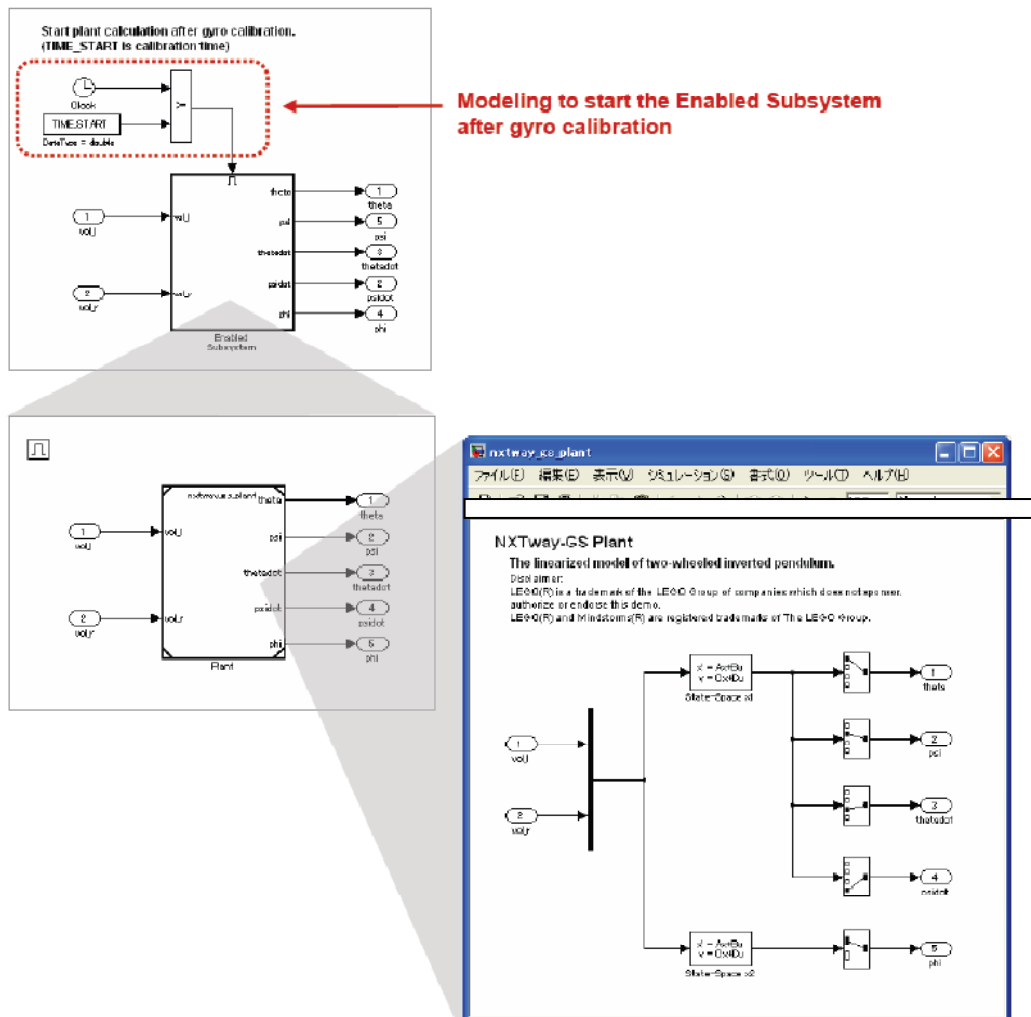
Η τάση των DC κινητήρων και η ταχύτητα περιστροφής είναι ανάλογες προς την τάση της μπαταρίας.



Σχήμα 5.18 : Πειραματικά αποτελέσματα από την τάση της μπαταρίας και την ταχύτητα περιστροφής για PWM=100%

5.6.3 Εγκατάσταση (Plant)

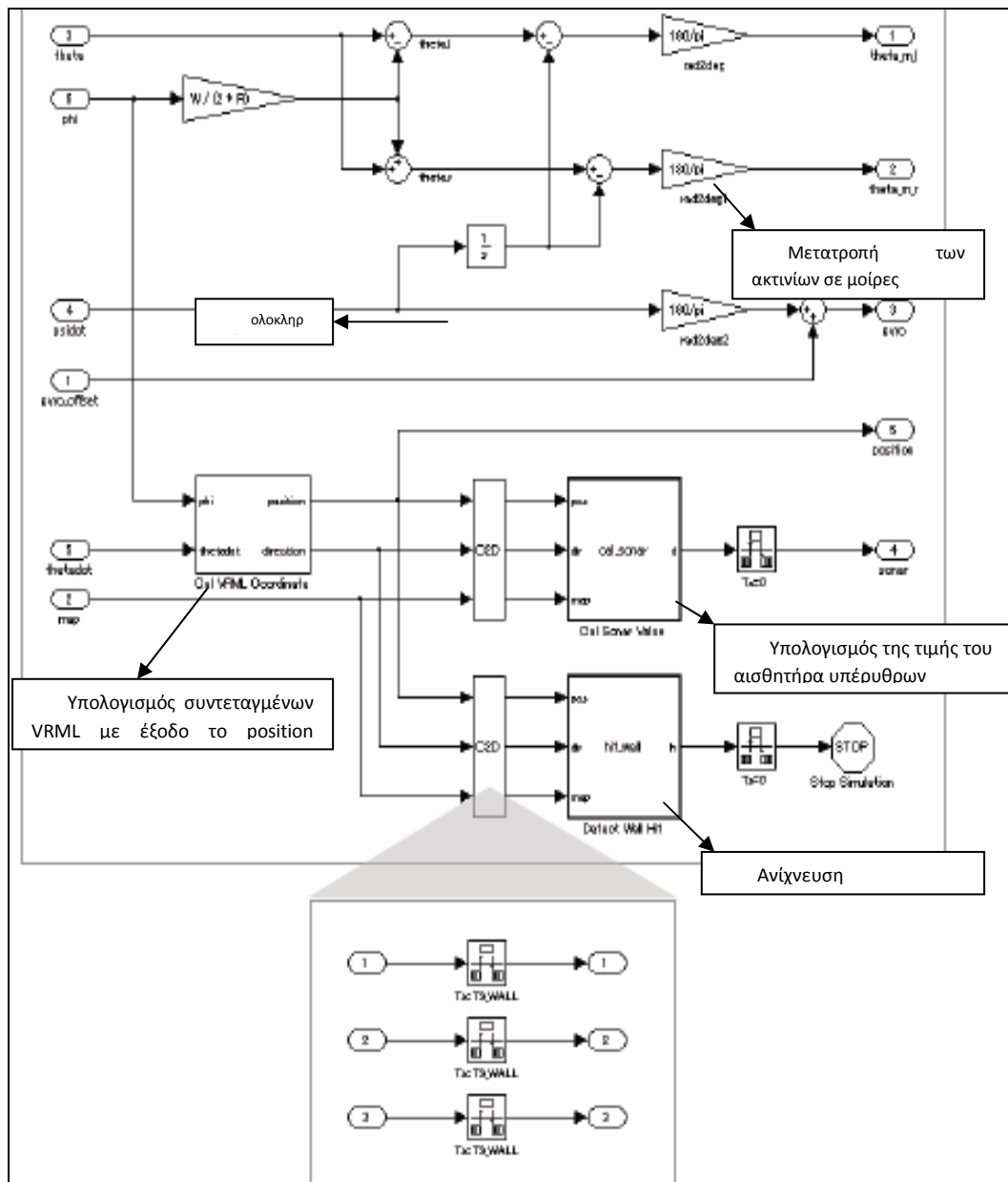
Το `nxtway_gs_plant.mdl` είναι το μοντέλο της καταστατικής εξίσωσης που παράγεται από τις καταστατικές εξισώσεις του δίτρουχου εκκρεμούς (κεφάλαιο 5.3). Αναφέρεται και στο ενεργοποιημένο υποσύστημα του Model Block. Λαμβάνοντας υπόψη την ρύθμιση του offset του γυροσκοπικού αισθητήρα, η εγκατάσταση διαμορφώνεται έτσι ώστε να αρχίζουν οι υπολογισμοί αφού έχει προηγηθεί η ρύθμιση του γυροσκοπικού αισθητήρα. Περισσότερες πληροφορίες για την ρύθμιση του γυροσκοπικού αισθητήρα υπάρχουν στην αναφορά 5.7 Συνοπτική περιγραφή προγράμματος.



5.6.4 Αισθητήρας (Sensor)

Το υποσύστημα του αισθητήρα μετατρέπει τις τιμές που υπολογίζονται στο υποσύστημα εγκατάστασης σε αρκετές εξόδους αισθητήρα. Επειδή το κόστος στάθμισης του υπολογισμού της απόστασης (η έξοδος από τον αισθητήρα υπέρηχων) και η ανίχνευση πρόσκρουσης σε τοίχο είναι μεγάλη περικλύπτουμε τα βήματα υπολογισμού με την εισαγωγή μοντέλων μετατροπής εύρους(των Rate Transition blocks). Περισσότερες πληροφορίες

σχετικά με τον υπολογισμό της απόστασης και της πρόσκρουσης σε τοίχο αναφέρονται στο παράρτημα Β.3.



Σχήμα 5.19 Το υποσύστημα του Αισθητήρα

5.7 Μοντέλο Ελεγκτή

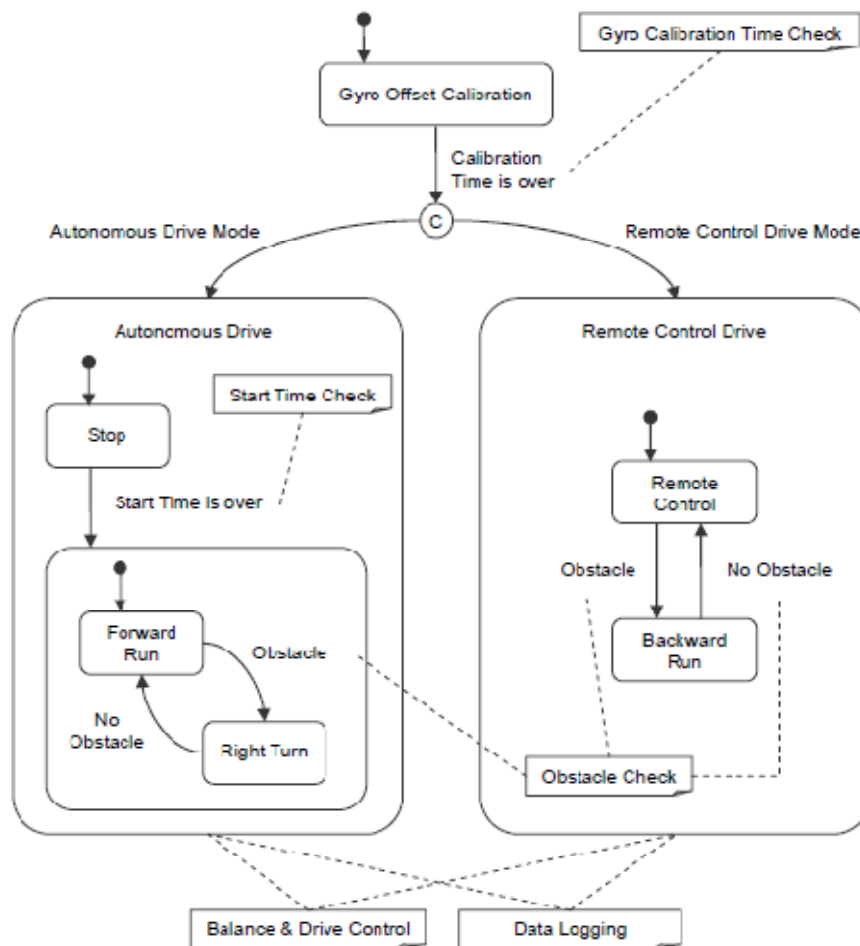
(Ενιαίας κινητής υποδιαστολή αριθμητικής ακρίβειας)

Αυτό το κεφάλαιο περιγράφει το πρόγραμμα ελέγχου , τη διαμόρφωση του στόχου και τα περιεχόμενα του μοντέλου `nxtway_gs_controller.mdl`.

5.7.1 Συνοπτική περιγραφή ελέγχου του προγράμματος

5.7.1.1 Διάγραμμα καταστάσεων μηχανής

Στο σχήμα 5.20 παρουσιάζεται το διάγραμμα καταστάσεων της μηχανής του ελεγκτή NXTway-GS .Υπάρχουν δύο καταστάσεις οδήγησης . Η μία είναι η κατάσταση αυτόνομης οδήγησης και η άλλη είναι η κατάσταση απομακρυσμένου ελέγχου με την χρήση αναλογικού τηλεχειριστηρίου για παιχνίδια H/Y (pc gamepad) μέσω της συνδεσιμότητας του NXT με gamepad.



Σχήμα 5.20 Διάγραμμα καταστάσεων μηχανής ελεγκτή NXTway-GS.

5.7.1.2 Διαμόρφωση εργασιών

Ο ελεγκτής του NXTway-GS εκτελεί 4 εργασίες που περιγράφονται στον πίνακα 5.7.1

Task	Period	Works
task_init	Initialization only	Initial value setting
task_ts1	4 [ms]	Balance & drive control Data logging Gyro offset calibration
task_ts2	20 [ms]	Obstacle check
task_ts3	100 [ms]	Time check Battery voltage averaging

Πίνακας 5.7.1 Εργασίες του ελεγκτή NXTway-GS

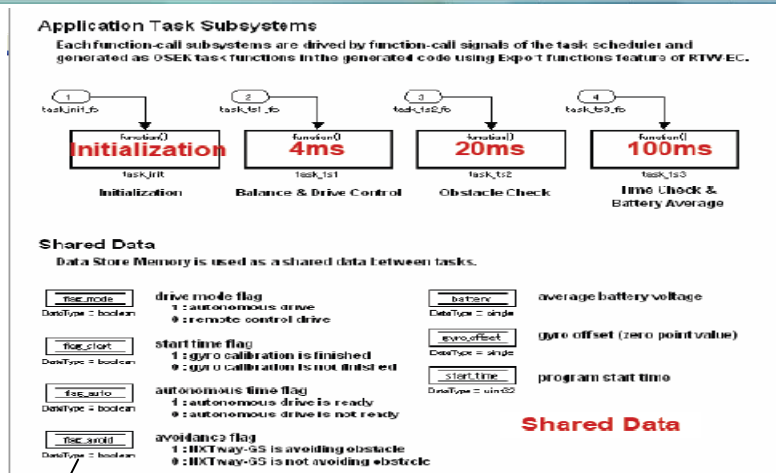
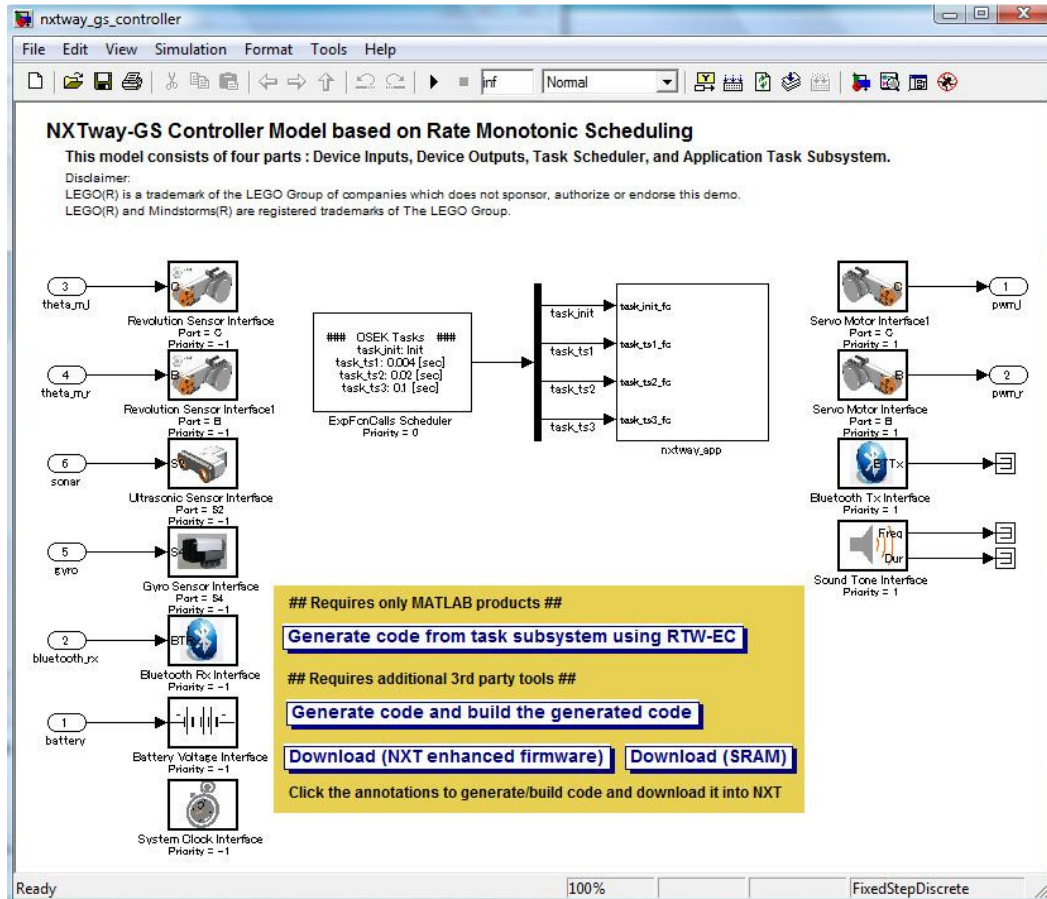
Χρησιμοποιείται περίοδος 4[ms] για τον έλεγχο της λειτουργίας ισορροπίας και οδήγησης θεωρώντας ως μέγιστο αριθμό δειγμάτων του γυροσκοπικού αισθητήρα το ένα δευτερόλεπτο. Με παρόμοιο τρόπο χρησιμοποιείται περίοδος 20[ms] για τον έλεγχο εμποδίων του αισθητήρα υπέρηχων. Ο ελεγκτής περιγράφεται στο κεφάλαιο 6.2 Σχεδιασμός Ελεγκτή που χρησιμοποιείτε για την ισορροπία και τον έλεγχο οδήγησης.

5.7.1.3 Τύπος Δεδομένων

Χρησιμοποιείται ενιαία κινητή υποδιαστολή αριθμητικής ακρίβειας δεδομένων (single precision floating-point data) για την ισορροπία και τον έλεγχο της κίνησης ώστε να μειώνονται τα λάθη των υπολογισμών. Δεν υπάρχει κάποια μονάδα επεξεργασίας αριθμού κινητής υποδιαστολής FPU(Floating point number processing Unit) στον επεξεργαστή ARM 7 που περιέχετε στον εγκέφαλο NXT, αλλά μπορεί να εκτελεστεί λογισμικό ενιαίας κινητής υποδιαστολής αριθμητικής ακρίβειας χρησιμοποιώντας την βιβλιοθήκη κινητής υποδιαστολής αριθμητικής ακρίβειας που παρέχεται από το GCC (δια- μεταγλωττιστής).

5.7.2 Σύνοψη Μοντέλου

Το `nxtway _GS_controller.mdl` βασίζεται στον Embedded Coder Robot NXT framework.



Σχήμα 5.21 : `nxtway_gs_controller.mdl`

Το `nxtway_app` subsystem περιέχει τα υποσυστήματα τα οποία ενεργοποιούνται από σήματα συναρτήσεων κλήσης τα οποία παράγονται από τον χρονοπρογραμματισμό (ExpFcnCall Scheduler)

5.7.2.1 Διεπαφή συσκευής (Device Interface)

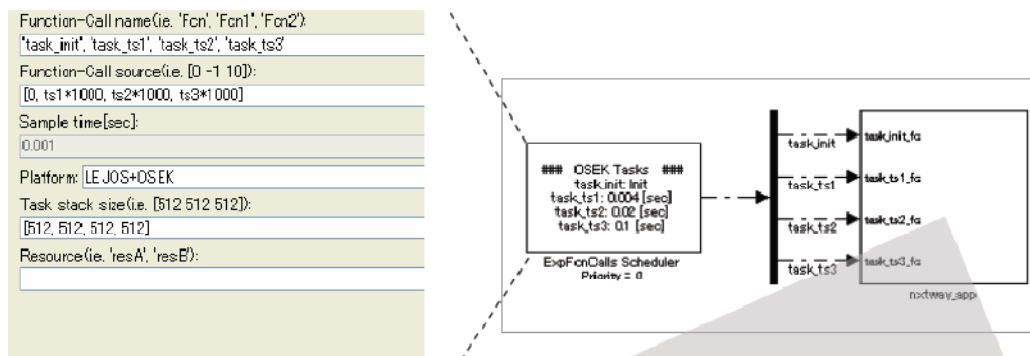
Οι διεπαφές συσκευών μπορούν να δημιουργηθούν με την χρήση των blocks των αισθητήρων και των ενεργοποιητών που παρέχονται από την βιβλιοθήκη του Embedded Coder Robot NXT.



Σχήμα 5.22 Διεπαφή DC κινητήρα

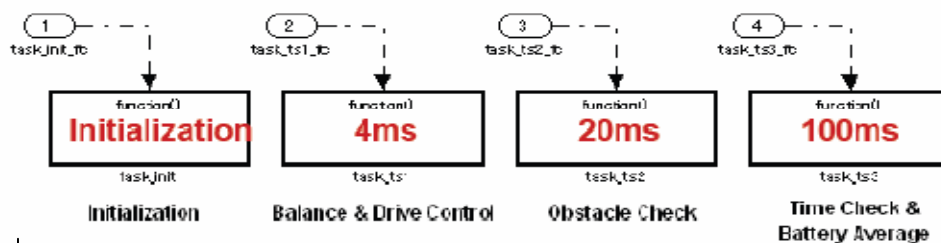
5.7.2.2 Χρονοπρογραμματισμός εργασιών. (Scheduler -Tasks)

Η συνάρτηση ExpFcnCalls Scheduler block περιλαμβάνει διαμόρφωση εργασιών όπως όνομα εργασίας, χρονική περίοδο, πλατφόρμα και μέγεθος στοιβάς εργασίας (stack size). Μπορούν να δημιουργηθούν υποσυστήματα εργασιών(subsystem tasks) συνδέοντας τα σήματα των συναρτήσεων κλήσης(function-call) από τον χρονοπρογραμματισμό με τα υποσυστήματα εργασιών.



Application Task Subsystems

Each function-call subsystems are driven by function-call signals of the task scheduler and generated as OSEK task functions in the generated code using Export functions feature of RTW-EC.



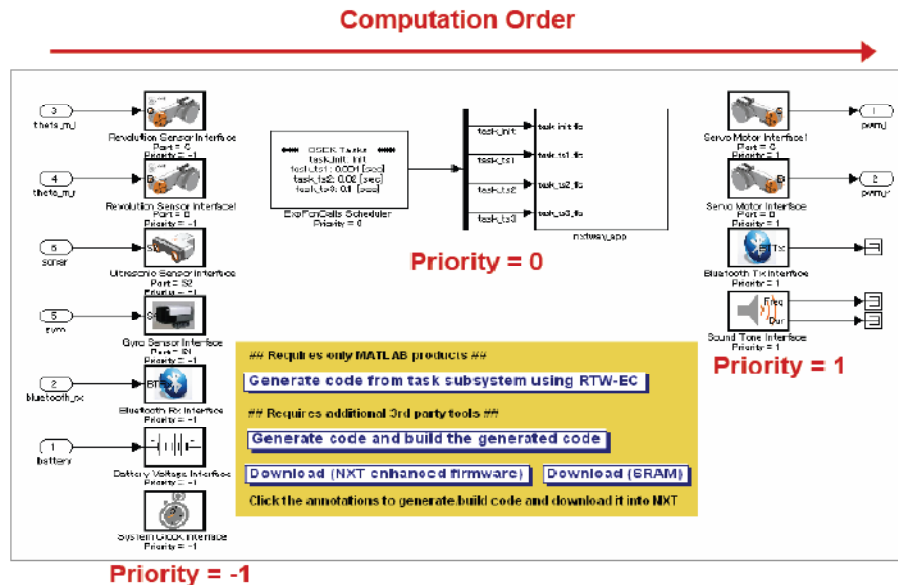
Σχήμα 5.23 : Χρονοπρογραμματισμός και Εργασίες (Scheduler and Tasks)

Κάθε υποσύστημα συνάρτησης κλήσης οδηγείται από τα σήματα των συναρτήσεων κλήσης του χρονοπρογραμματισμού εργασίας και παράγονται ως συναρτήσεις εφαρμογών της OSEK στον παραγόμενο κώδικα χρησιμοποιώντας συναρτήσεις εξαγωγής που παρουσιάζονται στο RTW-EC.

5.7.2.3 Προτεραιότητα (Priority)

Πρέπει να δοθεί προτεραιότητα των blocks των συσκευών και της ExpFcnCalls Scheduler Block στο πρωταρχικό στάδιο ταξινόμησης, με την ακόλουθη σειρά 1. Είσοδοι συσκευών -> 2. εργασίες -> 3. Έξοδοι συσκευών. Οι μικροί αριθμοί δηλώνουν υψηλή προτεραιότητα(επιτρέπονται και αρνητικοί αριθμοί).

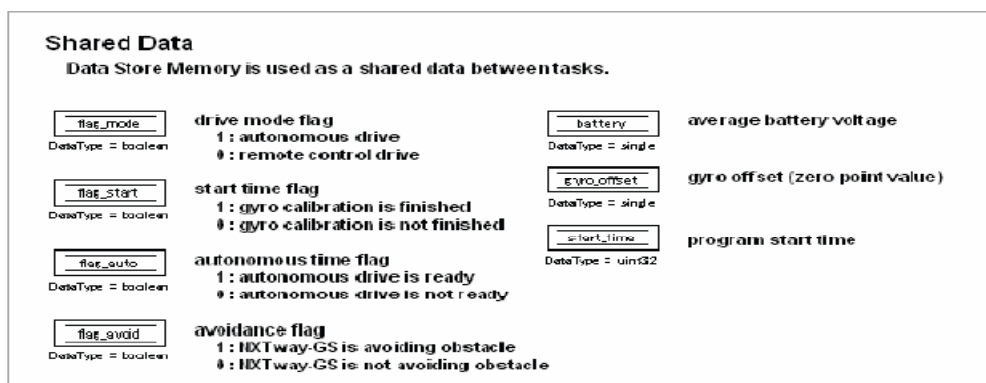
Για να εμφανίσετε την προτεραιότητα κάντε δεξί κλικ και επιλέξτε [Block priorities]



Σχήμα 5.24 Ρύθμιση Προτεραιότητας

5.7.2.4 Δεδομένα διαμοιρασμού (Shared Data)

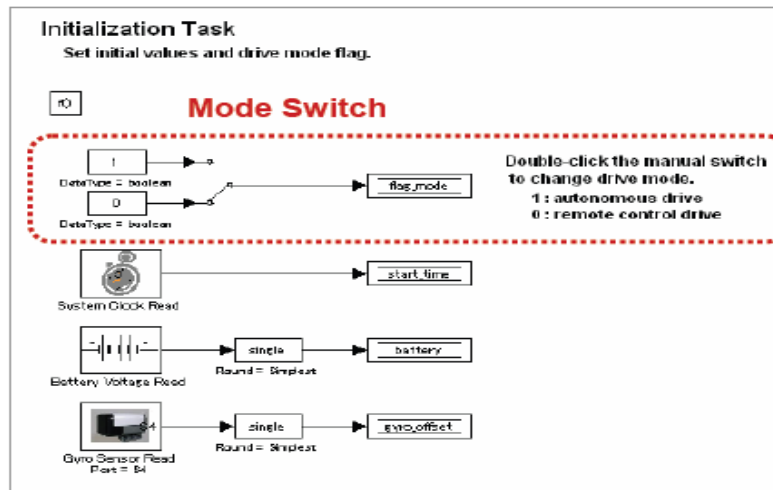
Για τον διαμοιρασμό των δεδομένων μεταξύ των εργασιών χρησιμοποιούνται τα Data Store Memory Blocks. Για παράδειγμα το flag mode χρησιμοποιείται για να κρατάει δεδομένα για την κατάσταση οδήγησης (1=αυτόνομη οδήγηση, 0=απομακρυσμένος έλεγχος)



Σχήμα 5.25 : Δεδομένα διαμοιρασμού (Shared Data)

5.8 Αρχικοποίηση Εργασίας : task_init

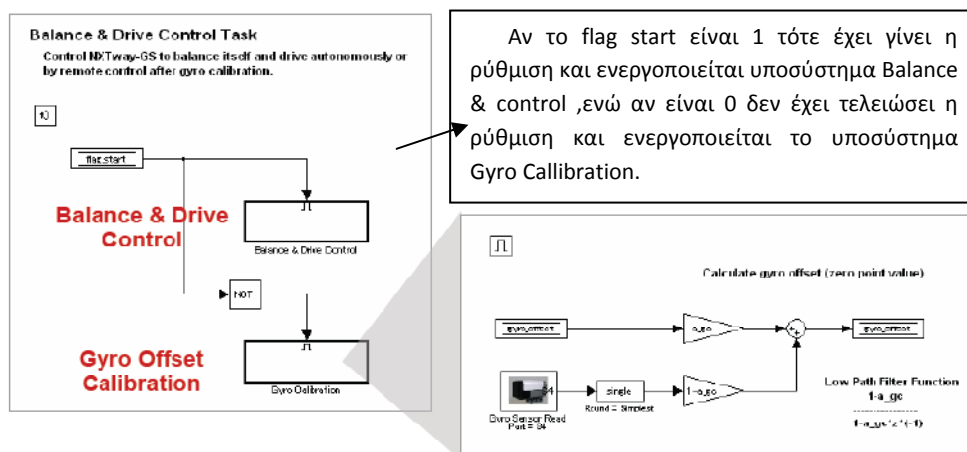
Σε αυτό το υποσύστημα(task) θέτονται οι αρχικές τιμές. Η κατάσταση οδήγησης μπορεί να αλλάξει σε κατάσταση αυτόνομης οδήγησης ή κατάσταση οδήγησης απομακρυσμένου ελέγχου. Όπως φαίνεται στο ακόλουθο υποσύστημα υπάρχει ένας χειροκίνητος διακόπτης για την επιλογή της καταστάσεως οδήγησης. 1=αυτόνομη οδήγηση, 0= απομακρυσμένη οδήγηση.



Σχήμα 5.26: Το υποσύστημα task_init

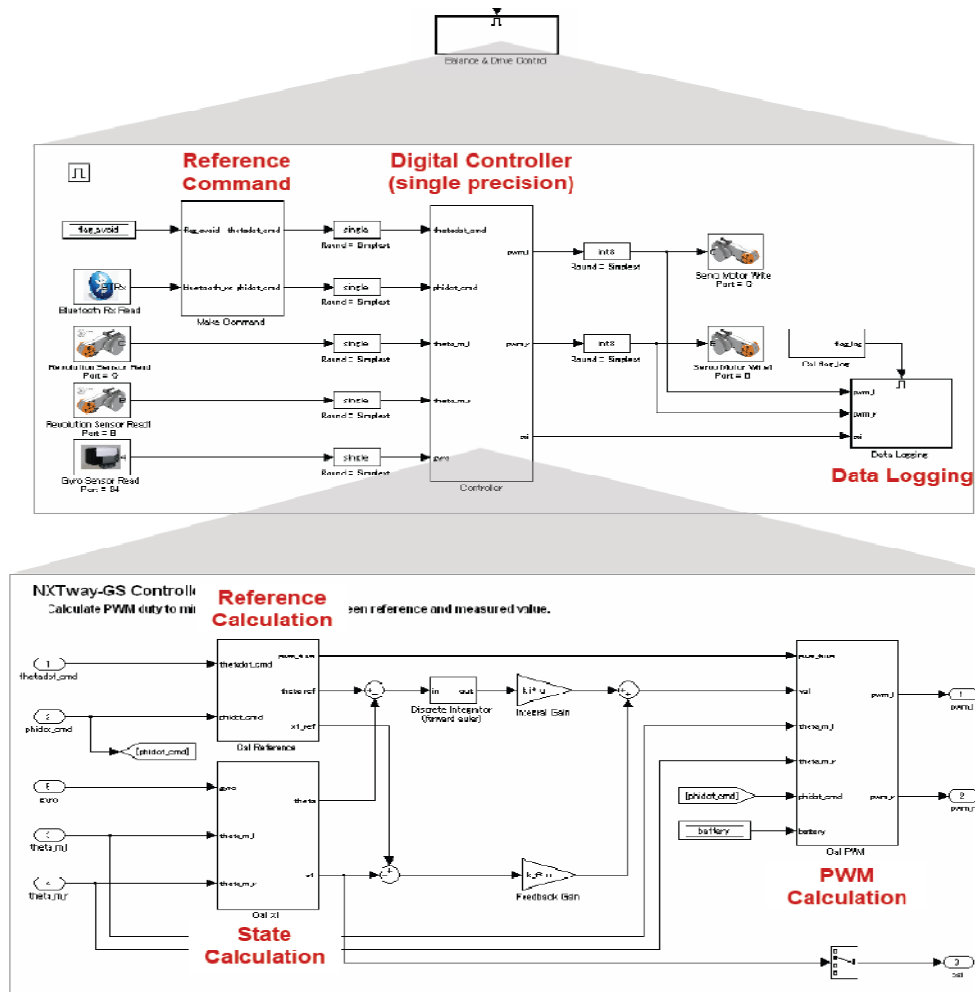
5.8.1 4ms Task : task_ts1

Αυτό είναι το task για την “ρύθμιση” του offset του γυροσκοπικού αισθητήρα, τον έλεγχο ισορροπίας & οδήγησης και την φόρτωση των δεδομένων. Η ισορροπία και ο έλεγχος οδήγησης εκτελείται μετά την ρύθμιση του γυροσκοπικού αισθητήρα. Ο χρόνος του offset της ρύθμισης του γυροσκοπικού αισθητήρα ορίζεται ως time_start στο param_controller.m.



Σχήμα 5.27 : Το υποσύστημα task_ts1

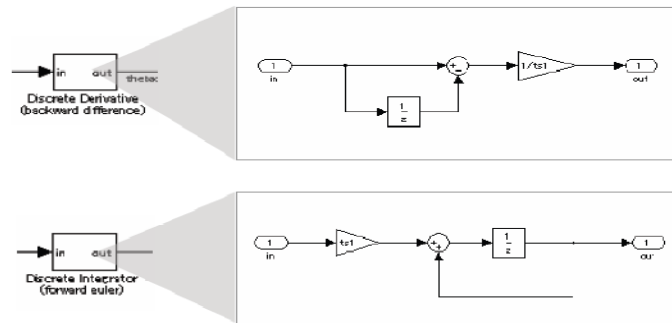
Το υποσύστημα ισορροπίας και ελέγχου οδήγησης είναι ο ελεγκτής που περιγράφεται στο κεφάλαιο 6.2 Σχεδιασμός Ελεγκτή



Σχήμα 5.28 Μοντέλο ελεγκτή για ισορροπία και έλεγχο οδήγησης

5.8.1.1 Block διακριτής Παραγώγου και διακριτού Ολοκληρώματος

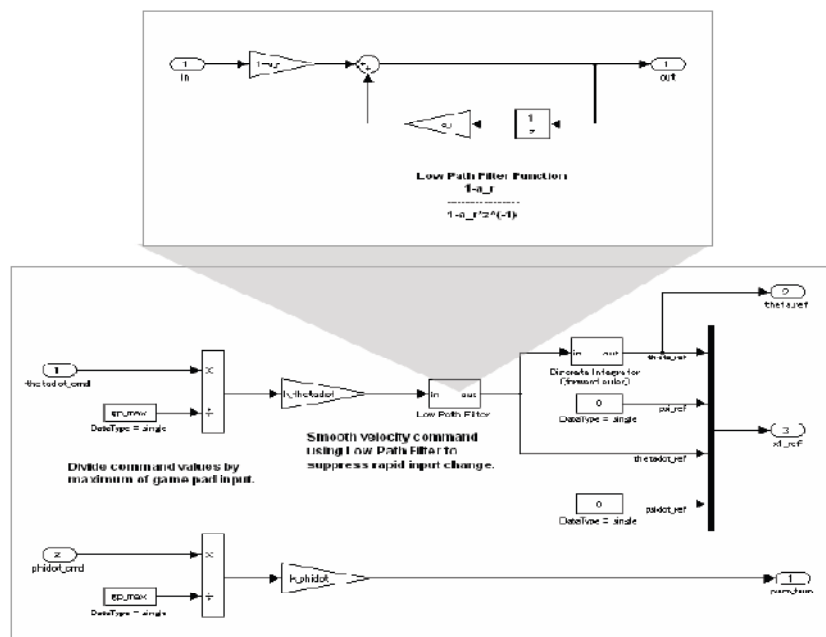
To block διακριτής Παραγώγου υπολογίζει την χρονική παράγωγο με την οπίσθια μέθοδο διαφοράς (backward difference method) και το block διακριτού Ολοκληρώματος κάνει χρονική ολοκλήρωση με την εμπρόσθια μέθοδο Euler (forward Euler method).



Σχήμα 5.29 : Block διακριτής παραγώγου και διακριτού Ολοκληρώματος

5.8.1.2 Υπολογισμός αναφοράς [(Calculation) -Cal Reference subsystem]

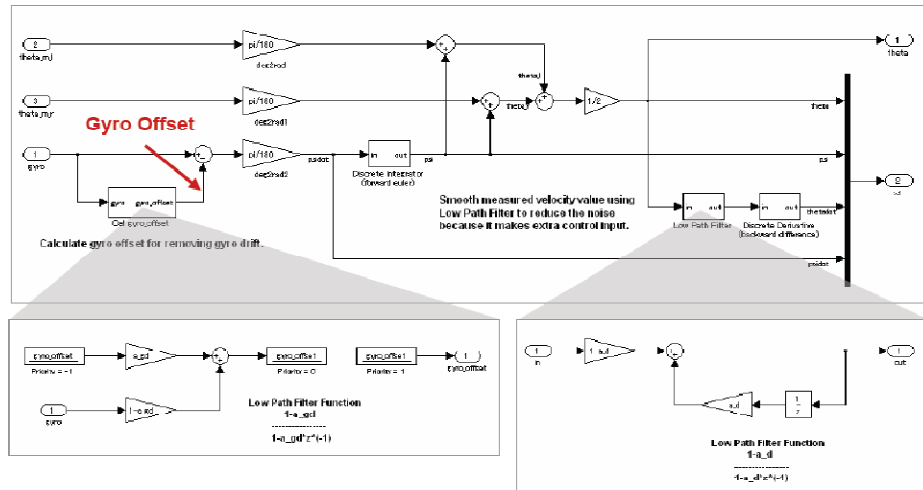
Στο υποσύστημα αυτό εξομαλύνεται η ταχύτητα χρησιμοποιώντας ένα χαμηλοπερατό (low path) φίλτρο για να μειώσει την υπέρβαση που προκαλείται από μια γρήγορη αλλαγή του σήματος αναφοράς της ταχύτητας.



Σχήμα 5.30 : Υπολογισμός Αναφοράς

5.8.1.3 Υπολογισμός Κατάστασης(Cal x1)

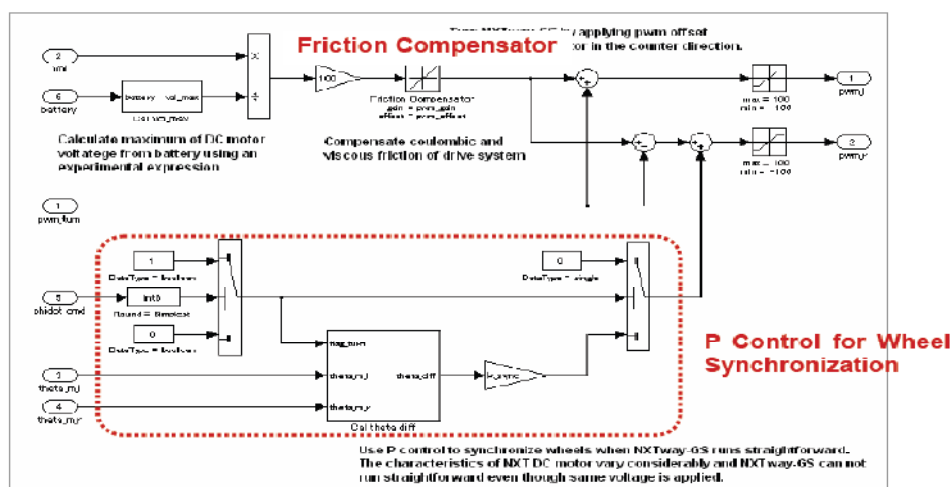
Αυτό το υποσύστημα υπολογίζει την κατάσταση χρησιμοποιώντας τις εξόδους των αισθητήρων. Ως γυροσκοπικό offset χρησιμοποιείται μεγάλης διάρκειας μέσος όρος της γυροσκοπικής εξόδου για την απομάκρυνση του φαινομένου απόκλισης του γυροσκοπικού αισθητήρα (gyro drift effect) και ένα χαμηλοπερατό φίλτρο για την απομάκρυνση του θορύβου από το σήμα ταχύτητας.



Σχήμα 5.31 Υπολογισμός Κατάστασης

5.8.1.4 Υπολογισμός PWM (Cal PWM)

Αυτό το υποσύστημα υπολογίζει την υπηρεσία του PWM που βασίζεται στο block διάγραμμα που παρουσιάζεται στο σχήμα 5.32 και την μέγιστη τάση των κινητήρων χρησιμοποιώντας την εξίσωση (8.1).

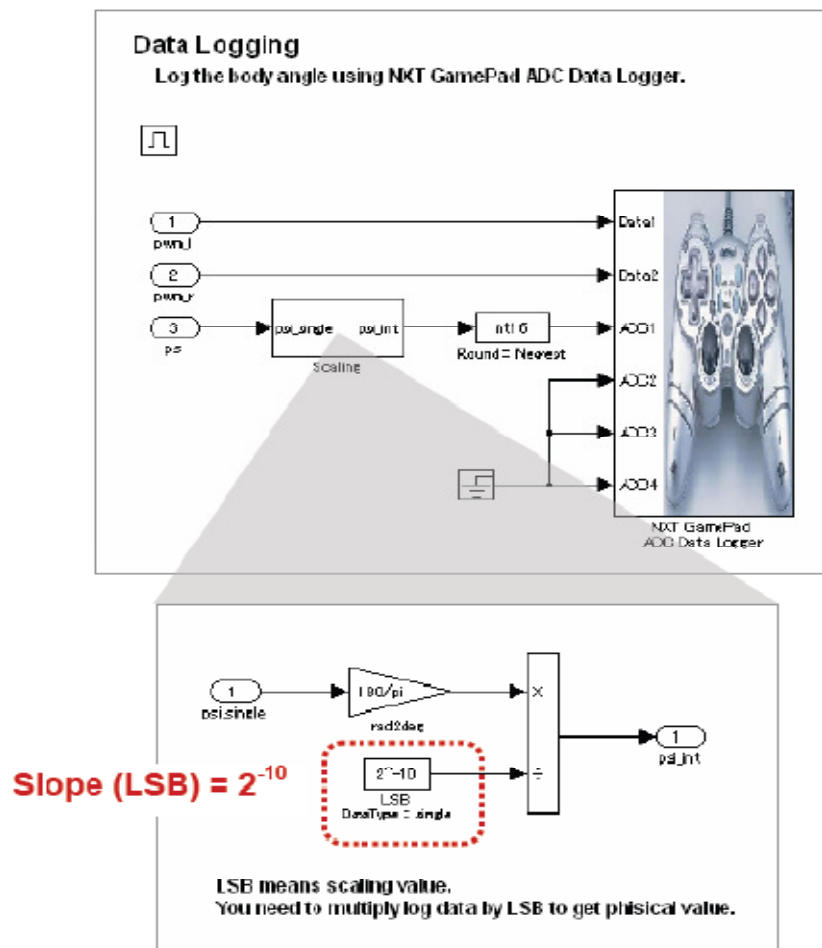


Σχήμα 5.32 : Υπολογισμός PWM

5.8.1.5 Φόρτωση Δεδομένων

Το NXT Gamepad ADC Data logger Block είναι τοποθετημένο σε αυτό το υποσύστημα για την φόρτωση των εξόδων του αισθητήρα και τα αποτελέσματα των υπολογισμών. Για την αλλαγή της περιόδου φόρτωσης, αλλάζτε το log_count που είναι ορισμένη στο αρχείο param_controller.m.

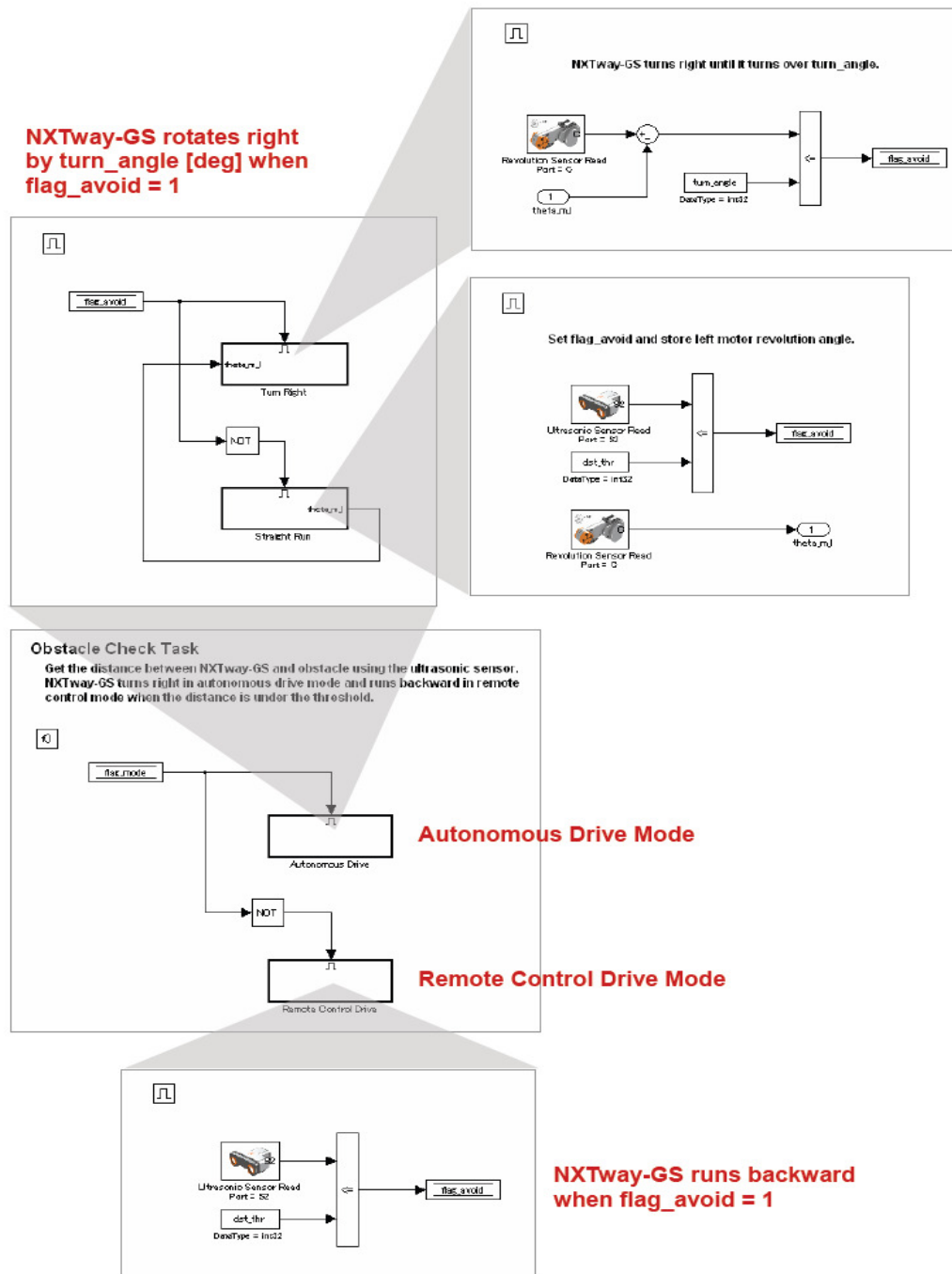
Το NXT Gamepad ADC Data logger Block μπορεί να φορτώσει τις εξόδους του αισθητήρα και τα 8bit και 16bit προσημασμένα ακέραια δεδομένα. Για την φόρτωση δεδομένων κινητής υποδιαστολής είναι απαραίτητη η κβάντιση με την διαίρεση της κατάλληλης τιμής, επειδή δεν υποστηρίζει δεδομένα κινητής υποδιαστολής απευθείας. Ο παρονομαστής που προκύπτει από την διαδικασία της διαίρεσης καλείται ως κλίση ή LSB, αυτό αντιστοιχεί σε μια τιμή ανά 1 μπιτ. Μια αρχική τιμή μπορεί να προσδιοριστεί με τον πολλαπλασιασμό των δεδομένων που έχουν φορτωθεί και της κλίσης (LSB).



Σχήμα 5.33 : Φόρτωση Δεδομένων

5.8.2 20ms task : task_ts2

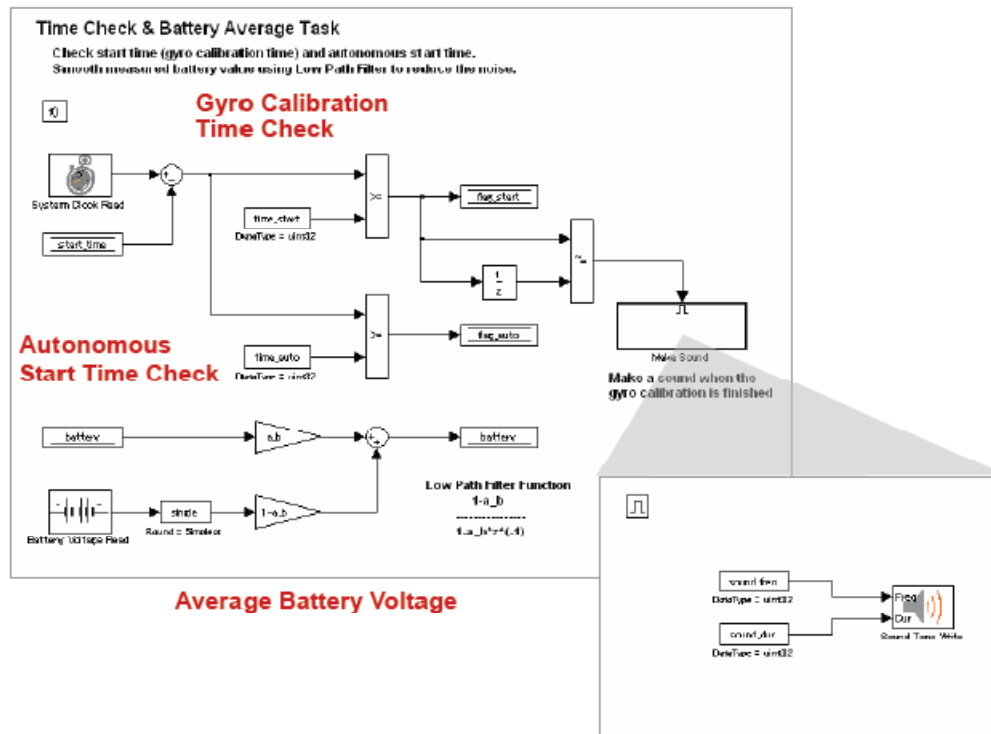
Αυτό το task ελέγχει αν υπάρχουν εμπόδια, με τον αισθητήρα υπέρηχων. Το NXTway-GS περιστρέφεται δεξιά στην αυτόνομη κατάσταση κίνησης και κινείται προς τα πίσω στην κατάσταση κίνησης με τηλεχειρισμό, κατά τον εντοπισμό ενός εμποδίου.



Σχήμα 5.34 :Το υποσύστημα `task_ts2`

5.8.3 100 ms Task : task_ts3

Αυτό είναι ένα task για τον έλεγχο του χρόνου και του μέσου όρου της τάσης της μπαταρίας. Το NXTway-GS ηχεί με διάρκεια `sound_dur[ms]` όταν η ρύθμιση του offset του γυροσκοπικού αισθητήρα τελειώσει.



Σχήμα 5.35 : Το υποσύστημα `task_ts3`

5.8.4 Παράμετροι συντονισμού

Όλες οι παράμετροι που χρησιμοποιούνται στο αρχείο `nxtway_gs_controller.mdl` ορίζονται στο αρχείο `param_controller.m`. Στον πίνακα 5.36 παρουσιάζονται οι παράμετροι συντονισμού για την ισορροπία και τον έλεγχο οδήγησης. Οι παράμετροι αυτοί πρέπει να συντονιστούν επειδή τα μέρη τα blocks, οι αισθητήρες και οι ενεργοποιητές είναι διαφορετικοί μεταξύ τους.

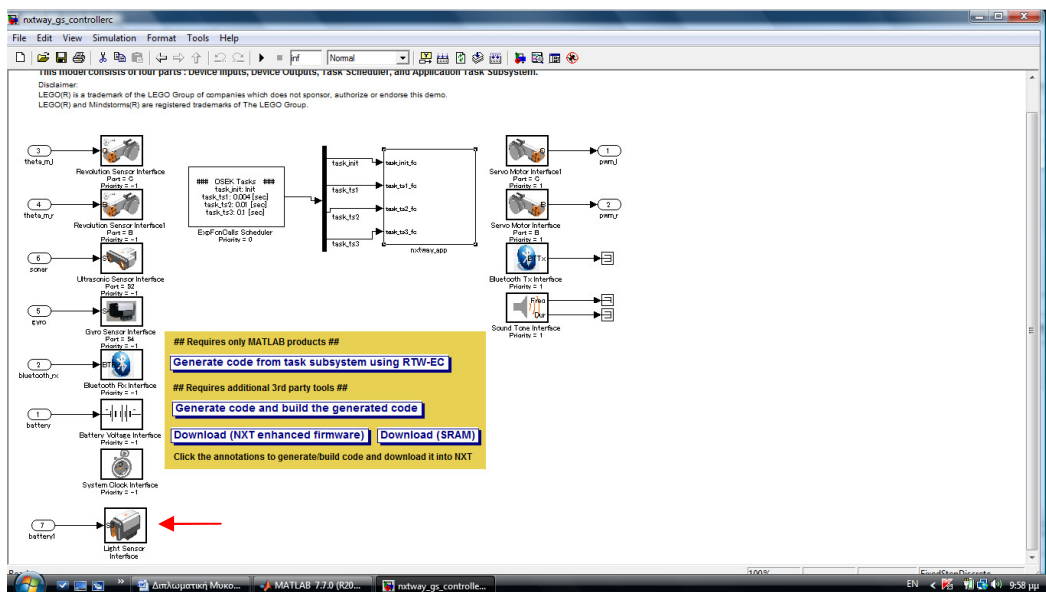
Παράμετρος	Περιγραφή
<code>k_f</code>	Servo control feedback gain
<code>k_i</code>	Servo control integral gain
<code>k_thetadot</code>	Speed reference gain
<code>k_phidot</code>	Rotation speed reference gain
<code>k_sync</code>	P control gain for wheel synchronization
<code>a_b</code>	Low path filter coefficient for averaging battery voltage
<code>a_d</code>	Low path filter coefficient

	for removing velocity noise
a_r	Low path filter coefficient for removing overshoot
a_gc	Low path filter coefficient for gyro calibration
a_gd	Low path filter coefficient for removing gyro drift effect
pwm_gain	Friction compensator gain
pwm_offset	Friction compensator offset
dst_thr	Obstacle avoidance threshold (distance)
turn_angle	Rotation angle in autonomous drive mode

Πίνακας 5.36 Συντονισμός Παραμέτρων

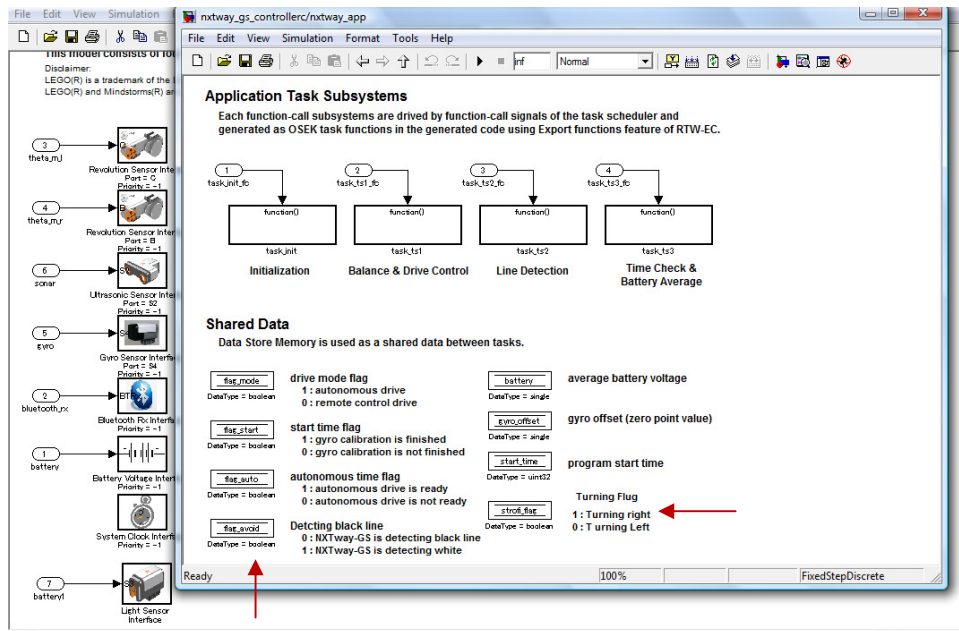
5.9 Εντοπισμός Μαύρης Γραμμής

Για τον εντοπισμό της μαύρης γραμμής προσθήσαμε στον ελεγκτή ένα αισθητήρα φωτός στις εισόδους.



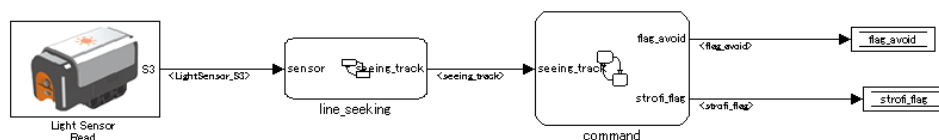
Σχήμα 5.37 Εισαγωγή του αισθητήρα φωτός

Στο Application Task Subsystem στα δεδομένα διαμοιρασμού θεωρήσαμε ότι το flag_avoid για 0 εντοπίζει την μαύρη γραμμή και για 1 εντοπίζει το άσπρο. Επίσης προστέθηκε επιπλέον και το strofi_flag όπου για 1 θα στρίβει δεξιά και για 0 θα στρίβει αριστερά.



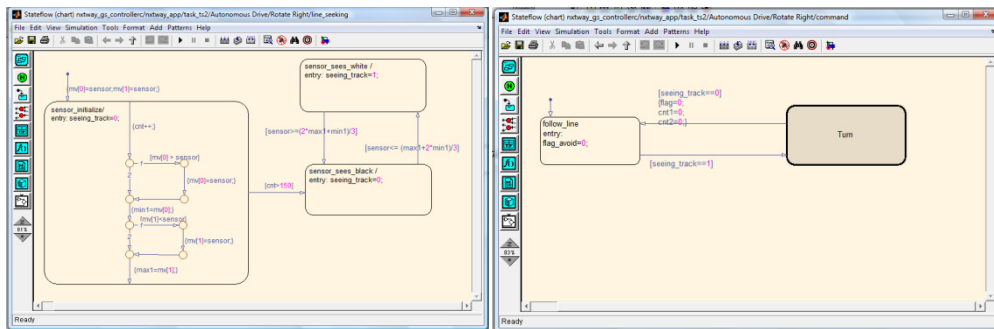
Σχήμα 5.38 Αλλαγές στα δεδομένα διαμοίρασμού

Στο υποσύστημα Line Detection η κατάσταση απομακρυσμένου ελέγχου(remote control)παρκαμένει ως έχει ενώ στο υποσύστημα αυτόνομου ελέγχου για τις περιπτώσεις των δύο υποσυστημάτων Rotate Right και Forward Run εκτελείται ο ακόλουθος αλγόριθμος για τον εντοπισμό της μαύρης γραμμής ώστε το nxtway_gs να κινείται σε αυτήν χωρίς να την χάνει.



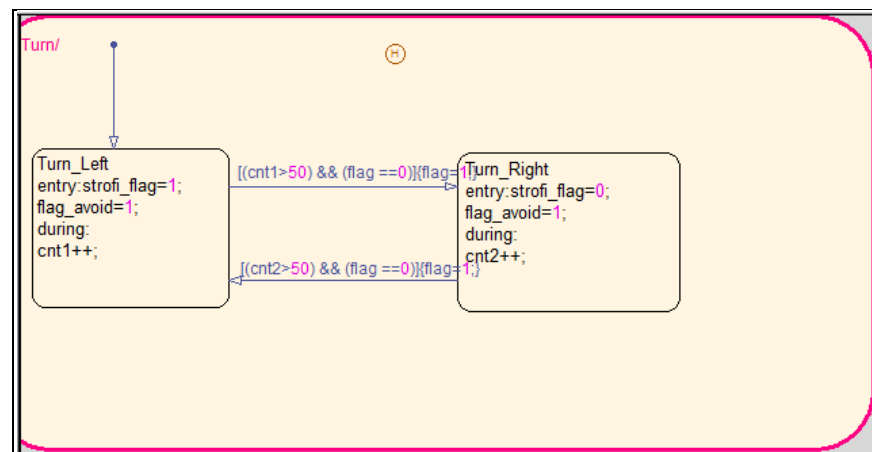
Σχήμα 5.39 Αλγόριθμος εντοπισμού Μαύρης Γραμμής

Ο αισθητήρας φωτός παίρνει δεδομένα τα οποία αποτελούν είσοδο για το υποσύστημα εντοπισμού γραμμής line seeking.Το υποσύστημα Line seeking περιέχει ένα διάγραμμα ροής στο οποίο γίνεται η αρχικοποίηση και ρύθμιση του αισθητήρα φωτός. Όταν ολοκληρωθεί η αρχικοποίηση και η ρύθμιση του αισθητήρα αντιστοιχίζει τα δεδομένα εισόδου για την τιμή seeing_track=1 αν είναι άσπρο και για την τιμή seeing_track=0 αν είναι μαύρο. Βάση πειραματικών μετρήσεων μέσω του Simulink η κλίμακα των τιμών για το μαύρο κυμαίνονται από 0-4 και για το άσπρο από 5-9.



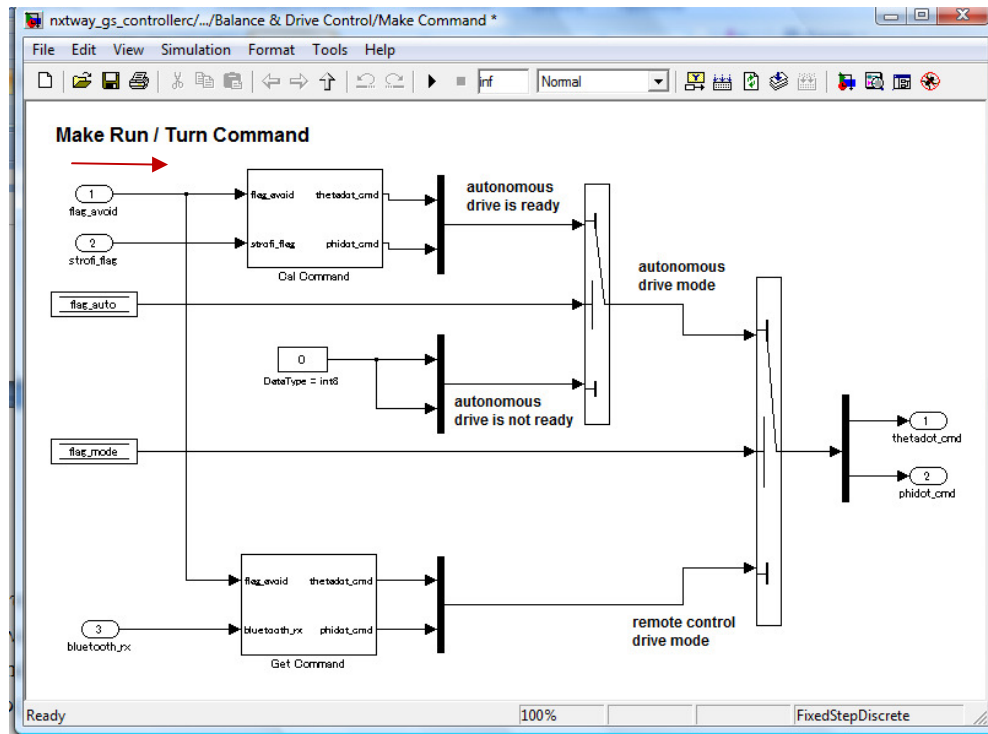
Σχήμα 5.40 Διάγραμμα ροής line seeking και command

Η τιμή του Seeing_track είναι είσοδος του υποσυστήματος command, όπου σε περίπτωση που το seeing track έχει την τιμή 0 τότε σημαίνει ότι βλέπει μαύρο και θα κινηθεί ευθεία, ενώ σε αντίθετη περίπτωση που έχει χάσει την μαύρη γραμμή ή βλέπει άσπρο το seeing_track είναι 1 και γίνεται είσοδος στο διάγραμμα ροής turn όπου το nwtway_gs θα στρίβει αριστερά και δεξιά μέχρις ότου να ξανά εντοπιστεί το μαύρο. Για να επιτευχθεί αυτό θέτουμε εντός του διαγράμματος ροής turn, το strofi_flag=0 και το flag_avoid=1 για αριστερή στροφή και το strofi_flag=1 και flag_avoid=1 για δεξιά στροφή.



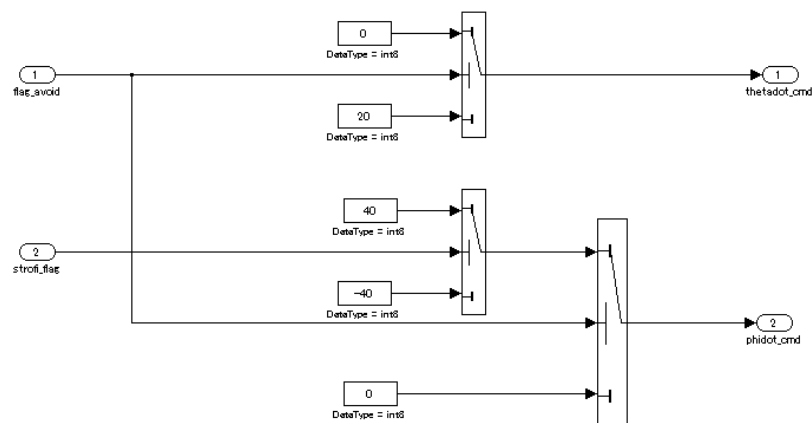
Σχήμα 5.41 Διάγραμμα ροής turn

Τέλος πραγματοποιήθηκαν κάποιες αλλαγές στο υποσύστημα balance & drive control προκειμένου το nwtway_gs να ισορροπεί και ταυτόχρονα να ακολουθεί την μαύρη γραμμή. Στο υποσύστημα Make command προστέθηκε ως είσοδος το strofi_flag το οποίο αποτελεί είσοδος και του υποσυστήματος Cal Command.



Σχήμα 5.42 Υποσύστημα Balance & Drive Control

Για $\text{flag_avoid}=0$, δηλαδή για τον εντοπισμό της μαύρης γραμμής επιλέγεται η κάτω τιμή του διακόπτη και δίνεται η τιμή ταχύτητας $\text{thetadot_cmd}=20$. Σε αντίθετη περίπτωση όπου το $\text{flag_avoid}=1$ δηλαδή όταν βλέπει άσπρο τότε ενεργοποιείται το flag_strofi όπου για 1 στρίβει δεξιά ενώ για 0 στρίβει αριστερά με ταχύτητα phidot_crm 40 και -40 αντίστοιχα.



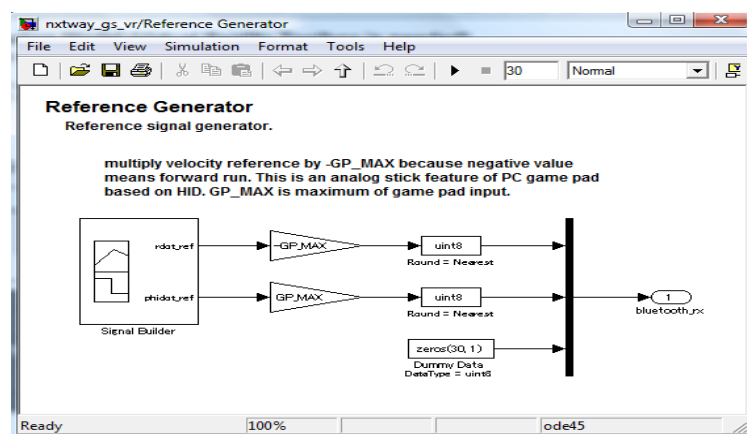
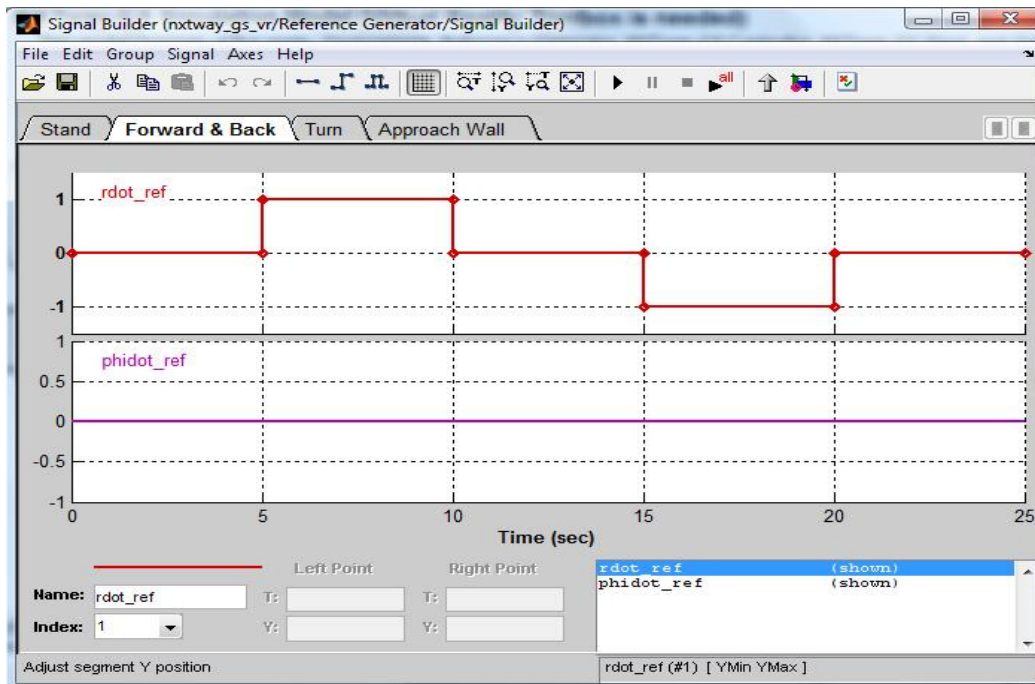
Σχήμα 5.43 Υποσύστημα Call_Command

5.10 Προσομοίωση

Αυτό το κεφάλαιο περιγράφει την προσομοίωση του μοντέλου NXTway_GS και τα αποτελέσματά του τρισδιάστατου παρουσιαστή στο `nxtway_gs_vr.mdl`.

5.10.1 Διαδικασία εκτέλεσης της προσομοίωσης

Η προσομοίωση εκτελείται με τον ίδιο τρόπο που εκτελούνται και τα συνηθισμένα μοντέλα.

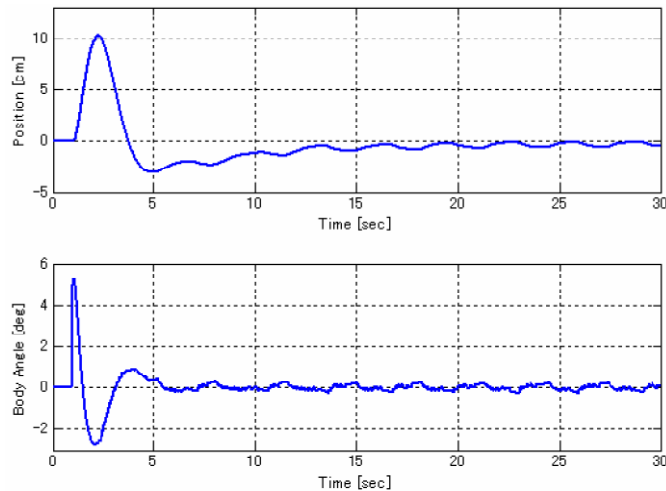


Σχήμα 5.44 : Το υποσύστημα Reference Generator

5.10.2 Αποτελέσματα Προσομοίωσης

5.10.2.1 Στάσιμη Εξισορρόπηση

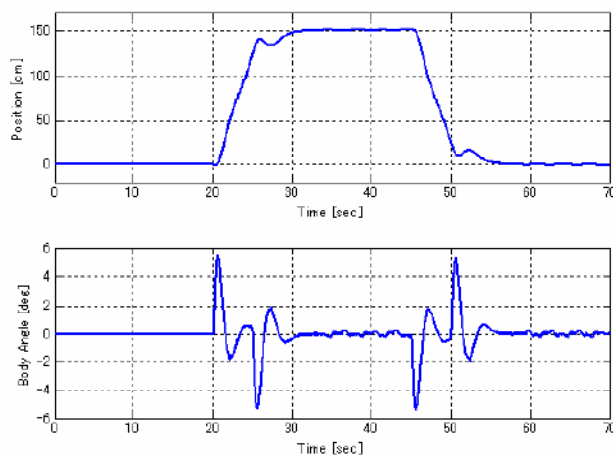
Στο σχήμα 5.45 παρουσιάζετε το αποτέλεσμα προσομοίωσης της στάσιμης εξισορρόπησης στην αρχική τιμή της γωνίας βήματος του σώματος και είναι ίσο με 5 [μοίρες]. Η γωνία βήματος τείνει στις [0 μοίρες] γρήγορα. Το NXTway –GS δεν εκτελείται προτού περάσει το 1[sec] που απαιτείται για την ρύθμιση του γυροσκοπικού αισθητήρα.



Σχήμα 5.45 : Γωνία θέσης και βήματος κατά την στάσιμη εξισορρόπηση

5.10.2.2 Κίνηση προς τα εμπρός και προς τα πίσω

Στο σχήμα 5.46 παρουσιάζονται τα αποτελέσματα της προσομοίωσης από την κίνηση προς τα εμπρός και προς τα πίσω. Η NXTway-GS αυξομειώνεται την χρονική στιγμή που ξεκινάει και σταματάει αλλά σταδιακά τείνει να γίνει σταθερή.

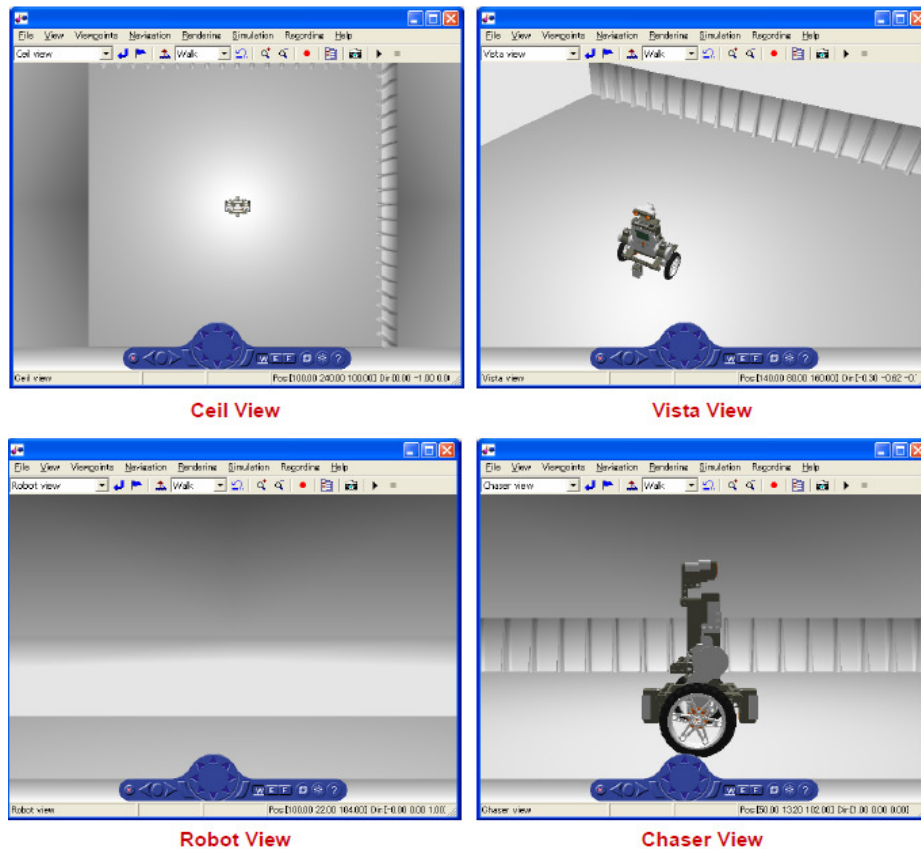


Σχήμα 5.46 Γωνία θέσης και βήματος κατά την κίνηση προς τα εμπρός και προς τα πίσω

5.10.3 Τρισδιάστατος Παρουσιαστής (3D viewer)

Στο αρχείο nxtway_gs_vr.mdl δίνεται η δυνατότητα τρισδιάστατης παρακολούθησης της προσομοίωσης από τον παρουσιαστή(viewer) που παρέχεται από την εργαλειοθήκη εικονικής πραγματικότητας (Virtual Reality Toolbox).Υπάρχει η δυνατότητα λήψης της εικόνας από διαφορετικά οπτικά σημεία αλλάζοντας την κατάσταση παρουσίασης (view mode) στο παράθυρο εικονικής πραγματικότητας . Το nxtway_gs_vr.mdl έχει τέσσερις καταστάσεις παρουσίασης.

- Ceil View : κάτοψη
- Vista View : όψη από σταθερό σημείο αναφοράς
- Robot View : όψη από τον αισθητήρα υπέρηχων του ρομπότ
- Chaser View :πλάγια όψη με κινούμενο σημείο αναφοράς



Σχήμα 5.47 : Καταστάσεις Παρουσίασης

5.11 Παραγωγή Κώδικα και Εφαρμογή

Σε αυτό το κεφάλαιο περιγράφεται ο τρόπος παραγωγής κώδικα από το `nxtway_gs_controller.mdl`, η φόρτωση(download) στο ευφρές τούβλο NXT καθώς επίσης και η παρουσίαση των πειραματικών αποτελεσμάτων.

5.11.1 Υλικό και Λογισμικό (Hardware & Software)

Ο πίνακας 11.1 παρουσιάζει το υλικό(hardware) του Lego mindstorms NXT και το λογισμικό(software) που χρησιμοποιείται στον Embedded Coder Robot NXT.

Hardware	Processor	ATMEL 32-bit ARM 7 (AT91SAM/S256) 48MHz
	Flash Memory	256 Kbytes (10000 time writing guarantee)
	RAM	64 Kbytes
Interface	Actuator	DC motor
	Sensor	Ultrasonic, Touch, Light, Sound
	Display	100 * 64 pixel LCD
	Communication	Bluetooth
Software	RTOS	LEJOS C / LEJOS OSEK
	Compiler	GCC
	Library	GCC library

Πίνακας 11.1 : Χαρακτηριστικά Lego mindstorms NXT & Embedded Coder Robot NXT

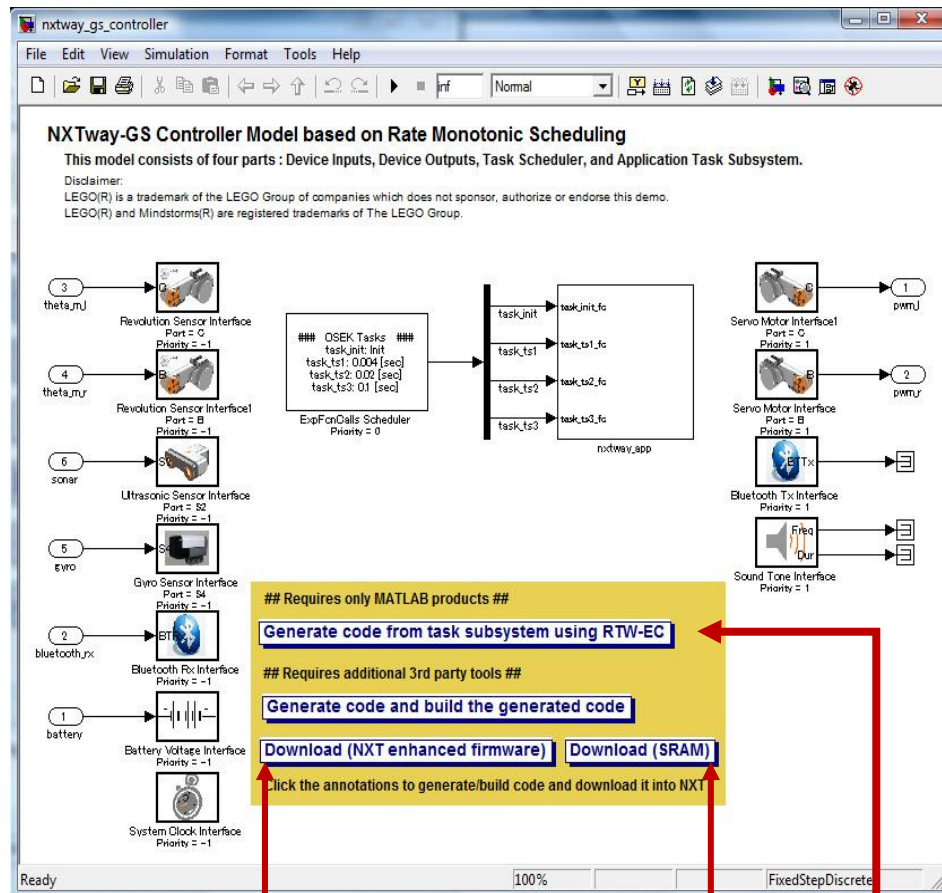
Σημείωση: Δεν είναι δυνατή η φόρτωση ενός προγράμματος όταν το μέγεθος του προγράμματος είναι μεγαλύτερο από την SRAM ή από το μέγεθος της flash memory (θα παρουσιάζεται σφάλμα φόρτωσης).

5.11.2 Τρόπος Παραγωγής και Φόρτωσης Κώδικα

Υπάρχει δυνατότητα παραγωγής κώδικα από το μοντέλο, κατασκευής του(build it), και φόρτωσης του προγράμματος στο NXT κάνοντας κλικ στις υποσημειώσεις του `nxtway_gs_controller.mdl` που παρουσιάζονται στο σχήμα 5.48

Στην συνέχεια ακολουθεί η διαδικασία:

1. Παραγωγή κώδικα και κατασκευή του παραγόμενου κώδικα κάνοντας κλικ στην υποσημείωση **[Generate code and built the generated code]**.
2. Σύνδεση του H/Y και του NXT μέσω καλωδίου USB. Φόρτωση του προγράμματος στο NXT κάνοντας κλικ στην υποσημείωση **[Download (NXT enhanced firmware)]** ή την **[Download (SRAM)]** βάση της κατάστασης εκκίνησης (boot mode) του NXT (enhanced firmware ή SRAM boot).



Βήμα 2.α **Φόρτωση**
προγράμματος **NXT** **enhanced**

Βήμα 1. **Παραγωγή**
Κώδικα και κατασκευή

Βήμα 2.β **Φόρτωση**
Προγράμματος (NXT SRAM)

Σχήμα 5.48: Υποσημειώσεις για παραγωγή κώδικα / κατασκευή / φόρτωση

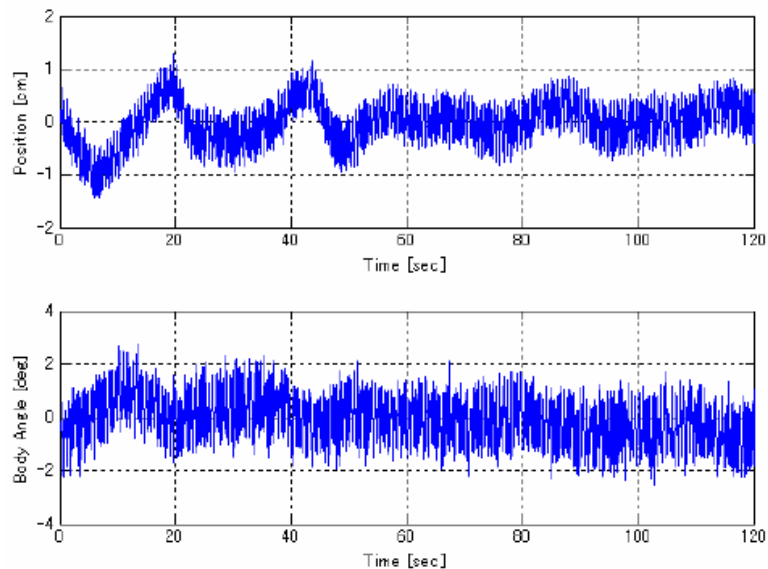
Σημείωση: Το κομμάτι της παραγωγής κώδικα περιγράφεται στο παράρτημα C.

5.11.3 Πειραματικά αποτελέσματα

Στην συνέχεια παρουσιάζονται τα πειραματικά αποτελέσματα. Τα αποτελέσματα που έχουν παραχθεί είναι παρόμοια με τα αποτελέσματα της προσομοίωσης.

5.11.3.1 Στάσιμη Εξισορρόπηση

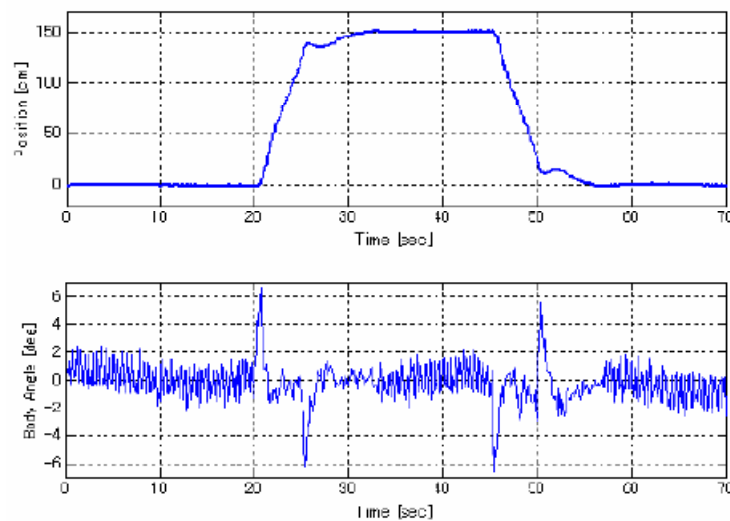
Στο σχήμα 5.49 παρουσιάζονται τα πειραματικά αποτελέσματα της στάσιμης εξισορρόπησης. Το NXTway-GS είναι πολύ σταθερό.



Σχήμα 5.49 Γωνία θέσης και βήματος κατά την στάσιμη εξισορρόπηση

5.11.3.2 Κίνηση προς τα εμπρός και προς τα πίσω

Στο σχήμα 5.50 παρουσιάζονται τα πειραματικά αποτελέσματα της κίνησης προς τα εμπρός και προς τα πίσω. Το σχήμα είναι περίπου το ίδιο με το σχήμα 5.46.



Σχήμα 5.50 Γωνία θέσης και βήματος της κίνησης προς τα εμπρός και προς τα πίσω

5.12 Μοντέλο Ελεγκτή [Αριθμός Σταθερής Υποδιαστολής (Fixed-Point Arithmetic)]

Αυτό το κεφάλαιο περιγράφει την έκδοση σταθερής υποδιαστολής (fixed point) του μοντέλου του ελεγκτή NXTway-GS και εξετάζει τον αντίκτυπο απώλειας και υπερχειρίσης της ακρίβειας στην απόδοση των ελεγκτών.

5.12.1 Ο Αριθμός Σταθερής Υποδιαστολής(Fixed point Number)

Ο αριθμός σταθερής υποδιαστολής(Fixed-Point number) αντιπροσωπεύει την προσέγγιση ενός πραγματικού αριθμού(Real number) στους υπολογισμούς. Τα δεδομένα που χρησιμοποιεί για την αντιπροσώπευση ενός πραγματικού αριθμού είναι ακέραιου τύπου δεδομένα και όχι δεδομένα τύπου κινητής υποδιαστολής(floating-Point data type). Στην ακόλουθη εξίσωση δίνεται ο τρόπος υπολογισμού ενός αριθμού σταθερής υποδιαστολής .

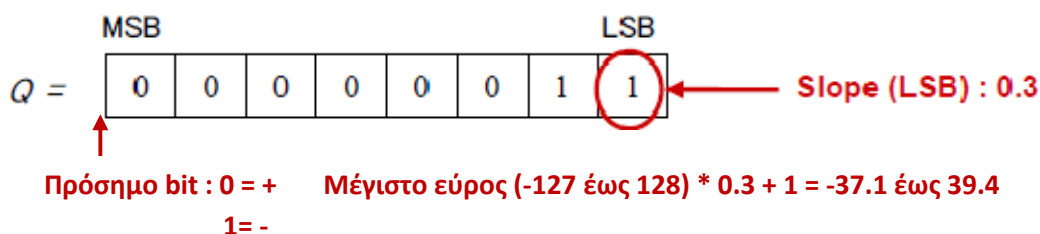
$$V_R \cong V_Q = SQ + B$$

$$V_R : \text{Real number} \quad V_Q : \text{Fixed-point number} \quad Q : \text{Integer} \quad S : \text{Slope} \quad B : \text{Bias}$$

Όπου Slope(η κλίση) αντιπροσωπεύει την πραγματική τιμή ανά ένα bit και το Bias είναι η πραγματική τιμή όταν η ακέραια τιμή (Q) μηδενίζεται. Η Slope καλείται και ως LSB (Less Significant Bit) ή ανάλυση και το Bias ως συμβατικό offset. Στον αριθμό σταθερής υποδιαστολής υπάρχει ένα σφάλμα στρογγυλοποίησης, επειδή αντιπροσωπεύει έναν πραγματικό αριθμό πεπερασμένης ανάλυσης. Επίσης το μέγιστο εύρος που αντιπροσωπεύεται από τον αριθμό σταθερής υποδιαστολής είναι περιορισμένο ανάλογα με το μήκος της λέξης. Το μήκος της λέξης είναι ένα σύνολο αριθμών bit ακέραιου τύπου δεδομένων. Γενικά ο αριθμός σταθερής υποδιαστολής έχει ακρίβεια έναντι του μέγιστου εύρους ανταλλαγής υπό την προϋπόθεση ότι το μήκος της λέξης είναι σταθερό.

Παράδειγμα:

Υπολογισμός του αριθμού σταθερής υποδιαστολής για $V_R = 2$ υπό την προϋπόθεση 8-bit προσημασμένου ακεραίου όπου, $S = 0.3$ και $B = 1$.



Οι αριθμοί σταθερής υποδιαστολής δίνουν την δυνατότητα υπολογισμού πραγματικών αριθμών στον μικροεπεξεργαστή, DSP χωρίς FPU(Floating Point number processing Unit). Γενικά οι αριθμοί σταθερής υποδιαστολής πλεονεκτούν ως προς την ταχύτητα εκτέλεσης από τους αριθμούς κινητής υποδιαστολής. Αντιθέτως όμως έχουν το μειονέκτημα περιορισμένου εύρους όπου μπορεί να εμφανιστεί υπερχειρίση. Στον πίνακα

12.1 παρουσιάζεται η σύγκριση μεταξύ σταθερής υποδιαστολής (fixed-point) και κινητής υποδιαστολής (floating-point).

Παράγοντες	Σταθερής Υποδιαστολής	Κινητής Υποδιαστολής
Ταχύτητα εκτέλεσης	Γρήγορη	Αργή
Κατανάλωση Ισχύος Υλικού	Χαμηλή	Υψηλή
Κατανάλωση RAM / ROM	Μικρή	Μεγάλη
Μέγεθος λέξης και κλιμάκωση	Προσαρμοστικότητα	Μη προσαρμοστικότητα
Μέγιστο Εύρος	Περιορισμένο	Ευρύ
Επιρρεπή σε σφάλμα	Υψηλό	Χαμηλό
Χρόνος Ανάπτυξης	Μακράς διάρκειας	Μικρής διάρκειας

Πίνακας 12.1 : Σταθερή Υποδιαστολή έναντι Κινούμενης Υποδιαστολής

5.12.2 Μετατροπή Κινητής Υποδιαστολής σε Σταθερή.

Η εργαλειοθήκη σταθερής υποδιαστολής / Simulink fixed point επιτρέπει εσωτερικές δυνατότητες σταθερής υποδιαστολής των προϊόντων της MATLAB/SIMULINK. Περισσότερες πληροφορίες σχετικά με την μοντελοποίηση της σταθερής υποδιαστολής στην αναφορά [3].

Το αρχείο `nxtway_gs_controller_fixpt.mdl` είναι η έκδοση σταθερής υποδιαστολής του `nxtway_gs_controller.mdl`. Χρησιμοποιεί σταθερής υποδιαστολής τύπο δεδομένων για την γωνία τροχού (wheel angle), την γωνία βήματος του σώματος και την μεταβλητή της μπαταρίας αντί του ενιαίου τύπου δεδομένων ακρίβειας. Τα κυριότερα σημεία της μετατροπής της σταθερής υποδιαστολής είναι τα ακόλουθα.

5.12.2.1 Τύπος Δεδομένων Σταθερής Υποδιαστολής

Το αρχείο `param_controller_fixpt.m` ορίζει τα αριθμητικού τύπου αντικείμενα Simulink, που χρησιμοποιούνται στο `nxtway_gs_controller_fixpt.mdl`. Τα αριθμητικά δεδομένα Simulink (Simulink Numeric Type) καθορίζουν οποιοδήποτε τύπο δεδομένων για παράδειγμα, ακέραιους, κινούμενης υποδιαστολής και σταθερής υποδιαστολής.

`param_controller_fixpt.m`

```
% Fixed-Point Parameters
% We can calculate how far NXTway-GS can move by using the following equation

% >> double(intmax('int32')) * pi / 180 * R * S
% where R is the wheel radius and S is the slope of dt_theta (S = 2^-14 etc.)

%
```

% S : Range [m]
% 2 ⁻¹⁰ : 1464.1
% 2 ⁻¹⁴ : 91.5
% 2 ⁻¹⁸ : 5.7
% 2 ⁻²² : 0.36
% Simulink.NumericType for wheel angle
% signed 32-bit integer, slope = 2 ⁻¹⁴ , bias = 0
dt_theta = fixdt(true, 32, 2 ⁻¹⁴ , 0);
% Simulink.NumericType for body pitch angle
% signed 32-bit integer, slope = 2 ⁻²⁰ , bias = 0
dt_psi = fixdt(true, 32, 2 ⁻²⁰ , 0);
% Simulink.NumericType for battery
% signed 32-bit integer, slope = 2 ⁻¹⁷ , bias = 0
dt_battery = fixdt(true, 32, 2 ⁻¹⁷ , 0);

Γενικά η απόδοση του ελεγκτή εξαρτάται από τους τύπους δεδομένων που χρησιμοποιούνται σε αυτόν. Υπάρχει μία ανταλλαγή μεταξύ της ακρίβειας και του εύρους κατά την χρήση του ίδιου μήκους λέξης που διευκρινίζεται στο Κεφάλαιο 12.1. Είναι απαραίτητο για έναν σχεδιαστή ελεγκτών να θεωρήσει αυτήν την ανταλλαγή προκειμένου να βρεθούν οι καλύτερες ρυθμίσεις σταθερής υποδιαστολής.

Στην περίπτωση του NXTway-GS οι ρυθμίσεις σταθερής υποδιαστολής για την γωνία τροχού και την γωνία βήματος του σώματος είναι πολύ σημαντικές επειδή το NXTway-GS δεν θα μπορεί να ισορροπήσει αν υπάρξει απώλεια ακρίβειας ή εμφάνιση υπερχείλισης.

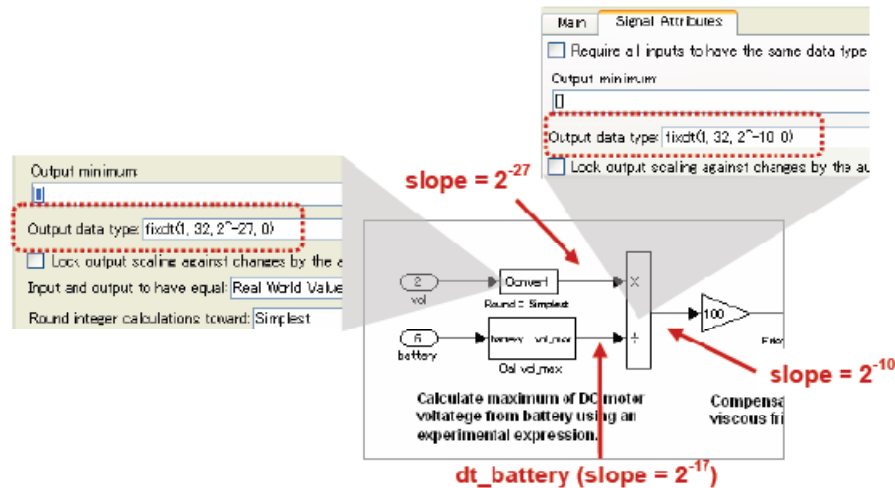
Μπορούμε να υπολογίσουμε πόσο μακριά μπορεί να μετακινηθεί το NXTway-GS από την ακόλουθη εξίσωση:

$$>> \text{double}(\text{intmax}('int32')) * \pi / 180 * R * S$$

Όπου R είναι η ακτίνα του τροχού, και S είναι η κλίση(slope) της γωνίας dt_theta. Στον πίνακα 12.2 παρουσιάζεται η μέγιστη απόσταση μετακίνησης του NXTway-GS χρησιμοποιώντας διάφορες κλίσεις.

Κλίση dt_theta	2 ⁻¹⁰	2 ⁻¹⁴	2 ⁻¹⁸	2 ⁻²²
Μέγιστη Απόσταση [m]	1464.1	91.5	5.7	0.36

Πίνακας 12.2 : Μέγιστη Απόσταση Κίνησης NXTway-GS



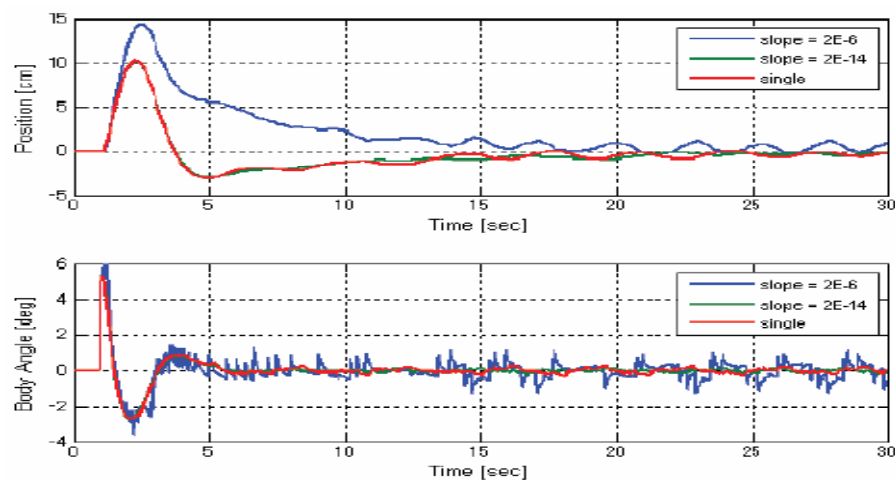
Σχήμα 5.53 Ρύθμιση κλίσης στον υπολογισμό του PWM

15.12.3 Αποτελέσματα Προσομοίωσης

Για την προσομοίωση του μοντέλου `nxtway_gs_controller_fixpt.mdl` είναι αναγκαία η αλλαγή του μοντέλου αναφοράς του ελεγκτή από `nxtway_gs.mdl` στο μοντέλο `nxtway_gs_vr.mdl`.

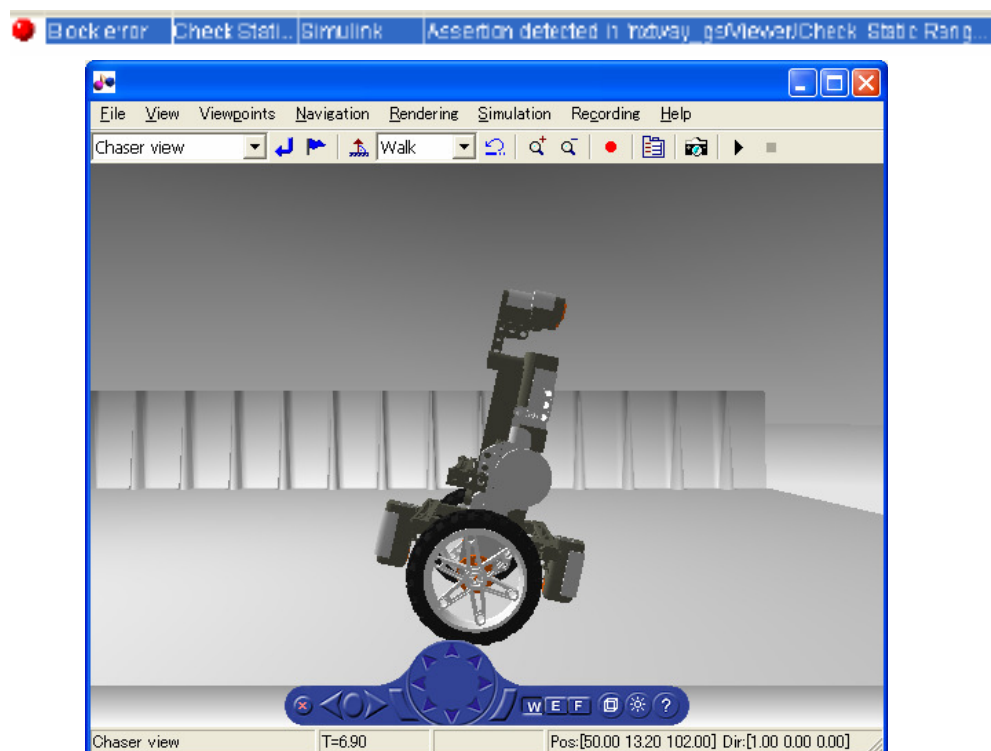
Για να αλλάξετε το μοντέλο αναφοράς κάντε δεξί κλικ στο block, επιλέξτε **[Model Reference Parameter]** και **edit [Model Name]**.

Παρατηρείτε ότι η απόδοση του ελεγκτή ποικίλει ανάλογα με την ακρίβεια του τύπου δεδομένων σταθερής υποδιαστολής (κλίση) με τη χρήση διαφόρων τιμών κλίσης της γωνίας των τροχών (`dt_theta`). Στο σχήμα 5.54 παρουσιάζονται τα αποτελέσματα προσομοίωσης σταθερής εξισορρόπησης για διαφορετικές τιμές κλίσης της `dt_theta` (η αρχική τιμή της γωνίας βήματος του σώματος είναι [5 μοίρες]). Οι χρόνος σύγκλισης και η διακύμανση είναι ανάλογα με την κλίση. Το NXTway-GS δεν μπορεί να ισορροπήσει αν η κλίση είναι μεγαλύτερη του 2^{-5} λόγω απώλειας της ακρίβειας.



Σχήμα 5.54 Αποτέλεσμα προσομοίωσης με διαφορετική κλίση ($2E-6=2^{-6}$)

Αφ' ενός, μικρότερη κλίση σημαίνει και συντομότερη μέγιστη απόσταση μετακίνησης του NXTway-GS. Για παράδειγμα όταν η κλίση είναι ρυθμισμένη στο 2^{-22} , το NXTway-GS θα πέσει αφού διανύσει απόσταση 0.36[m] λόγω της υπερχειλίσσης (κοίτα πίνακα 12.2). Το σχήμα 5.55 είναι το στιγμιότυπο της εικόνας του NXTway-GS την στιγμή που πέφτει από υπερχειλίσση.



Σχήμα 5.55 Το NXTway-GS πέφτει λόγω υπερχειλίσσης

5.12.4 Παραγωγή Κώδικα και Πειραματικά Αποτελέσματα

Η διαδικασία παραγωγής κώδικα και φόρτωσης του προγράμματος από το `nxtway_gs_controller_fixpt.mdl` είναι παρόμοια με αυτήν του `nxtway_gs_controller.mdl`. Ο παραγόμενος κώδικας χρησιμοποιεί μόνο ακέραιου τύπου δεδομένα. Παρατηρούνται τα ίδια αποτελέσματα με την προσομοίωση (απόδοση ελεγκτή και υπερχειλίσση) όταν φορτώνονται οι κώδικες σταθερής υποδιαστολής στο ευφρές τούβλο NXT.

Παράρτημα Α : Σύγχρονη θεωρία Συστημάτων Αυτόματου Ελέγχου

Αυτό το παράρτημα περιγράφει τη σύγχρονη θεωρία συστημάτων αυτόματου ελέγχου που χρησιμοποιείται στο σχεδιασμό του ελεγκτή του NXTway-GS εν συντομία. Για περισσότερες λεπτομέρειες σχετικά με την θεωρία συστημάτων αυτόματου ελέγχου στα εγχειρίδια.

A.1 Σταθερότητα

Καθορίζουμε ένα σύστημα το οποίο είναι ασυμπτωτικά σταθερό όταν η κατάσταση τείνει στο μηδέν ανεξάρτητα με το αν η αρχική τιμή εισαγωγής u είναι ίση με το μηδέν.

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = 0 \quad (\text{A.1})$$

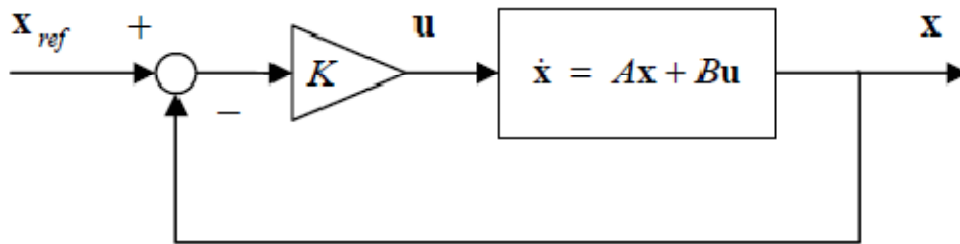
Η καταστατική εξίσωση δίδεται ως :

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (\text{A.2})$$

Αναγκαία ικανή συνθήκη ώστε να είναι ασυμπτωτικά σταθερό είναι ότι το πραγματικό μέρος όλων των ιδιοτήτων της μήτρας \mathbf{A} του συστήματος είναι αρνητικό. Το σύστημα είναι ασταθές όταν έχει κάποιες ιδιοτιμές με θετικό πραγματικό μέρος.

A.2 Κατάσταση Ελέγχου Ανατροφοδότησης

Η κατάσταση ελέγχου ανατροφοδότησης είναι μια τεχνική ελέγχου που πολλαπλασιάζει το κέρδος ανάδρασης K και την απόκλιση μεταξύ της αναφερόμενης τιμής της κατάστασης \mathbf{x}_{Ref} και της μετρίσιμης τιμής \mathbf{x} και τα ανατροφοδοτεί στο σύστημα. Είναι παρόμοια με τον έλεγχο PD στην κλασική θεωρία ελέγχου. Στο σχήμα A.1 παρουσιάζεται το διάγραμμα block της κατάστασης ελέγχου ανατροφοδότησης.



Σχήμα A.1: Block Διάγραμμα Κατάστασης Ελέγχου Ανατροφοδότησης

Οι εισοδοι και η καταστατική εξίσωση που παρουσιάζεται στο σχήμα A.1 δίνονται στη συνέχεια:

$$\mathbf{u}(t) = -K(\mathbf{x}(t) - \mathbf{x}_{\text{ref}}) \quad (\text{A.3})$$

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) + \mathbf{BK}\mathbf{x}_{\text{ref}} \quad (\text{A.4})$$

Το σύστημα μπορεί να γίνει σταθερό ρυθμίζοντας την ανατροφοδότηση του κέρδους K λόγω του ότι μπορεί να γίνει αλλαγή των ιδιοτιμών του συστήματος πινάκων $\mathbf{A}-\mathbf{BK}$.

Είναι απαραίτητη η χρήση κατάστασης ελέγχου ανατροφοδότησης έτσι ώστε το σύστημα να είναι ελεγχόμενο. Ικανή και αναγκαία συνθήκη ώστε να είναι ελεγχόμενος ο πίνακας M_c είναι να είναι πλήρως ταξινομημένος (full rank). ($\text{rank}(M_c) = n$, όπου το n είναι η τάξη του x)

$$M_c = [B, AB, \dots, A^{n-1}B] \quad (A.5)$$

Η εργαλειοθήκη συστημάτων ελέγχου (Control System Toolbox) παρέχει την συνάρτηση **ctrb** για την εκτίμηση του ελεγχόμενου πίνακα.

Παράδειγμα

Είναι το επόμενο σύστημα ελεγχόμενο; $A = [0, 1; -2, -3]$, $B = [0; 1] \rightarrow$ ελεγχόμενο

```
>> A = [0, 1; -2, -3]; B = [0; 1];
>> Mc = ctrb(A, B);
>> rank(Mc)
ans =
     2
```

Υπάρχουν δύο τρόποι για τον υπολογισμό της ανάδρασης του κέρδους K .

1. Άμεση τοποθέτηση Πόλου

Αυτή η μέθοδος υπολογίζει την ανάδραση του κέρδους K , καθορίζοντας τους πόλους (ιδιοτιμές) του συστήματος πινάκων A - BK στις επιθυμητές θέσεις. Οι παράμετροι ρύθμισης είναι οι πόλοι και η απόφαση για την κατάλληλη τιμή του κέρδους γίνεται με την μέθοδο δοκιμής σφάλματος. Η εργαλειοθήκη συστημάτων ελέγχου (Control System Toolbox) παρέχει την συνάρτηση **place** για την άμεση τοποθέτηση πόλου.

Παράδειγμα

Υπολογισμός ανάδρασης κέρδους για το σύστημα $A=[0,1; -2, -3]$, $B=[0;1]$ ώστε να καθοριστούν οι πόλοι σε $[-5,-6]$.

```
>> A = [0, 1; -2, -3]; B = [0; 1];
>> poles = [-5, -6];
>> K = place(A, B, poles)
K =
    23.0000    8.0000
```

2. Γραμμικός Δευτεροβάθμιος Ρυθμιστής (linear Quadratic Regulator)

Αυτή η μέθοδος υπολογίζει την ανάδραση του κέρδους K , ώστε να ελαχιστοποιείται το κόστος της συνάρτησης J που δίνεται στη συνέχεια ως:

$$J = \int_0^{\infty} (\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t)) dt$$

Οι ρυθμιστικές παράμετροι είναι πίνακες βαρών για την κατάσταση Q και για την είσοδο R . Η απόφαση για την κατάλληλη τιμή του κέρδους γίνεται με την μέθοδο δοκιμής σφάλματος. Η εργαλειοθήκη συστημάτων ελέγχου (Control System Toolbox) παρέχει την συνάρτηση **lqr** για τον γραμμικό δευτεροβάθμιο ρυθμιστή.

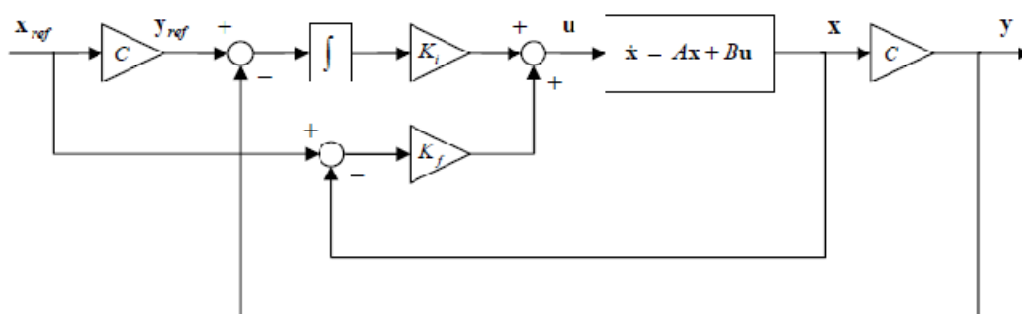
Παράδειγμα

Υπολογισμός ανάδρασης κέρδους για το σύστημα $A = [0, 1; -2 \ -3]$, $B = [0; 1]$ χρησιμοποιώντας $Q = [100, 0; 0, 1]$, $R = 1$

```
>> A = [0, 1; -2, -3]; B = [0; 1];
>> Q = [100, 0; 0, 1]; R = 1;
>> K = lqr(A, B, Q, R)
K =
    8.1980    2.1377
```

A.3 Σέρβο- Έλεγχος (Servo Control)

Ο σέρβο έλεγχος είναι μια μέθοδος ελέγχου ώστε η έξοδος από το σύστημα να αντιμετωπίζει μία αναμενόμενη συμπεριφορά. Ο έλεγχος PID (Αναλογικός-Ολοκληρωτικός-Διαφορικός Ελεγκτής) και ο έλεγχος I-PD είναι ένα είδος ελέγχου του σέρβο στην κλασική θεωρία ελέγχου. Αναγκαίο είναι να προστεθεί ένας ολοκληρωτής στον κλειστό βρόγχο ώστε να οι έξοδοι να εντοπίζουν καθορισμένο βήμα αναφοράς. Στο σχήμα A.2 παρουσιάζεται το διάγραμμα block του servo-ελέγχου(τύπου PID).



Σχήμα A.2 : Διάγραμμα Block σέρβο ελέγχου(τύπου PID)

Μπορεί να υπολογιστούν τα κέρδη του σέρβο ελέγχου με τον ίδιο τρόπο του ελέγχου ανάδρασης λαμβάνοντας δεδομένο ένα διευρυμένο σύστημα. Αυτό έχει μία καινούργια κατάσταση η οποία είναι η διαφορά ανάμεσα στην έξοδο και την αναφερόμενη τιμή. Τα κέρδη του σέρβο ελέγχου παραγωγίζονται στη συνέχεια.

Η καταστατική εξίσωση και η συνάρτηση εξόδου δίνονται στην συνέχεια ως:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \tag{A.6}$$

Χρησιμοποιείται η διαφορά $e(t) = \mathbf{C}(\mathbf{x}(t) - \mathbf{x}_{ref})$, το ολοκλήρωμα της διαφοράς $z(t)$, καταστάσεις του διευρυμένου συστήματος $\bar{\mathbf{x}}(t) = [\mathbf{x}(t), z(t)]^T$ και οι τάξεις του \mathbf{x} και του \mathbf{u} είναι n και m αντίστοιχα. Η καταστατική εξίσωση του διευρυμένου συστήματος είναι η ακόλουθη.

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix} = \begin{bmatrix} A & 0_{n \times m} \\ C & 0_{m \times n} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{z}(t) \end{bmatrix} + \begin{bmatrix} B \\ 0_{m \times n} \end{bmatrix} \mathbf{u}(t) - \begin{bmatrix} 0_{n \times n} \\ I_{m \times m} \end{bmatrix} C \mathbf{x}_{ref} \quad (\text{A.7})$$

Η εξίσωση (A.7) συγκλίνει με την εξίσωση (A.8) αν το διευρυμένο σύστημα θεωρείται σταθερό.

$$\begin{bmatrix} \dot{\mathbf{x}}(\infty) \\ \dot{\mathbf{z}}(\infty) \end{bmatrix} = \begin{bmatrix} A & 0_{n \times m} \\ C & 0_{m \times n} \end{bmatrix} \begin{bmatrix} \mathbf{x}(\infty) \\ \mathbf{z}(\infty) \end{bmatrix} + \begin{bmatrix} B \\ 0_{m \times n} \end{bmatrix} \mathbf{u}(\infty) - \begin{bmatrix} 0_{n \times n} \\ I_{m \times m} \end{bmatrix} C \mathbf{x}_{ref} \quad (\text{A.8})$$

Η καταστατική εξίσωση (A.9) παράγεται λαμβάνοντας υπόψη την αφαίρεση των εξισώσεων μεταξύ της εξίσωσης (A.7) και της (A.8).

$$\begin{bmatrix} \dot{\mathbf{x}}_e(t) \\ \dot{\mathbf{z}}_e(t) \end{bmatrix} = \begin{bmatrix} A & 0_{n \times m} \\ C & 0_{m \times n} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(t) \\ \mathbf{z}_e(t) \end{bmatrix} + \begin{bmatrix} B \\ 0_{m \times n} \end{bmatrix} \mathbf{u}_e(t) \rightarrow \frac{d}{dt} \bar{\mathbf{x}}_e(t) = \bar{A} \bar{\mathbf{x}}_e(t) + \bar{B} \mathbf{u}_e(t) \quad (\text{A.9})$$

$$\text{Όπου : } \mathbf{x}_e = \mathbf{x}(t) - \mathbf{x}(\infty), \quad \mathbf{z}_e = \mathbf{z}(t) - \mathbf{z}(\infty), \quad \mathbf{u}_e = \mathbf{u}(t) - \mathbf{u}(\infty).$$

Μπορεί να χρησιμοποιηθεί το σύστημα ελέγχου ανάδρασης ώστε να γίνει το διευρυμένο

$$\mathbf{u}_e(t) = -K \bar{\mathbf{x}}_e(t) = -K_f \mathbf{x}_e(t) - K_i \mathbf{z}_e(t) \quad (\text{A.10})$$

σύστημα σταθερό. Η είσοδος είναι :

Αν θεωρήσουμε $\mathbf{x}(\infty) \rightarrow \mathbf{x}_{ref}$, $\mathbf{z}(\infty) \rightarrow 0$, $\mathbf{u}(\infty) \rightarrow 0$ παράγουμε την είσοδο $\mathbf{u}(t)$ ως εξής:

$$\mathbf{u}(t) = -K_f(\mathbf{x}(t) - \mathbf{x}_{ref}) - K_i C \int (\mathbf{x}(t) - \mathbf{x}_{ref}) dt \quad (\text{A.11})$$

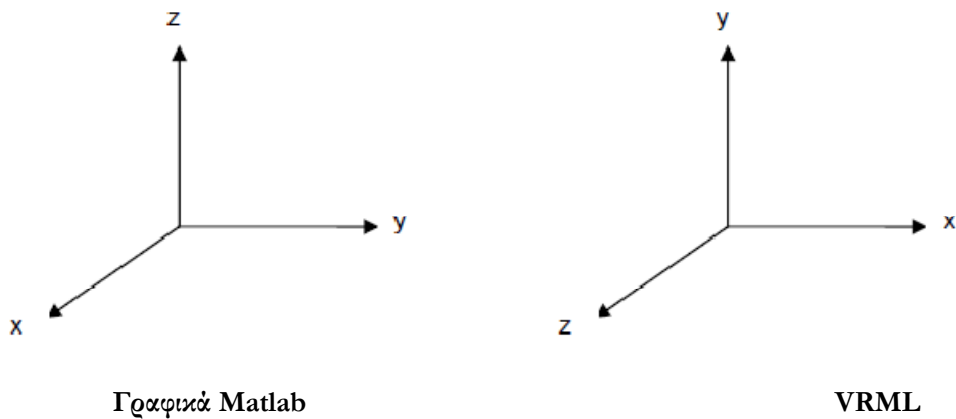
Τα βιβλία της σύγχρονης θεωρίας ελέγχου περιγράφουν συνήθως τον τύπο έκφρασης I-PD (\mathbf{x}_{ref} στον πρώτο όρο της εξίσωσης (A.11) ίσο με μηδέν) ως είσοδο σέρβο ελέγχου.

Παράρτημα Β : Χώρος Εικονικής Πραγματικότητας

Σε αυτό το παράρτημα περιγράφεται ο χώρος εικονικής πραγματικότητας που χρησιμοποιείται στο μοντέλο NXTway-GS. Για παράδειγμα το σύστημα συντεταγμένων, το αρχείο χάρτη(map file), και ο εντοπισμός πρόσδεσης σε τοίχο.

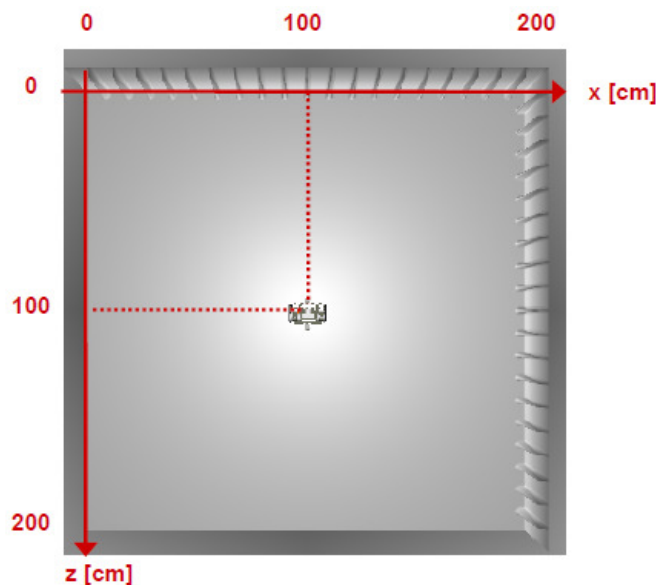
B.1 Σύστημα Συντεταγμένων

Χρησιμοποιείται η εργαλειοθήκη εικονικής πραγματικότητας (Virtual Reality Toolbox) για τρισδιάστατη απεικόνιση του NXTway-GS. Η εργαλειοθήκη εικονικής πραγματικότητας (Virtual Reality Toolbox) απεικονίζει ένα αντικείμενο που βασίζεται στην γλώσσα VRML. Το σύστημα συντεταγμένων VRML ορίζεται όπως φαίνεται στο σχήμα B.1.



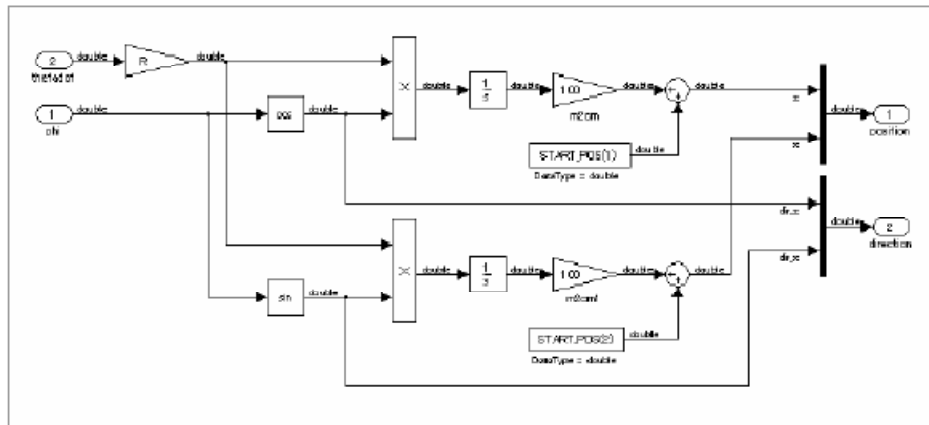
Σχήμα B.1 : Σύστημα συντεταγμένων Matlab και VRML

Στο σχήμα B.2 παρουσιάζεται το σύστημα συντεταγμένων που ορίζεται από το αρχείο track.wrl που είναι ένα αρχείο χάρτη(map file) γραμμένο σε VRML.



Σχήμα B.2 : Το σύστημα συντεταγμένων που ορίζεται από το αρχείο track.wrl

Η VRML θέση του NXTway-GS υπολογίζεται στο NXTway-GS/ Sensor/ Cal VRML υποσύστημα συντεταγμένων (Coordinate Subsystem).



Σχήμα B.3 : Υποσύστημα Συντεταγμένων Cal VRML

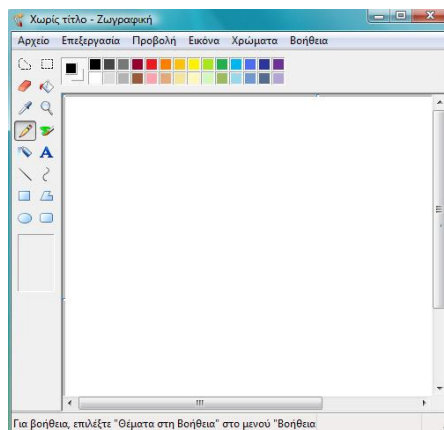
B.2 Δημιουργία ενός Αρχείου Χάρτη (Map File)

Το αρχείο track.wrl μπορεί να δημιουργηθεί από το αρχείο track.bmp χρησιμοποιώντας το αρχείο mywritevrtrack.m. Πληκτρολογώντας την ακόλουθη εντολή δημιουργείται το αρχείο track.wrl.

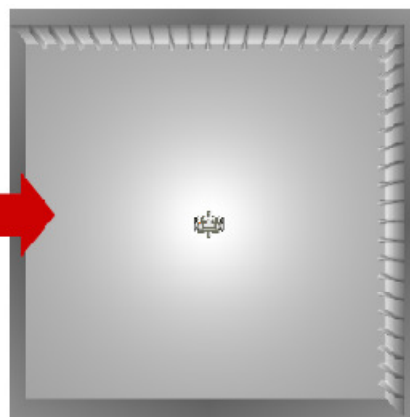
```
>> mywritevrtrack('track.bmp')
```

Το αρχείο mywritevrtrack.m δημιουργεί το αρχείο track.wrl με την χρήση των κανόνων μετατροπής που ακολουθούν στην συνέχεια.

- 1 pixel → 1 * 1 [cm²]
- white pixel (RGB = [255, 255, 255]) → floor
- gray pixel (RGB = [128, 128, 128]) → wall (default height is 20 [cm])
- black pixel (RGB = [0, 0, 0]) → black line



Track.bmp

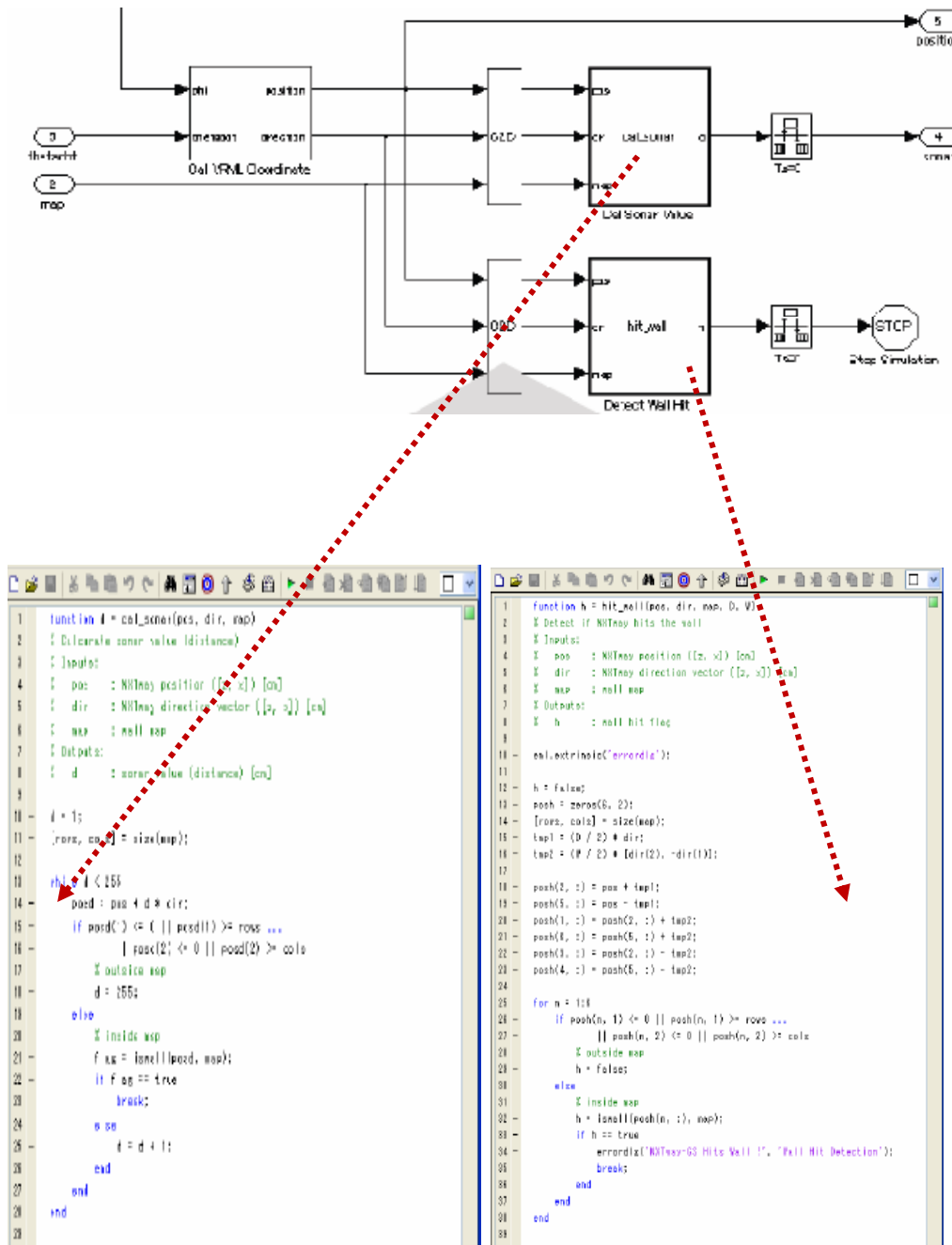


Track.wrl

Σχήμα B.4 : Δημιουργία ενός Αρχείου Χάρτη (map file)

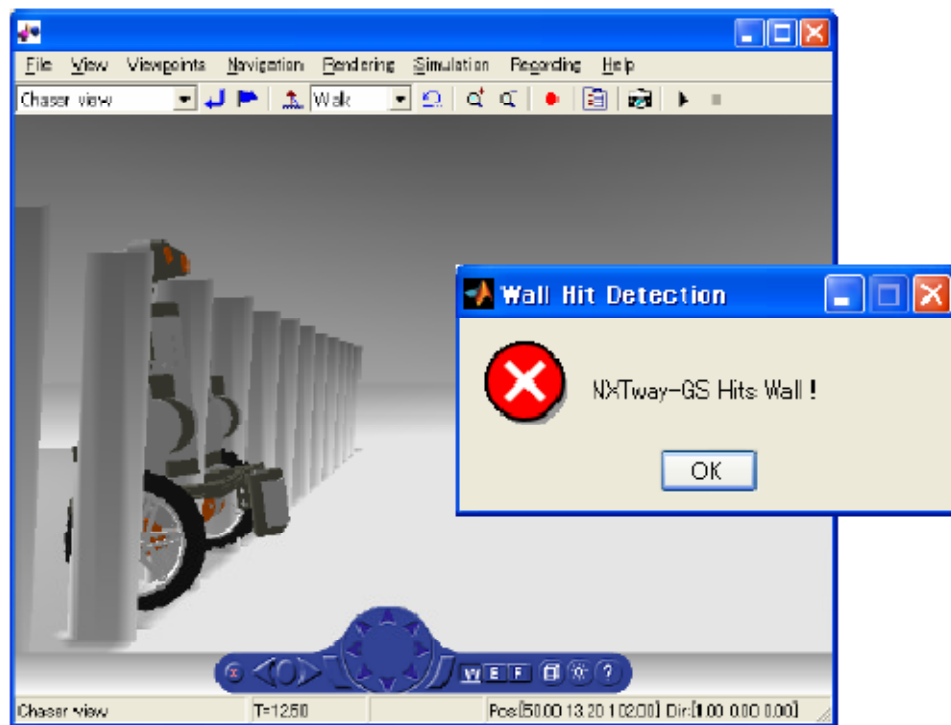
B.3 Υπολογισμός Απόστασης και Εντοπισμός Πρόσκρουσης σε τοίχο

Η απόσταση ανάμεσα στο NXTway-GS και τον τοίχο και ο εντοπισμός πρόσκρουσης σε τοίχο υπολογίζονται με το Embedded MATLAB function blocks στο υποσύστημα Αισθητήρα.



Σχήμα B.5 : Embedded MATLAB function blocks Υπολογισμού Απόστασης και Πρόσκρουσης σε Τοίχο

Ένα μήνυμα σφάλματος εμφανίζεται σε ένα παράθυρο όταν το NXTway-GS προσκρούσει σε τοίχο.



Σχήμα Β.6 : Πρόσκρουση σε Τοίχο

Παράρτημα Γ. Παραγόμενος Κώδικας

Σε αυτό το παράρτημα περιγράφεται η παραγωγή του κυρίως κώδικα από το `nxtway_gs_controller.mdl`. Αυτός είναι ο κώδικας προεπιλογής (default code) που παράγεται από τον RTW-EC. Ο RTW-EC επιτρέπει στον χρήστη να ορίσει τις μεταβλητές ιδιότητες, όπως το όνομα μεταβλητής, την κλάση αποθήκευσης, τροποποιήσεις κ.τ.λ., με την χρήση του Simulink Data Object, όμως δεν χρησιμοποιείται εδώ.

```
nxtway_app.c

#include "nxtway_app.h"
#include "nxtway_app_private.h"

BlockIO rtB;
D_Work rtDWork;
void task_init(void)
{
    rtDWork.battery = (real32_T)ecrobot_get_battery_voltage();
    rtDWork.flag_mode = 0U;
    rtDWork.gyro_offset = (real32_T)ecrobot_get_gyro_sensor(NXT_PORT_S4);
    rtDWork.start_time = ecrobot_get_sysclock_ms();
}

void task_tsl(void)
{
    real32_T rtb_UnitDelay;
    real32_T rtb_IntegralGain;
    real32_T rtb_Sum_g;
    real32_T rtb_UnitDelay_j;
    real32_T rtb_Sum_a;
    real32_T rtb_UnitDelay_l;
    real32_T rtb_DataTypeConversion1;
    real32_T rtb_theta;
    real32_T rtb_Sum2_b[4];
    real32_T rtb_Gain2_d;
    real32_T rtb_DataTypeConversion2;
    real32_T rtb_psidot;
    real32_T rtb_Sum_b;
    real32_T rtb_Saturation1;
    int16_T rtb_DataTypeConversion2_g1;
    int8_T rtb_DataTypeConversion;
    int8_T rtb_DataTypeConversion6;
    int8_T rtb_Switch;
    int8_T rtb_Switch2_e;
    int8_T rtb_DataTypeConversion_m[2];
    uint8_T rtb_UnitDelay_lv;
    boolean_T rtb_DataStoreRead;
    boolean_T rtb_DataStoreRead1;
    boolean_T rtb_Switch3_m;

    {
        int32_T i;
        rtb_DataStoreRead = rtDWork.flag_start;
        if (rtDWork.flag_start) {
            rtb_UnitDelay = rtDWork.UnitDelay_DSTATE;
            rtb_IntegralGain = -4.472125901E-001F * rtDWork.UnitDelay_DSTATE;
            rtb_UnitDelay_l = rtDWork.UnitDelay_DSTATE_j;
            rtb_DataStoreRead1 = rtDWork.flag_avoid;
            if (rtDWork.flag_mode) {
                if (rtDWork.flag_auto) {
                    if (rtDWork.flag_avoid) {
                        rtb_Switch = 0;
                    } else {
                        rtb_Switch = 100;
                    }
                }
            }
        }
    }
}
```

```

    }

    if (rtDWork.flag_avoid) {
        rth_Switch2_e = 100;
    } else {
        rth_Switch2_e = 0;
    }
} else {
    rth_Switch = 0;
    rth_Switch2_e = 0;
}
} else {
    ecrobot_read_bt_packet(rtB.BluetoothRxRead, 32);
    for (i = 0; i < 2; i++) {
        rth_DataTypeConversion_m[i] = (int8_T)rtB.BluetoothRxRead[i];
    }

    if (rth_DataStoreRead1) {
        rth_Switch = 100;
    } else {
        rth_Switch = rth_DataTypeConversion_m[0];
    }

    rth_Switch = (int8_T)(-rth_Switch);
    rth_Switch2_e = rth_DataTypeConversion_m[1];
}

rth_Sum_g = (real32_T)rth_Switch / 100.0F * 7.8F * 4.000000190E-002F +
9.959999919E-001F * rtDWork.UnitDelay_DSTATE_e;
rth_DataTypeConversion3 = (real32_T)ecrobot_get_motor_rev(NXT_PORT_C);
rth_UnitDelay_j = rtDWork.UnitDelay_DSTATE_h;
rth_DataTypeConversion4 = (real32_T)ecrobot_get_motor_rev(NXT_PORT_B);
rth_theta = ((1.745329238E-002F * rth_DataTypeConversion3 +
rtDWork.UnitDelay_DSTATE_h) + (1.745329238E-002F *
rth_DataTypeConversion4 + rtDWork.UnitDelay_DSTATE_h)) / 2.0F;
rth_Sum_a = 2.3000000030E-001F * rth_theta + 8.000000119E-001F *
rtDWork.UnitDelay_DSTATE_hp;
rth_Sum_h = (real32_T)ecrobot_get_gyro_sensor(NXT_PORT_R4);
rtDWork.gyro_offset = 9.990000129E-001F * rtDWork.gyro_offset +
1.000000047E-003F * rth_Sum_h;
rth_psidot = (rth_Sum_b - rtDWork.gyro_offset) * 1.715329238E-002F;
rth_Sum2_b[0] = rth_UnitDelay_l - rth_theta;
rth_Sum2_b[1] = 0.0F - rtDWork.UnitDelay_DSTATE_h;
rth_Sum2_b[2] = rth_Sum_g - (rth_Sum_a - rtDWork.UnitDelay_DSTATE_m) /
4.000000190E-002F;
rth_Sum2_b[3] = 0.0F - rth_psidot;

{
    static const int_T dims[3] = { 1, 4, 1 };

    rt_MatMulRR_Sgl((real32_T *)rth_Sum_h, (real32_T *)
&rtConstP.FeedbackGain_Gain[0],
(real32_T *)rth_Sum2_b, &dims[0]);
}

rth_IntegralGain += rth_Sum_b;
rth_Sum_b = rtDWork.battery;
rth_Sum_b = rth_IntegralGain / (1.089000027E-003F * rth_Sum_b - 0.625F) *
100.0F;
rth_Sum_b += 0.0F * rt_F32N(rth_Sum_b);
rth_IntegralGain = (real32_T)rth_Switch2_e;
rth_Gain2_d = rth_IntegralGain / 100.0F * 25.0F;
rth_Saturation1 = rth_Sum_b + rth_Gain2_d;
rth_Saturation1 = rt_SATURATE(rth_Saturation1, -100.0F, 100.0F);
rth_DataTypeConversion = (int8_T)rth_Saturation1;
ecrobot_set_motor_speed(NXT_PORT_C, rth_DataTypeConversion);
if ((int8_T)rth_IntegralGain != 0) {
    rth_DataStoreRead1 = 1U;
} else {
    rth_DataStoreRead1 = 0U;
}

```

```

    }

    rtb_IntegralGain = rtb_DataTypeConversion3 - rtb_DataTypeConversion4;
    if ((!rtb_DataStoreRead1) && rtDWork.UnitDelay_DSTATE_c) {
        rtB.theta_diff = rtb_IntegralGain;
    }

    if (rtb_DataStoreRead1) {
        rtb_Saturation1 = 0.0F;
    } else {
        rtb_Saturation1 = (rtb_IntegralGain - rtB.theta_diff) *
            3.499999940E-001F;
    }

    rtb_Saturation1 += rtb_Sum_b - rtb_Gain2_d;
    rtb_Saturation1 = rt_SATURATE(rtb_Saturation1, -100.0F, 100.0F);
    rtb_DataTypeConversion6 = (int8_T)rtb_Saturation1;
    ecrobot_set_motor_speed(NXT_PORT_B, rtb_DataTypeConversion6);
    rtb_UnitDelay_lv = rtDWork.UnitDelay_DSTATE_k;
    if (rtDWork.UnitDelay_DSTATE_k != 0U) {
        rtb_Switch3_m = 0U;
    } else {
        rtb_Switch3_m = 1U;
    }

    if (rtb_Switch3_m) {
        rtb_DataTypeConversion2_g1 = (int16_T)floor((real_T)(5.729578018E+001F *
            rtb_UnitDelay_j / 0.0009768628F) - 0.5);
        ecrobot_bt_adc_data_logger(rtb_DataTypeConversion,
            rtb_DataTypeConversion6, rtb_DataTypeConversion2_g1, 0, 0, 0);
    }

    rtb_UnitDelay_lv = (uint8_T)(1U + (uint32_T)rtb_UnitDelay_lv);
    if (25U == rtb_UnitDelay_lv) {
        rtDWork.UnitDelay_DSTATE_k = 0U;
    } else {
        rtDWork.UnitDelay_DSTATE_k = rtb_UnitDelay_lv;
    }

    rtDWork.UnitDelay_DSTATE = (rtb_UnitDelay_1 - rtb_theta) *
        4.000000190E-003F + rtb_UnitDelay;
    rtDWork.UnitDelay_DSTATE_j = 4.000000190E-003F * rtb_Sum_g +
        rtb_UnitDelay_1;
    rtDWork.UnitDelay_DSTATE_e = rtb_Sum_g;
    rtDWork.UnitDelay_DSTATE_h = 4.000000190E-003F * rtb_psidot +
        rtb_UnitDelay_j;
    rtDWork.UnitDelay_DSTATE_hp = rtb_Sum_a;
    rtDWork.UnitDelay_DSTATE_m = rtb_Sum_a;
    rtDWork.UnitDelay_DSTATE_c = rtb_DataStoreRead1;
}

if (!rtb_DataStoreRead) {
    rtDWork.gyro_offset = 8.0000000119E-001F * rtDWork.gyro_offset +
        2.000000030E-001F * (real32_T)ecrobot_get_gyro_sensor(NXT_PORT_S4);
}
}

void task_ts2(void)
{
    boolean_T rtb_DataStoreRead2_j;
    if (rtDWork.flag_mode) {
        rtb_DataStoreRead2_j = rtDWork.flag_avoid;
        if (!rtDWork.flag_avoid) {
            rtDWork.flag_avoid = (ecrobot_get_sonar_sensor(NXT_PORT_S2) <= 20);
            rtB.RevolutionSensor_C = ecrobot_get_motor_rev(NXT_PORT_C);
        }

        if (rtb_DataStoreRead2_j) {
            rtDWork.flag_avoid = (ecrobot_get_motor_rev(NXT_PORT_C) -

```

```

            rtB.RevolutionSensor_C <= 210);
        }
    }

    if (!rtDWork.flag_mode) {
        rtDWork.flag_avoid = (ecrobot_get_sonar_sensor(NXT_PORT_S2) <= 20);
    }
}

void task_ts3(void)
{
    uint32_T rtb_Sum1_i;
    boolean_T rtb_RelationalOperator_e0;
    rtDWork.battery = 8.0000000119E-001F * rtDWork.battery + 2.000000030E-001F *
        (real32_T)ecrobot_get_battery_voltage();
    rtb_Sum1_i = (uint32_T)((int32_T)ecrobot_get_sysclock_ms() - (int32_T)
        rtDWork.start_time);
    rtb_RelationalOperator_e0 = (rtb_Sum1_i >= 1000U);
    rtDWork.flag_start = rtb_RelationalOperator_e0;
    rtDWork.flag_auto = (rtb_Sum1_i >= 8000U);
    if (rtb_RelationalOperator_e0 != rtDWork.UnitDelay_DSTATE_a) {
        sound_freq(440U, 800U);
    }

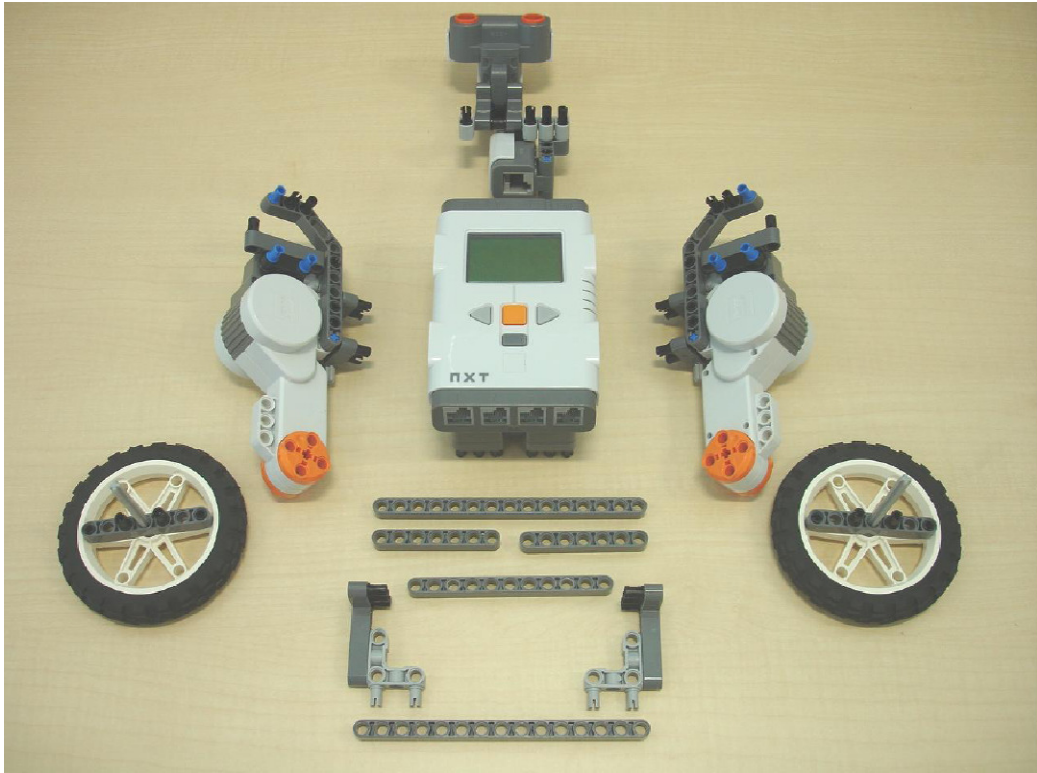
    rtDWork.UnitDelay_DSTATE_a = rtb_RelationalOperator_e0;
}

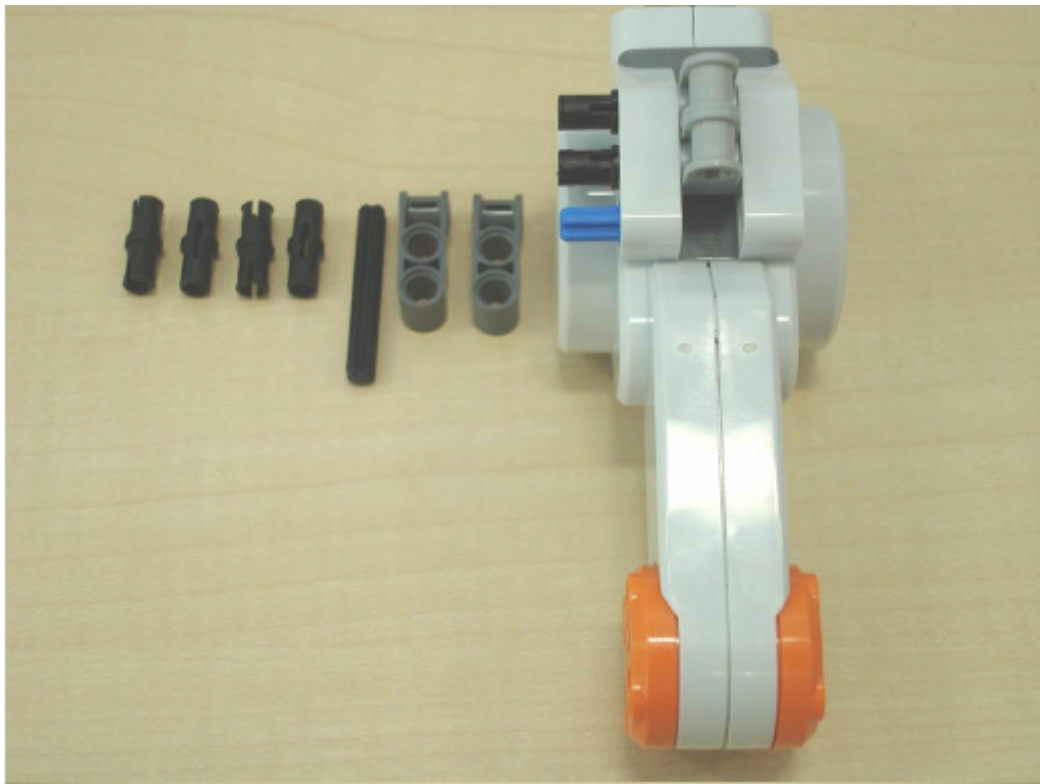
void matway_app_initialize(void)
{
}

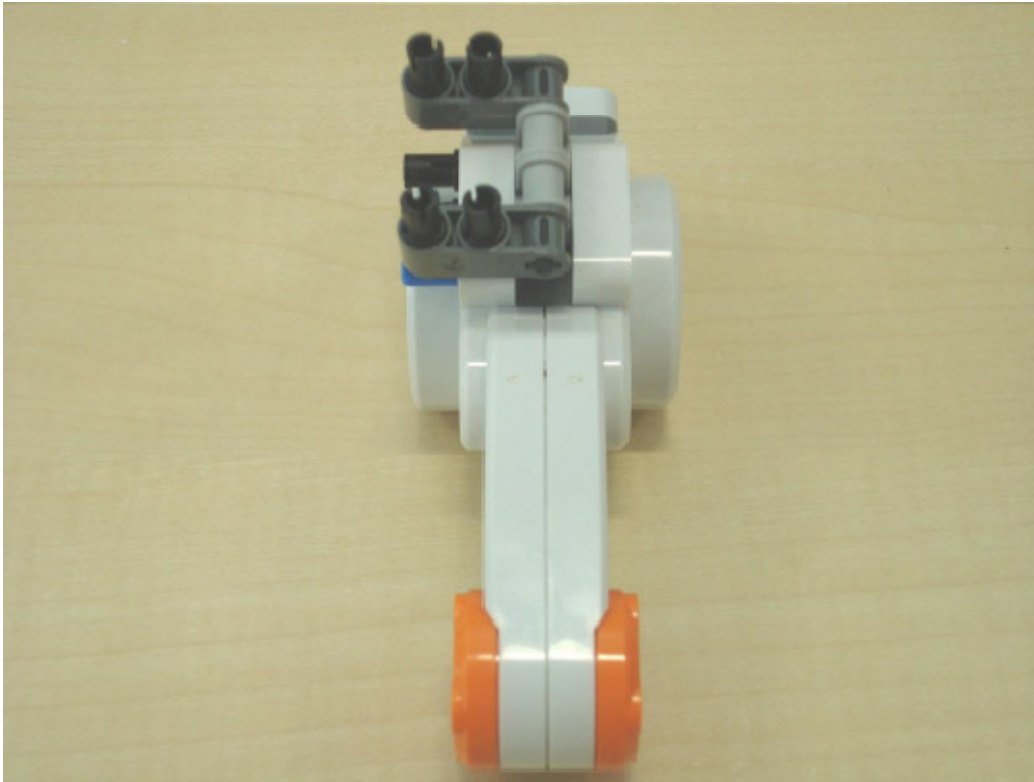
```

Παράρτημα Δ. Οδηγίες Κατασκευής του NXTway-GS

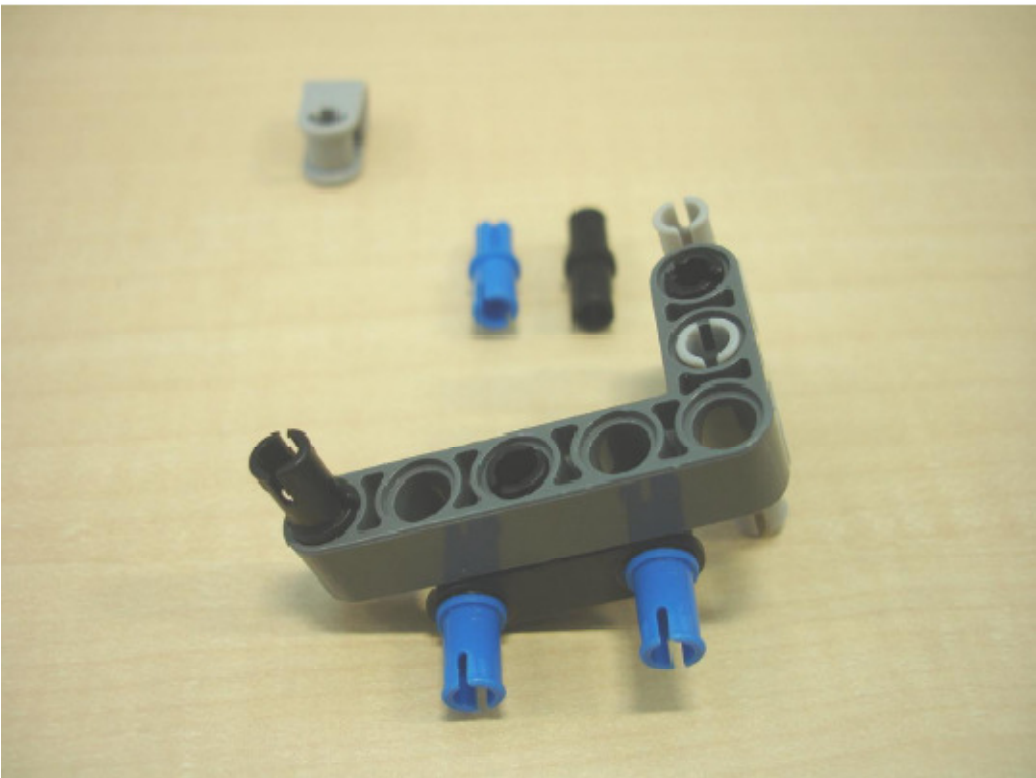




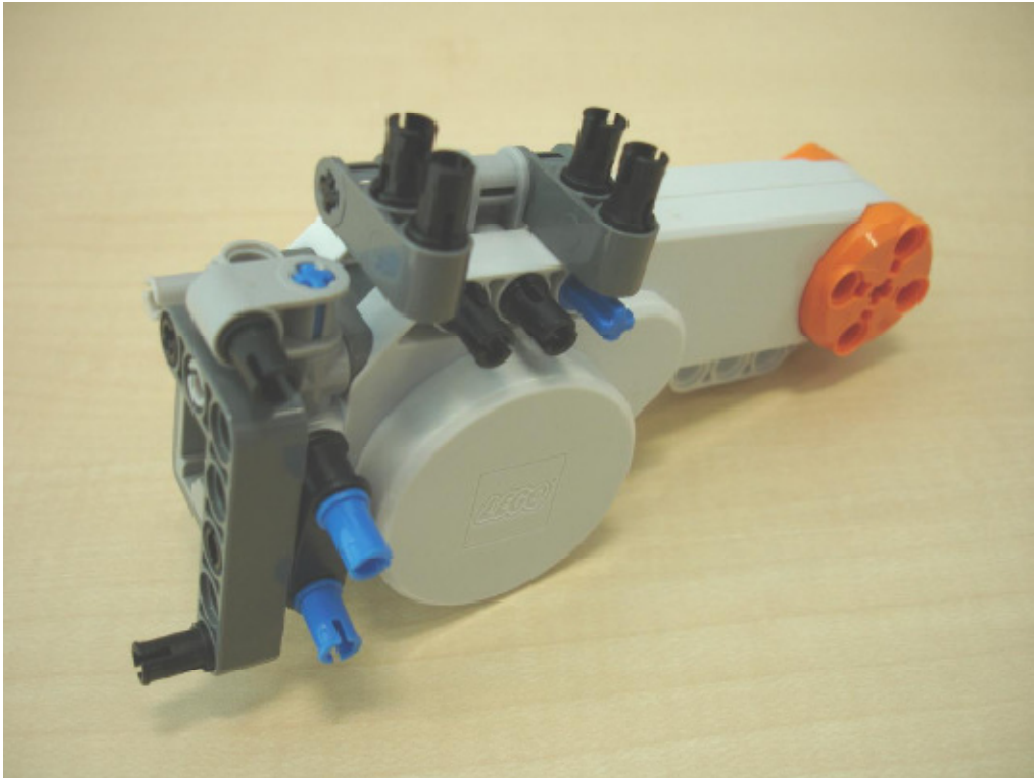


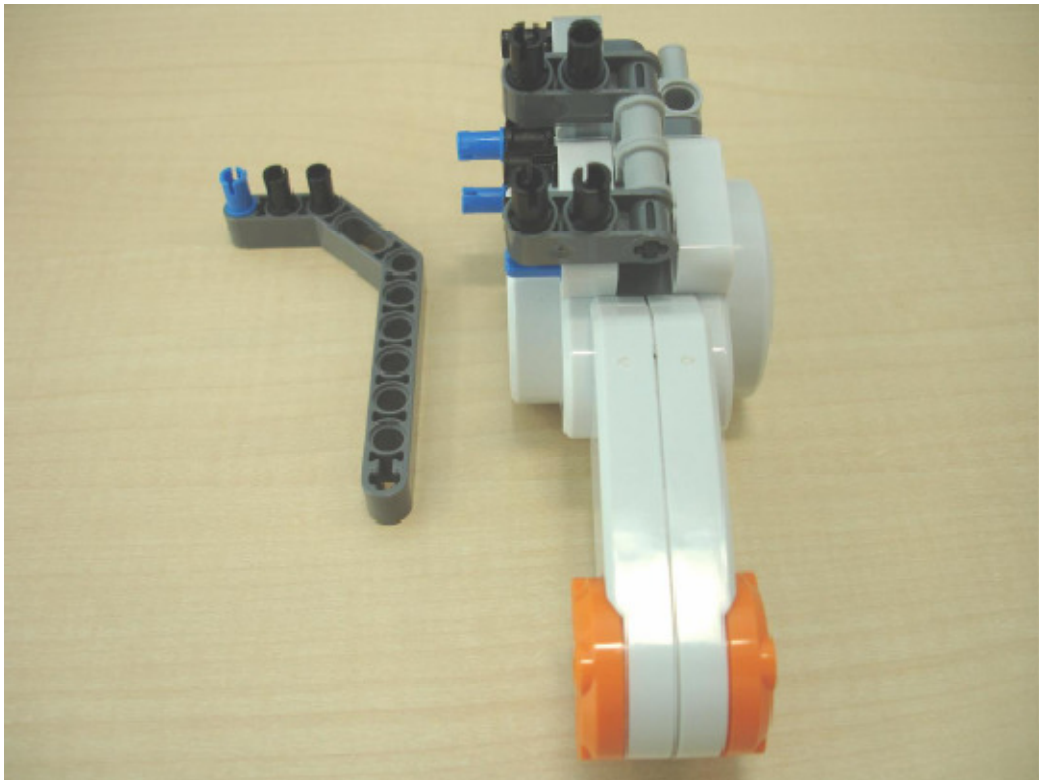


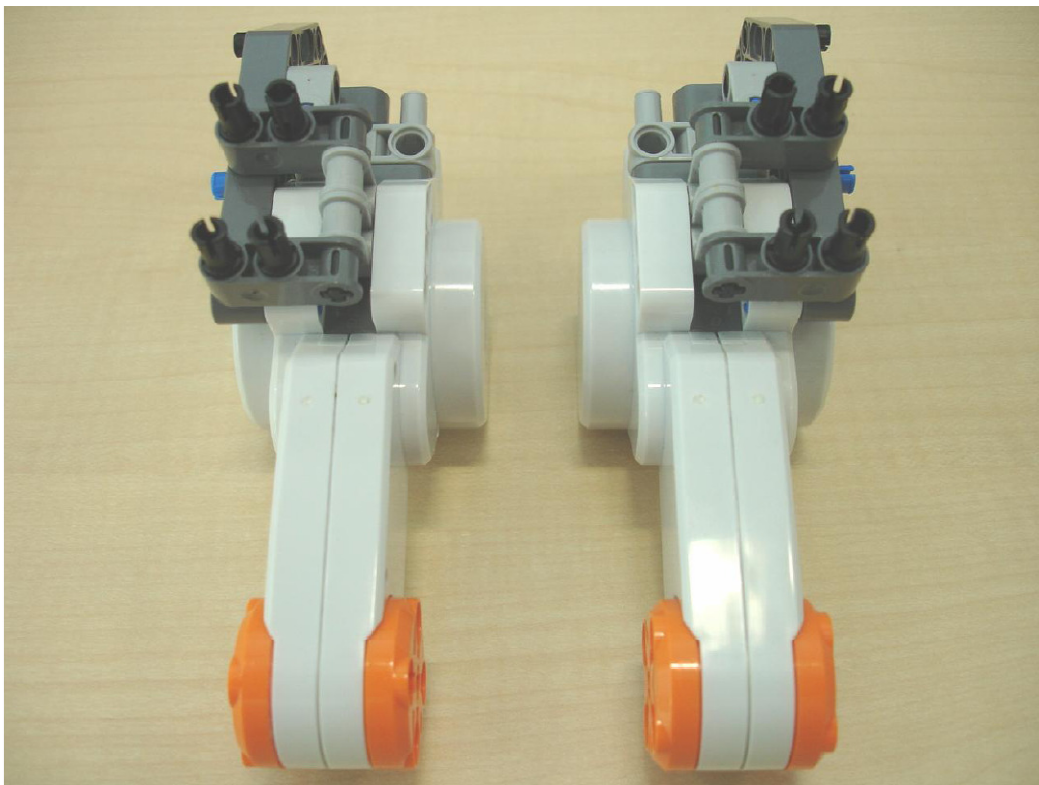
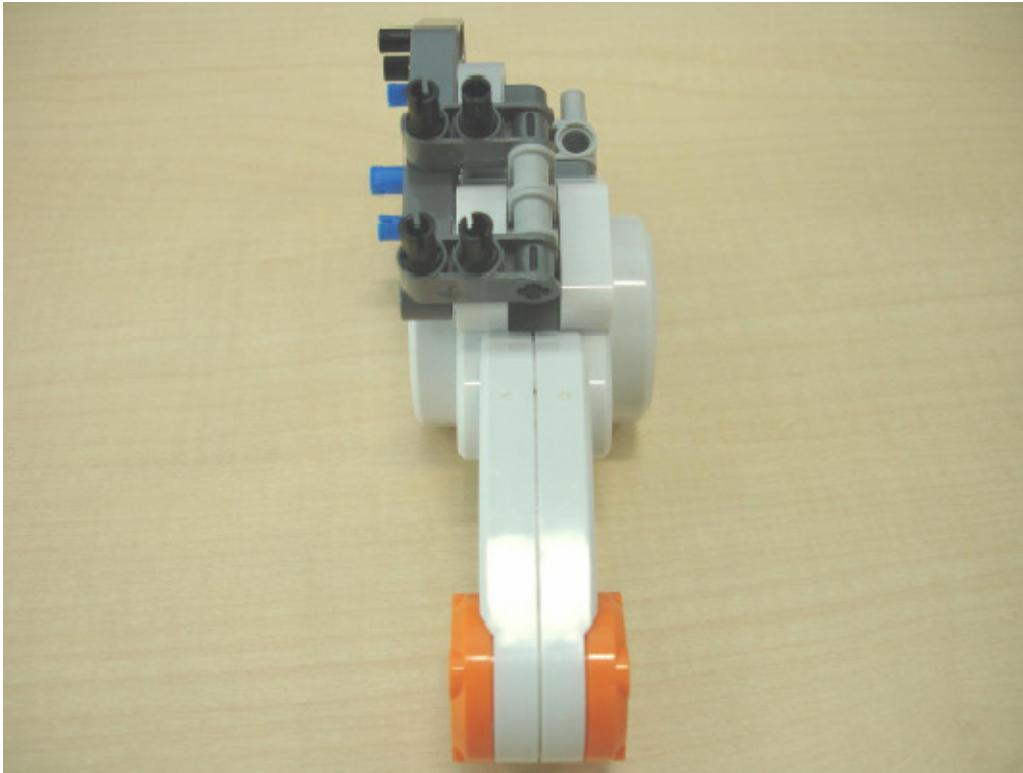


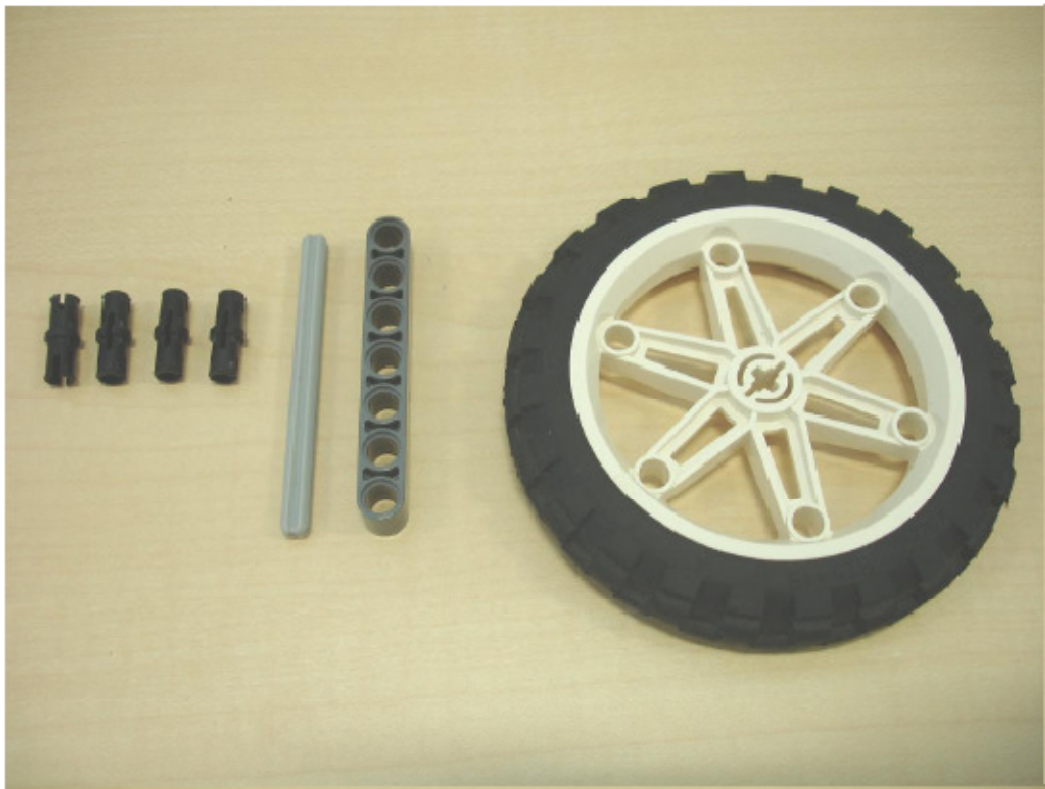




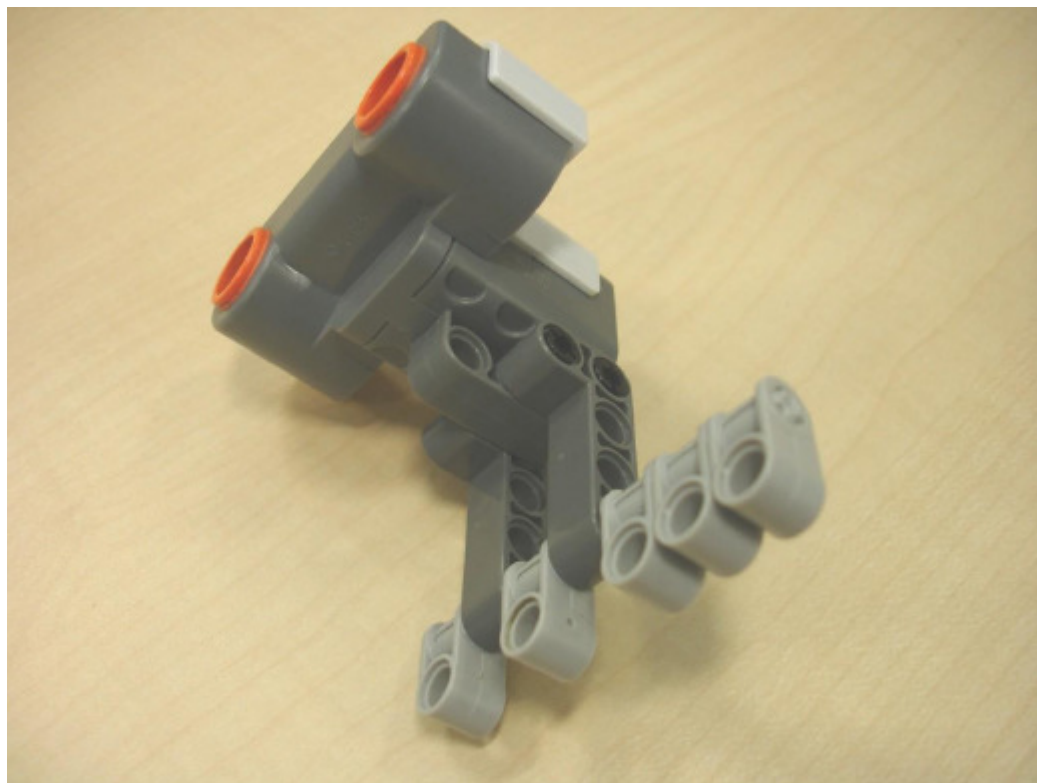




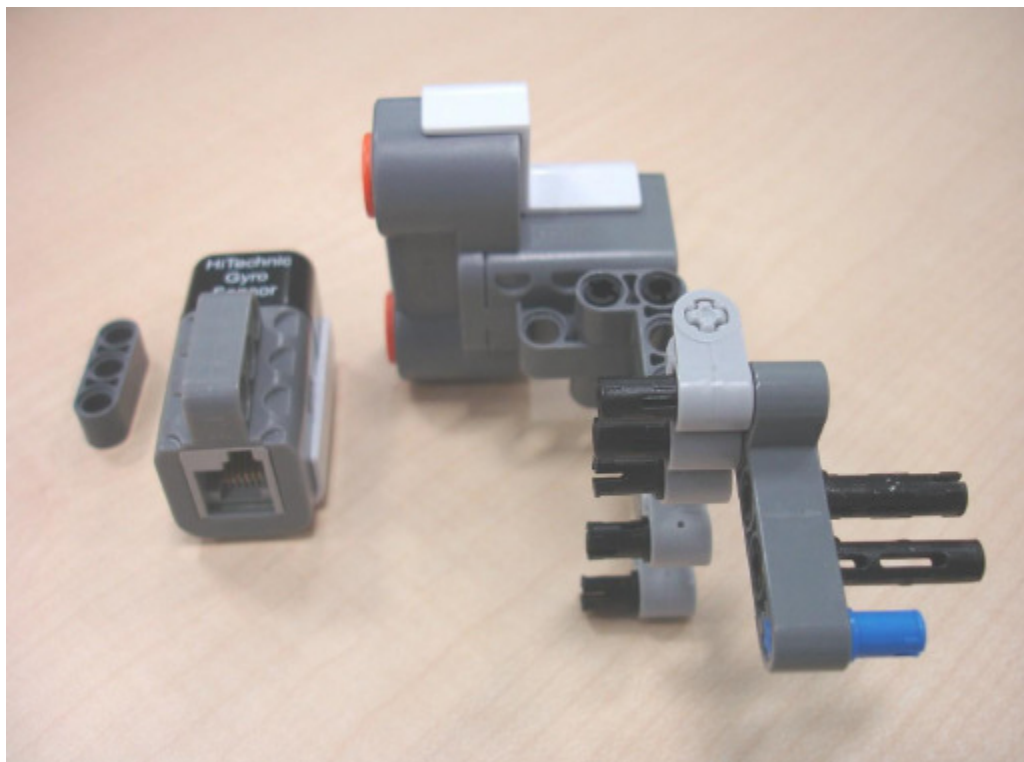




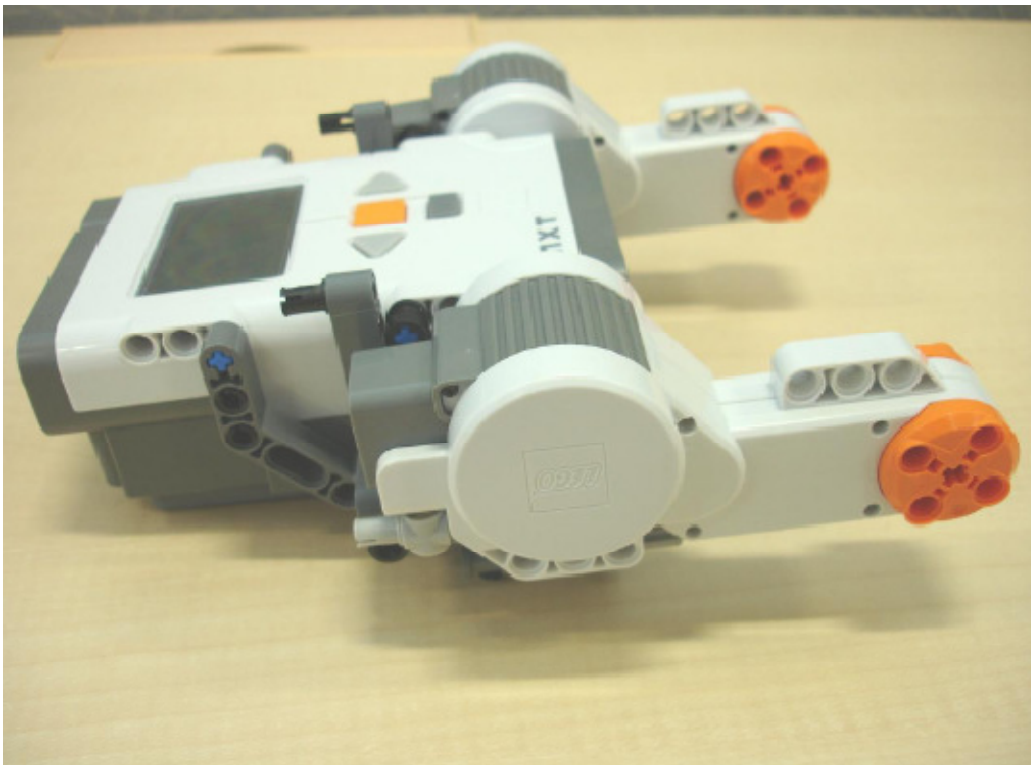
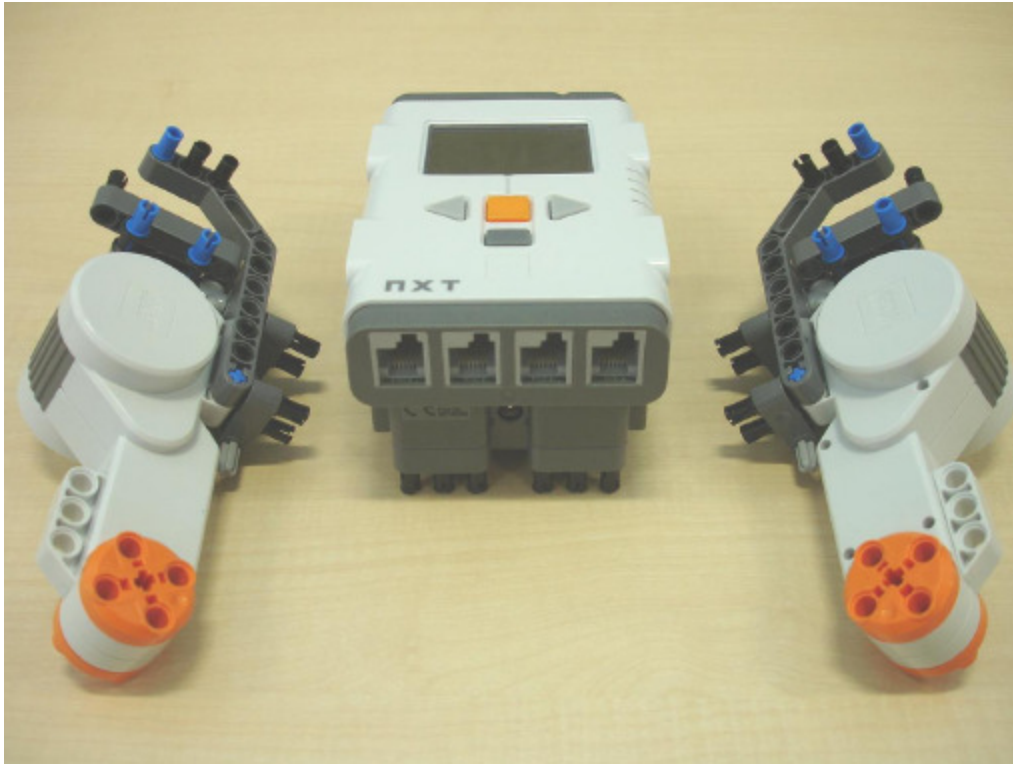


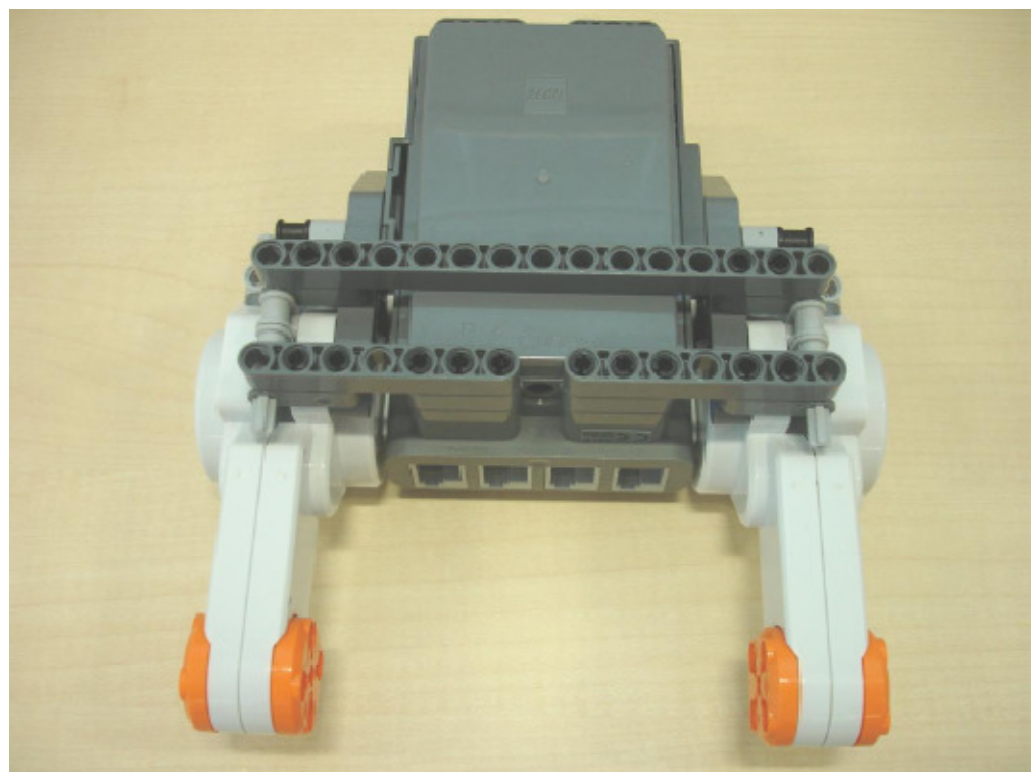
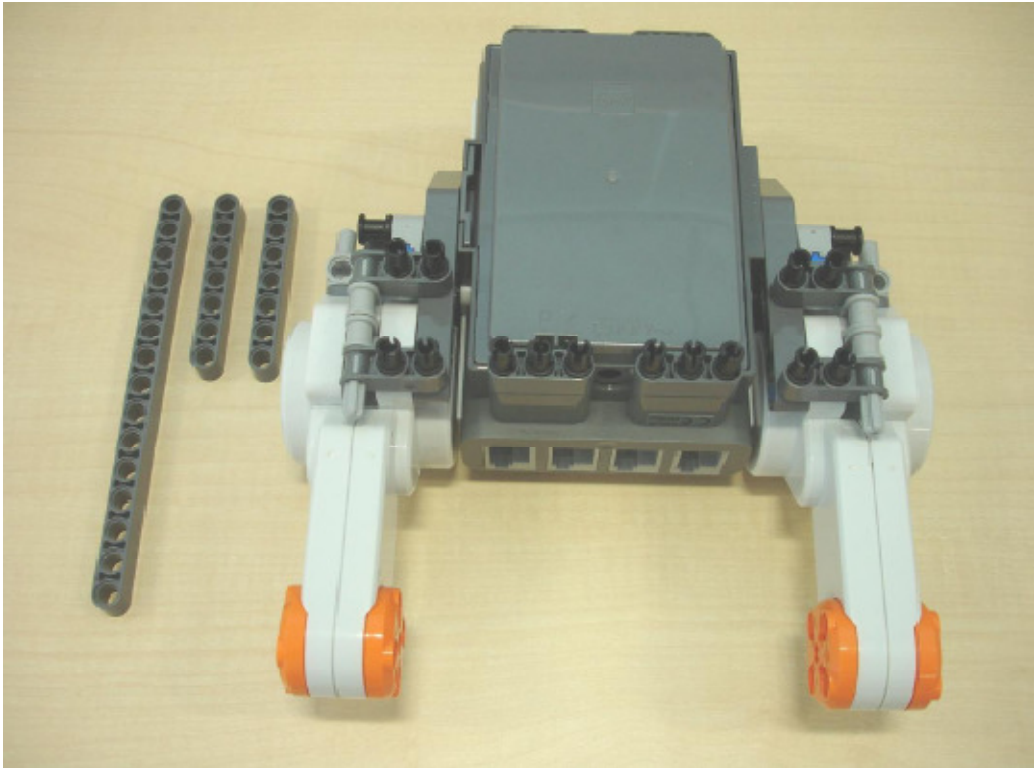


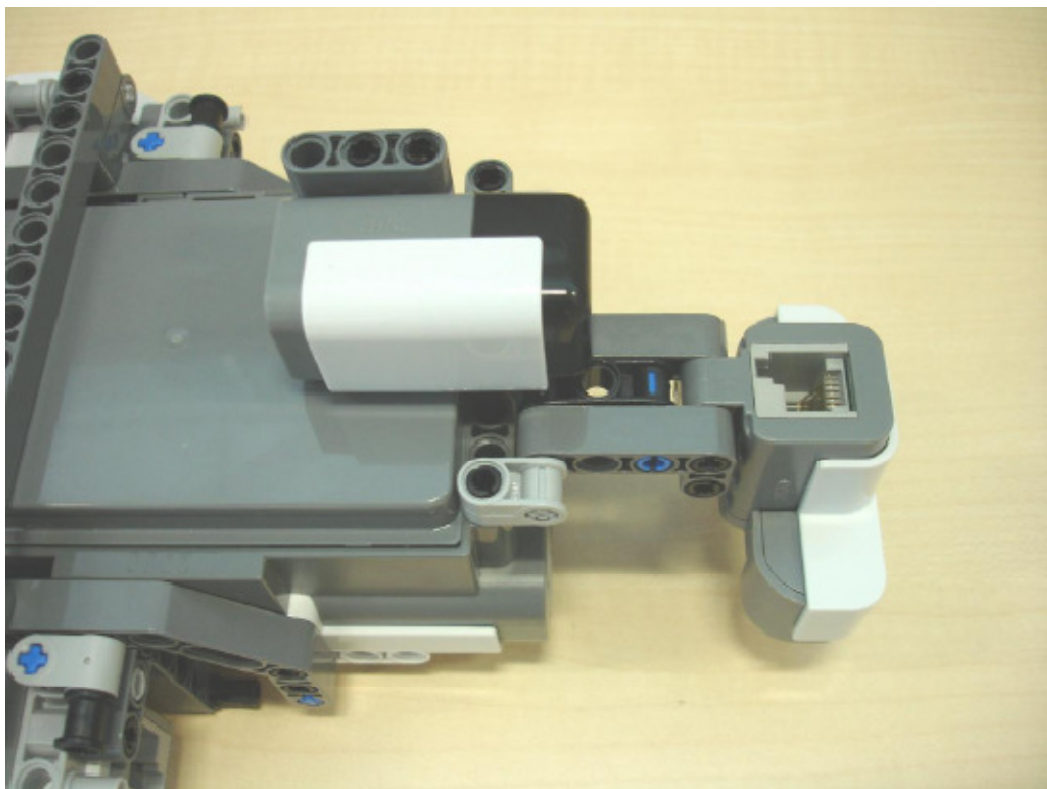
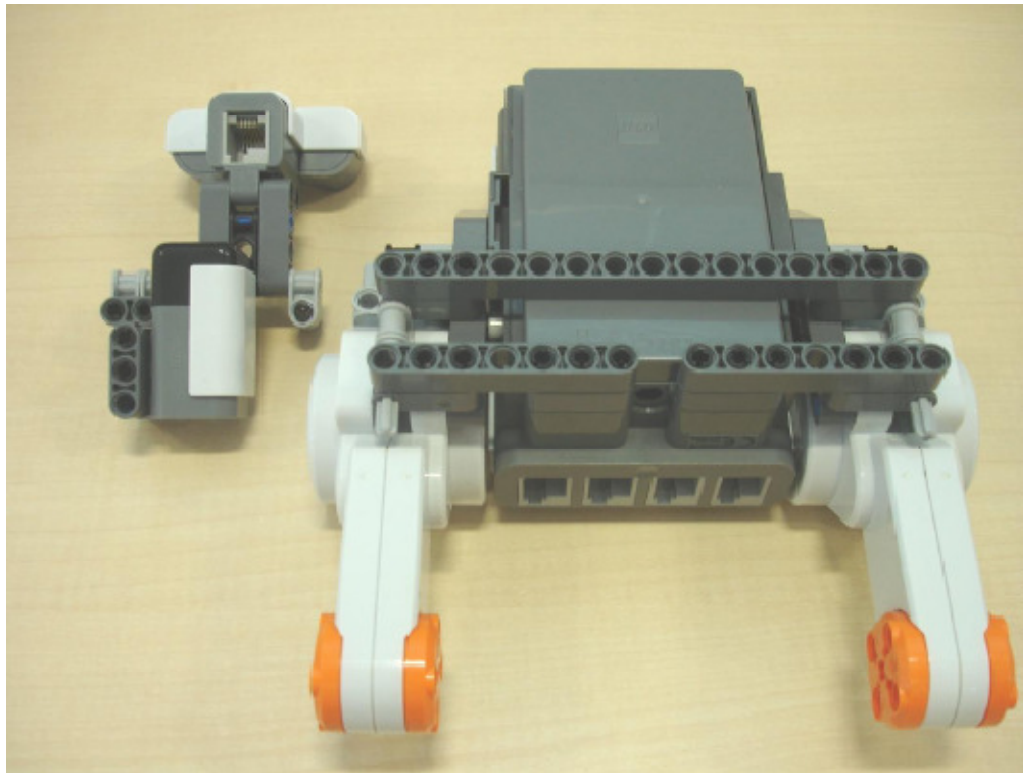


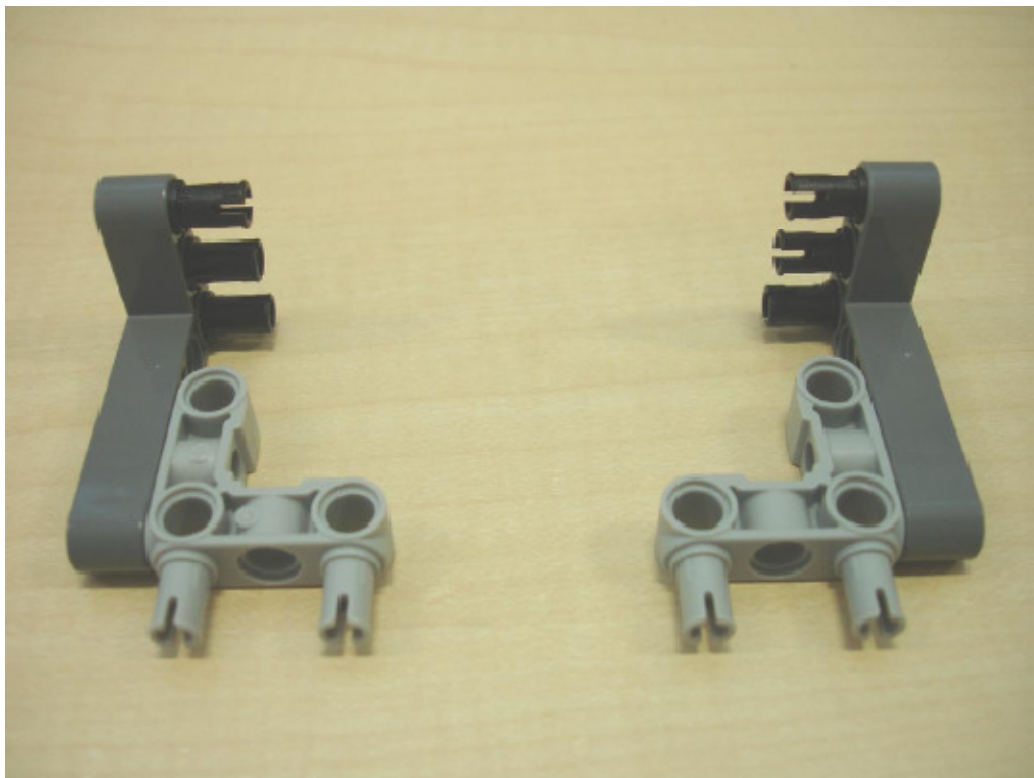


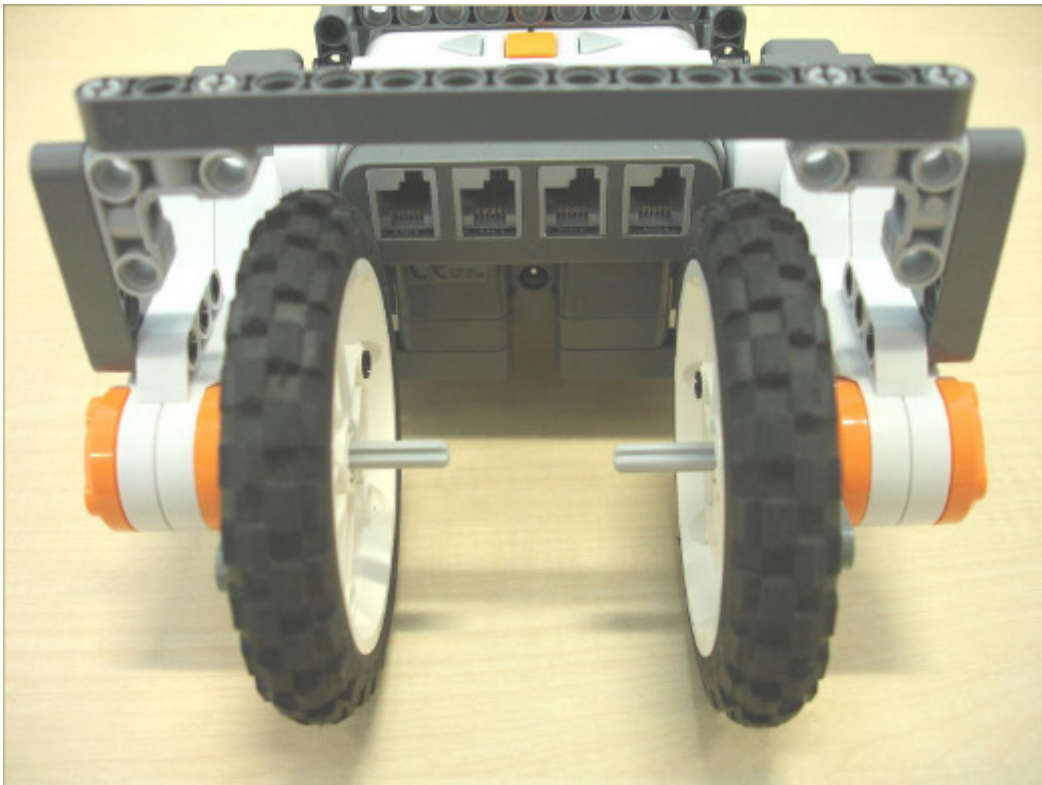
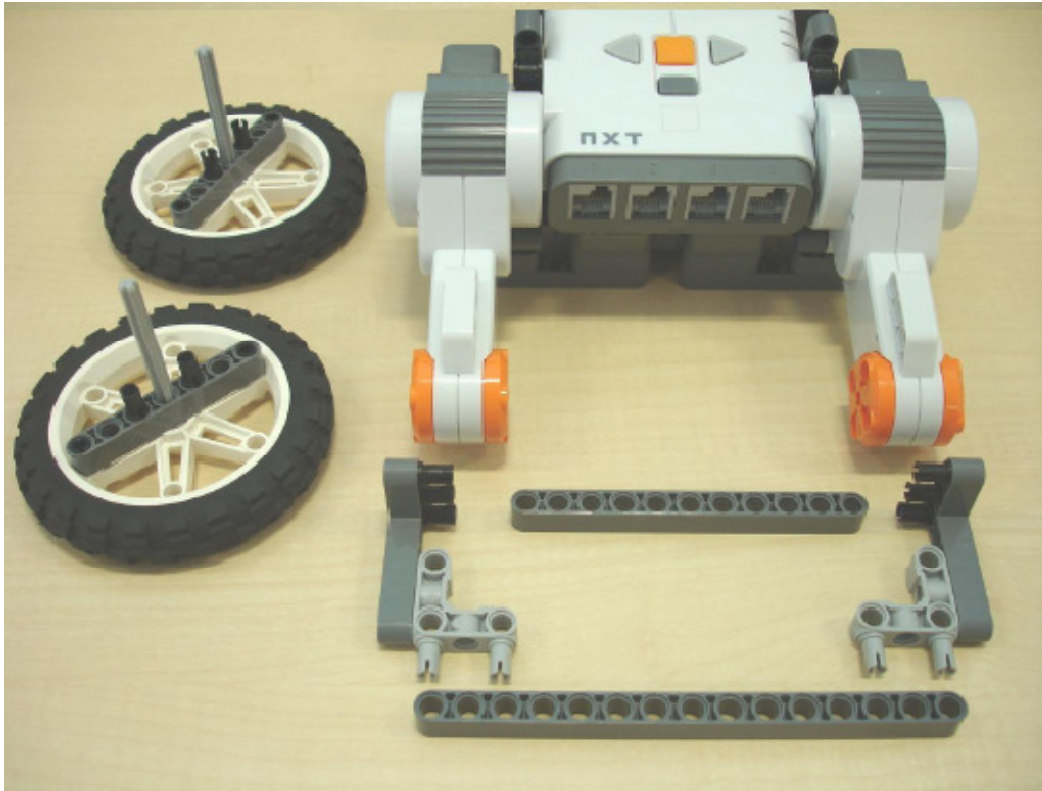




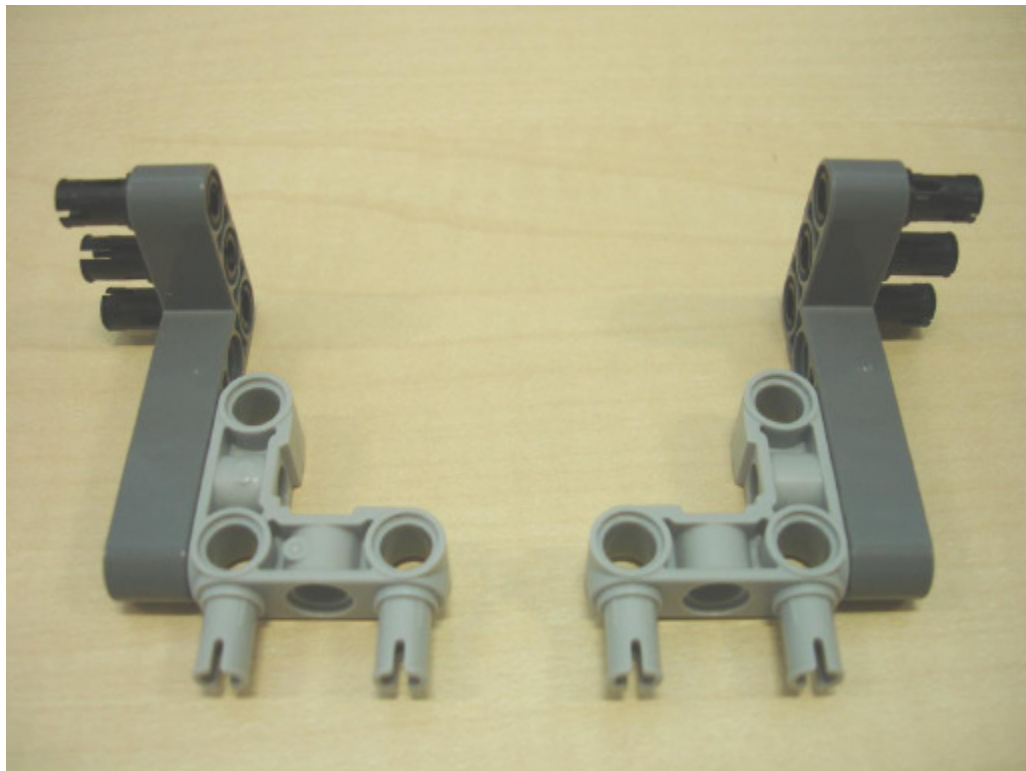


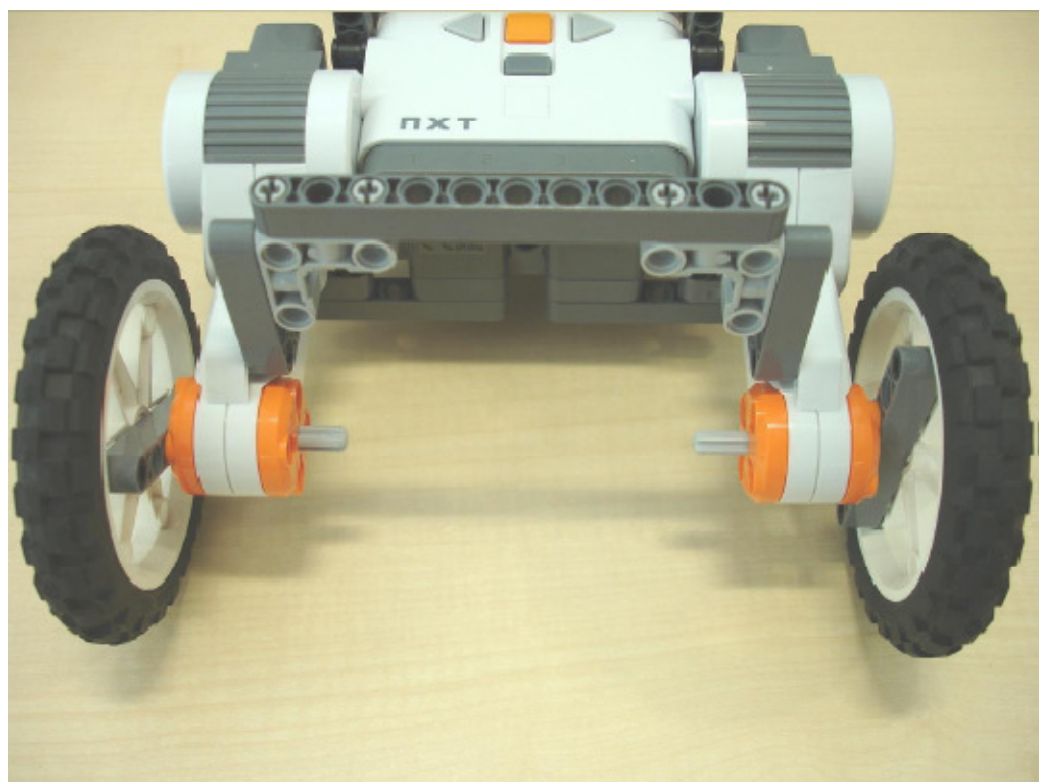
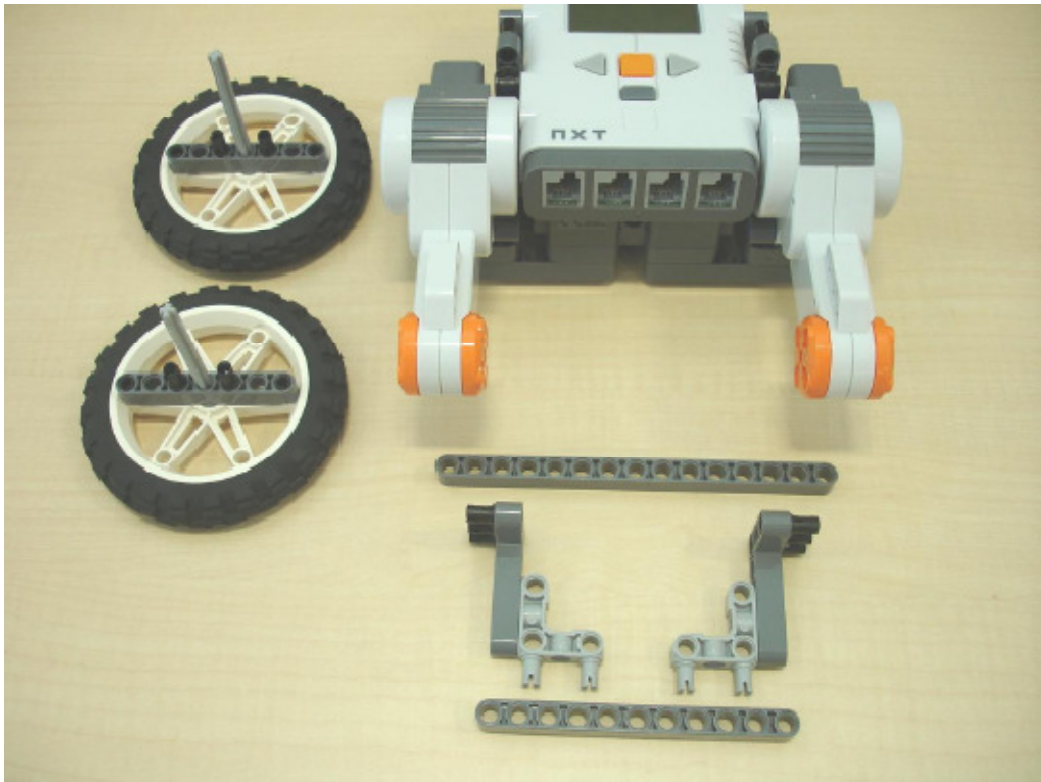




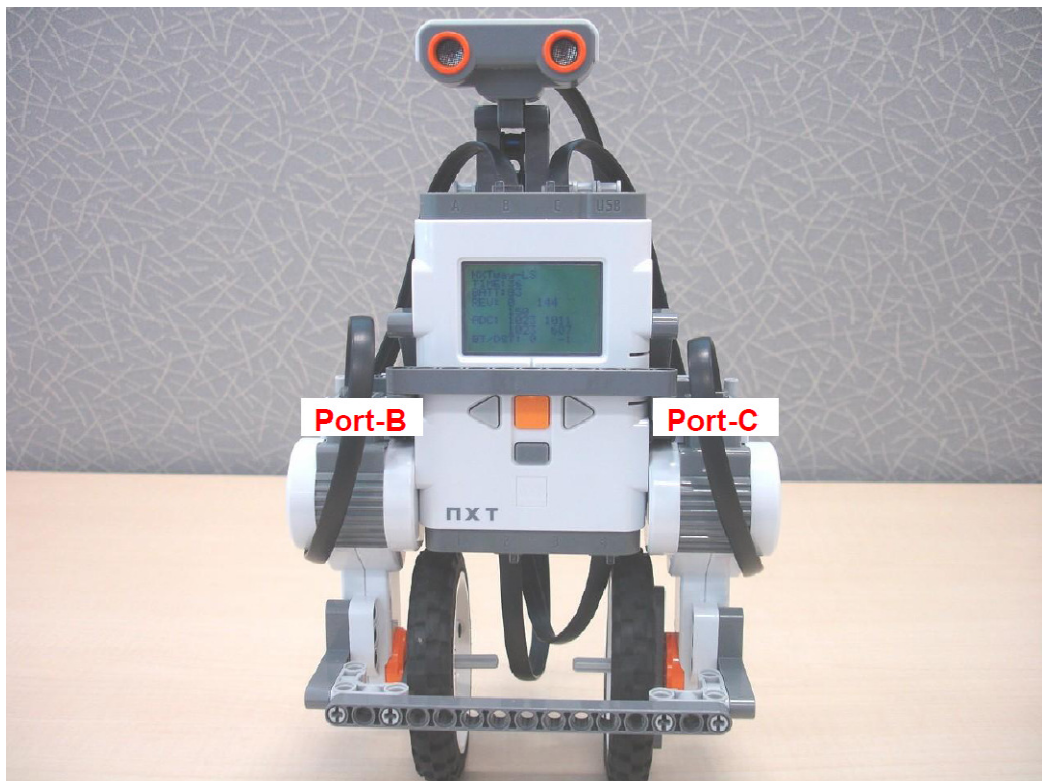














STR750Fxx STR751Fxx STR752Fxx STR755Fxx

ARM7TDMI-S™ 32-bit MCU with Flash, SMI, 3 std 16-bit timers, PWM timer, fast 10-bit ADC, I2C, UART, SSP, USB and CAN

Features

- Core
 - ARM7TDMI-S 32-bit RISC CPU
 - 54 DMIPS @ 60 MHz
- Memories
 - Up to 256 KB Flash program memory (10k W/E cycles, retention 20 yrs @ 85°C)
 - 16 KB Read-While-Write Flash for data (100k W/E cycles, retention 20 yrs@ 85°C)
 - Flash Data Readout and Write Protection
 - 16KBytes embedded high speed SRAM
 - Memory mapped interface (SMI) to ext. Serial Flash (64 MB) w. boot capability
- Clock, reset and supply management
 - Single supply 3.3V ±10% or 5V ±10%
 - Embedded 1.8V Voltage Regulators
 - Int. RC for fast start-up and backup clock
 - Up to 60 MHz operation using internal PLL with 4 or 8 MHz crystal/ceramic osc.
 - Smart Low Power Modes: SLOW, WFI, STOP and STANDBY with backup registers
 - Real-time Clock, driven by low power internal RC or 32.768 kHz dedicated osc, for clock-calendar and Auto Wake-up
- Nested interrupt controller
 - Fast interrupt handling with 32 vectors
 - 16 IRQ priorities, 2 maskable FIQ sources
 - 16 external interrupt / wake-up lines
- DMA
 - 4-channel DMA controller
 - Circular buffer management
 - Support for UART, SSP, Timers, ADC
- 6 Timers
 - 16-bit watchdog timer (WDG)
 - 16-bit timer for system timebase functions
 - 3 synchronizable timers each with up to 2 input captures and 2 output compare/PWMs.
- 16-bit 6-ch. synchronizable PWM timer
- Dead time generation, edge/center-aligned waveforms and emergency stop
- Ideal for induction/brushless DC motors
- 8 Communications interfaces
 - 1 I²C interface
 - 3 HiSpeed UARTs w. Modem/LIN capability
 - 2 SSP interfaces (SPI or SSI) up to 16 Mb/s
 - 1 CAN interface (2.0B Active)
 - 1 USB full-speed 12 Mb/s interface with 8 configurable endpoint sizes
- 10-bit A/D converter
 - 16/11 chan. with prog. Scan Mode & FIFO
 - Programmable Analog Watchdog feature
 - Conversion time: min. 3.75 µs
 - Start conversion can be triggered by timers
- Up to 72/38 I/O ports
 - 72/38 GPIOs with High Sink capabilities
 - Atomic bit SET and RES operations

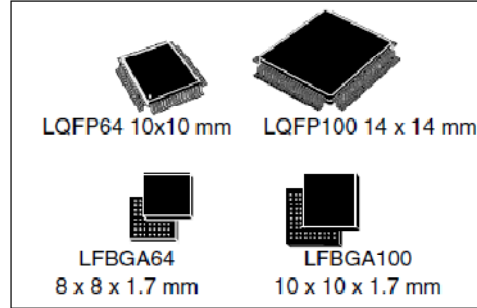


Table 1. Device summary

Reference	Part number
STR750Fxx	STR750FV0, STR750FV1, STR750FV2
STR751Fxx	STR751FR0, STR751FR1, STR751FR2
STR752Fxx	STR752FR0, STR752FR1, STR752FR2
STR755Fxx	STR755FR0, STR755FR1, STR755FR2 STR755FV0, STR755FV1, STR755FV2

Βιβλιογραφία

- [1] Αρναουτάκης Νεκτάριος (2005), “Εισαγωγή στο MATLAB Simulink”.
- [2] Π.Ν. Παρασκευόπουλου, “Εισαγωγή στο Αυτόματο Έλεγχο” Τόμος Β’ Εφαρμογές, σελ 128 με 131
- [3] Takashi Chikamasa “Embedded Coder Robot NXT Instruction manual”.
- [4] Takashi Chikamasa “Embedded Coder Robot for LEGO® Mindstorms® NXT Rev.2.02” .
- [5] Takashi Chikamasa “Embedded Coder Robot NXT Modeling Tips”.
- [6] Yamamoto Y, (2008) “NXTway-GS Model-Based Design - Control of self balancing two-wheeled robot built with LEGO Mindstorms NXT”.
- [7] Jeong S, Takahshi T (2007) “Wheeled inverted pendulum type assistant robot: Design concept and mobile control”.
- [8] Grasser F, D'arigo A, Colombi S, Rufer A (2001) JOE, “A mobile, inverted pendulum”
- [9] Mike Rinderknecht, “Tutorial for Programming the LEGO® MINDSTORMS™ NXT”
- [10]Yorihisa Yamamoto. NXTway-GS Model-Based Design-Control of self-balancing two-wheeled robot built with LEGO Mindstorms NXT. CYBERNET SYSTEMS CO, LTD, 2008.
- [11]Yorihisa Yamamoto. LEGO MINDSTORMS NXTway-GS Building Instructions. CYBERNET SYSTEMS CO, LTD., 2008.
- [12]The Mathworks, Real Time Workshop For Use with Simulink, Users Guide Version 3
- [13]The Mathworks, Real Time Workshop Embedded Coder For Use with Real Time Workshop, Users Guide Version 3
- [14]The Mathworks, Simulink Using, Version 6
- [15]The Mathworks, Stateflow and Stateflow Coder, Users Guide Version 5
- [16]The Mathworks, Writing S-Functions, Version 6
- [17]LEGO Mindstorms. The NXT
http://mindstorms.lego.com/Overview/The_NXT.aspx
- [18]Philippe E. Hurbain. Ultrasonic Sensor.
<http://www.philohome.com/nxtway/nxtway.htm>
- [19]Farid Golnaraghi Benjamin C. Kuo, “Automatic Control Systems”. John Wiley and Sons, Inc, 2003.
- [20]Johnny Rieper, Bent Bisballe Nyeng, and Kasper Sohn. Marvin , “The Balancing Robot”. Aarhus University, 2009.
- [21]Richard C. Dorf and Robert H. Bishop. Modern Control Systems 10th Edition.Pearson Education, Inc., 2005.
- [22] HiTechnic. NXT EOPD <http://www.hitechnic.com/contents/en-us/d28.html>
- [23]J. Sluka , “A PID Controller For Lego Mindstorms Robots
http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html
- [24]“what is nxtOsek” <http://lejos-osek.sourceforge.net/whatislejososek.htm>
- [25]“MATLAB Bluetooth router” http://robotics.benedettelli.com/BT_router.htm

- [26]“NXT GamePad” <http://lejos-osek.sourceforge.net/nxtgamepad.htm>
- [27]“Self-Balancing Wheeled Robot”
<http://blogs.mathworks.com/pick/2009/08/14/self-balancing-wheeled-robot/>
- [28]“What is Embedded Coder Robot NXT”
<http://blogs.mathworks.com/pick/2009/08/14/self-balancing-wheeled-robot/>
- [29]“What is BrickOS” <http://brickos.sourceforge.net/>
- [30]“Elements of Electrical Engineering”, lecture notes, Lecture 3
<http://elec101.rice.edu/trac/wiki/lectureNotes/Lecture3>
- [31] “Inverted pendulum” http://en.wikipedia.org/wiki/Inverted_pendulum
- [32]“LEGO Mindstorms” <http://mindstorms.lego.com/en-us/Default.aspx>
- [33]“Lego Mindstorms NXT”
http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT
- [34]“ARM 7”, <http://en.wikipedia.org/wiki/ARM7TDMI>
- [35]“LEGO Mindstorms. Servo Motors.” http://mindstorms.lego.com/Overview/Interactive_Servo_Motors.aspx
- [36]HiTechnic. Sensor Multiplexer. <http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NSX2020>
- [37]HiTechnic. IR Beacon.
<http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=FTCBCN>
- [38]HiTechnic. NXT Programming Software <http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NSK1042>
- [39]Ryo’s Holiday LEGO Mindstorms NXT
http://web.mac.com/ryo_watanabe/iWeb/Ryo%27s%20Holiday/LEGO%20Mindstorms%20NXT.html
- [40]Tips for Fixed-Point Modeling and Code Generation for Simulink 6
<http://www.mathworks.com/matlabcentral/fileexchange/7197>