

Μάθε το Mindstorms NXT εύκολα και γρήγορα



Σενάρια Ρομποτικής με το Mindstorms NXT

Συγγραφή : Γιώργος Κυριακού

Ιούνιος 2010

στον αγαπημένο μου αδερφό

Περιεχόμενα

Ενότητα 1^η

Τι είναι το Mindstorms NXT	9
<i>Τι είναι το Mindstorms NXT</i>	10
<i>Mindstorms NXT και Προγραμματισμός</i>	12
<i>Mindstorms NXT και Εκπαίδευση</i>	14
<i>Χαρακτηριστικά και Παιδαγωγική αξία</i>	15

Ενότητα 2η

Ρομποτική	19
Τι είναι ένα ρομπότ ;	20
Αλγόριθμος και Ψευδοκώδικας	20
Τι ονομάζουμε πρόγραμμα.....	20
Η γλώσσα προγραμματισμού NXT-G.....	21
Η γλώσσα προγραμματισμού Robolab.....	21

Ενότητα 3η

Τεμάχια πακέτου Mindstorms NXT	22
Πακέτο Lego Mindstorms NXT	23
Τεμάχια του πακέτου Lego Mindstorms.....	24
1. Τουβλάκια (Bricks)	24
2. Πλακίδια (Plates)	24
3. Δοκοί (Beams).....	25
4. Άξονες (Axles)	26
5. Δακτύλιοι (Bushings).....	26
6. Πιράκια (Pegs)	26
7. Σύνδεσμοι (Connectors)	27
8. Ρόδες (Wheels)	28
9. Γρανάζια (Gears).....	28
10. Ιμάντες και Τροχαλίες (Belts and Pulleys)	29
11. Διάφορα τεμάχια	29
12. Κινητήρες και Λαμπτήρες (Motors and Lamps).....	30
13. Αισθητήρες (Sensors)	30
14. Τούβλο ή μικροεπεξεργαστής NXT (NXT Brick)	30
15. Καλώδια	30
Γρανάζια.....	31
1. Γρανάζια με δόντια (Spur gears)	31
Λόγος γραναζιών (gear ratio).....	31
2. Γρανάζια με λοξά δόντια (Bevel gears)	32
3. Δοκός με δόντια (Rack and Pinion)	33
4. Κοχλιωτό γρανάτζι (Worm gears)	33
5. Γρανάζι με ολίσθηση (Slip clutch)	34
Απλοί μηχανισμοί με γρανάτζια :	34
Τροχαλίες και ιμάντες.....	35
Κινητήρες.....	36
Αισθητήρες.....	36
1. Αισθητήρας αφής.....	36
2. Αισθητήρας φωτός.....	37

Σενάρια Ρομποτικής με το Mindstorms NXT

3. Αισθητήρας περιστροφής (Ενσωματωμένος στον κινητήρα).....	38
4. Αισθητήρας ήχου.....	38
5. Αισθητήρας Υπέρηχων.....	39
Αισθητήρες παλιού τύπου (NXT)	40
1. Αισθητήρας αφής.....	40
2. Αισθητήρας φωτός.....	40
3. Αισθητήρας περιστροφής.....	41
4. Αισθητήρας θερμοκρασίας.....	41
Μικροεπεξεργαστής ή τούβλο NXT.....	42
Περιβάλλον προγραμματισμού του NXT.....	45
Πηγαίνετε στην περιοχή <i>NXT Software Download (PC/MAC)</i> και κάνετε 'κλικ' στο κουμπί.....	45
Λειτουργία του NXT.....	46
Υπομενού.....	48
Τα αρχεία μου (My Files)	48
Προγράμματα για δοκιμή (Try Me)	49
Εμφάνιση Πληροφοριών (View)	50
Προγράμματα απ' ευθείας στο NXT (NXT Program)	51
Δειγματοληψία NXT (NXT Datalog)	55
Ρυθμίσεις (Settings)	55
Σύνδεση Bluetooth.....	56
Επιλογή <i>NXT Window</i> από το πεδίο ελέγχου (<i>Controller</i>)	58
Ενότητα 4η	
Προγραμματιστικό περιβάλλον NXT-G.....	61
Το προγραμματιστικό περιβάλλον NXT-G.....	62
Εκκίνηση του προγράμματος.....	62
Εξοικείωση με το NXT-G.....	69
Δραστηριότητα 1η : κίνηση μπροστά για 2 περιστροφές.....	71
Δραστηριότητα 2η : κίνηση προς τα πίσω για 2 περιστροφές.....	78
Δραστηριότητα 3η : κίνηση μπροστά για 2 περιστροφές και επιστροφή.....	79
Δραστηριότητα 4η : Στροφή αριστερά	83
Δραστηριότητα 5η : Στροφή δεξιά	85
Ενότητα 5η	
Δραστηριότητες στο περιβάλλον του NXT-G	86
Λίγα λόγια για τους κινητήρες	87
Δραστηριότητα 6η : κίνηση μπροστά συγχρονισμένα	88
Δραστηριότητα 7η : παρκάρισμα	89
Δραστηριότητα 8η : τετράγωνο	90
Λίγα λόγια για τους αισθητήρες	90
Δραστηριότητα 9η : σταμάτημα με αισθητήρα υπέρηχων	91
Δραστηριότητα 10η : σταμάτημα με αισθητήρα αφής	92
Δραστηριότητα 11η : εκκίνηση με αισθητήρα ήχου και σταμάτημα με αισθητήρα υπέρηχων.....	93
Δραστηριότητα 12η : άναμμα φωτός σε τούνελ.....	95
Δραστηριότητα 13η : υλοποίηση συναγερμού	97
Δραστηριότητα 14η : κίνηση μπροστά μέχρι να σκοτεινιάσει.....	97
Δραστηριότητα 15η : κίνηση μπροστά μέχρι τη μαύρη γραμμή.....	97
Λίγα λόγια για τους ήχους και τις εικόνες	98
Δραστηριότητα 16η : σταμάτημα με αισθητήρα αφής με ενημέρωση.....	98
Λίγα λόγια για τη δομή επιλογής	99
Δραστηριότητα 17η : διακλάδωση με αισθητήρα αφής	99
Δραστηριότητα 18η : διακλάδωση με αισθητήρα φωτός	101

Σενάρια Ρομποτικής με το Mindstorms NXT

Δραστηριότητα 19η : διακλάδωση μέσα σε διακλάδωση	102
Λίγα λόγια για τη δομή επανάληψης (αέναο βρόχο)	103
Δραστηριότητα 20η : εκκίνηση με αισθητήρα ήχου και σταμάτημα	103
με αισθητήρα υπερήχων και επανάληψη	103
Δραστηριότητα 21η : κίνηση φιδωτή	104
Δραστηριότητα 22η : ξεκίνημα και σταμάτημα με αισθητήρα αφής και επανάληψη	105
Δραστηριότητα 23η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη	105
Δραστηριότητα 24η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη και με δυνατότητας τερματισμού	106
Δραστηριότητα 25η : υλοποίηση συναγερμού	106
Δραστηριότητα 26η : κίνηση μπροστά και ήχος για κάθε μαύρη γραμμή	107
Λίγα λόγια για τη δομή επανάληψης (ν φορές)	107
Δραστηριότητα 27η : τετράγωνο με δομή επανάληψης	107
Λίγα λόγια για τη δομή επανάληψης (μέχρι να)	108
Δραστηριότητα 28η : κίνηση μπροστά με σλάλομ και σταμάτημα με αισθητήρα αφής	108
Λίγα λόγια για τις παράλληλες διεργασίες	109
Δραστηριότητα 29η : παράλληλες διεργασίες	109
Δραστηριότητα 30η : παράλληλες διεργασίες	110
Λίγα λόγια για τους κατανεμητές δεδομένων	111
Δραστηριότητα 31η : προβολή τιμής αισθητήρα υπερήχων	111
Λίγα λόγια για τη Δομή επανάληψης με κατανεμητή δεδομένων	112
Δραστηριότητα 32η : προβολή αριθμού επαναλήψεων	113
Λίγα λόγια για τη Δομή επιλογής με κατανεμητή δεδομένων	114
Δραστηριότητα 33η : κίνηση μπροστά μέχρι τη μαύρη γραμμή	115
Δραστηριότητα 34η : Τυχαία κίνηση	116
Δραστηριότητα 35η : υπολογισμός απόστασης	117
Δραστηριότητα 36η : αντίσταση στη μετακίνηση	118
Λίγα λόγια για τις μεταβλητές	118
Δραστηριότητα 37η : μέτρηση αριθμού πατημάτων αισθητήρα αφής	118
Δραστηριότητα 38η : μετατροπή φωτεινότητας σε νότες (sequencer)	120
Δραστηριότητα 39η : μετατροπή φωτεινότητας σε νότες και διαχωρισμός	120
Δραστηριότητα 40η : αναμονή για μετρητή	121
Δραστηριότητα 41η : κίνηση με επιτάχυνση	122
Δραστηριότητα 42η : αλλαγή ταχύτητας με διακόπτη αφής	122
Δραστηριότητα 43η : μέτρηση πατημάτων	123
Λίγα λόγια για τους χρονομετρητές	124
Δραστηριότητα 44η : αναμονή με χρονομετρητή	124
Δραστηριότητα 45η : έλεγχος λάμπας	126
Δραστηριότητα 46η : έλεγχος λάμπας με χρονοκαθυστέρηση	126
Δραστηριότητα 47η : αναμονή για τυχαίο χρόνο	127
Λίγα λόγια για την αποστολή και λήψη μηνυμάτων	128
Δραστηριότητα 48η : σκυταλοδρομία	128
Δραστηριότητα 49η : δημιουργία και αποστολή μηνύματος	129
Ενότητα 6η	
Σύνθετες Δραστηριότητες στο NXT-G	131
A. Αποφυγή εμποδίων	132
Δραστηριότητα 1η : αποφυγή εμποδίου με 1 αισθητήρα αφής	132
Δραστηριότητα 2η : αποφυγή εμποδίου με 2 αισθητήρες αφής	133
Δραστηριότητα 3η : αποφυγή εμποδίου με αισθητήρα υπερήχων	134
A2. Έξοδος από σπηλιά	135
B. Έλεγχος κίνησης με αισθητήρες αφής	135

Σενάρια Ρομποτικής με το Mindstorms NXT

Δραστηριότητα 4η : έλεγχος κίνησης με 2 αισθητήρες αφής με παράλληλες διεργασίες	135
Δραστηριότητα 5η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση.....	136
Δραστηριότητα 6η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση.....	137
μέσα σε διακλάδωση.....	137
Δραστηριότητα 7η : έλεγχος κίνησης με 3 αισθητήρες αφής.....	138
Γ. Έξοδος από κυκλικό σχήμα.....	141
Δραστηριότητα 8η : έξοδος από κυκλικό σχήμα.....	141
Δ. Μέτρηση μαύρων γραμμών.....	143
Δραστηριότητα 9η : μέτρηση μαύρων γραμμών	143
Δραστηριότητα 10η : μπροστά πάνω από 3 μαύρες γραμμές και επιστροφή	144
Ε. Ακολουθώντας τη γραμμή (Line follower).....	144
Δραστηριότητα 11η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι	145
Δραστηριότητα 12η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως	146
Δραστηριότητα 13η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με διακλάδωση	146
Δραστηριότητα 14η: Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με επανάληψη και παράλληλες διεργασίες.....	148
Δραστηριότητα 15η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση και μία διεργασία.....	149
Δραστηριότητα 16η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση μέσα σε διακλάδωση και μία διεργασία	149
Ζ. Ακολουθώντας τον τοίχο.....	152
Δραστηριότητα 17η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα αφής	152
Δραστηριότητα 18η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα υπερήχων.....	152
Η. Άλλα προγράμματα	153
Δραστηριότητα 19η : φύλαξη με αισθητήρα υπερήχων	153
Δραστηριότητα 20η : Σχεδιαστήριο.....	153
Ενότητα 7η	
Διαδικαστικό Σενάριο Ρομποτικής.....	158
Ρομποτική και το πακέτο Mindstorms NXT της Lego.....	174
Γνωριμία με το προγραμματιστικό περιβάλλον NXT-G.....	184
Εντολές κίνησης και Δομή Επανάληψης.....	215
Εντολές αναμονής και Δομή Επιλογής.....	233
Σύνθετη δραστηριότητα “ έξοδος από σπηλιά ”	248
Σύνθετη δραστηριότητα “ ακολουθώντας τη γραμμή ”	263
Ενότητα 8η	
Προγραμματιστικό περιβάλλον ROBO LAB 2.9.....	282
Το προγραμματιστικό περιβάλλον Robolab	283
Εκκίνηση του προγράμματος	284
Ρυθμίσεις στο πρόγραμμα Robolab	286
Γνωριμία με το πρόγραμμα Robolab	289
Δραστηριότητες.....	290
Παλέτα Εργαλείων	291
Δραστηριότητες.....	291
Διαδικασία για τη δημιουργία ενός προγράμματος στο inventor	292
Δραστηριότητες.....	292
Παλέτες εντολών του Inventor.....	296
Εξοικείωση με το Robolab	299
Δραστηριότητα 1η : κίνηση δεξιά για 2s	301

Σενάρια Ρομποτικής με το Mindstorms NXT

Δραστηριότητα 2η : κίνηση μπροστά για 6s	307
Δραστηριότητα 3η : κίνηση μπροστά για 6s και επιστροφή.....	310
Ενότητα 9η	313
Δραστηριότητες στο περιβάλλον του ROBOLAB.....	313
Λίγα λόγια για τους κινητήρες	314
Δραστηριότητα 3η : κίνηση μπροστά για 4s στάση και επιστροφή (β' έκδοση)	315
Δραστηριότητα 4η : κίνηση σε καμπύλη δεξιά	315
Δραστηριότητα 4η : κίνηση σε καμπύλη (β' έκδοση).....	316
Δραστηριότητα 5η : στροφή επιτόπου δεξιά.....	316
Δραστηριότητα 5η : στροφή επιτόπου αριστερά (β' έκδοση)	316
Δραστηριότητα 6η : κίνηση μπροστά για 6s και στροφή επιτόπου	317
Δραστηριότητα 7η : παρκάρισμα	317
Δραστηριότητα 7η : κίνηση και ελιγμοί (β' έκδοση)	317
Δραστηριότητα 8η : κίνηση σε τετράγωνο	318
Δραστηριότητα 9η : χορός Salsa	318
Δραστηριότητα 10η : ξεκίνημα με επιτάχυνση.....	319
Λίγα λόγια για τον αισθητήρα αφής	319
Δραστηριότητα 11η : ξεκίνημα με αισθητήρα αφής.....	320
Δραστηριότητα 12η : ξεκίνημα και σταμάτημα με αισθητήρα αφής.....	320
Δραστηριότητα 13η : ξεκίνημα, αλλαγή κατεύθυνσης και σταμάτημα με αισθητήρα αφής.....	321
Δραστηριότητα 14η : ξεκίνημα και σταμάτημα με δύο αισθητήρες αφής.....	321
Λίγα λόγια για τον αισθητήρα φωτός	321
Δραστηριότητα 15η : υλοποίηση συναγερμού.....	322
Δραστηριότητα 16η : άναμμα φωτός σε τούνελ.....	322
Δραστηριότητα 16η : κίνηση μπροστά μέχρι να σκοτεινιάσει (β' έκδοση)	323
Δραστηριότητα 17η : κίνηση μπροστά μέχρι τη μαύρη γραμμή.....	323
Δραστηριότητα 18η : κίνηση μπροστά μέχρι τη μαύρη γραμμή και πίσω	323
Λίγα λόγια για τον αισθητήρα περιστροφής	324
Δραστηριότητα 19η : κίνηση μπροστά για 1 περιστροφή.....	324
Λίγα λόγια για τη δομή επιλογής	325
Δραστηριότητα 20η : νότες με διακλάδωση αφής.....	328
Δραστηριότητα 20η : κίνηση με διακλάδωση αφής (β' έκδοση)	328
Δραστηριότητα 21η : νότες με διακλάδωση φωτός.....	328
Δραστηριότητα 21η : κίνηση με διακλάδωση φωτός (β' έκδοση)	329
Λίγα λόγια για τη δομή επανάληψης (αέναο βρόχο).....	329
Δραστηριότητα 22η : κίνηση φιδωτή	330
Δραστηριότητα 22η : κίνηση σύρσιμο (β' έκδοση)	331
Δραστηριότητα 23η : ήχος συνέχεια	331
Δραστηριότητα 24η : νότες με διακλάδωση αφής και επανάληψη με άλμα	331
Δραστηριότητα 25η : έλεγχος λάμπας με αισθητήρα αφής.....	331
Δραστηριότητα 26η : ξεκίνημα και σταμάτημα με αισθητήρα αφής και επανάληψη με άλμα.....	332
Δραστηριότητα 27η : εμπόδιο με αισθητήρα αφής.....	332
Δραστηριότητα 28η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη με άλμα ...	332
Δραστηριότητα 28η : ξεκίνημα και σταμάτημα με δυνατότητας τερματισμού με διακλάδωση αισθητήρα αφής και επανάληψη με άλμα (β' έκδοση)	333
Δραστηριότητα 29η : υλοποίηση συναγερμού με διακλάδωση αισθητήρα φωτός και επανάληψη με άλμα ..	333
Δραστηριότητα 30η : κίνηση μπροστά και ήχος για κάθε μαύρη γραμμή	333
Λίγα λόγια για τη δομή επανάληψης (επανάλαβε μέχρι)	334
Δραστηριότητα 31η : ήχος 10 φορές.....	335
Δραστηριότητα 32η : νότες με διακλάδωση αφής και επανάληψη 6 φορές	335
Δραστηριότητα 33η : κίνηση σε τετράγωνο με επανάληψη.....	335

Σενάρια Ρομποτικής με το Mindstorms NXT

Λίγα λόγια για τις παράλληλες διεργασίες.....	336
Δραστηριότητα 34η : παράλληλες διεργασίες	337
Δραστηριότητα 35η : τρεις παράλληλες διεργασίες.....	337
Λίγα λόγια για τις υπορουτίνες.....	338
Δραστηριότητα 36η : χορευτικές κινήσεις.....	338
Λίγα λόγια για τις μεταβλητές	339
Δραστηριότητα 37η : αναμονή για μετρητή	340
Δραστηριότητα 38η : αλλαγή φωτεινότητας λάμπας	341
Δραστηριότητα 39η : μετατροπή φωτεινότητας σε νότες (sequencer).....	341
Δραστηριότητα 40η : αντίσταση στη μετακίνηση.....	341
Δραστηριότητα 41η : μέτρηση πατημάτων	342
Λίγα λόγια για τους χρονομετρητές	343
Δραστηριότητα 42η : αναμονή αριθμού δευτερολέπτων	344
Δραστηριότητα 43η : έλεγχος λάμπας.....	345
Δραστηριότητα 44η : έλεγχος λάμπας με χρονοκαθυστέρηση	345
Λίγα λόγια για την αποστολή και λήψη μηνυμάτων.....	346
Δραστηριότητα 45η : σκυταλοδρομία.....	346
Δραστηριότητα 46η : δημιουργία και αποστολή μηνύματος	347
Δραστηριότητα 47η : αναπαραγωγή νότας σύμφωνα με εισερχόμενο μήνυμα	347
Ενότητα 10η	
Σύνθετες Δραστηριότητες για το Robolab.....	348
A. Αποφυγή εμποδίων	349
Δραστηριότητα 1η : αποφυγή εμποδίου με 1 αισθητήρα αφής.....	349
Δραστηριότητα 2η : αποφυγή εμποδίου με 2 αισθητήρες αφής	350
Δραστηριότητα 3η : αποφυγή εμποδίου με αισθητήρα υπερήχων.....	351
A2. Έξοδος από σπηλιά	351
B. Έλεγχος κίνησης με αισθητήρες αφής	352
Δραστηριότητα 4η : έλεγχος κίνησης με 2 αισθητήρες αφής με παράλληλες διεργασίες	352
Δραστηριότητα 5η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση.....	353
Δραστηριότητα 6η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση μέσα σε διακλάδωση.....	353
Δραστηριότητα 7η : έλεγχος κίνησης με 2 αισθητήρες περιστροφής.....	355
Δραστηριότητα 8η : έλεγχος κίνησης με 3 αισθητήρες αφής.....	355
Δραστηριότητα 8η : έλεγχος κίνησης με 3 αισθητήρες αφής και 3 κινητήρες (8' έκδοση)	357
Γ. Έξοδος από κυκλικό σχήμα	358
Δραστηριότητα 9η : έξοδος από κυκλικό σχήμα.....	358
Δ. Μέτρηση μαύρων γραμμών.....	360
Δραστηριότητα 10η : μέτρημα μαύρων γραμμών	361
Δραστηριότητα 11η : μπροστά πάνω από 3 μαύρες γραμμές και επιστροφή	361
E. Ακολουθώντας τη γραμμή (Line follower).....	362
Δραστηριότητα 12η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι	362
Δραστηριότητα 13η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως	363
Δραστηριότητα 14η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με διακλάδωση	364
Δραστηριότητα 15η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με εντολή αναμονής για σκοτάδι	366
Δραστηριότητα 16η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως	366
Δραστηριότητα 17η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση και παράλληλες διεργασίες.....	367

Σενάρια Ρομποτικής με το Mindstorms NXT

Δραστηριότητα 17η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με επανάληψη και παράλληλες διεργασίες (β' έκδοση)	367
Δραστηριότητα 18η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση και μία διεργασία.....	368
Δραστηριότητα 19η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση μέσα σε διακλάδωση και μία διεργασία	368
Ζ. Ακολουθώντας τον τοίχο.....	371
Δραστηριότητα 20η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα αφής	371
Δραστηριότητα 21η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα υπερήχων.....	371
Η. Άλλα προγράμματα	372
Δραστηριότητα 22η : φύλαξη με αισθητήρα υπερήχων	372
Δραστηριότητα 23η : Σχεδιαστήριο.....	372

ΠΑΡΑΡΤΗΜΑ Ι

Παρουσίαση των εντολών του NXT-G	373
Βασική Παλέτα (Common Palette).....	375
Πλήρης παλέτα (Complete Palette).....	393
Μπλοκ Ενεργειών.....	393
Action Blocks	393
Μπλοκ Αισθητήρων.....	397
Sensor Blocks.....	397
Μπλοκ Ροής - Δομές	411
Flow Blocks.....	411
Μπλοκ Δεδομένων	412
Data Blocks	412
Μπλοκ για προχωρημένους.....	419
Advanced Blocks.....	419
Προσαρμοσμένη παλέτα (Custom Palette)	426

ΠΑΡΑΡΤΗΜΑ ΙΙ

Παρουσίαση των εντολών του Robolab 2.9	429
Εντολές αρχής και τέλους	431
Εντολές εξόδου (Output Commands).....	432
Προχωρημένες εντολές εξόδου (Advanced Output Commands)	437
Εντολές αναμονής (Wait for Commands).....	441
Δομές (Structures)	452
Δομές Επανάληψης (Loop)	468
Παράλληλες Διεργασίες	480
Υπορουτίνες.....	481
Μουσική.....	482
Μηδενισμός (Reset).....	484
Μεταβλητές (Containers).....	490
Επικοινωνία μεταξύ των NXT (NXT Communication)	501
Τροποποιητές (Modifiers).....	513

ΠΑΡΑΡΤΗΜΑ ΙΙΙ

Κατασκευή οχήματος – ρομπότ TankBot NXT	519
Βιβλιογραφία	529



Ενότητα 1η

Τι είναι το Mindstorms NXT

Τι είναι το Mindstorms NXT

Το **Mindstorms NXT** της Lego (<https://education.lego.com>) είναι ένα εκπαιδευτικό πακέτο ρομποτικής το οποίο αποτελεί ένα ευέλικτο μέσο δημιουργικής μάθησης που παρέχει ευκαιρίες για σχεδιασμό και κατασκευές σε μικρό χρονικό διάστημα και με μικρό προϋπολογισμό. Περιλαμβάνει ένα σύνολο από δομικά στοιχεία (τουβλάκια, γρανάζια, τροχούς, κ.α.), αισθητήρες, κινητήρες και άλλα εξαρτήματα με τα οποία μπορεί κανείς να κτίσει φυσικά μοντέλα. Τα εξαρτήματα αυτά προσαρμόζονται πάνω σε ένα τουβλάκι της LEGO μεγαλύτερων διαστάσεων, στο οποίο είναι ενσωματωμένος ο επεξεργαστής NXT.

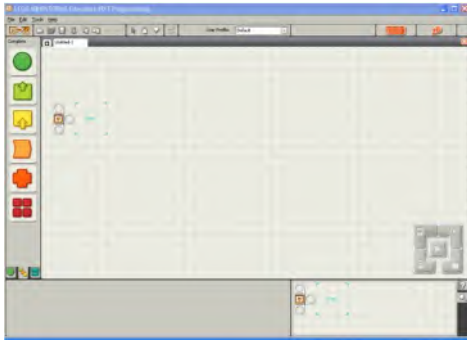


Ο NXT είναι ένας αυτόνομος μικροϋπολογιστής που ενσωματώνεται σε ένα τουβλάκι της LEGO, μπορεί να προγραμματιστεί μέσω ενός υπολογιστή και επιτρέπει στην κατασκευή να αλληλεπιδρά με το περιβάλλον αυτόνομα και ανεξάρτητα από τον υπολογιστή. Μια τυπική συμπεριφορά της ρομποτικής κατασκευής είναι η αντίδραση σε ένα πιθανό ερέθισμα, ένα χαρακτηριστικό ξεκάθαρα ανθρωπομορφικό. Ο μικροϋπολογιστής αυτός αποτελεί τον εγκέφαλο των μοντέλων και διαθέτει 5 θέσεις εισόδου (1,2,3,4) για αισθητήρες από τους οποίους συλλαμβάνει συμβάντα ή καταστάσεις του περιβάλλοντος (όπως αισθητήρες υπερήχων και αισθητήρες αφής για αναγνώριση εμποδίων, αισθητήρες εντάσεως του ήχου και του φωτός) και ενεργεί αναλόγως, καθώς και τρεις θέσεις εξόδου (A,B,C) για κινητήρες και λαμπτήρες, ώστε να θέτει σε κίνηση το ρομπότ, μια οθόνη υγρών κρυστάλλων (LCD), ένα μικρόφωνο και τέσσερα κουμπιά (αφορούν το άνοιγμα – κλείσιμο του NXT, την επιλογή του προγράμματος που θα εκτελεστεί και την έναρξη – τερματισμό της εκτέλεσης). Η προγραμματιζόμενη μονάδα NXT (τούβλο NXT) μπορεί να αποθηκεύει διαφορετικά προγράμματα στη μνήμη της τα οποία μεταδίδονται σε αυτήν από τον Η/Υ του χρήστη μέσω θύρας USB ή εναλλακτικά μέσω σύνδεσης Bluetooth και να τα εκτελεί σε πλήρη αυτονομία.

Οι ρομποτικές κατασκευές με τη χρήση κατάλληλων περιβαλλόντων ανάπτυξης προγραμμάτων μπορούν να προγραμματιστούν ώστε να εκτελούν μία σειρά ενεργειών και να αντιδρούν σε ερεθίσματα που δέχονται οι αισθητήρες τους.

Το προγραμματιστικό περιβάλλον NXT-G που συνοδεύει το πακέτο της Mindstorms NXT παρέχει ένα απλό γραφικό περιβάλλον για τον προγραμματισμό “συμπεριφορών” των ρομποτικών κατασκευών της Lego. Βασίζεται στη χρήση εικονιδίων. Ο προγραμματισμός ακολουθεί τις αρχές του οπτικού προγραμματισμού, όπου ο χρήστης (μαθητής) επιλέγει από μια συλλογή εικονιδίων που αντιστοιχούν σε συγκεκριμένες δράσεις και τη λογική διασύνδεση μεταξύ τους.





Το λογισμικό αυτό έχει μια διαισθητική διεπαφή “σύρε και άφησε” (drag and drop) και ένα γραφικό προγραμματιστικό περιβάλλον, το οποίο καθιστά την εφαρμογή προσιτή για έναν αρχάριο, αλλά και εξίσου δυναμική για έναν εξειδικευμένο χρήστη. Η πλήρης παλέτα εικονιδίων περιλαμβάνει εντολές δράσης που επιτρέπουν τον έλεγχο διάφορων εξωτερικών συσκευών (διαδραστικού κινητήρα, ήχων, λαμπτήρων κ.ά.) Οι εντολές ροής επιτρέπουν τη δημιουργία σύνθετων συμπεριφορών. Περιλαμβάνουν τον έλεγχο για την επανάληψη, την αναμονή και τις συνθήκες μεταβλητών, για τη διακοπή συμπεριφοράς ή τον καθορισμό μιας λογικής σειράς σε ένα πρόγραμμα και τη λήψη αποφάσεων για τον προγραμματισμό αντιδράσεων σε καθορισμένες τιμές των αισθητήρων. Οι παλέτες προγραμματισμού προσφέρουν όλες τις δομές προγραμματισμού που απαιτούνται για να δημιουργηθούν τα προγράμματα. Ένα πρόγραμμα δημιουργείται με συνδυασμό διαφορετικών εντολών. Τα διαθέσιμα εικονίδια-εντολές περιλαμβάνουν μεταξύ άλλων εντολές κίνησης (κάνουν τα ρομπότ να κινούνται), εντολές αναμονής (κάνουν το ρομπότ να περιμένει για την ενεργοποίηση των αισθητήρων του ή για τη λήξη ενός οριζόμενου χρονικού διαστήματος), εντολές επανάληψης (Loop) (το ρομπότ επαναλαμβάνει την ίδια συμπεριφορά όσες φορές ορίσουμε ή μέχρι να ενεργοποιηθεί κάποιος αισθητήρας), εντολές επιλογής (If) (επιτρέπουν στο ρομπότ να παίρνει τις δικές του αποφάσεις).



Λόγω των δυνατοτήτων του NXT να ελέγχει κινητήρες ή φώτα και να συγκεντρώνει δεδομένα με τη βοήθεια αισθητήρων, παιδιά και ενήλικες μπορούν εύκολα να δημιουργήσουν κατασκευές που κινούνται, σκέφτονται, και αντιδρούν (Portsmore 1999).

Η φιλοσοφία του εκπαιδευτικού υλικού της εταιρείας Lego, στηρίζεται στο ότι το παιδί πρέπει από μόνο του να οικοδομεί τη γνώση και ειδικότερα στην άποψη ότι η μάθηση επέρχεται μέσα από το παιχνίδι (*“learning through play”*).

Οι κατασκευές αυτές δίνουν τη δυνατότητα διεξαγωγής νέων τύπων επιστημονικών πειραμάτων με τα οποία οι μαθητές ερευνούν καθημερινά φαινόμενα της ζωής τους εντός και εκτός της σχολικής αίθουσας.

Η δημιουργία του αυτόνομου ρομπότ συνίσταται σε 4 βήματα:

1. Κατασκευή του ρομπότ σύμφωνα με τα σχέδια που παρέχει το kit ή τη φαντασία του χρήστη.
2. Ανάπτυξη προγράμματος, χρησιμοποιώντας το οπτικό προγραμματιστικό περιβάλλον NXT-G που συνοδεύει το kit ή μια άλλη γλώσσα προγραμματισμού.
3. Φόρτωση του προγράμματος στο ρομπότ, με τη μεταφορά του προγράμματος από τον υπολογιστή στο μικροϋπολογιστή NXT.
4. Εκτέλεση του προγράμματος.

Για να εκδηλώσει το ρομπότ μια συγκεκριμένη συμπεριφορά, συνήθως τα βήματα 2 έως 4 επαναλαμβάνονται αρκετές φορές (Patterson-McNeill, Binkerd, 2001).



Mindstorms NXT και Προγραμματισμός



Σύμφωνα με τους Κόμη Β. και Τσοβόλα Σπ. (Προγραμματισμός ρομποτικών κατασκευών. Μια μελέτη περίπτωσης με μαθητές δημοτικού, 2008), ο προγραμματισμός αποτελεί μια δεξιότητα, με την οποία δεν μπορεί να συγκριθεί καμιά άλλη γνωστική δεξιότητα. Η δεξιότητα αυτή έχει σαφή παιδαγωγική διάσταση κυρίως όσον αφορά τις διαδικασίες ανάλυσης προβλημάτων ή καταστάσεων, που προηγούνται της συγγραφής του προγράμματος, και συμβάλλει

σε μεγάλο βαθμό στην απόκτηση μεθόδων στρατηγικής και στην ανάπτυξη της λογικής σκέψης. Πρόκειται λοιπόν για μια νοητική δεξιότητα υψηλού επιπέδου που εντάσσεται στη μεγάλη κατηγορία έργων που οι ψυχολόγοι ονομάζουν επίλυση προβλημάτων.

Για την αντιμετώπιση των δυσκολιών στη διδασκαλία του προγραμματισμού έχουν κατά καιρούς προταθεί μία σειρά από εναλλακτικούς τρόπους διδασκαλίας. Μία συστηματική απαρίθμηση των προτάσεων αυτών μπορούμε να συναντήσουμε και πάλι στη μελέτη περίπτωσης των Χαρίση Χ. και Μικρόπουλο Τ. Α (Ρομποτική, Οπτικός Προγραμματισμός και Βασικές Προγραμματιστικές Δομές, 2008). Μεταξύ αυτών συγκαταλέγονται η χρήση μικρόκοσμων και μικρογλωσσών (Logo, Karel the Robot), η βελτίωση των διαγνωστικών δυνατοτήτων των μεταγλωττιστών, τα κατανοητά μηνύματα λάθους, η χρήση εικονικών γλωσσών προγραμματισμού, συστήματα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων, η χρήση τεχνικών δυναμικής προσομοίωσης εκτέλεσης αλγορίθμων, η δυναμική ηχητική προσομοίωση της εκτέλεσης προγραμμάτων (Ξυνόγαλος κ.α., 2000). Επίσης, προτείνεται η χρήση λογισμικών με οπτικοποίηση των διεργασιών, των τιμών των μεταβλητών και των δομών προγραμματισμού κατά την εκτέλεση προγραμμάτων (Κάππας κ.α., 2002; Yehezkeila et al., 2007), η διδασκαλία αλγορίθμων σε περιβάλλον συνεργατικής μάθησης με εποπτικό μέσο διδασκαλίας τον υπολογιστή (Βογιατζάκη κ.α., 2004), η χρήση λογισμικού παρουσιάσεων (Μπέλλου & Μικρόπουλος, 2005) και οι διαγραμματικές αναπαραστάσεις επαναληπτικών δομών (Carlisle, 2000).

Στο πλαίσιο μιας εποικοδομιστικής προσέγγισης στη μάθηση του προγραμματισμού, δεν πρέπει να επικεντρωθούμε στη διδασκαλία των εννοιών του προγραμματισμού, αλλά πρέπει να βοηθήσουμε τους μαθητές να οικοδομήσουν τα απαραίτητα νοητικά πλαίσια για να αποκτήσουν από μόνοι τους προγραμματιστικές δεξιότητες (Κόμης, 2005).

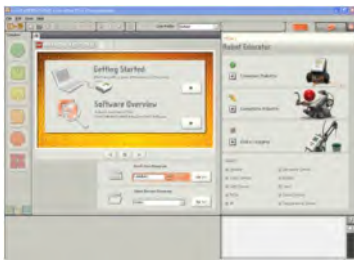
Το λογισμικό NXT-G είναι ένα βασισμένο σε ένα περιβάλλον οπτικού προγραμματισμού που επιτρέπει στο χρήστη να προγραμματίσει χρησιμοποιώντας εικόνες που αναπαριστούν όλους τους τύπους δεδομένων και τις βασικές εντολές και δομές. Έτσι, η διδασκαλία του προγραμματισμού δεν ακολουθεί το τυπικό μοντέλο διδασκαλίας αλλά μια κατασκευαστική προσέγγιση (Τσοβόλας & Αντωνίου, 2005). Ένα από τα βασικότερα πλεονεκτήματα τέτοιων γλωσσών προγραμματισμού είναι η ελαχιστοποίηση των συντακτικών κανόνων με τις οποίες έρχεται αντιμέτωπος ο μαθητής. Η διαδικασία σύνταξης ενός προγράμματος είναι αρκετά απλή και βασίζεται στη σωστή σύνδεση των



Σενάρια Ρομποτικής με το Mindstorms NXT

κατάλληλων εικονιδίων. Οι χρήστες στη γλώσσα προγραμματισμού NXT-G, μαθαίνουν να προγραμματίζουν μέσω οπτικών εντολών χρησιμοποιώντας την τεχνική του συρσίματος (drag & drop) για να δημιουργήσουν προγράμματα και δε χρειάζεται να έχουν γνώσεις πάνω στο υλικό (Κασκάλης κ.α., 2001). Η ανάπτυξη προγραμμάτων με τη χρήση εικονο-εντολών απαλλάσσει τους μαθητές από την εκμάθηση μιας γλώσσας προγραμματισμού και την απομνημόνευση των συντακτικών της κανόνων. Ειδικά τα εικονίδια λειτουργίας των εξόδων είναι προφανή. Στο περιβάλλον συγγραφής του κώδικα γίνεται ταυτόχρονα και συντακτικός έλεγχος, έτσι απαλλάσσεται ο προγραμματιστής από συντακτικά λάθη. (Τσοβόλας & Κόμης, 2005).

Η διδασκαλία του προγραμματισμού στα πλαίσια της προσέγγισης αυτής επικεντρώνεται στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων και σχεδίασης αλγορίθμων και όχι στην εκμάθηση της γλώσσας προγραμματισμού και είναι ιδιαίτερα αποτελεσματική (Ξυνόγαλος κ.α. 2000, Lawhead et al 2002).



Ο προγραμματισμός με τη χρήση του λογισμικού NXT-G λοιπόν αποφέρει θετικά μαθησιακά αποτελέσματα με χαρακτηριστικό τη χρήση της σωστής και επιθυμητής δομής σε κάθε περίπτωση. Επίσης, αυξάνει και διατηρεί το κίνητρο των μαθητών για τον προγραμματισμό.

Με αυτήν τη μέθοδο διδασκαλίας ο εκπαιδευτικός καταφέρνει να διδάξει την ίδια θεωρία και τις ίδιες έννοιες όχι μόνο πιο αποτελεσματικά αλλά και σε λιγότερο χρόνο (Fagin 2000). Τα εικονίδια που χρησιμοποιούνται, αναπαριστούν τις εντολές αρκετά επιτυχημένα. Ειδικά τα εικονίδια των εντολών εξόδου είναι προφανή. Στην εντολή επιλογής (Fork) πχ μπορεί κανείς να δει τις αντίστοιχες διαδρομές που θα ακολουθηθούν ανάλογα με το αποτέλεσμα από τον έλεγχο της συνθήκης. Η παραστατικότητα των εικονιδίων επιτρέπει στον εκπαιδευτικό να αφιερώσει λιγότερο χρόνο στην παρουσίαση της θεωρίας, αφήνοντας στους μαθητές περισσότερο χρόνο για πρακτική εφαρμογή και του δίνει την απαραίτητη άνεση χρόνου για συστηματική παρακολούθηση της όλης εκπαιδευτικής διαδικασίας και των δυσκολιών που εμφανίζονται. Το γεγονός αυτό σημαίνει ότι ο εκπαιδευτικός έχει στη διάθεσή του περισσότερο χρόνο για να παρακολουθήσει την πορεία του κάθε μαθητή ξεχωριστά και να εντοπίσει τις αδυναμίες, τις παρανοήσεις και τις δυσκολίες που αυτός αντιμετωπίζει. Επίσης, ο μαθητής εργάζεται ο καθένας με το δικό του ρυθμό μάθησης, έτσι ώστε να μπορεί να σταθεί όσο χρόνο χρειάζεται στα σημεία που τον ενδιαφέρουν ή το δυσκολεύουν. Μπορεί κανείς δηλαδή να μιλάει για εξατομικευμένη μάθηση.

Το τελικό αποτέλεσμα όπως σημειώνεται από τους Χαρίσης Χ. και Μικρόπουλο Τ. Α (Ρομποτική, Οπτικός Προγραμματισμός και Βασικές Προγραμματιστικές Δομές, 2008) είναι ο μαθητής να μπορεί να δημιουργήσει ένα πρόγραμμα για να κατευθύνει τη ρομποτική του κατασκευή σε σύντομο χρονικό διάστημα. Δεν χρειάζεται «να μάθει πολλά για να κάνει λίγα» αλλά με μια απλή επίδειξη του περιβάλλοντος, μπορεί να κατασκευάσει αμέσως ένα πρόγραμμα για να μπορέσει να ελέγξει μία ρομποτική κατασκευή.

Ένα μεγάλο πλεονέκτημα από τη χρήση των Lego Mindstorms στη διδασκαλία του προγραμματισμού αποτελεί το γεγονός ότι η εκτέλεση των προγραμμάτων που δημιουργούν οι μαθητές δεν γίνεται στον ίδιο τον υπολογιστή στον οποίο θα δημιουργήσουν το πρόγραμμα, αλλά σε μία άλλη εντελώς

Σενάρια Ρομποτικής με το Mindstorms NXT

διαφορετική μηχανή η οποία λειτουργεί αυτόνομα και ανεξάρτητα με τον παραπάνω υπολογιστή : Όταν οι μαθητές μαθαίνουν να προγραμματίζουν με τις γλώσσες γενικού σκοπού χρησιμοποιούν τον υπολογιστή ως μηχανή για να αναπτύξουν το πρόγραμμά τους και ταυτόχρονα χρησιμοποιούν την ίδια μηχανή για να εκτελέσει το πρόγραμμα ώστε να ελέγξουν τη συμπεριφορά του ίδιου του υπολογιστή. Προγραμματίζοντας όμως τα Lego Mindstorms, χρησιμοποιείται ο υπολογιστής για την ανάπτυξη του προγράμματος, αλλά η εκτέλεση του πραγματοποιείται από μια άλλη μηχανή - το μικρουπολογιστή NXT - που ελέγχει με τη σειρά του τη συμπεριφορά του ρομπότ.

Mindstroms NXT και Εκπαίδευση

Μια διαφορετική προσέγγιση διδασκαλίας είναι αυτή που αξιοποιεί την εκπαιδευτική ρομποτική σε συνδυασμό με τον οπτικό προγραμματισμό. Με την εισαγωγή και αξιοποίηση των προγραμματιζόμενων ρομποτικών κατασκευών στη δευτεροβάθμια εκπαίδευση οι μαθητές καλούνται να εργαστούν πάνω σε ένα ρεαλιστικό σχέδιο εργασίας (project). Το θεωρητικό πλαίσιο στην όλη διαδικασία που ακολουθείται από τους μαθητές στηρίζεται στον εποικοδομισμό και ιδιαίτερα οι αρχές του κατασκευαστικού εποικοδομισμού. Σε αυτή την κατηγορία ανήκει και το εκπαιδευτικό πακέτο ρομποτικής Lego Mindstorms NXT Education.



Το πακέτο αυτό περιλαμβάνει ένα μεγάλο πλήθος από δομικά στοιχεία, με τα οποία οι μαθητές έχουν την ευκαιρία να συνθέσουν μία μηχανική κατασκευή στηριζόμενοι στη γνώση και την εμπειρία που έχουν αποκομίσει από το φυσικό τους περιβάλλον αλλά και να την επεκτείνουν με εφόδιο τη φαντασία που διαθέτουν. Επίσης διαθέτει και ένα φιλικό και εύχρηστο γραφικό περιβάλλον προγραμματισμού, με το οποίο οι μαθητές έχουν τη δυνατότητα να δώσουν ζωή στο δημιούργημά τους.



Με το πλήθος των εξαρτημάτων που διαθέτει δίνει τη δυνατότητα στους μαθητές να δημιουργήσουν ένα φυσικό μηχανικό μοντέλο το οποίο σε συνδυασμό με τον κατάλληλο προγραμματισμό του, μπορεί να κινείται και είναι σε θέση να ανταποκρίνεται σε μία σειρά ερεθισμάτων που θα δεχτεί από το περιβάλλον στο οποίο θα το βάλουμε να δραστηριοποιηθεί. Με αυτό τον τρόπο οι μαθητές μπορούν να δημιουργήσουν ένα είδος αυτόματου μηχανισμού, δηλ ένα ρομπότ.

Τα τουβλάκια Lego Mindstorms έχουν χρησιμοποιηθεί συστηματικά τα τελευταία χρόνια για την εισαγωγή αρχάριων μαθητών στην εκμάθηση του προγραμματισμού (Lego Dacta A/S, 1999; Δαγδιλέλης, Σαρτατζέμη & Καγκάνη, 2005; Beisser, 2006; Hussain, Lindh, & Shukur, 2006). Η σειρά Lego Mindstorms έχει χρησιμοποιηθεί σε αρκετές περιπτώσεις στη διδασκαλία της τεχνολογίας και των φυσικών επιστημών από την πρωτοβάθμια ως την πανεπιστημιακή εκπαίδευση για την υποστήριξη της εποικοδομιστικής μάθησης (Resnick et al., 1996; Portsmouth, 1999; Αλιμήσης κ.α., 2005; Καρατράντου κ.ά., 2006; Αλιμήσης κ.α., 2007) αλλά και στη διδασκαλία βασικών αρχών προγραμματισμού και εισαγωγικών εννοιών της επιστήμης των υπολογιστών (Lawhead et al., 2002; Fagin & Merkle, 2003; Καρατράντου κ.ά., 2005; Καγκάνη κ.ά., 2005).

Θετικό στοιχείο της προσέγγισης αυτής αποτελεί το γεγονός ότι πολλοί μαθητές έχουν εξοικειωθεί με τα τουβλάκια της Lego από μικρές ηλικίες (Ringwood et al., 2005) και αποτελούν μια χαμηλού κόστους λύση (Hirst et al., 2002). Με το πακέτο ρομποτικής της Lego ενισχύονται οι βασικοί στόχοι της διδασκαλίας του προγραμματισμού (Lawhead et al., 2003).

Χαρακτηριστικά και Παιδαγωγική αξία

Η ρομποτική μπορεί να αποτελέσει ένα πολύ χρήσιμο εργαλείο στην εκπαιδευτική διαδικασία. Η χρήση της ρομποτικής για εκπαιδευτικούς σκοπούς δίνει στους μαθητές τη δυνατότητα να δημιουργήσουν καταρχήν μία μηχανική κατασκευή και στη συνέχεια μέσω ενός απλού και φιλικού προγραμματιστικού περιβάλλοντος να την προμηθεύσουν με τις κατάλληλες οδηγίες και με αυτό τον τρόπο να δώσουν λύση σε ένα ρεαλιστικό πρόβλημα. Η εκπαιδευτική ρομποτική παρέχει ένα καινοτόμο περιβάλλον το οποίο εμπλέκει ενεργά τους μαθητές στην αναζήτηση λύσεων σε πραγματικά προβλήματα, ενισχύοντας με αυτό τον τρόπο τη διερευνητική τους στάση καθώς τους επιτρέπει να πειραματιστούν και μέσα από διάφορες δοκιμές να οδηγηθούν αβίαστα στο τελικό αποτέλεσμα.



Όπως σημειώνουν οι Νικολός Δ., Καρατράντου Α. και Παναγιωτακόπουλος Χ (Αξιοποίηση του MicroWorlds EX Robotics για την κατανόηση βασικών δομών προγραμματισμού, 2006) τα Lego Mindstorms αλλάζουν ριζικά τη λογική διδασκαλίας του προγραμματισμού αφού οι μαθητές δεν επικεντρώνονται πλέον σε μία σειρά υπολογισμών αλλά ο σκοπός τους είναι να αποδώσουν μία συγκεκριμένη συμπεριφορά σε μία ρομποτική κατασκευή ώστε αυτή να μπορέσει να αλληλεπιδρά με τον ενδεδειγμένο τρόπο με το άμεσο φυσικό περιβάλλον του. Τα δεδομένα του προβλήματος δεν είναι πλέον απλές τιμές αλλά οντότητες που θα παρατηρηθούν και τα αποτελέσματα δεν είναι νέες τιμές αλλά ενέργειες μέσα σε μια δυναμική διαδικασία. Μια τέτοια λογική υλοποιείται σε ένα αντικειμενοστραφές πλαίσιο όπου η συσχέτιση ανάμεσα στη συμπεριφορά του φυσικού αντικειμένου (της ρομποτικής κατασκευής) και στο πρόγραμμα του μαθητή είναι ισχυρή και άμεσα παρατηρήσιμος (Lawhead et al., 2002).



Ένα μαθησιακό περιβάλλον εκπαιδευτικής ρομποτικής εμπλέκει τους μαθητές με δύο διαφορετικές δραστηριότητες : μια κατασκευαστική και μια προγραμματιστική. Όπως περιγράφεται από του Τσοβόλα Σπ. και Κόμης Β (Προγραμματισμός ρομποτικών κατασκευών μελέτη περίπτωσης με μαθητές δημοτικού, 2008) τα βασικά χαρακτηριστικά των ρομποτικών κατασκευών είναι η δυνατότητα συλλογής πληροφοριών/ερεθισμάτων από το περιβάλλον και εκτέλεσης ενεργειών, παρουσιάζοντας συγκεκριμένες συμπεριφορές ως αντίδραση στα εξωτερικά ερεθίσματα. Κατασκευαστικά οι

ρομποτικές κατασκευές συνδυάζουν αισθητήρες αφής, φωτός, θερμοκρασίας κλπ. με τους οποίους συλλέγουν πληροφορίες από το περιβάλλον και συσκευές εξόδου όπως κινητήρες, λαμπτήρες κλπ με τα οποία εκτελούν συγκεκριμένες ενέργειες. Απαραίτητο συμπλήρωμα των παραπάνω είναι το λογισμικό με το οποίο γίνεται ο προγραμματισμός των κατασκευών.

Σενάρια Ρομποτικής με το Mindstorms NXT

Οι μαθητές καλούνται να κατασκευάσουν μία μηχανική κατασκευή αξιοποιώντας ένα σύνολο δομικών υλικών, όπως για παράδειγμα τα Lego Mindstorms, για την εξοπλίσουν (αποφασίζοντας για τους κατάλληλους αισθητήρες τους οποίους θα ενσωματώσουν στην κατασκευή αυτή) και στη συνέχεια να δημιουργήσουν το κατάλληλο πρόγραμμα στον Η/Υ τους, ώστε να δώσουν στην κατασκευή τους την επιθυμητή συμπεριφορά ως ένα σύνολο οδηγιών. Ο προγραμματισμός μίας ρομποτικής κατασκευής διαφέρει από τον παραδοσιακό προγραμματισμό αφού η ενασχόληση δεν είναι πλέον μία διαδικασία αριθμητικών μόνο υπολογισμών αλλά το ζητούμενο τώρα, είναι η απόδοση στοιχειώδους ευφυΐας σε μία μηχανική κατασκευή.



Πρόκειται για μια εναλλακτική προσέγγιση διδασκαλίας, η οποία στηρίζεται στη χρήση φυσικών μηχανικών μοντέλων, με τα οποία οι μαθητές προσανατολίζονται ως πρώτο βήμα στην κατασκευή ενός ρομποτικού μηχανισμού και μετά από κατάλληλο σχεδιασμό και αρκετούς πειραματισμούς οδηγούνται στην επίλυση πραγματικών προβλημάτων. Στο πλαίσιο των δραστηριοτήτων αυτών οι μαθητές δημιουργούν φυσικά μοντέλα (ένα αυτοκίνητο, ένα σπίτι κα,) τα οποία αφού τα εφοδιάσουν με το κατάλληλο πρόγραμμα, αυτά είναι σε θέση να συλλέγουν πληροφορίες από το περιβάλλον και να αντιδρούν ανάλογα με τα ερεθίσματα που λαμβάνουν. Στο πλαίσιο της εκπαιδευτικής διαδικασίας αυτής οι μαθητές εξασκούνται στον προγραμματισμό για να καθορίσουν τη συμπεριφορά του μοντέλου που έχουν δημιουργήσει οι ίδιοι, όχι στην οθόνη του υπολογιστή τους αλλά στο φυσικό τους περιβάλλον.

Οι Καγκάνης Κ, Δαγδιλέλης Β., Σατρατζέμη Μ. και Ευαγγελίδης Γ. (Μία μελέτη Περίπτωσης της Διδασκαλίας του Προγραμματισμού στη Β/θμια Εκπαίδευση με τα LEGO Mindstorms, 2005) έχουν διαπιστώσει ότι η μέθοδος αυτή λαμβάνει υπόψη τα ενδιαφέροντα των μαθητών και τους εμπλέκει σε δραστηριότητες που έχουν πραγματικά χρησιμότητα και σημασία για αυτούς. Χαρακτηρίζεται από υψηλό βαθμό αλληλεπίδρασης μεταξύ του υπολογιστή και του πραγματικού αντικειμένου, με αποτέλεσμα, ο μαθητής να μπορεί να αναγνωρίσει εύκολα την αντιστοίχιση που υπάρχει ανάμεσα στις



εντολές του προγράμματος και τις αντιδράσεις του μοντέλου και να παρατηρήσει τις συνέπειες που έχουν στη συμπεριφορά του μοντέλου, οι αλλαγές που πραγματοποιεί στο πρόγραμμα. (Eden et al., 1996). Τα φυσικά μοντέλα λοιπόν προσφέρουν άμεση ανατροφοδότηση στους μαθητές ώστε να μπορούν να ελέγχουν την αποτελεσματικότητα των προγραμμάτων τους (Barnes, 2002). Η δυνατότητα αυτή βοηθά τους μαθητές να οδηγούνται σταδιακά σε καλύτερες, αποτελεσματικότερες, πληρέστερες και ακριβέστερες λύσεις.

Συμπερασματικά όπως διαπιστώνεται από τη μελέτη περίπτωσης των Καγκάνη Κ., Δαγδιλέλη Β., Σατρατζέμη Μ. και Ευαγγελίδη Γ. (2005) με τα φυσικά μηχανικά μοντέλα όπως αυτό της Lego: α) πετυχαίνεται υψηλός βαθμός αλληλεπίδρασης μεταξύ υπολογιστή και πραγματικού αντικειμένου, β) υπάρχει άμεση ανατροφοδότηση, γ) υπάρχει πειραματισμός και ενεργός συμμετοχή από τους μαθητές,

Σενάρια Ρομποτικής με το Mindstorms NXT

δ) αναπτύσσεται η κριτική σκέψη, ε) καλλιεργείται η δημιουργική σκέψη, η διορατικότητα και η πρωτοτυπία, στ) υπάρχει άμεση εμπειρία και ο μαθητής απαλλάσσεται από την εκμάθηση και απομνημόνευση συντακτικών κανόνων μιας γλώσσας προγραμματισμού και ζ) υλοποιείται ένα είδος εξατομικευμένης μάθησης αφού ο εκπαιδευτικός διαθέτει περισσότερο χρόνο για κάθε μαθητή και ο κάθε μαθητής εργάζεται με το δικό του ρυθμό μάθησης.



Η χρήση της εκπαιδευτικής ρομποτικής μπορεί να βελτιώσει τις δεξιότητες συνεργασίας, την αυτοπεποίθηση, τη δημιουργικότητα, τα κίνητρα των παιδιών και τις δεξιότητες χειρισμού του υπολογιστή (Palumbo & Palumbo, 1993; Wagner, 1998). Η διδασκαλία της ρομποτικής είναι κατάλληλη για μαθητές ανεξάρτητα από την ηλικία και το υπόβαθρό τους και αποτελεί έναν τρόπο ενθάρρυνσης της μάθησης. Επιπλέον παρέχει στα παιδιά το κατάλληλο περιβάλλον για να διερευνήσουν τα προβλήματα του πραγματικού κόσμου, συμβάλλοντας έτσι στη δημιουργία τεχνολογικά ενημερωμένων πολιτών σύμφωνα με τις ανάγκες της σύγχρονης κοινωνίας.

Έρευνες όσο αφορά στην ψυχαγωγικής διάσταση της εκπαιδευτικής ρομποτικής έχουν δώσει ενθαρρυντικά αποτελέσματα (Chandana et al., 2000; Lund et al., 2002). Οι Chandana, Hafner και Bongard (2000) διαπίστωσαν πως οι μαθητές όχι μόνο μπόρεσαν να κατανοήσουν τις βασικές έννοιες του μαθήματος αλλά κατάφεραν και τις ενσωμάτωσαν στις υπάρχουσες γνώσεις και δεξιότητες τους ως εργαλεία τα οποία ήταν σε θέση να χρησιμοποιήσουν μελλοντικά. Μάλιστα οι ερευνητές αναφέρουν πως ο μόνος αρνητικός παράγοντας στη διδασκαλία των μαθημάτων, σύμφωνα με τα λεγόμενα των μαθητών, είναι ότι «έπρεπε να διαρκούσαν περισσότερο».

Το πακέτο της Lego Mindstorms όπως αναφέρεται από τους Ατματζίδου Σ., Μαρκέλης Η. και Δημητριάδης (Χρήση των LEGO Mindstorms στο Δημοτικό και Λύκειο: Το παιχνίδι ως έναυσμα μάθησης, 2008) με την απλή και εύχρηστη διεπαφή του προγραμματιστικού περιβάλλοντος NXT-G μπορεί να χρησιμοποιηθεί ως ένα μέσο διδασκαλίας μεθόδων επίλυσης προβλημάτων, αποτελώντας μία ευχάριστη και ενδιαφέρουσα ενασχόληση. Η πτυχή του παιχνιδιού, αποτελεί ένα σημαντικό παράγοντα θετικού κινήτρου και παρότρυνσης στην εκπαίδευση (Κόμης, 2005b). Καθώς η πλειοψηφία των μαθητών έχει έρθει κατά καιρούς σε επαφή με τα τουβλάκια της Lego, τα αντιμετωπίζουν περισσότερο ως παιχνίδι και συμμετέχουν πρόθυμα στη διαδικασία της μάθησης χωρίς να βιώνουν την αίσθηση του εξαναγκασμού.



Σενάρια Ρομποτικής με το Mindstorms NXT

Παράλληλα, όπως έχει διατυπωθεί από τον Dewey (1997) η εκπαίδευση πρέπει να στηρίζεται στις φυσικές παρορμήσεις των παιδιών για έρευνα, κατασκευή, επικοινωνία και έκφραση και να εκμεταλλεύεται τη φυσική παρόρμηση τους, που αφορά στην κατασκευή και αλληλεπίδραση με ενεργές αναπαραστάσεις. Όπως αναφέρουν και οι Καρατράντου Α., Παναγιωτακόπουλος Χρ., Πιερρή Ε. (Οι Ρομποτικές Κατασκευές Lego Mindstorms στην κατανόηση Εννοιών Φυσικής στο Δημοτικό Σχολείο Μια Μελέτη Περίπτωσης, 2006) η κατασκευή ενεργών αναπαραστάσεων και η αλληλεπίδραση με αυτές παρουσιάζεται από πολύ μικρή ηλικία κατά τη διάρκεια του παιχνιδιού ενώ σε μεγαλύτερες ηλικίες οι ενεργές αναπαραστάσεις παίρνουν τη μορφή έξυπνων μοντέλων, έξυπνων μηχανών ή προσομοιώσεων. Αυτή η εμπλοκή των παιδιών στην κατασκευή ενεργών αναπαραστάσεων καθώς και η αλληλεπίδραση τους παίζει σημαντικό ρόλο στη μάθηση (Resnick et al. 1996, Ackermann 1997).



Σε αυτό το περιβάλλον εμφανίζεται και μια μορφή γνώσης που ονομάζεται άτυπη μάθηση. Η άτυπη μάθηση σε ένα περιβάλλον εκπαιδευτικής ρομποτικής έχει σχέση με τη γνώση που μοιράζονται οι ίδιοι οι μαθητές που εμπλέκονται σε μία κοινή δραστηριότητα, μεταξύ τους καθώς και από την αλληλεπίδραση τους με τον εκπαιδευτικό. Η άτυπη μάθηση αποτελεί ένα πολύ σημαντικό παράγοντα των γνώσεων και των δεξιοτήτων που αποκτά ένα πρόσωπο. Μπορεί να είναι χαοτική, να είναι δύσκολο να κωδικοποιηθεί και να αποθηκευτεί σε βάσεις δεδομένων και σε εσωτερικά δίκτυα και να μεταδοθεί και να μοιραστεί, αλλά περιέχει εικόνες, σύμβολα και αξίες, διορατικότητα, προαισθήματα, προκαταλήψεις, αισθήματα και είναι πολύτιμη και πλούσια πηγή εμπειρίας και μάθησης. Η άρρητη γνώση λαμβάνει χώρα κατά την αλληλεπίδραση των μαθητών, την ενασχόληση τους με κάποιες δραστηριότητες και την κοινή προσπάθεια που καταβάλλουν για την επίλυση προβλημάτων.



Ενότητα 2η

Ρομποτική

Τι είναι ένα ρομπότ ;

Το **ρομπότ** είναι μια μηχανική συσκευή η οποία είναι σε θέση, να υποκαθιστά τον άνθρωπο σε διάφορες εργασίες. Ένα ρομπότ μπορεί να δράσει κάτω από τον απ' ευθείας έλεγχο ενός ανθρώπου ή αυτόνομα, κάτω από τον έλεγχο ενός προ-προγραμματισμένου υπολογιστή.

Διαθέτει ένα πλήθος αισθητήρων για να μπορεί αντιληφθεί το εξωτερικό του περιβάλλον. Είναι μία συσκευή που είναι κατασκευασμένη να δρα ανεξάρτητα, αλληλεπιδρώντας με το περιβάλλον.

Τα ρομπότ δεν κάνουν του κεφαλιού τους, αλλά εκτελούν μια σειρά οδηγιών βήμα προς βήμα. Ένα ρομπότ δηλαδή μπορεί να εκτελέσει μία εργασία χωρίς τη δική μας παρέμβαση αλλά όμως, αφού έχει πρώτα προγραμματιστεί κατάλληλα από εμάς τους ανθρώπους.

Ένα ρομπότ δε χρειάζεται να έχει κατ' ανάγκη πόδια και γενικά ανθρωπόμορφη εμφάνιση. Μπορεί να είναι αυτοκινούμενο με ρόδες ή και να μη μετακινείται καθόλου.

Η λέξη ρομπότ προέρχεται από το σλαβικό *robota*, που σημαίνει εργασία. Καθιερώθηκε ως όρος με τη σημερινή του έννοια το 1920 από τον Τσέχο θεατρικό συγγραφέα Karel Čapek.

Τα ρομπότ μπορούν να χρησιμοποιηθούν ώστε να κάνουν εργασίες οι οποίες είτε είναι δύσκολες ή επικίνδυνες για να γίνουν από έναν άνθρωπο. (για παράδειγμα στη βαρεία βιομηχανία, στο διάστημα, στο στρατό, στην αστυνομία για την εξουδετέρωση εκρηκτικών μηχανισμών).

Σε άλλες περιπτώσεις, χρησιμοποιούνται για να εκτελέσουν εργασίες ταχύτερα ή φθηνότερα απ' ότι ο άνθρωπος. Έτσι, μπορούν να χρησιμοποιηθούν στην αυτόματη παραγωγή μεγάλων ποσοτήτων κάποιου προϊόντος και με χαμηλότερο κόστος (για παράδειγμα, στις αλυσίδες παραγωγής).

Επίσης μπορούν να χρησιμοποιηθούν για να κάνουμε λεπτούς χειρισμούς που απαιτούν μεγάλη επιδεξιότητα (για παράδειγμα στη χειρουργική ιατρική για επεμβάσεις ακριβείας, στην κατασκευή ηλεκτρονικών πλακετών).

Αλγόριθμος και Ψευδοκώδικας

Προτού προχωρήσουμε στον προγραμματισμό ενός επεξεργαστή πρέπει πρώτα να σκεφτούμε τα βήματα που πρέπει να γίνουν για να οδηγηθούμε στη λύση του προβλήματος μας. Έτσι πρώτα, πρέπει να περιγράψουμε τα βήματα που πρέπει να πούμε στο ρομπότ μας να εκτελέσει, για να επιτύχουμε το επιθυμητό αποτέλεσμα.

Η βήμα προς βήμα λεκτική περιγραφή, του τι πρέπει να κάνει ένας επεξεργαστής για να υλοποιήσουμε την εργασία που επιθυμούμε, ονομάζεται **αλγόριθμος**. Είναι σημαντικό να γράψουμε πρώτα τον αλγόριθμο πριν ξεκινήσουμε να δημιουργούμε ένα πρόγραμμα στον υπολογιστή.

Στην περίπτωση που κάθε βήμα του αλγορίθμου είναι διατυπωμένο έτσι, ώστε να αντιστοιχεί σε μία εντολή που το ρομπότ μας είναι ικανό να εκτελέσει, τότε μια τέτοια λεκτική περιγραφή των βημάτων ονομάζεται **ψευδοκώδικας**.

Τι ονομάζουμε πρόγραμμα

Με τη βοήθεια ενός προγράμματος προμηθεύουμε το ρομπότ μας με κάποια στοιχειώδη ευφυΐα. Ένα πρόγραμμα είναι ένα σύνολο από οδηγίες για το ρομπότ μας, τις οποίες πρέπει αυτό να ακολουθήσει προκειμένου να υλοποιήσει μία συγκεκριμένη εργασία.

Τις οδηγίες αυτές τις γράφουμε σε κάποια ειδική γλώσσα (γλώσσα προγραμματισμού) την οποία το ρομπότ μας είναι σε θέση να καταλαβαίνει.

Η γλώσσα προγραμματισμού NXT-G

Ο μικροϋπολογιστής NXT του πακέτου Lego Mindstorms NXT έρχεται από το εργοστάσιο ρυθμισμένος να καταλαβαίνει τη γλώσσα προγραμματισμού NXT-G. Μέχρι τώρα υπάρχουν δύο διαφορετικά πακέτα του προγραμματιστικού εργαλείου NXT-G : Η πρώτη συνοδεύει το πακέτο ρομποτικής NXT της Lego που πωλείται στα καταστήματα και είναι γνωστή ως εμπορική έκδοση (retail version) και η δεύτερη έρχεται με το εκπαιδευτικό πακέτο ρομποτικής (education version). Στα πακέτα αυτά δεν υπάρχει ουσιαστική διαφορά παρά μόνο κάποια διαφοροποίηση αφού η εκπαιδευτική έκδοση απευθύνεται στα σχολεία και στους εκπαιδευτικούς, αλλά βέβαια μπορεί να την προμηθευτεί και οποιοσδήποτε άλλος θέλει.

Η γλώσσα προγραμματισμού Robolab

Για να γράψουμε ένα πρόγραμμα για το μικροϋπολογιστή NXT μπορούμε να χρησιμοποιήσουμε και τη γλώσσα προγραμματισμού Robolab στην έκδοση τουλάχιστον 2.9. Η έκδοση αυτή υποστηρίζει και τους μικροϋπολογιστές NXT και Scout από τις προηγούμενες σειρές τις Lego Mindstorms. Οι εκδόσεις του Robolab 2.5 και κάτω υποστηρίζουν μόνο τη σειρά των μικροεπεξεργαστών NXT. Για να χρησιμοποιήσουμε το Robolab πρέπει να αλλάξουμε το firmware στο μικροϋπολογιστή NXT.

Μια παρουσίαση για τα ρομπότ θα βρείτε στο σύνδεσμο :

[Ρομπότ](#)

Μια παρουσίαση για τη Ρομποτική και το προγραμματιστικό περιβάλλον NXT-G θα βρείτε στους συνδέσμους : [Ρομποτική και NXT-G \(απλό\)](#) ή [Ρομποτική και NXT-G \(σε μορφή ερωτήσεων\)](#)



Ενότητα 3η

Τεμάχια πακέτου Mindstorms NXT

Πακέτο Lego Mindstorms NXT

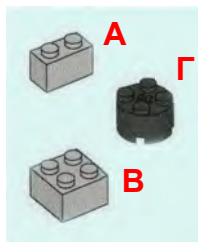
Το πακέτο Mindstorms NXT της Lego μπορεί να μοιάζει ως ένα ενδιαφέρον παιχνίδι αλλά στην πραγματικότητα είναι ένα ολοκληρωμένο περιβάλλον ρομποτικής. Πρόκειται για ένα εξαιρετικό εργαλείο ρομποτικής. Μπορεί να χρησιμοποιηθεί από σχολεία, πανεπιστήμια αλλά και χομπίστες. Με αυτό μπορείς να κατασκευάσεις από απλά ρομπότ μέχρι ότι πιο πολύπλοκο με μόνο περιορισμό τη φαντασία του καθενός.

Το πακέτο Lego Mindstorm Robotics δίνει την ευκαιρία στους χρήστες του, να έχουν μια βασιζόμενη σε χειρονακτική εργασία ανοικτού τύπου εμπειρία και μία βασιζόμενη στην επίλυση πραγματικών προβλημάτων ρεαλιστική εξερεύνηση, συνδυάζοντας και τη διασκέδαση.

Τεμάχια του πακέτου Lego Mindstorms

Το πακέτο Mindstorms NXT της Lego αποτελείται από τα παρακάτω είδη τεμαχίων :

1. Τουβλάκια (Bricks)



- A) 1x2 τουβλάκι
- B) 2x2 τουβλάκι
- Γ) 2x2 στρογγυλό τουβλάκι

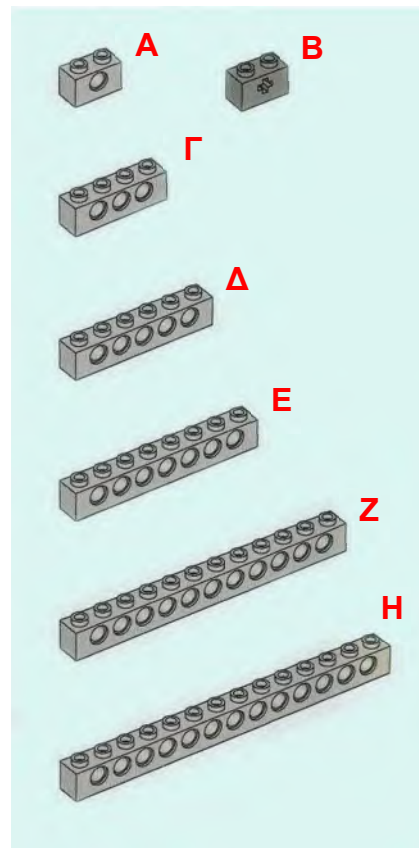
Τα τουβλάκια της Lego είναι γνωστά σε όλους μας. Τα συναντάμε σε διάφορα χρώματα και διαστάσεις. Για να τα ξεχωρίζουμε μεταξύ τους, τα ονομάζουμε ανάλογα με τον αριθμό των εξογκωμάτων που περιέχουν κατά τις δύο διαστάσεις τους.

Την απόσταση ανάμεσα σε δύο εξογκώματα την ονομάζουμε "μήκος lego".

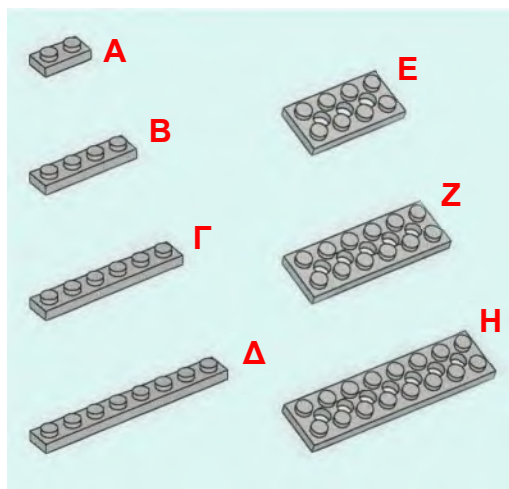
Εκτός από τα κλασικά τουβλάκια υπάρχουν και μερικοί άλλοι τύποι, όπως κυκλικά τουβλάκια και σφήνες.

Επίσης υπάρχουν και τα τουβλάκια δοκοί :

- A) 1x2 τουβλάκι δοκός
- B) 1x2 τουβλάκι με σπή +
- Γ) 1x4 τουβλάκι δοκός
- Δ) 1x6 τουβλάκι δοκός
- E) 1x8 τουβλάκι δοκός
- Z) 1x12 τουβλάκι δοκός
- H) 1x14 τουβλάκι δοκός



2. Πλακίδια (Plates)

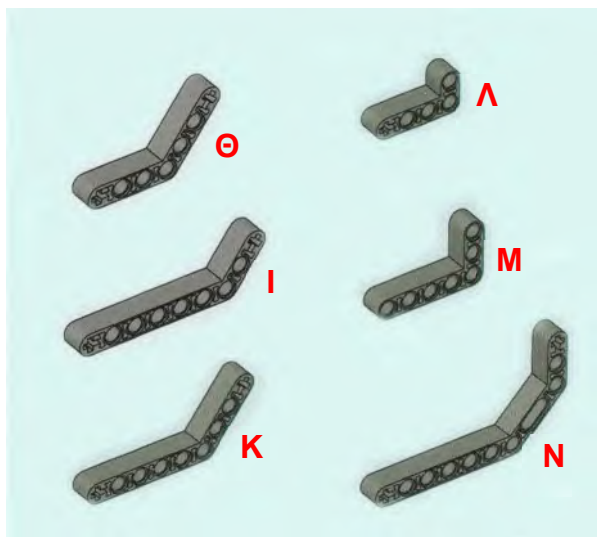
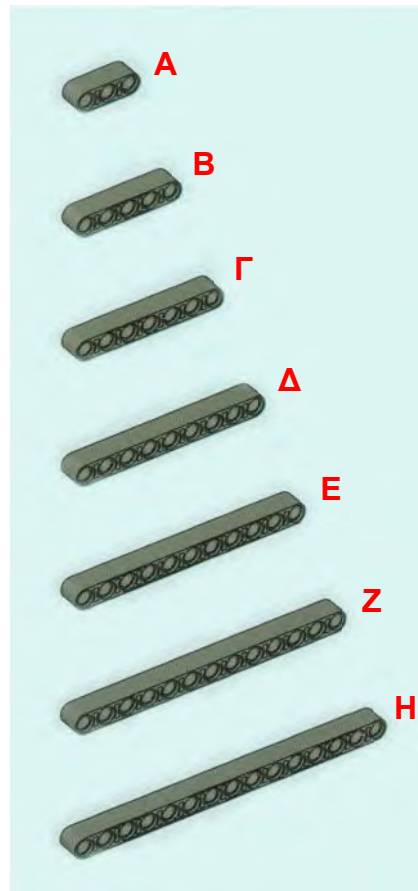


- A) 1x2 πλακίδιο
- B) 1x4 πλακίδιο
- Γ) 1x6 πλακίδιο
- Δ) 1x8 πλακίδιο
- E) 2x4 πλακίδιο με σπές
- Z) 2x6 πλακίδιο με σπές
- H) 2x8 πλακίδιο με σπές

Τα πλακίδια διαφέρουν από τα τουβλάκια στο πάχος. Το πάχος από ένα τουβλάκι είναι ίσο με το πάχος 3 πλακιδίων. Εκτός από τα κλασικά πλακίδια, υπάρχουν και πλακίδια με οπές, καθώς και ειδικά πλακίδια.

3. Δοκοί (Beams)

- A) δοκός 3 οπών
- B) δοκός 5 οπών
- Γ) δοκός 7 οπών
- Δ) δοκός 9 οπών
- Ε) δοκός 11 οπών
- Ζ) δοκός 13 οπών
- Η) δοκός 15 οπών



- Θ) γωνιακή δοκός 135° 4+3 οπών
- Ι) γωνιακή δοκός 135° 7+2 οπών
- Κ) γωνιακή δοκός 135° 6+3 οπών
- Λ) γωνιακή δοκός 90° 4+1 οπών
- Μ) γωνιακή δοκός 90° 5+2 οπών
- Ν) γωνιακή δοκός με 2 γωνίες

Οι δοκοί μαζί με τα τουβλάκια και τα πλακίδια αποτελούν τα δομικά στοιχεία μιας κατασκευής. Οι δοκοί διαθέτουν οπές ώστε να μπορούμε να στηρίξουμε μια σειρά από άξονες. Υπάρχουν σε μία μεγάλη ποικιλία μηκών, καθώς και ειδικοί δοκοί.

4. Άξονες (Axles)



- #2 κόκκινος άξονας
- Β) #3 άξονας
- Γ) #4 άξονας
- Δ) #5 άξονας
- Ε) #6 άξονας
- Ζ) #7 άξονας
- Η) #8 άξονας
- Θ) #10 άξονας
- Ι) #12 άξονας
- Κ) #2 άξονας με καπακωτό άκρο
- Λ) #3 άξονας με καπακωτό άκρο

Οι άξονες χρησιμοποιούνται για να τοποθετήσουμε γρανάζια και ρόδες. Το μήκος τους το μετράμε σε μήκη lego. Υπάρχει μια μεγάλη ποικιλία μηκών, από δύο μονάδες lego ως και 12 μονάδες.

Είναι σημαντικό να στηρίζουμε τους άξονες σε περισσότερα από ένα σημεία. Αυτό βοηθάει τον άξονα να γυρίζει με μεγαλύτερη ελευθερία και κάνει την κατασκευή πιο στιβαρή.

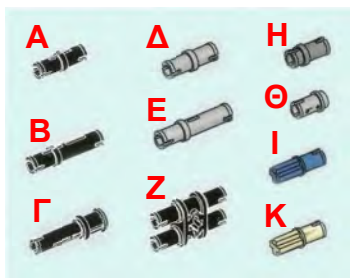
5. Δακτύλιοι (Bushings)



- Α) πλήρης δακτύλιος
- Β) μισός δακτύλιος

Οι δακτύλιοι τοποθετούνται στο τέλος των αξόνων, για να τους συγκρατούν στη θέση τους.

6. Πιράκια (Pegs)

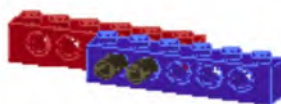


- Α) μαύρο πιράκι
- Β) μαύρο επεκταμένο πιράκι
- Γ) μαύρο πιράκι με κεφαλή για άξονα
- Δ) γκρι πιράκι
- Ε) γκρι επεκταμένο πιράκι
- Ζ) διπλό πιράκι
- Η) κοντό πιράκι
- Θ) πιράκι με καπακωτό άκρο
- Ι) μπλε πιράκι με άκρο άξονα
- Κ) λευκό πιράκι με άκρο άξονα

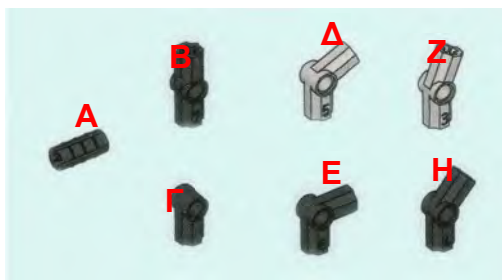
Τα πιράκια τοποθετούνται στις οπές των δοκών και συνδέουν δύο δοκούς μεταξύ τους.

Αν συνδέσουμε δύο δοκούς με ένα γκρι πιράκι, αυτοί μπορούν να περιστραφούν ελεύθερα. Έτσι μπορούμε να κατασκευάσουμε μια άρθρωση.

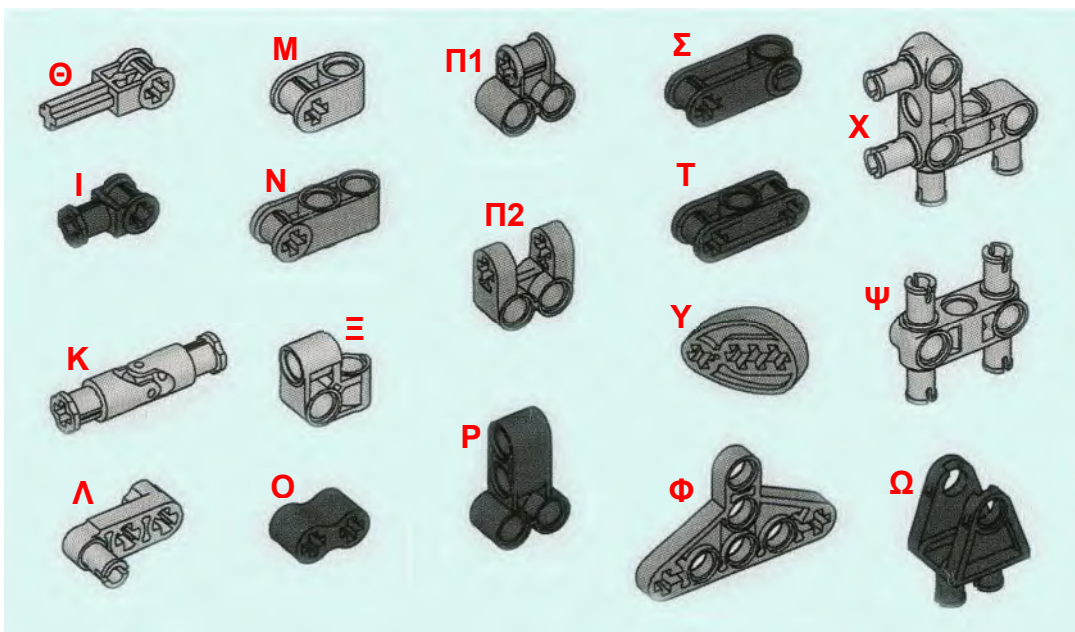
Αντίθετα το μαύρο πιράκι δεν επιτρέπει σε δύο δοκούς να περιστραφούν εύκολα. Το χρησιμοποιούμε όταν θέλουμε να συνδέσουμε τους δοκούς σταθερά μεταξύ τους. Το μαύρο πιράκι υπάρχει και σε διπλό μήκος από τη μία πλευρά του.



7. Σύνδεσμοι (Connectors)



- Α1) μούφα
- Α2) κύλινδρος
- Β) ίσιος σύνδεσμος
- Γ) μισός σύνδεσμος
- Δ) σύνδεσμος 120°
- Ε) σύνδεσμος 90°
- Ζ) σύνδεσμος 150°
- Η) σύνδεσμος 135°



- Θ) του άξονα αρσενικό – θηλυκό
- Κ) άρθρωση αξόνων
- Μ) του άξονα με πιράκι
- Ξ) γκρι του για 3 πιράκια
- Π1) του άξονα με 2 πιράκια σχήματος 'Τ'
- Π2) του άξονα με 2 πιράκια σχήματος 'Π'
- Ρ) του με 4 πιράκια σχήματος 'Τ'
- Τ) του 2 αξόνων με πιράκι
- Φ) επίπεδος βραχίονας 'Τ'
- Ψ) δοκός με πιράκια
- Ω) κοτσαδόρος
- Ι) του άξονα αρσενικό – αρσενικό
- Λ) μανιβέλα
- Ν) του άξονα με 2 πιράκια στη σειρά
- Ο) ελαστικός σύνδεσμος
- Σ) μακρύ του άξονα με πιράκι
- Υ) έκκεντρο
- Χ) γωνιακή δοκός με πιράκια

Οι σύνδεσμοι μας δίνουν τη δυνατότητα να συνδέσουμε τους άξονες και τα πιράκια. Χρησιμοποιώντας τους συνδέσμους, σε συνδυασμό με τους άξονες και τα πιράκια, μπορούμε να κατασκευάσουμε μια ποικιλία από αρθρώσεις και συνδέσεις.

8. Ρόδες (Wheels)

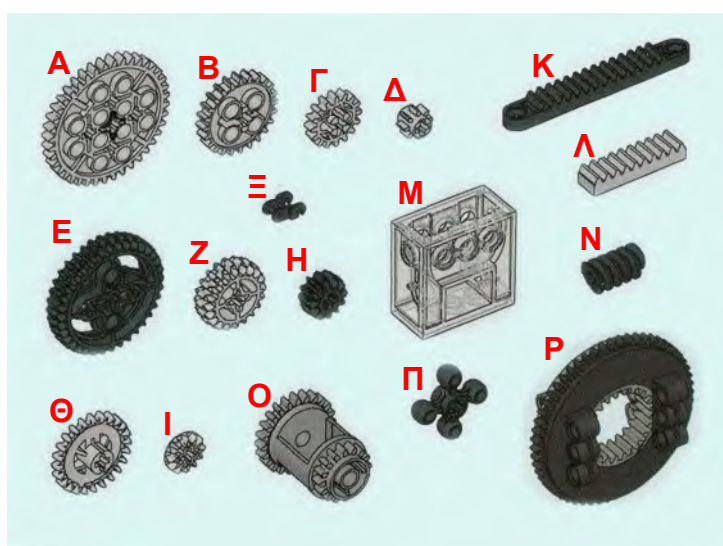


- Α) μεγάλη άσπρη ζάντα και
ρόδα
Β) μεσαία άσπρη ζάντα και
ρόδες
Γ) μικρή άσπρη ζάντα και
ρόδα
Δ) ζάντα τροχαλία και ρόδα
Ε) ερπύστρια

Υπάρχει μια μεγάλη ποικιλία στο μέγεθος των ροδών.

Παρέχεται επίσης η δυνατότητα χρήσης ερπυστριών. Με τις ερπύστριες ένα ρομπότ παρόλο που δεν μπορεί να επιταχύνει εύκολα, μπορεί να κινηθεί πιο σταθερά ακόμη και σε ανώμαλο έδαφος. Επίσης σε αντίθεση με ένα ρομπότ που διαθέτει τέσσερις ρόδες με δυνατότητα των μπροστινών ροδών να στρίβουν, το ρομπότ με ερπύστριες μπορεί να εκτελεί τις στροφές πιο αποτελεσματικά (ακόμη και επιτόπου).

9. Γρανάζια (Gears)



- A) 40 δοντιών γρανάξι
- B) 24 δοντιών γρανάξι
- Γ) 16 δοντιών γρανάξι
- Δ) 8 δοντιών γρανάξι
- E) 36 δοντιών χοντρό μαύρο γρανάξι
- Z) 20 δοντιών χοντρό γκρι γρανάξι
- H) 12 δοντιών χοντρό μαύρο γρανάξι
- Θ) 24 λοξών δοντιών γρανάξι
- I) 12 λοξών δοντιών γρανάξι
- K) μαύρος δοκός με 14 δόντια
- Λ) γκρι δοκός με 10 δόντια
- M) κιβώτιο κοχλιωτού γραναζιού
- N) κοχλιωτό γρανάξι
- Ξ) κρίκος καδένας
- O) διαφορικό
- Π) γρανάξι με 4 πτερύγια
- P) περιστροφικός δακτύλιος

Σενάρια Ρομποτικής με το Mindstorms NXT

Με τα γρανάζια μπορούμε να :

- μεταφέρουμε την κίνηση από ένα σημείο σε άλλο.
- αλλάξουμε την κατεύθυνση κίνησης ενός άξονα.
- να αλλάξουμε την ταχύτητα κίνησης και τη ροπή σε έναν άξονα. (πχ να μειώσουμε την ταχύτητα κίνησης στο ρομπότ μας και ταυτόχρονα να του δώσουμε περισσότερη ισχύ).

Υπάρχει μια πολύ μεγάλη ποικιλία γραναζιών με τα οποία μπορούμε να υλοποιήσουμε διάφορες σύνθετες κατασκευές.

10. Ιμάντες και Τροχαλίες (Belts and Pulleys)



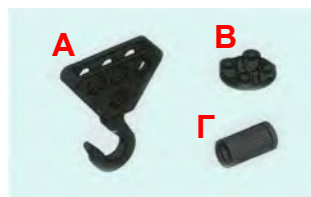
- A) μεσαία τροχαλία
B) μισός δακτύλιος
C) κόκκινος ιμάντας
D) κίτρινος ιμάντας

Μπορούμε επίσης να μεταφέρουμε την κίνηση χρησιμοποιώντας ιμάντες με τροχαλίες στη θέση των γραναζιών, με τη διαφορά ότι όταν σημειωθεί κάποιο μπλοκάρισμα, οι ιμάντες σε αντίθεση με τα γρανάζια, επιτρέπουν την ολίσθηση ενός άξονα σε σχέση με έναν άλλο. Στα γρανάζια το μπλοκάρισμα ενός άξονα, μπλοκάρει και τους συνδεδεμένους σε αυτόν, άξονες.

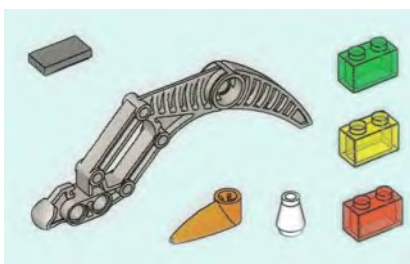
11. Διάφορα τεμάχια



Βραχίονες με αρθρώσεις



- A) άγκιστρο
B) ολισθητήρας
C) κυλινδράκι

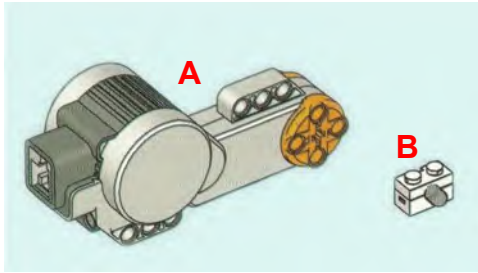


Διακοσμητικά τεμάχια



Κανόνι με βέλος

12. Κινητήρες και Λαμπτήρες (Motors and Lamps)

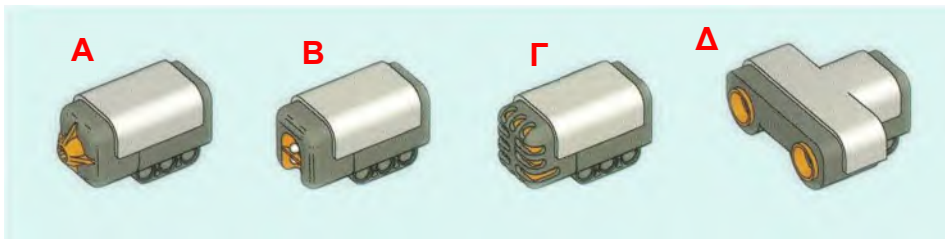


A) κινητήρας
B) λαμπτήρας

Οι κινητήρες δίνουν κίνηση στο ρομπότ μας.

13. Αισθητήρες (Sensors)

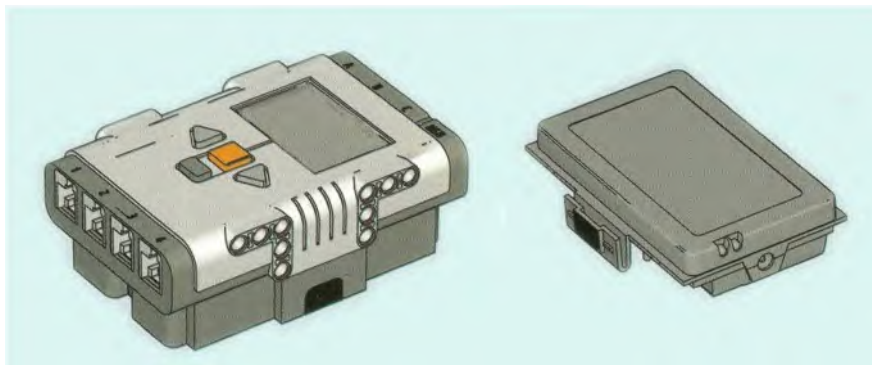
Οι αισθητήρες παρέχουν στο ρομπότ μας πληροφορίες για τον κόσμο που τους περιβάλλει.



A) αισθητήρας αφής
B) αισθητήρας φωτός
Γ) αισθητήρας ήχου
Δ) αισθητήρας υπερήχων

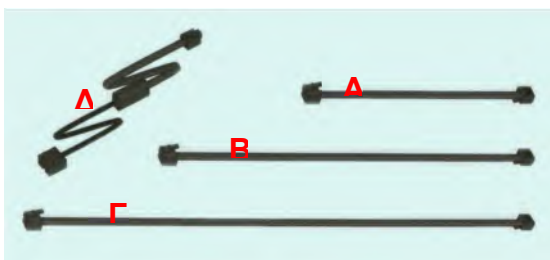
14. Τούβλο ή μικροεπεξεργαστής NXT (NXT Brick)

Το τούβλο NXT αποτελεί το μυαλό του ρομπότ μας. Αποθηκεύει τα προγράμματα τα οποία διαβάζουν τα δεδομένα από τους αισθητήρες και δίνουν την κατάλληλη κίνηση στους κινητήρες.



A) τούβλο NXT
B) επαναφορτιζόμενη μπαταρία

15. Καλώδια



A) καλώδιο 20 cm
B) καλώδιο 35 cm
Γ) καλώδιο 45 cm
Δ) καλώδιο προσαρμογής

Μια παρουσίαση για τα τεμάχια του Πακέτο NXT θα βρείτε στο σύνδεσμο :

[το Πακέτο NXT](#)

Παρακάτω θα αναφερθούμε πιο αναλυτικά στα γρανάζια, τις τροχαλίες και τους ιμάντες, τους κινητήρες και τους αισθητήρες.

Γρανάζια

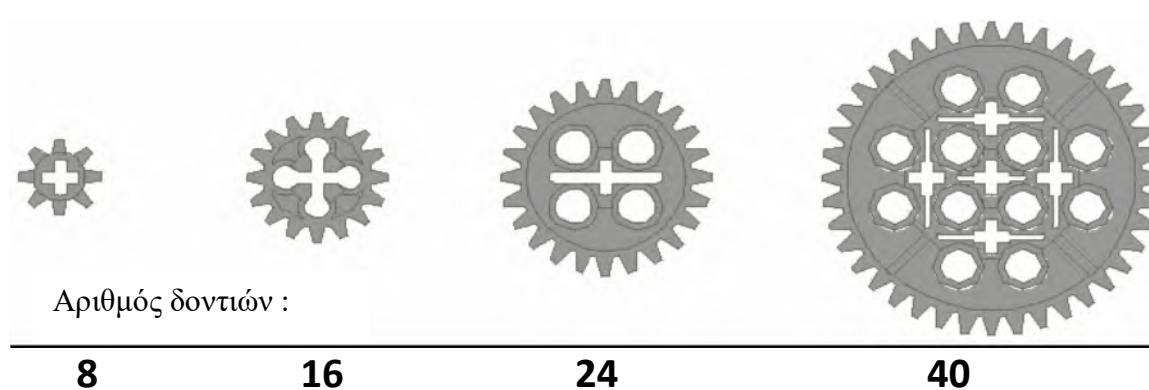
Τα γρανάζια όπως είδαμε χρησιμοποιούνται για να αλλάξουν την κατεύθυνση κίνησης, την ταχύτητα περιστροφής και τη ροπή ενός κινητήρα.

Μπορούμε να χρησιμοποιήσουμε γρανάζια για να :

- Αυξήσουμε ή να μειώσουμε την ταχύτητα κίνησης ενός ρομπότ.
- Αυξήσουμε ή να μειώσουμε την ισχύ στην κίνηση ενός ρομπότ.

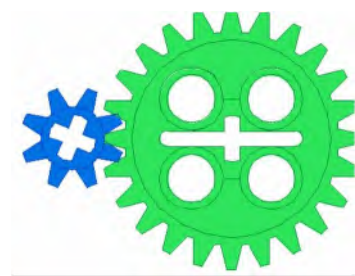
Υπάρχουν διάφοροι τύποι γραναζιών, με τους οποίους μπορούμε να πετύχουμε διάφορες υλοποιήσεις:

1. Γρανάζια με δόντια (Spur gears).



Πρόκειται για τα κλασικά γρανάζια.

A) Βάζοντας δύο γρανάζια στη σειρά, αντιστρέφουμε τη φορά περιστροφής και μετατρέπουμε την ταχύτητα αντιστρόφως ανάλογα με το λόγο των γραναζιών και τη ροπή ανάλογα με το λόγο των γραναζιών.



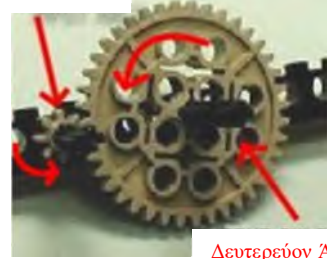
Λόγος γραναζιών (gear ratio):

Για να υπολογίσουμε τη μετατροπή στην ταχύτητα περιστροφής δύο συνδεδεμένων μεταξύ τους γραναζιών - και αντίστοιχα στη δύναμη περιστροφής που ασκείται στους άξονες τους, δηλαδή στη ροπή τους - αρκεί να μετρήσουμε τον αριθμό των γραναζιών που διαθέτει κάθε ένα από τα γρανάζια αυτά, ως εξής :

Σενάρια Ρομποτικής με το Mindstorms NXT

Ας πάρουμε για παράδειγμα ένα γρανάζι με 8 δόντια το οποίο είναι συνδεδεμένο στον πρωτεύοντα άξονα, ο οποίος κινείται από έναν κινητήρα και ένα γρανάζι με 40 δόντια το οποίο είναι συνδεδεμένο στο δευτερεύοντα άξονα, στον οποίο έχουμε τοποθετήσει τις ρόδες μας. Ως **Λόγο Γραναζιών** ονομάζουμε το πηλίκο του αριθμού των δοντιών του γραναζιού στο δευτερεύοντα άξονα προς τον αριθμό των δοντιών του γραναζιού στον πρωτεύοντα άξονα. Στην περίπτωση μας ο λόγος των γραναζιών είναι 40:8 ή 5:1.

Πρωτεύον Άξονας



Δευτερεύον Άξονας

Πρακτικά μπορούμε να πούμε ότι το γρανάζι στο δευτερεύοντα άξονα θα κάνει 1 πλήρη περιστροφή όταν το γρανάζι στον πρωτεύοντα άξονα συμπληρώσει 5 περιστροφές. Άρα έχουμε μείωση της ταχύτητας περιστροφής ανάλογη με το λόγο των γραναζιών. Επίσης προκύπτει ότι έχουμε αύξηση της ροπής (της δύναμης δηλαδή που ασκείται για να περιστρέψει τους άξονες μας) ανάλογη με το λόγο γραναζιών. Ο δευτερεύοντας άξονας μας είναι λοιπόν 5 φορές πιο αργός από τον πρωτεύοντα αλλά και 5 φορές πιο ισχυρός.

Συμβουλή : Δεν είναι βολικό να συνδέουμε μία ρόδα κατευθείαν στον άξονα του κινητήρα επειδή έτσι θα έχουμε μεγάλη ταχύτητα περιστροφής με μικρή όμως ροπή. Συνηθίζεται να μεταδίδουμε την κίνηση από τον κινητήρα με συζευγμένα γρανάζια, τα οποία να μειώνουν την ταχύτητα περιστροφής.

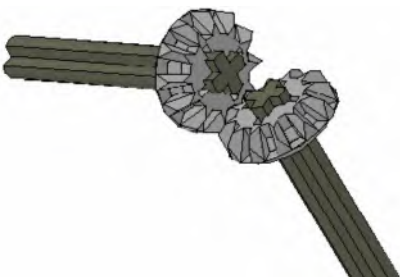
Β) Βάζοντας ένα γρανάζι ανάμεσα σε δύο άλλα πετυχαίνουμε να κρατήσουμε την ίδια φορά περιστροφής ανάμεσα στο πρώτο γρανάζι και στο τρίτο. Η ταχύτητα περιστροφής και η ροπή σε αυτήν την περίπτωση, υπολογίζονται από το λόγο των γραναζιών ανάμεσα στο πρώτο και στο τρίτο γρανάζι. Το μεσαίο γρανάζι δηλαδή δεν παίζει κανέναν ρόλο στην αλλαγή της ταχύτητας και της ροπής, αλλά επηρεάζει μόνο τη φορά περιστροφής.

2. Γρανάζια με λοξά δόντια (Bevel gears).

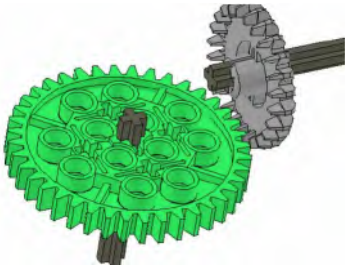


Με τα γρανάζια αυτά μπορούμε να αλλάξουμε την κατεύθυνση της περιστροφής κατά 90 °. Υπάρχουν δύο τύποι γραναζιών με λοξά δόντια :

Α) 12 δοντιών, τα οποία μπορούν να συνδεθούν μόνο μεταξύ τους.

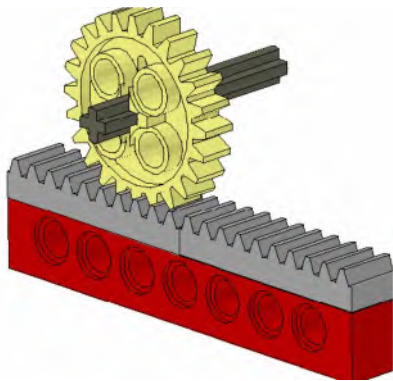


B) 24 δοντιών τα οποία ονομάζονται και γρανάζια κορώνες, και στα οποία μπορεί να συνδεθεί και ένα απλό γρανάζι.



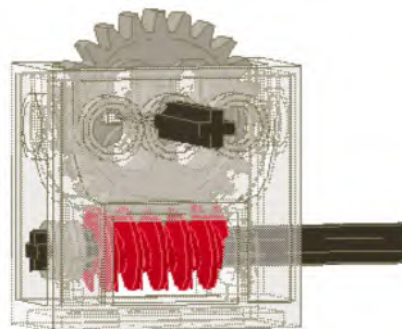
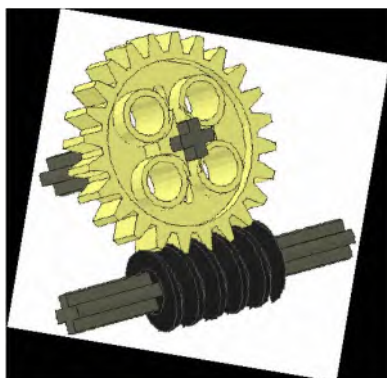
3. Δοκός με δόντια (Rack and Pinion).

Σε συνδυασμό με ένα κλασικό γρανάζι (το οποίο για την περίπτωση αυτή το ονομάζουμε πηνίο (pinion)) μπορούμε να αλλάξουμε την κατεύθυνση της κίνησης από περιστροφική σε ευθύγραμμη και αντίστροφα.



4. Κοχλιωτό γρανάζι (Worm gears).

Μοιάζει με τις βίδες. Σε συνδυασμό με ένα κλασικό γρανάζι μπορούμε να αλλάξουμε την κατεύθυνση της περιστροφής κατά 90° , (όπως και με τα γρανάζια με λοξά δόντια) αλλά με μία ιδιαιτερότητα. Το κοχλιωτό γρανάζι μπορεί να γυρίσει το κλασικό γρανάζι αλλά το αντίθετο είναι αδύνατο. Έτσι το κοχλιωτό γρανάζι είναι πάντα αυτό που δίνει την κίνηση.



5. Γρανάζι με ολίσθηση (Slip clutch)

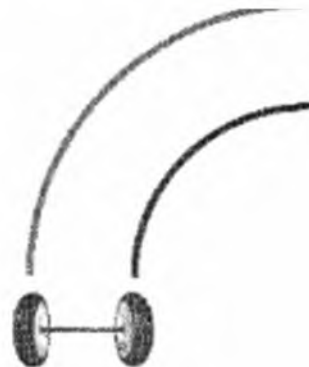
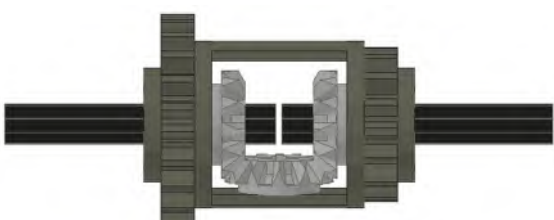
Πρόκειται για έναν ειδικό τύπο του κλασικού γραναζιού. Διαθέτει μία ιδιαιτερότητα : σε αντίθεση με ένα κλασικό γρανάζι, επιτρέπει την ολίσθηση του άξονα στον οποίο είναι συνδεδεμένο, όταν τα δόντια του μπλοκάρουν. Συμπεριφέρεται δηλαδή παρόμοια με μία τροχαλία με ιμάντα.



Απλοί μηχανισμοί με γρανάζια :

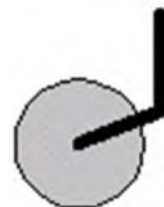
1. Διαφορικό (differential)

Τοποθετώντας ένα διαφορικό στο οποίο συνδέονται δύο άξονες, αντί να συνδέσουμε τους τροχούς μας πάνω στον ίδιο άξονα, πετυχαίνουμε μία σχετική ανεξαρτησία στην κίνηση των δύο τροχών, με αποτέλεσμα το όχημα – ρομπότ μας να στρίβει πιο ευέλικτα. Με το διαφορικό οι δύο άξονες περιστρέφονται με διαφορετική μεταξύ τους ταχύτητα, με τον περιορισμό ότι ο μέσος όρος των ταχυτήτων αυτών παραμένει πάντα σταθερός.

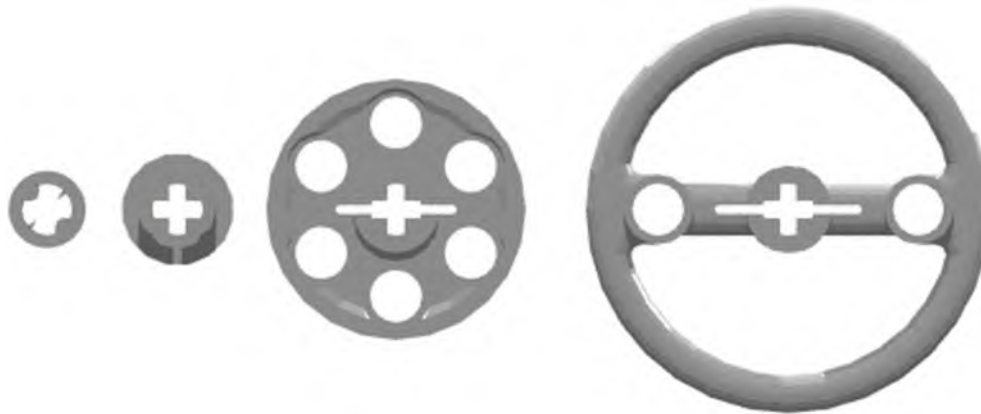


2. Σύστημα τροχού κάστερ (caster wheel)

Το σύστημα τροχού κάστερ, επιτρέπει το όχημα – ρομπότ μας να στρίβει χωρίς την παραμικρή δυσκολία αφού οι ρόδες ακολουθούν πάντα την κατεύθυνση του οχήματος μας.



Τροχαλίες και ιμάντες



Περίμετρος σε cm :

2

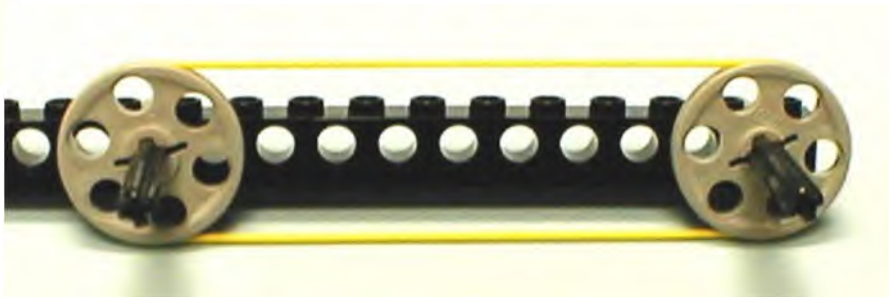
3

7

11

Όπως τα γρανάζια, με τον ίδιο τρόπο και οι ιμάντες με τροχαλίες χρησιμοποιούνται για να μεταφέρουν την κίνηση από ένα άξονα σε ένα άλλο. Υπάρχουν όμως σημαντικές διαφορές :

α) Στους ιμάντες με τροχαλίες διατηρείται η ίδια φορά περιστροφής.



β) Οι ιμάντες δε μεταφέρουν πάντα όλη την ισχύ από τον ένα άξονα στον άλλο. Στην περίπτωση που ο δεύτερος άξονας ζοριστεί, ο ιμάντας αρχίζει να ολισθαίνει.



Και στην περίπτωση των τροχαλιών, με τον ίδιο τρόπο όπως και με τα γρανάζια, μπορούμε να υπολογίσουμε το λόγο μετατροπής της περιστροφικής ταχύτητας και της ροπής. Στην περίπτωση αυτή, αντί για τον αριθμό των δοντιών χρησιμοποιούμε την περίμετρο των τροχαλιών.

Μια παρουσίαση για τα Γρανάζια και τις Τροχαλίες του Πακέτο NXT θα βρείτε στο σύνδεσμο :

[Πακέτο NXT – Γρανάζια, Τροχαλίες](#)

Κινητήρες



Οι τρεις κινητήρες που μπορούν να συνδεθούν στο ρομπότ μας, του δίνουν τη δυνατότητα να κινείται. Αν μάλιστα σε μία ρομποτική κατασκευή η οποία κινείται με τη βοήθεια δύο κινητήρων χρησιμοποιήσουμε στο πρόγραμμα που θα δημιουργήσουμε, την εντολή Move, οι δύο κινητήρες θα συγχρονιστούν αυτόματα και το ρομπότ μας θα κινείται απολύτως σε ευθεία γραμμή.



Αισθητήρες

Οι αισθητήρες παίζουν καθοριστικό ρόλο σε ένα ρομπότ. Παρέχουν πληροφορίες στον επεξεργαστή του ρομπότ για το περιβάλλον, μέσα στο οποίο δραστηριοποιείται αυτό. Στη συνέχεια ο επεξεργαστής του ρομπότ αποφασίζει τι θα κάνει, βασιζόμενος σε αυτές τις πληροφορίες και με βάση τις εντολές του προγράμματος που του έχουν δοθεί. Στο πακέτο Lego Mindstorms συναντάμε 5 τύπους αισθητήρων :

1. Αισθητήρας αφής

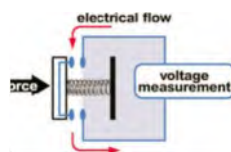


Ο αισθητήρας αφής δίνει τη δυνατότητα στο ρομπότ μας να αποκτήσει την αίσθηση της αφής. Με τον αισθητήρα αφής το ρομπότ μας είναι σε θέση να αντιληφθεί πότε έχει έρθει σε επαφή ή σταμάτησε να έχει επαφή με κάποιο αντικείμενο από το κοντινό περιβάλλον μέσα στο οποίο δραστηριοποιείται. Ο αισθητήρας αφής ανιχνεύει πότε πιέζεται από κάτι και πότε απελευθερώνεται πάλι. Ο νέος

αισθητήρας αφής διαθέτει πιο μεγάλο πλήκτρο σε σύγκριση με αυτόν της σειράς NXT, οπότε συνήθως δε θα χρειαστεί να κατασκευαστεί κάποια διάταξη προέκτασης. Επιπρόσθετα διαθέτει και μία υποδοχή για άξονα στο κέντρο του πλήκτρου η οποία μας δίνει τη δυνατότητα να κατασκευάσουμε μία προέκταση με ευκολία.

Τόσο ο καινούριος όσο και ο προϋπάρχων αισθητήρας αφής στηρίζουν τη λειτουργία τους με τον ίδιο τρόπο : Με το πάτημα του πλήκτρου κλείνει ένα κύκλωμα και διέρχεται ηλεκτρικό ρεύμα ενώ με την απελευθέρωση του πλήκτρου το κύκλωμα ανοίγει.

Παραδείγματα χρήσης του



Ο αισθητήρας αφής μπορεί να χρησιμοποιηθεί ως χειριστήριο για να δίνουμε εντολές ελέγχου στο ρομπότ μας. Μπορεί δηλαδή, να χρησιμοποιηθεί ως διακόπτης για τον έλεγχο μίας διαδικασίας ώστε αν τον πατήσουμε χειροκίνητα, το ρομπότ μας να εκτελέσει ένα συγκεκριμένο σύνολο εντολών. Για παράδειγμα να κάνουμε το ρομπότ μας να κινηθεί, να μιλήσει κτλ.

Ένα κινούμενο όχημα ρομπότ μπορεί να αντιληφθεί τότε συγκρούεται με κάποιο εμπόδιο και να εκτελεί τις κατάλληλες εντολές ώστε να μπορεί να το παρακάμψει.

Μία ρομποτική κατασκευή εφοδιασμένη με ένα ρομποτικό βραχίονα ο οποίος είναι εξοπλισμένος με έναν αισθητήρα αφής στο εσωτερικό του, μπορεί να αντιληφθεί τότε υπάρχει κάποιο αντικείμενο εντός του βραχίονα ώστε να τον κλείσει για να γραπώσει επιτυχώς το αντικείμενο αυτό.

2. Αισθητήρας φωτός



Ο αισθητήρας φωτός είναι ένας από τους δύο αισθητήρες που δίνουν τη δυνατότητα στο ρομπότ μας να αποκτήσει μερικώς την αίσθηση της όρασης (Ο άλλος είναι ο αισθητήρας υπερήχων). Με τον αισθητήρα φωτός το ρομπότ μας είναι σε θέση να ξεχωρίσει τις καταστάσεις ανάμεσα στο φως και το σκοτάδι, στο κοντινό περιβάλλον μέσα στο οποίο δραστηριοποιείται. Ο νέος αισθητήρας φωτός είναι μία βελτιωμένη

έκδοση από αυτόν της σειράς NXT : διαθέτει μεγαλύτερη ευαισθησία και επιτρέπει πιο ακριβείς μετρήσεις στην κλίμακα του 0 (καθόλου φως) ως 100 (πολύ έντονο φως). Μπορεί επιπρόσθετα να λειτουργήσει με δύο τρόπους : μπορούμε να επιλέξουμε αν θα ενεργοποιηθεί ή όχι η λυχνία υπέρυθρης ακτινοβολίας που διαθέτει αυτός. Έτσι όταν η παραπάνω λυχνία είναι κλειστή το ρομπότ μας μπορεί να αντιληφθεί τη ένταση του φωτός σε ένα δωμάτιο ενώ στην περίπτωση που είναι ενεργοποιημένη μπορεί να μετρήσει τη ένταση του φωτός που αντανακλάται από μία χρωματιστή επιφάνεια.

Τη ένταση του φωτός μας την εμφανίζει για λόγους ευκολίας σε ποσοστό μιας ελάχιστης και μιας μέγιστης στάθμης οι οποίες μπορούν να καθοριστούν και από εμάς. Όσο μικρότερο είναι αυτό το ποσοστό, τόσο μικρότερη είναι και η φωτεινότητα του μετρούμενου φωτός.



Έτσι αντιλαμβάνεται τα παραπάνω χρώματα ένα ρομπότ που χρησιμοποιεί τον αισθητήρα φωτός.

Παραδείγματα χρήσης του

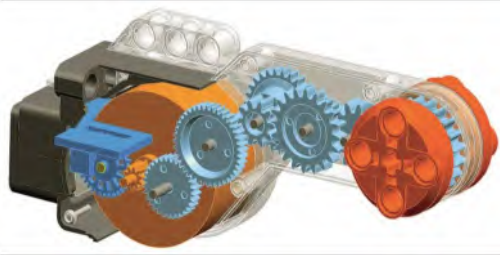


Με έναν αισθητήρα φωτός μπορούμε να κατασκευάσουμε ένα σύστημα συναγερμού. Όταν κάποιος ανάψει το φως στο δωμάτιο το ρομπότ μας να μπορεί να ενεργεί κατάλληλα για να αποτρέψει το διαρρήκτη.

Ένα κινούμενο όχημα ρομπότ μπορεί να ακολουθεί μία γραμμή χαραγμένη στο πάτωμα με διαφορετικό χρώμα από αυτό.

Μπορούμε επίσης να χρησιμοποιήσουμε τον αισθητήρα φωτός για να βάλουμε ένα ρομπότ να ταξινομήσει διάφορα αντικείμενα ως προς το χρώμα τους.

3. Αισθητήρας περιστροφής (Ενσωματωμένος στον κινητήρα)



Σε αντίθεση με την προηγούμενη σειρά NXT, η σειρά NXT της Lego δεν περιλαμβάνει ξεχωριστούς αισθητήρες περιστροφής. Στη σειρά NXT, οι νέοι κινητήρες με τους οποίους έρχεται εφοδιασμένη, περιλαμβάνουν έναν ενσωματωμένο αισθητήρα περιστροφής, οποίος μπορεί να πληροφορήσει το μικροεπεξεργαστή NXT ανά πάσα στιγμή, για τη θέση στη οποία έχει περιστραφεί ένας κινητήρας. Πρόκειται δηλαδή για υλοποιήσεις σερβομηχανισμού. Από τις θύρες αυτές A, B και C λοιπόν, σε αντίθεση με το τούβλο NXT της προηγούμενης γενιάς Lego, μπορούμε με τη χρήση των ενσωματωμένων αισθητήρων περιστροφής να πάρουμε πληροφορίες για την τρέχουσα θέση περιστροφής των κινητήρων.

Οι αισθητήρες αυτοί μπορούν να μετρήσουν την περιστροφή του κινητήρα σε μοίρες (degrees) ή πλήρεις περιστροφές (μία πλήρη περιστροφή αντιστοιχεί σε 360ο) με ακρίβεια ± 1 μοίρα. Αυτός είναι και ο λόγος που τώρα στη νέα σειρά NXT μπορούμε να έχουμε πολλαπλές επιλογές στην ταχύτητα περιστροφής (επίπεδο ισχύος) του κινητήρα.

Σημείωση : Σε περίπτωση που χρειαζόμαστε κάποιον αυτόνομο αισθητήρα περιστροφής, μπορούμε να χρησιμοποιήσουμε αυτούς της σειράς NXT συνδέοντας τους στο τούβλο NXT με το καλώδιο προσαρμογής (converter cable) που περιλαμβάνεται στο πακέτο.

4. Αισθητήρας ήχου



Ο αισθητήρας ήχου δίνει τη δυνατότητα στο ρομπότ μας να αποκτήσει μερικώς την αίσθηση της ακοής. Ο αισθητήρας ήχου ανιχνεύει τη στάθμη ενός ήχου (την πίεση που δημιουργείται στον αέρα από ένα ηχητικό κύμα). Ο αισθητήρας αυτός μπορεί να διακρίνει τη στάθμη ενός ηχητικού κύματος είτε σε ντεσιμπέλ (dB) είτε σε προσαρμοσμένα ντεσιμπέλ (dBA).

Στο σύστημα μέτρησης ήχου dBA η στάθμη του ήχου μετριέται λαμβάνοντας υπόψη και τις συχνότητες στις οποίες παράγεται αυτός και προσαρμόζοντας την σύμφωνα την ευαισθησία του αυτιού μας στις διάφορες συχνότητες αυτές. Πρόκειται για τους ήχους, όπως τους αντιλαμβάνεται το ανθρώπινο αυτί. Από την άλλη στο σύστημα μέτρησης ήχου dBA η στάθμη του ήχου μετριέται αντικειμενικά χωρίς να παίρνουμε υπόψη την ευαισθησία του ανθρώπινου αυτιού. Έτσι στην περίπτωση αυτή το ρομπότ μας μπορεί να ανιληφθεί και ήχους με χαμηλές ή υψηλές συχνότητες που εμείς οι άνθρωποι δε μπορούμε να ακούσουμε.

Ο αισθητήρας ήχου μπορεί να μετρήσει μία ηχητική πίεση μέχρι και της τάξης των 90dB, ως ποσοστό μιας ελάχιστης και μιας μέγιστης στάθμης οι οποίες μπορούν να καθοριστούν και από εμάς. Όσο μικρότερο είναι αυτό το ποσοστό, τόσο μικρότερη είναι και η ένταση του μετρούμενου ήχου.

Ενδεικτικά :

Ο θόρυβος που δημιουργείται σε ένα ήσυχο καθιστικό είναι της τάξης του 4-5 %

Ένας άνθρωπος που μιλά από κάποια απόσταση 5-10%

Η κανονική συνομιλία κοντά στον αισθητήρα ή μουσική σε κανονική στάθμη 10-30%

Άνθρωποι που φωνάζουν ή μουσική σε δυνατή ένταση 30-100%

Παραδείγματα χρήσης του



Ο αισθητήρας ήχου μπορεί να χρησιμοποιηθεί ως διακόπτης για τον έλεγχο μίας διαδικασίας ώστε πχ αν η ένταση του ήχου αυξηθεί πάνω από ένα επίπεδο, το ρομπότ μας να εκτελέσει ένα συγκεκριμένο σύνολο εντολών.

Με τον αισθητήρα ήχου υπάρχει η δυνατότητα το ρομπότ μας να αναγνωρίζει συγκεκριμένα ηχητικά μοτίβα πχ να ξεχωρίζει το ένα κύπημα των χεριών από δύο συνεχόμενα καθώς επίσης και συγκεκριμένους ηχητικούς τόνους πχ μπάσα από πρίμα. Για να το πετύχουμε αυτό θα πρέπει να τοποθετήσουμε μία σειρά από εντολές αναμονής για ήχο.

5. Αισθητήρας Υπερήχων



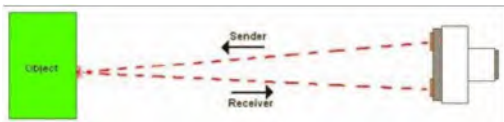
Ο αισθητήρας υπερήχων αποτελεί μία ενδιαφέρουσα προσθήκη στη ρομποτική της lego.

Ο αισθητήρας υπερήχων, όπως και ο αισθητήρας φωτός, δίνει τη δυνατότητα στο ρομπότ μας να αποκτήσει μερικώς την αίσθησης της όρασης. Με τον αισθητήρα υπερήχων το ρομπότ μας είναι σε θέση όχι μόνο να ανιχνεύσει την ύπαρξη ενός αντικείμενου σε κάποια απόσταση μπροστά από αυτό αλλά και να διακρίνει την

απόσταση στην οποία βρίσκεται τοποθετημένο το αντικείμενο αυτό.

Ο αισθητήρας ήχου μπορεί να μετρήσει την απόσταση σε εκατοστόμετρα (cm) ή σε ίντσες (inches). Μπορεί να μετρήσει αποστάσεις με μέγιστη τιμή τα 255 εκατοστόμετρα και ακρίβεια +/- 3 cm.

Λειτουργεί με την ίδιο τρόπο που χρησιμοποιούν οι νυκτερίδες για να έχουν όραση ή τα ραντάρ-σόναρ. Υπολογίζει την απόσταση μετρώντας το χρόνο που χρειάζεται σε ένα ηχητικό κύμα υπερήχων να κτυπήσει σε ένα αντικείμενο και να επιστρέψει πάλι πίσω στον αισθητήρα. Με αυτόν τον τρόπο έχουμε πιο πετυχημένη ανίχνευση όταν πρόκειται για αντικείμενα μεγάλα και με σκληρή επιφάνεια και λιγότερο πετυχημένη όταν πρόκειται για μικρά αντικείμενα, φτιαγμένα από μαλακή ίνα, πολύ λεπτά ή με κυρτές επιφάνειες.



Σημείωση : Αν χρησιμοποιηθούν πάνω από ένας αισθητήρας υπερήχων στον ίδιο χώρο θα έχουμε διασταύρωση των υπέρηχων που παράγονται και άρα όχι αξιόπιστες μετρήσεις.

Παραδείγματα χρήσης του



Μπορούμε να χρησιμοποιήσουμε ένα αισθητήρα υπερήχων ώστε το ρομπότ του να αποφεύγει τα εμπόδια που βρίσκονται μπροστά του πριν να τα πλησιάσει ή για να μπορεί να αντιληφθεί τότε ένα αντικείμενο διέρχεται από μπροστά του ή έχει πλησιάσει επικίνδυνα.

Αισθητήρες παλιού τύπου (NXT)

Στο μικροϋπολογιστή NXT μπορούμε να συνδέσουμε και τους αισθητήρες από την προηγούμενη σειρά NXT της Lego Mindstorms χρησιμοποιώντας το ειδικό καλώδιο προσαρμογής που μας παρέχεται. Παρακάτω θα κάνουμε μία αναφορά στους παλιούς αισθητήρες τύπου NXT.

1. Αισθητήρας αφής

Ο αισθητήρας αφής είναι ένα ειδικό τουβλάκι 2x3 γκρι χρώματος με ένα κίτρινο κουμπί στη μία άκρη του.



Ο αισθητήρας αφής ανιχνεύει τότε υπάρχει επαφή με τα αντικείμενα, στο περιβάλλον που δραστηριοποιείται ένα ρομπότ.

Όταν χτυπήσει π.χ σε ένα εμπόδιο το κουμπί του αισθητήρα πιέζεται. Με αυτόν τον τρόπο μπορεί το ρομπότ να αντιληφθεί τότε πέφτει πάνω σε ένα αντικείμενο.

Δρα σαν ένας διακόπτης. Όταν πατηθεί το κουμπί στέλνει στο μικροεπεξεργαστή τον αριθμό 1, ενώ σε αντίθετη περίπτωση τον αριθμό 0.

2. Αισθητήρας φωτός

Ο αισθητήρας φωτός είναι ένα ειδικό τουβλάκι 2x4 μπλε χρώματος.



Σενάρια Ρομποτικής με το Mindstorms NXT

Ο αισθητήρας φωτός μετράει την ποσότητα του φωτός που ανακλάται από μία επιφάνεια. Αναφέρει αυτή την ποσότητα στο μικροεπεξεργαστή με έναν αριθμό από το 0 (απόλυτο σκοτάδι) ως το 100 (άπλετο φως). Με αυτόν τον τρόπο μπορεί να προσδιορίσει αν βρίσκεται πάνω από ένα λευκό ή από ένα μαύρο κομμάτι χαρτιού. Όταν βρίσκεται πάνω σε ένα λευκό κομμάτι χαρτιού στέλνει στο μικροεπεξεργαστή μια αριθμητική τιμή κοντά στον αριθμό 50, ενώ όταν βρίσκεται πάνω σε ένα μαύρο κομμάτι χαρτιού στέλνει μια αριθμητική τιμή κοντά στον αριθμό 33.

3. Αισθητήρας περιστροφής

Ο αισθητήρας περιστροφής είναι ένα ειδικό τουβλάκι 2x4 μπλε χρώματος. Διαθέτει μια οπή στο πλάι για να τοποθετήσουμε ένα άξονα.



Ο αισθητήρας περιστροφής μετράει την περιστροφή του άξονα στον οποίο συνδέεται. Αναφέρει μια πλήρη περιστροφή του άξονα στο μικροεπεξεργαστή με τον αριθμό 16. Με τον αισθητήρα αυτόν μπορούμε να υπολογίσουμε πόσες φορές έχει γυρίσει ένας άξονας.

Σημείωση : Μετρώντας πόσες φορές έχει περιστραφεί ο άξονας σε ένα όχημα – ρομπότ, και παίρνοντας υπόψη μας την περιφέρεια των τροχών του, μπορούμε να υπολογίσουμε την απόσταση που έχει διανύσει το ρομπότ μας. Από την άλλη γνωρίζοντας την απόσταση που έχει διανύσει το ρομπότ μας σε κάποιο συγκεκριμένο χρονικό διάστημα, μπορούμε να υπολογίσουμε την ταχύτητα κίνησης του.

4. Αισθητήρας θερμοκρασίας

Ο αισθητήρας θερμοκρασίας είναι ένα ειδικό τουβλάκι 2x3 κίτρινου χρώματος. Διαθέτει μια προεξοχή για τη μέτρηση της θερμοκρασίας.



Ο αισθητήρας θερμοκρασίας μετράει τη θερμοκρασία που ανιχνεύει. Αναφέρει τη θερμοκρασία αυτή στο μικροεπεξεργαστή, με έναν αριθμό που αντιστοιχεί στην πραγματική θερμοκρασία.

Μικροεπεξεργαστής ή τούβλο NXT



Η καρδιά του πακέτου NXT είναι το τούβλο –μικροϋπολογιστής NXT. Το NXT είναι ο εγκέφαλος οποιουδήποτε ρομπότ της lego Mindstorms. Πρόκειται για ένα ευφυές, ελεγχόμενο από υπολογιστή τούβλο, το οποίο δίνει στη ρομποτική κατασκευή μας, τη δυνατότητα να ζωντανέψει και να εκτελέσει μία σειρά από διαφορετικές διαδικασίες.

Μέσα στο τούβλο – μικροϋπολογιστή NXT βρίσκεται ένας αυτόνομος μικροεπεξεργαστής των 32 bit (σε αντίθεση με τα 16 bit της πρώτης γενιάς), ο οποίος μπορεί να προγραμματιστεί μέσω ηλεκτρονικού υπολογιστή PC.

Τεχνικά χαρακτηριστικά :

- Μικροεπεξεργαστής 32-bit ARM7
- Μνήμη 256 Kbytes FLASH, 64 Kbytes RAM
- Μικροελεγκτή 8-bit AVR με μνήμη 4 Kbytes FLASH, 512 Byte RAM
- Δυνατότητα ασύρματης επικοινωνίας Bluetooth (Bluetooth Class II V2.0 compliant)
- Θύρα υψηλής ταχύτητας USB (12 Mbit/s)
- 4 θύρες εισόδου με καλώδιο 6-αγωγών (Η μία θύρα ακολουθεί το πρότυπο IEC 61158 Type 4/EN 50 170 για μελλοντική χρήση)
- 3 θύρες εξόδου με καλώδιο 6-αγωγών.
- Οθόνη γραφικών 100 x 64 pixel LCD
- Μεγάφωνο ηχητικής ποιότητας 8 kHz με εύρος καναλιού 8-bit και συχνότητα δειγματοληψίας 2-16 KHz.
- Τροφοδοσία με 6 μπαταρίες AA.

Για την τροφοδοσία του NXT μπορεί κανείς να χρησιμοποιήσει την επαναφορτιζόμενη μπαταρία.



Το NXT διαθέτει τέσσερις θύρες εισόδου για τη σύνδεση των αισθητήρων καθώς και τρεις θύρες εξόδου για τη σύνδεση των κινητήρων.

Στην κάτω πλευρά του τούβλου NXT βρίσκονται οι θύρες εισόδου 1,2,3 και 4 όπου μπορούμε να συνδέσουμε μέχρι τέσσερις αισθητήρες. Στο πάνω μέρος του βρίσκονται οι θύρες εξόδου A, B και C για τη σύνδεση των κινητήρων.

Στην πάνω πλευρά επίσης και δίπλα στις θύρες για τους κινητήρες υπάρχει μία θύρα USB μέσω της οποίας μπορούμε να συνδέσουμε το μικροϋπολογιστή NXT με έναν προσωπικό υπολογιστή χρησιμοποιώντας ένα καλώδιο USB. Από αυτή τη θύρα μπορούμε να μεταφορτώσουμε στο μικροϋπολογιστή NXT ένα πρόγραμμα που έχουμε δημιουργήσει στο PC μας μέσα από το προγραμματιστικό περιβάλλον NXT-G και στη συνέχεια να το τρέξουμε από το NXT αυτόνομα. Στην περίπτωση που το καλώδιο USB δε περιορίζει την κίνηση του ρομπότ μας μπορούμε να τρέξουμε το πρόγραμμα απευθείας από το PC οπότε και μπορούμε να παρακολουθούμε τις τρέχουσες τιμές

Σενάρια Ρομποτικής με το Mindstorms NXT

διαφόρων παραμέτρων του προγράμματος μας μέσα από το περιβάλλον του NXT-G. Βέβαια για να μην έχουμε κανένα περιορισμό στην κίνηση του ρομπότ μας μπορούμε να χρησιμοποιήσουμε τη σύνδεση Bluetooth για τη μεταφόρτωση του προγράμματος μας καθώς και τη συνεχή παρακολούθηση των παραμέτρων του.



Με το πακέτο Mindstorms NXT της Lego μπορεί κανείς αφού συνδέσει στο μικροϋπολογιστή NXT έναν αριθμό αισθητήρων και κινητήρων, γράφοντας το κατάλληλο πρόγραμμα και στη συνέχεια στέλνοντας το (μεταφορτώνοντας το) στο NXT -μέσω της σύνδεσης USB ή ακόμη και ασύρματα με χρήση σήματος Bluetooth- να προσδώσει την επιθυμητή συμπεριφορά στη ρομποτική του κατασκευή, ελέγχοντας κατάλληλα τους κινητήρες με βάση τα ερεθίσματα που προσλαμβάνονται από τους αντίστοιχους αισθητήρες.



Πλήκτρα του NXT

Ο μικροϋπολογιστής NXT διαθέτει τα παρακάτω πλήκτρα :

Πορτοκαλί πλήκτρο : Για το άνοιγμα του NXT (On), την επιβεβαίωση μίας ενέργειας (Enter) και το τρέξιμο ενός προγράμματος (Run)

Ανοικτό γκρι βελάκια : Για να μπορέσουμε να περιηγηθούμε δεξιά και αριστερά στα μενού του NXT.

Σκούρο γκρι πλήκτρο: Για τον καθαρισμό (Clear) και την επιστροφή (Go back)

Οθόνη Γραφικών

Στην μπροστινή όψη το τούβλο NXT περιλαμβάνει μία οθόνη LCD με ανάλυση 100 x 64 εικονοστοιχεία, από όπου μπορούμε να δούμε διάφορες πληροφορίες όταν χειριζόμαστε το NXT χωρίς την παρέμβαση ενός προσωπικού υπολογιστή είτε όταν βρίσκεται σε εξέλιξη το τρέξιμο ενός προγράμματος μας.

Μεγάφωνο

Το τούβλο NXT διαθέτει ένα ενσωματωμένο μεγάφωνο ώστε το ρομπότ μας να έχει τη δυνατότητα να αναπαράγει διάφορους ήχους.

Ο μικροϋπολογιστής NXT κυκλοφορεί σε δύο εκδόσεις στην 1 και τη 2. Πρόκειται για το ίδιο ακριβώς μοντέλο και η μόνη διαφορά τους, εντοπίζεται στο firmware που διαθέτουν.

Μια παρουσίαση για τους Κινητήρες, τους Αισθητήρες, και το τούβλο NXT του Πακέτο NXT θα βρείτε στο σύνδεσμο : [Πακέτο NXT – Κινητήρες, Αισθητήρες, τούβλο NXT](#)

Περιβάλλον προγραμματισμού του NXT



Αφού κατασκευάσει κανείς το ρομπότ του, μπορεί να δημιουργήσει ένα δικό του πρόγραμμα χρησιμοποιώντας ένα εύχρηστο αλλά πλούσιο σε χαρακτηριστικά λογισμικό, το NXT-G. Το NXT-G βασίζεται στο προγραμματιστικό περιβάλλον ρομποτικών κατασκευών LabVIEW, το οποίο έχει σχεδιαστεί από τη National Instruments.

Γράφουμε ένα πρόγραμμα σέρνοντας κάποιο εικονίδιο (μπλοκ εντολής) από την περιοχή της παλέτας εντολών στο κεντρικό παράθυρο, το παράθυρο εργασίας. Τα εικονίδια ενώνονται αυτόματα μόλις τα τοποθετήσουμε πάνω στη δοκό σύνδεσης (sequence beam).

Υπάρχουν 3 διαφορετικές μπάρες στην παλέτα εντολών :

Η **βασική παλέτα** εντολών που όπως υποδηλώνει και το όνομα της περιέχει τις βασικές εντολές, αυτές δηλ που μας επιτρέψουν να δημιουργήσουμε το πρόγραμμα που θέλουμε για την πλειοψηφία των προβλημάτων που επιθυμούμε να επιλύσουμε. Στην παλέτα αυτή συναντούμε τις εντολές για να κινήσουμε τη ρομποτικής μας κατασκευή, να βάλουμε το τούβλο NXT να αναπαράγει κάποιο ήχο ή μελωδία, να μας εμφανίσει κάποιο κείμενο ή γράφημα στην οθόνη του αλλά και τις πιο σύνθετες δομές προγραμματισμού όπως η αναμονή μέχρι να συμβεί κάποιο γεγονός, ή επανάληψη μιας διαδικασίας και η λήψη απόφασης από το ρομπότ μας ανάλογα με τις επικρατούσες συνθήκες.

Η **πλήρης παλέτα** εντολών από την οποία μπορούμε να χρησιμοποιήσουμε το σύνολο των δυνατοτήτων που υποστηρίζει ο μικροεπεξεργαστής NXT για την απόδοση της κατάλληλης συμπεριφοράς στη ρομποτική κατασκευή μας.

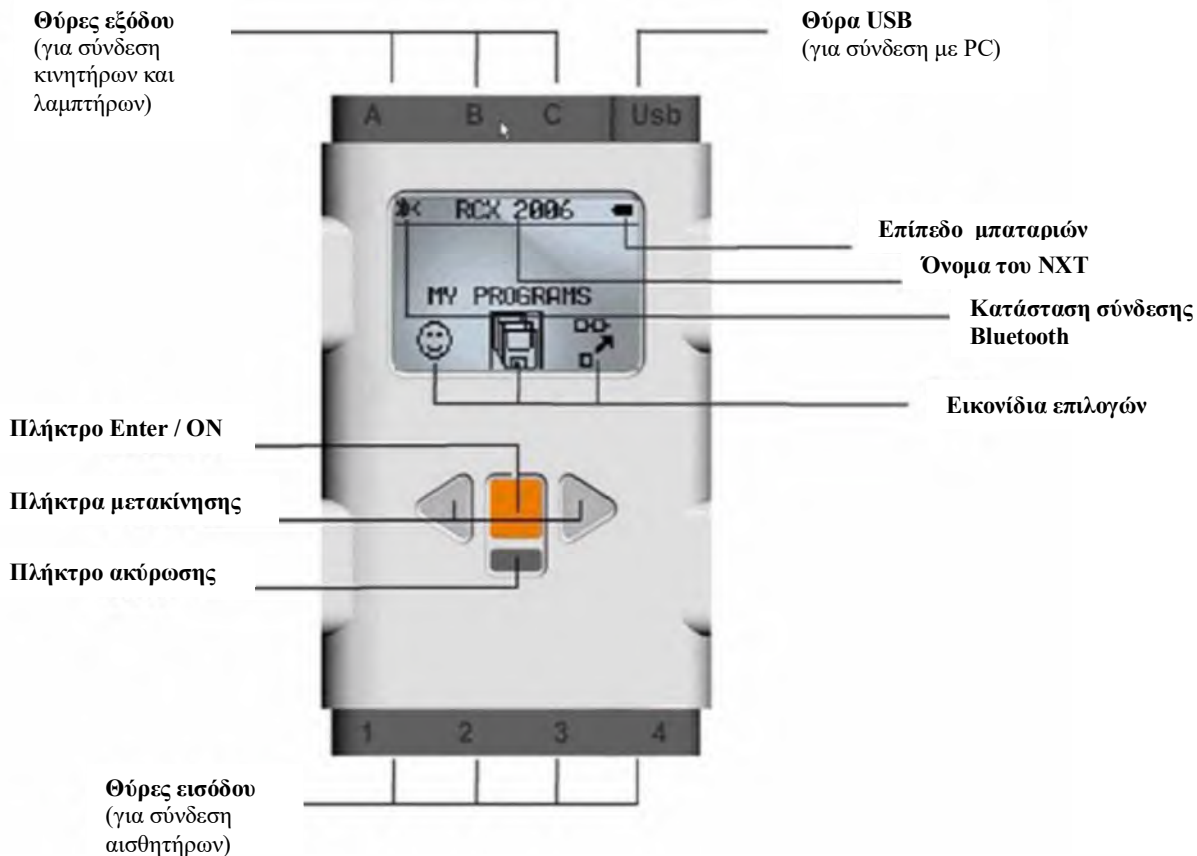
Η **προσαρμοσμένη παλέτα** εντολών στην οποία μπορούμε να βάλουμε κάποιες δικές μας εντολές. Τις εντολές αυτές μπορούμε να τις δημιουργήσουμε συνθέτοντας μία διαδικασία από ένα συνδυασμό εντολών από αυτές που διαθέτει το προγραμματιστικό περιβάλλον. Με αυτό τον τρόπο δημιουργούμε μία δική μας βιβλιοθήκη με ένα αριθμό από διαθέσιμες υπορουτίνες.

Το λογισμικό NXT-G προσφέρεται σε δύο εκδόσεις :

- Τη οικιακή (home edition) μπορείτε να την κατεβάσετε από την ιστοσελίδα :
<https://www.lego.com/en-gr/themes/mindstorms/downloads>
Πηγαίνετε στην περιοχή **NXT Software Download (PC/MAC)** και κάνετε 'κλικ' στο κουμπί **NXT Software Download (PC)**
- Την εκπαιδευτική (education edition) μπορείτε να την κατεβάσετε από την ιστοσελίδα :
<https://education.lego.com/en-us/downloads/retiredproducts/nxt/software>
κάνοντας 'κλικ' στο κουμπί **DOWNLOAD** από την περιοχή **Download your NXT software v. 2.1.0**, αφού επιλέξετε την έκδοση για windows.

Λειτουργία του NXT

Το τούβλο NXT αποτελεί τον εγκέφαλο των ρομποτικών κατασκευών του πακέτου LEGO MINDSTORMS. Πρόκειται για ένα ευφυές προγραμματιζόμενο τούβλο το οποίο προσδίδει στη ρομποτική κατασκευή - μετά από δική μας παρέμβαση- έξυπνες συμπεριφορές με δυνατότητα λήψης αποφάσεων.



Υποδοχές

Θύρες εισόδου

Το NXT διαθέτει τέσσερις θύρες εισόδου 1,2,3 και 4, για τη σύνδεση αισθητήρων.

Θύρες εξόδου

Το NXT διαθέτει τρεις θύρες εξόδου A,B και C για τη σύνδεση κινητήρων και λαμπτήρων.

Θύρα USB

Συνδέοντας τον προσωπικό υπολογιστή μας μέσω της θύρας USB αυτής κατεβάζουμε (μεταφορτώνουμε) τα προγράμματα που δημιουργούμε στο NTX ή ανεβάζουμε δεδομένα από αυτό.

Σημείωση : Μπορούμε εναλλακτικά να κάνουμε χρήση της σύνδεσης Bluetooth και να έχουμε διαρκή εποπτεία του NXT.

Υποδοχή τροφοδοσίας

Μπορούμε να συνδέσουμε το φιν τροφοδοσίας ρεύματος για να φορτίσουμε την επαναφορτιζόμενη μπαταρία του NXT. Μπορούμε επίσης στην περίπτωση που το καλώδιο τροφοδοσίας ρεύματος δεν μας

Σενάρια Ρομποτικής με το Mindstorms NXT

περιορίζει στη λειτουργία της ρομποτικής μας κατασκευής (πχ όταν πρόκειται για στατική ρομποτική κατασκευή ή με περιορισμένη κίνηση.) να αφήσουμε το φινις τροφοδοσίας μόνιμα συνδεδεμένο στο NXT.

Εικονίδια



Εικονίδιο λειτουργίας (Running icon)

Όταν το NXT είναι σε λειτουργία το εικονίδιο αυτό στριφογυρίζει.

Σημείωση: αν το εικονίδιο αυτό σταματήσει να κινείται σημαίνει ότι το NXT έχει μπλοκάρει και πρέπει να πατήσουμε το πλήκτρο reset για να αρχικοποιήσουμε τα κυκλώματα του.



Επίπεδο Μπαταριών (Battery level)

Το εικονίδιο αυτό εμφανίζει το επίπεδο ισχύος των μπαταριών.

Όταν αυτό πέσει κάτω από το 10 % της χωρητικότητας των μπαταριών το εικονίδιο αναβοσβήνει και θα πρέπει να προχωρήσουμε σε αντικατάσταση των μπαταριών.



Εικονίδιο σύνδεσης USB

Μόλις συνδέσουμε το NXT με έναν προσωπικό υπολογιστή εμφανίζεται το εικονίδιο USB το οποίο και εξαφανίζεται μόλις αποσυνδέσουμε το καλώδιο USB.

Στην περίπτωση που η σύνδεση USB δε λειτουργεί σωστά εμφανίζεται το παρακάτω εικονίδιο :



Εικονίδιο σύνδεσης Bluetooth

Το εικονίδιο αυτό μας πληροφορεί για την τρέχουσα κατάσταση μιας ασύρματης σύνδεσης Bluetooth. Αν δεν είναι ορατό αυτό το εικονίδιο η δυνατότητα σύνδεσης Bluetooth του NXT είναι απενεργοποιημένη.

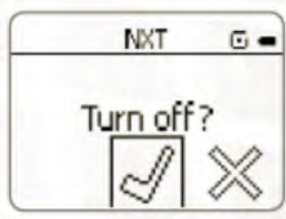
Αν η δυνατότητα Bluetooth είναι ενεργοποιημένη αλλά έχουμε ρυθμίσει το NXT μας να μην είναι ορατό από άλλες συσκευές με σήμα Bluetooth εμφανίζεται το παρακάτω εικονίδιο : *

Αν η δυνατότητα Bluetooth είναι ενεργοποιημένη και το NXT μας είναι ορατό από άλλες συσκευές με σήμα Bluetooth εμφανίζεται το παρακάτω εικονίδιο : *<

Αν η δυνατότητα Bluetooth είναι ενεργοποιημένη και το NXT μας έχει πραγματοποιήσει σύνδεση με μία άλλη συσκευή μέσω Bluetooth εμφανίζεται το παρακάτω εικονίδιο : *◇

Τερματισμός της λειτουργίας του NXT

Για να τερματίσουμε τη λειτουργία του NXT πατάμε το γκρι πλήκτρο (go back) μέχρι να εμφανιστεί η παρακάτω οθόνη :



Στο σημείο αυτό πατάμε το πορτοκαλί πλήκτρο (enter) για να επιβεβαιώσουμε τον τερματισμό. (Αν έχουμε μετανιώσει πατάμε πάλι το γκρι πλήκτρο για να επιστρέψουμε στο κυρίως μενού του NXT)

Απόδοση ονόματος στο NXT

Μπορούμε να αλλάξουμε το όνομα του NXT μας (το εξ' ορισμού όνομα είναι NXT) ώστε να το ξεχωρίζουμε από τυχόν άλλους μικροϋπολογιστές NXT. Στην περίπτωση που θέλουμε να υλοποιήσουμε μία δραστηριότητα με πολλά ρομπότ, στην οποία αυτά πρέπει να επικοινωνήσουν και μεταξύ τους μέσω Bluetooth, επιβάλλεται τα NXT που θα συμμετέχουν στη δραστηριότητα αυτή να έχουν και διαφορετικά ονόματα.

Η αλλαγή του ονόματος του NXT γίνεται μέσα από το περιβάλλον προγραμματισμού NXT-G. Επιλέγουμε το πλήκτρο NXT Window από το πεδίο ελέγχου (Controller) που βρίσκεται στην κάτω δεξιά γωνία του λογισμικού NXT-G. Αν δεν υπάρχει ήδη, αποκαθιστούμε μία σύνδεση με το NXT μας είτε μέσω του καλωδίου USB είτε ασύρματα μέσω Bluetooth. Στο πεδίο *Your current NXT is : Name* πληκτρολογούμε το νέο όνομα. Το όνομα ενός NXT επιτρέπεται να περιλαμβάνει μέχρι 8 χαρακτήρες.



Υπομενού

Τα αρχεία μου (My Files)



Στους υποφακέλους του φακέλου *My Files* του NXT αποθηκεύονται όλα τα προγράμματα που έχουμε δημιουργήσει απευθείας στο NXT (NXT files) ή έχουμε μεταφορτώσει από τον προσωπικό μας υπολογιστή (software files).

Τα αρχεία που μεταφορτώνουμε στο NXT τοποθετούνται αυτόματα στον κατάλληλο φάκελο. Για παράδειγμα όταν κατεβάσουμε στο NXT μας ένα πρόγραμμα που χρησιμοποιεί ένα αρχείο ήχου, το ίδιο το πρόγραμμα θα τοποθετηθεί στο φάκελο των αρχείων προγραμμάτων (Software files) ενώ το αρχείο ήχου θα τοποθετηθεί στο φάκελο των αρχείων ήχων (Sound files).

Υπάρχουν τέσσερεις διαφορετικοί υποφάκελοι :

Φάκελος αρχείων προγραμμάτων [Software files] για τα προγράμματα που μεταφορτώνουμε από τον προσωπικό μας υπολογιστή.

Φάκελος αρχείων NXT [NXT files] για τα προγράμματα που δημιουργούμε απευθείας στο NXT μας.

Φάκελος αρχείων ήχου [Sound files] για τους ήχους που περιέχονται σε ένα πρόγραμμα που έχουμε μεταφορτώσει.

Φάκελος αρχείων Δειγματοληψίας [Datalog files] για τα δεδομένα που έχουν καταγραφεί κατά τη διάρκεια μιας δειγματοληψίας δεδομένων.

Προγράμματα για δοκιμή (Try Me)



Το μενού δοκιμών του NXT [Try Me] απευθύνεται στους αρχάριους χρήστες και επιτρέπει την εξοικείωση -μέσω του πειραματισμού- με τους αισθητήρες και τους κινητήρες, χρησιμοποιώντας κάποια έτοιμα προγράμματα που βρίσκονται στο υπομενού αυτό.

Τα προγράμματα αυτά μετά την ολοκλήρωση της εξοικείωσης από ένα αρχάριο, δεν υπάρχει λόγος να παραμένουν στο NXT.

Σβήνοντας τα, μπορούμε να εξοικονομήσουμε κάποιο πολύτιμο χώρο στη μνήμη του NXT για τα δικές μας εργασίες. Η διαγραφή τους είναι δυνατή από το μενού Ρυθμίσεις [Settings] του NXT : [Settings] → [Delete files] → [Try me files]

Παρόλα αυτά, αν αργότερα θέλουμε να τα ανακτήσουμε δεν έχουμε παρά να κατεβάσουμε το firmware ξανά.

ΠΡΟΣΟΧΗ : Για να λειτουργήσουν σωστά τα προγράμματα αυτά θα πρέπει οι αισθητήρες και κινητήρες έχουν συνδεθεί στις εξ' ορισμού θύρες. Στα προγράμματα αυτά πρέπει να ακολουθηθεί η παρακάτω παραδοχή :

Ο αισθητήρας αφής είναι συνδεδεμένος τη θύρα εισόδου 1.

Ο αισθητήρας ήχου είναι συνδεδεμένος στη θύρα εισόδου 2.

Ο αισθητήρας φωτός είναι συνδεδεμένος στη θύρα εισόδου 3.

Ο αισθητήρας υπερήχων είναι συνδεδεμένος στη θύρα εισόδου 4.

Παράδειγματα δοκιμής :

Παρακάτω περιγράφεται η διαδικασία για τη δοκιμή λειτουργίας ενός αισθητήρα αφής.

Καταρχήν συνδέστε έναν αισθητήρα αφής στη θύρα εισόδου 1 του τούβλου NXT.

Αφού εισέρθετε στο μενού [Try me] επιλέξτε με χρήση των ανοικτών γκρι τριγωνικών πλήκτρων του NXT το εικονίδιο [Try-Touch] και πατήστε το πορτοκαλί πλήκτρο στο NXT δύο φορές για να ξεκινήσει η δοκιμή της λειτουργίας του αισθητήρα αφής.

Παρατηρήστε ότι κάθε φορά που πατάτε το διακόπτη του αισθητήρα αφής στην οθόνη του NXT το ανθρωπάκι που υπάρχει χαμογελάει και ακούγεται το επιφώνημα «ουπς».

Αν επιλέξετε το εικονίδιο [Try-Sound] αφού έχετε συνδέσει έναν αισθητήρα ήχου στη θύρα εισόδου 2 και δύο κινητήρες στις θύρες εξόδου B και C, κάθε φορά που μιλάτε γυρίζουν οι δύο κινητήρες με ταχύτητα ανάλογα με την ένταση της ομιλίας.

Αν επιλέξετε το εικονίδιο [Try-Light] αφού έχετε συνδέσει έναν αισθητήρα φωτός στη θύρα εισόδου 3, καθώς τοποθετείτε τον αισθητήρα σε διάφορα σημεία του χώρου που βρίσκεστε, αναπαράγεται από το NXT μία μελωδία ανάλογα με την ένταση φωτός που ανιχνεύεται από το περιβάλλον.

Αν επιλέξετε το εικονίδιο [Try-Ultrasound] αφού έχετε συνδέσει έναν αισθητήρα υπερήχων στη θύρα εισόδου 4, καθώς τοποθετείτε το χέρι σας μπροστά από τον αισθητήρα μετακινώντας το μπρος πίσω, αναπαράγεται από το NXT μία μελωδία ανάλογα με την απόσταση που βρίσκεται το χέρι σας.

Αν επιλέξετε το εικονίδιο [Try-Motor] αφού έχετε συνδέσει έναν κινητήρα στη θύρα εξόδου A, καθώς περιστρέφετε τον άξονα περιστροφής του κινητήρα, αναπαράγεται από το NXT μία μελωδία ανάλογα με τη θέση στην οποία έχετε περιστρέψει τον άξονα.

Εμφάνιση Πληροφοριών (View)



Με τη βοήθεια του μενού εμφάνισης πληροφοριών [View] μπορούμε να δούμε τις τρέχουσες τιμές των παραμέτρων των διαφόρων αισθητήρων και κινητήρων που είναι συνδεδεμένοι στο NXT.

Καταρχήν συνδέουμε στο τούβλο NXT τον αισθητήρα που θέλουμε (ή έναν κινητήρα όταν θέλουμε να χρησιμοποιήσουμε τον αισθητήρα περιστροφής) σε μία αντίστοιχη θύρα.

Πηγαίνουμε στο μενού εμφάνισης [View] και επιλέγουμε το υπομενού με την εικόνα του αισθητήρα ή ενός κινητήρα που θέλουμε ανάμεσα στα : Sound dB, Sound dBA, Reflected Light, Ambient Light, Light Sensor*, Temperature °C*, Temperature °F*, Rotation*, Motor Rotations, Motor Degrees, Touch, Ultrasonic Inch και Ultrasonic cm.

Στη συνέχεια επιλέγουμε τη θύρα στην οποία έχουμε συνδέσει την αντίστοιχη συσκευή. Αμέσως στην οθόνη μας εμφανίζονται δεδομένα από τη συσκευή που έχουμε συνδέσει.

Παράδειγματα δοκιμής :

Παρακάτω περιγράφεται η διαδικασία για τη εμφάνιση πληροφοριών από έναν αισθητήρα αφής.

Συνδέστε έναν αισθητήρα αφής σε μία θύρα εισόδου του τούβλου NXT.

Αφού εισέρθετε στο μενού [View] επιλέξτε με χρήση των ανοικτών γκρι τριγωνικών πλήκτρων του NXT το εικονίδιο του αισθητήρα αφής [Touch] και πατήστε το πορτοκαλί πλήκτρο στο NXT. Στη συνέχεια επιλέξτε τη θύρα εισόδου στην οποία έχετε συνδέσει τον αισθητήρα αυτόν. Πατήστε και κρατήστε πατημένο το διακόπτη του αισθητήρα αφής. Στην οθόνη του NXT εμφανίζεται η τιμή 1. Μόλις αφήσετε το διακόπτη αμέσως η τιμή θα αλλάξει σε 0.

Συνδέστε έναν αισθητήρα ήχου και επιλέξτε ένα από τα εικονίδια [Sound dB] ή [Sound dBA]. Στην οθόνη του NXT, κάθε φορά που μιλάτε ή δημιουργείται διαφορετικούς ήχους κοντά στον αισθητήρα ήχου, θα εμφανίζεται η ένταση του ήχου που θα καταγράφει ο αισθητήρας αυτός. Στη δεύτερη περίπτωση για τη μέτρηση λαμβάνεται υπόψη και η ευαισθησία του ανθρώπινου αυτιού στις διαφορετικές συχνότητες με αποτέλεσμα η μέτρηση να ταιριάζει με την ένταση που αντιλαμβανόμαστε με την ακοή μας.

Συνδέστε έναν αισθητήρα φωτός και επιλέξτε το εικονίδιο [Reflected Light]. Στην οθόνη του NXT, κάθε φορά που τοποθετείτε τον αισθητήρα φωτός πάνω από μία επιφάνεια διαφορετικού χρώματος, θα εμφανίζεται η φωτεινότητα του φωτός που θα ανακλάται από τις επιφάνειες αυτές. Αν επιλέξετε το εικονίδιο [Ambient Light] και περιφέρετε τον αισθητήρα φωτός σε διαφορετικά σημεία του δωματίου στην οθόνη του NXT θα εμφανίζεται η ένταση του φωτός στα σημεία αυτά.

Συνδέστε έναν αισθητήρα υπερήχων και επιλέξτε το εικονίδιο [Ultrasonic Inch] ή [Ultrasonic cm]. Στην οθόνη του NXT, κάθε φορά που τοποθετείτε μπροστά από τον αισθητήρα υπερήχων κάποιο εμπόδιο, θα εμφανίζεται η απόσταση σε ίντσες ή εκατοστά που υπολογίζεται από τον αισθητήρα αυτόν.

Συνδέστε έναν κινητήρα σε μία θύρα εξόδου, προσαρμόστε έναν τροχό στον άξονα του κινητήρα και επιλέξτε το εικονίδιο [Motor Rotations] ή [Motor Degrees]. Στην οθόνη του NXT, θα εμφανίζονται οι πλήρες περιστροφές ή οι περιστροφές σε μοίρες που θα πραγματοποιεί ο τροχός αυτός καθώς τον κυλάμε. (Στην περίπτωση αυτή η μέτρηση γίνεται από τον ενσωματωμένο αισθητήρα περιστροφής με τον οποίο είναι εφοδιασμένος ο κινητήρας).

Προγράμματα απ' ευθείας στο NXT (NXT Program)



Σε αντίθεση με το προηγούμενο μικρουπολογιστή RCX της σειράς Lego Mindstorms στον οποίο δε μπορούσαμε να δώσουμε εντολές απευθείας από τον ίδιο το μικροεπεξεργαστή, στη σειρά NXT δε χρειαζόμαστε απαραίτητα τον προσωπικό υπολογιστή για να προγραμματίσουμε μία ρομποτική κατασκευή.

Μπορούμε να χρησιμοποιήσουμε τις εντολές που περιέχονται στο μενού NXT προγράμματα [NXT Program].

Πρέπει όμως να αναφερθεί ότι με αυτό τον τρόπο προγραμματισμού μπορούμε να δημιουργήσουμε μόνο κάποια απλά προγράμματα (παρόμοια με τον τρόπο προγραμματισμού Pilot του προγραμματιστικού περιβάλλοντος Robolab) της μορφής "ένέργεια – αναμονή – άλλη ενέργεια – ξανά αναμονή – σταμάτημα ή επανάληψη" (Output-Input- Output-Input-Next).

Δε μπορούμε να δώσουμε πχ. συμπεριφορές στο NXT μας που να επιτρέπουν τη λήψη αποφάσεων. Αν ο στόχος μας είναι να δημιουργήσουμε κάποιο αξιόλογο πρόγραμμα πρέπει να στραφούμε στο προγραμματιστικό περιβάλλον NXT-G.

Για να διαγράψουμε ένα πρόγραμμα NXT μετακινούμαστε στο μενού Ρυθμίσεις [Settings] του NXT : [Settings] → [Delete files] → [NXT files]

ΠΡΟΣΟΧΗ : Για να λειτουργήσουν σωστά τα προγράμματα αυτά θα πρέπει οι αισθητήρες και κινητήρες έχουν συνδεθεί στις εξ' ορισμού θύρες. Στα προγράμματα αυτά πρέπει να ακολουθηθεί η παρακάτω παραδοχή :

Ο αισθητήρας αφής είναι συνδεδεμένος τη θύρα εισόδου 1.

Ο αισθητήρας ήχου είναι συνδεδεμένος στη θύρα εισόδου 2.

Ο αισθητήρας φωτός είναι συνδεδεμένος στη θύρα εισόδου 3.

Ο αισθητήρας υπερήχων είναι συνδεδεμένος στη θύρα εισόδου 4.

Ο αριστερός (κοιτώντας από μπροστά) κινητήρας είναι συνδεδεμένος στη θύρα εξόδου B.

Ο δεξιός (κοιτώντας από μπροστά) κινητήρας είναι συνδεδεμένος στη θύρα εξόδου C.

Παράδειγμα δημιουργίας προγράμματος :

Παρακάτω περιγράφεται η διαδικασία δημιουργίας ενός προγράμματος με το οποίο το ρομπότ θα αλλάζει φορά κίνησης κάθε φορά που πατάμε το διακόπτη σε έναν αισθητήρα αφής. Πρώτα βάζουμε το ρομπότ μας να κινηθεί προς τα μπροστά. Στη συνέχεια όταν πατηθεί ο διακόπτης του αισθητήρα αφής το ρομπότ μας θα κινηθεί προς τα πίσω. Όταν πατήσουμε ξανά το διακόπτη του αισθητήρα αφής το ρομπότ μας θα κινηθεί και πάλι προς τα εμπρός. Αυτό θα εξακολουθεί να συμβαίνει συνέχεια μέχρι εμείς να διακόψουμε την εκτέλεση αυτού του προγράμματος.

Αφού εισέρθετε στο μενού NXT Programs επιλέξτε με χρήση των ανοικτών γκρι τριγωνικών πλήκτρων του NXT την εντολή *Forward* και πατήστε το πορτοκαλί πλήκτρο στο NXT για να οριστικοποιήσουμε την επιλογή αυτή.

Με τον ίδιο τρόπο επιλέξτε διαδοχικά τις εντολές *Touch*, *Backward*, ξανά *Touch* και τέλος *Loop* για να πούμε στο πρόγραμμα τρέχει ξανά και ξανά μέχρι να κλείσουμε το NXT.

Το πρόγραμμα θα έχει την παρακάτω μορφή :



Σενάρια Ρομποτικής με το Mindstorms NXT

Για να τρέξετε το πρόγραμμα επιλέξτε το υπομενού εκτέλεσης προγράμματος [Run].

Για να αποθηκεύσετε το πρόγραμμα επιλέξτε το υπομενού αποθήκευσης [Save] και πληκτρολογήστε ένα όνομα για το πρόγραμμα αυτό.

Το σύνολο των εντολών που υποστηρίζεται από το μενού NXT Program φαίνεται στον παρακάτω πίνακα :

Κατηγορία Ενεργειών (Outputs)	
	Κίνηση μπροστά. (κινητήρες B και C μπροστά) Οι κινητήρες θα συνεχίσουν να κινούνται ή θα σταματήσουν ανάλογα με την επόμενη εντολή που ακολουθεί.
	Κίνηση μπροστά για 5 δευτερόλεπτα. (κινητήρες B και C μπροστά)
	Κίνηση μπροστά με ελαφριά στροφή δεξιά.
	Κίνηση μπροστά με στροφή 45° δεξιά.
	Κίνηση μπροστά με ελαφριά στροφή αριστερά.
	Κίνηση μπροστά με στροφή 45° αριστερά
	Κίνηση προς τα πίσω. (κινητήρες B και C μπροστά) Οι κινητήρες θα συνεχίσουν να κινούνται ή θα σταματήσουν ανάλογα με την επόμενη εντολή που ακολουθεί.
	Κίνηση προς τα πίσω για 5 δευτερόλεπτα.

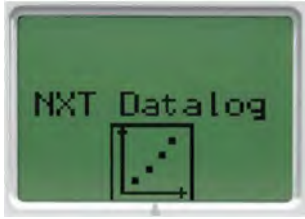
Σενάρια Ρομποτικής με το Mindstorms NXT

	Κίνηση προς τα πίσω με ελαφριά στροφή αριστερά.
	Κίνηση προς τα πίσω με στροφή 45° αριστερά
	Κίνηση προς τα πίσω με ελαφριά στροφή δεξιά.
	Κίνηση προς τα πίσω με στροφή 45° αριστερά
	Αναπαραγωγή τόνου χαμηλής συχνότητας για 1 δευτερόλεπτο.
	Αναπαραγωγή τόνου υψηλής συχνότητας για 1 δευτερόλεπτο.
	Κενό : Το NXT θα συνεχίσει με την εκτέλεση της επόμενης εντολής.
Κατηγορία Αναμονής (Inputs)	
	Αναμονή για 2 δευτερόλεπτα
	Αναμονή για 5 δευτερόλεπτα

Σενάρια Ρομποτικής με το Mindstorms NXT

	Αναμονή για 10 δευτερόλεπτα
	Αναμονή μέχρι να πατηθεί ο διακόπτης του αισθητήρα αφής. Χρησιμοποιείται συνήθως ώστε το ρομπότ μας να είναι σε θέση να ανιχνεύει τότε συγκρούεται σε ένα εμπόδιο.
	Αναμονή μέχρι ο αισθητήρας φωτός να ανιχνεύσει κάποια αύξηση της φωτεινότητας. (Με αυτήν την εντολή μπορούμε να αλλάξουμε την κατεύθυνση της κίνησης όταν πχ ανάψουμε τη λάμπα στο δωμάτιο.)
	Αναμονή μέχρι ο αισθητήρας φωτός να ανιχνεύσει κάποια μείωση της φωτεινότητας. (Με αυτήν την εντολή μπορούμε να αλλάξουμε την κατεύθυνση της κίνησης όταν πχ σβήσουμε τη λάμπα στο δωμάτιο.)
	Αναμονή μέχρι ο αισθητήρας ήχου να ανιχνεύσει κάποιον ήχο. (Με αυτήν την εντολή μπορούμε να αλλάξουμε την κατεύθυνση της κίνησης όταν πχ χτυπήσουμε παλαμάκια.)
	Αναμονή μέχρι ο αισθητήρας υπερήχων να ανιχνεύσει κάποιο εμπόδιο που βρίσκεται σε κάποια απόσταση. Χρησιμοποιείται συνήθως ώστε το ρομπότ μας να είναι σε θέση να ανιχνεύει ένα εμπόδιο που βρίσκεται μπροστά στην πορεία κίνησης του ή που διέρχεται από μπροστά του.
	Κενό : Το NXT θα συνεχίσει με την εκτέλεση της επόμενης εντολής.
Επανάληψη ή σταμάτημα του προγράμματος	
	Επανάληψη των προηγούμενων εντολών.
	Τερματισμός του προγράμματος.

Δειγματοληψία NXT (NXT Datalog)



Το μενού Δειγματοληψία NXT [NXT Datalog] μας επιτρέπει να συλλέξουμε μια σειρά από δεδομένα χωρίς να χρειάζεται εκείνη τη στιγμή το NXT μας να έχει επικοινωνία με τον προσωπικό μας υπολογιστή.

Τρέχοντας το πρόγραμμα NXT Datalog δημιουργείται ένα αρχείο τύπου log με δεδομένα, το οποίο αποθηκεύεται στη μνήμη του NXT. Το αρχείο αυτό στη συνέχεια μπορούμε να το εισάγουμε στον προσωπικό μας υπολογιστή για να προχωρήσουμε στην ανάλυση των δεδομένων μας.

Καταρχήν συνδέουμε στο τούβλο NXT τον αισθητήρα που θέλουμε (ή έναν κινητήρα όταν θέλουμε να χρησιμοποιήσουμε τον αισθητήρα περιστροφής) σε μία αντίστοιχη θύρα.

Πηγαίνουμε στο μενού δειγματοληψίας [NXT Datalog] και επιλέγουμε το υπομενού με την εικόνα του αισθητήρα ή ενός κινητήρα που θέλουμε ανάμεσα στα : Sound dB, Sound dBA, Reflected Light, Ambient Light, Light Sensor*, Temperature °C*, Temperature °F*, Rotation*, Motor Rotations, Motor Degrees, Touch, Ultrasonic Inch και Ultrasonic cm.

Στη συνέχεια επιλέγουμε τη θύρα στην οποία έχουμε συνδέσει την αντίστοιχη συσκευή και ξεκινάμε τη συλλογή των δεδομένων με την επιλογή εκτέλεσης [Run]. Το πρόγραμμα ξεκινάει αμέσως να συλλέγει δεδομένα από τη συσκευή που έχουμε συνδέσει. Για να σταματήσουμε τη συλλογή δεδομένων πατάμε το γκρι πλήκτρο στο NXT οπότε και τα δεδομένα που συλλέχτηκαν αποθηκεύονται αυτόματα σε ένα αρχείο στο υποφάκελο Datalog του NXT.

Ρυθμίσεις (Settings)



Με τη βοήθεια του μενού ρυθμίσεις [Settings] μπορούμε να καθορίσουμε τις τιμές για διάφορες ρυθμίσεις στο μικροϋπολογιστή NXT, όπως :

Αδράνεια (Sleep) :

Από το υπομενού *Sleep* μπορούμε να ρυθμίσουμε το NXT να σβήνει αυτόματα όταν έχουν περάσει 2,5,10,30,60 λεπτά της ώρας χωρίς να χρησιμοποιηθεί, ώστε να εξοικονομήσουμε ενέργεια. Μπορούμε επίσης να το ρυθμίσουμε να μη σβήνει ποτέ από μόνο του. Σε αυτή την περίπτωση όμως πρέπει να είμαστε προσεκτικοί και να κλείνουμε το NXT όταν δε το χρησιμοποιούμε, διαφορετικά οι μπαταρίες του θα αποφορτιστούν πολύ γρήγορα.

Σημαντικό : Πριν τρέξουμε κάποιο πρόγραμμα στο οποίο έχουμε εισάγει κάποια εντολή αναμονής και μπορεί να χρειαστεί να περιμένει για αρκετό χρόνο χωρίς να κάνει τίποτα, επιβάλλεται να ελέγξουμε τη ρύθμιση Sleep στο τούβλο NXT και αν χρειάζεται να ορίσουμε έναν αρκετά μεγάλο χρόνο αναμονής.

Ρύθμιση έντασης ήχου (Volume) :

Από το υπομενού *Volume* μπορούμε να ρυθμίσουμε την ένταση του ήχου για το ενσωματωμένο μεγάφωνο του NXT στην περιοχή από 0 (κλειστός) έως 4 (δυνατά).

Διαγραφή αρχείων (Delete) :

Από το υπομενού *Delete* μπορούμε να διαγράψουμε αρχεία του NXT από τους υποφακέλους του : Software files, NXT files, Sound files και Try Me files.

Σύνδεση Bluetooth



Από το μενού *Bluetooth* μπορούμε να ρυθμίσουμε τις παραμέτρους για την επίτευξη μιας ασύρματης σύνδεσης Bluetooth ανάμεσα στο NXT μας και άλλες συσκευές που υποστηρίζουν αυτό το πρωτόκολλο σύνδεσης, όπως άλλες μονάδες NXT, υπολογιστές και κινητά τηλέφωνα.

Εναλλακτικά αντί της σύνδεσης USB μπορούμε να μεταφορτώσουμε τα προγράμματα μας στο NXT με μία ασύρματη σύνδεση Bluetooth.

Ενεργοποιώντας μία ασύρματη επικοινωνία Bluetooth μεταξύ διαφορετικών μονάδων NXT οι ρομποτικές κατασκευές που λαμβάνουν μέρος σε μία δραστηριότητα είναι σε θέση να ανταλλάσσουν πληροφορίες και να αλληλεπιδρούν μεταξύ τους. Μπορούμε επίσης να στείλουμε προγράμματα είτε μεμονωμένα σε κάθε NXT είτε και μαζικά σε όλα μαζί ως ομάδα (group). Κάθε τέτοια ομάδα μπορεί να περιλαμβάνει μέχρι τρεις μονάδες NXT.

Ρυθμίζοντας μία ασύρματη επικοινωνία Bluetooth ανάμεσα σε ένα NXT και ένα κινητό τηλέφωνο μπορούμε με τη χρήση κατάλληλου προγράμματος από το κινητό μας να ελέγχουμε διαδραστικά μία ρομποτική μας κατασκευή.

Υπομενού Bluetooth

Ενεργή / ανενεργή (On/Off)

Για να μπορέσει να πραγματοποιηθεί μία σύνδεση Bluetooth θα πρέπει να ενεργοποιήσουμε τη λειτουργία αυτή. Αν την απενεργοποιήσουμε δεν θα είναι πλέον εφικτή η αποστολή και η λήψη δεδομένων στο NXT μας, οπότε θα πρέπει να χρησιμοποιήσουμε τη σύνδεση USB για τη μεταφόρτωση προγραμμάτων από τον προσωπικό μας υπολογιστή. Για να εξοικονομήσουμε ενέργεια μπορούμε να απενεργοποιούμε τη λειτουργία Bluetooth για το διάστημα που δε σκοπεύουμε να τη χρησιμοποιήσουμε. Αρχικά αυτή η λειτουργία είναι απενεργοποιημένη.

Ορατό ή όχι (Visibility : Visible/Invisible)

Από το υπομενού αυτό επιλέγουμε αν το NXT μας θα είναι ορατό (θα μπορεί να ανιχνευτεί από άλλες συσκευές Bluetooth, εκτός αυτών που υπάρχουν στη λίστα των επαφών [my contacts]). Από τη στιγμή που μετά την αναζήτηση (search) έχουμε εντοπίσει τις συσκευές με τις οποίες επιθυμούμε να επικοινωνούμε ασύρματα και έχουμε ήδη πραγματοποιήσει μία ασύρματη σύνδεση μαζί τους, οπότε αυτές έχουν προστεθεί στη λίστα των επαφών μας, μπορούμε να απενεργοποιήσουμε την ιδιότητα αυτή και να καταστήσουμε το NXT μας αόρατο σε οποιαδήποτε άλλη ξένη συσκευή Bluetooth.

Αναζήτηση (Search)

Με τη εντολή αυτή το NXT μας ξεκινάει μία αναζήτηση για να εντοπίσει τις συσκευές Bluetooth που βρίσκονται στο πεδίο δράσης του (Οι συσκευές αυτές για να μπορέσουν να ανιχνευτούν θα πρέπει να έχουν ενεργοποιημένη τη ιδιότητα Visible). Στη συνέχεια μπορούμε να πραγματοποιήσουμε σύνδεση και να πετύχουμε ασύρματη επικοινωνία με κάποιες από τις συσκευές που έχει εντοπίσει το NXT μας (με μία όμως τη φορά) αν βέβαια γνωρίζουμε το συνθηματικό τους.

Συνδέσεις (Connections)

Αυτή ή λίστα περιέχει όλες τις συσκευές Bluetooth με τις οποίες το NXT μας έχει συνδεθεί την τρέχουσα στιγμή. Κάθε μονάδα NXT μπορεί να πραγματοποιήσει σύνδεση με τρεις συσκευές Bluetooth κάθε στιγμή (με τη χρήση των γραμμών σύνδεσης 1,2 και 3) και να αποδεχτεί τη σύνδεση από άλλη μία συσκευή (μέσω της γραμμής 0). Ανεξάρτητα όμως από τον αριθμό αυτών των συνδέσεων μπορεί να συνομιλεί -ανταλλάσσει δεδομένα ή προγράμματα- κάθε στιγμή με μία μόνο από αυτές.

Οι Επαφές μου (My Contacts)

Η λίστα αυτή περιλαμβάνει όλες τις συσκευές Bluetooth με τις οποίες το NXT μας έχει πραγματοποιήσει σύνδεση Bluetooth κάποια στιγμή προηγουμένως. Οι συσκευές που βρίσκονται σε αυτή τη λίστα μπορούν να συνδεθούν αυτόματα με το NXT μας χωρίς να απαιτείται να δώσουν ξανά το συνθηματικό για τη σύνδεση.

Η προσθήκη νέων συσκευών Bluetooth στη λίστα αυτή γίνεται με τη χρήση της λειτουργίας αναζήτησης (Search)

Συνθηματικό (Passkey)

Το συνθηματικό εξασφαλίζει ότι μόνο οι συσκευές οι οποίες έχουν την έγκριση από εμάς θα μπορέσουν να συνδεθούν στο NXT μας. Κάθε φορά που το NXT προχωράει στην πραγματοποίηση μία σύνδεσης με μία νέα συσκευή Bluetooth για πρώτη φορά, η συσκευή αυτή θα ζητήσει να πληκτρολογήσουμε το συνθηματικό που έχει οριστεί για να επιτρέψει τη σύνδεση. Με τον ίδιο τρόπο και οι άλλες συσκευές Bluetooth πρέπει να γνωρίζουν το συνθηματικό του NXT μας, για να πραγματοποιήσουν μία σύνδεση με αυτό. Μετά την επαλήθευση του συνθηματικού οι έμπιστες συσκευές τοποθετούνται αυτόματα στη λίστα των επαφών [my contacts]. Για τη σύνδεση με μία από αυτές κάποια άλλη στιγμή, δεν απαιτείται πλέον η χρήση του συνθηματικού.



Στην οθόνη που εμφανίζεται για να πληκτρολογήσουμε το συνθηματικό μας προτείνεται ως εξ' ορισμού τιμή το 1234. Αν θέλουμε να δώσουμε ένα νέο συνθηματικό από την αρχή πατάμε το γκρι πλήκτρο στο NXT για να καθαρίσουμε το συνθηματικό που μας προτείνεται και στη συνέχεια με τα βελάκια βρίσκουμε τους χαρακτήρες που θέλουμε και τους εισάγουμε πατώντας το πορτοκαλί πλήκτρο στο NXT. Αν θέλουμε να διαγράψουμε κάποιο χαρακτήρα που εισάγαμε κατά λάθος πατάμε το γκρι πλήκτρο στο NXT.

Δημιουργία σύνδεσης Bluetooth του NXT με τον προσωπικό υπολογιστή.

Για να προχωρήσουμε σε μία ασύρματη σύνδεση Bluetooth πρέπει ο προσωπικός υπολογιστής μας να υποστηρίζει τη δυνατότητα αυτής της σύνδεσης. Αν δεν υπάρχει ενσωματωμένη θύρα Bluetooth στον υπολογιστή μας μπορεί κανείς να προμηθευτεί ένα USB στικ Bluetooth.

Καταρχήν ανοίγουμε τη μονάδα NXT και αφού ενεργοποιήσουμε τη δυνατότητα σύνδεσης Bluetooth από το μενού [Bluetooth]→[On] επιλέγουμε το NXT μας να είναι ορατό από άλλες συσκευές Bluetooth από το μενού [Bluetooth]→[Visible].

Επίσης σιγουρευόμαστε ότι η δυνατότητα σύνδεσης Bluetooth είναι εγκατεστημένη και ενεργοποιημένη στον προσωπικό μας υπολογιστή.

Στη συνέχεια κάνουμε κλικ στο πλήκτρο NXT Window του πεδίο ελέγχου (Controller) μέσα από το περιβάλλον προγραμματισμού NXT-G. Στο παράθυρο NXT Window που εμφανίζεται κάνουμε κλικ στο πλήκτρο *Scan* και ο υπολογιστής αυτόματα ξεκινάει την ανίχνευση για συσκευές Bluetooth.

Μετά από μια μικρή αναμονή εμφανίζεται μία λίστα με τις συσκευές που εντοπίστηκαν. Επιλέγουμε τη συσκευή με την οποία επιθυμούμε να πραγματοποιήσουμε σύνδεση και κάνουμε κλικ στο πλήκτρο *Connect*.

Αν η σύνδεση με μία συσκευή Bluetooth γίνεται για πρώτη φορά, εμφανίζεται ένα νέο παράθυρο, όπου μας ζητείται να πληκτρολογήσουμε το συνθηματικό (*Passkey*) της σύνδεσης με το NXT (Το ίδιο συνθηματικό πρέπει να έχουμε δώσει και στο NXT μας). Για να συνδεθούμε ασύρματα με μία άλλη συσκευή Bluetooth πρέπει να γνωρίζουμε το συνθηματικό της. Στο παράθυρο NXT window η κατάσταση του NXT εμφανίζεται πλέον αλλαγμένη από διαθέσιμο (*Available*) σε συνδεδεμένο (*Connected*) και έχει αποκατασταθεί ασύρματη σύνδεση ανάμεσα στον προσωπικό μας υπολογιστή και το αντίστοιχο NXT, οπότε και μπορεί να γίνει ανταλλαγή δεδομένων.

Δημιουργία σύνδεσης Bluetooth του NXT με άλλη μονάδα NXT.

Καταρχήν ανοίγουμε τις μονάδες NXT και αφού ενεργοποιήσουμε τη δυνατότητα σύνδεσης Bluetooth από το υπομενού [Bluetooth]→[On] επιλέγουμε σε καθεμία από τις δύο μονάδες NXT να είναι ορατές από άλλες συσκευές Bluetooth από το υπομενού [Bluetooth]→[Visible]. Επίσης φροντίζουμε να έχουμε δώσει διαφορετικά ονόματα σε κάθε μία από τις μονάδες NXT.

Δίνουμε εντολή στο NXT μας να προχωρήσει σε αναζήτηση τυχόν συσκευών Bluetooth που βρίσκονται δίπλα του επιλέγοντας το υπομενού [Bluetooth]→[Search]. Το NXT μας άμεσα θα ανιχνεύσει όλες τις συσκευές Bluetooth που βρίσκονται σε μία ακτίνα 0-30 μέτρα. Ανάλογα με πόσες συσκευές Bluetooth βρίσκονται μέσα σε αυτό το πεδίο δράσης το NXT θα εμφανίσει μία λίστα τους μετά από λίγα δευτερόλεπτα.

Επιλέγουμε τη συσκευή με την οποία επιθυμούμε να πραγματοποιήσουμε σύνδεση και κατόπιν επιλέγουμε τη γραμμή (*line*) με την οποία θα συνομιλεί το NXT μας με τη συσκευή αυτή (1,2 ή 3). Μπορούμε να συνδέσουμε το NXT με τρεις διαφορετικές συσκευές την ίδια στιγμή. Στην περίπτωση μας επιλέγουμε το NXT μας να συνδεθεί με το άλλο NXT που επιθυμούμε.

Αν η σύνδεση με μία συσκευή Bluetooth γίνεται για πρώτη φορά θα απαιτηθεί να πληκτρολογήσουμε στο NXT μας, το συνθηματικό (*Passkey*) της άλλης συσκευής. Αυτό σημαίνει ότι και στις δύο μονάδες NXT χρειάζεται να πληκτρολογήσουμε το ίδιο συνθηματικό για να γίνει εφικτή η σύνδεση τους. Για να συνδεθούμε ασύρματα με μία άλλη συσκευή Bluetooth πρέπει να γνωρίζουμε το συνθηματικό της.

Δημιουργία σύνδεσης Bluetooth του NXT με πάνω από μία μονάδες NXT.

Αν επιθυμούμε να συνδέσουμε το NXT μας με περισσότερες από μία -και μέχρι τρεις- πρέπει να πάμε στο υπομενού των επαφών [Bluetooth]→[My Contacts] αν είχαμε συνδεθεί με αυτή προηγουμένως -όπου βρίσκονται οι συνδέσεις που εμπιστεύεται το NXT μας- ή θα πρέπει να ξεκινήσουμε μία καινούρια αναζήτηση αν πρόκειται για μία νέα συσκευή που φέραμε στο πεδίο δράσης του NXT μας, μόλις τώρα. Το NXT έχει τη δυνατότητα να συνδεθεί με τρεις άλλες συσκευές Bluetooth την ίδια στιγμή. Παρόλα αυτά μπορεί να επικοινωνήσει μόνο με μία συσκευή τη φορά.

Για να στείλουμε ένα αρχείο από το NXT μας σε ένα άλλο, αφού αποκαταστήσουμε τη σύνδεση ανάμεσα τους επιλέγουμε το μενού αρχείων [My Files] και στη συνέχεια το πρόγραμμα που θέλουμε να αποστείλουμε. Στη συνέχεια επιλέγουμε αποστολή [Send] και καθορίζουμε τη συσκευή στη οποία θέλουμε να στείλουμε το αρχείο επιλέγοντας τη γραμμή (*Line*) την οποία έχουμε αποδώσει για την επικοινωνία με τη συγκεκριμένη συσκευή. Η αποστολή ολοκληρώνεται.

Επιλογή NXT Window από το πεδίο ελέγχου (Controller)

Μπορούμε επίσης να δούμε πληροφορίες για την κατάσταση των NXT που βρίσκονται σε σύνδεση Bluetooth ή USB με τον προσωπικό μας υπολογιστή κάνοντας κλικ στο πλήκτρο NXT Window του πεδίου ελέγχου (Controller) μέσα από το περιβάλλον προγραμματισμού NXT-G. Από εδώ όπως είδαμε, μπορούμε να αλλάξουμε το όνομα μιας μονάδας, να ελέγξουμε την κατάσταση της μπαταρίας του, να δούμε τη χρήση της μνήμης του και να σβήσουμε διάφορα αρχεία στο NXT μας.



My Files



NXT Program



View



Bluetooth



Settings



Try Me

- Software
- Files
- NXT Files
 - Run
 - Send
 - Delete

- Setup Instructions
- Programming UI

- Sound dB
- Sound dBA
- Reflected Light
- Ambient Light
- Light Sensor*
- Temperature °C*
- Temperature °F*
- Rotation*
- Motor Rotations
- Motor Degrees
- Touch
- Ultrasonic Inch
- Ultrasonic cm

- Port 1
- Port 2
- Port 3
- Port 4

- My Contacts
 - Select
- Connections
 - 0 1 2 3
- Visibility
 - Visible
 - Invisible
- On/Off
 - On
 - Off
- Search

- Volume
 - Volume UI
- Sleep
 - Sleep UI
- NXT Version
 - Version Info
- Delete Files
 - Software Files
 - NXT Files
 - Sound Files
 - Try Me Files

Try Me
Programs

Για να κατασκευάσετε το όχημα – ρομπότ tankbot δείτε τις οδηγίες συνομολόγησης στο ΠΑΡΑΡΤΗΜΑ ΙΙΙ.



Ενότητα 4η

Προγραμματιστικό περιβάλλον NXT-G

Το προγραμματιστικό περιβάλλον NXT-G

Το NXT-G είναι μια γλώσσα προγραμματισμού που χρησιμοποιούμε για να πούμε στο ρομπότ μας τι να κάνει. Γράφουμε το πρόγραμμα μας στον προσωπικό υπολογιστή και μετά το κατεβάζουμε στον επεξεργαστή NXT μέσω σύνδεσης USB ή Bluetooth. Το NXT-G χρησιμοποιεί εικονίδια για να αναπαραστήσει τις διαφορετικές λειτουργίες, που είναι σε θέση να εκτελέσει ένα ρομπότ εφοδιασμένο με τον επεξεργαστή NXT. Τοποθετώντας αυτά τα εικονίδια με μια συγκεκριμένη σειρά πάνω σε μία δοκό έχουμε το επιθυμητό πρόγραμμα.

Εκκίνηση του προγράμματος

Το πρόγραμμα NXT-G είναι ένα οπτικό περιβάλλον προγραμματισμού στο οποίο οι εντολές έχουν τη μορφή εικονιδίων.

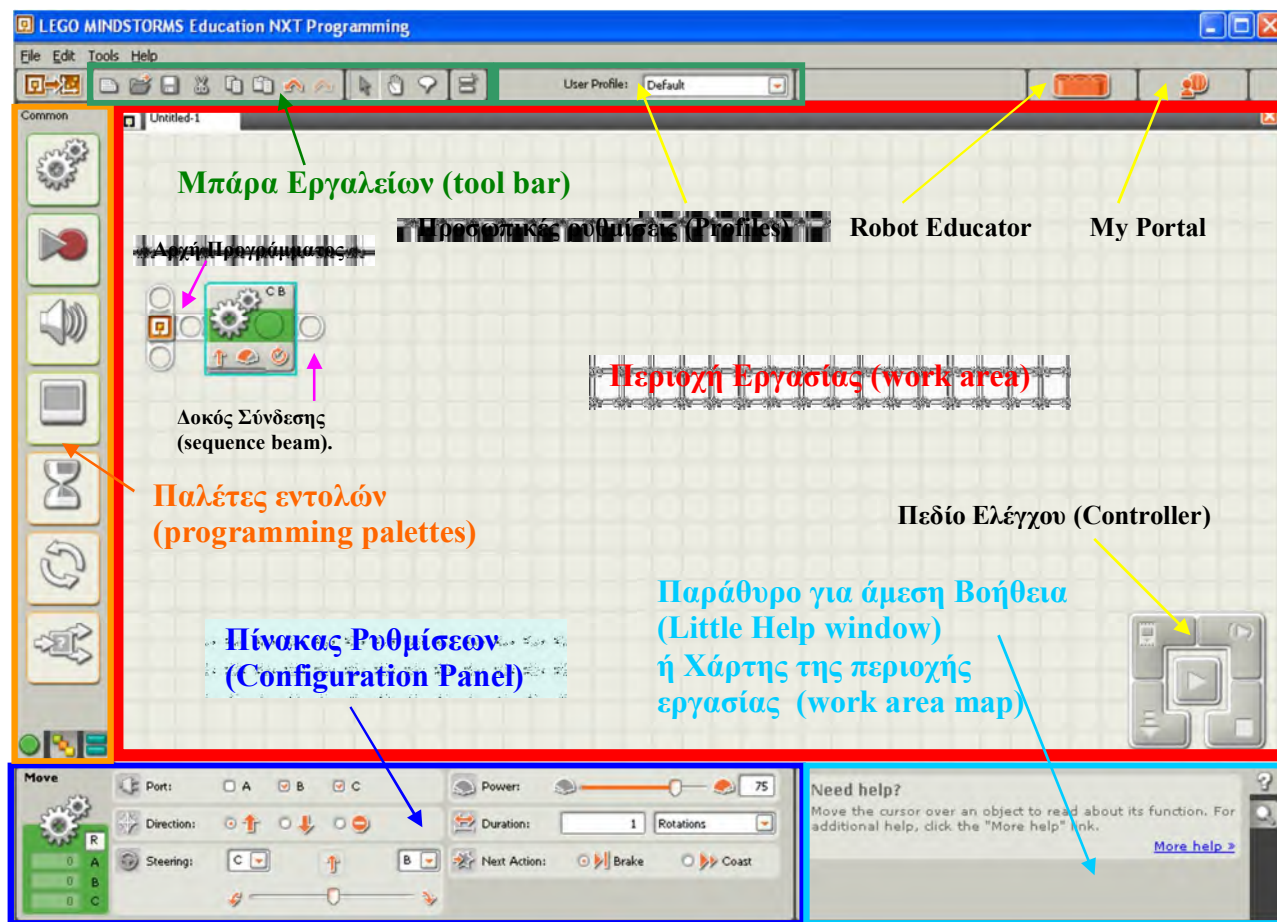
Όταν εκτελέσουμε το πρόγραμμα NXT-G εμφανίζεται η αρχική καρτέλα του προγράμματος :



Για να ξεκινήσουμε τη συγγραφή ενός νέου προγράμματος πληκτρολογούμε ένα όνομα στο πεδίο κειμένου για την εκκίνηση ενός νέου προγράμματος "Start New Program" και κάνουμε κλικ στο κουμπί «Go>>».

Για να ανοίξουμε ένα υπάρχον πρόγραμμα, το οποίο το έχουμε επεξεργαστεί τελευταία, το επιλέγουμε από τη λίστα των πρόσφατα ανοιγμένων προγραμμάτων "Open Recent Program" και κάνουμε κλικ στο κουμπί «Go>>».

Περιγραφή του προγράμματος NXT-G



Robot Educator

Περιέχει οδηγίες βήμα προς βήμα για να κατασκευάσει κανείς κάποιο από τα έτοιμα μοντέλα και να υλοποιήσει μία σειρά από προγραμματιστικές δραστηριότητες.



My Portal

Παρέχει σύνδεση στην ιστοσελίδα www.MINDSTORMSEducation.com από όπου μπορεί κανείς να βρει εργαλεία και σχετικό υλικό για να κατεβάσει και διάφορες πληροφορίες.



Το παράθυρο του Robot Educator ή του My Portal συνήθως πριν ξεκινήσουμε τη συγγραφή ενός προγράμματος το εξαφανίζουμε, ώστε να μεγιστοποιήσουμε το χώρο εργασίας μας. Για να κλείσουμε τα παράθυρα αυτά κάνουμε κλικ στο πλήκτρο με το κόκκινο X πάνω δεξιά. Αν θέλουμε να τα

ξαναεμφανίσουμε κάνουμε κλικ στις καρτέλες



ή





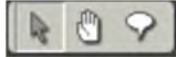
Διακόπτης δειγματοληψίας Δεδομένων (Switch to NXT Data Logging)

Μας δίνει τη δυνατότητα να κάνουμε μια δική μας ερευνητική εργασία ως μικροί επιστήμονες. Μπορούμε να χρησιμοποιήσουμε το NXT να συλλέξει μια σειρά από δεδομένα, τα οποία στη συνέχεια μπορούμε να επεξεργαστούμε κατάλληλα και να καταλήξουμε σε κάποια συμπεράσματα για ένα φυσικό φαινόμενο.

Μπάρα Εργαλείων (tool bar)



Η μπάρα εργασιών είναι παρόμοια με την μπάρα εργασιών που συναντάμε στις συνηθισμένες εφαρμογές των windows. Περιλαμβάνει τις εντολές που χρησιμοποιούμε πιο συχνά.

Στην μπάρα εργαλείων περιλαμβάνεται και η μικρή εργαλειοθήκη  με τα παρακάτω εργαλεία :



Εργαλείο Κατάδειξης (Pointer tool): Χρησιμοποιείτε για να μεταφέρουμε τα εικονίδια εντολών από την παλέτα εντολών στο Παράθυρο Εργασίας καθώς και για να επιλέξουμε και να μετακινήσουμε τα εικονίδια εντολών.



Εργαλείο Ολίσθησης (Pan tool): Χρησιμοποιείτε για να μετακινήσουμε το χώρο εργασίας ώστε να αλλάξουμε το μέρος που θα είναι ορατό στην οθόνη μας.



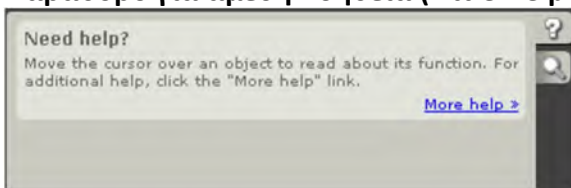
Εργαλείο Κειμένου (Comment tool) : Χρησιμοποιείτε για να προσθέσουμε και να επεξεργαστούμε κείμενο ή επεξηγηματικά σχόλια μέσα στο πρόγραμμα.

Στο σημείο αυτό να αναφέρουμε ότι για να δημιουργήσουμε μία διακλάδωση σε ένα σημείο της συνδετικής δοκού ώστε να δημιουργηθεί στο σημείο αυτό ένας παράλληλος κλάδος κάνουμε κλικ στο σημείο που μας ενδιαφέρει έχοντας πατημένο το πλήκτρο *Shift* του πληκτρολογίου. Στις άλλες περιπτώσεις που θέλουμε να καλωδιώσουμε εντολές μεταξύ τους ή μέσω των σημείων σύνδεσης των κατανεμητών δεδομένων τους το εργαλείο καλωδίωσης  ενεργοποιείται αυτόματα.

Περιοχή Εργασίας (work area)

Είναι η περιοχή στην οποία δημιουργούμε ένα πρόγραμμα μας. Για να υλοποιήσουμε ένα πρόγραμμα για το NXT ρομπότ μας σέρνουμε έναν αριθμό εντολών από την παλέτα εντολών στην περιοχή εργασίας και τις τοποθετούμε κατάλληλα σε μία σειρά πάνω στη δοκό σύνδεσης (sequence beam).

Παράθυρο για άμεση Βοήθεια (Little Help window)



Από εδώ μπορεί κανείς να βρει τις πληροφορίες που χρειάζεται για τη χρήση των εντολών.

Χάρτης της περιοχής εργασίας (work area map)

Κάνοντας κλικ πάνω στο καρτελάκι  στην κάτω δεξιά γωνία στην περιοχή που προβάλλεται η βοήθεια για την τρέχουσα επιλεγμένη εντολή, εμφανίζεται τώρα μία προεπισκόπηση ολόκληρου του προγράμματος που έχει υλοποιήσει κανείς μέχρι τη στιγμή εκείνη. Με τον τρόπο αυτό στην περίπτωση που το πρόγραμμα μας είναι πολύ μεγάλο, μπορούμε να μετακινηθούμε πολύ γρήγορα από το ένα μέρος του σε ένα άλλο. Διαφορετικά η μετακίνηση μέσα στο πρόγραμμα μας μπορεί να γίνει με τη χρήση του εργαλείου pan tool της μπάρας εργαλείων.

Παλέτες εντολών (programming palettes)

Στις παλέτες εντολών βρίσκονται όλες οι διαθέσιμες εντολές που μπορούμε να χρησιμοποιήσουμε για να δημιουργήσουμε κάποιο πρόγραμμα μας. Κάθε εικονίδιο εντολής αντιστοιχεί σε μία σειρά οδηγιών τις οποίες το NXT πρέπει να μεταγλωττίσει πριν να τις εκτελέσει. Ένα πρόγραμμα προκύπτει από κατάλληλο συνδυασμό των μπλοκ αυτών. Όταν ολοκληρώσουμε ένα πρόγραμμα το μεταφορτώνουμε (κατεβάζουμε) στο NXT μας. (Θυμηθείτε ότι προτού μεταφορτώσουμε ένα πρόγραμμα μας, πρέπει πρώτα να ανοίξουμε την τροφοδοσία και να συνδέσουμε το NXT στον υπολογιστή μας).



Τα καρτελάκια (tabs) στο κάτω μέρος μας επιτρέπουν να επιλέξουμε ανάμεσα :

1. στη βασική (common palette) η οποία περιέχει τις εντολές που χρησιμοποιούμε πιο συχνά. Είναι κατάλληλη για την πρώτη επαφή με το προγραμματιστικό αυτό περιβάλλον.
2. στην πλήρη (complete palette) η οποία περιέχει το σύνολο των εντολών ομαδοποιημένων σε κατηγορίες
3. και στην προσαρμοσμένη (custom palette) η οποία περιέχει τις εντολές που μπορεί κανείς να δημιουργήσει ο ίδιος ή να κατεβάσει από το ιντερνέτ.



1. Βασική Παλέτα (Common Palette)

Περιλαμβάνει τις εντολές :



Move block

Με την εντολή αυτή πετυχαίνουμε τη μετακίνηση της ρομποτικής μιας κατασκευής.

Record/Play block

Μας επιτρέπει να προγραμματίσουμε μία σειρά από κινήσεις, μετακινώντας χειροκίνητα το ρομπότ μας και στη συνέχεια να αναπαραγάγουμε τις κινήσεις αυτές, σε κάποιο άλλο σημείο του προγράμματος.

Sound block

Με την εντολή αυτή το NXT μας αναπαράγει διάφορους ήχους στους οποίους συμπεριλαμβάνονται και κάποιες έτοιμες ηχογραφημένες φράσεις.

Display block

Με την εντολή αυτή το NXT μας εμφανίζει στην οθόνη του κάποιο κείμενο που έχουμε εισαγάγει, έτοιμες εικόνες ή σχέδια που έχουμε δημιουργήσει.

Wait block

Με την ομάδα αυτή των εντολών το ρομπότ μας περιμένει μέχρι να λάβει κάποιο συγκεκριμένο ερέθισμα, όπως π.χ. να ανιχνευτεί κάποιος ήχος από τον αισθητήρα ήχου ή να περάσει κάποιο συγκεκριμένο χρονικό διάστημα.

Loop block

Με την ομάδα αυτή των εντολών το ρομπότ μας θα επαναλάβει ξανά και ξανά μία σειρά ενεργειών μέχρι να λάβει κάποιο συγκεκριμένο ερέθισμα, όπως π.χ. να πατηθεί ο διακόπτης του αισθητήρα αφής.

Switch block

Με την ομάδα αυτή των εντολών το ρομπότ μας είναι ικανό να λάβει μία απόφαση μετά από την ενέργεια ενός ελέγχου ανάλογα με το ερέθισμα που θα δεχτεί τη στιγμή εκείνη όπως π.χ. να στρίβει δεξιά αν ανιληφθεί ένα δυνατό ήχο ή αριστερά αν ανιληφθεί ένα απαλό ήχο.



2. Πλήρης Παλέτα (Complete Palette)

Περιλαμβάνει τις κατηγορίες :



Common blocks

Περιλαμβάνει τις εντολές που περιέχονται και στη βασική παλέτα

Action blocks

Μας επιτρέπει να ελέγξουμε μία σειρά από ενέργειες σχετικές με τις συσκευές εξόδου : τον έλεγχο των κινητήρων, την αναπαραγωγή ήχων, την εμφάνιση γραφημάτων στην οθόνη, την αποστολή μηνυμάτων Bluetooth και τον έλεγχο των λαμπτήρων.

Sensor blocks

Μας επιτρέπει να προσδώσουμε στο ρομπότ μας έναν αριθμό από συμπεριφορές σε συνδυασμό με την κατάσταση που ανιχνεύεται από διάφορους αισθητήρες. Με τις εντολές αυτές το ρομπότ μας μπορεί να λάβει πληροφορίες από : τους αισθητήρες αφής, ήχου, φωτός, υπερήχων, περιστροφής, από χρονομετρητές και λήψη μηνυμάτων Bluetooth.

Flow blocks

Με τις εντολές αυτές μπορούμε να δημιουργήσουμε σύνθετες συμπεριφορές για το ρομπότ μας. Περιλαμβάνονται οι εντολές για αναμονή, οι δομές επανάληψης και επιλογής και η εντολή διακοπής.

Data blocks

Περιλαμβάνει μία σειρά από εντολές μαθηματικών πράξεων : αλγεβρικών και λογικών πράξεων, σύγκρισης με μία τιμή ή με κάποιο εύρος τιμών, χρήσης μεταβλητών και παραγωγής τυχαίου αριθμού.

Advanced blocks

Περιλαμβάνει εντολές για μετατροπή χαρακτήρων σε αριθμούς, συνένωση χαρακτήρων διαχείριση αρχείων στη μνήμη του NXT, απενεργοποίηση της κατάστασης αδράνειας, καλιμπράρισμα αισθητήρων, συλλογή δεδομένων και σύνδεση μέσω Bluetooth.



3. Προσαρμοσμένη Παλέτα (Custom Palette)



My Blocks

Εδώ βρίσκονται οι εντολές οι οποίες αποτελούν τη βιβλιοθήκη με τις υπορουτίνες μας.

Web Downloads

Εδώ βρίσκονται οι εντολές που έχουμε κατεβάσει από το ιντερνέτ.

Πίνακας Ρυθμίσεων (Configuration Panel)



Κάθε εντολή που τοποθετούμε στην περιοχή εργασίας διαθέτει το δικό της πίνακα ρυθμίσεων από όπου μπορούμε να προσαρμόσουμε κατάλληλα, ανάλογα με το πρόβλημα που έχουμε να επιλύσουμε, μία σειρά από ιδιότητες και επιλογές. Όταν επιλέξουμε το εικονίδιο μιας εντολής στην περιοχή εργασίας, ακριβώς κάτω από την περιοχή εργασίας εμφανίζεται ο πίνακας με τις ρυθμίσεις που μπορούμε να προσαρμόσουμε στην εντολή αυτή. Αλλάζοντας κάποιες από τις παραμέτρους στον πίνακα ρυθμίσεων μπορούμε να τροποποιήσουμε τη συμπεριφορά που η συγκεκριμένη εντολή θα προσδώσει στο ρομπότ μας. Για παράδειγμα, αλλάζοντας την τιμή στην ισχύ [*power*] στον πίνακα ρυθμίσεων μιας εντολής μετακίνησης (*move*) θα αλλάξουμε και την ταχύτητα με την οποία θα μετακινηθεί το ρομπότ μας.

Πεδίο Ελέγχου (Controller)



Περιλαμβάνει πέντε τα πλήκτρα με τα οποία μπορούμε να επικοινωνήσουμε με ένα τούβλο NXT από τον προσωπικό μας υπολογιστή.

Με τα τέσσερα από αυτά μπορούμε να μεταφορτώσουμε ένα πρόγραμμα ή μέρος αυτού στο NXT :

Με το πλήκτρο μεταφόρτωσης (download button) μεταφορτώνουμε (κατεβάζουμε) το πρόγραμμα μας στο NXT. Στη συνέχεια μπορούμε να προχωρήσουμε στην εκτέλεση του κατευθείαν από το τούβλο NXT. Με το πλήκτρο μεταφόρτωσης και εκτέλεσης (download and run button) μεταφορτώνουμε (κατεβάζουμε) το πρόγραμμα μας στο NXT το οποίο και εκτελείται αυτόματα μόλις ολοκληρωθεί το κατέβασμα του. Πρέπει να είμαστε προσεκτικοί με τη χρήση αυτής της επιλογής ώστε να μη συμβεί κάποιο ατύχημα στο NXT, όπως στην περίπτωση που η ρομποτική μας κατασκευή βρίσκεται πάνω σε ένα τραπέζι και αρχίσει να κινείται με αποτέλεσμα να πέσει από αυτό.

Με το πλήκτρο μεταφόρτωσης και εκτέλεσης ενός τμήματος (download and run selected button) έχουμε τη δυνατότητα να μεταφορτώσουμε (κατεβάσουμε) ένα τμήμα του προγράμματος μας, -το οποίο και έχουμε προηγουμένως επιλέξει- το οποίο και θα εκτελεστεί αυτόματα μόλις ολοκληρωθεί το κατέβασμα του. Την επιλογή αυτή τη χρησιμοποιούμε για να κάνουμε δοκιμές για την ορθή λειτουργία του προγράμματος μας και την ανίχνευση τυχόν λογικών σφαλμάτων.

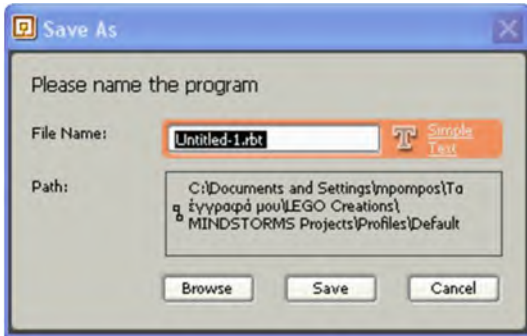
Με το πλήκτρο διακοπής (stop button) μπορούμε να σταματήσουμε την εκτέλεση ενός προγράμματος στο NXT.

Από το πλήκτρο "NXT window" μπορούμε να αλλάξουμε τις ρυθμίσεις του μικροϋπολογιστή NXT. Επιλέγοντας το πλήκτρο αυτό εμφανίζεται ένα παράθυρο από το οποίο μπορούμε να δούμε πληροφορίες για το NXT, τη μνήμη και τον τρόπο σύνδεσης του με τον υπολογιστή.

Αποθήκευση των αρχείων

Η αποθήκευση μιας εργασίας μας γίνεται μέσα από την επιλογή *Save* του μενού *File* του προγραμματιστικού περιβάλλοντος NXT-G.

Τα προγράμματα που δημιουργούμε με το λογισμικό NXT-G αποθηκεύονται στο φάκελο "...\\My Documents\\LEGO Creations\\MINDSTORMS Projects\\profiles\\default" ως αρχεία μορφής .rbt



Προσωπικές ρυθμίσεις (Profiles)

Στην περίπτωση που λογισμικό NXT-G χρησιμοποιείται και από άλλους χρήστες τότε μπορούμε να επιλέξουμε τα αρχεία μας να αποθηκεύονται σε ξεχωριστούς φακέλους για κάθε χρήστη, συμπεριλαμβανομένων και κάποιων προσωπικών μας ρυθμίσεων ως προς την εμφάνιση του προγράμματος.

Η δημιουργία ενός προσωπικού μας προφίλ γίνεται μέσα από την επιλογή *Manage Profiles* του μενού *Edit* του προγραμματιστικού περιβάλλοντος NXT-G.

Με την κατασκευή του προσωπικού μας προφίλ δημιουργείται αυτόματα ένας φάκελος με το όνομα που δηλώσαμε στην περιοχή :

"...\\My Documents\\LEGO Creations\\MINDSTORMS Projects\\profiles\\".

Ο φάκελος αυτός περιέχει τα αρχεία προγράμματος που έχουμε δημιουργήσει (αρχεία .rbt), τα δικά μας blocks και το αρχείο settings.ini με τις προσωπικές ρυθμίσεις.

Για να προσθέσουμε νέους ήχους αντιγράφουμε τα αρχεία των ήχων (.rso) στο φάκελο C:\\Program Files\\LEGO Software\\LEGO MINDSTORMS NXT\\engine\\Sounds

Για να προσθέσουμε νέες εικόνες αντιγράφουμε τα αρχεία των εικόνων (.ric) στο φάκελο C:\\Program Files\\LEGO Software\\LEGO MINDSTORMS NXT\\engine\\Pictures

Για να προσθέσουμε νέα Blocks αντιγράφουμε τα αρχεία των blocks στο φάκελο

My Documents/LEGO Creations/MINDSTORMS/Projects/Profiles/Default/Blocks/ My Blocks

Φόρτωση ή αναβάθμιση Firmware στο μικροϋπολογιστή NXT

Για να φορτώσουμε ένα νέο Firmware στο μικροϋπολογιστή NXT κάνουμε κλικ στην επιλογή *Update NXT Firmware* από το μενού *Tools* του προγράμματος NXT-G.

Μια παρουσίαση για το προγραμματιστικό περιβάλλον NXT-G θα βρείτε στο σύνδεσμο :

[Προγραμματιστικό περιβάλλον NXT-G](#)

Εξοικείωση με το NXT-G

Το NXT-G είναι ένα προγραμματιστικό περιβάλλον το οποίο χρησιμοποιούμε για να πούμε στο ρομπότ μας τι να κάνει.

Γράφουμε το πρόγραμμα μας στον προσωπικό υπολογιστή μας και μετά το κατεβάζουμε (κάνουμε downloading) στο NXT του ρομπότ μας χρησιμοποιώντας καλωδίωση USB ή εναλλακτικά ασύρματη σύνδεση Bluetooth

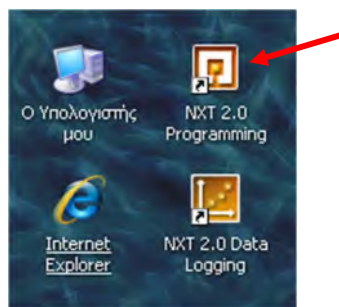
Το NXT-G χρησιμοποιεί εικονίδια (*blocks*) τα οποία αναπαριστούν διαφορετικές εργασίες που πρέπει να εκτελέσει το ρομπότ μας. Για να δημιουργήσουμε μια λογική σειρά από εντολές που θα εκτελέσει το ρομπότ μας, βάζουμε τα εικονίδια αυτά σε μία συγκεκριμένη σειρά και τα συνδέουμε μεταξύ τους.

Στις παρακάτω δραστηριότητες θα χρησιμοποιήσουμε το όχημα - ρομπότ NXTbot, που έχουμε ήδη κατασκευάσει.

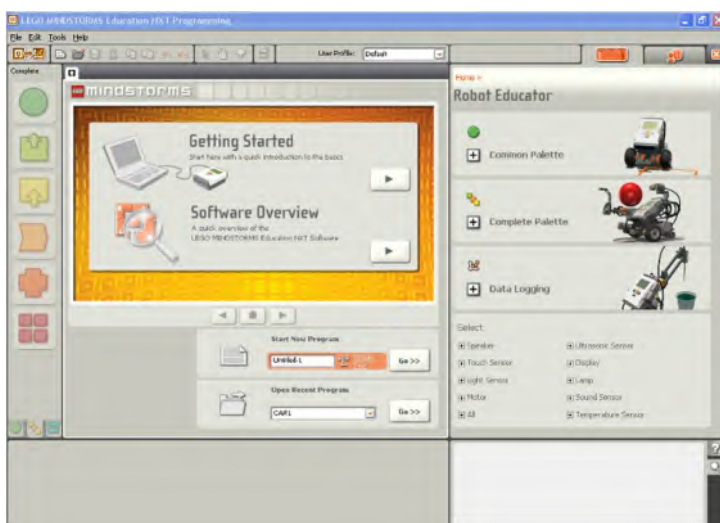
Το όχημα αυτό διαθέτει δύο κινητήρες :

Ο αριστερός κινητήρας κινεί την αριστερή ρόδα και είναι συνδεδεμένος στη θύρα εξόδου B του NXT, ενώ ο δεξιός κινητήρας κινεί τη δεξιά ρόδα και είναι συνδεδεμένος στη θύρα εξόδου C.

• Ανοίξτε την εφαρμογή NXT-G κάνοντας διπλό κλικ πάνω στη συντόμευση του Lego Mindstorms NXT



Θα ξεκινήσει η εφαρμογή και θα εμφανιστεί το παρακάτω παράθυρο :

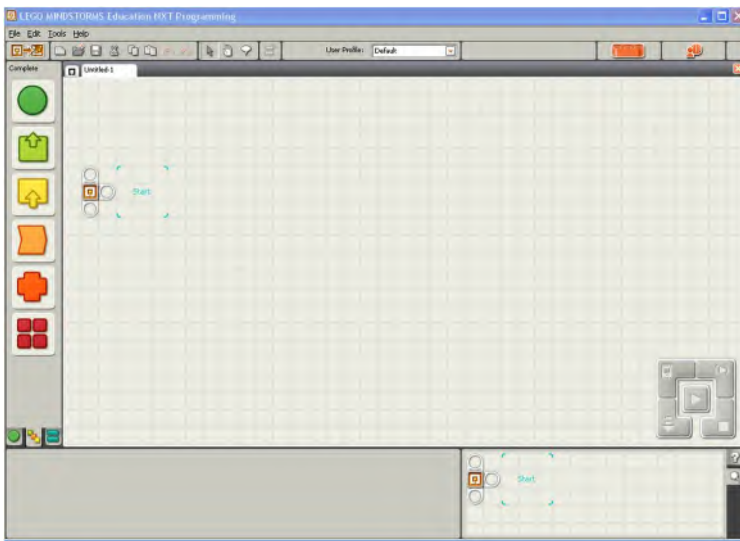


Σενάρια Ρομποτικής με το Mindstorms NXT

- Πληκτρολογήστε ένα όνομα για τη νέα εργασία σας στο πεδίο κειμένου για την εκκίνηση ενός νέου προγράμματος *Start New Program* και κάντε κλικ στο κουμπί *Go>>*.



Για να κατασκευάσουμε ένα νέο πρόγραμμα πιάνουμε με το ποντίκι ένα εικονίδιο από την παλέτα εντολών στη δεξιά πλευρά της εφαρμογής και το μετακινούμε στην περιοχή εργασίας (ή παράθυρο προγραμματισμού), όπου και το αφήνουμε.



Για τις δραστηριότητες που ακολουθούν θα χρησιμοποιήσουμε την πλήρη παλέτα εντολών.

- Από την περιοχή με τις παλέτες εντολών στο αριστερό μέρος της εφαρμογής κάντε κλικ στο μεσαίο από τα τρία καρτελάκια στο κάτω μέρος της.

Επίσης θα κλείσουμε το παράθυρο του Robot Educator στο δεξιό μέρος της εφαρμογής για να εκμεταλλευτούμε ολόκληρο το διαθέσιμο χώρο εργασίας.

- Κλείστε το παράθυρο του Robot Educator.

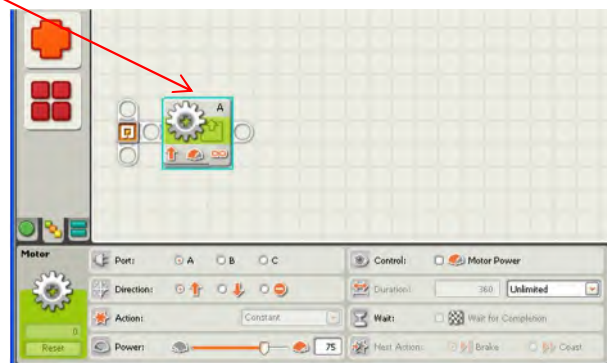
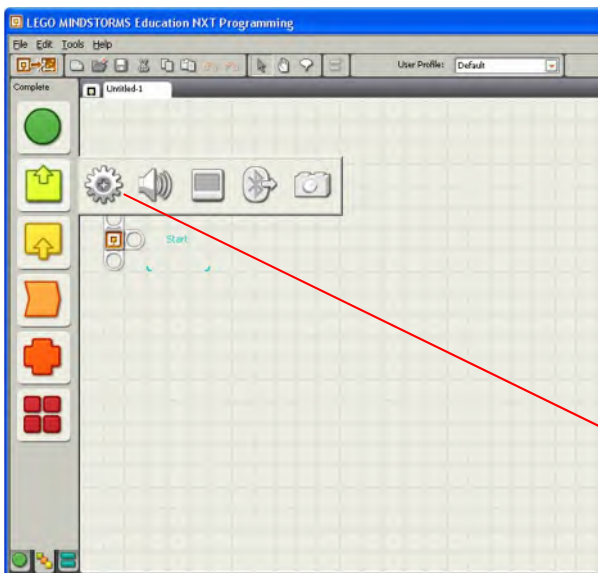
Δραστηριότητα 1η : κίνηση μπροστά για 2 περιστροφές

Στη δραστηριότητα αυτή θα χρησιμοποιήσουμε το ρομπότ nxtbot που ήδη έχουμε κατασκευάσει. Προσπαθήστε να καταλάβετε τι αποτέλεσμα θα έχει το παρακάτω πρόγραμμα:



Ας ξεκινήσουμε δημιουργώντας ένα απλό πρόγραμμα. Αυτό το πρόγραμμα θα μετακινήσει το ρομπότ μας προς τα μπροστά για 2 περιστροφές των κινητήρων.

- Από το υπομενού για ενέργειες **Action** (έχει ως σχήμα ένα τετράγωνο με ένα βελάκι προς τα επάνω και πράσινο ανοιχτό χρώμα) επιλέξτε το εικονίδιο για την εντολή κίνησης κινητήρα, το οποίο έχει ως σχήμα ένα γρανάζι. Για να τοποθετήσετε το εικονίδιο αυτό μέσα στο πρόγραμμά μας κάντε κλικ πάνω του, κρατήστε πατημένο το πλήκτρο του ποντικιού, σύρετε το εικονίδιο στο χώρο εργασίας της εφαρμογής μας και τοποθετήστε το στο σημείο εκκίνησης (το οποίο εμφανίζεται με τη σήμανση ενός μπλε τετραγώνου εντός του οποίου περιέχεται η λέξη start) δίπλα στο σύμβολο της εκκίνησης προγράμματος.



Σενάρια Ρομποτικής με το Mindstorms NXT

- Σε αυτό το σημείο ελέγξτε αν η εντολή κούμπωσε σωστά πάνω στη λευκή συνδετική δοκό. Στην αντίθετη περίπτωση η εντολή θα εμφανίζεται αχνά. Αν δεν έχει κουμπώσει σωστά, μετακινήστε την εντολή οπουδήποτε αλλού στην περιοχή εργασίας και τοποθετείστε την ξανά πάνω στη συνδετική δοκό.

Μόλις το εικονίδιο της εντολής τοποθετηθεί στην περιοχή εργασίας, στο κάτω δεξιό μέρος της εφαρμογής μας, εμφανίζεται ο πίνακας ρυθμίσεων της εντολής. Από εδώ μπορούμε να τροποποιήσουμε την κατεύθυνση, την ταχύτητα και τη διάρκεια της κίνησης για τον κινητήρα του ρομπότ μας.

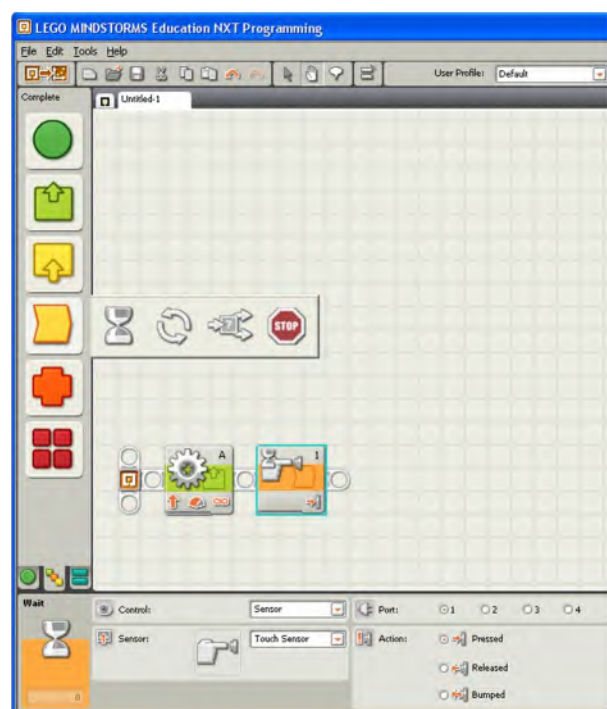


Στο σημείο αυτό μπορείτε να αλλάξετε τις παραμέτρους του πίνακα ρυθμίσεων και να πειραματιστείτε με τους διαφορετικούς τρόπους που μπορούμε να επιβάλουμε στον κινητήρα να κινηθεί.

- Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως Θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη θύρα C αντί για την A. Μην πειράξετε καμία άλλη ρύθμιση.

Με την εντολή αυτή ο κινητήρας μας θα ξεκινήσει να λειτουργεί και θα σταματήσει μόνο όταν το πρόγραμμα μας φτάσει στο τέλος του. Εμείς όμως θέλουμε ο κινητήρας να κινηθεί για ένα συγκεκριμένο χρονικό διάστημα. Για να το πετύχουμε αυτό πρέπει να τοποθετήσουμε μία εντολή αναμονής.

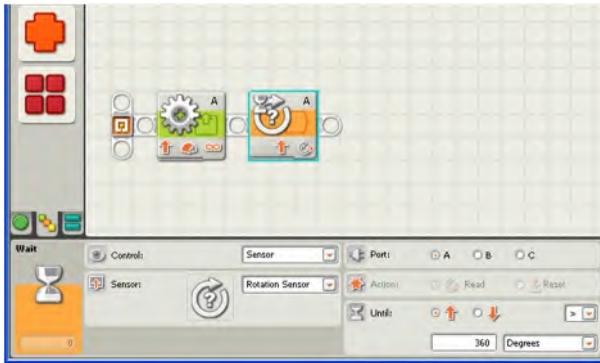
- Ανοίξτε την υποπαλέτα ροής **flow** (έχει ως σχήμα ένα τσακισμένο τετράγωνο και πορτοκαλί χρώμα) και επιλέξτε το εικονίδιο για την εντολή αναμονής, το οποίο έχει ως σχήμα μία κλεψύδρα. Σύρετε το στην περιοχή εργασίας και τοποθετείστε το στο σημείο αμέσως μετά την εντολή κίνησης κινητήρα, ακριβώς στα δεξιά της.



Σενάρια Ρομποτικής με το Mindstorms NXT

Οι κινητήρες που συνοδεύουν το πακέτο Lego Mindstorms NXT διαθέτουν έναν ενσωματωμένο αισθητήρα περιστροφής, ο οποίος μας προμηθεύει με πληροφορίες για την περιστροφή που εκτελεί κάθε στιγμή ένας κινητήρας. Μπορούμε να αξιοποιήσουμε τις πληροφορίες αυτές στην περίπτωση που θέλουμε να ελέγξουμε τον κινητήρα, ώστε να περιστραφεί όσο ακριβώς θέλουμε.

- Στον πίνακα ρυθμίσεων αλλάζτε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα περιστροφής "Rotation Sensor" από αισθητήρα αφής "Touch Sensor".



Αφού ο κινητήρας μας είναι συνδεδεμένος στη θύρα C την πληροφορία για την περιστροφή του, πρέπει να την πάρουμε από τον αισθητήρα περιστροφής του συγκεκριμένου κινητήρα.

- Στον πίνακα ρυθμίσεων επιλέξτε ως θύρα (Port) που θα παρακολουθείται από τον αισθητήρα περιστροφής, τη θύρα C αντί για την A. Αλλάξτε επίσης τον αριθμό των μοιρών που θέλουμε να περιστραφεί ο κινητήρας μας σε 720 (2 περιστροφές). Μην πειράξετε καμία άλλη ρύθμιση.



Τώρα μπορούμε αν θέλουμε, να γράψουμε και μερικές σημειώσεις πάνω από τις εντολές μας για να θυμόμαστε τι ακριβώς κάνουν. Οι σημειώσεις μας βοηθούν να θυμηθούμε τι ακριβώς κάνει το πρόγραμμα μας όταν θέλουμε να το χρησιμοποιήσουμε μετά από την πάροδο κάποιου χρονικού διαστήματος.

- Κάντε διπλό κλικ ακριβώς πάνω από την εντολή αναμονής για περιστροφή και πληκτρολογήστε την πρόταση <<περιστροφή 720 μοίρες>>.

Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το ρομπότ μας θα εκτελέσει μία στροφή προς τα δεξιά για 2 περιστροφές.

Σενάρια Ρομποτικής με το Mindstorms NXT

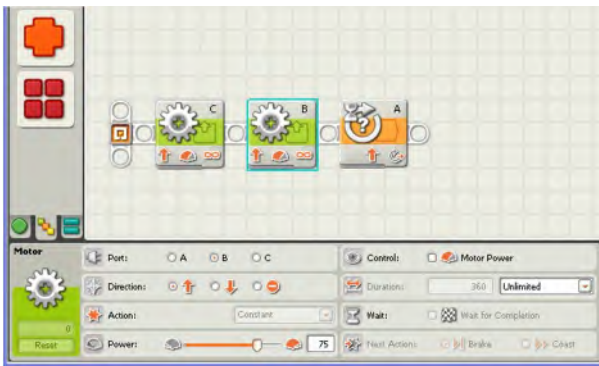
Σημείωση: Αν δε βάλουμε την εντολή της αναμονής ο κινητήρας δε θα γυρίσει καθόλου, αφού μόλις ξεκινούσε θα σταματούσε ακαριαία με το πέρας του προγράμματος μας, χωρίς στην ουσία να έχει προλάβει καν να ξεκινήσει.

Για να κινηθεί το ρομπότ μας σε ευθεία γραμμή μπροστά θα πρέπει να κινήσουμε και το δεύτερο κινητήρα μας, ο οποίος είναι συνδεδεμένος στη θύρα B, με τον ίδιο τρόπο όπως και τον πρώτο.

• Από το υπομενού για ενέργειες **Action** επιλέξτε το εικονίδιο για την εντολή κίνησης κινητήρα και τοποθετήστε το αμέσως μετά την πρώτη εντολή για τον κινητήρα C και πριν την εντολή αναμονής για περιστροφή. Παρατηρήστε ότι η εντολή αναμονής αυτόματα απομακρύνεται προς τα δεξιά για να δημιουργηθεί ο απαραίτητος χώρος.



Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη θύρα B αντί για την A.



Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το ρομπότ μας θα κινηθεί ευθεία προς τα μπροστά για 2 τουλάχιστον περιστροφές.

Σημείωση: Ο κινητήρας μόλις συμπληρωθούν οι δύο περιστροφές δε θα σταματήσουν ακαριαία αλλά θα συνεχίσουν να κινούνται για ένα πολύ μικρό χρονικό διάστημα. Το πρόγραμμα αυτό θέτει σε λειτουργία τον κινητήρα C και αμέσως μετά και τον κινητήρα B και στη συνέχεια περιμένει μέχρι ο κινητήρας C να συμπληρώσει δύο περιστροφές. Μόλις συμβεί αυτό δεν υπάρχει καμία εντολή για να πει στους κινητήρες να σταματήσουν να λειτουργούν προτού το πρόγραμμα τερματιστεί. Με τον τερματισμό ενός προγράμματος διακόπτεται η τροφοδοσία σε όλες τις θύρες εξόδου του NXT οπότε οι κινητήρες σταματάνε αλλά όχι ακαριαία όπως θα θέλαμε, παρά μόνο μετά από ολίσθηση μέχρι να σταματήσουν εντελώς λόγω της τριβής. Έτσι αν παρατηρήσετε πιο προσεκτικά το ρομπότ, θα

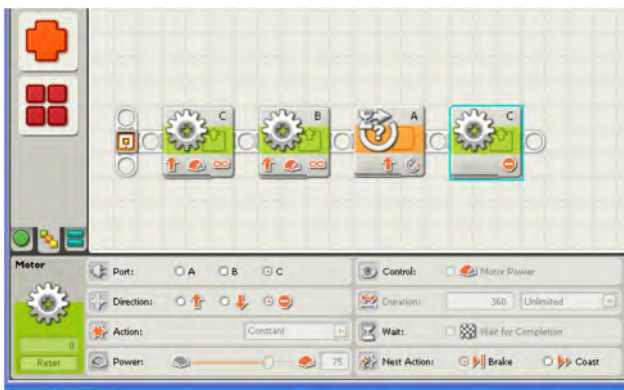
Σενάρια Ρομποτικής με το Mindstorms NXT

διαπιστώσετε ότι αυτό θα συνεχίσει να κινείται για λίγο ακόμη αφού ολοκληρωθούν οι δύο περιστροφές που θέσαμε με την εντολή της αναμονής, μέχρι να σταματήσει εντελώς λόγω της τριβής.

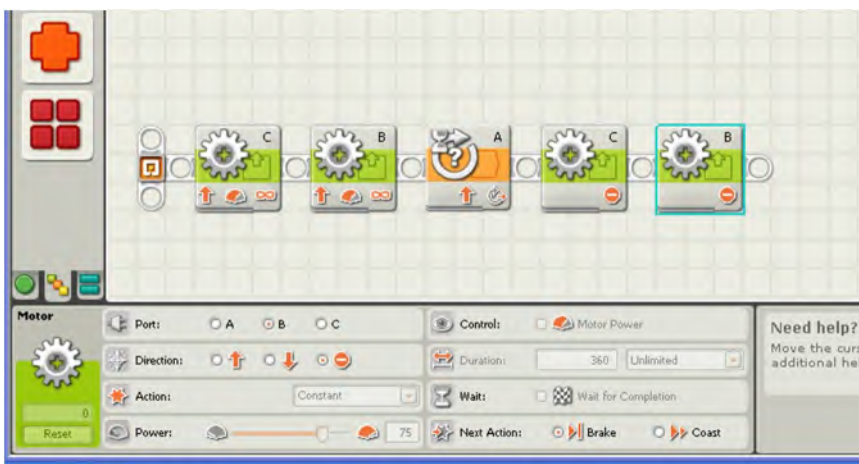
Αν θέλουμε το ρομπότ μας να κινηθεί για 720° ακριβώς, θα πρέπει να πούμε στο πρόγραμμα αμέσως μετά εντολή της αναμονής για 2 περιστροφές, να φρενάρει τους κινητήρες.

Θυμηθείτε ότι τα ρομπότ εκτελούν αυστηρά τις οδηγίες που τους δίνουμε και όχι αυτό που θέλουμε και επιθυμούμε.

- Από την υποπαλέτα για ενέργειες **Action** επιλέξτε την εντολή κίνησης κινητήρα και σύρετε την στην περιοχή εργασίας, αμέσως μετά την τελευταία εντολή του προγράμματος. Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη θύρα C αντί για την A και ως κατεύθυνση κίνησης (Direction) επιλέξτε το σήμα του στοπ.



- Επαναλάβετε την ίδια διαδικασία ακόμη μία φορά. Στον πίνακα ρυθμίσεων αυτή τη φορά επιλέξτε, ως θύρα τη B.



Το πρόγραμμά σας πρέπει τώρα να είναι ίδιο με το παραπάνω.

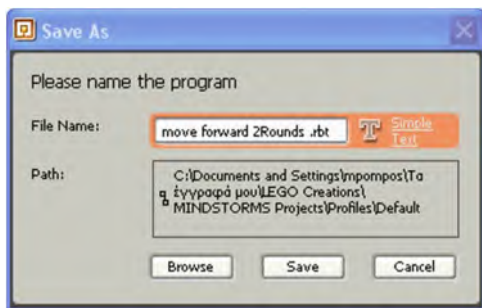
Αποθήκευση του προγράμματος

Δεν πρέπει να ξεχνάτε να αποθηκεύουμε την εργασία που κάνουμε, ανά τακτά χρονικά διαστήματα και πάντοτε πριν να στείλουμε ένα πρόγραμμα στο NXT για να το εκτελέσει.

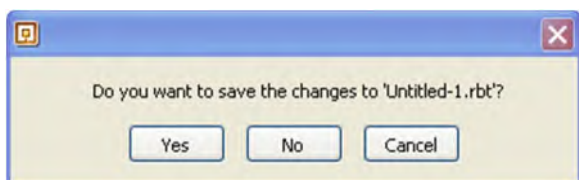
• Για να αποθηκεύσετε το πρόγραμμα που δημιουργήσατε επιλέξτε το *Save As...* από το μενού *File* του παραθύρου σχεδίασης.

Στο παράθυρο που εμφανίζεται πληκτρολογήστε την πρόταση **κίνηση μπροστά για 2 περιστροφές** ως όνομα για την εργασία μας, καθορίστε το φάκελο **nxt programs** της επιφάνειας εργασίας ως θέση αποθήκευσης και πατήστε το κουμπί *Αποθήκευση*.

Στην καρτέλα για την αποθήκευση προτείνεται από την εφαρμογή η αποθήκευση των προγραμμάτων μας στο φάκελο που έχει δημιουργηθεί για το λόγο αυτό, από την εφαρμογή κατά την εγκατάσταση της. Για να μετακινηθείτε λοιπόν στο φάκελο **nxt programs** πατήστε το κουμπί *Browse*.



Προσοχή : Όταν κλείσουμε το παράθυρο του προγραμματιστικού περιβάλλοντος ενώ προηγουμένως έχουμε προχωρήσει σε κάποιες αλλαγές στην εργασία μας, εμφανίζεται το παρακάτω παράθυρο το οποίο μας προτρέπει να σώσουμε τις αλλαγές που έχουμε κάνει, στο πρόγραμμα που επεξεργαζόμαστε.



Κατέβασμα του προγράμματος στο NXT και εκτέλεση του

Για να κατεβάσουμε ένα πρόγραμμα στο ρομπότ μας πρέπει πρώτα να σιγουρευτούμε ότι το τούβλο NXT είναι σε λειτουργία και συνδεδεμένο με τον Η/Υ μας.

• Συνδέστε το τούβλο NXT με τον προσωπικό σας υπολογιστή χρησιμοποιώντας το καλώδιο USB. Εναλλακτικά μπορούμε να πραγματοποιήσουμε μία ασύρματη σύνδεση Bluetooth ανάμεσα στις δύο συσκευές.

Πατήστε το πορτοκαλί πλήκτρο **enter** στο NXT για να το θέσετε σε λειτουργία.



Σενάρια Ρομποτικής με το Mindstorms NXT

• Πατήστε το κουμπί Download στο πεδίο ελέγχου, που βρίσκεται στην κάτω δεξιά γωνία της περιοχής εργασίας, για να κατεβάσετε το πρόγραμμα που δημιουργήσατε στο NXT.
Το πρόγραμμα θα αρχίσει να μεταφέρεται και όταν το πρόγραμμα σας φορτωθεί επιτυχώς στο NXT, αυτό θα σφυρίξει.

Όταν κατεβάσετε ένα πρόγραμμα σας στο NXT, αυτό αποθηκεύεται ως αρχείο στη μνήμη του με το όνομα που του έχουμε δώσει. Για να εκτελέσουμε το πρόγραμμα αυτό στο ρομπότ μας, πρέπει να πάμε στο φάκελο *Software Files* και να βρούμε το συγκεκριμένο αρχείο.



- Από το κυρίως μενού στο τούβλο NXT πατήστε το πορτοκαλί πλήκτρο για να μεταβείτε στο μενού *My Files* και ακόμη μία φορά για να μπούμε στο μενού *Software Files*. Βρείτε το πρόγραμμα σας - αν αυτό δεν εμφανίζεται ήδη στην οθόνη - με τη βοήθεια των πλήκτρων αριστερό και δεξιό βελάκι.
- Όταν το πρόγραμμα σας εμφανιστεί στην οθόνη του NXT πατήστε μία φορά το πορτοκαλί πλήκτρο *enter* για να το επιλέξετε, και ακόμη μία φορά για να το τρέξετε.

Προσοχή : πριν τρέξετε το πρόγραμμα σας, αν είχατε συνδέσει το NXT στον προσωπικό σας υπολογιστή με το καλώδιο USB, μη ξεχάσετε να αποσυνδέσετε την άκρη του καλωδίου αυτού από το NXT του ρομπότ σας.

- Τρέξτε το πρόγραμμα από το NXT πατώντας δύο φορές το πορτοκαλί πλήκτρο *enter*.

Σημείωση : Στην περίπτωση που αφαιρέσουμε από το παραπάνω πρόγραμμα την εντολή αναμονής για περιστροφή, το όχημα ρομπότ μας δε θα κουνηθεί καθόλου από τη θέση του, αφού μόλις μπουν οι κινητήρες σε λειτουργία την ίδια στιγμή και θα σταματήσουν



Η εντολή για αναμονή είναι αυτή που ελέγχει το χρονικό διάστημα που θα κινηθούν οι κινητήρες προτού σταματήσει η λειτουργία τους. Με την εντολή αναμονής το ρομπότ μας θα περάσει στην επόμενη εντολή μετά το χρονικό διάστημα που ορίζεται από την εντολή αναμονής. Μέχρι τότε θα συνεχίσει να περιστρέφει τους κινητήρες του. Αν δεν υπάρχει καμία εντολή για αναμονή θα θέσει σε λειτουργία τους κινητήρες και αμέσως μετά θα τους σταματήσει πριν καν προλάβουν αυτοί να ξεκινήσουν.

Συγχαρητήρια !!!

Ολοκληρώσατε την 1η δραστηριότητα

Δραστηριότητα 2η : κίνηση προς τα πίσω για 2 περιστροφές

(τροποποίηση προγράμματος)

Για να ξεκινήσουμε τη συγγραφή ενός νέου προγράμματος υπάρχουν όπως είναι γνωστό δύο επιλογές :

1. Να μετονομάσουμε και μετά να επεξεργαστούμε ένα υπάρχον πρόγραμμα.
2. Να ξεκινήσουμε με ένα άδειο φύλλο και να κάνουμε ένα νέο πρόγραμμα από την αρχή.

Στη δραστηριότητα θα τροποποιήσουμε το προηγούμενο πρόγραμμα ώστε το ρομπότ μας να κινηθεί προς τα πίσω για 2 περιστροφές.

Άνοιγμα του προγράμματος

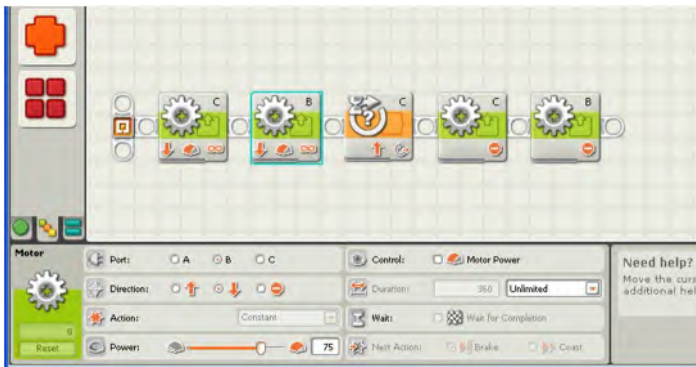
Για να ανοίξουμε ένα υπάρχον πρόγραμμα που έχουμε δημιουργήσει, επιλέγουμε το *Open...* από το μενού *File* της εφαρμογής.

- Στο παράθυρο που εμφανίζεται καθορίστε το φάκελο **nxt programs**, επιλέξτε το αρχείο **κίνηση μπροστά για 2 περιστροφές** και πατήστε το κουμπί *ok*.

Τροποποίηση του προγράμματος

Για να κινηθεί το όχημα-ρομπότ μας προς τα πίσω θα πρέπει να αλλάξουμε τη φορά περιστροφής στις εντολές για την κίνηση των δύο κινητήρων του.

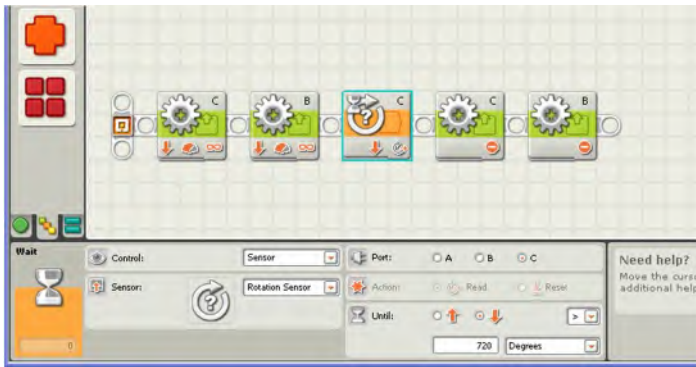
- Επιλέξτε την εντολή του κινητήρα C και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (*Direction*) επιλέγοντας το βελάκι που δείχνει προς τα κάτω. Κάντε το ίδιο και με την εντολή του κινητήρα B.



Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το ρομπότ μας θα κινηθεί προς τα πίσω αλλά δε θα σταματήσει να κινείται μετά από τη συμπλήρωση 2 περιστροφών. Αυτό θα συμβεί επειδή η εντολή της αναμονής για περιστροφή, έτσι όπως τη ρυθμίσαμε προηγουμένως, θα περιμένει μέχρι το ρομπότ μας να ολοκληρώσει δύο περιστροφές προς τα μπροστά, πράγμα που δεν πρόκειται να συμβεί ποτέ αφού τώρα έχουμε βάλει το ρομπότ μας να κινείται προς τα πίσω. Άρα για να σταματήσει το ρομπότ μας μετά από δύο περιστροφές θα πρέπει να αλλάξουμε και τη φορά κίνησης που θα παρακολουθείται από την εντολή αναμονής για περιστροφή.

Σενάρια Ρομποτικής με το Mindstorms NXT

- Επιλέξτε την εντολή αναμονής για περιστροφή του κινητήρα C και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (*Until*) επιλέγοντας το βελάκι που δείχνει προς τα κάτω.



- Αποθηκεύστε το πρόγραμμά σας ως **κίνηση πίσω για 2 περιστροφές**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Συγχαρητήρια !!!

Ολοκληρώσατε την 2η δραστηριότητα

Δραστηριότητα 3η : κίνηση μπροστά για 2 περιστροφές και επιστροφή

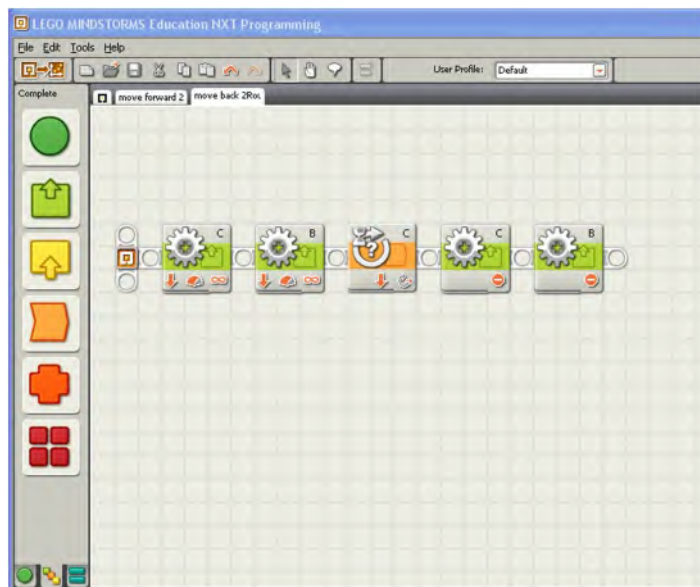
(επέκταση προγράμματος)

Θα τροποποιήσουμε το προηγούμενο πρόγραμμα ώστε το ρομπότ μας να κινηθεί αρχικά προς τα μπροστά για 2 περιστροφές και στη συνέχεια να κινηθεί προς τα πίσω για 2 περιστροφές, ώστε να επιστρέψει στο σημείο από το οποίο ξεκίνησε.

Το πρόγραμμα αυτό αποτελείται από δύο μέρη. Το πρώτο μέρος (κίνηση του οχήματος-ρομπότ προς τα εμπρός για 2 περιστροφές) το έχουμε αναπτύξει ήδη στην 1η δραστηριότητα. Το δεύτερο μέρος (Κίνηση του οχήματος-ρομπότ προς τα πίσω για 2 περιστροφές) το έχουμε αναπτύξει στην προηγούμενη εργασία μας. Αρκεί λοιπόν να ανοίξουμε το πρόγραμμα που δημιουργήσαμε στην 1η δραστηριότητα και να το συμπληρώσουμε με το πρόγραμμα που είχαμε δημιουργήσει προηγουμένως στην 2η δραστηριότητα.

Άνοιγμα του προγράμματος

- Ανοίξτε το πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** που δημιουργήσατε στην 1^η δραστηριότητα.
- Ανοίξτε επίσης και το πρόγραμμα **κίνηση πίσω για 2 περιστροφές** που δημιουργήσατε προηγουμένως.



Βλέπουμε ότι το λογισμικό NXT-G ανοίγει μία καινούργια καρτέλα για κάθε πρόγραμμα που επεξεργαζόμαστε. Με αυτόν τον τρόπο μπορούμε να έχουμε ανοιχτά πολλαπλά προγράμματα την ίδια στιγμή.

Τροποποίηση του προγράμματος

Για να δημιουργήσουμε το νέο μας πρόγραμμα θα επικολλήσουμε τις εντολές του προγράμματος **κίνηση πίσω για 2 περιστροφές** στο τέλος του προγράμματος **κίνηση μπροστά για 2 περιστροφές**.

- Πατήστε και κρατήστε πατημένο το αριστερό πλήκτρο του ποντικιού σε ένα σημείο της περιοχής εργασίας, πάνω και αριστερά από το πρώτο εικονίδιο του προγράμματος **κίνηση πίσω για 2 περιστροφές**. Τώρα χωρίς να αφήσετε το πλήκτρο του ποντικιού σύρετε το ποντίκι μέχρι ένα σημείο, κάτω και δεξιά από το τελευταίο εικονίδιο του προγράμματος, ώστε να σχηματιστεί ένα ορθογώνιο παραλληλόγραμμο το οποίο να περιέχει όλα τα εικονίδια του προγράμματος. Στο ορθογώνιο παραλληλόγραμμο πρέπει να προσέξουμε ώστε να μη συμπεριλάβουμε και το σημείο εκκίνησης του προγράμματος.

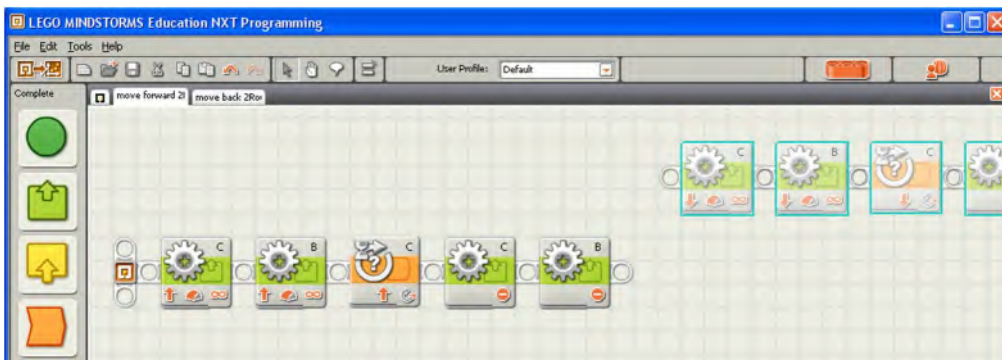


Στο σημείο αυτό όλα τα εικονίδια του προγράμματος πρέπει να περιβάλλονται από ένα μπλε περιθώριο.



• Από το μενού *edit* του λογισμικού NXT-G κάντε κλικ στην επιλογή *copy* και στη συνέχεια μεταβείτε στο πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** κάνοντας κλικ στην αντίστοιχη καρτέλα.

Στην περιοχή εργασίας της καρτέλα **κίνηση μπροστά για 2 περιστροφές** κάντε κλικ σε ένα σημείο δεξιά της τελευταίας εντολής του προγράμματος και από το μενού *edit* του λογισμικού NXT-G κάντε κλικ στην επιλογή *paste*.

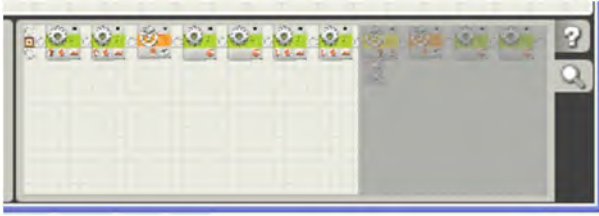


Βλέπουμε μπροστά μας το αντίγραφο που έχει δημιουργηθεί, το οποίο και μεταφέρουμε αμέσως μετά την τελευταία εντολή.

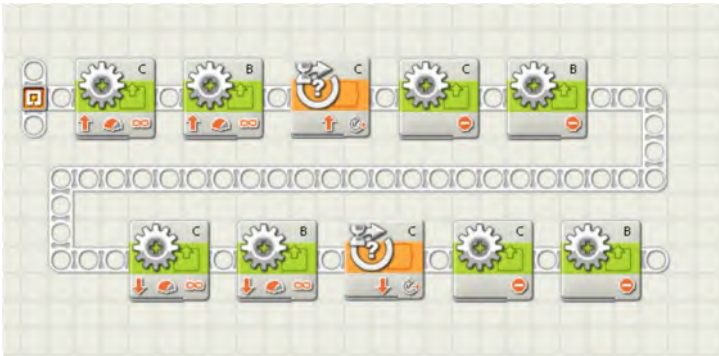


• Για να μετακινήσετε τη νέα ομάδα εντολών που μόλις αντιγράψατε σύρετε το πρώτο εικονίδιο της ομάδας αυτής και τοποθετείστε το στο τέλος της συνδετικής δοκού.

Αν το πρόγραμμά μας δε χωράει ολόκληρο στην ενεργή επιφάνεια της περιοχής εργασίας (αυτή που μπορούμε να δούμε στην οθόνη σας), μπορούμε να χρησιμοποιήσουμε το εργαλείο ολίσθησης (*pan tool*) της μπάρας εργαλείων για να μεταφερθούμε σε διάφορα σημεία του προγράμματος μας. Ένας πιο βολικός τρόπος είναι να μετακινούμε την ενεργή επιφάνεια στο χάρτη του προγράμματος, στο κάτω δεξιό μέρος της εφαρμογής μας.



Εναλλακτικά για να εισάσετε σε θέση να βλέπετε ολόκληρο το πρόγραμμα στην οθόνη μπορείτε να κατεβάσετε μερικές εντολές σε άλλη σειρά.

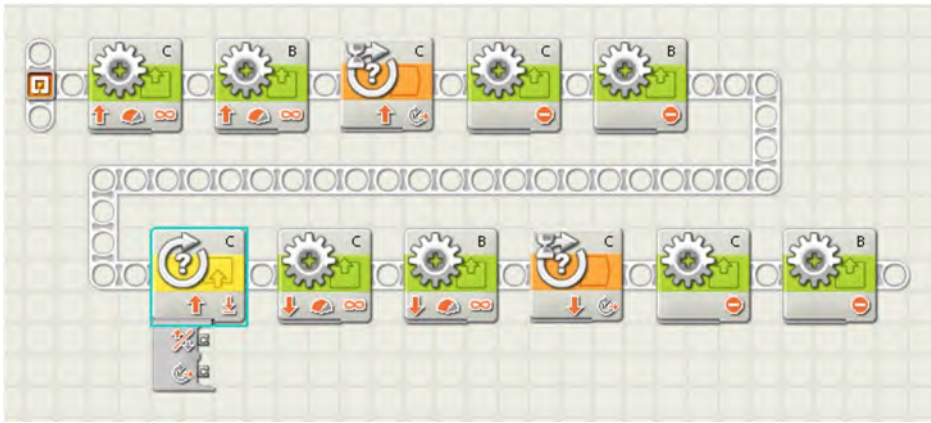


- Αποθηκεύστε το πρόγραμμά σας ως **κίνηση μπροστά για 2 περιστροφές και επιστροφή**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

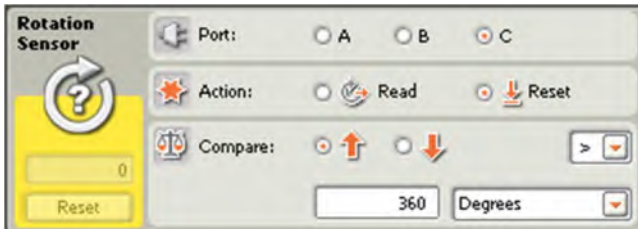
Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το ρομπότ μας θα κινηθεί προς τα μπροστά για 2 περιστροφές ακριβώς, θα σταματήσει και θα κινηθεί προς τα πίσω για 4 περιστροφές. Δεν θα σταματήσει δηλαδή στο σημείο εκκίνησης αλλά θα κινηθεί επιπλέον για άλλες 720 μοίρες (2 περιστροφές). Το σημείο αναφοράς του αισθητήρα περιστροφής είναι το σημείο που βρίσκεται το ρομπότ τη στιγμή που ξεκινάει η εκτέλεση του προγράμματος. Θα πρέπει να αλλάξουμε τη συνθήκη αναμονής στο δεύτερο εντολή αναμονής του προγράμματος έτσι ώστε να περιμένει μέχρι να περιστραφεί ο κινητήρας στη θέση 0 μοίρες και όχι 720° πίσω.

Εναλλακτικά μπορούμε πριν από δεύτερο μέρος του προγράμματος, να τοποθετήσουμε μία εντολή για να αναγκάσουμε τον αισθητήρα περιστροφής να αρχίσει να μετράει από την αρχή προτού αρχίσει να κινείται προς τα πίσω. Για το πετύχουμε αυτό χρησιμοποιούμε μία εντολή αισθητήρα περιστροφής.

- Από την υποπαλέτα αισθητήρων **Sensors Action** (έχει ως σχήμα ένα τετράγωνο με ένα βελάκι προς τα κάτω και κίτρινο χρώμα) επιλέξτε το εικονίδιο του αισθητήρα περιστροφής το οποίο έχει ως σχήμα ένα κύκλο με ένα ερωτηματικό, και σύρετε το στην περιοχή εργασίας πριν από το δεύτερο μέρος του προγράμματος.



Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη θύρα C αντί για την A και ως ενέργεια (Action) την αρχικοποίηση "Reset" αντί για την ανάγνωση "Read".



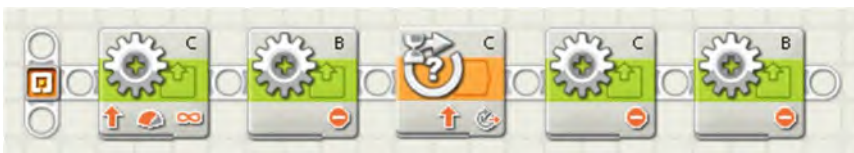
- Αποθηκεύστε τις αλλαγές στο πρόγραμμά σας ως **κίνηση μπροστά για 2 περιστροφές και επιστροφή**
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Συγχαρητήρια !!!

Ολοκληρώσατε την 3η δραστηριότητα

Δραστηριότητα 4η : Στροφή αριστερά

- Ανοίξτε το πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** που δημιουργήσατε στην δραστηριότητα 1.



- Επιλέξτε την εντολή κινητήρα B και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (Until) επιλέγοντας το σήμα του στοπ.

- Αποθηκεύστε το πρόγραμμα σας ως **στροφή αριστερά**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Στο πρόγραμμα γυρίζει η δεξιά ρόδα από το ρομπότ αφού ξεκινάει ο κινητήρας C αλλά η αριστερή του ρόδα δεν ακολουθεί (ο κινητήρας B είναι φρεναρισμένος) παρά παραμένει καθηλωμένη στη θέση του. Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το ρομπότ μας θα κινηθεί εκτελώντας μία στροφή αριστερά και θα σταματήσει μετά από δύο περιστροφές του κινητήρα C.

Σημείωση : Δεν είναι απαραίτητο να τοποθετήσουμε τις δύο εντολές με τις οποίες φρενάρουμε τον κινητήρα B.

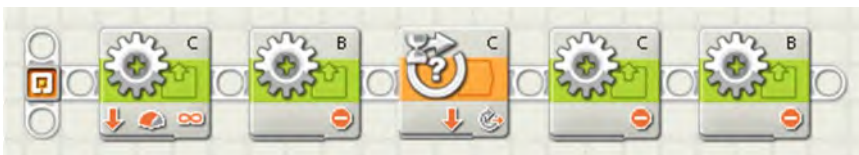
β) Στην περίπτωση που αντί να φρενάρουμε τον κινητήρα B, τον ρυθμίζουμε να γυρίζει προς τα μπροστά με μικρή ταχύτητα (αρκετά μικρότερη σε σχέση με τον κινητήρα C) το όχημα-ρομπότ θα εκτελέσει μία ομαλή στροφή αριστερά διαγράφοντας ένα μεγάλο τόξο με κέντρο τη ρόδα που είναι ακινητοποιημένη.



γ) Στην περίπτωση που αντί να φρενάρουμε τον κινητήρα B, τον ρυθμίζουμε να γυρίζει προς τα πίσω (δηλ. προς την αντίθετη κατεύθυνση) το όχημα-ρομπότ θα εκτελέσει μία στροφή αριστερά επιτόπου με κέντρο το διάμεσο ανάμεσα στους τροχούς του.



δ) Στην περίπτωση που ρυθμίσουμε τον κινητήρα C να γυρίζει προς τα πίσω αντί για μπροστά το όχημα-ρομπότ θα εκτελέσει μία στροφή δεξιά με την όπισθεν.



Συγχαρητήρια !!!

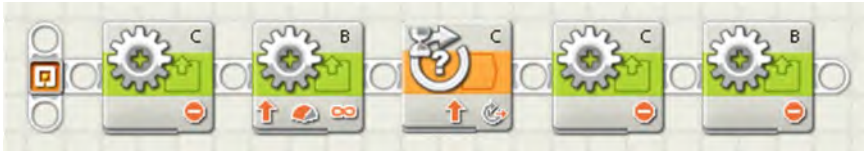
Ολοκληρώσατε την 4η δραστηριότητα

Δραστηριότητα 5η : Στροφή δεξιά

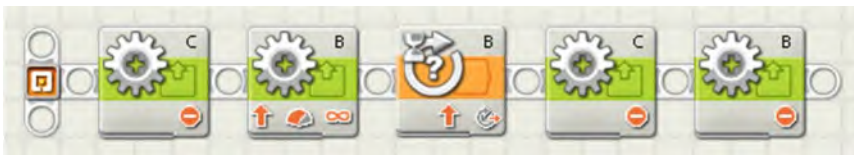
- Ανοίξτε το πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** που δημιουργήσαμε στη δραστηριότητα 1.

Θέλουμε το πρόγραμμα να κινήσει το όχημα-ρομπότ μας ώστε να εκτελέσει μία στροφή δεξιά και να σταματήσει μετά από 2 περιστροφές.

- Αυτή τη φορά επιλέξτε την εντολή κινητήρα C και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (*Until*) επιλέγοντας το σήμα του στοπ.



Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το ρομπότ μας θα κινηθεί εκτελώντας μία στροφή δεξιά αλλά δε θα σταματήσει να κινείται ποτέ. Στο πρόγραμμα αυτό ξεκινάει η λειτουργία του κινητήρα B και στη συνέχεια η εντολή αναμονής βάζει το πρόγραμμα να περιμένει ώσπου ο κινητήρας C συμπληρώσει δύο περιστροφές !!! Όμως ο κινητήρας C δε κινείται καθόλου, είναι σταματημένος. Το ρομπότ μας θα κινείται συνέχεια εκτελώντας κύκλους προς τα δεξιά. Για να λειτουργήσει το πρόγραμμα όπως θα θέλαμε, πρέπει στην εντολή αναμονής για περιστροφή να αλλάξουμε τη θύρα του κινητήρα που θα παρακολουθείται, ορίζοντας τη θύρα B αντί για τη C.



- Αποθηκεύστε το πρόγραμμά σας ως **στροφή δεξιά**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

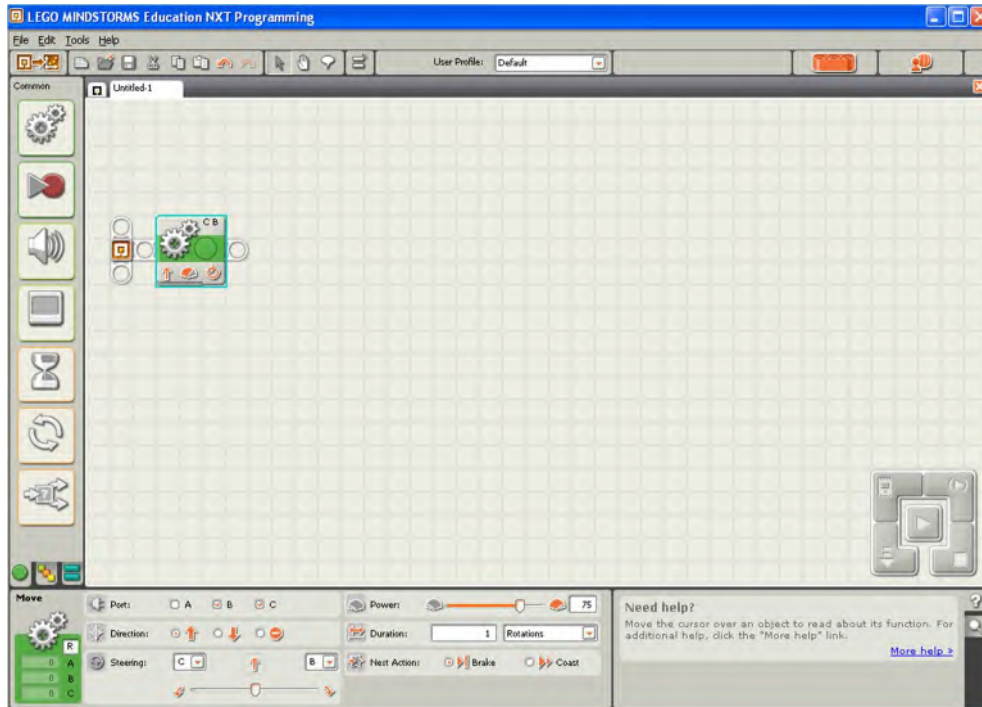
Συγχαρητήρια !!!

Ολοκληρώσατε την 5η δραστηριότητα

Τον κώδικα του NXT-G v2 των δραστηριοτήτων εξοικείωσης θα τον βρείτε στο σύνδεσμο :

[Εξοικείωση με το NXT-G](#)

Για περισσότερες πληροφορίες για τις εντολές του προγραμματιστικού περιβάλλοντος NXT-G ανατρέξτε στο ΠΑΡΑΡΤΗΜΑ Ι.



Ενότητα 5η

Δραστηριότητες στο περιβάλλον του NXT-G

Λίγα λόγια για τους κινητήρες

Στις προηγούμενες δραστηριότητες για να κινήσουμε το ρομπότ μας χρησιμοποιήσαμε έναν αριθμό από εντολές κινητήρα. Το λογισμικό NXT-G διαθέτει ακόμη μία εντολή με την οποία μπορούμε να κινήσουμε ένα ρομπότ, το οποίο κινείται με δύο κινητήρες -όπως αυτό που χρησιμοποιούμε. Πρόκειται για την εντολή μετακίνησης (*move block*) η οποία αποτελεί μία από τις πιο δημοφιλείς εντολές.

Με αυτήν την εντολή μπορούμε να πούμε στο ρομπότ μας το οποίο είναι εφοδιασμένο με δύο τουλάχιστον κινητήρες υπεύθυνους για την κίνηση του, να κινηθεί μπροστά ή πίσω σε ευθεία γραμμή ή να στρίψει ακολουθώντας μία καμπύλη γραμμή. Επίσης χρησιμοποιώντας την ίδια εντολή μπορούμε να πούμε στο ρομπότ μας να σταματήσει να κινείται.

Η εντολή κίνησης κινητήρα **motor** είναι πολύ απλή. Μία εντολή τέτοια ελέγχει μόνο έναν κινητήρα. Επομένως για να κάνουμε ένα πρόγραμμα για ένα τυπικό ρομπότ με δύο κινητήρες χρειαζόμαστε πάνω από μία τέτοια εντολή.

Η εντολή μετακίνησης **move** από την άλλη μπορεί να μας δώσει πλήρη έλεγχο στην κίνηση του ρομπότ μας. Η εντολή αυτή διαθέτει και ένα μεγάλο πλεονέκτημα : κατά τη διάρκεια της κίνησης του ρομπότ, παρακολουθεί τους ενσωματωμένους στους κινητήρες, αισθητήρες περιστροφής. Με τον τρόπο αυτό γνωρίζει ανά πάσα στιγμή τη θέση του ρομπότ και ελέγχει κατάλληλα τους κινητήρες ώστε να ακολουθηθεί η επιθυμητή διαδρομή που έχουμε ορίσει. Επίσης με τον ίδιο τρόπο, φροντίζει ώστε και οι δύο κινητήρες στο ρομπότ μας να περιστρέφονται συγχρονισμένα έτσι ώστε αν κάποια στιγμή το ρομπότ αρχίσει να αποκλίνει από την καθορισμένη πορεία του, με την εντολή αυτή το πρόγραμμα φροντίζει να το επαναφέρει στην αρχική του πορεία.

Αν επιλέξουμε το ρομπότ μας να κινηθεί για ένα συγκεκριμένο διάστημα (σε πλήρεις περιστροφές "*Rotations*", μοίρες περιστροφής "*Degrees*" ή χρονικό διάστημα "*Seconds*") το πρόγραμμα θα περιμένει να ολοκληρωθεί αυτή η κίνηση, προτού προχωρήσει στην εκτέλεση της επόμενης εντολής. Στην περίπτωση αυτή μπορούμε να καθορίσουμε και αν με το πέρας της κίνησης, οι κινητήρες θα φρενάρουν αστραπιαία "*Brake*" ή θα συνεχίσουν να ολισθαίνουν "*Coast*" μέχρι να σταματήσουν, λόγω τριβής, ενώ θα έχει αρχίσει η εκτέλεση της επόμενης εντολής (*Next Action*).

Αν επιλέξουμε το ρομπότ μας να κινείται συνεχώς "*Unlimited*" το πρόγραμμα θα θέσει σε κίνηση το ρομπότ και θα περάσει αμέσως στην εκτέλεση της επόμενης εντολής. Στην περίπτωση αυτή για να σταματήσουμε το ρομπότ, χρησιμοποιούμε αργότερα μέσα στο πρόγραμμα μας κάποια άλλη εντολή (συνήθως αναμονής από έναν αισθητήρα).

Η εντολή μετακίνησης εμφανίζεται πρώτη, πάνω ψηλά στη βασική παλέτα. Στην πλήρη παλέτα θα τη βρούμε στο υπομενού των βασικών εντολών.

Τώρα θα δούμε πως μπορούν να τροποποιηθούν τα προγράμματα που υλοποιήσαμε στις προηγούμενες δραστηριότητες με τη χρήση αυτή τη φορά της εντολής μετακίνησης.

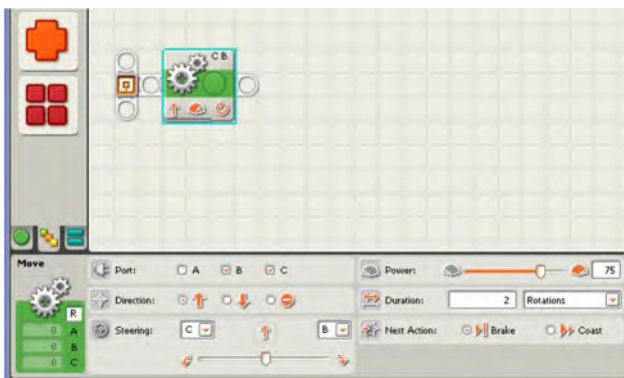
Δραστηριότητα 6η : κίνηση μπροστά συγχρονισμένα

- Ξεκινήστε ένα νέο πρόγραμμα.
- Από το υπομενού των βασικών ενεργειών **Common** (έχει ως σχήμα ένα πράσινο κύκλο) επιλέξτε το εικονίδιο για την εντολή μετακίνησης το οποίο έχει ως σχήμα δύο γρανάζια συζευγμένα, και τοποθετείστε το στο χώρο εργασίας δίπλα στο σημείο της αρχής του προγράμματος.

α) Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης αλλάξτε τη διάρκεια της μετακίνησης (Duration) σε 2 περιστροφές. Μην πειράξετε καμία άλλη ρύθμιση.



Το πρόγραμμά μας είναι έτοιμο. Δε χρειάζεται να τοποθετήσουμε κάποια εντολή για να σταματήσουμε τους κινητήρες με το πέρας της κίνησης μετά από 2 περιστροφές, αφού όπως έχουμε αναφέρει αυτοί θα φρενάρουν αυτόματα.

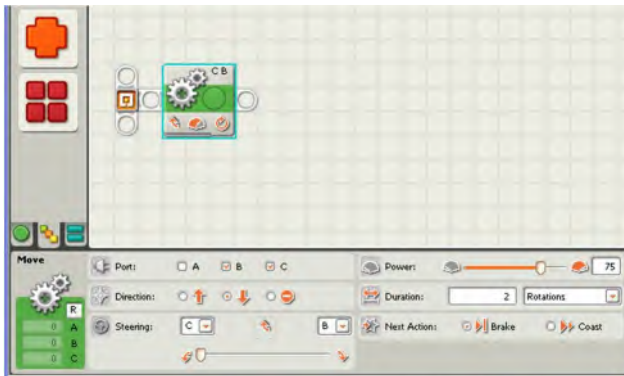


β) Για να μετακινηθεί το όχημα-ρομπότ μας προς τα πίσω για 2 περιστροφές αρκεί να αλλάξουμε την κατεύθυνση (Direction) της κίνησης, στον πίνακα ρυθμίσεων της εντολής μετακίνησης, προς τα πίσω.

γ) Για να μετακινηθεί το όχημα-ρομπότ μας προς τα μπροστά για 2 περιστροφές και να επιστρέψει πίσω, αρκεί να τοποθετήσουμε δύο εντολές μετακίνησης. Στην περίπτωση μάλιστα που χρησιμοποιούμε εντολές μετακίνησης, δε χρειάζεται να τοποθετήσουμε την εντολή του αισθητήρα περιστροφής με την κίτρινη λουρίδα (Rotation Sensor) για να μηδενίζουμε το μετρητή.



δ) Για να στρίψει το όχημα-ρομπότ μας προς τα αριστερά για δύο περιστροφές αρκεί να μετακινήσουμε ανάλογα την μπάρα του τιμονιού (Steering) στον πίνακα ρυθμίσεων της εντολής μετακίνησης.

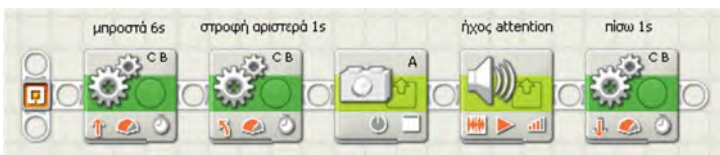


Το ρομπότ μας μπορεί να πάρει μία ομαλή ή πιο απότομη στροφή ανάλογα με τη θέση που θα τοποθετήσουμε το σημάδι στην μπάρα του τιμονιού. Δεν υπάρχει όμως τρόπος να στρίψει το ρομπότ μας επιτόπου με μία μόνο εντολή μετακίνησης.

Δραστηριότητα 7η : παρκάρισμα

Το όχημα-ρομπότ θέλουμε να κινηθεί προς τα μπροστά για 6 δευτερόλεπτα, να στρίψει προς τα αριστερά για 1 δευτερόλεπτο, να σταματήσει και στη συνέχεια να κινηθεί προς τα πίσω για 1 δευτερόλεπτο και να σταματήσει.

Καθώς πηγαίνει με την όπισθεν να ανάβει μία λάμπα και να ακούγεται ένας ήχος.



Η εντολή για το άναμμα ενός λαμπτήρα βρίσκεται στο υπομενού ενέργειας **Action**. Στον πίνακα ρυθμίσεων της, αφήνουμε τη θύρα εξόδου (Port) στο A και την ενέργεια (Action) στο "On". Για να πούμε στο ρομπότ μας να αναπαράγει έναν ήχο τοποθετούμε την εντολή ήχου ή οποία βρίσκεται και αυτή στο υπομενού ενέργειας. Στον πίνακα ρυθμίσεων της επιλέγουμε (File) να γίνει η αναπαραγωγή του αρχείου "attention" και ως λειτουργία (Function) επιλέγουμε ο ήχος να επαναλαμβάνεται "Repeat".

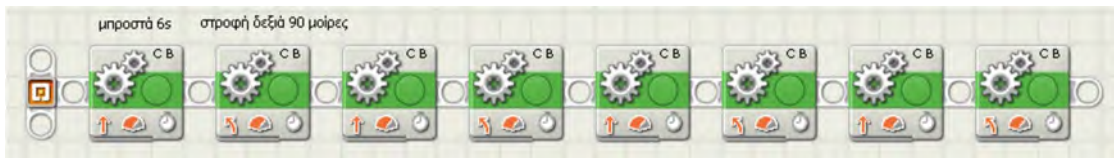
Με το πέρας του προγράμματος οι κινητήρες θα σταματήσουν να λειτουργούν αυτόματα. Το ίδιο θα συμβεί με το φως του λαμπτήρα και με την αναπαραγωγή του ήχου που τοποθετήσαμε στο πρόγραμμα μας, αφού με το πέρας ενός προγράμματος διακόπτεται η τροφοδοσία σε όλες τις θύρες εξόδου του NXT.

Δραστηριότητα 8η : τετράγωνο

Το όχημα-ρομπότ μας θέλουμε να κινηθεί ακολουθώντας μία διαδρομή ή οποία να έχει σχήμα τετραγώνου.

Ξεκινάμε βάζοντας το ρομπότ μας να διανύσει μία απόσταση σε ευθεία γραμμή και στη συνέχεια να στρίψει κατά 90° φρενάροντας τον έναν του κινητήρα. Για να πετύχουμε μία στροφή 90° ακριβώς, πρώτα πειραματιζόμαστε ώστε να βρούμε τις μοίρες που χρειάζεται να κινηθεί ο κινητήρας C, κρατώντας φρεναρισμένο τον κινητήρα B για να εκτελέσει το ρομπότ μας τη στροφή αυτή.

Στη συνέχεια επαναλαμβάνουμε την παραπάνω διαδικασία άλλες τρεις φορές ώστε το ρομπότ μας να επιστρέψει στο σημείο από όπου ξεκίνησε.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τους κινητήρες θα τον βρείτε στο σύνδεσμο :

[Κινητήρες](#)

Λίγα λόγια για τους αισθητήρες

Με την εντολή μετακίνησης (τα ίδια ισχύουν και στην εντολή κίνησης κινητήρα) μπορούμε να καθορίσουμε το ρομπότ μας να διανύσει μία συγκεκριμένη απόσταση ορίζοντας στον πίνακα ρυθμίσεων τη διάρκεια της μετακίνησης (Duration) σε συγκεκριμένο αριθμό περιστροφών "Rotations" ή μοιρών περιστροφής "Degrees" των κινητήρων ή καθορίζοντας συγκεκριμένο χρονικό διάστημα κίνησης σε δευτερόλεπτα "Seconds". Στις περιπτώσεις αυτές το ρομπότ μας προτού να περάσει στην εκτέλεση της επόμενης εντολής, θα περιμένει να ολοκληρωθεί η κίνηση που έχουμε καθορίσει. Μπορούμε να πούμε επίσης στο ρομπότ μας αν οι κινητήρες του θα φρενάρουν ή θα συνεχίσουν να ολισθαίνουν με την ολοκλήρωση της μετακίνησης.

Στις προηγούμενες δραστηριότητες προγραμματίσαμε το ρομπότ μας να εκτελέσει μία σειρά από προκαθορισμένες κινήσεις τις οποίες γνωρίζαμε και είχαμε υπολογίσει εκ των προτέρων.

Υπάρχει και ακόμη μία επιλογή που μπορούμε να εφαρμόσουμε για τη διάρκεια της μετακίνησης: μπορούμε να πούμε στο ρομπότ μας να θέσει σε λειτουργία τους κινητήρες χωρίς να προσδιορίσουμε εκ των προτέρων συγκεκριμένη διάρκεια "Unlimited". Σε αυτήν την περίπτωση το ρομπότ μας θα ξεκινήσει τους κινητήρες και θα περάσει αμέσως στην εκτέλεση της επόμενης εντολής. Την επιλογή αυτή τη χρησιμοποιούμε όταν δεν γνωρίζουμε εκ των προτέρων για πόσο διάστημα θα πρέπει να λειτουργήσουν οι κινητήρες μέχρι να επιτευχθεί μία συγκεκριμένη συνθήκη. Η συνθήκη αυτή μπορεί να γίνει αντιληπτή από το ρομπότ μας με τη χρήση των αισθητήρων, με τους οποίους μπορούμε να το εφοδιάσουμε. Με τον τρόπο αυτό μπορούμε να πούμε στο ρομπότ να κινείται συνεχώς μέχρι να συμβεί ένα συγκεκριμένο γεγονός για το οποίο θα ενημερωθεί με τη βοήθεια των αισθητήρων του.

Είδαμε ήδη από το πρώτο παράδειγμα μας τη χρήση του αισθητήρα περιστροφής με τον οποίο το ρομπότ μας μπόρεσε να αντιληφθεί το μέγεθος της περιστροφής ενός κινητήρα. Αντί να πούμε στο ρομπότ μας, από πιο μπροστά να κινηθεί μόνο για δύο περιστροφές και αμέσως μετά να σταματήσει, του είπαμε εναλλακτικά να ξεκινήσει και να συνεχίσει να κινείται μέχρις ότου συμπληρώσει δύο πλήρεις περιστροφές. Με τη διατύπωση αυτή πετυχαίνουμε περισσότερη ελευθερία δράσεως. Μας δίνεται η δυνατότητα να προσδώσουμε στο ρομπότ μας ένα πλήθος από συμπεριφορές που μόνο η φαντασία μας μπορεί να περιορίσει.

Οι αισθητήρες είναι ηλεκτρονικές συσκευές οι οποίες λειτουργούν με τον ίδιο τρόπο όπως τα αισθητήρια όργανα που διαθέτουμε εμείς οι άνθρωποι ή τα άλλα ζωντανά όντα. Οι αισθητήρες μπορούν να μετρήσουν μία σειρά από φυσικά χαρακτηριστικά όπως επαφή με ένα αντικείμενο, απόσταση από ένα αντικείμενο, χρώμα, φωτεινή ένταση, στάθμη ήχου, θερμοκρασία κτλ. Τη μέτρηση αυτή τη μετατρέπουν σε μία αριθμητική τιμή αντίστοιχη με το φυσικό μέγεθος που αντιλαμβάνονται.

Με τους αισθητήρες το ρομπότ μας μπορεί να αντιληφθεί διάφορα συμβάντα που βρίσκονται σε εξέλιξη μέσα στο περιβάλλον που δραστηριοποιείται και να αντιδράσει αναλόγως. Με τη βοήθεια των αισθητήρων, το ρομπότ μας μπορεί να αναγνωρίσει δυναμικά τις καταστάσεις τις οποίες συναντά κατά τη διάρκεια μιας αποστολής του και με τις αντίστοιχες οδηγίες που του δίνουμε για κάθε μία από τις καταστάσεις αυτές που πιθανόν να συναντήσει, το καθιστούμε ικανό να ανταπεξέρχεται επιτυχώς. Δουλειά λοιπόν του προγραμματιστή είναι να προβλέψει όλες τις πιθανές καταστάσεις που μπορεί να συναντήσει το ρομπότ του ώστε πρώτα από όλα να το σχεδιάσει κατάλληλα και να το εφοδιάσει με τους κατάλληλους αισθητήρες και στη συνέχεια να το προμηθεύσει με το κατάλληλο πρόγραμμα, με το οποίο θα μπορεί να ανταποκρίνεται με το σωστό τρόπο σε όλα τα συμβάντα που θα συναντήσει.

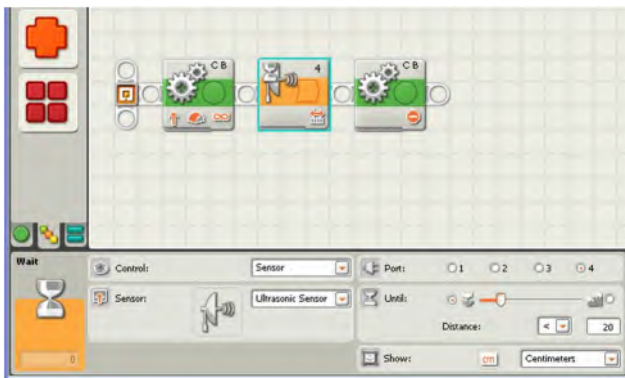
Το ρομπότ μας μπορεί να αντιληφθεί την απόσταση από ένα αντικείμενο με τη βοήθεια του αισθητήρα υπερήχων και αν έχει έρθει σε επαφή μαζί του με τον αισθητήρα αφής, μία συγκεκριμένη ένταση φωτός με τη βοήθεια του αισθητήρα φωτός, ένα συγκεκριμένο χρώμα με τη βοήθεια του αισθητήρα χρώματος, μία συγκεκριμένη στάθμη ήχου με τη βοήθεια του αισθητήρα ήχου, μία συγκεκριμένη θερμοκρασία με τη βοήθεια του αισθητήρα θερμοκρασίας, μία συγκεκριμένη περιστροφή με τη βοήθεια του αισθητήρα περιστροφής κ.α.

Δραστηριότητα 9η : σταμάτημα με αισθητήρα υπερήχων

Θέλουμε το όχημα-ρομπότ μας να κινείται μέχρι να συναντήσει ένα εμπόδιο στα 20 cm μπροστά του οπότε και να σταματήσει.

Για το σκοπό αυτό θα δώσουμε πρώτα στο ρομπότ μας την εντολή να ξεκινήσει να κινείται επ' αόριστο "Unlimited" αφού δεν ξέρουμε εκ των προτέρων πότε θα συναντήσει κάποιο εμπόδιο. Στη συνέχεια για να είναι σε θέση το ρομπότ μας να αντιληφθεί ένα εμπόδιο μπροστά του, θα τοποθετήσουμε στο πρόγραμμα μας μία εντολή αναμονής με αισθητήρα υπερήχων.

Η εντολή αυτή βρίσκεται στην υποπαλέτα ροής **flow**. Στον πίνακα ρυθμίσεων της, αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα υπερήχων "Ultrasonic Sensor" από αισθητήρα αφής "Touch Sensor". Μπορούμε να παρατηρήσουμε ότι το πρόγραμμα θεωρεί ότι ο αισθητήρας μας είναι συνδεδεμένος στη θύρα εισόδου 4, στην οποία ενδείκνυται να τοποθετούμε έναν αισθητήρα υπερήχων για να μην μπερδευόμαστε. Αν για κάποιο λόγο χρειαστεί να συνδέσουμε τον αισθητήρα αυτόν σε μία άλλη θύρα, θα πρέπει να τροποποιήσουμε τη ρύθμιση αυτή αντίστοιχα. Στην επιλογή (Until) για τη συνθήκη που θα πρέπει εκπληρωθεί ώστε να τερματιστεί η αναμονή από το ρομπότ μας, πληκτρολογούμε την αριθμητική τιμή 20, αφού πρώτα έχουμε επιλέξει ως μονάδα μέτρησης (Show) τα εκατοστά του μέτρου "Centimeters" αντί για τις ίντσες "Inches". Τέλος για να πούμε στο ρομπότ να σταματήσει μόλις ικανοποιηθεί η συνθήκη που καθορίσαμε προηγουμένως, τοποθετούμε μία εντολή μετακίνησης με επιλεγμένη τη ρύθμιση στοπ στον πίνακα ρυθμίσεων της.



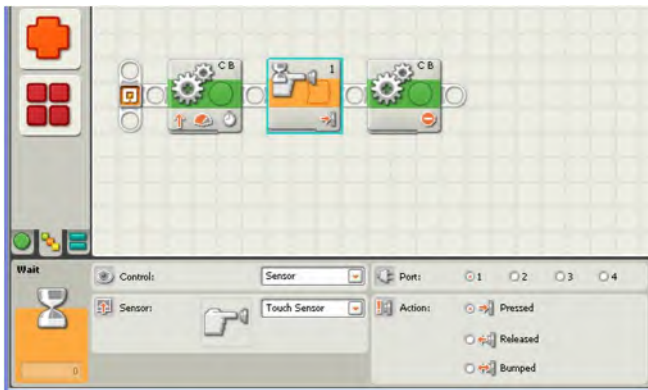
Αν δεν τοποθετήσουμε πριν από το τέλος του προγράμματος μας την εντολή για το σταμάτημα των κινητήρων, το ρομπότ μας θα σταματήσει να κινείται και πάλι αφού όπως έχουμε αναφέρει με τον τερματισμό ενός προγράμματος το NXT διακόπτει την τροφοδοσία σε όλες τις θύρες εξόδου του. Το ρομπότ μας όμως δε θα σταματήσει ακαριαία αλλά θα συνεχίσει για ένα πολύ μικρό διάστημα να ολισθαίνει.

Δραστηριότητα 10η : σταμάτημα με αισθητήρα αφής

Στην περίπτωση που επιθυμούμε το όχημα-ρομπότ μας να κινείται μέχρι να πέσει πάνω σε ένα τοίχο πρέπει να χρησιμοποιήσουμε έναν αισθητήρα αφής. Τον αισθητήρα αυτόν, τον τοποθετούμε στην μπροστινή μεριά του οχήματος-ρομπότ μας πάνω σε ένα κατάλληλο προφυλακτήρα, που πρέπει να κατασκευάσουμε.

Ένας αισθητήρας αφής μας δίνει την πληροφορία, αν ο διακόπτης του είναι πατημένος ή όχι σε μια συγκεκριμένη χρονική στιγμή. Αναφέρει το γεγονός αυτό στο NXT με την αριθμητική τιμή 1 για την περίπτωση που ο διακόπτης βρεθεί πατημένος και 0 για την περίπτωση που είναι απελευθερωμένος.

Με τον αισθητήρα αφής το ρομπότ μας μπορεί να διαπιστώσει αν έχει χτυπήσει σε κάποιο εμπόδιο. Η διαφορά ανάμεσα σε έναν αισθητήρα αφής και έναν αισθητήρα υπερήχων είναι ότι ο αισθητήρας αφής πρέπει να έρθει σε επαφή με κάποιο αντικείμενο για να το ανιχνεύσει ενώ ο αισθητήρας υπερήχων μπορεί να ανιχνεύσει ένα αντικείμενο που θα βρεθεί μπροστά του σε κάποια απόσταση την οποία εμείς έχουμε καθορίσουμε. Με τον τρόπο αυτόν ο αισθητήρας υπερήχων επιτρέπει σε ένα ρομπότ να σταματήσει πριν πέσει σε κάποιο αντικείμενο, ενώ με τον αισθητήρα αφής το ρομπότ δεν θα σταματήσει παρά μόνο όταν συγκρουστεί με το αντικείμενο.



Δραστηριότητα 11η : εκκίνηση με αισθητήρα ήχου και σταμάτημα με αισθητήρα υπερήχων

Θα τροποποιήσουμε το πρόγραμμα της δραστηριότητας 9 ώστε το όχημα-ρομπότ μας να μην ξεκινήσει μόλις τρέξουμε το πρόγραμμα μας αλλά να περιμένει από εμάς να κτυπήσουμε παλαμάκια. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε μία εντολή αναμονής για ήχο. Την εντολή αυτή την τοποθετούμε στην αρχή του προγράμματος έτσι ώστε το ρομπότ να περιμένει από εμάς να κτυπήσουμε παλαμάκια προτού εκτελέσει το υπόλοιπο μέρος του προγράμματος. Ο αισθητήρας ήχου ανιχνεύει τη στάθμη του ήχου που παράγεται κάθε χρονική στιγμή στο περιβάλλον που δραστηριοποιείται και τη μετατρέπει σε μία αντίστοιχη αριθμητική τιμή, την οποία και στέλνει στο NXT μέσω της θύρας εισόδου που είναι συνδεδεμένος.

Πριν χρησιμοποιήσουμε έναν αισθητήρα ήχου (τα ίδια ισχύουν και για τον αισθητήρα φωτός όπως θα δούμε παρακάτω) πρέπει να κάνουμε μία σειρά από μετρήσεις με σκοπό να πληροφορηθούμε σε ποιες αριθμητικές τιμές αντιστοιχούν οι διάφοροι ήχοι που θα συναντήσει το ρομπότ μας στο συγκεκριμένο περιβάλλον που θα δραστηριοποιηθεί. Στη συνέχεια πρέπει να υπολογίσουμε τη στάθμη του ήχου (κατώφλι) πάνω ή κάτω από την οποία μας ενδιαφέρει το ρομπότ μας να αλλάζει συμπεριφορά.

Μπορούμε να λάβουμε μία εκτίμηση του σημείου αυτού μετρώντας τις τιμές σε δύο ακραίες καταστάσεις και υπολογίζοντας το μέσο όρο τους. Έτσι αν θέλουμε να το ρομπότ μας να διακρίνει έναν απαλό από έναν δυνατό ήχο θα πρέπει να κάνουμε μία μέτρηση όταν επικρατεί ησυχία και μία άλλη μέτρηση όταν έχουμε αυξημένο θόρυβο. Για να προσδιορίσουμε στη συνέχεια την τιμή για το κατώφλι μας στη συγκεκριμένη περίπτωση, υπολογίζουμε το μέσο όρο αυτών των τιμών. Ήχος κάτω από την τιμή αυτή θα εκλαμβάνεται από το ρομπότ μας ως απαλός ήχος και πάνω από την τιμή αυτή ως δυνατός. Ενδεικτικά αναφέρουμε τη στάθμη που κατά προσέγγιση παράγουν οι παρακάτω δραστηριότητες :

Ησυχία 0%

Θόρυβος σε δωμάτιο 6%

Ομιλίες 45%

Χειροκρότημα 70%

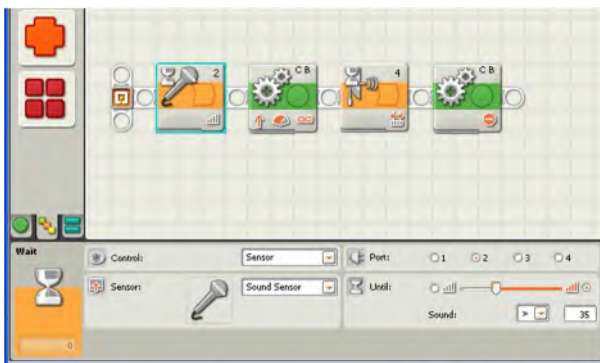
Δυνατός θόρυβος 100%

Για να κάνουμε μία μέτρηση για τη στάθμη ενός ήχου χρησιμοποιούμε την επιλογή View από το μενού του επεξεργαστή NXT.

Στη δική μας περίπτωση, μας ενδιαφέρει το ρομπότ μας να διακρίνει πότε θα ακουστεί ένας ήχος με τη στάθμη που παράγεται, όταν κτυπήσουμε παλαμάκια με τα χέρια μας. Στο παράδειγμα μας εκτελέσαμε μία μέτρηση χωρίς να κάνουμε κάποιο θόρυβο και μία μέτρηση τη στιγμή που χτυπήσαμε παλαμάκια. Από τις μετρήσεις αυτές λάβαμε τις αριθμητικές τιμές 5 και 65. Στο πρόγραμμα μας λοιπόν θα χρησιμοποιήσουμε ως κατώφλι το μέσο όρο $5+65/2=35$. Οποιοσδήποτε θόρυβος της τάξης του 35% και άνω θα εκλαμβάνεται από το ρομπότ μας ως χειροκρότημα.

Η εντολή για την αναμονή με αισθητήρα ήχου βρίσκεται στην υποπαλέτα ροής **flow**. Την εντολή αυτήν την τοποθετούμε στην αρχή του προγράμματος της δραστηριότητας 9.

Στον πίνακα ρυθμίσεων της αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα ήχου "Sound Sensor" από αισθητήρα αφής "Touch Sensor". Μπορούμε να παρατηρήσουμε ότι το πρόγραμμα θεωρεί ότι ο αισθητήρας μας είναι συνδεδεμένος στη θύρα εισόδου 2, στην οποία ενδείκνυται να τοποθετούμε έναν αισθητήρα ήχου για να μην μπερδευόμαστε. Στην επιλογή (Until) για τη συνθήκη που θα πρέπει εκπληρωθεί ώστε να τερματιστεί η αναμονή από το ρομπότ μας, πληκτρολογούμε την αριθμητική τιμή 35.



Αν θέλουμε το ρομπότ μας να ξεκινήσει όταν θα έχουν ολοκληρωθεί τα παλαμάκια, πρέπει να τοποθετήσουμε ακόμη μία εντολή αναμονής για ήχο αμέσως μετά την πρώτη, στην οποία ορίζουμε ως κατώφλι μία πολύ χαμηλή τιμή.



Σημείωση : Προτού χρησιμοποιήσουμε στο πρόγραμμα μας ένα αισθητήρα ήχου (το ίδιο ισχύει και για τον αισθητήρα φωτός) πρέπει να κάνουμε ρεγουλάρισμα του αισθητήρα ώστε να τον προσαρμόζουμε στις συνθήκες που επικρατούν στο συγκεκριμένο περιβάλλον που το ρομπότ μας πρόκειται να δραστηριοποιηθεί.

Το ρεγουλάρισμα αυτό γίνεται με τη χρήση δύο εντολών καλιμπραρίσματος. Ένα ενδεικτικό πρόγραμμα για την εργασία αυτή είναι το παρακάτω :



Δραστηριότητα 12η : άναμμα φωτός σε τούνελ

Θέλουμε το όχημα-ρομπότ μας να ξεκινήσει να κινείται και να ανάψει τα φώτα όταν μπει σε μία σκοτεινή περιοχή.

Ο αισθητήρας φωτός ανιχνεύει την ποσότητα του περιβάλλοντος φωτός ή εναλλακτικά το φως που ανακλάται από μία επιφάνεια. Αναφέρει αυτή την ποσότητα στο NXT με μία αριθμητική τιμή από το 0 (απόλυτο σκοτάδι) ως το 100 (άπλετο φως). Μπορούμε να τον χρησιμοποιήσουμε για να ξεκινάμε και να σταματάμε έναν κινητήρα ή ένα λαμπτήρα.

Πριν χρησιμοποιήσουμε έναν αισθητήρα φωτός πρέπει να κάνουμε μία σειρά από μετρήσεις με σκοπό να πληροφορηθούμε σε ποιες αριθμητικές τιμές αντιστοιχούν τα διάφορα επίπεδα φωτός που θα συναντήσει το ρομπότ μας στο συγκεκριμένο περιβάλλον που θα δραστηριοποιηθεί. Στη συνέχεια πρέπει να υπολογίσουμε τη στάθμη του φωτός (κατώφλι) πάνω ή κάτω από την οποία μας ενδιαφέρει το ρομπότ μας να αλλάζει συμπεριφορά. Μπορούμε να λάβουμε μία εκτίμηση του σημείου αυτού μετρώντας τις τιμές σε δύο ακραίες καταστάσεις και υπολογίζοντας το μέσο όρο τους.

Έτσι αν θέλουμε να το ρομπότ μας να διακρίνει πότε υπάρχει επαρκές φως και πότε όχι θα πρέπει να κάνουμε δύο αντίστοιχες μετρήσεις στο χώρο που θα δραστηριοποιηθεί το ρομπότ μας και στις περιοχές με επαρκές και ελλιπές φως. Για να προσδιορίσουμε στη συνέχεια την τιμή για το κατώφλι μας, υπολογίζουμε το μέσο όρο αυτών των τιμών. Στάθμη φωτός πάνω από

την τιμή αυτή θα εκλαμβάνεται από το ρομπότ μας ως κανονικό φως και κάτω από την τιμή αυτή ως σκοτάδι.

Για να κάνουμε μία μέτρηση για τη στάθμη ενός ήχου χρησιμοποιούμε την επιλογή View από το μενού του επεξεργαστή NXT.

Στην περίπτωση τη δική μας, μας ενδιαφέρει το ρομπότ μας να διακρίνει πότε θα σκοτεινιάσει. Στο παράδειγμα μας εκτελέσαμε μία μέτρηση σε ένα σημείο με κανονικό φως δωματίου και σε ένα άλλο μέσα στο τούνελ. Από τις μετρήσεις αυτές λάβαμε τις αριθμητικές τιμές 55 και 35. Στο πρόγραμμα μας λοιπόν θα χρησιμοποιήσουμε ως κατώφλι για το σκοτάδι το $55+35/2=40$. Οποιοσδήποτε τιμή της τάξης του 40 και κάτω θα εκλαμβάνεται από το ρομπότ μας ως σκοτάδι.

Η εντολή για την αναμονή με αισθητήρα φωτός βρίσκεται στην υποπαλέτα ροής **flow**. Την εντολή αυτήν την τοποθετούμε μετά από μία εντολή μετακίνησης του ρομπότ μας.

Στον πίνακα ρυθμίσεων της αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα φωτός "Light Sensor" από αισθητήρα αφής "Touch Sensor". Μπορούμε να παρατηρήσουμε ότι το πρόγραμμα θεωρεί ότι ο αισθητήρας μας είναι συνδεδεμένος στη θύρα εισόδου 3, στην οποία ενδείκνυται να τοποθετούμε έναν αισθητήρα φωτός για να μην μπερδευόμαστε.

Στο πρόγραμμα αυτό, μας ενδιαφέρει το επίπεδο του φωτός από το περιβάλλον. Για το λόγο αυτό στη ρύθμιση της λειτουργίας (Function) απενεργοποιούμε την επιλογή για την παραγωγή υπέρυθρης ακτινοβολίας "Generate light" αφού δε χρειαζόμαστε ο αισθητήρας φωτός να εκπέμπει τη δική του υπέρυθρη ακτινοβολία.

Στην επιλογή (Until) για τη συνθήκη που θα πρέπει να εκπληρωθεί ώστε να τερματιστεί η αναμονή από το ρομπότ μας πληκτρολογούμε την αριθμητική τιμή 40 και αλλάζουμε την κατεύθυνση της ανισότητας σε μικρότερο από "<" αντί του μεγαλύτερο από ">". Στη συνέχεια αφού τοποθετήσουμε μία εντολή λαμπτήρα για το άναμμα των φώτων, για να προσδιορίσουμε για πόσο διάστημα θα μείνουν τα φώτα αναμμένα, τοποθετούμε άλλη μία εντολή αναμονής με αισθητήρα φωτός στον πίνακα ρυθμίσεων της οποίας αυτή τη φορά ρυθμίζουμε τη συνθήκη για φως μεγαλύτερο από ">" 40. Με αυτό τον τρόπο διασφαλίζουμε ότι όσο το ρομπότ μας βρίσκεται μέσα στο τούνελ το φως θα παραμένει ανοικτό. Αμέσως μετά την εντολή της αναμονής, τοποθετούμε μία εντολή λαμπτήρα στον πίνακα ρυθμίσεων της οποίας ορίζουμε ως ενέργεια (Action) το "Off" για το σβήσιμο των φώτων.

Στο πρόγραμμα αυτό αφήνουμε το ρομπότ μας να κινηθεί για άλλα 3 δευτερόλεπτα ώστε να σιγουρευτούμε ότι θα βγει εντελώς από το τούνελ.



Δραστηριότητα 13η : υλοποίηση συναγερμού

Όταν περάσει ένα αντικείμενο μπροστά από έναν αισθητήρα φωτός ή όταν σκοτεινιάσει το NXT ανάβει ένα λαμπάκι και παίζει έναν ήχο.



Δραστηριότητα 14η : κίνηση μπροστά μέχρι να σκοτεινιάσει

Το όχημα - ρομπότ κινείται προς τα μπροστά μέχρι να σκοτεινιάσει, οπότε περιμένει για 2 δευτερόλεπτα, κινείται προς τα πίσω μέχρι να βγει από το σκοτάδι και σταματάει.



Δραστηριότητα 15η : κίνηση μπροστά μέχρι τη μαύρη γραμμή

Το όχημα - ρομπότ κινείται προς τα μπροστά μέχρι να περάσει πάνω από μία μαύρη γραμμή οπότε και σταματάει.



Λίγα λόγια για τους ήχους και τις εικόνες

Πολλές φορές κατά την εκτέλεση ενός προγράμματος θέλουμε να γνωρίζουμε την κατάσταση στην οποία βρίσκεται εκείνη τη στιγμή το ρομπότ μας. Ειδικά αν βρισκόμαστε στο στάδιο δοκιμής ενός προγράμματος μας, θέλουμε να ξέρουμε που βρίσκεται ανά πάσα στιγμή σε ποιο σημείο δεν λειτούργησε σωστά το πρόγραμμα μας. Στις περιπτώσεις αυτές μπορούμε να χρησιμοποιήσουμε τις εντολές ήχου και προβολής. Τις εντολές αυτές μπορούμε να τις χρησιμοποιήσουμε για να προσδώσουμε στο ρομπότ μας ανθρωπόμορφα χαρακτηριστικά, όπως να απευθύνεται σε μας βγάζοντας κατάλληλους ήχους ή και σχεδιάζοντας διάφορες εικόνες στην οθόνη του και να εκφράζει έτσι μία σειρά από συναισθήματα.

Τους ήχους και τις εικόνες μπορούμε να τους χρησιμοποιήσουμε και για να προσδιορίσουμε ακριβώς σε μία συγκεκριμένη χρονική στιγμή, ποιο σημείο του προγράμματος εκτελείται από το ρομπότ μας.

Δραστηριότητα 16η : σταμάτημα με αισθητήρα αφής με ενημέρωση

Θα χρησιμοποιήσουμε το πρόγραμμα από τη δραστηριότητα 10 το οποίο και θα τροποποιήσουμε ώστε το ρομπότ μας να απευθύνεται σε εμάς αναπαράγοντας έναν ήχο και σχεδιάζοντας μία εικόνα

Οι εντολές για αναπαραγωγή ήχου και εμφάνισης μίας εικόνας στην οθόνη του NXT βρίσκονται και οι δύο στο υπομενού ενέργειας **Action**. Τοποθετούμε καταρχήν μία εντολή ήχου αμέσως μετά την εντολή της αναμονής. Στον πίνακα ρυθμίσεων της επιλέγουμε (File) να γίνει η αναπαραγωγή του αρχείου "Crying 02". Στη συνέχεια τοποθετούμε μία εντολή εμφάνισης αμέσως μετά την εντολή για το σταμάτημα των κινητήρων. Στον πίνακα ρυθμίσεων της επιλέγουμε (File) να γίνει η εμφάνιση της εικόνας "Sad 02": 

Σημείωση : Για να μπορέσουμε να προλάβουμε να δούμε την εικόνα θα πρέπει να τοποθετήσουμε δίπλα από την εντολή της εμφάνισης μία εντολή αναμονής χρόνου. Διαφορετικά η εμφάνιση θα διαρκέσει για τόσο λίγο ώστε να μη δούμε να εμφανίζεται τίποτα στην οθόνη του NXT.



Αν θέλουμε να είμαστε σίγουροι ότι αυτός που θα τρέξει το πρόγραμμα θα μπορέσει να δει εικόνα που βάλαμε να εμφανιστεί, μία καλή τεχνική είναι αντί για την εντολή της αναμονής για χρόνο να τοποθετήσουμε μία εντολή αναμονής για πάτημα ενός πλήκτρου του NXT. Στην περίπτωση αυτή η προβολή της εικόνας μας θα σταματήσει μόνο όταν κάποιος πατήσει το πλήκτρο Enter του NXT ή όποιο άλλο πλήκτρο έχουμε επιλέξει στον πίνακα ρυθμίσεων της εντολή αυτής.



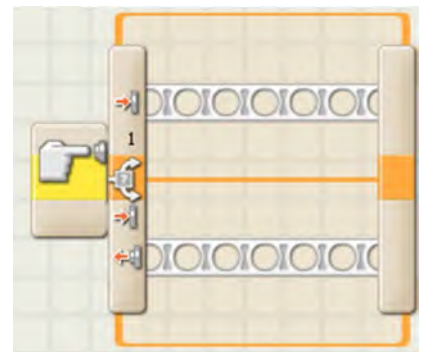
Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τους αισθητήρες θα τον βρείτε στο σύνδεσμο :

[Αισθητήρες](#)

Λίγα λόγια για τη δομή επιλογής

Συχνά μέσα σε ένα πρόγραμμα χρειάζεται να πούμε στο ρομπότ μας τι να κάνει όταν συναντήσει μία μελλοντική κατάσταση, την οποία δεν μπορούμε να ξέρουμε από πριν αν θα τη συναντήσει και πότε. Πχ να πέσει πάνω σε ένα εμπόδιο. Σε αυτήν την περίπτωση θα πρέπει να έχουμε προγραμματίσει το ρομπότ μας κατάλληλα, ώστε να είναι σε θέση τη στιγμή εκείνη να πάρει μια απόφαση για τι θα κάνει, στηριζόμενο στην τιμή κάποιας παραμέτρου, η οποία μπορεί να είναι η τιμή ενός αισθητήρα ή ένα χρονικό διάστημα.

Για να το πετύχουμε αυτό χρησιμοποιούμε μία δομή επιλογής. Με μία εντολή επιλογής δημιουργούνται μέσα στο πρόγραμμα δύο πιθανές διαδρομές (μονοπάτια) τις οποίες μπορεί ακολουθήσει το πρόγραμμα μας. Από τα δύο μονοπάτια αυτά, το πρόγραμμα θα ακολουθήσει μόνο το ένα, ανάλογα με την κατάσταση που θα έχει διαμορφωθεί εκείνη τη στιγμή, μετά από την πραγματοποίηση ενός ελέγχου.



Δραστηριότητα 17η : διακλάδωση με αισθητήρα αφής

Θέλουμε το ρομπότ μας να κινηθεί μπροστά για 5 δευτερόλεπτα και στη συνέχεια να κινηθεί προς τα πίσω για 2 δευτερόλεπτα ή να σταματήσει ανάλογα με την κατάσταση ενός αισθητήρα αφής τη στιγμή εκείνη.

Τοποθετούμε καταρχήν την εντολή για την κίνηση του ρομπότ μας για 5 δευτερόλεπτα και στη συνέχεια εισάγουμε στο πρόγραμμα μας μία εντολή επιλογής με αισθητήρα αφής.

Το εικονίδιο για τη δομή επιλογής βρίσκεται στο υπομενού για δομές **Flow** και έχει ως εικόνα ένα τετράγωνο με ερωτηματικό με ένα βελάκι για είσοδο και δύο βελάκια για έξοδο. Με την εντολή επιλογής με αισθητήρα αφής μπορούμε να προγραμματίσουμε το ρομπότ μας να κάνει διαφορετικά πράγματα, ανάλογα με το αν ο διακόπτης του αισθητήρα τη στιγμή που θα γίνει ο έλεγχος θα βρίσκεται πατημένος ή απελευθερωμένος.

Στον πίνακα ρυθμίσεων της εντολής επιλογής με αισθητήρα αφής δε χρειάζεται να κάνουμε καμία αλλαγή.

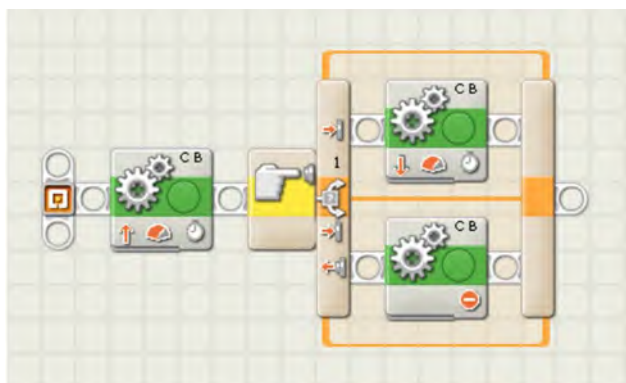


Βλέπουμε ότι στο πρόγραμμά μας με τη δομή επιλογής δημιουργήθηκαν δύο εναλλακτικές διαδρομές τις οποίες μπορεί να εκτελέσει το ρομπότ μας. Σε αυτές θα πρέπει να εισάγουμε τις εντολές που επιθυμούμε για την κάθε μία περίπτωση.

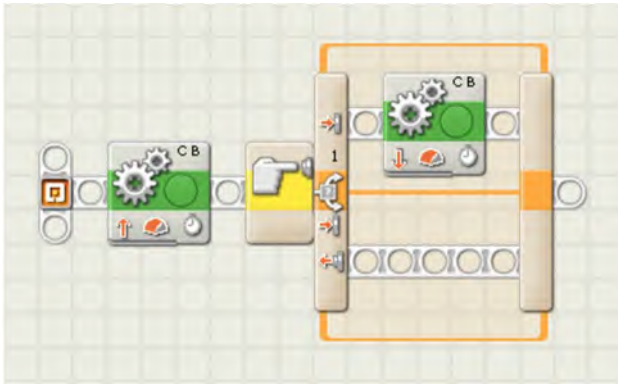
Αυτό που θέλουμε είναι :

αν ο διακόπτης του αισθητήρα αφής που είναι συνδεδεμένος στη θύρα 1 βρεθεί πατημένος τότε το ρομπότ μας να κινηθεί προς τα πίσω για 2 δευτερόλεπτα, **αλλιώς (αν δηλαδή** ο διακόπτης του αισθητήρα αφής που είναι συνδεδεμένος στη θύρα 1 βρεθεί απελευθερωμένος **τότε)** το ρομπότ μας να σταματήσει.

Τοποθετούμε λοιπόν τις κατάλληλες εντολές σε κάθε μία από της δύο διαδρομές: Την εντολή για την κίνηση του ρομπότ μας στο πάνω μέρος της εντολή επιλογής για την περίπτωση που ο διακόπτης του αισθητήρας αφήs είναι πατημένος και στο κάτω μέρος την εντολή για το σταμάτημα του ρομπότ μας αντίστοιχα.



Την εντολή για να σταματήσει το ρομπότ μας στην περίπτωση που ο διακόπτης του αισθητήρα αφής είναι απελευθερωμένος τη στιγμή του ελέγχου μπορούμε να την παραλείψουμε, αφού το ρομπότ μας μετά την ολοκλήρωση της κίνησης του προς τα εμπρός έχει προγραμματιστεί να σταματήσει.



Δραστηριότητα 18η : διακλάδωση με αισθητήρα φωτός

Με το εικονίδιο διακλάδωσης αισθητήρα φωτός μπορούμε να προγραμματίσουμε το ρομπότ μας να κάνει διαφορετικά πράγματα, ανάλογα με το αν η τιμή που θα διαβάσει ο αισθητήρας τη στιγμή που θα γίνει ο έλεγχος, θα είναι μεγαλύτερη ή όχι από μία συγκεκριμένη τιμή, πχ. 55. Η απόφαση που θα λάβει το ρομπότ μας δηλαδή, εξαρτάται από την ένταση του φωτός που θα μετρηθεί από τον αισθητήρα, τη στιγμή του ελέγχου.

Θέλουμε το ρομπότ μας να κινηθεί προς τα μπροστά για 5 δευτερόλεπτα και στη συνέχεια να στρίψει για 1 δευτερόλεπτα προς τα δεξιά ή προς τα αριστερά ανάλογα την τιμή που θα ανιχνεύσει ο αισθητήρα φωτός τη στιγμή εκείνη.

Τον αισθητήρα αυτόν τον τοποθετούμε πάνω σε ένα κατάλληλο βραχίονα που πρέπει να κατασκευάσουμε έτσι ώστε η λυχνία υπέρυθρης ακτινοβολίας του αισθητήρα να κοιτάει προς τα κάτω και να βρίσκεται σε μικρή απόσταση από το έδαφος.

Τοποθετούμε καταρχήν την εντολή για την κίνηση του ρομπότ μας για 5 δευτερόλεπτα και στη συνέχεια εισάγουμε στο πρόγραμμά μας μία εντολή επιλογής. Η εντολή αυτή όταν την τοποθετούμε στην περιοχή εργασίας εμφανίζεται ως εντολή επιλογής με χρήση του αισθητήρα αφής. Από τον πίνακα ρυθμίσεων μπορούμε να επιλέξουμε οποιοδήποτε άλλο αισθητήρα, η τιμή του οποίου θέλουμε να χρησιμοποιηθεί στη συνθήκη ελέγχου.

Στον πίνακα ρυθμίσεων της αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα φωτός "Light Sensor" από αισθητήρα αφής "Touch Sensor".

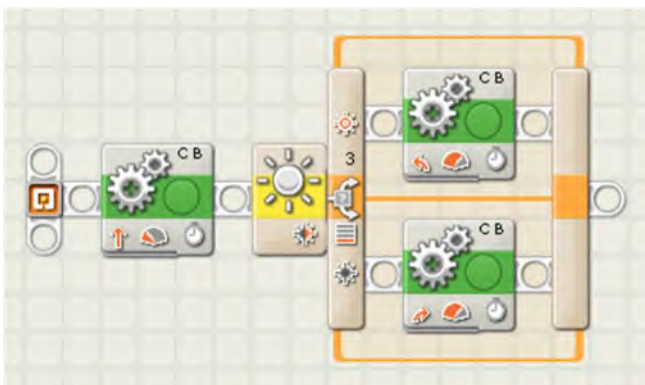
Στο πρόγραμμα αυτό, μας ενδιαφέρει ο αισθητήρας φωτός να μπορεί να διακρίνει τη απόχρωση στο δάπεδο στην κλίμακα του γκρι. Για το λόγο αυτό στη ρύθμιση της λειτουργίας (Function) αφήνουμε ενεργή την παράμετρο της παραγωγής υπέρυθρης ακτινοβολίας "Generate light" ώστε ο αισθητήρας φωτός να εκπέμπει τη δική του υπέρυθρη ακτινοβολία με τη χρήση της οποίας μπορεί να ανιληφθεί την απόχρωση μίας επιφάνειας.

Στην επιλογή (Until) για τη συνθήκη που θα πρέπει εκπληρωθεί πληκτρολογούμε την αριθμητική τιμή 40.



Αυτό που θέλουμε είναι :

αν ο αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα 3 ανιχνεύσει μία ποσότητα φωτός πάνω από την αριθμητική τιμή 40 **τότε** το ρομπότ να στρίψει αριστερά,
αλλιώς (αν δηλαδή ο αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα 3 ανιχνεύσει μία ποσότητα φωτός ίση ή κάτω από την αριθμητική τιμή 40 **τότε**) το ρομπότ να στρίψει δεξιά,
Τοποθετούμε λοιπόν τις κατάλληλες εντολές σε κάθε μία από της δύο διαδρομές.

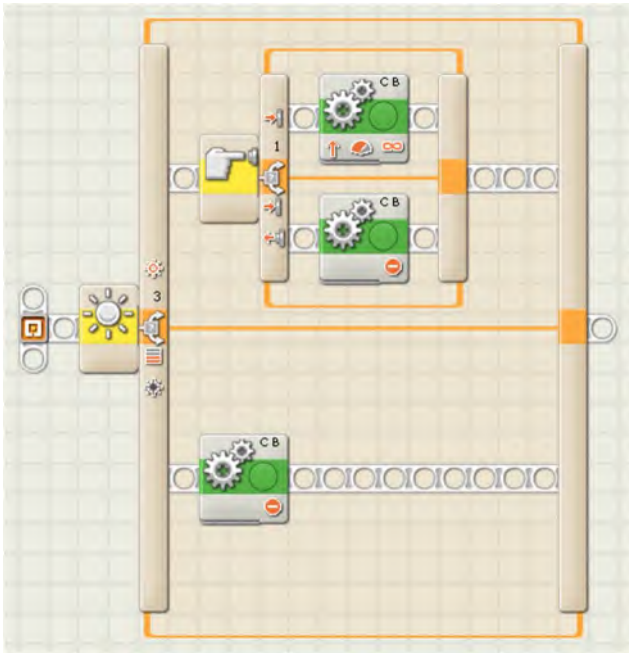


Δραστηριότητα 19η : διακλάδωση μέσα σε διακλάδωση

Για να προσδώσουμε πιο σύνθετες συμπεριφορές σε ένα ρομπότ μπορούμε να εισάγουμε στο πρόγραμμα μας μία δομή επιλογής μέσα σε μία άλλη δομή επιλογής.

Αυτό που θέλουμε είναι :

αν ο αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα 3 ανιχνεύσει μία ποσότητα φωτός πάνω από την αριθμητική τιμή 30 (δηλαδή υπάρχει φως) **τότε**
αν ο διακόπτης του αισθητήρα αφής που είναι συνδεδεμένος στη θύρα 1 βρεθεί πατημένος **τότε** το ρομπότ μας να σταματήσει
αλλιώς (αν δηλαδή ο διακόπτης του αισθητήρα αφής που είναι συνδεδεμένος στη θύρα 1 βρεθεί απελευθερωμένος **τότε**) το ρομπότ μας να κινηθεί προς τα εμπρός,
αλλιώς (αν δηλαδή ο αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα 3 ανιχνεύσει μία ποσότητα φωτός ίση ή κάτω από την αριθμητική τιμή 30 **τότε**) το ρομπότ να σταματήσει.



Με το παραπάνω συνδυασμό πετυχαίνουμε το ρομπότ μας να κινηθεί μόνο όταν υπάρχει ικανοποιητικό φως στο περιβάλλον που δραστηριοποιείται και αν ο διακόπτης στον αισθητήρα αφήσ βρεθεί πατημένος τη στιγμή του ελέγχου.

Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τις δομές επιλογής θα τον βρείτε στο σύνδεσμο :

[Δομές Επιλογής](#)

Λίγα λόγια για τη δομή επανάληψης (αέναο βρόχο)

Συχνά μέσα στο πρόγραμμα χρειάζεται να πούμε στο ρομπότ μας να επαναλάβει μια συγκεκριμένη διαδικασία περισσότερες από μία φορές.

Για να το πετύχουμε αυτό χρησιμοποιούμε μία δομή επανάληψης.



Δραστηριότητα 20η : εκκίνηση με αισθητήρα ήχου και σταμάτημα με αισθητήρα υπερήχων και επανάληψη

Στη δραστηριότητα 11 προγραμματίσαμε το ρομπότ μας να ξεκινήσει όταν χτυπήσουμε παλαμάκια και να σταματήσει όταν πλησιάσει σε ένα εμπόδιο (πχ βάλουμε το χέρι μας μπροστά από τον αισθητήρα υπερήχων). Τις ενέργειες αυτές το ρομπότ θα τις εκτελέσει μία φορά και το πρόγραμμα θα τερματιστεί.

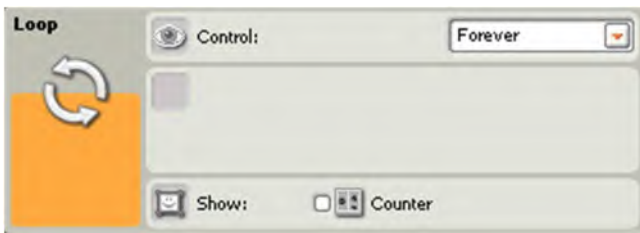
Για να υιοθετήσει το ρομπότ μας τη συγκεκριμένη συμπεριφορά μόνιμα, ώστε να αντιδρά κάθε φορά στα ερεθίσματα που θα δέχεται με τον κατάλληλο τρόπο και όχι μόνο μία και μοναδική φορά, μπορούμε να τοποθετήσουμε τις ενέργειες που επιθυμούμε να επαναληφθούν στο εσωτερικό μιας εντολής επανάληψης

Το εικονίδιο για τη δομή επανάληψης βρίσκεται στο υπομενού για δομές **Flow** και έχει ως εικόνα δύο βέλη τα οποία σχηματίζουν ένα κύκλο. Μέσα στο εικονίδιο της εντολής επανάληψης τοποθετούμε τις εντολές που είχαμε χρησιμοποιήσει στη δραστηριότητα 11.

Ένας βολικός τρόπος είναι να ανοίξουμε τη δραστηριότητα 11 και αφού εισάγουμε μία εντολή επανάληψης στην αρχή του προγράμματος να επιλέξουμε όλα τα εικονίδια των εντολών και να τα σύρουμε μέσα στο εσωτερικό του εικονιδίου της εντολής επανάληψης.



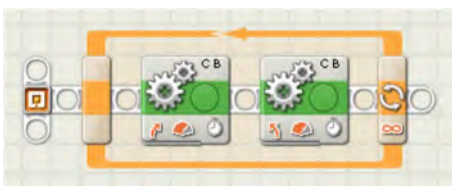
Στον πίνακα ρυθμίσεων της εντολής αυτής δεν χρειάζεται να πειράξουμε καμία ρύθμιση.



Με το πρόγραμμα αυτό κάθε φορά που το ρομπότ μας είναι ακίνητο και κτυπάμε παλαμάκια θα ξεκινάει να κινείται και κάθε φορά που κινείται και τοποθετούμε το χέρι μας μπροστά στον αισθητήρα υπερήχων θα σταματάει. Η εκτέλεση του προγράμματος αυτού δε θα σταματήσει παρά μόνο όταν εμείς θελήσουμε να διακόψουμε το πρόγραμμα αυτό.

Δραστηριότητα 21η : κίνηση φιδωτή

Το όχημα-ρομπότ κινείται φιδωτά, δηλαδή στρίβει μία αριστερά και μία δεξιά διαγράφοντας καμπύλες γραμμές.



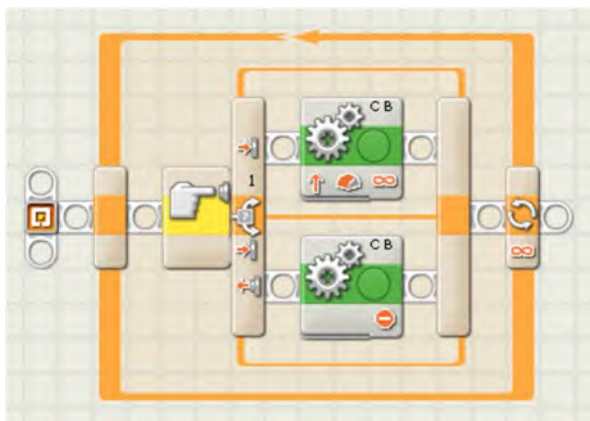
Δραστηριότητα 22η : ξεκίνημα και σταμάτημα με αισθητήρα αφής και επανάληψη

Το όχημα-ρομπότ κινείται προς τα εμπρός όταν πατιέται ο διακόπτης του αισθητήρα αφής και να σταματάει όταν ο διακόπτης του αισθητήρα αφής πατιέται ξανά.



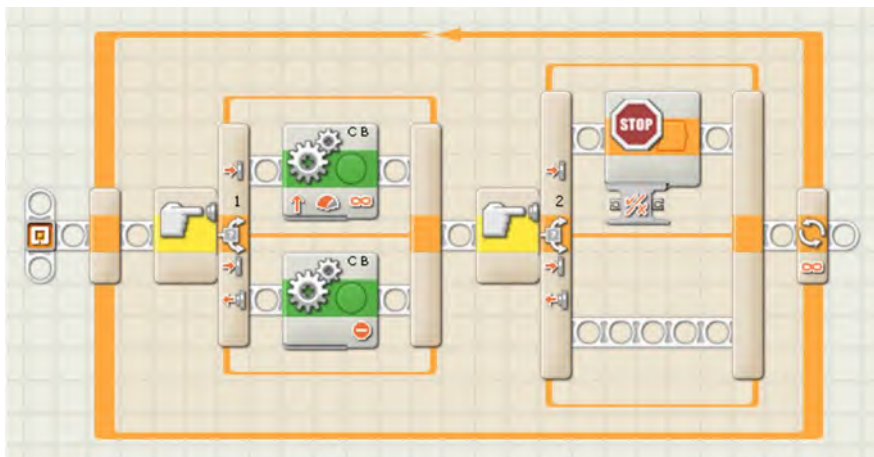
Δραστηριότητα 23η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη

Το όχημα-ρομπότ κινείται προς τα εμπρός αν ο διακόπτης αφής είναι πατημένος και να σταματάει αν ο αισθητήρας αφής είναι απελευθερωμένος.



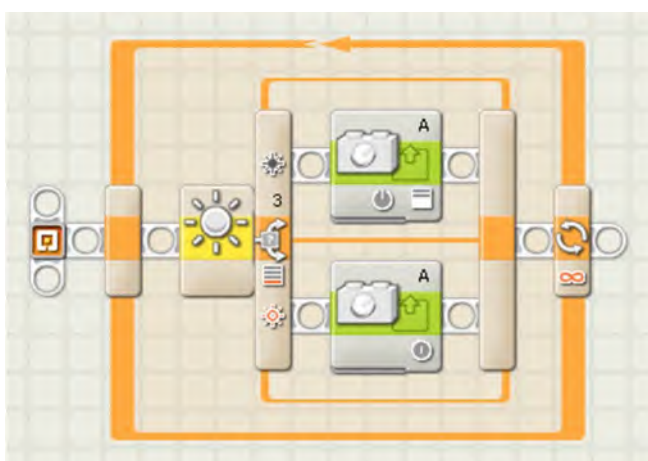
Δραστηριότητα 24η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη και με δυνατότητα τερματισμού

Το όχημα-ρομπότ κινείται προς τα εμπρός αν ο διακόπτης του αισθητήρα αφής 1 είναι πατημένος και να σταματάει αν ο διακόπτης του αισθητήρα αφής 1 είναι απελευθερωμένος. Το πρόγραμμα αυτό τερματίζει με το πάτημα του διακόπτη του αισθητήρα αφής 2.



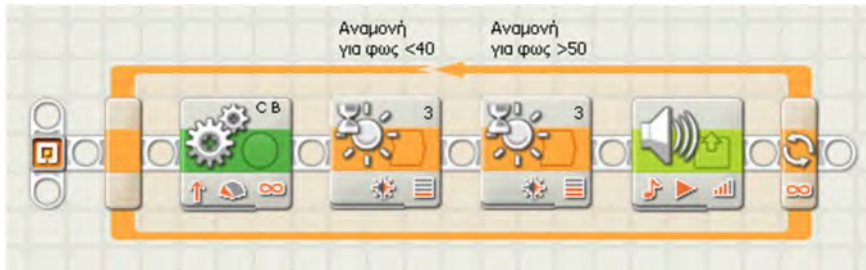
Δραστηριότητα 25η : υλοποίηση συναγερμού

Αν η ποσότητα φωτός που ανιχνεύεται από έναν αισθητήρα φωτός πέσει κάτω από την τιμή 45, το NXT ανάβει ένα λαμπτήρα.



Δραστηριότητα 26η : κίνηση μπροστά και ήχος για κάθε μαύρη γραμμή

Το όχημα-ρομπότ κινείται προς τα μπροστά και κάθε φορά που περνάει από μία μαύρη γραμμή παίζει ένα ήχο.



Σημείωση : το όχημα πρέπει να κινείται αργά για να προλαβαίνει ο αισθητήρας φωτός να ανιχνεύσει τις μαύρες γραμμές. Επίσης πρέπει να περάσει τελείως μία γραμμή για να μετρήσει την επόμενη

Λίγα λόγια για τη δομή επανάληψης (n φορές)

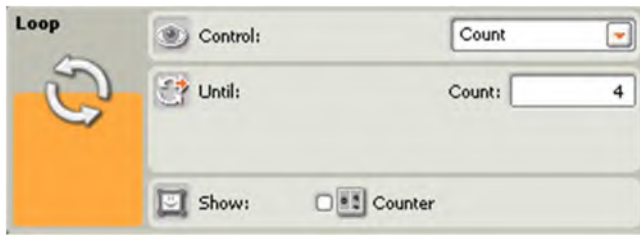
Πολλές φορές σε ένα πρόγραμμα θέλουμε μία σειρά ενεργειών να επαναληφθεί ένα συγκεκριμένο αριθμό φορές και όχι να επαναλαμβάνονται συνεχώς έως ότου διακόψουμε εμείς την εκτέλεση του προγράμματος. Στην περίπτωση αυτή γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων που πρέπει να γίνουν.

Δραστηριότητα 27η : τετράγωνο με δομή επανάληψης

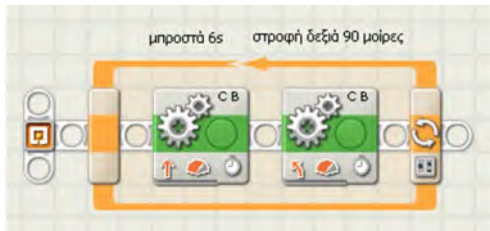
Θέλουμε το όχημα-ρομπότ μας να κινηθεί ακολουθώντας μία διαδρομή ή οποία να έχει σχήμα τετραγώνου, όπως και στη δραστηριότητα 7, χρησιμοποιώντας αυτή τη φορά τη δομή της επανάληψης.

Όπως είχαμε δει στη δραστηριότητα 7 για να κινηθεί το ρομπότ σε ένα τετράγωνο χρειάζεται να επαναλάβει την ίδια διαδικασία (κίνηση μπροστά και στροφή δεξιά) 4 φορές.

Ξεκινάμε πρόγραμμα μας εισάγοντας μία εντολή επανάληψης. Η εντολή αυτή όταν την τοποθετούμε στην περιοχή εργασίας, είναι ρυθμισμένη να επαναλαμβάνει το περιεχόμενο των εντολών που βρίσκονται στο εσωτερικό της για πάντα "Forever". Από τον πίνακα ρυθμίσεων μπορούμε να διαλέξουμε η επανάληψη να γίνει ένα αριθμό φορές αλλάζοντας τη επιλογή του ελέγχου της επανάληψης (Control) σε "Count". Πληκτρολογούμε στο αντίστοιχο πεδίο τον αριθμό 4.



Βλέπουμε λοιπόν ότι η δομή επανάληψης μας επιτρέπει να επαναλάβουμε ένα μέρος του προγράμματος μας, κάποιες συγκεκριμένες φορές. Με αυτό τον τρόπο δεν χρειάζεται να ξαναγράψουμε το ίδιο κομμάτι του προγράμματος μας (βρόχο) ξανά και ξανά.



Λίγα λόγια για τη δομή επανάληψης (μέχρι να)

Μία ακόμη περίπτωση επανάληψης έχουμε όταν μία διαδικασία επαναλαμβάνεται συνέχεια όχι όμως για πάντα, αλλά μέχρι να ικανοποιηθεί ένας στόχος ή όσο ισχύει μία συνθήκη. Στην περίπτωση αυτή δε γνωρίζουμε από πιο μπροστά τον αριθμό των επαναλήψεων που θα συμβούν.

Δραστηριότητα 28η : κίνηση μπροστά με σλάλομ και σταμάτημα με αισθητήρα αφής

Θέλουμε το όχημα-ρομπότ μας να κινείται κάνοντας σλάλομ (δηλαδή στρίβοντας μία δεξιά και μία αριστερά εναλλάξ) μέχρι να κτυπήσει σε ένα εμπόδιο.

Καταρχήν εισάγουμε το πρόγραμμα μας μία εντολή επανάληψης. Από τον πίνακα ρυθμίσεων μπορούμε να διαλέξουμε η επανάληψη να γίνεται μέχρι να ικανοποιηθεί μία συγκεκριμένη συνθήκη αλλάζοντας τη επιλογή του ελέγχου της επανάληψης (Control) σε "Sensor". Στην περίπτωση αυτή έχουμε τη δυνατότητα να επιλέξουμε οποιοδήποτε αισθητήρα. Εμείς εδώ χρειαζόμαστε τον αισθητήρα αφής, οπότε δεν πειράζουμε καμία άλλη ρύθμιση.

Στο εσωτερικό της δομής επανάληψης τοποθετούμε δύο εντολές μετακίνησης. Με την πρώτη, το ρομπότ μας εκτελεί ανοικτή στροφή προς τα δεξιά ενώ με τη δεύτερη ανοικτή στροφή προς

τα αριστερά. Στον πίνακα ρυθμίσεων των εντολών αυτών καθορίζουμε τη διάρκεια της κίνησης σε 2 δευτερόλεπτα. Επίσης στο πεδίο για την επόμενη ενέργεια (Next Action) επιλέγουμε την ολίσθηση "Coast" των κινητήρων αντί για το φρενάρισμα τους "Break", ώστε η κίνηση του ρομπότ μας να είναι πιο ομαλή χωρίς διακοπές, αφού εξάλλου στο πρόγραμμα αυτό δεν μας ενδιαφέρει η ακρίβεια των κινήσεων.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τις δομές επανάληψης θα τον βρείτε στο σύνδεσμο : [Δομές Επανάληψης](#)

Λίγα λόγια για τις παράλληλες διεργασίες

Μέχρι τώρα τα προγράμματα που δημιουργήσαμε περιείχαν μία μόνο διεργασία, μία σειρά δηλαδή εντολών. Το ρομπότ μας όμως μπορεί να προγραμματιστεί να εκτελεί περισσότερες από μία διαφορετικές εργασίες ταυτόχρονα. Μπορούμε δηλαδή μέσα σε ένα πρόγραμμα να έχουμε περισσότερες από μία διεργασίες, οι οποίες θα εκτελεστούν παράλληλα. Για να το πετύχουμε αυτό τοποθετούμε τη δεύτερη σειρά των εντολών στο πρόγραμμα μας και στη συνέχεια την ενώνουμε με το σημείο αρχής του προγράμματος, σύροντας το ποντίκι με κατεύθυνση από το σημείο αρχής προς το πρώτο εικονίδιο της δεύτερης σειράς των εντολών.

Αν θέλουμε η δεύτερη αυτή διεργασία να ξεκινάει από κάποιο σημείο ενδιάμεσα μέσα στο πρόγραμμα τότε για να δημιουργήσουμε έναν παράλληλο κλάδο σέρνουμε το ποντίκι από το σημείο της συνδετικής δοκού που μας ενδιαφέρει, πατώντας αρχικά το πλήκτρο SHIFT.

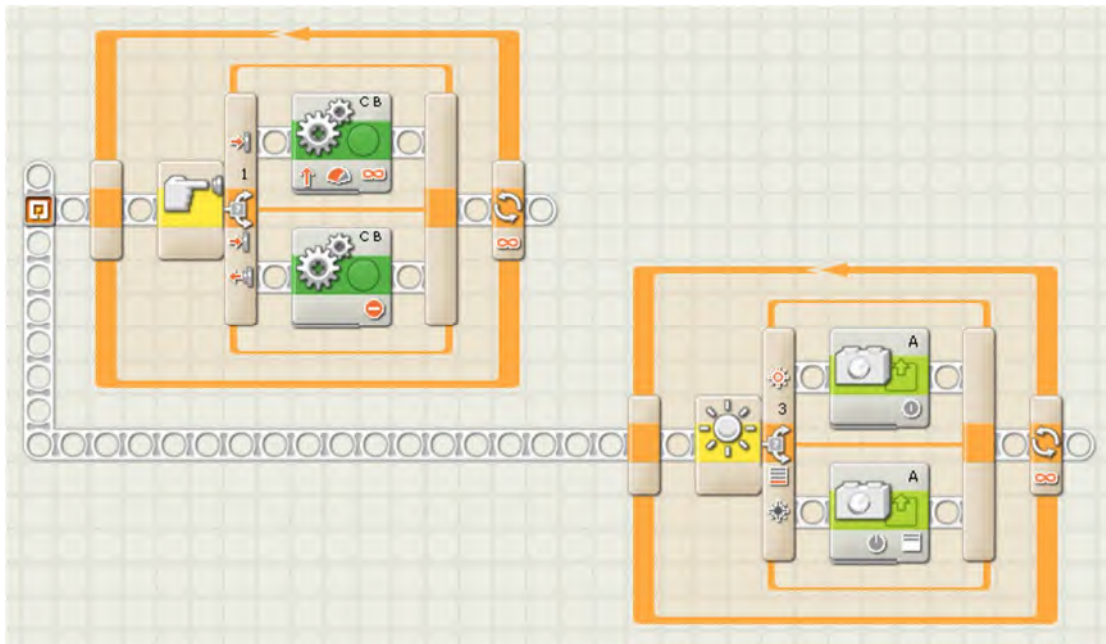
Στην περίπτωση παράλληλων διεργασιών χρειάζεται ιδιαίτερη προσοχή ώστε να μη συμβεί ποτέ το ρομπότ μας να εκτελέσει δύο εντολές ταυτόχρονα, οι οποίες να είναι αντιφατικές μεταξύ τους (πχ να στρίβει την ίδια στιγμή δεξιά και αριστερά ή να προχωρήσει μπροστά και πίσω ταυτόχρονα).

Δραστηριότητα 29η : παράλληλες διεργασίες

Το όχημα-ρομπότ εκτελεί 2 διεργασίες παράλληλα :

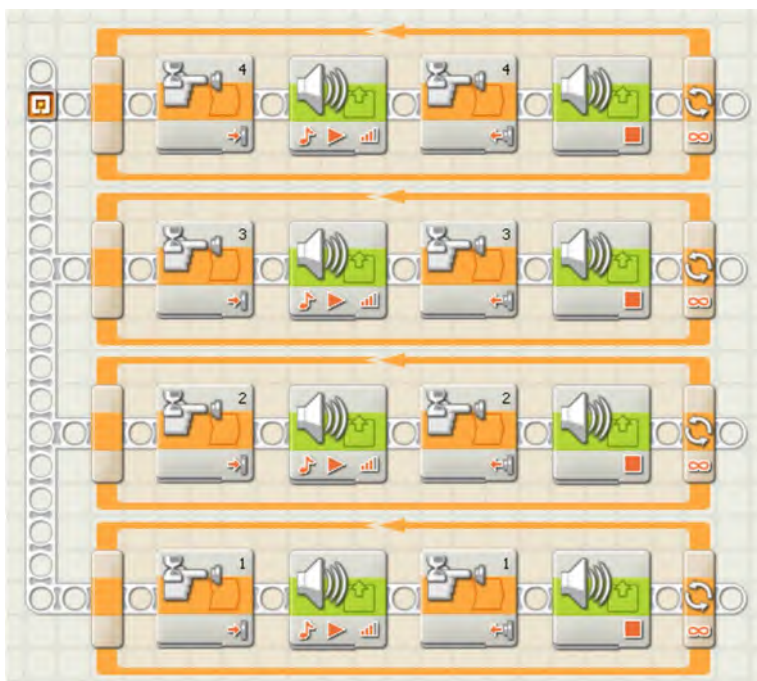
Στην πρώτη διεργασία το όχημα-ρομπότ κινείται προς τα μπροστά αν ο διακόπτης του αισθητήρα αφής είναι πατημένος και σταματάει αν ο διακόπτης του αισθητήρα αφής είναι απελευθερωμένος.

Στη δεύτερη διεργασία αν η ποσότητα φωτός που ανιχνεύεται από έναν αισθητήρα φωτός πέσει κάτω από την τιμή 45, ανάβει ένα λαμπάκι.



Δραστηριότητα 30η : παράλληλες διεργασίες

Το NXT αναπαράγει μία σειρά από 4 διαφορετικούς ήχους πατώντας τους διακόπτες από 4 αισθητήρες αφής.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τις παράλληλες διεργασίες θα τον βρείτε στο σύνδεσμο : [Παράλληλες Διεργασίες](#)

Λίγα λόγια για τους καταναμεμητές δεδομένων

Αν παρατηρήσουμε πιο προσεκτικά τα εικονίδια των εντολών θα διακρίνουμε στα περισσότερα από αυτά ότι στο κάτω μέρος τους υπάρχει μία εσοχή. Αν κάνουμε κλικ πάνω στην εσοχή αυτή εμφανίζεται μία προέκταση η οποία είναι γνωστή με το όνομα **καταναμεμητής δεδομένων (Data Hubs)**.

Ο καταναμεμητής δεδομένων περιέχει μία σειρά από σημεία σύνδεσης εισόδου στην αριστερή του πλευρά και μία αντίστοιχη σειρά από σημεία σύνδεσης εξόδου στη δεξιά του πλευρά. Με τη βοήθεια αυτών των σημείων σύνδεσης μπορούμε να στείλουμε πληροφορίες από μία εντολή σε μία άλλη. Για να το πετύχουμε αυτό αρκεί να καλωδιώσουμε ένα σημείο σύνδεσης εξόδου μίας εντολής με το αντίστοιχο σημείο σύνδεσης εισόδου μίας άλλης εντολής.

Με τη βοήθεια των καταναμεμητών δεδομένων μπορούμε να προσδώσουμε στο ρομπότ μας ακόμη πιο πολύπλοκες συμπεριφορές.

Δραστηριότητα 31η : προβολή τιμής αισθητήρα υπερήχων

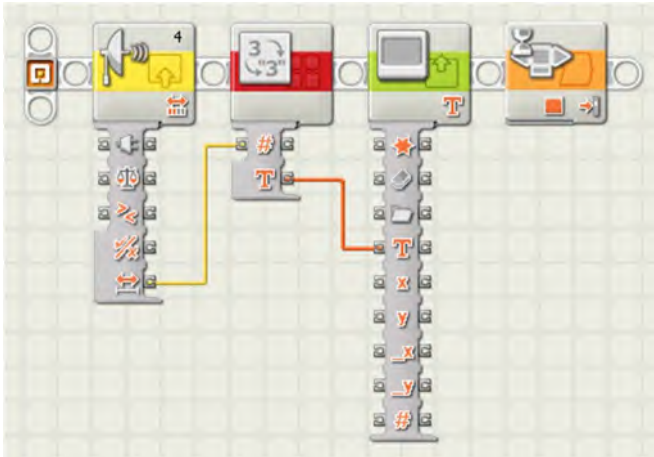
Θέλουμε να εμφανίσουμε στην οθόνη του NXT την τιμή από έναν αισθητήρα υπερήχων. Για το σκοπό αυτό θα χρησιμοποιήσουμε μία εντολή αισθητήρα υπερήχων.

Οι εντολές αισθητήρα έχουν την ίδια εμφάνιση με τις εντολές της αναμονής αισθητήρα, με τις εντολές αισθητήρα να έχουν κίτρινο χρώμα αντί για το πορτοκαλί των αντίστοιχων εντολών αναμονής αισθητήρα. Διαφέρουν όμως εντελώς ως προς τη λειτουργία τους. Μία εντολή αισθητήρα μας δίνει απλά μία πληροφορία για την κατάσταση του αντίστοιχου αισθητήρα την οποία μπορούμε να αξιοποιήσουμε μέσω κατάλληλων καλωδιώσεων των καταναμεμητών δεδομένων. Βέβαια όπως ήδη είδαμε, για τον αισθητήρα περιστροφής υπάρχει και η δυνατότητα να αρχικοποιήσουμε την τιμή του, ώστε να ξεκινήσει τη μέτρηση για την περιστροφή ενός κινητήρα από την αρχή

Από τον καταναμεμητή δεδομένων της εντολής αισθητήρα υπερήχων μπορούμε να λάβουμε την απόσταση [*Distance*] που έχει μετρηθεί από κάποιο αντικείμενο. Η απόσταση αυτή μας δίνεται ως μία αριθμητική τιμή. Το πρόβλημα μας σε αυτήν την περίπτωση είναι ότι η εντολή εμφάνισης δεν μπορεί να δεχθεί ως όρισμα της έναν αριθμό, αλλά αυτό πρέπει να είναι της μορφής κειμένου. Για να μπορέσουμε να εμφανίσουμε την τιμή της απόστασης που έχει μετρηθεί από τον αισθητήρα υπερήχων θα πρέπει να τη μετατρέψουμε προηγουμένως σε κείμενο [*Text*]. Για το σκοπό αυτό χρησιμοποιούμε την εντολή μετατροπής αριθμού σε κείμενο. Καλωδιώνοντας λοιπόν κατάλληλα τις εντολές που χρειαζόμαστε, θα εμφανιστεί στην οθόνη του NXT η τιμή της απόστασης που έχει μετρηθεί από τον αισθητήρα υπερήχων.

Για να συνδέσουμε δύο σημεία σύνδεσης από δύο εντολές, αφού εμφανίσουμε πρώτα τους κατανεμητές δεδομένων των εντολών αυτών, κάνουμε κλικ στο σημείο σύνδεσης εξόδου που μας ενδιαφέρει, της μίας εντολής και στη συνέχεια κάνουμε κλικ στο σημείο σύνδεσης εισόδου που θέλουμε, στην άλλη εντολή. Τα δύο σημεία σύνδεσης που πρόκειται να συνδέσουμε πρέπει να υποστηρίζουν παρόμοιο τύπο ορίσματος. Σε διαφορετική περίπτωση η καλωδίωση μας εμφανίζεται αχνή και διακεκομμένη.

Για να προλάβουμε να δούμε την τιμή αυτή τοποθετούμε στο πρόγραμμα μας μία εντολή αναμονής για πάτημα ενός πλήκτρου του NXT, αμέσως μετά την εντολή εμφάνισης.



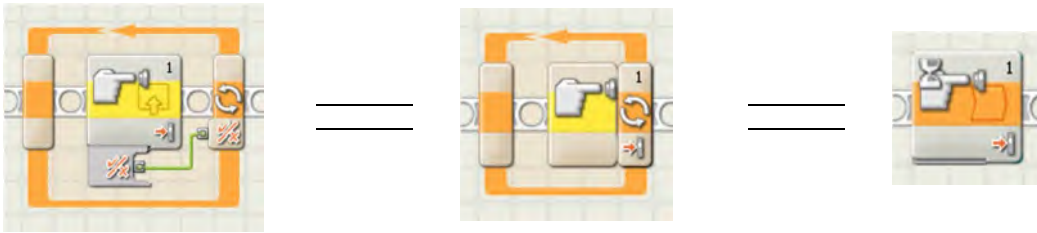
Λίγα λόγια για τη Δομή επανάληψης με κατανεμητή δεδομένων

Μία άλλη μορφή της εντολής επανάληψης, προκύπτει αν στον πίνακα ρυθμίσεων επιλέξουμε η επανάληψη να γίνεται μέχρις ότου καταφθάσει ένα αληθές (*true*) λογικό σήμα από ένα σημείο σύνδεσης εξόδου κάποιας εντολής. Για να υλοποιήσουμε αυτή τη μορφή της επανάληψης - που είναι παρόμοια με την εντολή επανάληψης μέχρι να συμβεί κάποιο γεγονός σε έναν αισθητήρα - αλλάζουμε τη επιλογή του ελέγχου της επανάληψης (*Control*) σε "*Logic*". Στην περίπτωση αυτή θα εμφανιστεί ένα σημείο σύνδεσης εισόδου στο τέλος του βρόχου επανάληψης. Στο σημείο αυτό καλωδιώνουμε το σημείο σύνδεσης εξόδου τύπου λογικού σήματος μιας άλλης εντολής.

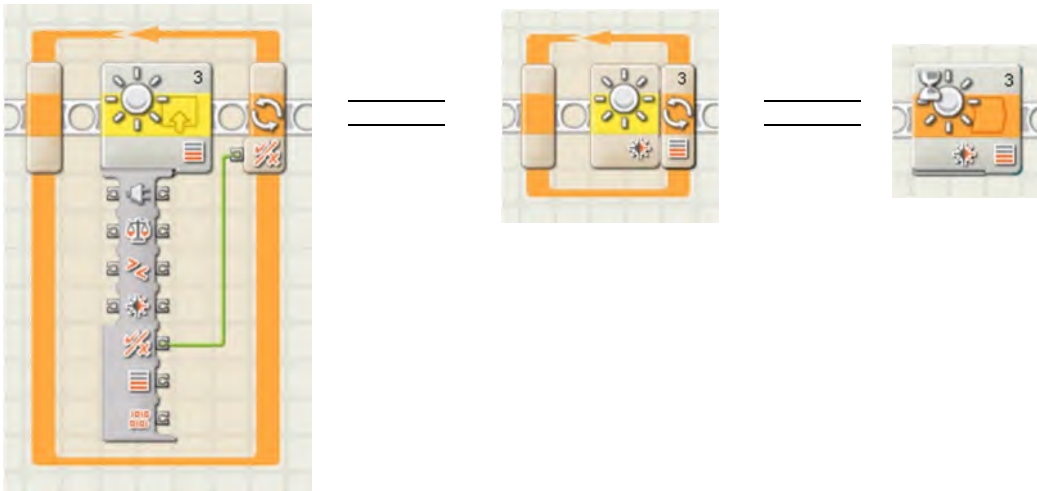


Αν στο βρόχο επανάληψης τοποθετήσουμε μόνο μία εντολή αισθητήρα και καλωδιώσουμε το σημείο σύνδεσης εξόδου του, τύπου λογικού σήματος, με το σημείο σύνδεσης εισόδου της εντολής επανάληψης δημιουργείται μία δομή ταυτόσημη με τη εντολή επανάληψης με τον αισθητήρα αυτόν. Με τον τρόπο αυτόν μία εντολή επανάληψης με αισθητήρα αφής μπορεί να συνταχθεί και ως :

Σενάρια Ρομποτικής με το Mindstorms NXT



και αντίστοιχα μία εντολή επανάληψης με αισθητήρα φωτός ως :

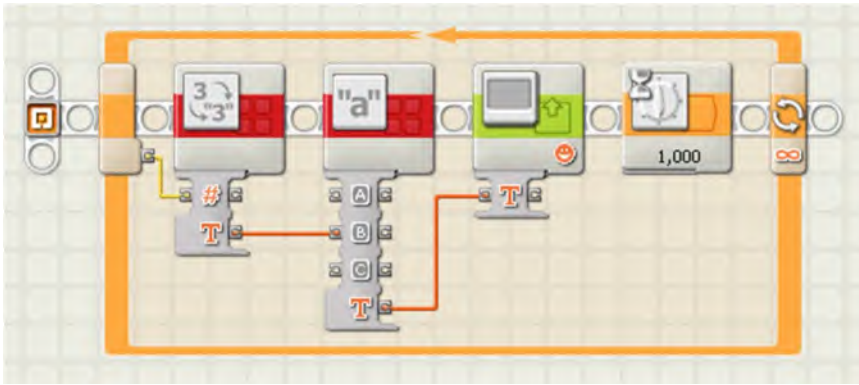


Η παραπάνω σύνταξη ισοδυναμεί με μία εντολή αναμονής.

Δραστηριότητα 32η : προβολή αριθμού επαναλήψεων

Αν στον πίνακα ρυθμίσεων μιας εντολής επανάληψης ενεργοποιήσουμε την εμφάνιση (Show) του μετρητή "Counter" θα εμφανιστεί ένα σημείο σύνδεσης εξόδου, στην αρχή του βρόχου της εντολής επανάληψης. Από αυτό το σημείο σύνδεσης μπορούμε να λαμβάνουμε την πληροφορία για τον αριθμό των επαναλήψεων που έχουν συμβεί οποιαδήποτε στιγμή, και να τη αξιοποιήσουμε κατάλληλα καλωδιώνοντας το σημείο αυτό, με ένα σημείο σύνδεσης εισόδου μιας άλλης εντολής.

Με το παρακάτω πρόγραμμα μας μπορούμε να βλέπουμε στην οθόνη του NXT τον αριθμό της επανάληψης που εκτελείται οποιαδήποτε στιγμή.



Στο πρόγραμμα αυτό έχουμε συμπεριλάβει και μία εντολή συνένωσης κειμένου στην οποία έχουμε πληκτρολογήσει στα πεδία κειμένου A και C τις φράσεις «Έχουν γίνει» και «επαναλήψεις» αντίστοιχα, ενώ ως όρισμα B εισάγεται μέσω καλωδίωσης στο σημείο σύνδεσης της εισόδου B ο αριθμός των επαναλήψεων. Τον αριθμό των επαναλήψεων τον έχουμε μετατρέψει πρώτα σε μορφή κειμένου, με τη βοήθεια μίας εντολής μετατροπής αριθμού σε κείμενο.



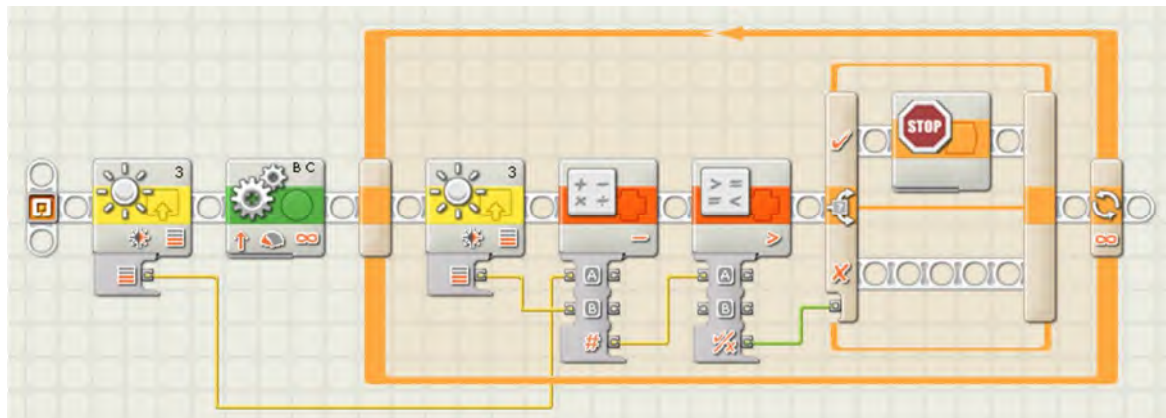
Λίγα λόγια για τη Δομή επιλογής με κατανεμητή δεδομένων

Μία άλλη μορφή της εντολής επιλογής, προκύπτει αν στον πίνακα ρυθμίσεων επιλέξουμε η συνθήκη της εντολής επιλογής να ελέγχεται (*Control*) από μία λογική τιμή "Value" αντί από έναν αισθητήρα. Στην περίπτωση αυτή θα εμφανιστεί ένα σημείο σύνδεσης εισόδου στην αρχή της εντολής επιλογής. Στο σημείο αυτό καλωδιώνουμε το σημείο σύνδεσης εξόδου, τύπου λογικού σήματος κάποιας άλλης εντολής.



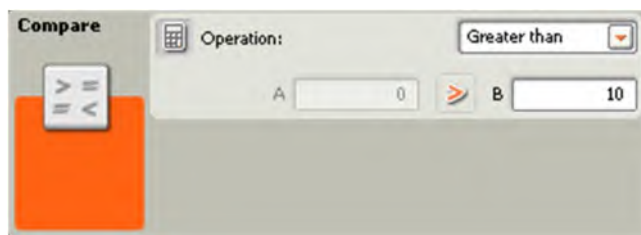
Δραστηριότητα 33η : κίνηση μπροστά μέχρι τη μαύρη γραμμή

Στη δραστηριότητα αυτή δίνουμε μία άλλη πιο πολύπλοκη σύνταξη για λύση του προβλήματος της δραστηριότητα 15. Στην παρακάτω δραστηριότητα το όχημα - ρομπότ κινείται προς τα μπροστά μέχρι να περάσει πάνω από μία μαύρη γραμμή οπότε και σταματάει.



Στο πρόγραμμα αυτό προτού ξεκινήσουμε το όχημα-ρομπότ, παίρνουμε μία μέτρηση για την ένταση του φωτός που αντανακλάται από την επιφάνεια στην οποία βρίσκεται εκείνη τη στιγμή το ρομπότ μας. Στη συνέχεια καθώς το ρομπότ μας κινείται, το πρόγραμμα εκτελεί συνεχώς νέες μετρήσεις για την ένταση του φωτός για κάθε σημείο από το οποίο διέρχεται το ρομπότ μας. Η τιμή αυτή συγκρίνεται με την αρχική, και αν η διαφορά του είναι μεγαλύτερη από 10 μονάδες τότε το ρομπότ θεωρεί ότι βρίσκεται πάνω σε μία μαύρη γραμμή και σταματάει να κινείται.

Για τη σύγκριση ανάμεσα στις δύο τιμές χρησιμοποιούμε μία εντολή μαθηματικών πράξεων με την οποία αφαιρούμε την τρέχουσα μέτρηση της ένταση του φωτός από την αρχική, και μία εντολή σύγκρισης με την οποία συγκρίνουμε αν το αποτέλεσμα της αφαίρεσης είναι μεγαλύτερο από 10 μονάδες.



Το λογικό αποτέλεσμα της σύγκρισης το δίνουμε ως είσοδο στο σημείο σύνδεσης εισόδου τύπου λογικού σήματος μίας εντολής επιλογής. Αν το λογικό σήμα είναι αληθές τότε το πρόγραμμα αυτό θα τερματιστεί.

Το πλεονέκτημα της λύσης αυτής, είναι ότι δεν στηριζόμαστε σε μία απόλυτη τιμή της έντασης του φωτός που μετράται από τον αισθητήρα φωτός για τη μαύρη γραμμή, αλλά στη σχετική τιμή

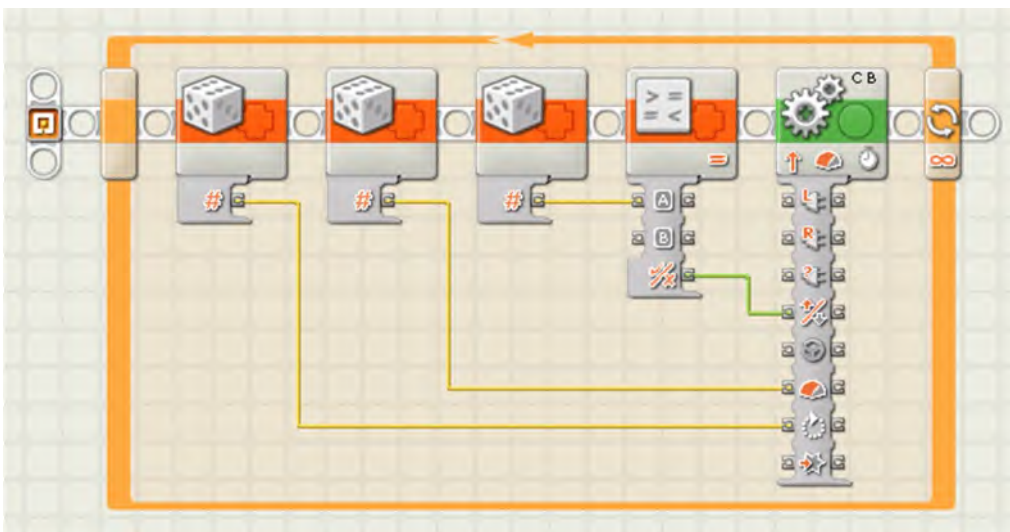
της σε σχέση με την τιμή που θα μετρηθεί στην υπόλοιπη επιφάνεια. Με τον τρόπο αυτόν το πρόγραμμα αυτό θα λειτουργήσει άψογα κάτω από διάφορες συνθήκες φωτισμού του περιβάλλοντος χώρου, χωρίς να χρειάζεται να γίνει ρεγουλάρισμα του αισθητήρα φωτός πριν από την εκτέλεση του προγράμματος. Στην περίπτωση της δραστηριότητας 15 για να λειτουργήσει σωστά το πρόγραμμα θα πρέπει να προσαρμόσουμε τον αισθητήρα φωτός στις εκάστοτε συνθήκες φωτισμού, με την πραγματοποίηση ρεγουλαρίσματος του αισθητήρα φωτός.

Δραστηριότητα 34η : Τυχαία κίνηση

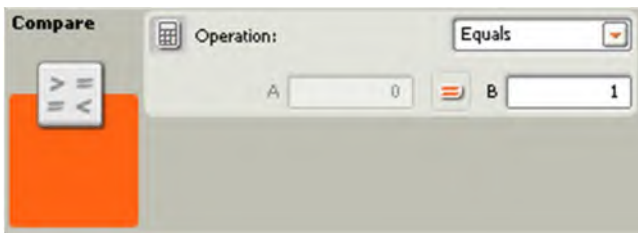
Θέλουμε να προσδώσουμε μία εντελώς τυχαία κίνηση στο ρομπότ μας. Για το σκοπό αυτό χρησιμοποιούμε μία εντολή τυχαίου αριθμού. Μία εντολή τυχαίου αριθμού παράγει έναν ακέραιο αριθμό με τυχαίο τρόπο, μέσα σε ένα διάστημα που καθορίζουμε δίνοντας ως ορίσματα μία ελάχιστη A και μία μέγιστη τιμή B.



Στην εντολή μετακίνησης τα ορίσματα της ταχύτητας περιστροφής, της διάρκειας της κίνησης καθώς και της φοράς της δίνονται δυναμικά μέσω καλωδίωσης των αντίστοιχων σημείων σύνδεσης εισόδων της εντολή μετακίνησης με τα σημεία σύνδεσης εξόδου από 3 εντολές τυχαίου αριθμού.



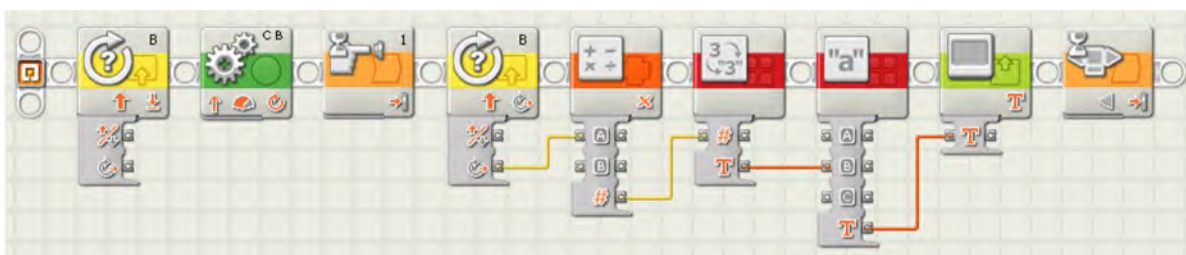
Η πρώτη εντολή τυχαίου αριθμού παράγει έναν αριθμό από το 0 ως το 3600 ο οποίος αντιστοιχεί στη χρονική διάρκεια κίνησης του ρομπότ σε δευτερόλεπτα, Η δεύτερη εντολή έναν αριθμό από το 0 ως το 100 ο οποίος αντιστοιχεί στην ταχύτητα περιστροφής των κινητήρων, Η τρίτη εντολή έναν αριθμό από το 0 ως το 1 ο οποίος αντιστοιχεί στη φορά κίνησης των κινητήρων. Η τρίτη εντολή μπορεί να μας δώσει μόνο έναν από τους αριθμούς 0 και 1. Το σημείο σύνδεσης εισόδου για τη φορά περιστροφής στην εντολή μετακίνησης πρέπει να είναι ένα λογικό σήμα. Για να μετατρέψουμε τους αριθμούς 0 και 1 σε λογικές τιμές, παρεμβάλουμε μία εντολή σύγκρισης στην οποία καθορίζουμε να γίνει σύγκριση του αριθμού που προκύπτει από την τρίτη εντολή τυχαίου αριθμού με τον αριθμό 1. Αν ο αριθμός αυτός είναι ο 1 παράγεται το αληθές λογικό σήμα (*true*) ενώ αν είναι ο 0 το ψευδές λογικό σήμα (*false*).



Δραστηριότητα 35η : υπολογισμός απόστασης

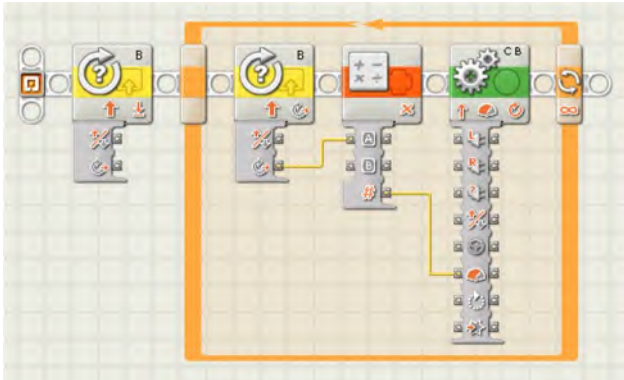
Θέλουμε να υπολογίσουμε την απόσταση που έχει μετακινηθεί ένα όχημα-ρομπότ τη στιγμή που θα πιέσουμε το διακόπτη ενός αισθητήρα αφής.

Για να το κατορθώσουμε αυτό μηδενίζουμε καταρχήν την τιμή του αισθητήρα περιστροφής του κινητήρα στη θύρα εξόδου B. Η απόσταση που θα έχει διανύσει το ρομπότ υπολογίζεται από τον αριθμό N των περιστροφών του κινητήρα B χρησιμοποιώντας τον τύπο της περιμέτρου των τροχών του : $S = N * 2\pi R$, όπου R ή ακτίνα των τροχών του ρομπότ. Στη δική μας περίπτωση $S = N * 12 \text{ cm}$. Το αποτέλεσμα της πράξης του πολλαπλασιασμού της περιστροφής του κινητήρα B με το 12 εμφανίζεται στην οθόνη του NXT.



Δραστηριότητα 36η : αντίσταση στη μετακίνηση

Όταν πάμε να μετακινήσουμε με τα χέρια μας ένα όχημα-ρομπότ αυτό αντιστέκεται και παραμένει στην αρχική του θέση.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τους κατανεμητές θα τον βρείτε στο σύνδεσμο :
[Κατανεμητές](#)

Λίγα λόγια για τις μεταβλητές

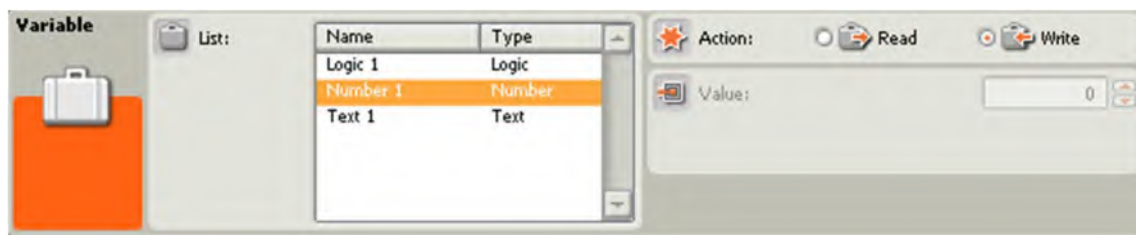
Μια **μεταβλητή (Container)** περιέχει έναν αριθμό τον οποίο δε γνωρίζουμε μέχρι τη στιγμή που θα εκτελεσθεί το πρόγραμμα μας. Αυτός μπορεί να είναι η τιμή από ένα αισθητήρα, ή για παράδειγμα πόσες φορές έχει πατηθεί ο διακόπτης του αισθητήρα αφής, το χρονικό διάστημα που πέρασε ανάμεσα σε δύο συγκεκριμένα γεγονότα ή μία άλλη τιμή η οποία δε μας είναι γνωστή μέχρι και τη στιγμή που θα ξεκινήσει το πρόγραμμα μας.

Δραστηριότητα 37η : μέτρηση αριθμού πατημάτων αισθητήρα αφής

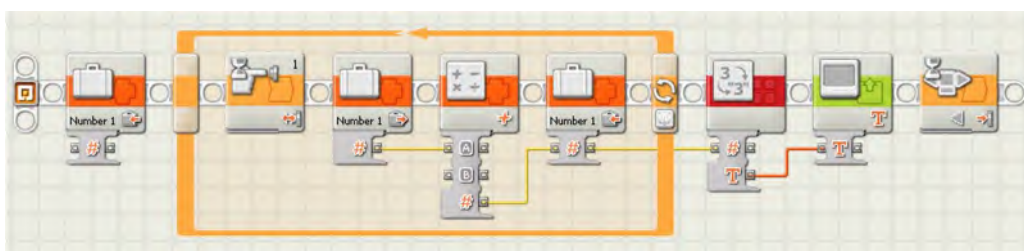
Θέλουμε να υπολογίσουμε τον αριθμό που θα έχει πατηθεί ο διακόπτης ενός αισθητήρα αφής μέσα σε ένα χρονικό διάστημα 60 δευτερόλεπτα.

α) Για να το πετύχουμε αυτό χρησιμοποιούμε μία εντολή μεταβλητής που δέχεται αριθμητικές τιμές. Η μεταβλητή αυτή είναι ήδη δηλωμένη στο πρόγραμμα μας ως number 1.

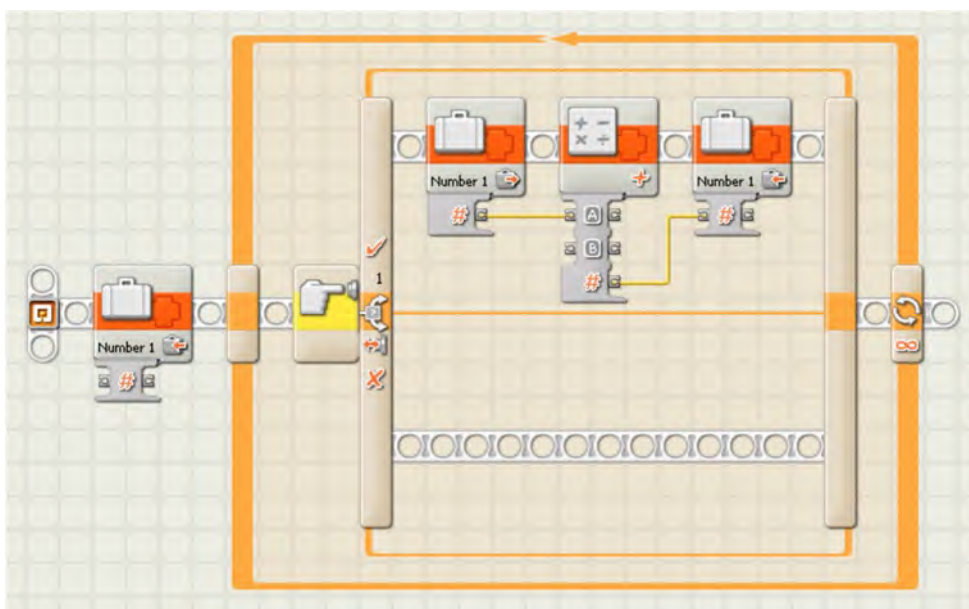
Σενάρια Ρομποτικής με το Mindstorms NXT



Στο πρόγραμμα αυτό χρησιμοποιούμε τη μεταβλητή σαν ένα μετρητή. Αρχικά δίνουμε στη μεταβλητή μας την τιμή μηδέν και στη συνέχεια μέσα σε ένα βρόχο επανάληψης, για κάθε πάτημα και απελευθέρωση του διακόπτη του αισθητήρα αφής αυξάνουμε την τιμή της μεταβλητής κατά μία μονάδα. Με την ολοκλήρωση του χρονικού διαστήματος των 60 δευτερολέπτων που έχουμε ορίσει στην εντολή επανάληψης η τελική τιμή της μεταβλητής εμφανίζεται στην οθόνη του NXT.

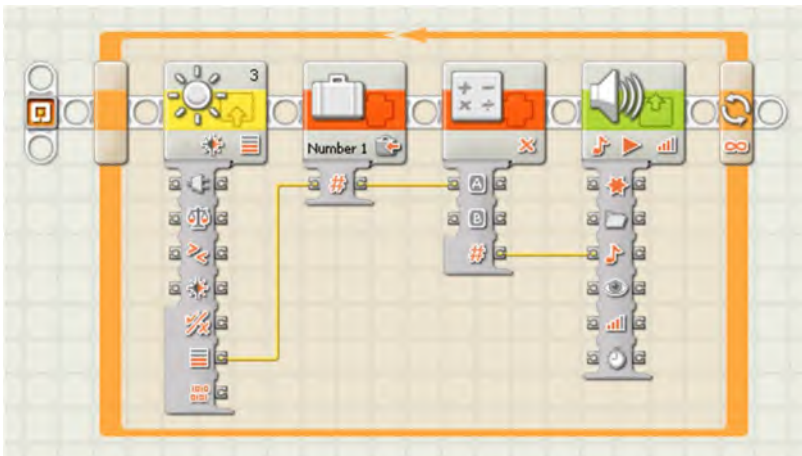


β. Εναλλακτικά μπορούμε να υλοποιήσουμε και το παρακάτω πρόγραμμα στο οποίο κάθε φορά που πατάμε και απελευθερώνουμε το διακόπτη του αισθητήρα αφής, η τιμή του μετρητή αυξάνεται κατά μία μονάδα.



Δραστηριότητα 38η : μετατροπή φωτεινότητας σε νότες (sequencer)

Το παρακάτω πρόγραμμα αναπαράγει μουσικές νότες μετατρέποντας τις τιμές που διαβάζει ένας αισθητήρας φωτός σε ηχητικές συχνότητες. Αρχικά γίνεται ανάγνωση της τρέχουσας τιμής της έντασης του φωτός από μία εντολή αισθητήρα φωτός και στη συνέχεια αφού πρώτα αποθηκεύσουμε την τιμή αυτής στη μεταβλητή Number 1, την πολλαπλασιάζουμε με τον αριθμό 100 με μία εντολή μαθηματικών πράξεων. Το αποτέλεσμα του πολλαπλασιασμού αυτού το δίνουμε ως είσοδο στην εντολή αναπαραγωγής ήχου, καλωδιώνοντας το σημείο σύνδεσης εξόδου της εντολής μαθηματικών πράξεων με το σημείο σύνδεσης εισόδου που καθορίζει τη συχνότητα του ήχου που θα αναπαραχθεί από την εντολή αναπαραγωγής ήχου.

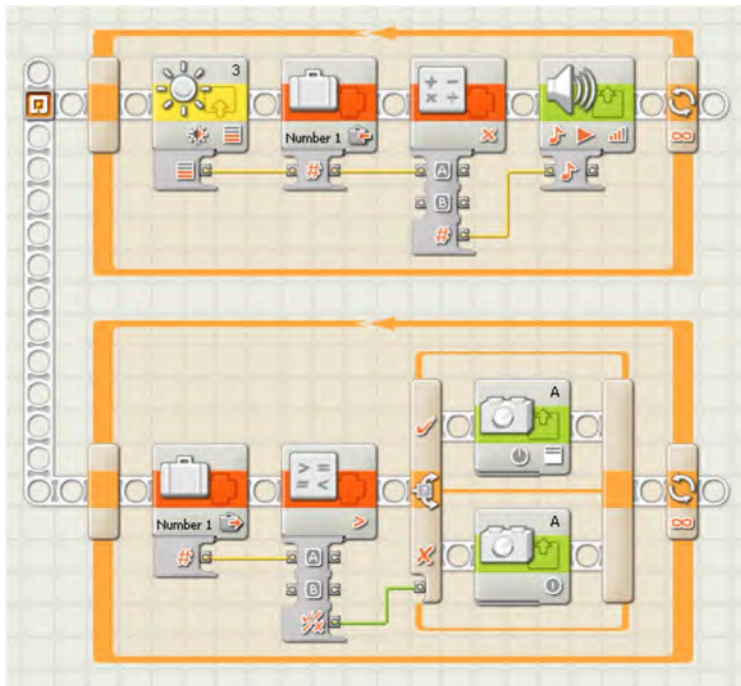


Σημείωση : Το πρόγραμμα αυτό θα λειτουργήσει άψογα και χωρίς την ύπαρξη της μεταβλητής. Η μεταβλητή θα ήταν χρήσιμη μόνο στην περίπτωση που υπήρχε κάποιο άλλο τμήμα στο πρόγραμμα μας, στο οποίο θα έπρεπε να χρησιμοποιήσουμε την τιμή της έντασης που μετρήθηκε από την εντολή ανάγνωσης του αισθητήρα φωτός. Μία μεταβλητή τη χρησιμοποιούμε όταν θέλουμε να αποθηκεύσουμε μία τιμή, την οποία θα τη χρησιμοποιήσουμε αργότερα ή να μεταφέρουμε μία τιμή από ένα σημείο του προγράμματος σε άλλο

Δραστηριότητα 39η : μετατροπή φωτεινότητας σε νότες και διαχωρισμός

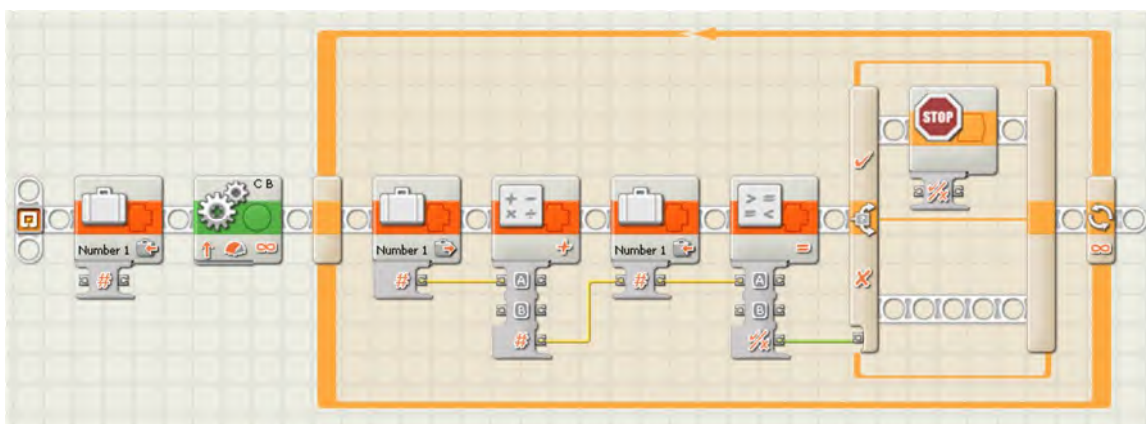
Στο παρακάτω πρόγραμμα επεκτείνοντας το προηγούμενο πρόγραμμα, στην περίπτωση που η συχνότητα που παράγεται από την εντολή ήχου είναι μεγαλύτερη από $40 \times 100 = 4 \text{ KHz}$, ανάβει ένας λαμπτήρας. Για να το πετύχουμε αυτό δημιουργούμε έναν παράλληλο κλάδο στο οποίο η τιμή της μεταβλητής συγκρίνεται με τον αριθμό 40 και στην περίπτωση που αυτή είναι μεγαλύτερη από το 40 ανάβει ο λαμπτήρας στη θύρα εξόδου A. Διαφορετικά ο λαμπτήρας σβήνει.

Σενάρια Ρομποτικής με το Mindstorms NXT



Δραστηριότητα 40η : αναμονή για μετρητή

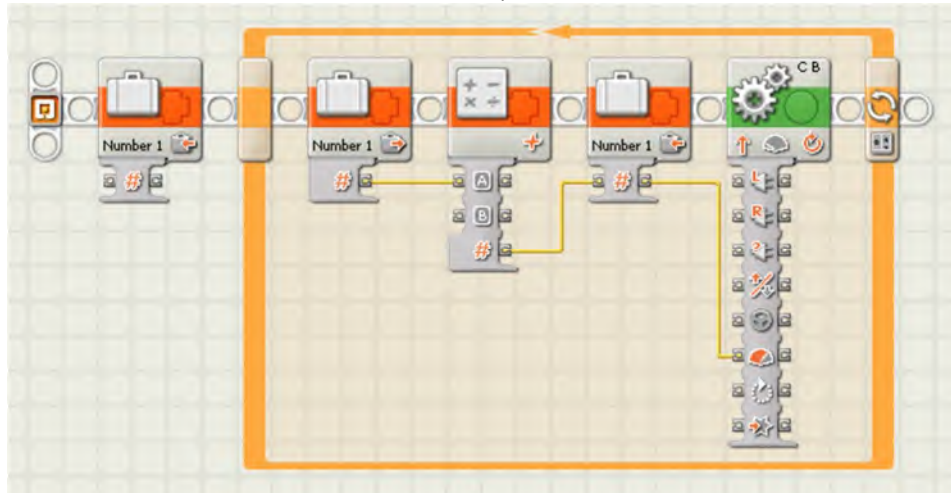
Το όχημα-ρομπότ ξεκινάει να κινείται μπροστά και σταματάει όταν ένας μετρητής φτάσει από το 0 στο 100.



Δραστηριότητα 41η : κίνηση με επιτάχυνση

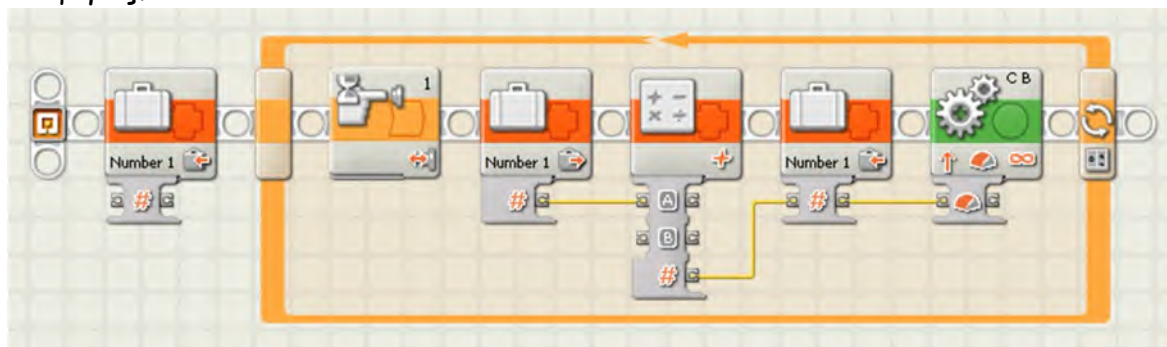
Στο παρακάτω πρόγραμμα το όχημα-ρομπότ ξεκινάει να κινείται μπροστά με επίπεδο ισχύος του κινητήρα 10 μονάδες και αυξάνει συνεχώς την ταχύτητα του κατά 10 μονάδες μετά από κάθε πλήρη περιστροφή, μέχρι να πιάσει την τελική ταχύτητα του. Αυτό θα συμβεί μετά από 10 επαναλήψεις. Στην αρχή του προγράμματος μηδενίζουμε τη μεταβλητή και μέσα στο βρόχο της επανάληψης αυξάνουμε την τιμή της κατά 10 μονάδες. Την τρέχουσα τιμή της μεταβλητής την περνάμε με καλωδίωση στο σημείο σύνδεσης εισόδου για την ταχύτητα περιστροφής [Power] της εντολής μετακίνησης. Στην εντολή μετακίνησης η διάρκεια της κίνησης έχει οριστεί σε μία πλήρη περιστροφή.

Στην εντολή επανάληψης έχουμε ορίσει η επανάληψη να γίνει 10 φορές.



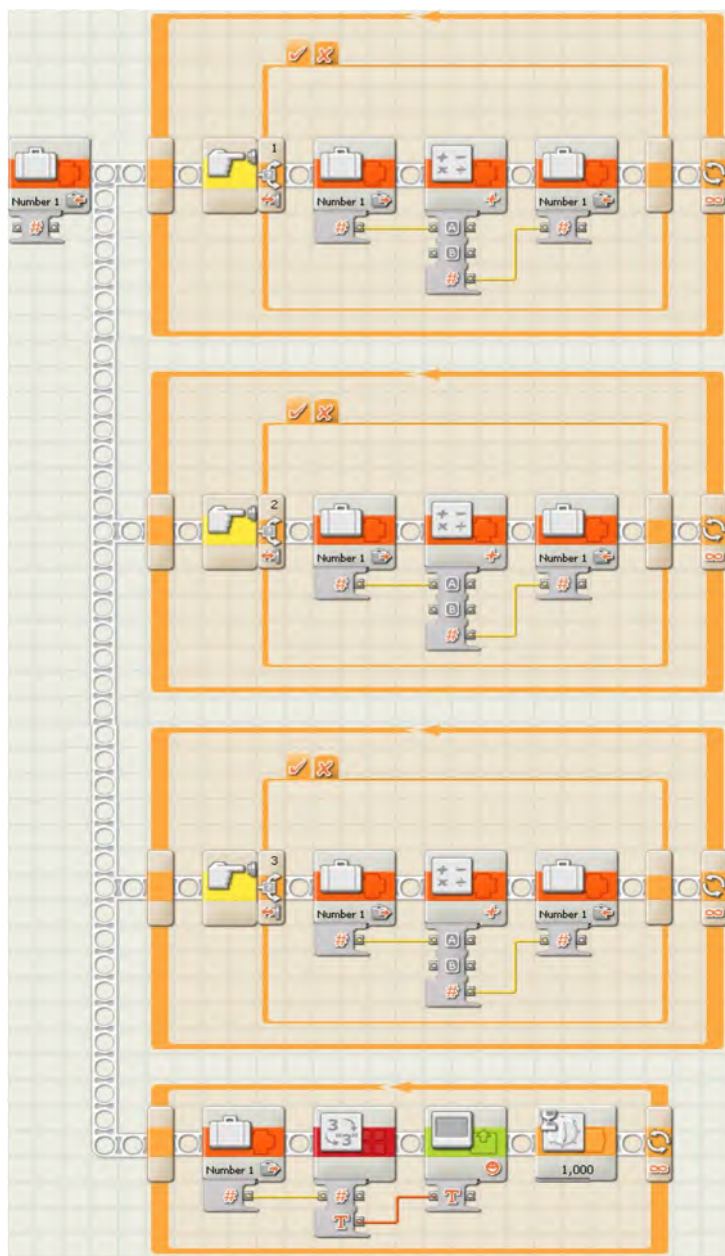
Δραστηριότητα 42η : αλλαγή ταχύτητας με διακόπτη αφής

Στο παρακάτω πρόγραμμα το όχημα-ρομπότ ξεκινάει να κινείται μπροστά με επίπεδο ισχύος του κινητήρα 10 μονάδες μετά από το πάτημα του διακόπτη ενός αισθητήρα αφής και κάθε φορά που πατιέται ο διακόπτης ξανά αυξάνει την ταχύτητα του κατά 10 μονάδες. Εναλλακτικά μπορούμε να πετύχουμε η αλλαγή της ταχύτητας να συμβαίνει μετά από το πάτημα του διακόπτη ενός αισθητήρα αφής. Στην εντολή επανάληψης έχουμε ορίσει η επανάληψη να γίνει 10 φορές.



Δραστηριότητα 43η : μέτρηση πατημάτων

Το NXT υπολογίζει έναν αριθμό ως εξής : κάθε φορά που πατιέται ο διακόπτης του αισθητήρα αφής 1 προστίθεται μία μονάδα, κάθε φορά που πατιέται ο διακόπτης του αισθητήρα αφής 2 προστίθεται μία δεκάδα και κάθε φορά που πατιέται ο διακόπτης του αισθητήρα αφής 3 προστίθεται μία εκατοντάδα. Ο αριθμός αυτός εμφανίζεται στην οθόνη του NXT.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τις μεταβλητές θα τον βρείτε στο σύνδεσμο :

[Μεταβλητές](#)

Λίγα λόγια για τους χρονομετρητές

Το εσωτερικό ρολόι του NXT μπορεί να χρησιμοποιηθεί ως ακόμα μία είσοδος μέσα στο πρόγραμμα μας. Οι χρονομετρητές (Timers) είναι ένας τρόπος να ελέγξουμε τη ροή του προγράμματος χρησιμοποιώντας την παράμετρο του χρόνου που πέρασε.

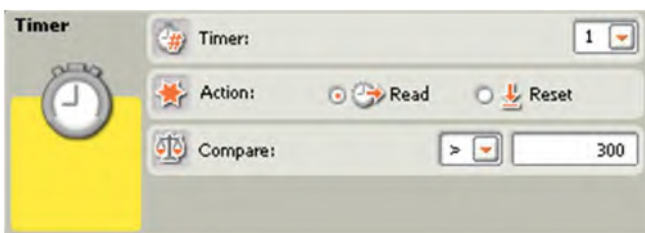
Τους χρονομετρητές μπορούμε να τους χρησιμοποιήσουμε για να βάλουμε το πρόγραμμα να περιμένει μέχρι να περάσει ένα καθορισμένο χρονικό διάστημα, προτού συνεχίσει με την εκτέλεση της επόμενης εντολής. Είναι σημαντικό να μηδενίζουμε ένα χρονομετρητή προτού να ξεκινήσουμε.

Η διαφορά τους με μία εντολή αναμονής για χρόνο είναι ότι τους χρονομετρητές μπορούμε να τους χρησιμοποιήσουμε για τη μέτρηση του χρόνου σε κάποιο σημείο ενός προγράμματος ενώ παράλληλα εκτελούνται κάποιες άλλες εντολές ενδιάμεσα.

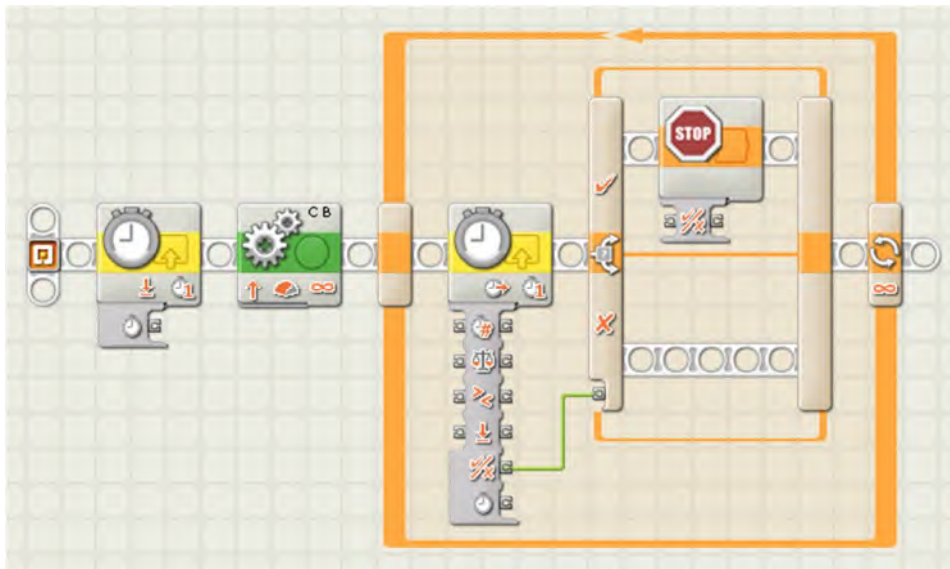
Δραστηριότητα 44η : αναμονή με χρονομετρητή

Στο παρακάτω πρόγραμμα το όχημα-ρομπότ κινείται προς τα μπροστά για 300 δευτερόλεπτα.

α) Για να το πετύχουμε αυτό χρησιμοποιούμε αντί για την εντολή αναμονής για χρόνο, μία εντολή χρονομετρητή. Αρχικά με μία εντολή χρονομετρητή μηδενίζουμε το χρονομετρητή 1. Στη συνέχεια αφού ξεκινήσουμε το ρομπότ μας, τοποθετούμε μέσα σε ένα βρόχο επανάληψης μία εντολή χρονομετρητή στην οποία ορίζουμε ως συνθήκη "Compare", ο χρόνος να ξεπεράσει τα 300 δευτερόλεπτα.



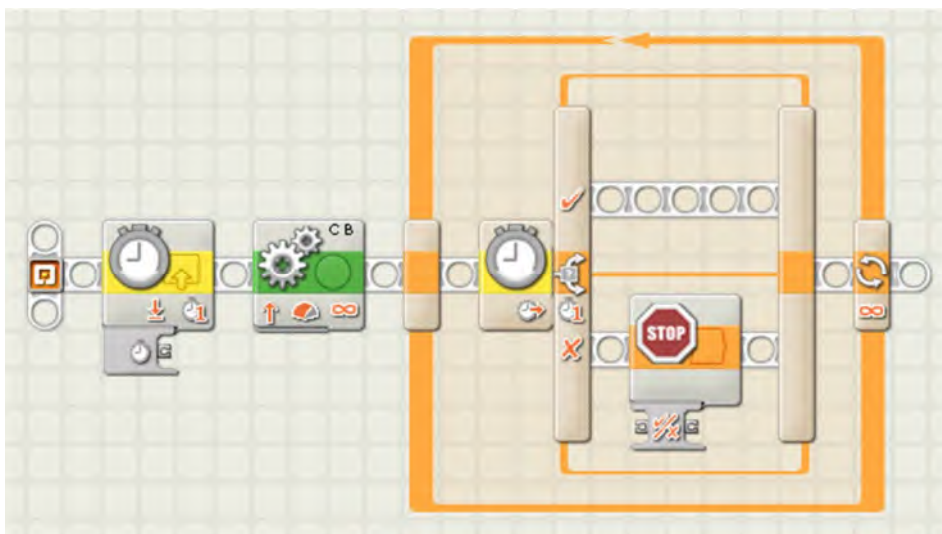
Όταν το ρολόι του χρονομετρητή μετρήσει 300 δευτερόλεπτα η εντολή χρονομετρητή αποστέλλει μέσω καλωδίωσης ένα αληθές λογικό σήμα (*true*) και ο βρόχος επανάληψης καθώς και το πρόγραμμα τερματίζονται.



β. Η παραπάνω δραστηριότητα μπορεί να απλοποιηθεί όπως φαίνεται στο παρακάτω πρόγραμμα. Στο βρόχο επανάληψης αυτή τη φορά έχουμε τοποθετήσει μία εντολή επιλογής χρονομετρητή στην οποία έχουμε ορίσει ως συνθήκη, μία σύγκριση με τον αριθμό 300.

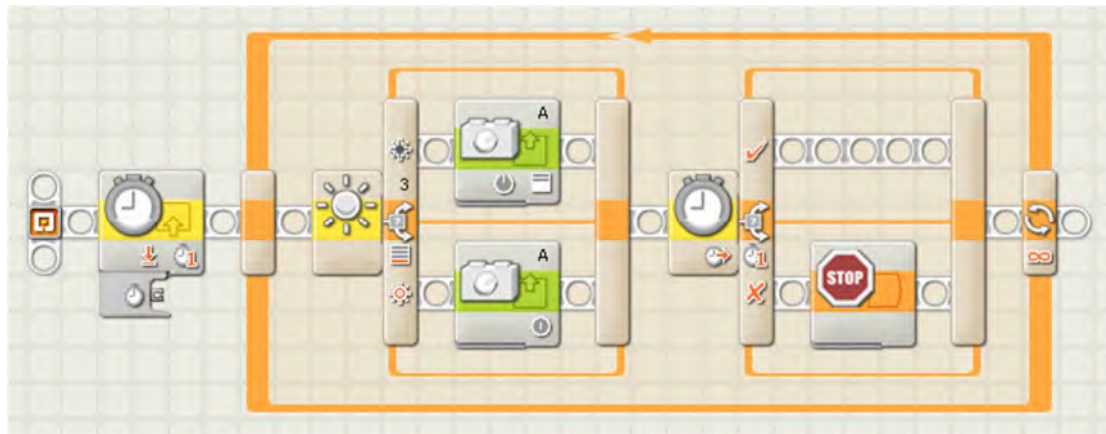


Όσο η τιμή του χρονομετρητή είναι μικρότερη από τα 300 δευτερόλεπτα το πρόγραμμα βρίσκεται σε αναμονή



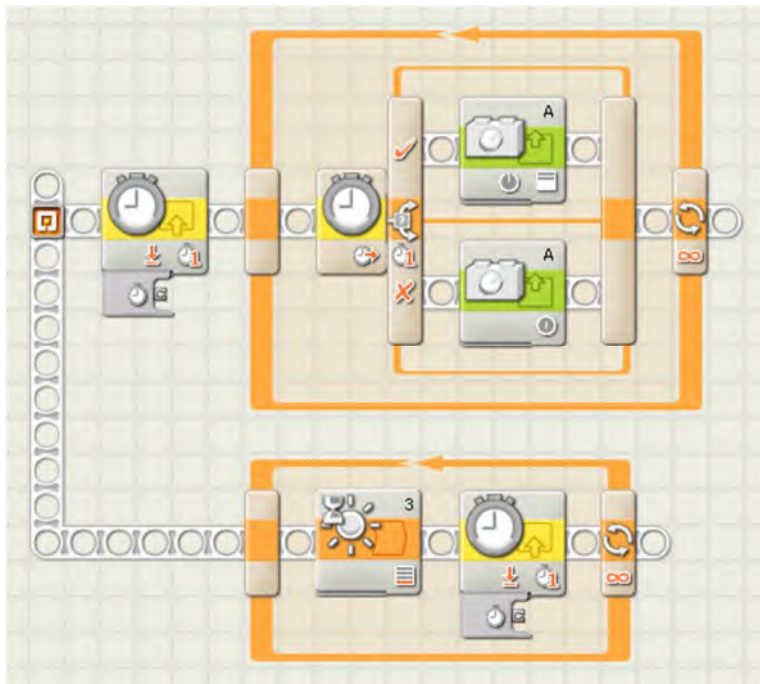
Δραστηριότητα 45η : έλεγχος λάμπας

Το NXT ελέγχει ένα λαμπτήρα Α με τη βοήθεια ενός αισθητήρα φωτός για χρόνο 20 δευτερόλεπτα. Όταν ο χρόνος που μετρίεται από ένα χρονομετρητή ξεπεράσει τα 20 δευτερόλεπτα το πρόγραμμα τερματίζεται.



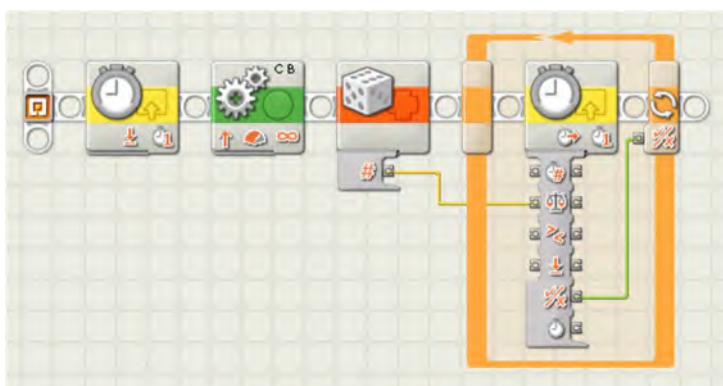
Δραστηριότητα 46η : έλεγχος λάμπας με χρονοκαθυστέρηση

Το NXT ανάβει ένα λαμπτήρα Α όταν έχει σκοτάδι για πάνω από 2 δευτερόλεπτα. Μας ενδιαφέρει ο λαμπτήρας Α να ανάψει, όχι αμέσως μόλις σκοτεινιάσει αλλά αφού έχουν περάσει δύο δευτερόλεπτα και εξακολουθεί να επικρατεί σκοτάδι. Αρχικά με μία εντολή χρονομετρητή μηδενίζουμε το χρονομετρητή 1. Με έναν παράλληλο κλάδο βάζουμε το πρόγραμμά μας να περιμένει μέχρι να αυξηθεί η ένταση του φωτός. Στην περίπτωση αυτή μηδενίζουμε ξανά το χρονομετρητή μας και τον αναγκάζουμε να ξεκινήσει μία νέα μέτρηση από την αρχή. Στην περίπτωση που υπάρχει συνέχεια επαρκές φως ο χρονομετρητής δεν θα καταφέρει να μετρήσει ποτέ μέχρι τα 2 δευτερόλεπτα. Μόνο αφού πέσει αρκετά η φωτεινότητα και παραμείνει στα ίδια επίπεδα για πάνω από 2 δευτερόλεπτα θα ικανοποιηθεί η συνθήκη της εντολής επιλογής με χρονομετρητή στον πρώτο κλάδο και θα ανάψει ο λαμπτήρας στη θύρα εισόδου Α. Από την άλλη μόλις ανέβει πάλι η ένταση του φωτός ο λαμπτήρας θα σβήσει αμέσως.



Δραστηριότητα 47η : αναμονή για τυχαίο χρόνο

Το NXT περιμένει μέχρι να ολοκληρωθεί ένα τυχαίο χρονικό διάστημα. Αρχικά με μία εντολή χρονομετρητή μηδενίζουμε το χρονομετρητή 1. Στη συνέχεια με μία εντολή τυχαίου αριθμού παράγεται ένας τυχαίος αριθμός. Όταν το ρολόι του χρονομετρητή φτάσει στην τιμή του τυχαίου αριθμού ο βρόχος επανάληψης καθώς και το πρόγραμμα τερματίζονται.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τους χρονομετρητές θα τον βρείτε στο σύνδεσμο :
[Χρονομετρητές](#)

Λίγα λόγια για την αποστολή και λήψη μηνυμάτων

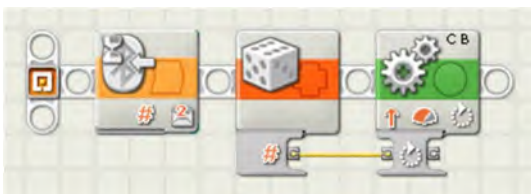
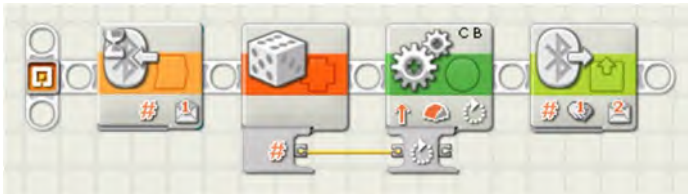
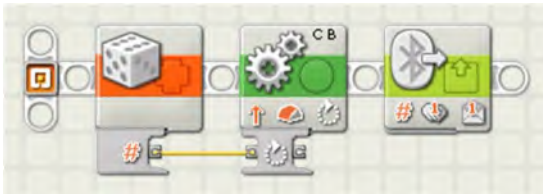
Οι μικροεπεξεργαστές NXT έχουν τη δυνατότητα να επικοινωνήσουν μεταξύ τους και να ανταλλάξουν μηνύματα τα οποία μπορούν να περιέχουν κάποιο κείμενο, έναν αριθμό ή ένα λογικό αριθμό. Η αποστολή και η λήψη των μηνυμάτων γίνεται μέσω της υπέρυθρης διασύνδεσης από ένα τούβλο NXT σε οποιοδήποτε άλλο τούβλο NXT που θα βρεθεί μέσα στην εμβέλεια του σήματος. Για να στείλουμε ένα μήνυμα από ένα ρομπότ NXT σε ένα άλλο χρησιμοποιούμε την εντολή της αποστολής μηνύματος. Για να λάβει κάποιο ρομπότ NXT ένα μήνυμα από ένα άλλο NXT χρησιμοποιούμε την εντολή της λήψης μηνύματος.

Με τη δυνατότητα αποστολής και λήψης μηνυμάτων μπορούμε ελέγξουμε τη συμπεριφορά ενός ρομπότ ασύρματα στέλνοντας μηνύματα με διάφορες πληροφορίες στις οποίες το ρομπότ θα ανταποκρίνεται αναλόγως.

Επίσης ένα ρομπότ μπορεί να ειδοποιήσει ένα άλλο για την ολοκλήρωση μίας ενέργειας, ώστε να αναλάβει με τη σειρά του κάποια δράση -για την οποία έπρεπε πρώτα να περιμένει την ολοκλήρωση μίας άλλης ενέργειας.

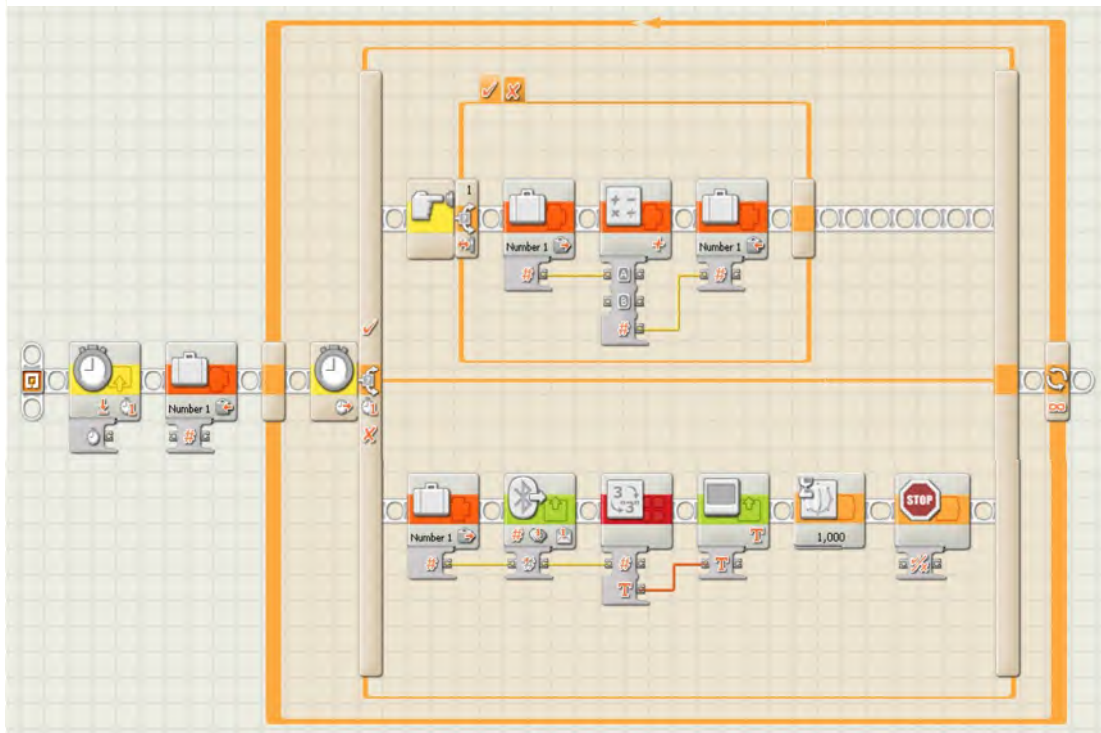
Δραστηριότητα 48η : σκυταλοδρομία

Έχουμε 3 οχήματα-ρομπότ. Μόλις τα θέσουμε σε λειτουργία το πρώτο από αυτά θα ξεκινήσει να κινείται για ένα τυχαίο χρονικό διάστημα ενώ τα άλλα δύο θα τεθούν σε κατάσταση αναμονής μέχρι να λάβουν ένα συγκεκριμένο μήνυμα. Το δεύτερο όχημα περιμένει να λάβει ένα μήνυμα στη γραμματοθυρίδα του 1 που θα περιέχει την αριθμητική τιμή 1. Το τρίτο όχημα περιμένει να λάβει ένα μήνυμα στη γραμματοθυρίδα του 1 που θα περιέχει την αριθμητική τιμή 2. Μόλις το πρώτο όχημα ολοκληρώσει την κίνηση του στέλνει, ένα μήνυμα από τη γραμματοθυρίδα 1 το οποίο έχει ως περιεχόμενο την αριθμητική τιμή 1. Το μήνυμα αυτό θα το λάβουν και τα δύο από τα υπόλοιπα οχήματα, αλλά μόνο το δεύτερο όχημα θα βγει από την κατάσταση αναμονής αφού μόνο για αυτό θα εκπληρωθούν τα κριτήρια του μηνύματος που έλαβε από το πρώτο όχημα. Με παρόμοιο τρόπο το δεύτερο όχημα θα ξεκινήσει να κινείται για ένα τυχαίο χρονικό διάστημα και μόλις ολοκληρώσει την κίνηση του ειδοποιεί το τρίτο όχημα να ξεκινήσει με τη σειρά του. Στο τέλος αφού το τρίτο όχημα λάβει και αυτό το μήνυμα που περιμένει, κινείται και αυτό με τη σειρά του για ένα τυχαίο χρονικό διάστημα.



Δραστηριότητα 49η : δημιουργία και αποστολή μηνύματος

Το NXT στέλνει ένα μήνυμα το οποίο περιλαμβάνει τον αριθμό των πατημάτων και απελευθερώσεων του διακόπτη ενός αισθητήρα αφής που πραγματοποιήθηκε μέσα σε ένα συγκεκριμένο χρονικό διάστημα. Αρχικά μηδενίζουμε το χρονομετρητή 1 και δίνουμε την τιμή 0 στη μεταβλητή Number 1. Στη συνέχεια όσο η τιμή του χρόνου στο χρονομετρητή είναι μικρότερη από τα 300 δευτερόλεπτα, ο αριθμός των πατημάτων και απελευθερώσεων του διακόπτη του αισθητήρα αφής, αποθηκεύεται στη μεταβλητή. Όταν ο χρόνος ξεπεράσει τα 300 δευτερόλεπτα η τιμή που βρίσκεται αποθηκευμένη στη μεταβλητή αποστέλλεται με μία εντολή αποστολής μηνύματος. Η τιμή αυτή εμφανίζεται στη συνέχεια στην οθόνη του NXT και το πρόγραμμα τερματίζεται.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για την αποστολή και λήψη μηνυμάτων θα τον βρείτε στο σύνδεσμο : [Μηνύματα](#)

Για περισσότερες πληροφορίες για τις εντολές του προγραμματιστικού περιβάλλοντος NXT-G ανατρέξτε στο ΠΑΡΑΡΤΗΜΑ Ι.



Ενότητα 6η

Σύνθετες Δραστηριότητες στο NXT-G

Α. Αποφυγή εμποδίων

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να παρακάμπτει τα εμπόδια που θα συναντήσει στην πορεία του.

Δραστηριότητα 1η : αποφυγή εμποδίου με 1 αισθητήρα αφής

Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας αφής.

Τον αισθητήρα αφής αυτόν τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Ανάλυση προβλήματος :

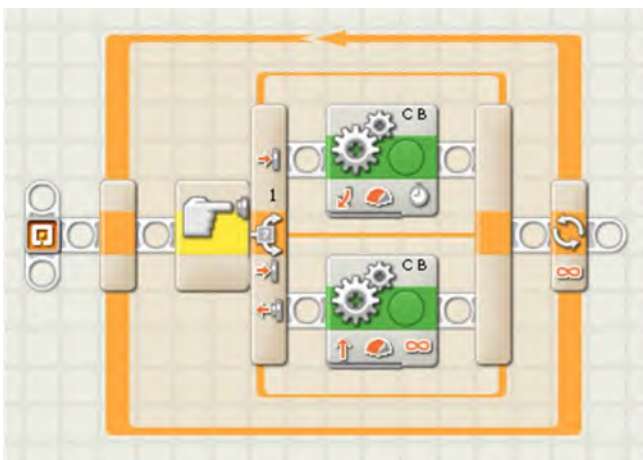
Το όχημα - ρομπότ μας κινείται μπροστά, σε ευθύγραμμη πορεία. Με αυτό τον τρόπο κάποια στιγμή θα συναντήσει κάποιο εμπόδιο και θα παραμείνει κολλημένο στο σημείο εκείνο.

α) Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει αισθητήρα αφής στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι έχει συναντήσει ένα εμπόδιο στην πορεία του.

Για να είναι σε θέση το ρομπότ μας να προσπεράσει το εμπόδιο αυτό και να συνεχίσει την πορεία του μπορούμε να του πούμε να πάει πίσω για λίγο, εκτελώντας στροφή προς μία κατεύθυνση πχ δεξιά. Στη συνέχεια να συνεχίσει ευθεία και να επαναλάβει την ίδια διαδικασία για κάθε εμπόδιο.



β. Το ίδιο πρόβλημα με διακλάδωση αφής :



Δραστηριότητα 2η : αποφυγή εμποδίου με 2 αισθητήρες αφής

Αυτή τη φορά στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκονται τοποθετημένοι δύο αισθητήρες αφής. Οι αισθητήρες αφής είναι τοποθετημένοι ο ένας στην αριστερή και ο άλλος στη δεξιά μεριά του προφυλακτήρα.

Τον αριστερό αισθητήρα αφής τον συνδέουμε στη θύρα εισόδου 1 ενώ το δεξιό στη θύρα εισόδου 3 του NXT.

Τοποθετώντας δύο αισθητήρες αφής το όχημα είναι σε θέση να παρακάμπτει, με πιο αποτελεσματικό τρόπο, τα εμπόδια που θα συναντήσει στην πορεία του.

Ανάλυση προβλήματος :

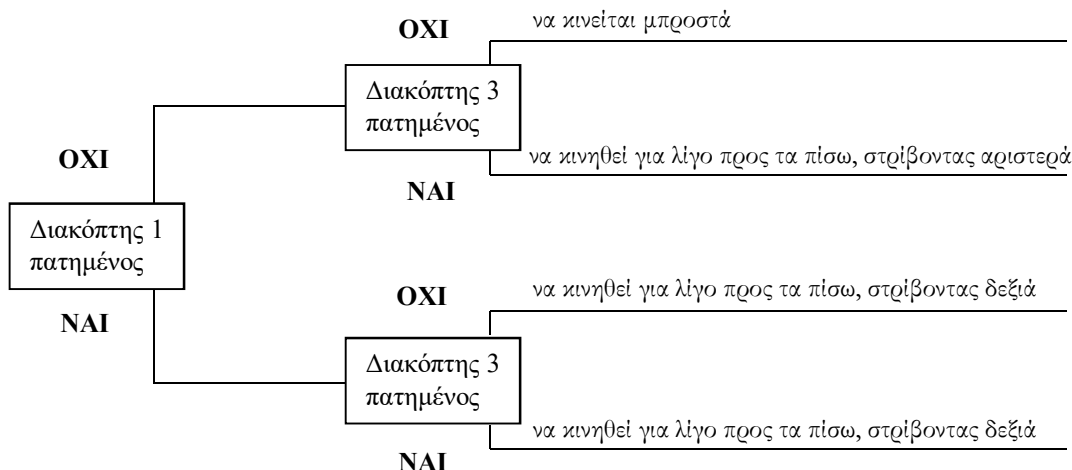
Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει δύο αισθητήρες αφής, δεξιά και αριστερά, στο μπροστινό του μέρος "είναι σε θέση να καταλάβει", όχι μόνο ότι έχει συναντήσει ένα εμπόδιο στην πορεία του, αλλά και σε ποιά πλευρά του.

Μπορούμε τώρα να του πούμε να πάει πίσω για λίγο και να στρίψει ελαφρώς ανάλογα με τον αισθητήρα που πατήθηκε. Στη συνέχεια να συνεχίσει ευθεία και να επαναλάβει την ίδια διαδικασία για κάθε εμπόδιο.

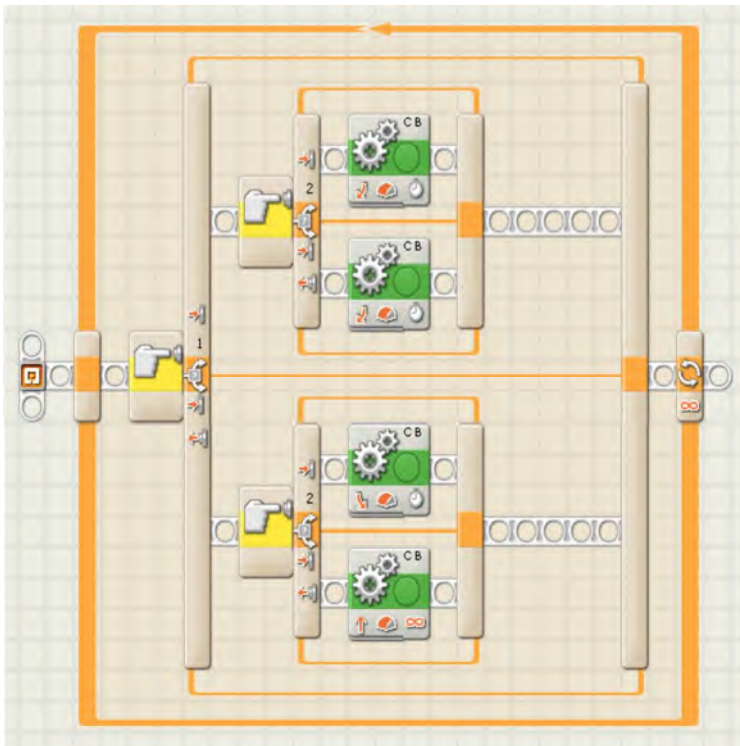
Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

Αριστερός Διακόπτης 1 πατημένος	Δεξιός Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	να κινείται μπροστά
OXI	NAI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας αριστερά
NAI	OXI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας δεξιά
NAI	NAI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας πχ δεξιά

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.

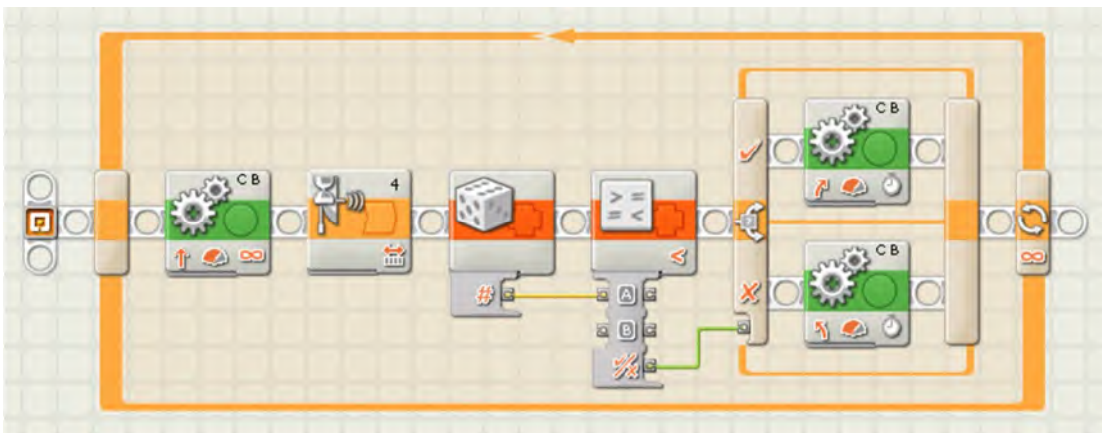


Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



Δραστηριότητα 3η : αποφυγή εμποδίου με αισθητήρα υπερήχων

Στην περίπτωση που αντί για τον αισθητήρα αφής χρησιμοποιήσουμε έναν αισθητήρα υπερήχων τα πράγματα γίνονται πιο απλά, αφού τώρα το όχημα μας δεν χρειάζεται πλέον να πέσει πάνω στο εμπόδιο για να το αντιληφθεί. Μπορεί να εντοπίσει ένα εμπόδιο το οποίο βρίσκεται στην πορεία της κίνησης του, από αρκετά μεγάλη απόσταση και να το παρακάμψει εκτελώντας μία μικρή στροφή και συνεχίζοντας πάλι μπροστά. Μπορούμε μάλιστα να προγραμματίσουμε το όχημα μας, να στρίβει δεξιά ή αριστερά με τυχαίο τρόπο και για τυχαίο χρονικό διάστημα.

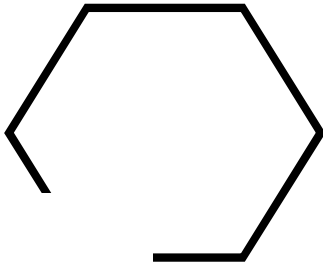


Τον κώδικα του NXT-G v2 των δραστηριοτήτων για την αποφυγή εμποδίων θα τον βρείτε στο σύνδεσμο:

[Αποφυγή εμποδίων](#)

A2. Έξοδος από σπηλιά

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να βρει την έξοδο από μία σπηλιά και να βγει από αυτήν.



Τα παραπάνω προγράμματα με κατάλληλες τροποποιήσεις στις παραμέτρους τους, μπορούμε να τα χρησιμοποιήσουμε και στην περίπτωση που θέλουμε το όχημα-ρομπότ μας να βγει από έναν κλειστό χώρο. Το χώρο αυτό μπορούμε να τον κατασκευάσουμε, φράζοντας τον με τοίχιο ύψους τουλάχιστον 5 εκατοστά. Στο χώρο αυτόν αφήνουμε μία έξοδο τουλάχιστον 30 εκατοστά.

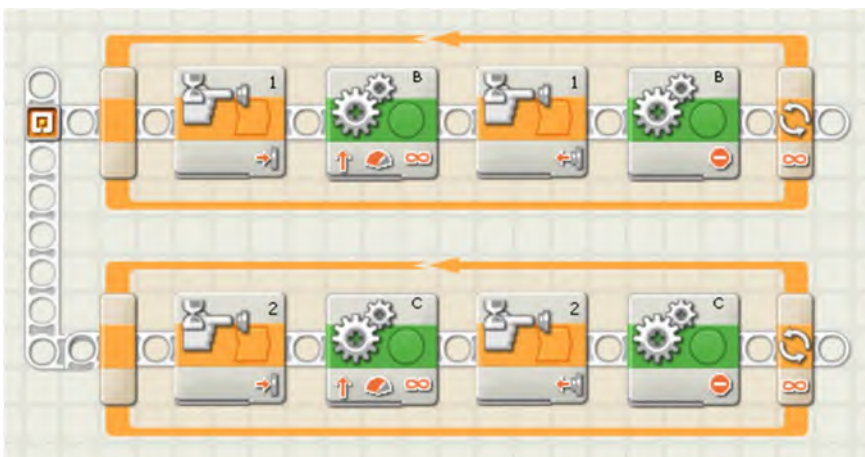
B. Έλεγχος κίνησης με αισθητήρες αφής

Θέλουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να μπορούμε να το οδηγήσουμε με χειριστήριο να ακολουθεί μία διαδρομή που επιθυμούμε.

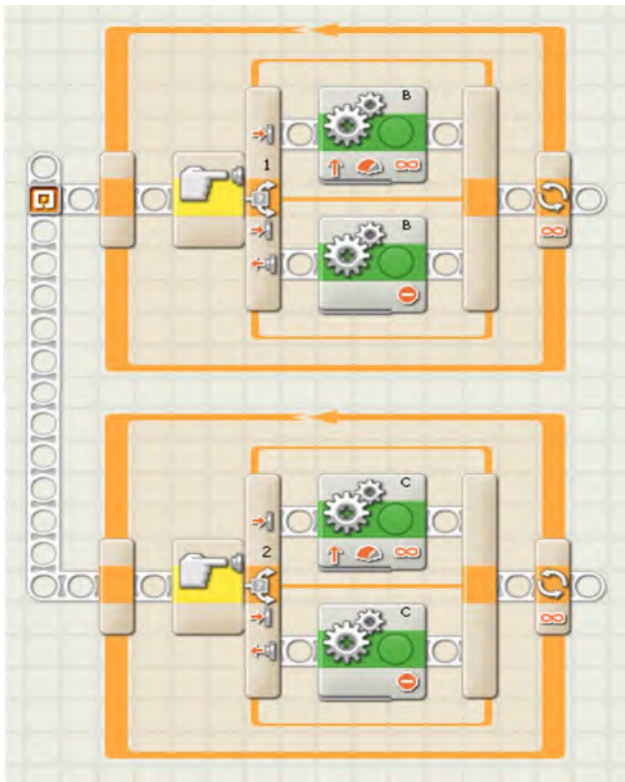
Δραστηριότητα 4η : έλεγχος κίνησης με 2 αισθητήρες αφής με παράλληλες διεργασίες

Συνδέουμε στο όχημα 2 αισθητήρες αφής με μακρύ καλώδιο. Τους αισθητήρες αφής τους συνδέουμε στις θύρες εισόδου 1 και 3 του NXT με τη βοήθεια καλωδίων με μεγάλο μήκος ώστε να υπάρχει μία σχετική ελευθερία στην κίνηση.

α) Το όχημα μας μπορεί να κινηθεί μόνο μπροστά ευθεία ή εκτελώντας στροφή. Στο παρακάτω πρόγραμμα κάθε αισθητήρας αφής ελέγχει έναν κινητήρα. Χρησιμοποιούμε δύο παράλληλες διεργασίες. Στην πρώτη ο αισθητήρας αφής στη θύρα 1 ελέγχει τον κινητήρα B και στη δεύτερη ο αισθητήρας αφής στη θύρα 3 ελέγχει τον κινητήρα C.

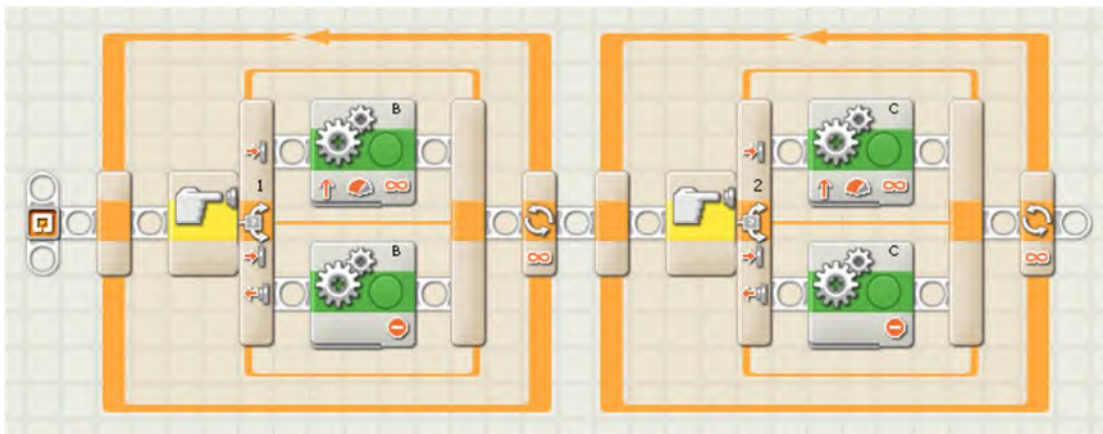


β. Το ίδιο πρόβλημα με διακλαδώσεις αισθητήρα αφής:



Δραστηριότητα 5η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση

Μπορούμε αντί για παράλληλες διεργασίες να χρησιμοποιήσουμε κάλλιστα μόνο μία διεργασία.



Δραστηριότητα 6η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση μέσα σε διακλάδωση

Είναι πιο βολικό για να έχουμε καλύτερο έλεγχο των συνδυασμών που προκύπτουν από δύο αισθητήρες αφής να χρησιμοποιήσουμε μία εντολή διακλάδωσης αισθητήρα αφής μέσα σε μία άλλη εντολή διακλάδωσης αφής.

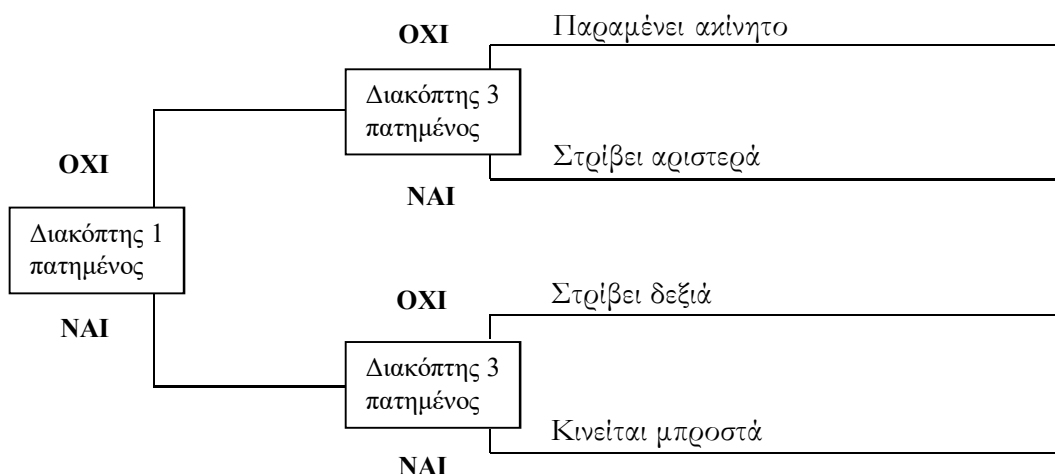
Ανάλυση προβλήματος :

Θέλουμε να μπορούμε να οδηγούμε το ρομπότ μας, χρησιμοποιώντας το χειριστήριο ως τιμόνι και ταχύτητες, έτσι ώστε να μπορούμε να δίνουμε εντολή στο ρομπότ να κινείται μπροστά και να στρίβει δεξιά και αριστερά, ως τηλεκατευθυνόμενο.

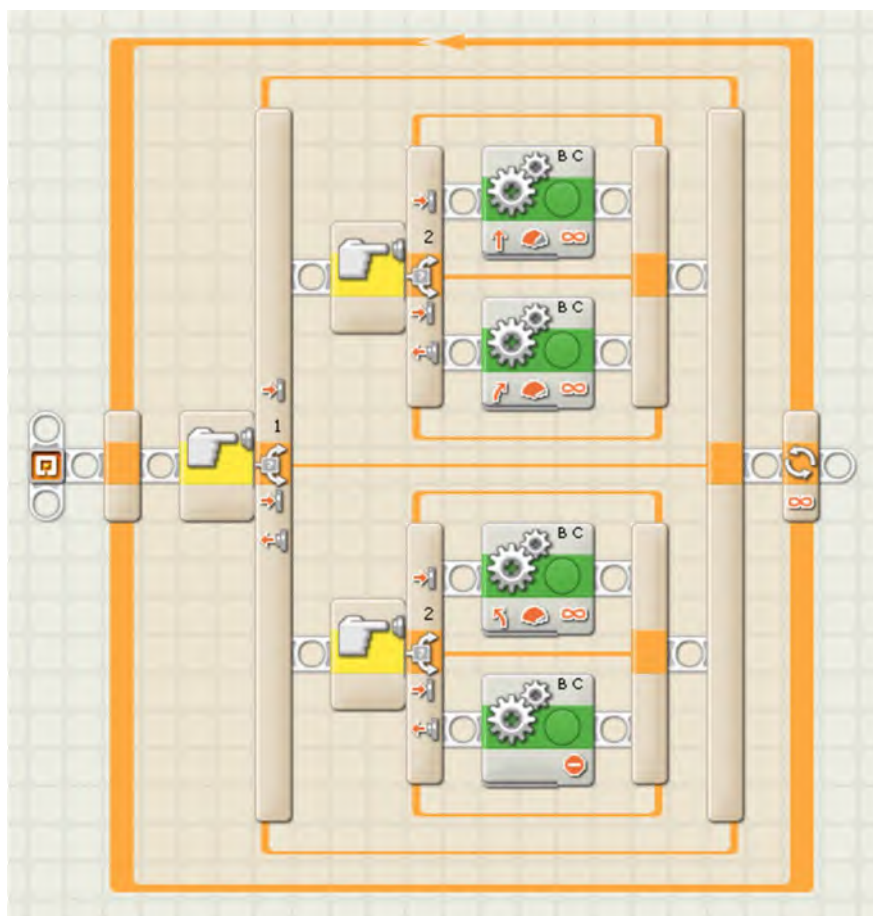
Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

Διακόπτης 1 πατημένος	Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	Παραμένει ακίνητο
OXI	NAI	Στρίβει αριστερά
NAI	OXI	Στρίβει δεξιά
NAI	NAI	Κινείται μπροστά

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.



Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



Δραστηριότητα 7η : έλεγχος κίνησης με 3 αισθητήρες αφής

Αυτή τη φορά συνδέουμε στο όχημα 3 αισθητήρες αφής με μακρύ καλώδιο. Τους αισθητήρες αφής τους συνδέουμε στις θύρες εισόδου 1, 2 και 3 του NXT με τη βοήθεια καλωδίων με μεγάλο μήκος ώστε να υπάρχει μία σχετική ελευθερία στην κίνηση.

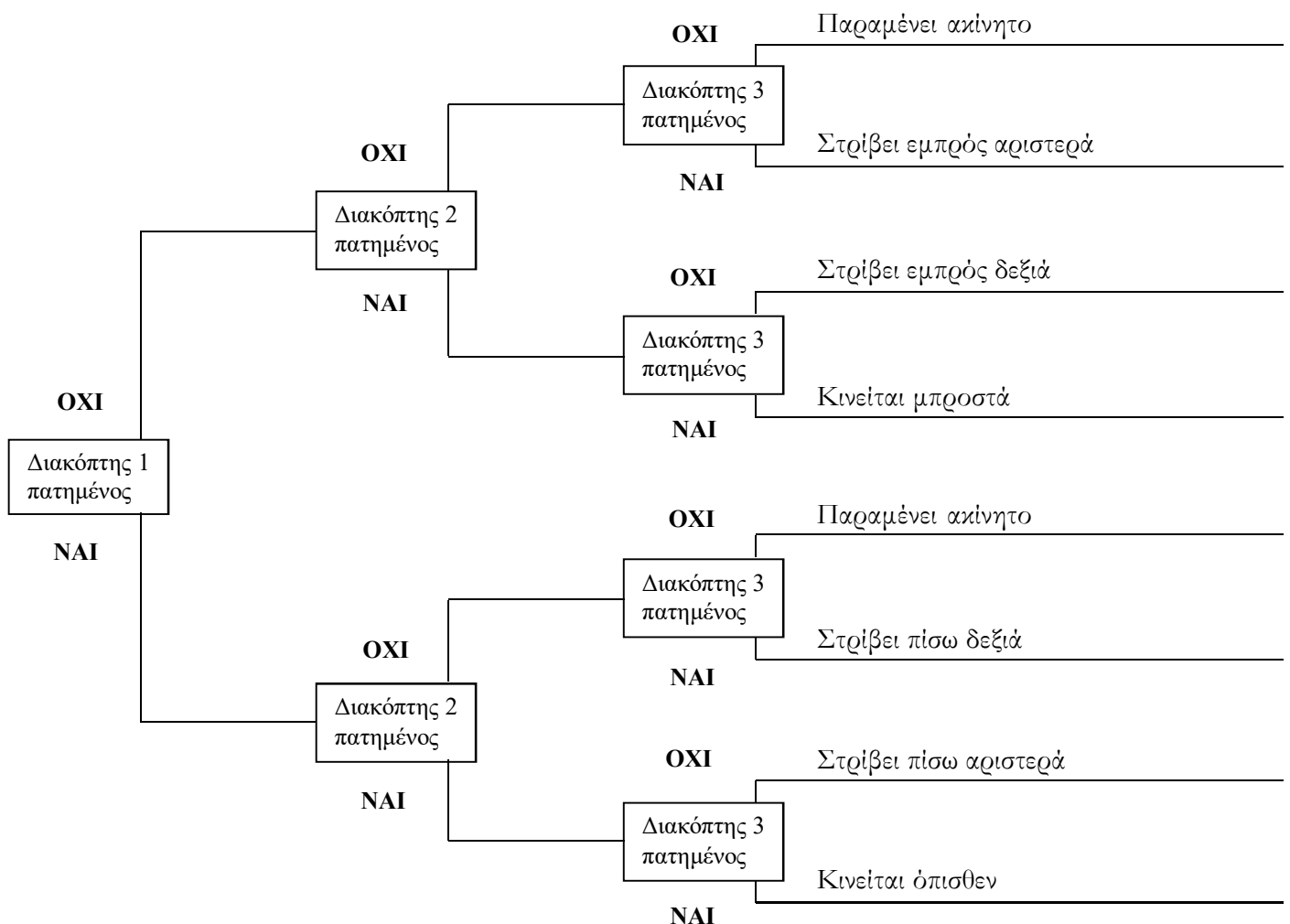
Το πλεονέκτημα χρησιμοποιώντας 3 αισθητήρες αφής για να ελέγξουμε ένα όχημα - ρομπότ είναι ότι τώρα μπορούμε να ελέγξουμε 8 συνδυασμούς συμπεριφοράς του οχήματος μας σε σύγκριση με τους 4 συνδυασμούς που προκύπτουν αν χρησιμοποιήσουμε 2 αισθητήρες αφής. Τώρα έχουμε τη δυνατότητα να κατευθύνουμε και προς τα πίσω. Έτσι μπορούν να προκύψουν αρκετά εναλλακτικά σενάρια.

Σενάρια Ρομποτικής με το Mindstorms NXT

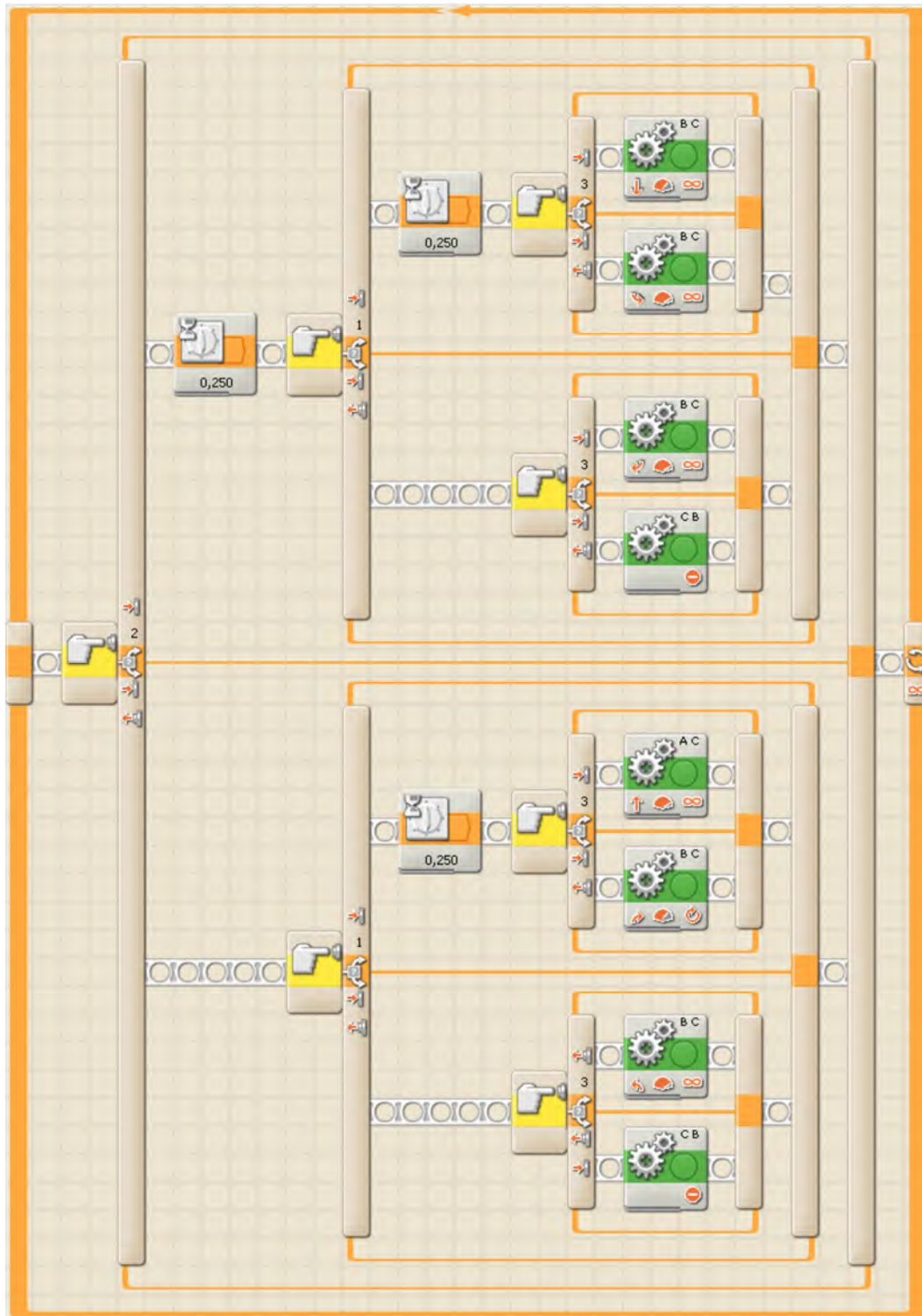
Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

Διακόπτης 1 πατημένος	Διακόπτης 2 πατημένος	Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	OXI	Παραμένει ακίνητο
OXI	OXI	NAI	Στρίβει εμπρός αριστερά
OXI	NAI	OXI	Στρίβει εμπρός δεξιά
OXI	NAI	NAI	Κινείται μπροστά
NAI	OXI	OXI	Παραμένει ακίνητο
NAI	OXI	NAI	Στρίβει πίσω δεξιά
NAI	NAI	OXI	Στρίβει πίσω αριστερά
NAI	NAI	NAI	Κινείται όπισθεν

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.



Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



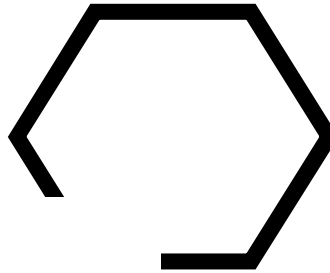
Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τον έλεγχο της κίνησης θα τον βρείτε στο σύνδεσμο: [Έλεγχος κίνησης](#)

Γ. Έξοδος από κυκλικό σχήμα

Θα επιδιώξουμε να προγραμματίσουμε το ρομπότ μας, ώστε να είναι σε θέση να βρει την έξοδο από ένα κλειστό χώρο, σχεδιασμένο στο πάτωμα με μια μαύρη γραμμή και να βγει από αυτό.

Δραστηριότητα 8η : έξοδος από κυκλικό σχήμα

Σχεδιάζουμε ένα κλειστό σχήμα με μαύρο χρώμα σε μία λευκή επιφάνεια. Στο σχήμα αυτό αφήνουμε μία έξοδο τουλάχιστον 30 εκατοστά.



Στο μπροστινό μέρος από το όχημα ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας φωτός. Τον αισθητήρα φωτός αυτόν, τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Προετοιμασία

Ένας αισθητήρας φωτός, δε μπορεί να αναγνωρίσει διαφορετικά χρώματα ή αποχρώσεις του μαύρου. Αυτό που μας δίνει είναι το ποσοστό (0-100%) του φωτός που ανακλάται από μία επιφάνεια.

Όταν έχουμε να κάνουμε με έναν αισθητήρα φωτός πριν προχωρήσουμε στη δημιουργία ενός προγράμματος, πρέπει να δούμε πως αντιλαμβάνεται ο αισθητήρας φωτός τις διαφορετικές επιφάνειες που θα συναντήσει στο περιβάλλον που θα δραστηριοποιηθεί. Το γεγονός αυτό επηρεάζεται από τη χρωματική απόχρωση που έχει μία επιφάνεια αλλά και από τις συνθήκες φωτισμού που επικρατούν στο περιβάλλοντα χώρο. Για να βρούμε τις τιμές που θα αναγνωρίσει ο αισθητήρα φωτός τον τοποθετούμε πάνω από τις διαφορετικές επιφάνειες που μας ενδιαφέρουν και εκτελούμε μία σειρά μετρήσεων.

Στην περίπτωση μας έχουμε μία μαύρη γραμμή την οποία θέλουμε να διακρίνουμε από την υπόλοιπη λευκή επιφάνεια εκτελούμε μία μέτρηση της έντασης του φωτός που αντανάκλαται για τη μαύρη γραμμή και μία για τη λευκή επιφάνεια. Στη συνέχεια υπολογίζουμε τη μέση τιμή των δύο αυτών μετρήσεων την οποία και χρησιμοποιούμε ως τιμή κατωφλίου για να διακρίνει το ρομπότ μας τις δύο διαφορετικές επιφάνειες.

Για παράδειγμα αν το μαύρο διαβαστεί ως 30% και το λευκό ως 60%

$$30 + 60 = 90$$

$$90/2 = 45$$

τότε το 45, θα είναι η τιμή που θα χρησιμοποιήσουμε στη συνθήκη για την εντολή με αισθητήρα φωτός.

Οδηγίες

Στα προγράμματα στα οποία ένα όχημα-ρομπότ πρέπει να ξεχωρίσει δύο επιφάνειες διαφορετικής απόχρωσης, χρησιμοποιούμε τον αισθητήρα φωτός με ενεργοποιημένη την εκπομπή της υπέρυθρης ακτινοβολίας. Επίσης για να μπορέσει ένα όχημα ρομπότ να αναγνωρίσει σωστά τις επιφάνειες θα πρέπει ο αισθητήρας φωτός να βρίσκεται σε μικρή απόσταση από την επιφάνεια του δαπέδου και το όχημα να κινείται με μικρή ταχύτητα.

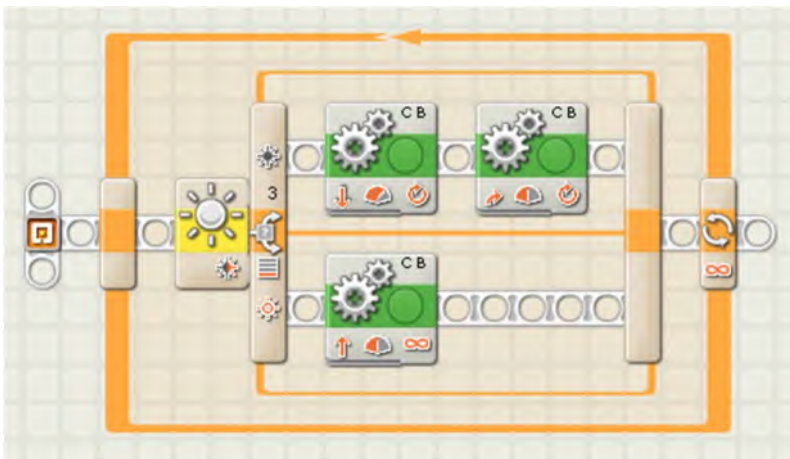
Ανάλυση προβλήματος :

Το όχημα - ρομπότ μας κινείται μπροστά σε ευθύγραμμη πορεία. Με αυτό τον τρόπο κάποια στιγμή θα συναντήσει τη μαύρη γραμμή.

α) Λαμβάνοντας υπόψη, ότι το ρομπότ μας διαθέτει αισθητήρα φωτός στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι έχει συναντήσει μία μαύρη γραμμή στην πορεία του. Μόλις συναντήσει τη μαύρη γραμμή θα κινηθεί λίγο προς τα πίσω, στρίβοντας ελαφρώς. Στη συνέχεια θα επαναλάβει τα βήματα αυτά μέχρι να βρει την έξοδο.



β. Το ίδιο πρόβλημα με διακλάδωση φωτός :

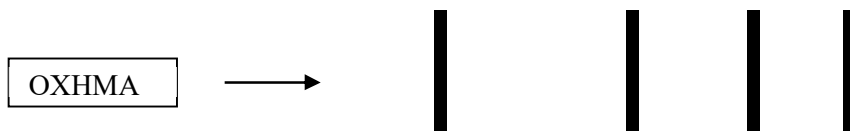


Τον κώδικα του NXT-G v2 των δραστηριοτήτων για την έξοδο από σπηλιά θα τον βρείτε στο σύνδεσμο: [Έξοδος από σπηλιά](#)

Δ. Μέτρηση μαύρων γραμμών

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να μετρήσει τις μαύρες γραμμές που θα συναντήσει στην πορεία του.

Σχεδιάζουμε μία σειρά από μαύρες γραμμές με μαύρο χρώμα σε μία λευκή επιφάνεια. Οι γραμμές αυτές βρίσκονται μπροστά από την πορεία του οχήματος μας και είναι παράλληλες μεταξύ τους.



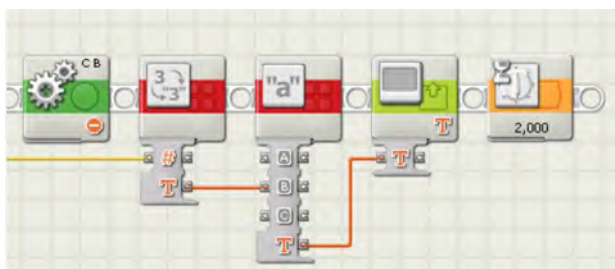
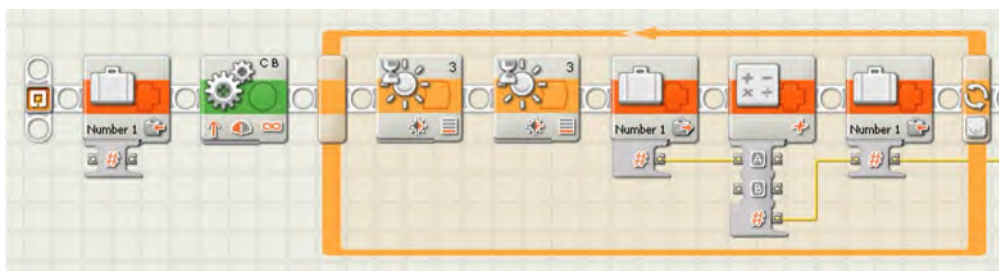
Δραστηριότητα 9η : μέτρημα μαύρων γραμμών

Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας φωτός. Τον αισθητήρα φωτός αυτόν, τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Ανάλυση προβλήματος :

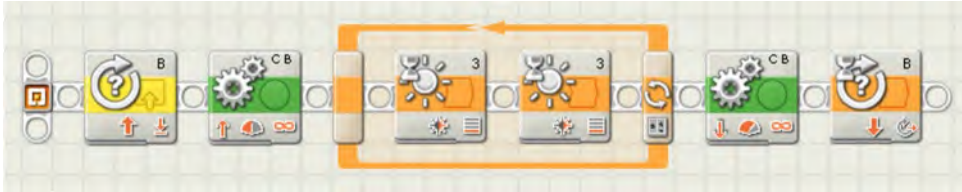
Το όχημα - ρομπότ μας κινείται μπροστά σε ευθύγραμμη πορεία για κάποιο χρονικό διάστημα. Στην πορεία θα συναντήσει έναν αριθμό από μαύρες γραμμές. Για να μπορέσει να μετρήσει τον αριθμό των γραμμών αυτών χρησιμοποιούμε έναν μετρητή.

Μόλις συναντήσει μία μαύρη γραμμή αυξάνει το μετρητή αυτόν κατά μία μονάδα. Η δεύτερη εντολή του αισθητήρα φωτός για αναμονή για φως είναι απαραίτητη για είμαστε σίγουροι ότι το όχημα έχει διασχίσει μία μαύρη γραμμή ώστε να μην ξαναμετρήσει κάποια χοντρή γραμμή πάνω από μία φορά.



Δραστηριότητα 10η : μπροστά πάνω από 3 μαύρες γραμμές και επιστροφή

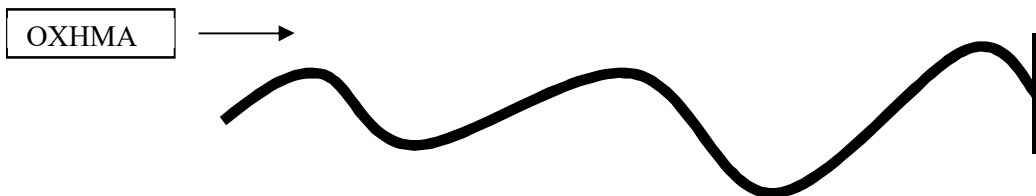
Στο παρακάτω πρόγραμμα το όχημα ρομπότ το οποίο είναι εφοδιασμένο με βραχίονα με έναν αισθητήρα φωτός, προχωράει προς τα μπροστά μέχρι να διασχίσει 3 μαύρες γραμμές και στη συνέχεια επιστρέφει ξανά πίσω στην αρχική του θέση.



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για τη μέτρηση γραμμών θα τον βρείτε στο σύνδεσμο : [Μέτρηση γραμμών](#)

Ε. Ακολουθώντας τη γραμμή (Line follower)

Θα επιδιώξουμε να προγραμματίσουμε το ρομπότ μας, ώστε να είναι σε θέση να προχωράει κατά μήκος μίας μαύρης καμπύλης γραμμής, σχεδιασμένης στο πάτωμα.



Πως μπορούμε να κατορθώσουμε ένα όχημα-ρομπότ με δύο κινητήρες να ακολουθεί μία μαύρη γραμμή χρησιμοποιώντας έναν αισθητήρα φωτός.

Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας φωτός. Τον αισθητήρα φωτός αυτόν, τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Δραστηριότητα 11η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι

Ανάλυση προβλήματος :

Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει αισθητήρα φωτός στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι στην πορεία του έχει συναντήσει τη μία μαύρη γραμμή. Θεωρήστε ότι υπάρχει μία αόρατη σε εσάς γραμμή χαραγμένη στο πάτωμα, την οποία αντιλαμβάνεστε μόνο τη στιγμή που θα τη συναντήσετε. (πχ μία γραμμή από αόρατη μελάνη την οποία δεν βλέπετε, αλλά μόλις πατήσετε πάνω της ειδοποιείστε από ένα σφύριγμα.) Αρχικά είστε τοποθετημένοι δίπλα στη γραμμή αυτή. Σας ενημερώνουν επίσης, ότι η γραμμή βρίσκεται στα **δεξιά** σας.

Τοποθετούμε το όχημα στην **αριστερή** πλευρά της γραμμής.

Ξεκινάμε κινώντας μόνο τον έναν κινητήρα ώστε να στρίψει προς την κατεύθυνση που βρίσκεται η γραμμή.

Μόλις τη συναντήσει (ο αισθητήρας φωτός ανιχνεύσει σκοτάδι) του δίνουμε εντολή να απομακρυνθεί από τη γραμμή κινώντας μόνο τον άλλο κινητήρα και μετά να ξαναεπιστρέψει προς αυτήν.



Με το πρόγραμμα αυτό προκύπτει μία σειρά προβλημάτων:

Αν ο χρόνος όπου το ρομπότ απομακρύνεται από τη μαύρη γραμμή είναι αρκετά μεγάλος υπάρχει το ενδεχόμενο στην περίπτωση που αυτή στρίβει απότομα μακριά από το ρομπότ μας, να απομακρυνθεί το ρομπότ μας μακριά από αυτήν και τη χάσει εντελώς. Όταν μειώνουμε το χρόνο, στον οποίο το ρομπότ μας απομακρύνεται από τη μαύρη γραμμή, τότε το ρομπότ μας κινείται πιο ομαλά ακολουθώντας τη γραμμή. Από την άλλη όμως αν μειώσουμε πάρα πολύ το χρόνο που κινείται το ρομπότ όταν απομακρύνεται από τη μαύρη γραμμή στην περίπτωση που αυτή στρίβει απότομα προς το ρομπότ, το ρομπότ θα συνεχίσει να βρίσκεται μέσα στη γραμμή και μετά την απομάκρυνση του από αυτήν με συνέπεια να αργοπορήσει ή και να περάσει από την άλλη μεριά με αποτέλεσμα να τη χάσει εντελώς.

Δραστηριότητα 12η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως

Μπορούμε να βελτιώσουμε το προηγούμενο πρόβλημα αν αντικαταστήσουμε την εντολή αναμονής για χρόνο με μία εντολή αναμονής για φως ώστε να απομακρύνεται από τη μαύρη γραμμή μέχρι να βρεθεί στη λευκή επιφάνεια.

Στην περίπτωση αυτήν το ρομπότ μας θα κινηθεί περισσότερο ομαλά.



Δραστηριότητα 13η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με διακλάδωση

Το παραπάνω πρόγραμμα μπορεί να απλοποιηθεί χρησιμοποιώντας την εντολή διακλάδωσης αισθητήρα φωτός.

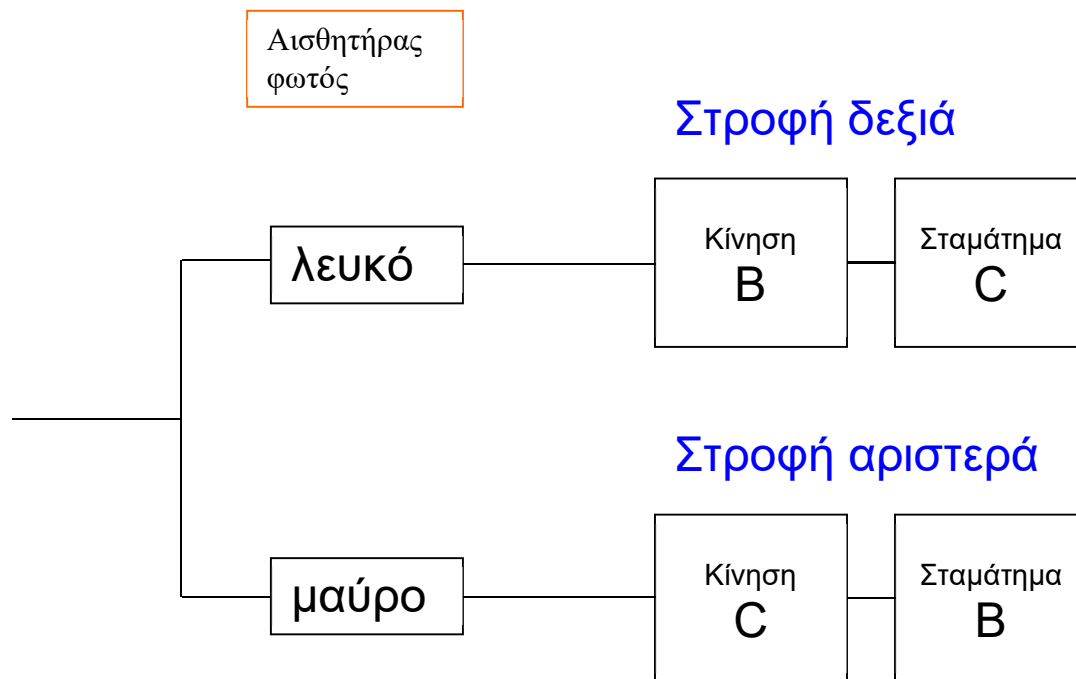
Ανάλυση προβλήματος :

Κάνουμε τη σύμβαση το όχημα να βρίσκεται **αριστερά** από τη γραμμή.

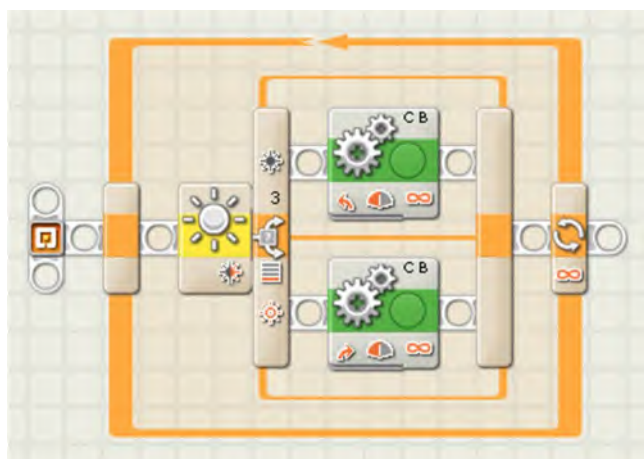
Αυτό που θέλουμε είναι να δώσουμε εντολή στο όχημα ρομπότ μας, να στρίβει δεξιά (κίνηση κινητήρα B, σταμάτημα κινητήρα C) όταν ο αισθητήρας φωτός ανιχνεύει φωτεινό χρώμα και να στρίβει αριστερά (κίνηση κινητήρα C, σταμάτημα κινητήρα B) όταν ο αισθητήρας φωτός ανιχνεύει σκοτεινό χρώμα.

Το ρομπότ διαβάζει την τιμή από τον αισθητήρα φωτός στη θύρα 1 και τη συγκρίνει με την αριθμητική τιμή 45 που έχουμε ορίσει ως κατώφλι. Αν η τιμή είναι πάνω από το κατώφλι, τότε ο αισθητήρας φωτός βρίσκεται έξω από τη μαύρη γραμμή. Το πρόγραμμα κινεί τον αριστερό κινητήρα και σταματάει το δεξιό κινητήρα, στρίβοντας το ρομπότ μας δεξιά προς το αριστερό σύνορο της μαύρης γραμμής. Ομοίως αν η τιμή είναι κάτω από το κατώφλι το ρομπότ στρίβει αριστερά και απομακρύνεται από το αριστερό σύνορο της μαύρης γραμμής. Ο κύκλος αυτός επαναλαμβάνεται συνεχώς μέχρι να σταματήσουμε το πρόγραμμα.

Το παρακάτω λογικό διάγραμμα περιγράφει την επιθυμητή διαδικασία:

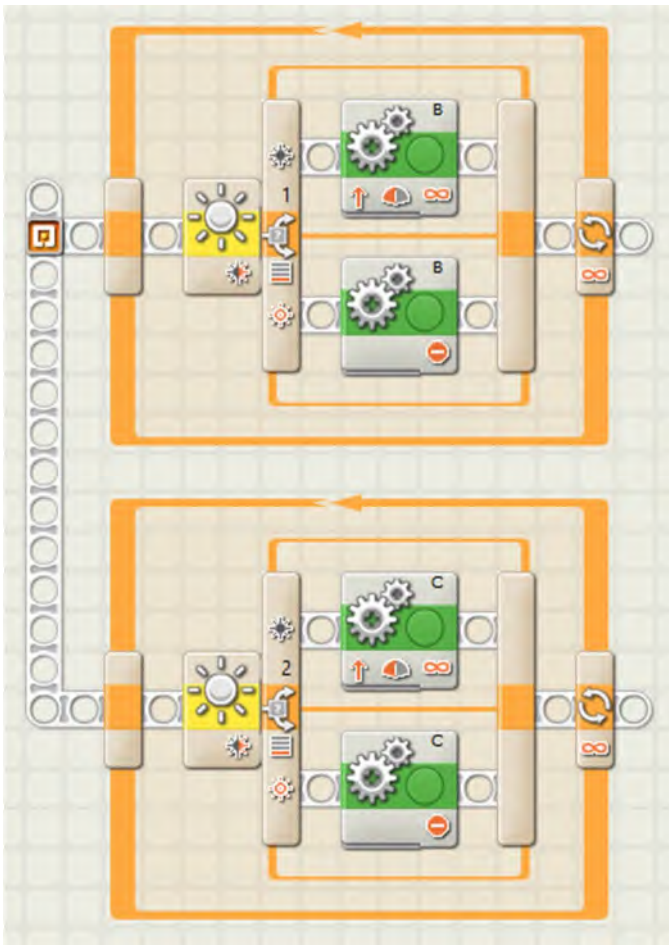


Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



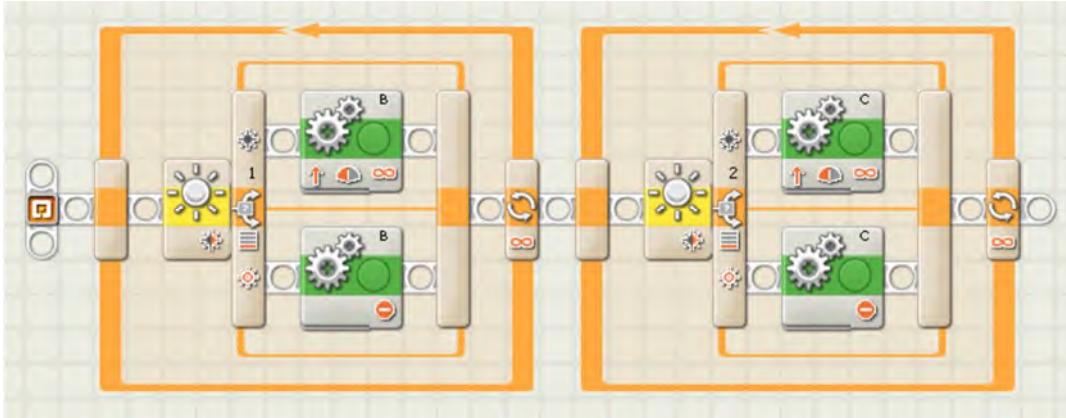
Δραστηριότητα 14η: Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με επανάληψη και παράλληλες διεργασίες

Το ίδιο πρόβλημα με εντολή επανάληψης αισθητήρα φωτός :



Δραστηριότητα 15η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση και μία διεργασία

Μπορούμε αντί για παράλληλες διεργασίες να χρησιμοποιήσουμε κάλλιστα μόνο μία διεργασία.



Δραστηριότητα 16η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση μέσα σε διακλάδωση και μία διεργασία

Είναι πιο βολικό για να έχουμε καλύτερο έλεγχο των συνδυασμών που προκύπτουν από δύο αισθητήρες φωτός να χρησιμοποιήσουμε μία εντολή διακλάδωσης αισθητήρα φωτός μέσα σε μία άλλη εντολή διακλάδωση φωτός.

Ανάλυση προβλήματος :

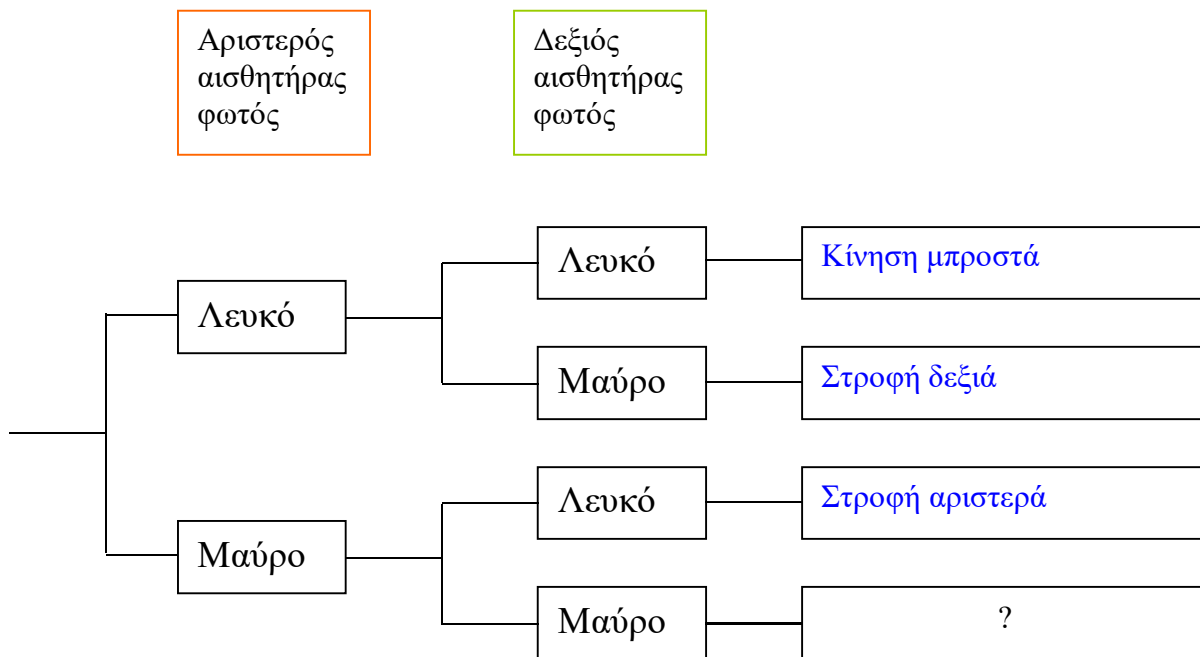
Τοποθετούμε το όχημα - ρομπότ πάνω στη μαύρη γραμμή, ώστε οι αισθητήρες φωτός να βρίσκονται δεξιά και αριστερά της μαύρης γραμμής.

Αυτό που θέλουμε είναι να δώσουμε εντολή στο όχημα - ρομπότ μας κινείται ευθεία όταν και οι δύο αισθητήρες ανιχνεύουν φωτεινό χρώμα, να στρίβει αριστερά (κίνηση κινητήρα C, σταμάτημα κινητήρα A) όταν ο αισθητήρας φωτός 1 ανιχνεύει σκοτεινό χρώμα και να στρίβει δεξιά (κίνηση κινητήρα A, σταμάτημα κινητήρα C) όταν ο αισθητήρας φωτός 2 ανιχνεύει σκοτεινό χρώμα.

Μπορούμε επίσης να το κάνουμε να σταματάει όταν και οι δύο αισθητήρες ανιχνεύουν σκοτεινό χρώμα.

Σενάρια Ρομποτικής με το Mindstorms NXT

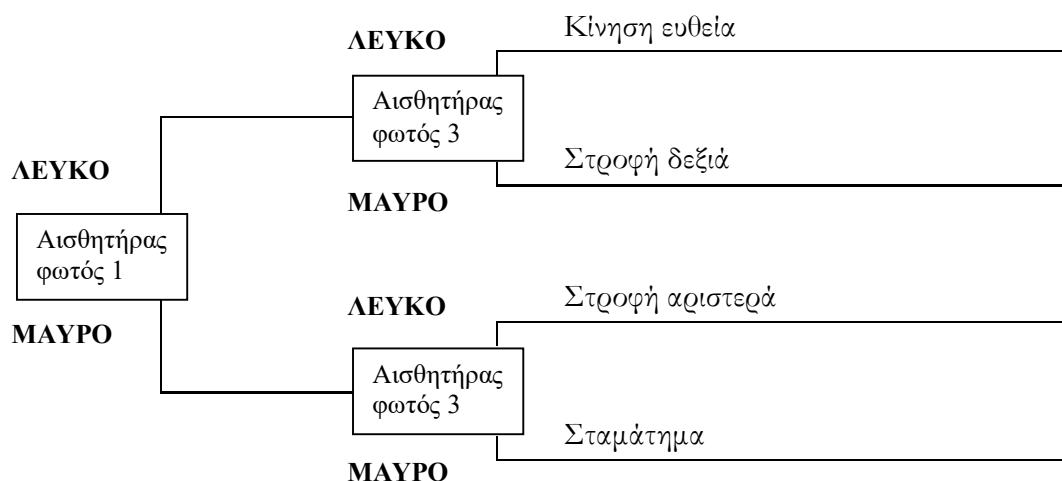
Το παρακάτω λογικό διάγραμμα περιγράφει την επιθυμητή διαδικασία:



Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

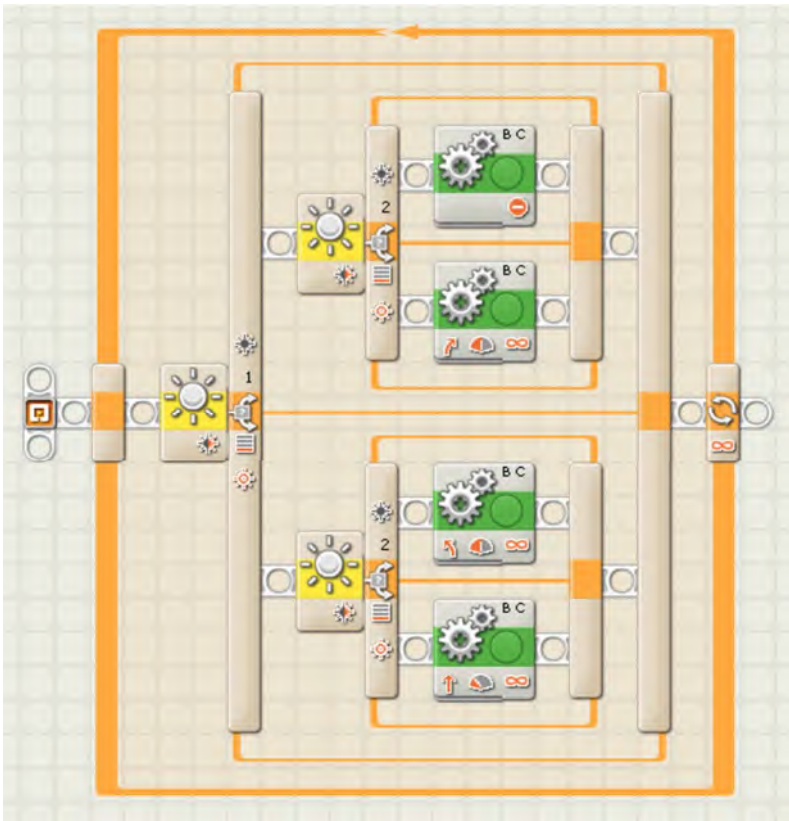
Αισθητήρας φωτός 1	Αισθητήρας φωτός 3	Συμπεριφορά κίνησης
ΛΕΥΚΟ	ΛΕΥΚΟ	Κίνηση ευθεία
ΛΕΥΚΟ	ΜΑΥΡΟ	Στροφή δεξιά
ΜΑΥΡΟ	ΛΕΥΚΟ	Στροφή αριστερά
ΜΑΥΡΟ	ΜΑΥΡΟ	Σταμάτημα

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.



Σενάρια Ρομποτικής με το Mindstorms NXT

Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



Τον κώδικα του NXT-G v2 των δραστηριοτήτων για να ακολουθήσει το ρομπότ μας μία γραμμή θα τον βρείτε στο σύνδεσμο : [Ακολουθώντας τη γραμμή](#)

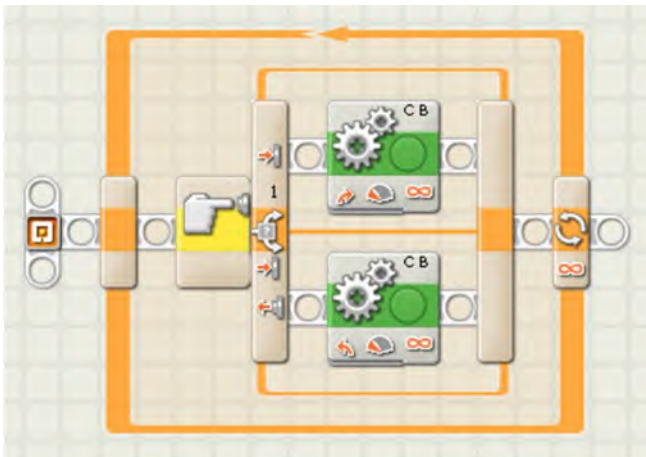
Ζ. Ακολουθώντας τον τοίχο

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να ακολουθεί μία πορεία παράλληλη με έναν τοίχο.

Δραστηριότητα 17η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα αφής

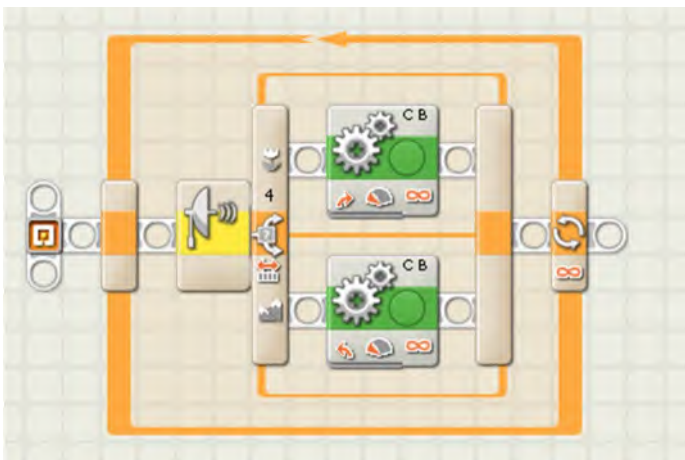
Στο μπροστινό μέρος από το όχημα ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας αφής.

Τον αισθητήρα αφής αυτόν τον συνδέουμε στη θύρα εισόδου 1 του NXT.



Δραστηριότητα 18η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα υπερήχων

Στην περίπτωση που αντί για τον αισθητήρα αφής χρησιμοποιήσουμε έναν αισθητήρα υπερήχων τα πράγματα γίνονται πιο απλά, αφού τώρα το όχημα μας δεν χρειάζεται πλέον να πέσει πάνω στον τοίχο για να τον αντιληφθεί. Μπορεί να εντοπίσει τον τοίχο, από αρκετά μεγάλη απόσταση

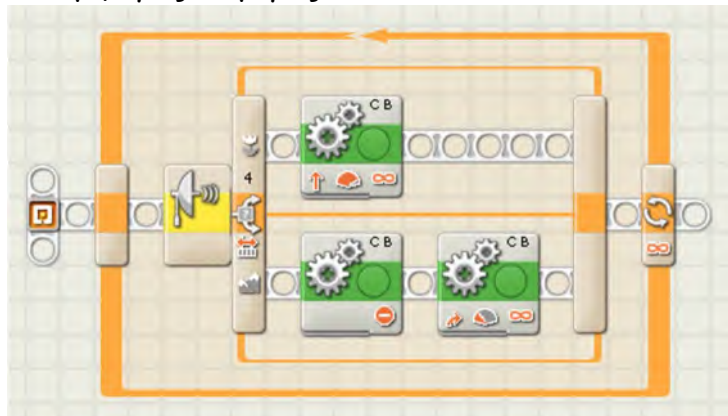


Τον κώδικα του NXT-G v2 των δραστηριοτήτων για να ακολουθήσει το ρομπότ μας έναν τοίχο θα τον βρείτε στο σύνδεσμο : [Ακολουθώντας τον τοίχο](http://kyrgeo.sites.sch.gr)

Η. Άλλα προγράμματα

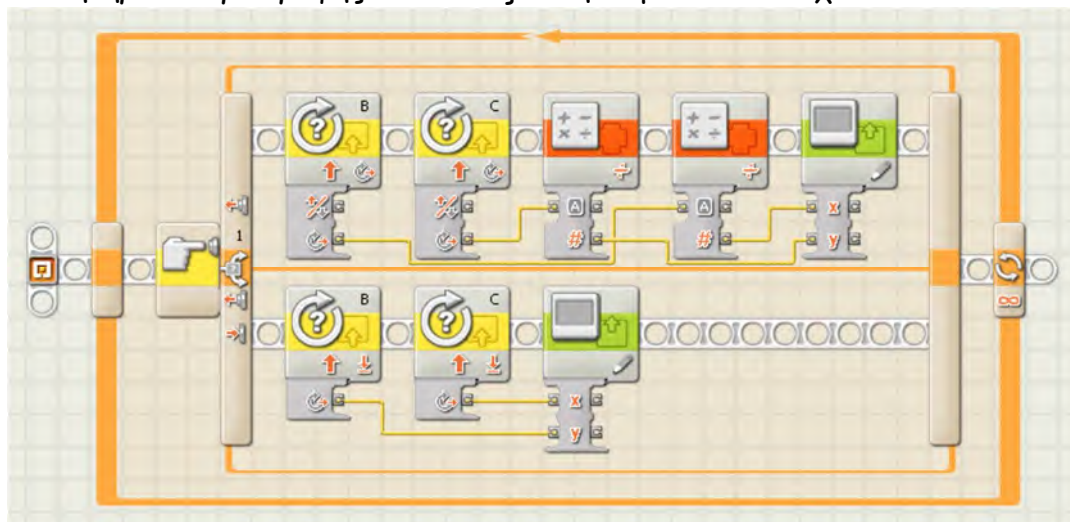
Δραστηριότητα 19η : φύλαξη με αισθητήρα υπερήχων

Στο παρακάτω πρόγραμμα το όχημα-ρομπότ εποπτεύει ένα χώρο κινούμενο με μικρή ταχύτητα εκτελώντας κυκλική πορεία. Αν κάποιος εισβολέας βρεθεί στο οπτικό του πεδίο τρέχει με δύναμη προς το μέρος του.



Δραστηριότητα 20η : Σχεδιαστήριο

Με το παρακάτω πρόγραμμα μπορούμε να σχεδιάζουμε μία εικόνα στην οθόνη του NXT ελέγχοντας τη θέση του μολυβιού μας με τη βοήθεια δύο αισθητήρων περιστροφής. Όταν ο διακόπτης του αισθητήρα αφής βρίσκεται απελευθερωμένος ο χρήστης μπορεί να σχεδιάσει στην οθόνη του NXT χρησιμοποιώντας ως χειριστήρια τους αισθητήρες περιστροφής B και C. Οι τιμές από τους αισθητήρες περιστροφής πολλαπλασιάζονται με τον παράγοντα 20 για να γίνει προσαρμογή τους στην κλίμακα της οθόνης του NXT. Όταν ο διακόπτης του αισθητήρα αφής πατηθεί, η οθόνη του NXT καθαρίζεται και γίνεται αρχικοποίηση των αισθητήρων περιστροφής ώστε να ξεκινήσουμε ένα νέο σχέδιο.

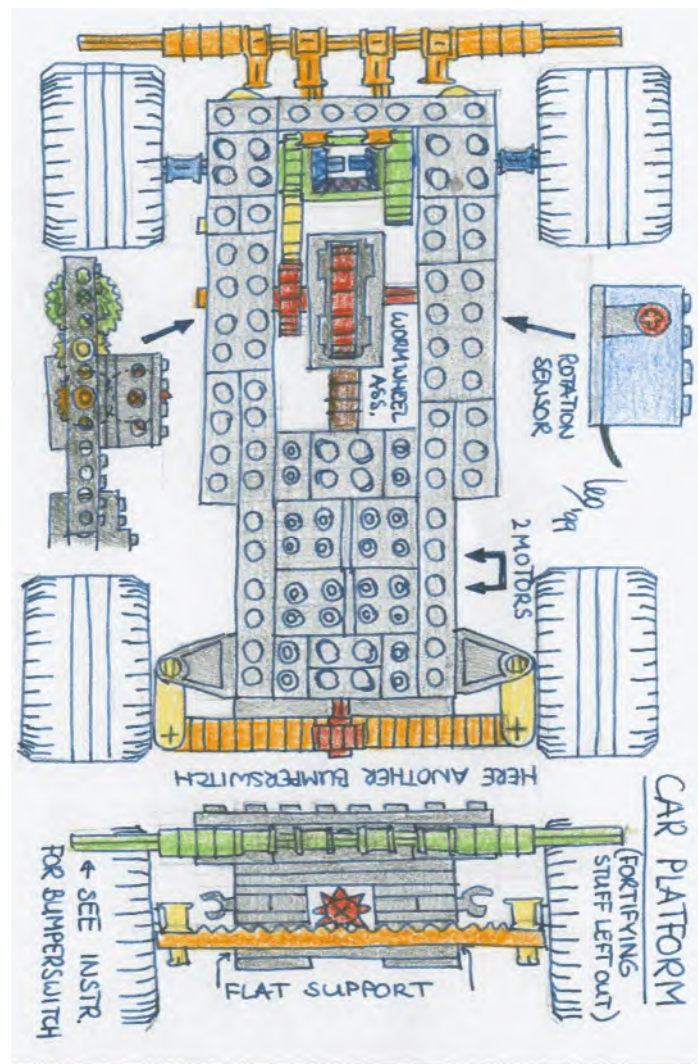


Τον κώδικα του NXT-G v2 αυτών των δραστηριοτήτων θα τον βρείτε στο σύνδεσμο :

[Άλλα προγράμματα](#)

Μια παρουσίαση για τις Εντολές του NXT-G θα βρείτε στο σύνδεσμο :

[Εντολές του NXT-G](#)



Ενότητα 7η

Διδακτικό Σενάριο Ρομποτικής

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Εισαγωγή στον Προγραμματισμό με το πακέτο MINDSTORMS NXT της LEGO

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 12 διδακτικές ώρες (6 δίωρες συναντήσεις).

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Το διδακτικό σενάριο σχετίζεται άμεσα τόσο με το Διαθεματικό Ενιαίο Πλαίσιο Προγράμματος Σπουδών (Δ.Ε.Π.Π.Σ.) για την Πληροφορική όσο και με τη διδασκαλία -πρόγραμμα σπουδών των νέων διδακτικών αντικειμένων που έχουν εισαχθεί στα ολοήμερα δημοτικά σχολεία που λειτουργήσουν με Ενιαίο Αναμορφωμένο Εκπαιδευτικό Πρόγραμμα (ΕΑΕΠ) (ΦΕΚ 1139/28-8-2010).

Επίσης σύμφωνα με το νέο αναλυτικό πιλοτικό πρόγραμμα και ειδικότερα στο επιστημονικό πεδίο Τεχνολογίας Πληροφορίας & Επικοινωνιών στην Πρωτοβάθμια Εκπαίδευση ορίζεται στους βασικούς άξονες μαθησιακών στόχων του Π.Σ. για τις ΤΠΕ ο προγραμματισμός του υπολογιστή (Ενότητα: Διερευνώ, ανακαλύπτω και λύνω προβλήματα με ΤΠΕ). Ειδικότερα, σύμφωνα με το νέο αναλυτικό πιλοτικό πρόγραμμα βασικός στόχος της υποενότητας «Προγραμματίζω τον υπολογιστή μου (10 ώρες)» είναι η σταδιακή εξοικείωση των μαθητών με τον προγραμματισμό μέσα από την αξιοποίηση διαθέσιμων εκπαιδευτικών περιβαλλόντων οπτικού προγραμματισμού. Οι μαθητές σε κατάλληλα εκπαιδευτικά περιβάλλοντα προγραμματισμού, χειρίζονται και διερευνούν έτοιμα προγράμματα και εισάγονται στην έννοια του αλγορίθμου. Ο εκπαιδευτικός, προκαλεί το ενδιαφέρον των μαθητών, ώστε να δημιουργήσουν μικρές εφαρμογές, στις οποίες θα προκαλείται η δράση αντικειμένων. Όσον αφορά το εκπαιδευτικό υλικό προτείνεται και το εκπαιδευτικό πακέτο MINDSTORMS NXT της LEGO.

Επίσης, σύμφωνα με τα παρόντα ΑΠΣ και ΔΕΠΠΣ Πληροφορικής το παρόν διδακτικό σενάριο μπορεί να αξιοποιηθεί στο μάθημα της Πληροφορικής της Γ' Γυμνασίου.

Ομοίως, στο Πρόγραμμα Σπουδών του Γενικού και του Τεχνολογικού Λυκείου η εισαγωγή σε βασικές έννοιες του προγραμματισμού αποτελούν στόχο των μαθημάτων Πληροφορικής. Το μαθησιακό περιβάλλον της LEGO αποτελεί μια εναλλακτική προσέγγιση γλώσσας προγραμματισμού που βοηθά στην ενεργοποίηση των μαθητών, στην οικοδόμηση νέας γνώσης και στην εφαρμογή της νέας γνώσης στην επίλυση προβλημάτων. Το σενάριο θα μπορούσε να διδαχθεί και σε μαθητές της Β Λυκείου στο πλαίσιο του μαθήματος «Εφαρμογές Υπολογιστών».

Το πακέτο Mindstorms NXT for Education της Lego παρέχει ένα περιβάλλον οπτικού προγραμματισμού με το οποίο μπορούν οι μαθητές να προσδίδουν σε ένα ρομπότ συμπεριφορές που μπορούν να επινοήσουν χωρίς περιορισμούς, εξαντλώντας τη φαντασία του καθενός, εύκολα και γρήγορα, ενώ παράλληλα θα εξοικειώνονται με τις βασικές αρχές του προγραμματισμού. Ο προγραμματισμός μίας ρομποτικής κατασκευής διαφέρει από τον παραδοσιακό προγραμματισμό αφού η ενασχόληση δεν

Σενάρια Ρομποτικής με το Mindstorms NXT

είναι πλέον μία διαδικασία αριθμητικών μόνο υπολογισμών αλλά το ζητούμενο τώρα, είναι η απόδοση στοιχειώδους ευφυΐας σε μία μηχανική κατασκευή.

Το πακέτο αυτό συγκεντρώνει ενδιαφέροντα στοιχεία για την εισαγωγή και εμβάθυνση στον προγραμματισμό συγκριτικά με άλλα περιβάλλοντα. Ο προγραμματισμός μίας ρομποτικής κατασκευής διαφέρει από τον παραδοσιακό προγραμματισμό αφού η ενασχόληση δεν είναι πλέον μία διαδικασία αριθμητικών μόνο υπολογισμών αλλά το ζητούμενο τώρα, είναι η απόδοση στοιχειώδους ευφυΐας σε μία μηχανική κατασκευή. Το λογισμικό **NXT-G** παρέχει ένα απλό γραφικό περιβάλλον για τον προγραμματισμό “συμπεριφορών” των ρομποτικών κατασκευών της Lego. Βασίζεται στη χρήση εικονιδίων. Ο προγραμματισμός ακολουθεί τις αρχές του οπτικού προγραμματισμού, όπου ο χρήστης (μαθητής) επιλέγει από μια συλλογή εικονιδίων που αντιστοιχούν σε συγκεκριμένες δράσεις και τη λογική διασύνδεση μεταξύ τους.

Λόγω των δυνατοτήτων του NXT να ελέγχει κινητήρες ή φώτα και να συγκεντρώνει δεδομένα με τη βοήθεια αισθητήρων, παιδιά και ενήλικες μπορούν εύκολα να δημιουργήσουν κατασκευές που κινούνται, σκέφτονται, και αντιδρούν (Portsmore 1999).

Η φιλοσοφία του εκπαιδευτικού υλικού της εταιρείας Lego, στηρίζεται στο ότι το παιδί πρέπει από μόνο του να οικοδομεί τη γνώση και ειδικότερα στην άποψη ότι η μάθηση επέρχεται μέσα από το παιχνίδι (“*learning through play*”).

Με αυτή την πλατφόρμα προγραμματισμού και αυτόν τον τρόπο διδακτικής προσέγγισης, οι μαθητές έρχονται σε επαφή με σημαντικές μαθηματικές και υπολογιστικές ιδέες, ενώ παράλληλα κατανοούν καλύτερα τη γενική διαδικασία του σχεδιασμού αλγορίθμων. Παράλληλα αναπτύσσεται η αναλυτική σκέψη των μαθητών και καλλιεργείται η δημιουργικότητά τους.

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Σκοπός του σεναρίου είναι οι μαθητές να έρθουν σε επαφή με το περιβάλλον οπτικού προγραμματισμού της γλώσσας NXT-G και να δημιουργήσουν τα πρώτα δικά τους απλά προγράμματα.

Στόχοι του σεναρίου είναι οι μαθητές :

- να κατανοήσουν με την έννοια του ρομπότ.
- να γνωρίσουν το πακέτο ρομποτικής της Lego.
- να εξοικειωθούν με το προγραμματιστικό περιβάλλον Mindstorms NXT-G Edu.
- να μπορούν να χρησιμοποιούν απλές εντολές κίνησης και αναμονής για αισθητήρα.
- να μπορούν να χρησιμοποιούν τις προγραμματιστικές δομές επανάληψης και επιλογής.
- να μπορούν να χρησιμοποιήσουν συνδυαστικά εντολές για να επιλύσουν σύνθετα προβλήματα.

Το σενάριο προσβλέπει σε μία προσπάθεια αξιοποίησης του εκπαιδευτικού πακέτου ρομποτικών κατασκευών Lego Mindstorms και του εκπαιδευτικού λογισμικού NXT-G, σε μια βασισμένη σε σχέδιο εργασίας (project-based), διδακτική προσέγγιση επίλυσης προβλημάτων, χρησιμοποιώντας παράλληλα και το στοιχείο του παιχνιδιού, προκειμένου οι μαθητές της Γ΄ Γυμνασίου να κατανοήσουν βασικές αρχές προγραμματισμού, όπως οι δομές ελέγχου και επανάληψης και να εργαστούν με αυτές.

Ανακεφαλαιώνοντας τα προβλήματα που παρατηρούνται στην τάξη κατά τη διδασκαλία του προγραμματισμού με την παραδοσιακή μέθοδο διδασκαλίας έχει διαπιστωθεί ότι υπάρχει μειωμένη προσοχή και έλλειψη ενδιαφέροντος από την πλευρά των μαθητών, άρνηση ενεργητικής συμμετοχής τους στην εκπαιδευτική διαδικασία και πολύ περισσότερο συνεργασίας μεταξύ τους καθώς και με τον εκπαιδευτικό και τέλος απουσία κάθε δημιουργικής προσπάθειας. Επίσης διαπιστώνεται ότι υπάρχει μεγάλη δυσκολία από τους μαθητές στην ενσωμάτωση των δομών ελέγχου και επανάληψης κατά τη δημιουργία ενός προγράμματος.

Ο σκοπός είναι η εκπαιδευτική ρομποτική σε συνδυασμό με τον οπτικό προγραμματισμό και με την προϋπόθεση της κατάλληλης παιδαγωγικής πλαισίωσης να χρησιμοποιηθεί για την επίλυση των προβλημάτων που παρουσιάζονται κατά τη διδασκαλία του προγραμματισμού και συγκεκριμένα :

1. η αξιοποίηση τη εκπαιδευτικής ρομποτικής σε συνδυασμό με τη δημιουργία κατάλληλου εκπαιδευτικού υλικού (σημειώσεις για το γνωστικό αντικείμενο της ρομποτικής, και φύλλα δραστηριοτήτων, εργασιών και ασκήσεων) να δημιουργήσουν το κατάλληλο περιβάλλον ώστε α) να τραβήξουν την προσοχή και το ενδιαφέρον των μαθητών και β) να πετύχουν την ενεργητική συμμετοχή και τη συνεργασία τους.
2. η διδασκαλία του προγραμματισμού με τη χρήση της εκπαιδευτικής ρομποτικής α) να ενθαρρύνει τους μαθητές ώστε να καταβάλουν δημιουργική προσπάθεια και β) να βοηθήσει στην ενσωμάτωση των δομών ελέγχου και επανάληψης στα προγράμματα των μαθητών με επιτυχία.

Ειδικότερος σκοπός των τελευταίων ενοτήτων του σεναρίου είναι οι μαθητές να μπορούν να αξιοποιούν τις γνώσεις και δεξιότητες που έχουν αποκτήσει για τη δημιουργία και παρουσίαση συνθετικών εργασιών. Είναι σημαντικό η επιλογή μιας συνθετικής δραστηριότητας να βασίζεται στην έμφυτη περιέργεια του μαθητή και την αυτενέργεια του. Ο μαθητής πρέπει να παρακινείται από τον εκπαιδευτικό στο να προσδιορίζει και να αξιοποιεί διάφορες πηγές και μέσα πληροφόρησης. Να συνδυάζει τη θεωρία με την πράξη και να στοχεύει κυρίως στην απόκτηση κριτικών δεξιοτήτων και δεξιοτήτων μεθοδολογικού χαρακτήρα. Οι μαθητές πρέπει να στηριχτούν στις γνώσεις που έχουν αποκτήσει, οι οποίες θα τους βοηθήσουν στον κατάλληλο σχεδιασμό και στην οργάνωση που απαιτείται για να επιτευχθεί το επιδιωκόμενο αποτέλεσμα της κάθε δραστηριότητας στον προγραμματισμένο χρόνο.

Με τις συνθετικές εργασίες οι μαθητές συμμετέχουν ενεργά στη μαθησιακή διαδικασία. Επιδιώκεται έτσι η ανάπτυξη της δημιουργικότητας, της συνεργατικότητας και της ικανότητας επικοινωνίας. Οι μαθητές εκπαιδεύονται να αναπτύξουν κριτικές δεξιότητες για την αντιμετώπιση προβλημάτων με τη χρήση υπολογιστή και να επιλύσουν απλά προβλήματα σε προγραμματιστικό περιβάλλον. Ο εκπαιδευτικός πρέπει να αναπτύξει στους μαθητές την πειραματική και ερευνητική διάθεση, δίνοντας τους απλά και διασκεδαστικά προβλήματα προς επίλυση. Οι μαθητές ενθαρρύνονται να κατανοήσουν το προς επίλυση πρόβλημα, να το αναλύσουν και με τη βοήθεια ενός κατάλληλου προγραμματιστικού περιβάλλοντος να συνθέσουν τη λύση του. Πρέπει να τονιστεί ότι η πειραματική διερεύνηση δε σημαίνει επίδειξη προγραμματιστικών τεχνικών που διεξάγονται από τον εκπαιδευτικό αλλά ανάπτυξη του πειραματικού πνεύματος του μαθητή.

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το σενάριο που περιγράφεται παρακάτω περιλαμβάνει 6 φύλλα εργασίας για τους μαθητές,

Η βασισμένες σε σχέδιο εργασίας δραστηριότητες περιλαμβάνουν 4 βασικές επιμέρους εκπαιδευτικές δραστηριότητες (φάσεις) :

1. τη φάση της εξοικείωσης
2. τη φάση της πρακτικής εφαρμογής και
3. τη φάση της πρόκλησης
4. τη φάση της αξιολόγησης

Η φάση της εξοικείωσης

Η πρώτη φάση περιλαμβάνει δύο συναντήσεις και έχει ως αντικείμενο την εξοικείωση με το πακέτο Lego Mindstorms και το λογισμικό NXT-G.

Στη φάση αυτή χρησιμοποιείται σε μεγάλο βαθμό η καθοδηγούμενη διδασκαλία.

1η συνάντηση:

Δίνεται η αφορμή και το κίνητρο για να προκληθεί το ενδιαφέρον των μαθητών με προβολή βίντεο και παρουσιάσεων σχετικών με τις ρομποτικές κατασκευές. Ρωτούνται οι μαθητές για τις γνώσεις τους πάνω στα ρομπότ (πως είναι ένα ρομπότ, που έχουν δει κάποιο, τι εργασίες μπορεί να εκτελέσει, γιατί τα ρομπότ χρειάζονται αισθητήρες, πως λειτουργούν τα πραγματικά ρομπότ) και μετά από συζήτηση καλούνται να συνθέσουν το ορισμό της έννοιας ρομπότ. Γίνεται μια εκτενής αναφορά στο ρόλο και στις δυνατότητες που μπορούν να έχουν τα ρομπότ στη ζωή μας. Αμέσως μετά, γίνεται επίδειξη της λειτουργίας της έτοιμης ρομποτικής κατασκευής στην οποία έχουν αποθηκευτεί έτοιμα προγράμματα

Στη συνέχεια γίνεται μια λεπτομερή παρουσίαση των τεμαχίων του πακέτου Lego Mindstorms ώστε οι μαθητές, όχι απλά να εξοικειωθούν με τη χρήση όλων των υλικών αλλά και να συλλάβουν τη λειτουργικότητα κάθε τεμαχίου (όπως πχ τα γρανάζια) σε συνδυασμό με τα υπόλοιπα.

Δίνεται έμφαση στην αναγνώριση των έξυπνων στοιχείων εισόδου, δηλαδή των αισθητήρων αφής, φωτός, ήχου και υπερήχων και στην κατανόηση της χρήσης των αισθητήρων. (‘τί’ και ‘πως’ αυτοί ‘μετρούν’) και στον τρόπο λειτουργίας και χειρισμού του μικροεπεξεργαστή NXT.

Ακολουθεί ο χωρισμός από τους ίδιους τους μαθητές σε ομάδες μετά από συνεννόηση και με βάση την επιθυμία τους. Ο εκπαιδευτικός θα πρέπει να καθορίσει τη σύνθεση των ομάδων πριν την εκκίνηση των δραστηριοτήτων καθώς κάθε ομάδα θα συντάξει ξεχωριστά το ίδιο πρόγραμμα. Παράλληλα μοιράζονται φυλλάδια με τα φύλλα εργασίας που θα χρησιμοποιούνταν μέχρι το πέρας των συναντήσεων.

Ως ομάδες πλέον οι μαθητές προχωρούν στην κατασκευή ενός συγκεκριμένου ρομπότ, του tankbot, με βάση τις οδηγίες που τους δίνονται από φυλλάδιο που τους διανεμήθηκε. Σε κάθε ομάδα δίνονται όλα τα απαραίτητα, για την κατασκευή αυτήν, τεμάχια.

2η συνάντηση:

Στη συνέχεια ο εκπαιδευτικός θα παρουσιάσει το προγραμματιστικό περιβάλλον που θα χρησιμοποιηθεί στο σενάριο δείχνοντας τις διεπαφές του και περιγράφοντας θεωρητικά τους διαφορετικούς τρόπους χρήστης του και τους τρόπους πλοήγησης στο περιβάλλον. Αρχικά γίνεται

μια γρήγορη παρουσίαση του προγραμματιστικού περιβάλλοντος NXT-G με σκοπό να έρθουν σε επαφή οι μαθητές με τον τρόπο λειτουργίας, τις δυνατότητες και τα εργαλεία του λογισμικού NXT-G. Στη συνέχεια πραγματοποιείται μια επίδειξη της διαδικασίας προγραμματισμού μέσα από το περιβάλλον του NXT-G και της μεταφόρτωσης (κατέβασμα) του προγράμματος στο μικροεπεξεργαστή NXT της ρομποτικής κατασκευής.

Συγκεκριμένα οι μαθητές παρακολουθούν τη διαδικασία συγγραφής ενός νέου προγράμματος στο οπτικό περιβάλλον του NXT-G, την αποστολή του στη ρομποτική κατασκευή μας (το tankbot) και την απόδοση της επιθυμητής συμπεριφοράς στο ρομπότ μας με την εκτέλεση αυτού του προγράμματος.

Στη συνέχεια με στόχο να εξοικειωθούν οι μαθητές στη χρήση του NXT-G και να αποκτήσουν την απαραίτητη ευχέρεια στο χειρισμό του ώστε να είναι σε θέση να ανταποκριθούν με άνεση στις δραστηριότητες που θα ακολουθήσουν, οι μαθητές εργάζονται με το, ειδικά για το σκοπό αυτό, πρώτο φύλλο εργασίας 'Εξοικείωση με το NXT-G'. Το φύλλο αυτό έχει τη μορφή αναλυτικών βήμα προς βήμα οδηγιών, με φωτογραφίες για το αναμενόμενο αποτέλεσμα κάθε βήματος και οι μαθητές εξασκήθηκαν στις βασικές λειτουργίες του προγράμματος..

Η φάση της πρακτικής εφαρμογής

Η δεύτερη φάση της πρακτικής εφαρμογής περιλαμβάνει δύο συναντήσεις και έχει ως αντικείμενο την επαφή με τις δυνατότητες του λογισμικού NXT-G και συγκεκριμένα με τις εντολές ελέγχου με τη χρήση αισθητήρων και τις προγραμματιστικές δομές τις επανάληψης και της επιλογής..

Συνειδητά δεν γίνεται απόπειρα διδασκαλίας των αρχών προγραμματισμού, και οι απαιτούμενες έννοιες (ροή προγράμματος, λογική επιλογή, βρόγχος επανάληψης, κ.λ.π.) ανακαλύπτονται με βιωματική προσέγγιση, μέσα από συνδυασμένες δραστηριότητες συζήτησης και πειραματισμού.

Όσον αφορά την πορεία της διδασκαλίας για κάθε νέο στοιχείο προγραμματισμού, γίνεται αρχικά μία σύντομη παρουσίαση των νέων εννοιών, και μετά οι μαθητές εργάζονται πάνω σε συγκεκριμένα φύλλα εργασιών. Σε αυτά προτού προχωρήσουν στην υλοποίηση δικών τους προγραμμάτων, καλούνται να μελετήσουν έτοιμα προγράμματα, να δώσουν λεκτικές περιγραφές σχετικά με το αποτέλεσμα του προγράμματος στη συμπεριφορά του ρομπότ και κατόπιν παροτρύνονται να τροποποιήσουν τις παραμέτρους των έτοιμων αυτών προγραμμάτων. Με τη μελέτη των έτοιμων προγραμμάτων, δίνεται η δυνατότητα να επισημανθούν σημαντικά στοιχεία, να αναδειχθούν συντακτικές ιδιαιτερότητες και να αποσαφηνιστούν δυσκολονόητα, για τους μαθητές, σημεία. Στη συνέχεια, οι μαθητές καλούνται να υλοποιήσουν συγκεκριμένες ασκήσεις – προγράμματα για τον έλεγχο της συμπεριφοράς του ρομπότ. Παράλληλα, παροτρύνονται να πειραματιστούν, να παρατηρήσουν και να καταγράψουν την επίδραση που έχουν οι αλλαγές των τιμών των διαφόρων παραμέτρων του προγράμματος, στη λειτουργία των ρομπότ.

Μέσα από αυτές τις διαδικασίες οι μαθητές εξοικειώνονται με την ιδέα ότι μια σειρά εντολών οδηγεί σε συγκεκριμένες δράσεις, και αντιστρόφως, αναγνωρίζουν ότι μια αλληλουχία δράσεων μπορεί να υλοποιηθεί από μια σειρά εντολών. Έρχονται σε επαφή με τις «λογικές ακολουθίες» ως μια αλληλουχία απλών εκτελέσιμων βημάτων και εξοικειώνονται με τις αρχές του λογικού προγραμματισμού και τις συνθήκες ελέγχου.

Στη φάση αυτή χρησιμοποιείται σε μικρότερο βαθμό η καθοδηγούμενη διδασκαλία.

3η συνάντηση:

Οι μαθητές εργαζόμενοι με το δεύτερο φύλλο εργασίας έρχονται παράλληλα και σε μία πρώτη επαφή με τους κινητήρες και πειραματίζονται με τις εντολές του λογισμικού που ελέγχουν τη λειτουργία τους (εντολές εξόδου και αναμονής για περιστροφή). Στα πλαίσια των δραστηριοτήτων του φύλλου αυτού αναπτύσσουν απλά προγράμματα χρησιμοποιώντας ηλεκτροκινητήρες και προγραμματίζουν το αυτοκινητάκι tankbot, το οποίο μπορούσε να κινείται προς τα εμπρός και προς τα πίσω για ένα ορισμένο χρονικό διάστημα.

Στη συνέχεια παρουσιάζονται από τον υπεύθυνο εκπαιδευτικό οι προγραμματιστικές δομές της επανάληψης. Ο κύριος στόχος είναι οι μαθητές να εξοικειωθούν με τον προγραμματισμό του ρομπότ έτσι ώστε να προετοιμαστούν κατάλληλα για την υλοποίηση των προγραμμάτων στις προκλήσεις που ακολουθούν.

4η συνάντηση:

Οι μαθητές εργαζόμενοι με το τρίτο φύλλο εργασίας καλούνται να πειραματιστούν με το σύνολο των βασικών δυνατοτήτων του NXT-G με στόχο να γνωρίσουν τις βασικές αρχές προγραμματισμού και να ολοκληρώσουν μία σειρά από προκλήσεις, σταδιακά αυξανόμενης δυσκολίας, οι οποίες απαιτούν προγραμματιστικές δεξιότητες ώστε να αποκτήσουν ευχέρεια στον προγραμματισμό ενός ρομπότ.

Αρχικά παρουσιάζονται από τον εκπαιδευτικό οι προγραμματιστικές δομές της επιλογής σε συνδυασμό με τις δυνατότητες του προγραμματιστικού περιβάλλοντος NXT-G.

Οι μαθητές αρχικά εξοικειώνονται με τη χρήση των βασικών εντολών εισόδου (αναμονή για είσοδο από αισθητήρα) κάνοντας χρήση των αισθητήρων υπερήχων, ήχου, αφής και φωτός. Οι μαθητές αφού κατανοήσουν, τον τρόπο λειτουργίας των αισθητήρων και τον τρόπο προγραμματισμού τους, καλούνται να πειραματιστούν με το αυτοκινητάκι tankbot και συγκεκριμένα, να ελέγξουν την κίνηση του χρησιμοποιώντας αισθητήρες υπερήχων, ήχου, αφής και φωτός.

Στο στάδιο αυτό ζητείται από τους μαθητές να προγραμματίσουν το αυτοκίνητο tankbot, προσδίδοντάς του διάφορες συμπεριφορές ολοένα και πιο σύνθετες.

Η φάση της πρόκλησης

Στη τελική φάση της πρόκλησης οι μαθητές καλούνται να αντιμετωπίσουν μόνοι τους με όσο το δυνατόν ελάχιστη καθοδήγηση πλέον, δύο σύνθετα προβλήματα προγραμματισμού κατά τη διάρκεια των δύο τελευταίων δώρων συναντήσεων.

Σε αυτήν τη φάση το περιβάλλον μάθησης έχει ως ουσιαστικό χαρακτηριστικό του ένα μοντέλο επικοινωνίας, που επιτρέπει στους συμμετέχοντες να αλληλεπιδράσουν μέσα σε ελεγχόμενες συνθήκες ανταγωνισμού.

Σε αυτή τη φάση ζητείται από τους μαθητές να προσδώσουν στο αυτοκινητάκι tankbot συγκεκριμένες συμπεριφορές με στόχο την επίλυση κάθε φορά ρεαλιστικών προβλημάτων, στα οποία πρέπει να ανταπεξέλθουν με βάση τις γνώσεις που αποκόμισαν από τις προηγούμενες φάσεις.

Αρχικά γίνεται παρουσίαση των προβλημάτων από τον υπεύθυνο καθηγητή με λεπτομερή περιγραφή των συνθηκών που πρέπει να λάβουν υπόψη τους οι μαθητές. Ακολουθεί συζήτηση

πάνω στον τρόπο επίλυσης του προβλήματος. Στη συνέχεια οι μαθητές έχοντας κατανοήσει το πρόβλημα προχωρούν στην ανάλυση του με τη μορφή αλγορίθμου, στην κωδικοποίηση του αλγορίθμου σε πρόγραμμα του NXT-G, στην εκτέλεση προγράμματος και στην απαραίτητη διόρθωση (όταν χρειαζόταν). Όλες αυτές τις κινήσεις τους τις καταγράφουν σε συγκεκριμένο φύλλο δραστηριοτήτων

Ο υπεύθυνος εκπαιδευτικός φροντίζει να είναι σαφείς οι στόχοι που πρέπει να επιτευχθούν και να κατανοήσουν οι ομάδες ότι συμμετέχοντας ενεργά στην παραπάνω εκπαιδευτική διαδικασία όλοι βγαίνουν νικητές.

5η συνάντηση:

Στη συνάντηση αυτή τίθεται το πρόβλημα της αποφυγής εμποδίων από το αυτοκινητάκι tankbot. Στο αυτοκινητάκι αυτό αρχικά τοποθετείται ένας ειδικά σχεδιασμένος βραχίονας με έναν αισθητήρα αφής και ζητείται από τους μαθητές να προγραμματίσουν το όχημα τους, ώστε να είναι σε θέση να βγει από μία ειδικά διαμορφωμένη σπηλιά. Για αυτό το σκοπό στήνεται από τον υπεύθυνο καθηγητή ένας κλειστός χώρος με τοίχιο και με ένα αρκετά μεγάλο άνοιγμα. Οι μαθητές καλούνται ως ομάδες ξεχωριστά, καταρχήν να αναλύσουν το πρόβλημα συζητώντας τις προτάσεις τους, τις οποίες ταυτόχρονα και καταγράφουν. Προς διευκόλυνση τους ζητείται να σκεφτούν τις ενέργειες που θα έκαναν αυτοί βήμα-βήμα, στην υποθετική περίπτωση που είχαν κλειστά τα μάτια τους και έπρεπε να κινηθούν σε ένα άγνωστο χώρο έχοντας μόνο το ένα χέρι τους, μπροστά τους, τεντωμένο και ακίνητο.

Στη συνέχεια στο αυτοκινητάκι τοποθετείται ένας δεύτερος ειδικά σχεδιασμένος βραχίονας που περιλαμβάνει δύο αισθητήρες αφής, δεξιό και αριστερό και αυτή τη φορά τους ζητείται να σκεφτούν τις ενέργειες που θα έκαναν αυτοί βήμα-βήμα, στην υποθετική περίπτωση που είχαν κλειστά τα μάτια σας και έπρεπε να κινηθούν σε ένα άγνωστο χώρο έχοντας και τα δύο χέρια τους, μπροστά τους ελαφρά ανοιγμένα, τεντωμένα και ακίνητα.

6η συνάντηση:

Στη συνάντηση αυτή τίθεται το πρόβλημα αν και κατά πόσο το αυτοκινητάκι tankbot μπορεί να ακολουθήσει μία μαύρη γραμμή με αρκετές καμπύλες, η οποία είναι ζωγραφισμένη σε μία λευκή επιφάνεια από χαρτί.

Οι μαθητές πρέπει διαπιστώσουν την ανάγκη χρήσης του αισθητήρα φωτός. Στο αυτοκινητάκι αυτό αρχικά τοποθετείται ένας ειδικά σχεδιασμένος βραχίονας με έναν αισθητήρα φωτός και ζητείται από τους μαθητές να προγραμματίσουν το όχημα τους ώστε να είναι σε θέση να ακολουθήσει τη συγκεκριμένη γραμμή. Για να γίνει η μύηση των μαθητών στην κατανόηση του τρόπου χειρισμού του συγκεκριμένου προβλήματος σταδιακά, πρώτα τους γίνεται μία επίδειξη με ένα αυτοκινητάκι, στο οποίο ο υπεύθυνος εκπαιδευτικός έχει ήδη φορτώσει το απαραίτητο πρόγραμμα. Οι μαθητές καλούνται να παρατηρήσουν τη συμπεριφορά του οχήματος και να προσπαθήσουν να αναγνωρίσουν τις εντολές με τις οποίες είναι εφοδιασμένο το όχημα αυτό. Προς διευκόλυνση τους ζητείται να σκεφτούν τις ενέργειες που θα έκαναν αυτοί βήμα-βήμα, στην υποθετική περίπτωση που υπήρχε μία αόρατη γραμμή χαραγμένη στο πάτωμα, την οποία μπορούσαν να αντιληφθούν μόνο τη στιγμή που θα τη συναντούσαν (πχ μία γραμμή από αόρατη

μελάνη την οποία δεν βλέπουν, αλλά μόλις πατήσουν πάνω της ειδοποιούνται από ένα σφύριγμα.).

Στο σημείο αυτό για την καλύτερη κατανόηση του προβλήματος κρίνεται σκόπιμο να στηθεί και το ανάλογο παιχνίδι ρόλων όπου ένας μαθητής καλείται να υποδυθεί το ρομπότ. Από τη βιωματική αυτή δραστηριότητα, τα παιδιά πρέπει να οδηγηθούν στο συμπέρασμα ότι για να αποφευχθεί η δυσκολία στην κίνηση του ρομπότ, πρέπει αυτό να κινείται σχετικά αργά ώστε να αναγνωρίζει το χρώμα που βλέπει, ο δρόμος πάνω στον οποίο κινείται να διαφέρει χρωματικά από τον υπόλοιπο χώρο, το φάρδος του δρόμου να είναι μεγάλο και ο αισθητήρας να είναι όσο το δυνατό χαμηλότερα στο έδαφος.

Στη συνέχεια στο αυτοκινητάκι τοποθετείται ένας δεύτερος ειδικά σχεδιασμένος βραχίονας που περιλαμβάνει δύο αισθητήρες φωτός, δεξιό και αριστερό και αυτή τη φορά τους ζητείται να βελτιώσουν την προηγούμενη συμπεριφορά του οχήματος τους.

Κατά τη διάρκεια των 2 τελευταίων συναντήσεων οι μαθητές παρουσιάζουν το αποτέλεσμα τους στην τάξη και αναπτύσσεται συζήτηση με βάση την κρίση των έργων τους.

Η φάση της αξιολόγησης

Στη διάρκεια της υλοποίησης του σεναρίου ο εκπαιδευτικός θα πρέπει να ελέγχει τα συμπεράσματα και τις απαντήσεις των μαθητών, να συνεργάζεται μαζί τους, να τους καθοδηγεί ώστε να αντιλαμβάνονται καλύτερα τα αποτελέσματα τους και να τους ενθαρρύνει να συνεχίσουν τη διερεύνηση.

Στο τέλος του εγχειρήματος όλοι οι μαθητές, συμπληρώνουν δύο ερωτηματολόγια ένα για να αξιολογηθεί κατά πόσο έχουν αφομοιώσει τη νέα γνώση και ένα δεύτερο σε μία προσπάθεια να καταγραφούν γραπτώς οι εντυπώσεις τους από τις δραστηριότητες με τις οποίες ασχολήθηκαν.

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ - ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Ένα μεγάλο πλεονέκτημα από τη χρήση των Lego Mindstorms στη διδασκαλία του προγραμματισμού αποτελεί το γεγονός ότι η εκτέλεση των προγραμμάτων που δημιουργούν οι μαθητές δεν γίνεται στον ίδιο τον υπολογιστή στον οποίο θα δημιουργήσουν το πρόγραμμα, αλλά σε μία άλλη εντελώς διαφορετική μηχανή η οποία λειτουργεί αυτόνομα και ανεξάρτητα με τον παραπάνω υπολογιστή : Όταν οι μαθητές μαθαίνουν να προγραμματίζουν με τις γλώσσες γενικού σκοπού χρησιμοποιούν τον υπολογιστή ως μηχανή για να αναπτύξουν το πρόγραμμά τους και ταυτόχρονα χρησιμοποιούν την ίδια μηχανή για να εκτελέσει το πρόγραμμα ώστε να ελέγξουν τη συμπεριφορά του ίδιου του υπολογιστή. Προγραμματίζοντας όμως τα Lego Mindstorms, χρησιμοποιείται ο υπολογιστής για την ανάπτυξη του προγράμματος, αλλά η εκτέλεση του πραγματοποιείται από μια άλλη μηχανή - το μικρουπολογιστή NXT - που ελέγχει με τη σειρά του τη συμπεριφορά του ρομπότ.

Όπως σημειώνουν οι Νικολός Δ., Καρατράντου Α. και Παναγιωτακόπουλος Χ (Αξιοποίηση του MicroWorlds EX Robotics για την κατανόηση βασικών δομών προγραμματισμού, 2006) τα Lego Mindstorms αλλάζουν ριζικά τη λογική διδασκαλίας του προγραμματισμού αφού οι μαθητές δεν επικεντρώνονται πλέον σε μία σειρά υπολογισμών αλλά ο σκοπός τους είναι να αποδώσουν μία

συγκεκριμένη συμπεριφορά σε μία ρομποτική κατασκευή ώστε αυτή να μπορέσει να αλληλεπιδρά με τον ενδεδειγμένο τρόπο με το άμεσο φυσικό περιβάλλον του. Τα δεδομένα του προβλήματος δεν είναι πλέον απλές τιμές αλλά οντότητες που θα παρατηρηθούν και τα αποτελέσματα δεν είναι νέες τιμές αλλά ενέργειες μέσα σε μια δυναμική διαδικασία. Μια τέτοια λογική υλοποιείται σε ένα αντικειμενοστραφές πλαίσιο όπου η συσχέτιση ανάμεσα στη συμπεριφορά του φυσικού αντικειμένου (της ρομποτικής κατασκευής) και στο πρόγραμμα του μαθητή είναι ισχυρή και άμεσα παρατηρήσιμος (Lawhead et al., 2002).

Η χρήση της εκπαιδευτικής ρομποτικής μπορεί να βελτιώσει τις δεξιότητες συνεργασίας, την αυτοπεποίθηση, τη δημιουργικότητα, τα κίνητρα των παιδιών και τις δεξιότητες χειρισμού του υπολογιστή (Palumbo & Palumbo, 1993; Wagner, 1998). Η διδασκαλία της ρομποτικής είναι κατάλληλη για μαθητές ανεξάρτητα από την ηλικία και το υπόβαθρό τους και αποτελεί έναν τρόπο ενθάρρυνσης της μάθησης. Επιπλέον παρέχει στα παιδιά το κατάλληλο περιβάλλον για να διερευνήσουν τα προβλήματα του πραγματικού κόσμου, συμβάλλοντας έτσι στη δημιουργία τεχνολογικά ενημερωμένων πολιτών σύμφωνα με τις ανάγκες της σύγχρονης κοινωνίας.

Πρόκειται για μια εναλλακτική προσέγγιση διδασκαλίας του προγραμματισμού, η οποία στηρίζεται στη χρήση φυσικών μηχανικών μοντέλων, με τα οποία οι μαθητές προσανατολίζονται ως πρώτο βήμα στην κατασκευή ενός ρομποτικού μηχανισμού και μετά από κατάλληλο σχεδιασμό και αρκετούς πειραματισμούς οδηγούνται στην επίλυση πραγματικών προβλημάτων. Στο πλαίσιο των δραστηριοτήτων αυτών οι μαθητές δημιουργούν φυσικά μοντέλα (ένα αυτοκίνητο, ένα σπίτι κα,) τα οποία αφού τα εφοδιάσουν με το κατάλληλο πρόγραμμα, αυτά είναι σε θέση να συλλέγουν πληροφορίες από το περιβάλλον και να αντιδρούν ανάλογα με τα ερεθίσματα που λαμβάνουν. Στο πλαίσιο της εκπαιδευτικής διαδικασίας αυτής οι μαθητές εξασκούνται στον προγραμματισμό για να καθορίσουν τη συμπεριφορά του μοντέλου που έχουν δημιουργήσει οι ίδιοι, όχι στην οθόνη του υπολογιστή τους αλλά στο φυσικό τους περιβάλλον.

Οι Καγκάνης Κ, Δαγδιλέλης Β., Σατρατζέμη Μ. και Ευαγγελίδης Γ. (Μία μελέτη Περίπτωσης της Διδασκαλίας του Προγραμματισμού στη Β/θμια Εκπαίδευση με τα LEGO Mindstorms, 2005) έχουν διαπιστώσει ότι η μέθοδος αυτή λαμβάνει υπόψη τα ενδιαφέροντα των μαθητών και τους εμπλέκει σε δραστηριότητες που έχουν πραγματικά χρησιμότητα και σημασία για αυτούς. Χαρακτηρίζεται από υψηλό βαθμό αλληλεπίδρασης μεταξύ του υπολογιστή και του πραγματικού αντικειμένου, με αποτέλεσμα, ο μαθητής να μπορεί να αναγνωρίσει εύκολα την αντιστοίχιση που υπάρχει ανάμεσα στις εντολές του προγράμματος και τις αντιδράσεις του μοντέλου και να παρατηρήσει τις συνέπειες που έχουν στη συμπεριφορά του μοντέλου, οι αλλαγές που πραγματοποιεί στο πρόγραμμα. (Eden et al., 1996). Τα φυσικά μοντέλα λοιπόν προσφέρουν άμεση ανατροφοδότηση στους μαθητές ώστε να μπορούν να ελέγχουν την αποτελεσματικότητα των προγραμμάτων τους (Barnes, 2002). Η δυνατότητα αυτή βοηθά τους μαθητές να οδηγούνται σταδιακά σε καλύτερες, αποτελεσματικότερες, πληρέστερες και ακριβέστερες λύσεις.

Συμπερασματικά όπως διαπιστώνεται από τη μελέτη περίπτωσης των Καγκάνη Κ., Δαγδιλέλη Β., Σατρατζέμη Μ. και Ευαγγελίδη Γ. (2005) με τα φυσικά μηχανικά μοντέλα όπως αυτό της Lego: α) πετυχαίνεται υψηλός βαθμός αλληλεπίδρασης μεταξύ υπολογιστή και πραγματικού αντικειμένου, β) υπάρχει άμεση ανατροφοδότηση, γ) υπάρχει πειραματισμός και ενεργός συμμετοχή από τους μαθητές, δ) αναπτύσσεται η κριτική σκέψη, ε) καλλιεργείται η δημιουργική σκέψη, η διορατικότητα και η πρωτοτυπία, στ) υπάρχει άμεση εμπειρία και ο μαθητής απαλλάσσεται από την εκμάθηση και

απομνημόνευση συντακτικών κανόνων μιας γλώσσας προγραμματισμού και ζ) υλοποιείται ένα είδος εξατομικευμένης μάθησης αφού ο εκπαιδευτικός διαθέτει περισσότερο χρόνο για κάθε μαθητή και ο κάθε μαθητής εργάζεται με το δικό του ρυθμό μάθησης.

Όμως, η χρήση φυσικών μοντέλων κατά τη διδασκαλία των αρχών του προγραμματισμού μπορεί να δημιουργήσει και διδακτικά προβλήματα, τα σημαντικότερα από τα οποία οφείλονται: α) σε φυσικούς περιορισμούς (περιορισμοί που μπορεί να δημιουργήσει η τριβή στην κίνηση ενός αντικειμένου π.χ. ενός κινητήρα), β) τεχνικούς περιορισμούς του υλικού (φόρτωσης του προγράμματος στον επεξεργαστή κ. ά., γ) χρονικούς περιορισμούς (χρόνος εκμάθησης του συστήματος) και δ) περιορισμούς κόστους επένδυσης (μεγάλη οικονομική επιβάρυνση για την απόκτηση και συντήρηση-επικαιροποίηση απαιτούμενου αριθμού συστημάτων).

Στο πλαίσιο της εργασίας αυτής, για τον προγραμματισμό των ρομποτικών κατασκευών της Lego στα πλαίσια των εκπαιδευτικών δραστηριοτήτων για την κατανόηση των βασικών αρχών, εννοιών και δομών του προγραμματισμού από μαθητές του Γυμνασίου επιλέχθηκε η χρήση του προγραμματιστικού περιβάλλοντος NXT-G που συνοδεύει το πακέτο Lego Mindstorm NXT Education..

Το λογισμικό NXT-G είναι ένα βασισμένο σε ένα περιβάλλον οπτικού προγραμματισμού που επιτρέπει στο χρήστη να προγραμματίσει χρησιμοποιώντας εικόνες που αναπαριστούν όλους τους τύπους δεδομένων και τις βασικές εντολές και δομές. Έτσι, η διδασκαλία του προγραμματισμού δεν ακολουθεί το τυπικό μοντέλο διδασκαλίας αλλά μια κατασκευαστική προσέγγιση (Τσοβόλας & Αντωνίου, 2005). Ένα από τα βασικότερα πλεονεκτήματα τέτοιων γλωσσών προγραμματισμού είναι η ελαχιστοποίηση των συντακτικών κανόνων με τις οποίες έρχεται αντιμέτωπος ο μαθητής. Η διαδικασία σύνταξης ενός προγράμματος είναι αρκετά απλή και βασίζεται στη σωστή σύνδεση των κατάλληλων εικονιδίων. Οι χρήστες στη γλώσσα προγραμματισμού NXT-G, μαθαίνουν να προγραμματίζουν μέσω οπτικών εντολών χρησιμοποιώντας την τεχνική του συρσίματος (drag & drop) για να δημιουργήσουν προγράμματα και δε χρειάζεται να έχουν γνώσεις πάνω στο υλικό (Κασκάλης κ.α., 2001). Η ανάπτυξη προγραμμάτων με τη χρήση εικονο-εντολών απαλλάσσει τους μαθητές από την εκμάθηση μιας γλώσσας προγραμματισμού και την απομνημόνευση των συντακτικών της κανόνων. Ειδικά τα εικονίδια λειτουργίας των εξόδων είναι προφανή. Στο περιβάλλον συγγραφής του κώδικα γίνεται ταυτόχρονα και συντακτικός έλεγχος, έτσι απαλλάσσεται ο προγραμματιστής από συντακτικά λάθη. (Τσοβόλας & Κόμης, 2005).

Η διδασκαλία του προγραμματισμού στα πλαίσια της προσέγγισης αυτής επικεντρώνεται στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων και σχεδίασης αλγορίθμων και όχι στην εκμάθηση της γλώσσας προγραμματισμού και είναι ιδιαίτερα αποτελεσματική (Ξυνόγαλος κ.α. 2000, Lawhead et al 2002).

Ο προγραμματισμός με τη χρήση του λογισμικού NXT-G λοιπόν αποφέρει θετικά μαθησιακά αποτελέσματα με χαρακτηριστικό τη χρήση της σωστής και επιθυμητής δομής σε κάθε περίπτωση. Επίσης, αυξάνει και διατηρεί το κίνητρο των μαθητών για τον προγραμματισμό.

Στα μαθήματα του προγραμματισμού ορισμένες έννοιες όπως οι δομές επιλογής και ειδικότερα οι εμφωλευμένες δομές επιλογής δεν είναι κατανοητές από τους μαθητές. Καλό θα είναι, μέσα από πειραματισμούς, προσομοιώσεις και στοχευμένες ερωτήσεις, οι μαθητές να μπορούν να οικοδομούν νέες γνώσεις πάνω στις υπάρχουσες και να αποσαφηνίζουν τυχόν μπερδεμένες έννοιες.

Όσον αφορά τις τελευταίες ενότητες, η επίτευξη των συνθετικών εργασιών μπορεί να γίνει είτε ατομικά είτε ομαδοσυνεργατικά. Η συνεργασία ομάδων βοηθάει στην ανάπτυξη της ικανότητας επικοινωνίας, στην ένταξη των μοναχικών μαθητών στο κοινωνικό σύνολο του σχολείου και εξασφαλίζει την ενεργό συμμετοχή των μαθητών στη μαθησιακή διαδικασία με φυσικό και αβίαστο τρόπο. Η εργασία σε ομάδες περιορίζει την ανταγωνιστικότητα μεταξύ των μαθητών και συγχρόνως κάνει τους μαθητές να αισθάνονται υπεύθυνα και σημαντικά άτομα, ικανά να αναλάβουν αρμοδιότητες και να συμβάλλουν στην ολοκλήρωση μιας κοινής εργασίας. Μία κατάλληλη μέθοδος για την εκπόνηση ομαδικών συνθετικών εργασιών είναι η μέθοδος Project ή η μέθοδος των σχεδίων.

Καθώς οι μαθητές βρίσκονται στα πρώτα στάδια της εφηβείας και ολοκληρώνουν την περίοδο της αφαιρετικής σκέψης (σύμφωνα με τα στάδια του Piaget), η διδασκαλία της Πληροφορικής πρέπει να ενθαρρύνει τους μαθητές:

- να συνεργάζονται για την εκτέλεση συγκεκριμένων μεγάλων δραστηριοτήτων.
- να αναπτύσσουν πρωτοβουλίες και να διατηρούν την ανεξαρτησία τους με σεβασμό στην άποψη των άλλων.
- να σχεδιάζουν, να προγραμματίζουν και να οργανώνουν τις εργασίες μιας ομάδας.
- να αναγνωρίζουν τη συμβολή της ομαδικής εργασίας στην παραγωγή έργου.
- να συζητούν και να κρίνουν την εργασία τους και την εργασία των άλλων.
- να αυτενεργούν και να πειραματίζονται, ώστε να ανακαλύπτουν μόνοι τους τη γνώση.

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Για τη διδασκαλία του σεναρίου μας θα χρειαστεί ένα εργαστήριο πληροφορικής που να έχει εγκατασταθεί το λογισμικό Mindstorms NXT-G Edu. Επίσης επιβάλλεται να υπάρχει ένας video-projector για να μπορεί ο εκπαιδευτικός να παρουσιάζει τις απαραίτητες έννοιες για τον προγραμματισμό που θα χρειαστούν να γνωρίζουν οι μαθητές καθώς και το προγραμματιστικό περιβάλλον.

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Οι πιθανές δυσκολίες των μαθητών σε ζητήματα αλγοριθμικής σκέψης πολλές φορές αποτελεί εμπόδιο για την εκμάθηση αρχών προγραμματισμού. Τα εισαγωγικά μαθήματα προγραμματισμού είναι πολλές φορές απογοητευτικά τόσο για τους μαθητές όσο και για τον ίδιο τον καθηγητή. Όπως αναφέρεται και από τους Καγκάνη Κ, Δαγδιλέλη Β., Σατρατζέμη Μ. και Ευαγγελίδη Γ. (Μία μελέτη Περίπτωσης της Διδασκαλίας του Προγραμματισμού στη Β/θμια Εκπαίδευση με τα LEGO Mindstorms, 2005) ένας από τους σημαντικότερους παράγοντες στον οποίο έχει διαπιστωθεί ότι οφείλονται οι δυσκολίες κατά την εκμάθηση του προγραμματισμού είναι η παραδοσιακή προσέγγιση της διδασκαλίας των αρχών του προγραμματισμού : οι μαθητές

διδάσκονται μια γλώσσα γενικού σκοπού (Pascal, Basic, C κλπ) σε ένα επαγγελματικό περιβάλλον προγραμματισμού το οποίο είναι προσανατολισμένο στην επίλυση μαθηματικών προβλημάτων που περιλαμβάνει την επεξεργασία μιας σειράς αριθμών και συμβόλων, γεγονός που απέχει από τις πραγματικές διδακτικές ανάγκες των μαθητών (Ξυνόγαλος κ.α. 2000).

Οι εμπορικές γλώσσες προγραμματισμού περιλαμβάνουν ένα μεγάλο λεξιλόγιο από εντολές, το οποίο πρέπει να χρησιμοποιηθεί ακολουθώντας ένα ιδιαίτερο συντακτικό, γεγονός που προσανατολίζει λανθασμένα τους μαθητές κατά τη χρήση μιας συγκεκριμένης γλώσσας προγραμματισμού, να ασχολούνται περισσότερο με τις τεχνικές λεπτομέρειες της γλώσσα αυτής, παρά να τη χρησιμοποιούν ως μέσο για να αναπτύξουν τη λογική σκέψη και να αποκτήσουν τις δεξιότητες της στρατηγικής που απαιτείται από το πρόβλημα για να οδηγηθούν στην επίλυση του.

Με την προσέγγιση αυτή οι μαθητές δυσκολεύονται να εμπεδώσουν τις καινούριες έννοιες. Τα προβλήματα που καλούνται να επιλύσουν οι μαθητές είναι μαθηματικού τύπου και απαιτούν από αυτούς να προχωρήσουν μηχανικά στην εκτέλεση μίας σειράς μαθηματικών υπολογισμών και στην εμφάνιση του τελικού αποτελέσματος. Επιπλέον τα προβλήματα αυτά είναι ξένα με τα προβλήματα που έχουν βιώσει οι μαθητές στην καθημερινότητα τους. Ως αποτέλεσμα των παραπάνω είναι η δημιουργία φραγμών που περιορίζουν την αντίληψη των μαθητών στην ανάλυση του προβλήματος και το σχεδιασμό της επίλυσης του. Έτσι οι μαθητές δίνουν περισσότερη βαρύτητα στην εκμάθηση της ίδιας της γλώσσας, παρά στη μεθοδολογία για την επίλυση προβλημάτων, και ασχολούνται με προβλήματα που δεν τους φαίνονται ενδιαφέροντα (Δαγδιλέλης, 1986).

Οι Ματάνας κ.α. (2005) επεκτείνουν το πρόβλημα στη διδασκαλία του προγραμματισμού αναφέροντας ότι ακόμα και όταν οι μαθητές κατανοήσουν τα βήματα που πρέπει να γίνουν για την επίλυση του προβλήματος στη φυσική γλώσσα, δυσκολεύονται στη μεταφορά του αλγορίθμου αυτού στην αυστηρή σύνταξη και δομή μίας γλώσσας προγραμματισμού.

Μία από τις δυσκολίες, επίσης, που αντιμετωπίζουν οι μαθητές όταν καλούνται να επιλύσουν ένα πρόβλημα προγραμματισμού, είναι οι αναπαραστάσεις που απαιτείται να οικοδομήσουν κατά τη διάρκεια της επίλυσής του (Κόμης, 2005α). Οι Ατματζίδου Σ., Μαρκέλης Η. και Δημητριάδης Σ. (Χρήση των LEGO Mindstorms στο Δημοτικό και Λύκειο. Το παιχνίδι ως έναυσμα μάθησης, 2008) σημειώνουν ότι τα προγραμματιστικά περιβάλλοντα στα οποία δεν παρέχεται η δυνατότητα προγραμματισμού με οπτικό τρόπο οι μαθητές κατανοούν δύσκολα τον τρόπο με τον οποίο εκτελείται ένα πρόγραμμα και τη διαδικασία με την οποία συντελείται η είσοδος και η έξοδος των δεδομένων.

Ένα ιδιαίτερο κεφάλαιο στο σύνολο των προβλημάτων που προκύπτουν κατά τη διδασκαλία του προγραμματισμού έχει να κάνει με τις δυσκολίες στη διδασκαλία των δομών επιλογής και επανάληψης.

Οι σχετικές δυσκολίες αφορούν στα:

- Αδυναμία γενίκευσης.
- Ανεπαρκή νοητικά μοντέλα για τις δομές επανάληψης και επιλογής, κυρίως για τις εμφωλευμένες. Επιπλέον, μερικές φορές τα λάθη στα νοητά μοντέλα, των αρχάριων

προγραμματιστών είναι ανεπαίσθητα και δύσκολα να εντοπιστούν.

- Αδυναμία εντοπισμού της σειράς εντολών.

Μία συστηματική απαρίθμηση για τις συγκεκριμένες δυσκολίες μπορούμε να συναντήσουμε στη μελέτη περίπτωσης των Χαρίση Χ. και Μικρόπουλο Τ. Α (Ρομποτική, Οπτικός Προγραμματισμός και Βασικές Προγραμματιστικές Δομές, 2008). Στη μελέτη αυτή διαβάζουμε ότι : Οι Τζιμογιάννης και Γεωργίου που εντοπίζουν δυσκολίες μαθητών σε δομές ελέγχου, τις δικαιολογούν στη διδασκαλία της γλώσσας προγραμματισμού καθαυτής και όχι στη μεθοδολογία επίλυσης προβλημάτων (1999). Ο Carlisle αναφέρει προβλήματα αρχαρίων στην κατανόηση επαναληπτικών δομών και προτείνει διαγραμματικές αναπαραστάσεις τους για την κατανόησή τους (2000). Δυσκολίες που αφορούν στις επαναληπτικές δομές αναφέρονται σε μαθητές Γ' Λυκείου και εντοπίζονται στον καθορισμό των συνθηκών ελέγχου, στον προσδιορισμό των χαρακτηριστικών των επαναληπτικών δομών και στην αναγνώριση αρχικής τιμής (Γρηγοριάδου κ.ά., 2004). Παρόμοιου τύπου δυσκολίες αναφέρονται και δικαιολογούνται στη δυσκολία χειρισμού μεταβλητών (Sajaniemi & Kuittinen, 2005).

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ - ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ

Κατά την εκτέλεση του σεναρίου εκτιμάται ότι δεν υπάρξουν σημαντικά προβλήματα σε σχέση με την ταχύτητα λειτουργίας του Η/Υ - Λογισμικού. Συνεπώς δεν θα υπάρξουν προβλήματα εκκίνησης του λογισμικού ή δυσλειτουργίες που θα επηρεάσουν το μάθημα (διδασκτικός θόρυβος). Επίσης το διδακτικό συμβόλαιο δεν θα ανατραπεί διότι τα φύλλα εργασίας είναι απλά, ρεαλιστικά και οδηγούν το μαθητή βήμα - βήμα στην ομαλή εξοικείωση του με το λογισμικό.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

<https://kyrgeo.sites.sch.gr>

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Βασική ιδέα στην οποία στηρίζεται το σενάριο είναι κατ' αρχάς ο κονστρουκτιβισμός. (constructivism) σύμφωνα με τη θεωρία του εποικοδομισμού της γνώσης του Piaget και ακολουθεί την εποικοδομιστική κατασκευαστική (constructionist) προσέγγιση της μάθησης σύμφωνα με τις αρχές που διατυπώθηκαν από τον Papert.

Οι μαθητές καλούνται να επιλύσουν μια σειρά από προβλήματα στο περιβάλλον του NXT-G. Οι μαθητές, μπορούν, εργαζόμενοι σε ομάδες, να διερευνήσουν τις δυνατότητες του περιβάλλοντος και να τις συνδυάσουν για την επίλυση των προβλημάτων. Η επίλυση των προβλημάτων που προτείνονται εξάλλου, καθιστά εμφανή τη λειτουργία και αξία των χρησιμοποιούμενων μεθόδων, εννοιών, τεχνικών κλπ..

Παράλληλα, στο σενάριο αυτό δίνεται μεγάλη βαρύτητα στη άποψη που έχει διατυπωθεί από τον Dewey, σύμφωνα με την οποία η εκπαίδευση πρέπει να στηρίζεται στις φυσικές παρορμήσεις των

παιδιών για έρευνα, κατασκευή, επικοινωνία και έκφραση και να εκμεταλλεύεται τη φυσική παρόρμηση τους, που αφορά στην κατασκευή και αλληλεπίδραση με ενεργές αναπαραστάσεις μιας και το περιβάλλον NXT-G, έχει ένα χαρακτήρα παιγνιώδη. Το στοιχείο αυτό, σε συνδυασμό με την οργάνωση της τάξης σε μικρές ομάδες, προσφέρει ένα πολύ καλό περιβάλλον για την ανάπτυξη μιας ισχυρής αλληλεπίδρασης που μπορεί να ευνοήσει τη μάθηση. Αναμένεται λοιπόν να υπάρξει ζωντανή συμμετοχή των μαθητών στις αρχικές δραστηριότητες. Επιπλέον, είναι ενδεχόμενο, πολύ γρήγορα οι μαθητές να θελήσουν να δοκιμάσουν τροποποιήσεις του περιβάλλοντος και των λοιπών στοιχείων των (μικρών αρχικά) προγραμμάτων, επιθυμώντας να τροποποιήσουν τα χαρακτηριστικά του σύμφωνα με τι επιθυμίες τους. Στο παρόν διδακτικό σενάριο λοιπόν, θα πραγματοποιηθεί πειραματισμός με τα στοιχεία του προγραμματιστικού περιβάλλοντος με ενεργητική συμμετοχή.

Εργαζόμενοι σε ομάδες εξάλλου, όπως υποστηρίζουν οι θεωρίες του Vygotsky, θα έχουν τη δυνατότητα να αλληλεπιδράσουν σε μεγάλο βαθμό με τους συμμαθητές τους να αντιμετωπίσουν κοινωνιογνωστικές συγκρούσεις και έτσι να αναγκαστούν να διατυπώσουν τις γνώμες τους, να εκφράσουν απόψεις, να επιχειρηματολογήσουν, να διαφωνήσουν και να εμπλουτίσουν έτσι και να εκλεπτύνουν τις αρχικές τους αντιλήψεις. Με την κατάλληλη διοργάνωση του μαθήματος, μπορούν ακόμη να δημιουργηθούν συνθήκες κατάλληλες για συνεργατική μάθηση υποστηριζόμενη από υπολογιστή.

Ευνοείται λοιπόν ιδιαίτερα η δημιουργία ενός τυπικού περιβάλλοντος κοινωνιο-κονστрукτιβιστικού. Πιο συγκεκριμένα, όσον αφορά τη διδακτική προσέγγιση υιοθετούμε τις βασικές ιδέες του Piaget και του Papert: «Ο διδάσκων οφείλει να δημιουργεί κατάλληλες συνθήκες για να μπορέσουν οι μαθητές να οικοδομήσουν τις γνώσεις τους. Το σενάριο είναι θεμελιωμένο στη θεωρία μάθησης του εποικοδομητισμού διότι ο μαθητής χτίζει τη γνώση του ανιχνεύοντας, διερευνώντας και αλληλεπιδρώντας οπτικά με τις εντολές της γλώσσας προγραμματισμού του NXT-G δημιουργώντας απλά προγράμματα.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟ ΜΕΤΑΒΟΛΩΝ

Δε φαίνεται να έχουμε.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ - ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Προτείνεται οι μαθητές να οργανωθούν σε ομάδες των 2-3 (το πολύ) ατόμων.

Η οργάνωση αυτή ευνοεί την αλληλεπίδραση μεταξύ των μαθητών οι οποίοι αντιμετωπίζουν την επίλυση των προτεινομένων προβλημάτων ως ένα κοινό project, ως μια κοινή προσπάθεια και όχι ως ατομικές δράσεις.

Θα πρέπει πάντως να δοθεί προσοχή στο γεγονός ότι η κατανόηση της λειτουργίας (της σημασίας δηλαδή) των εντολών του περιβάλλοντος NXT-G είναι πολύ σημαντική για την περαιτέρω πρόοδο τους.

Το μάθημα θα πραγματοποιηθεί στο σχολικό εργαστήριο πληροφορικής. Το σενάριο φαίνεται να είναι ρεαλιστικά πραγματοποιήσιμο στον προβλεπόμενο χρόνο.

14. ΕΠΕΚΤΑΣΕΙΣ / ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Ή ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Μετά την ολοκλήρωση του παρόντος σεναρίου οι μαθητές θα είναι προετοιμασμένοι για να εργαστούν με επόμενα σενάρια για τη διδασκαλία προχωρημένων προγραμματιστικών εννοιών, όπως οι μεταβλητές και οι παράλληλες διεργασίες.

Ο εκπαιδευτικός μπορεί να προετοιμάσει σχετικές επεκτάσεις των φύλλων εργασίας.

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Τα φύλλα εργασίας αφορούν στην εκμάθηση των προγραμματιστικών δομών στο περιβάλλον οπτικού προγραμματισμού NXT-G και συνοδεύονται από τα αντίστοιχες σημειώσεις του καθηγητή όπου υπάρχει η απαραίτητη ανάλυση τους.

16. ΑΞΙΟΛΟΓΗΣΗ

Οι βασικές διαδικασίες αξιολόγησης προέρχονται κυρίως μέσα από το βαθμό επίτευξης των (προγραμματιστικών) στόχων που θέτει το σενάριο. Εφόσον μάλιστα, αυτούς τους στόχους διδασκαλίας τους έχουμε εξηγήσει στους μαθητές, τους καθιστούμε εν μέρει ικανούς να αξιολογήσουν και μόνοι τους την εργασία τους.

Στο τέλος του εγχειρήματος όλοι οι μαθητές, συμπληρώνουν δύο ερωτηματολόγια ένα για να αξιολογηθεί κατά πόσο έχουν αφομοιώσει τη νέα γνώση και ένα δεύτερο σε μία προσπάθεια να καταγραφούν γραπτώς οι εντυπώσεις τους από τις δραστηριότητες με τις οποίες ασχολήθηκαν.

Επίσης, ο εκπαιδευτικός μπορεί να προτείνει τεστ αυτοαξιολόγησης (κλειστού τύπου), σταυρόλεξα, κρυπτόλεξα κλπ. Το λογισμικό Hot Potatoes (και άλλα ανάλογα, όπως και ορισμένες επιλογές του Moodle) προσφέρονται για συστηματική δημιουργία ερωτημάτων κλειστού τύπου..

17. ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ

Τα Σχέδια μαθήματος, τα φύλλα εργασίας και τα φύλλα αξιολόγησης του μαθητή και οι σημειώσεις του καθηγητή επισυνάπτονται παρακάτω :

Μαζί με τα φύλλα παρουσιάσεων και τα αντίστοιχα προγράμματα θα τα βρείτε στην ιστοσελίδα μου : <https://kyrgeo.sites.sch.gr/lego-mindstorms/>

Παρόμοιο Ψηφιακό Σενάριο μου θα βρείτε στην ιστοσελίδα του Αισώπου : <http://aesop.iep.edu.gr/node/18609>

ΓΝΩΣΤΙΚΟ ΑΝΤΙΚΕΙΜΕΝΟ:

**ΠΛΗΡΟΦΟΡΙΚΗ
ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
ΜΕ LEGO MINDSTORMS NXT**

Α) Περιγραφή Διδακτικής Παρέμβασης στην τάξη – με χρήση ΤΠΕ

Δημιουργός σεναρίου: Γεώργιος Κυριακού

Ημ/νίες εφαρμογής:

Τάξεις/Τμήματα: Γ' Γυμνασίου / Α' Λυκείου

Τίτλος διδακτικής παρέμβασης

Ρομποτική και το πακέτο Mindstorms NXT της Lego

Γνωστικό αντικείμενο και εμπλεκόμενες γνωστικές περιοχές

Πληροφορική Γ' Γυμνασίου. Διδακτική Ενότητα: 1.2 :

Ο Προγραμματισμός στην πράξη

Τάξη εφαρμογής

Γ' Γυμνασίου (Νέα πιλοτικά προγράμματα σπουδών)

Συμβατότητα και συσχέτιση με το Αναλυτικό Πρόγραμμα

Προγραμματίζω τον υπολογιστή - Οι μαθητές σχεδιάζουν και υλοποιούν προγράμματα σύμφωνα με τα νέα πιλοτικά προγράμματα σπουδών.

Υλικοτεχνική υποδομή

Εργαστήριο πληροφορικής, βιντεοπροβολέας, πίνακας μαρκαδόρου, πακέτο Lego Mindstorms NXT Education.

Το Λογισμικό

Λογισμικό παρουσιάσεων Powerpoint.

Σκοπός

Να αποκτήσει ο μαθητής σχετική ευχέρεια και εμπειρία στη δεξιότητα του δομημένου προγραμματισμού.

Διδακτικοί στόχοι:

α. Εισαγωγή στη Ρομποτική

- Να κατανοήσουν οι μαθητές την έννοια ρομπότ.
- Να αξιολογήσουν τις δυνατότητες και τη χρησιμότητας των ρομπότ.
- Να μάθουν πως λειτουργεί ένα ρομπότ.
- Να κατανοήσουν την έννοια του αλγορίθμου.
- Να έρθουν σε επαφή με τις βασικές αρχές του προγραμματισμού.

β. Γνωριμία με το Πακέτο Lego Mindstorms

- Να έρθουν σε επαφή οι μαθητές με τα διαφορετικά τεμάχια του πακέτου Lego Mindstorms.
- Να μπορούν να αναγνωρίζουν τα διαφορετικά τεμάχια και να τα ταξινομήσουν ως προς το χρώμα, την ονομασία τους και το μέγεθος τους.
- Να αναγνωρίζουν τους κινητήρες και να καθορίζουν τη κατεύθυνση κίνησης και πως μπορούν να την αντιστρέψουν.
- Να αναγνωρίζουν τους αισθητήρες και να κατανοήσουν για ποιο λόγο αυτοί χρησιμοποιούνται.
- Να αναγνωρίζουν το τούβλο (εγκέφαλο ή επεξεργαστή) NXT και να εξοικειωθούν με τη χρήση του και τις πληροφορίες που αυτό μπορεί να μας εμφανίσει.
- Να εξοικειωθούν με μερικές απλές κατασκευές.
- και τέλος να αποκτήσουν ευχέρεια στην κατασκευή ρομπότ με το πακέτο Lego Mindstorms.

γ. Κατασκευή του οχήματος – ρομπότ tankbot.

Γενικότεροι μαθησιακοί στόχοι

- Να αντιλαμβάνονται την έννοια της ρομποτικής.

Στόχοι ως προς την αξιοποίηση της τεχνολογίας

- Εξοικείωση των μαθητών με τις ρομποτικές κατασκευές.

Διδακτική μέθοδος

Εκτεταμένη διάλεξη και αντίστοιχη επίδειξη.

Διδακτικές στρατηγικές / τεχνικές

Μορφή καθοδήγησης (θέτουμε ερωτήματα και οι μαθητές επιλέγουν βήματα και δράσεις σε πλαίσιο καθοδήγησης), πειραματισμός, διερεύνηση, ανακάλυψη, επίλυση προβλήματος.

Οργάνωση διδασκαλίας

Οι μαθητές κάθονται σε διάταξη ημικυκλίου.

Ρόλος εκπαιδευτικού

Καθοδηγητικός. Καθοδηγούμε τη διαδικασία ώστε να αναδεικνύονται οι προϋπάρχουσες αντιλήψεις των μαθητών, οι στρατηγικές τους, προκαλούμε την αποτελεσματική συζήτηση και την αλληλεπίδραση στις ομάδες και στην ολομέλεια..

Εκτιμώμενη διάρκεια

2 διδακτικές ώρες

2. ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ

Το επιστημονικό περιεχόμενο

Προγραμματισμός → Ρομποτική

Προαπαιτούμενες γνώσεις : όχι

Περιγραφή δραστηριοτήτων

Αναλυτική παρουσίαση στο έγγραφο "[A2- Πορεία Διδασκαλίας](#)" (Επισυνάπτεται)

1. Αφόρμηση και παροχή κινήτρων.
2. Προβολή βίντεο από το αρχείο της NASA "[Mars Science Laboratory Curiosity Rover Animation](#)":
3. Προβολή της παρουσίασης "[1.1.1 - Ρομπότ](#)".
4. Προβολή της παρουσίασης "[1.1.2 - Ρομποτική](#)".
5. Συζήτηση γύρω από τα ρομπότ : Πως είναι ένα ρομπότ.
6. Περιγραφή των δραστηριοτήτων που θα ακολουθήσουν :
7. Κατασκευή Ρομποτικών κατασκευών χρησιμοποιώντας τεμάχια του πακέτου Lego Mindstorms και προγραμματισμός τους.
8. Προβολή της παρουσίασης "[1.2.3 - Lego Ρομπότ](#)".
9. [Επίδειξη συμπεριφορών](#) του ρομπότ με εκτέλεση έτοιμων προγραμμάτων.
10. Παρουσίαση των τεμαχίων του πακέτου Lego Mindstorms με ταυτόχρονη προβολή της παρουσίασης "[1.2.1- Πακέτο Lego Mindstrom](#)".
11. Αναλυτικότερη παρουσίαση των γραναζιών και τροχαλίων με ταυτόχρονη προβολή της παρουσίασης "[1.2.2- Πακέτο Lego Mindstrom - Γρανάζια, Τροχαλίες](#)".
12. Αναλυτικότερη παρουσίαση των κινητήρων, αισθητήρων και του μικροεπεξεργαστή NXT με ταυτόχρονη προβολή της παρουσίασης "[1.2.3- Πακέτο Lego Mindstrom - Κινητήρες, Αισθητήρες, NXT](#)".
13. Επίδειξη των λειτουργιών του μικροεπεξεργαστή NXT και πειραματισμός από τους μαθητές.

Σενάρια Ρομποτικής με το Mindstorms NXT

14. Κατασκευή από τους μαθητές του οχήματος – ρομπότ tankbot ακολουθώντας τις οδηγίες του αρχείου [tank.lfx](#) με χρήση του δωρεάν λογισμικού Lego Digital Designer της Lego (εναλλακτικά του αρχείου [tank.ldr](#) με χρήση του δωρεάν λογισμικού MLCad). Εναλλακτικά μπορεί να χρησιμοποιηθεί κάποιο άλλο όχημα.
15. Αξιολόγηση των μαθητών με τη συμπλήρωση εργασιών για το σπίτι δίνοντας τα παρακάτω Φύλλα αξιολόγησης :
- “[1.1 - Τεμάχια NXT](#)”
- “[1.2.1 - Γρανάζια](#)”
- “[1.2.2 - Γρανάζια και ταχύτητα](#)”
- “[1.3 - Αισθητήρες](#)”

Φύλλα εργασίας : όχι

Αξιολόγηση μαθητών

Αξιολόγηση της δεξιότητας χειρισμού προγραμματιστικού λογισμικού και της συμπλήρωσης του φύλλου αξιολόγησης.

Αξιολόγηση σεναρίου

Αξιολόγηση μέσα από τις ίδιες δραστηριότητες.

B) Επισυναπτόμενο υλικό:

Παρουσιάσεις :

[1.1.1 - Ρομπότ](#)

[1.1.2 - Ρομποτική](#)

[1.2.1- Πακέτο Lego Mindstrom](#)

[1.2.2- Πακέτο Lego Mindstrom - Γρανάζια, Τροχαλίες](#)

[1.2.3- Πακέτο Lego Mindstrom - Κινητήρες, Αισθητήρες, NXT](#)

Σημειώσεις του καθηγητή :

[A1- Γενική Θεώρηση και Διδακτική Μεθοδολογία](#)

[A2- Πορεία Διδασκαλίας](#)

[B1- Ρομποτική](#)

[B2- Πακέτο Lego Mindstorms NXT](#)

[Γ- Βιβλιογραφία](#)

[Φύλλο Εργασιών 0 - Επίδειξη συμπεριφορών \(Καθηγητή\)](#)

ΠΑΡΑΡΤΗΜΑ IV – [Εποικοδομισμός](#)

ΠΑΡΑΡΤΗΜΑ III - [Οδηγίες κατασκευής tankbot](#)

Φύλλα Αξιολόγησης μαθητών

[1.1 - Τεμάχια NXT](#)

[1.2.1 - Γρανάζια](#)

[1.2.2 - Γρανάζια και ταχύτητα](#)

[1.3 - Αισθητήρες](#)

Έτοιμα προγράμματα :

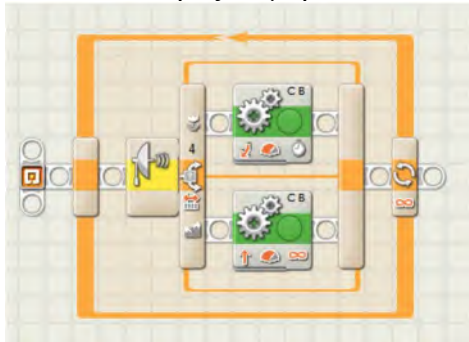
[Φάκελος 0 - Αφόρμηση](#)

Γ) Αποτελέσματα –Σχόλια – Προβληματισμοί (Αναστοχασμός)

Σημειώσεις Καθηγητή - έτοιμα προγράμματα για επίδειξη

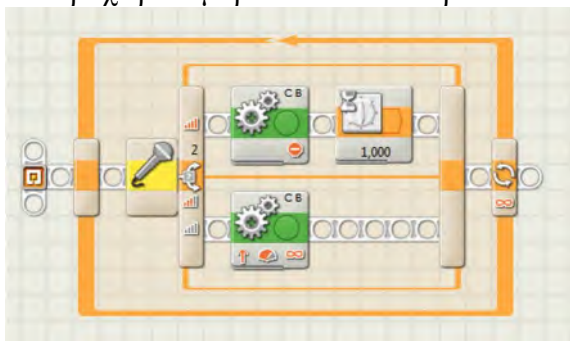
1. Ανίχνευση εμποδίου μπροστά και παράκαμψη του.

κινείται προς τα μπροστά και όταν συναντήσει εμπόδιο το παρακάμπτει.



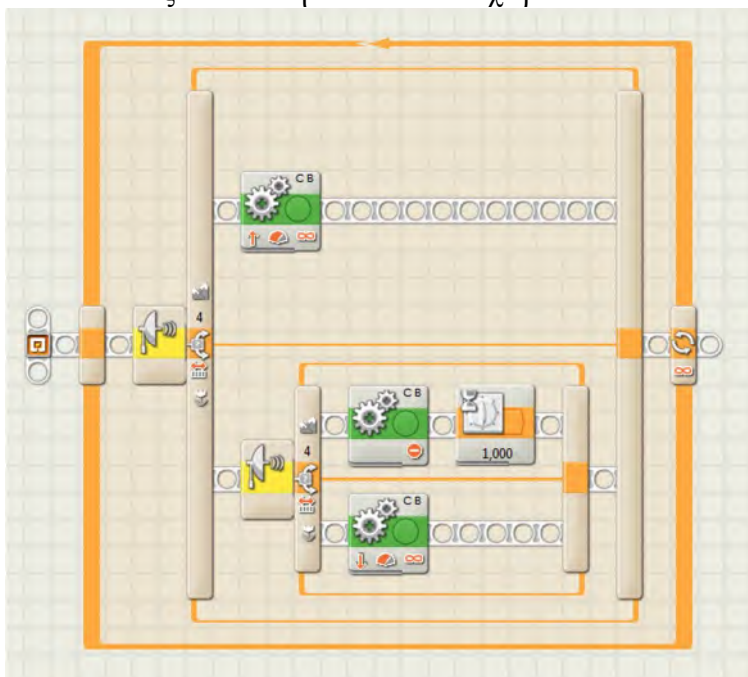
2. Προσομοίωση χελώνας.

προχωράει μπροστά όταν επικρατεί απόλυτη ησυχία, αλλιώς κάθετα ακίνητο



3. Προσομοίωση φοβισμένου σκύλου.

προχωράει μπροστά μέχρι να σε πλησιάσει οπότε και σταματάει.
Όταν πας να το πλησιάσει οπισθοχωρεί.



Φύλλο Αξιολόγησης : Τεμάχια

Όνομα _____

Ημερομηνία _____

1. Ταίριαξε τα ονόματα των τεμαχίων Lego με τις εικόνες



τροχαλία



ρόδα



άξονας



δακτύλιος



γρανάζι



δοκός

2. Γράψε το όνομα του κάθε τεμαχίου στην παρακάτω εικόνα.

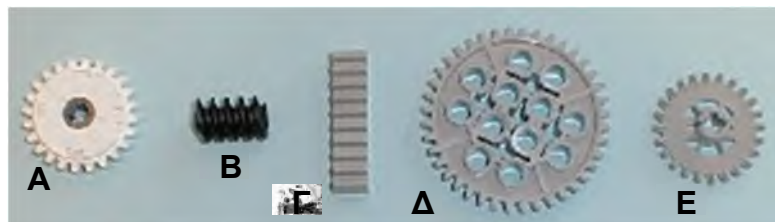


Φύλλο Αξιολόγησης : Γρανάζια

Όνομα _____

Ημερομηνία _____

1. Γράψε τα ονόματα των παρακάτω διαφορετικών γραναζιών :



- A. _____
- B. _____
- Γ. _____
- Δ. _____
- Ε. _____

2. Σε τι μετράμε το μέγεθος ενός γραναζιού ;

3. Τα γρανάζια με λοξά δόντια :

- α. Έχουνε παράλληλα δόντια μεταξύ τους.
- β. Αλλάζουν την κατεύθυνση της κίνησης κατά 90° .
- γ. Αλλάζουν την κατεύθυνση της κίνησης κατά 45° .
- δ. Αλλάζουν την κατεύθυνση της κίνησης κατά 180° .

4. Το κολχιωτό γρανάζι έχει το παρακάτω ειδικό χαρακτηριστικό :

- α. Αλλάζει την περιστροφική κίνηση σε ευθύγραμμη.
- β. Μπορεί να περιστρέψει ένα κανονικό γρανάζι αλλά είναι αδύνατο να περιστραφεί από αυτό.
- γ. Αλλάζει την κατεύθυνση της κίνησης κατά 45°

5. Το σύστημα των γραναζιών που αλλάζει την περιστροφική κίνηση σε ευθύγραμμο ονομάζεται :
- α. Κοχλιωτού γραναζιού.
 - β. Γραναζιών με λοξά δόντια.
 - γ. Κανονικών γραναζιών.
 - δ. Δοκού με δόντια.
6. Πέρα από τα γρανάζια, μια άλλη παράμετρο που επηρεάζει την ταχύτητα ενός οχήματος – ρομπότ είναι :
- α. Το μήκος των αξόνων.
 - β. Η κατεύθυνση περιστροφής του κινητήρα.
 - γ. Ο αριθμός των αισθητήρων.
 - δ. Η διάμετρος από τις ρόδες.
7. Για να αυξήσουμε την ταχύτητα σε ένα όχημα - ρομπότ μπορούμε να :
- α. Αυξήσουμε τη διάμετρο από τις ρόδες.
 - β. Μειώσουμε τη διάμετρο από τις ρόδες.
 - γ. Αυξήσουμε το πάχος από τις ρόδες.
 - δ. Χρησιμοποιήσουμε ερπύστριες αντί για ρόδες.
8. Ο μοναδικός τύπος γραναζιού που επιτρέπει τη μετάδοση της κίνησης προς τη μία μόνο κατεύθυνση είναι :
- α. Ένα κανονικό γρανάζι
 - β. Ένα γρανάζι με λοξά δόντια
 - γ. Ένα κοχλιωτό γρανάζι
 - δ. Ένα σύστημα διαφορικού.

Φύλλο Αξιολόγησης : Γρανάζια και ταχύτητα

Όνομα _____

Ημερομηνία _____

1. Τι ονομάζουμε λόγο γραναζιών.

2. Ποιο είναι το χαρακτηριστικό ενός κοχλιωτού γραναζιού.

3. Ποιο είναι το χαρακτηριστικό ενός γραναζιού με ολίσθηση ; Σε τι διαφέρει από ένα κανονικό γρανάτζι ;

4. Ποιες διαφορές υπάρχουν όταν έχουμε μετάδοση κίνησης με ιμάντες και τροχαλίες αντί για γρανάτζια ;

Φύλλο Αξιολόγησης : Αισθητήρες

Όνομα _____

Ημερομηνία _____

1. Περιέγραψε με δικά σου λόγια τι είναι ένας αισθητήρας.

2. Περιέγραψε με δικά σου λόγια τι είναι ένας αισθητήρας αφής.
Τι είδους πληροφορία μας δίνει ;

3. Σκέψου και σημείωσε για ποιο λόγο ένα όχημα - ρομπότ χρειάζεται έναν αισθητήρα αφής ;

4. Σκέψου και σημείωσε με ποιους τρόπους θα μπορούσαμε να χρησιμοποιήσουμε δύο αισθητήρες αφής σε ένα όχημα – ρομπότ ;

5. Περιέγραψε με δικά σου λόγια τι κάνει ένας αισθητήρας φωτός.
Τι είδους πληροφορία μας δίνει ;

6. Σκέψου και σημείωσε για ποιο λόγο ένα ρομπότ χρειάζεται έναν αισθητήρα φωτός ;

ΓΝΩΣΤΙΚΟ ΑΝΤΙΚΕΙΜΕΝΟ: ΠΛΗΡΟΦΟΡΙΚΗ
**ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
ΜΕ LEGO MINDSTORMS NXT**

A) Περιγραφή Διδακτικής Παρέμβασης στην τάξη – με χρήση ΤΠΕ

Δημιουργός σεναρίου: Γεώργιος Κυριακού

Ημ/νίες εφαρμογής:

Τάξεις/Τμήματα: Γ' Γυμνασίου / Α' Λυκείου

Τίτλος διδακτικής παρέμβασης

Γνωριμία με το προγραμματιστικό περιβάλλον NXT-G

Γνωστικό αντικείμενο και εμπλεκόμενες γνωστικές περιοχές

Πληροφορική Γ' Γυμνασίου. Διδακτική Ενότητα: 1.2 :

Ο Προγραμματισμός στην πράξη

Τάξη εφαρμογής

Γ' Γυμνασίου (Νέα πιλοτικά προγράμματα σπουδών)

Συμβατότητα και συσχέτιση με το Αναλυτικό Πρόγραμμα

Προγραμματίζω τον υπολογιστή - Οι μαθητές σχεδιάζουν και υλοποιούν προγράμματα σύμφωνα με τα νέα πιλοτικά προγράμματα σπουδών.

Υλικοτεχνική υποδομή

Εργαστήριο πληροφορικής, βιντεοπροβολέας, πίνακας μαρκαδόρου, πακέτο Lego Mindstorms NXT Education.

Το Λογισμικό

Περιβάλλον προγραμματισμού Lego NXT-G, λογισμικό παρουσιάσεων Powerpoint.

Σκοπός

Να αποκτήσει ο μαθητής σχετική ευχέρεια και εμπειρία στη δεξιότητα του δομημένου προγραμματισμού.

Διδακτικοί στόχοι:

Εξοικείωση με το NXT-G

- Να έρθουν σε επαφή οι μαθητές με το πρόγραμμα NXT-G.
- Να γνωρίσουν τη διαδικασία προγραμματισμού και μεταφόρτωσης (κατέβασμα) του προγράμματος στο NXT.
Αναλυτικότερα:
 - ο Πως ανοίγουμε το προγραμματιστικό περιβάλλον NXT-G και ξεκινάμε τη συγγραφή ενός νέου προγράμματος.
 - ο Πως δημιουργούμε ένα πρόγραμμα.
 - ο Πως το κατεβάζουμε και το πώς το τρέχουμε.
 - ο Πως κάνουμε αλλαγές και πως τροποποιούμε τις παραμέτρους.
 - ο Πως αποθηκεύουμε το πρόγραμμά μας.
- Να γνωρίσουν τις βασικές αρχές προγραμματισμού
- Να κατανοήσουν τον τρόπο με τον οποίο μπορείς να ελέγξεις μία ρομποτική κατασκευή.

Γενικότεροι μαθησιακοί στόχοι

- να γνωρίζουν τη διαδικασία δημιουργίας και εκτέλεσης προγραμμάτων.

Σενάρια Ρομποτικής με το Mindstorms NXT

Γενικότεροι εκπαιδευτικοί στόχοι

- συνεργασία των μαθητών για εξαγωγή αποτελεσμάτων και συμπερασμάτων

Στόχοι ως προς την αξιοποίηση της τεχνολογίας

- να γνωρίζουν τα περιβάλλοντα οπτικού προγραμματισμού.

Διδακτική μέθοδος

Αρχικά με διάλεξη περιορισμένου χρόνου και αντίστοιχη επίδειξη
Στη συνέχεια οι μαθητές εργάζονται σε ομάδες (κοινωνιο-επικοινωνιακή μάθηση),
έμμεση διδασκαλία – μαθητοκεντρική.

Διδακτικές στρατηγικές / τεχνικές

Μορφή καθοδήγησης (θέτουμε ερωτήματα και οι μαθητές επιλέγουν βήματα και δράσεις σε πλαίσιο καθοδήγησης), πειραματισμός, διερεύνηση, ανακάλυψη, επίλυση προβλήματος.

Οργάνωση διδασκαλίας

Ομάδες των τριών ατόμων. Αναθέτουμε σε ένα μέλος της ομάδας το ρόλο του εκπροσώπου, σε ένα άλλο το ρόλο του γραμματέα ο οποίος θα κρατάει τις σημειώσεις και στο τελευταίο μέλος το ρόλο του προγραμματιστή στο NXT-G. Κάθε ομάδα έχει το δικό της υπολογιστή.

Ρόλος εκπαιδευτικού

Καθοδηγητικός. Καθοδηγούμε τη διαδικασία ώστε να αναδεικνύονται οι προϋπάρχουσες αντιλήψεις των μαθητών, οι στρατηγικές τους, προκαλούμε την αποτελεσματική συζήτηση και την αλληλεπίδραση στις ομάδες και στην ολομέλεια, παροτρύνουμε τους μαθητές που συναντούν δυσκολίες.

Εκτιμώμενη διάρκεια

2 διδακτικές ώρες

2. ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ

Το επιστημονικό περιεχόμενο

Προγραμματισμός → Περιβάλλον προγραμματισμού NXT-G.

Προαπαιτούμενες γνώσεις

1. Εισαγωγή στη Ρομποτική (βλέπε σημειώσεις: [B1- Ρομποτική](#))
2. Γνωριμία με το Πακέτο Lego Mindstorms (βλέπε σημειώσεις: [B2- Πακέτο Lego Mindstorms NXT](#))

Περιγραφή δραστηριοτήτων

Αναλυτική παρουσίαση στο έγγραφο “[A2- Πορεία Διδασκαλίας](#)” (Επισυνάπτεται)

1. Παρουσίαση του προγραμματιστικού περιβάλλοντος NXT-G με ταυτόχρονη προβολή της παρουσίασης “[2.1 Προγραμματιστικό περιβάλλον NXT-G](#)”.
2. Πρακτική εξάσκηση και εξοικείωση των μαθητών με το προγραμματιστικό περιβάλλον NXT-G ακολουθώντας το Φύλλο Εργασιών “[Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G](#)”.
3. Αξιολόγηση των μαθητών με τη συμπλήρωση εργασιών για το σπίτι με το
4. Φύλλο Αξιολόγησης “[2.1 - Προγραμματιστικό περιβάλλον NXT-G](#)”.

Φύλλα εργασίας

[Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G](#) (Επισυνάπτεται)

Αξιολόγηση μαθητών

Αξιολόγηση της ικανότητας χρήσης του προγραμματιστικού περιβάλλοντος μέσα από τις δραστηριότητες οι οποίες θα υλοποιηθούν και της συμπλήρωσης του φύλλου εργασίας και αξιολόγησης.

Σενάρια Ρομποτικής με το Mindstorms NXT

Αξιολόγηση σεναρίου

Αξιολόγηση μέσα από τις ίδιες δραστηριότητες και της ικανότητας συνεργασίας των μαθητών, όπως και από τη συμπλήρωση του φύλλου αξιολόγησης.

B) Επισυναπτόμενο υλικό:

Παρουσίαση :

[Προγραμματιστικό περιβάλλον NXT-G](#)

Φύλλα εργασιών :

[Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G](#)

Σημειώσεις του καθηγητή :

[A2- Πορεία Διδασκαλίας](#)

[B3- Το προγραμματιστικό περιβάλλον NXT-G](#)

[B4- Οι εντολές του NXT-G](#)

[Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G \(Καθηγητή\)](#)

[ΠΑΡΑΡΤΗΜΑ Ι - Σετ εντολών του NXT-G](#)

Φύλλα Αξιολόγησης μαθητών :

[2.1 - Προγραμματιστικό περιβάλλον NXT-G](#)

Έτοιμα προγράμματα :

[Φάκελος 1 - Εξοικείωση](#)

Γ) Αποτελέσματα –Σχόλια – Προβληματισμοί (Αναστοχασμός)

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασίας 1^ο

Γνωριμία με το προγραμματιστικό περιβάλλον NXT-G

Όνομα _____

Ημερομηνία _____

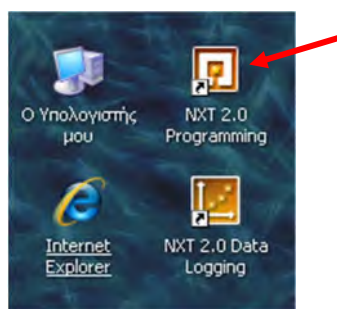
Εξοικείωση με το NXT-G

Στις παρακάτω ασκήσεις θα χρησιμοποιήσουμε το όχημα - ρομπότ NXTbot, που έχουμε ήδη κατασκευάσει.

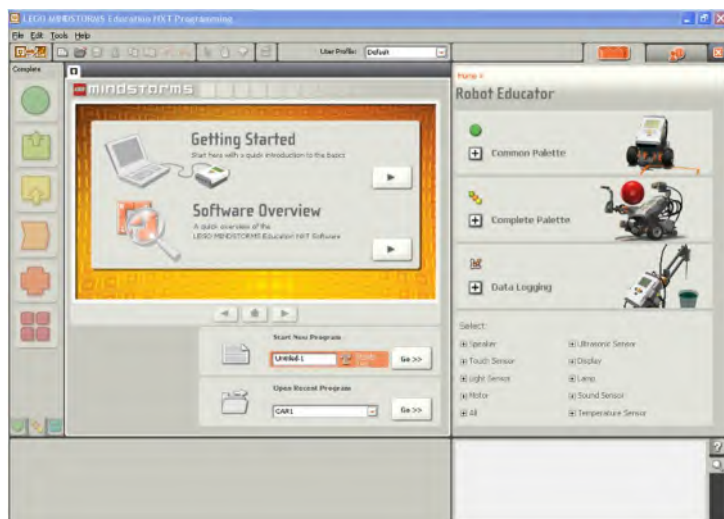
Το όχημα αυτό διαθέτει δύο κινητήρες :

Ο αριστερός κινητήρας κινεί την αριστερή ρόδα και είναι συνδεδεμένος στη θύρα εξόδου Β του NXT, ενώ ο δεξιός κινητήρας κινεί τη δεξιά ρόδα και είναι συνδεδεμένος στη θύρα εξόδου C.

- Ανοίξτε την εφαρμογή NXT-G κάνοντας διπλό κλικ πάνω στη συντόμευση του Lego Mindstorms NXT



Θα ξεκινήσει η εφαρμογή και θα εμφανιστεί το παρακάτω παράθυρο :

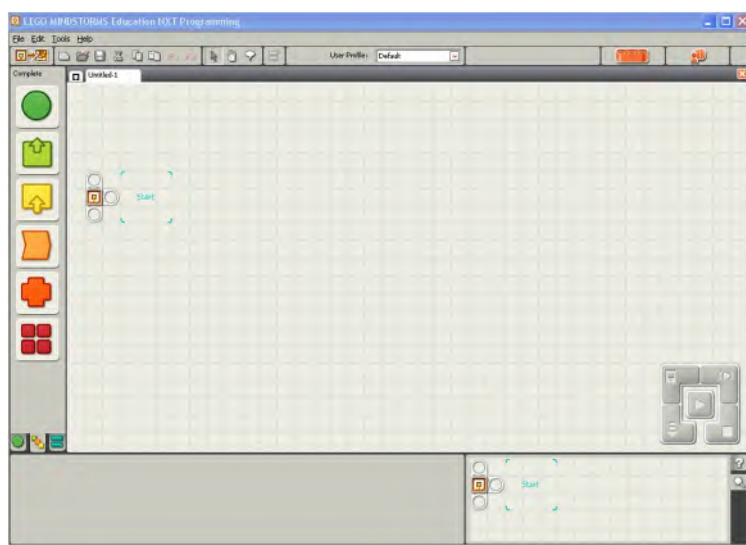


Σενάρια Ρομποτικής με το Mindstorms NXT

- Πληκτρολογήστε ένα όνομα για τη νέα εργασία σας στο πεδίο κειμένου για την εκκίνηση ενός νέου προγράμματος *Start New Program* και κάντε κλικ στο κουμπί *Go>>*.



Για να κατασκευάσουμε ένα νέο πρόγραμμα πιάνουμε με το ποντίκι ένα εικονίδιο από την παλέτα εντολών στη δεξιά πλευρά της εφαρμογής και το μετακινούμε στην περιοχή εργασίας (ή παράθυρο προγραμματισμού), όπου και το αφήνουμε.



Για τις δραστηριότητες που ακολουθούν θα χρησιμοποιήσουμε την πλήρη παλέτα εντολών.

- Από την περιοχή με τις παλέτες εντολών στο αριστερό μέρος της εφαρμογής κάντε κλικ στο μεσαίο από τα τρία καρτελάκια στο κάτω μέρος της.

Επίσης θα κλείσουμε το παράθυρο του Robot Educator στο δεξιό μέρος της εφαρμογής για να εκμεταλλευτούμε ολόκληρο το διαθέσιμο χώρο εργασίας.

- Κλείστε το παράθυρο του Robot Educator.

Δραστηριότητα 1^η : κίνηση μπροστά για 2 περιστροφές (Ένα απλό πρόγραμμα)

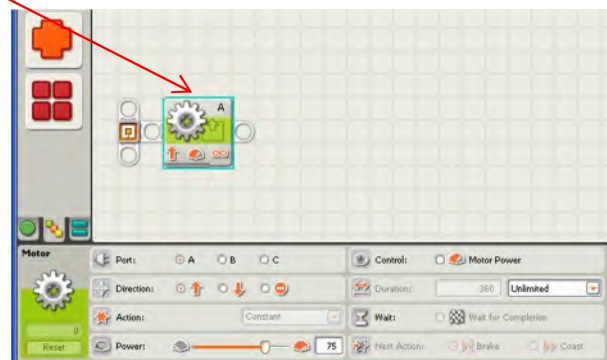
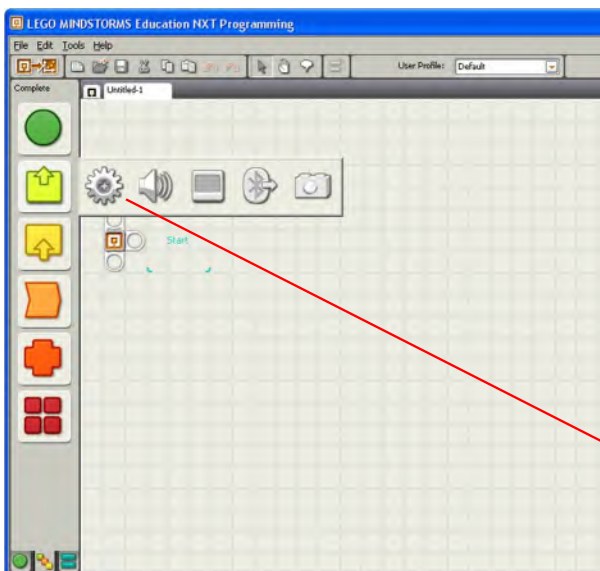
Στην άσκηση αυτή θα χρησιμοποιήσουμε το ρομπότ-όχημα nxtbot που ήδη έχουμε κατασκευάσει.



Προσπαθήστε να καταλάβετε τι αποτέλεσμα θα έχει το παρακάτω πρόγραμμα:

Ας ξεκινήσουμε δημιουργώντας ένα απλό πρόγραμμα. Αυτό το πρόγραμμα θα μετακινήσει το όχημα μας προς τα μπροστά για 2 περιστροφές των κινητήρων.

- Από το υπομενού για ενέργειες Action (έχει ως σχήμα ένα τετράγωνο με ένα βελάκι προς τα επάνω και πράσινο ανοιχτό χρώμα) επιλέξτε το εικονίδιο για την εντολή κίνησης κινητήρα, το οποίο έχει ως σχήμα ένα γρανάζι. Για να τοποθετήσετε το εικονίδιο αυτό μέσα στο πρόγραμμα μας κάντε κλικ πάνω του, κρατήστε πατημένο το πλήκτρο του ποντικιού, σύρετε το εικονίδιο στο χώρο εργασίας της εφαρμογής μας και τοποθετήστε το στο σημείο εκκίνησης (το οποίο εμφανίζεται με τη σήμανση ενός μπλε τετραγώνου εντός του οποίου περιέχεται η λέξη start) δίπλα στο σύμβολο της εκκίνησης προγράμματος.



Σενάρια Ρομποτικής με το Mindstorms NXT

- Σε αυτό το σημείο ελέγξτε αν η εντολή κούμπωσε σωστά πάνω στη λευκή συνδετική δοκό. Στην αντίθετη περίπτωση η εντολή θα εμφανίζεται αχνά. Αν δεν έχει κουμπώσει σωστά, μετακινήστε την εντολή οπουδήποτε αλλού στην περιοχή εργασίας και τοποθετείστε την ξανά πάνω στη συνδετική δοκό.

Μόλις το εικονίδιο της εντολής τοποθετηθεί στην περιοχή εργασίας, στο κάτω δεξιό μέρος της εφαρμογής μας, εμφανίζεται ο πίνακας ρυθμίσεων της εντολής. Από εδώ μπορούμε να τροποποιήσουμε την κατεύθυνση, την ταχύτητα και τη διάρκεια της κίνησης για τον κινητήρα του οχήματος μας.

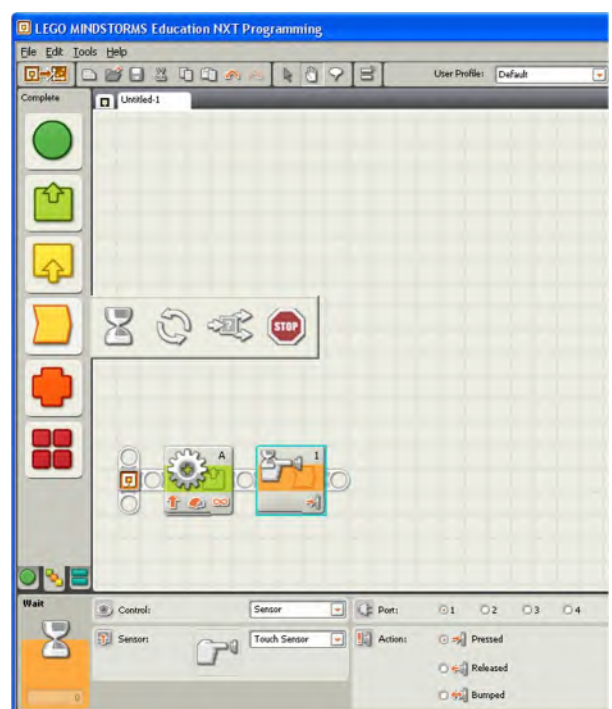


Στο σημείο αυτό μπορείτε να αλλάξετε τις παραμέτρους του πίνακα ρυθμίσεων και να πειραματιστείτε με τους διαφορετικούς τρόπους που μπορούμε να επιβάλουμε στον κινητήρα να κινηθεί.

- Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως Θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη θύρα C αντί για την A. Μην πειράξετε καμία άλλη ρύθμιση.

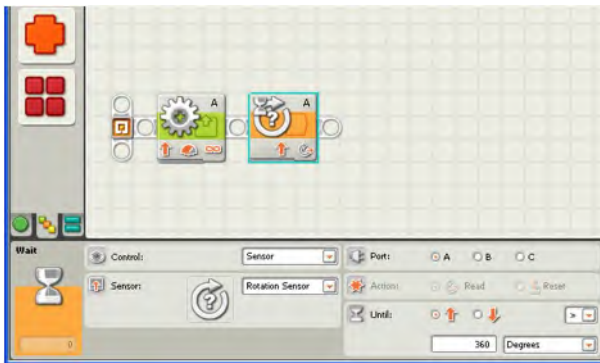
Με την εντολή αυτή ο κινητήρας μας θα ξεκινήσει να λειτουργεί και θα σταματήσει μόνο όταν το πρόγραμμα μας φτάσει στο τέλος του. Εμείς όμως θέλουμε ο κινητήρας να κινηθεί για ένα συγκεκριμένο χρονικό διάστημα. Για να το πετύχουμε αυτό πρέπει να τοποθετήσουμε μία εντολή αναμονής.

- Ανοίξτε την υποπαλέτα ροής **flow** (έχει ως σχήμα ένα τσακισμένο τετράγωνο και πορτοκαλί χρώμα) και επιλέξτε το εικονίδιο για την εντολή αναμονής, το οποίο έχει ως σχήμα μία κλεψύδρα. Σύρετε το στην περιοχή εργασίας και τοποθετείστε το στο σημείο αμέσως μετά την εντολή κίνησης κινητήρα, ακριβώς στα δεξιά της.



Οι κινητήρες που συνοδεύουν το πακέτο Lego Mindstorms NXT διαθέτουν έναν ενσωματωμένο αισθητήρα περιστροφής, ο οποίος μας προμηθεύει με πληροφορίες για την περιστροφή που εκτελεί κάθε στιγμή ένας κινητήρας. Μπορούμε να αξιοποιήσουμε τις πληροφορίες αυτές στην περίπτωση που θέλουμε να ελέγξουμε τον κινητήρα, ώστε να περιστραφεί όσο ακριβώς θέλουμε.

- Στον πίνακα ρυθμίσεων αλλάζτε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα περιστροφής "Rotation Sensor" από αισθητήρα αφής "Touch Sensor".



Αφού ο κινητήρας μας είναι συνδεδεμένος στη θύρα C την πληροφορία για την περιστροφή του, πρέπει να την πάρουμε από τον αισθητήρα περιστροφής του συγκεκριμένου κινητήρα.

- Στον πίνακα ρυθμίσεων επιλέξτε ως θύρα (Port) που θα παρακολουθείται από τον αισθητήρα περιστροφής, τη θύρα C αντί για την A. Αλλάξτε επίσης τον αριθμό των μοιρών που θέλουμε να περιστραφεί ο κινητήρας μας σε 720 (2 περιστροφές). Μην πειράξετε καμία άλλη ρύθμιση.



Τώρα μπορούμε αν θέλουμε, να γράψουμε και μερικές σημειώσεις πάνω από τις εντολές μας για να θυμόμαστε τι ακριβώς κάνουν. Οι σημειώσεις μας βοηθούν να θυμηθούμε τι ακριβώς κάνει το πρόγραμμά μας όταν θέλουμε να το χρησιμοποιήσουμε μετά από την πάροδο κάποιου χρονικού διαστήματος.

- Κάντε διπλό κλικ ακριβώς πάνω από την εντολή αναμονής για περιστροφή και πληκτρολογήστε την πρόταση <<περιστροφή 720 μοίρες>>.

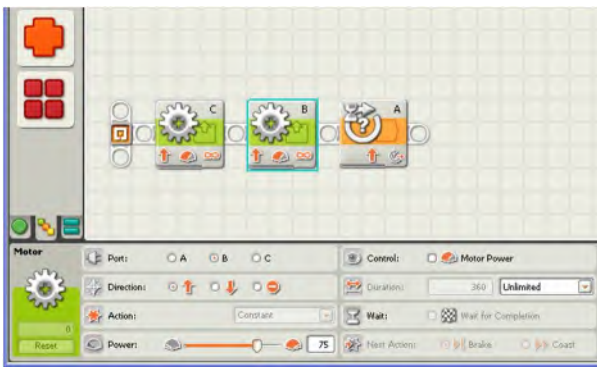
Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το όχημά μας θα εκτελέσει μία στροφή προς τα δεξιά για 2 περιστροφές.

Για να κινηθεί το όχημα μας σε ευθεία γραμμή μπροστά θα πρέπει να κινήσουμε και το δεύτερο κινητήρα μας, ο οποίος είναι συνδεδεμένος στη θύρα B, με τον ίδιο τρόπο όπως και τον πρώτο.

- Από το υπομενού για ενέργειες **Action** επιλέξτε το εικονίδιο για την εντολή κίνησης κινητήρα και τοποθετήστε το αμέσως μετά την πρώτη εντολή για τον κινητήρα C και πριν την εντολή αναμονής για περιστροφή. Παρατηρήστε ότι η εντολή αναμονής αυτόματα απομακρύνεται προς τα δεξιά για να δημιουργηθεί ο απαραίτητος χώρος.



Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη θύρα B αντί για την A.

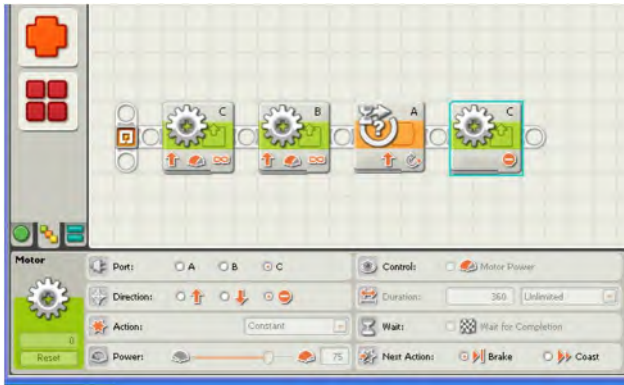


Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το όχημα μας θα κινηθεί ευθεία προς τα μπροστά για 2 τουλάχιστον περιστροφές.

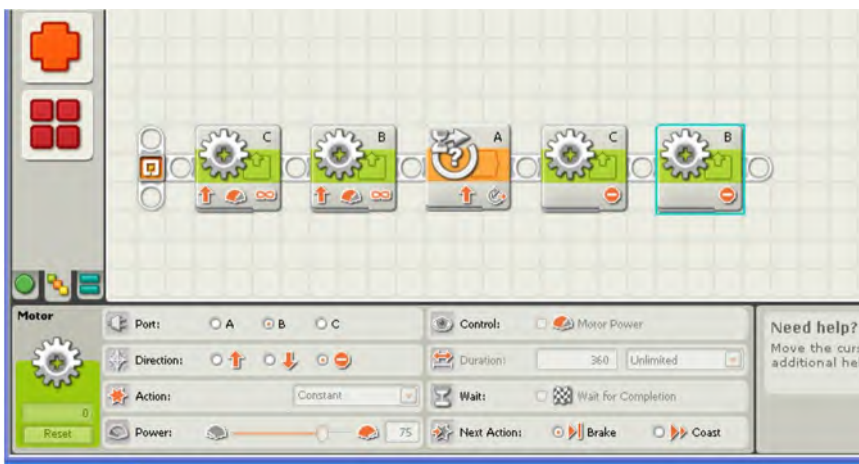
Αν θέλουμε το όχημα μας να κινηθεί για 720° ακριβώς, θα πρέπει να πούμε στο πρόγραμμα αμέσως μετά εντολή της αναμονής για 2 περιστροφές, να φρενάρει τους κινητήρες.

Θυμηθείτε ότι τα ρομπότ εκτελούν αυστηρά τις οδηγίες που τους δίνουμε και όχι αυτό που εμείς επιθυμούμε και θεωρούμε ότι θα συμβεί.

- Από την υποπαλέτα για ενέργειες **Action** επιλέξτε την εντολή κίνησης κινητήρα και σύρετε την στην περιοχή εργασίας, αμέσως μετά την τελευταία εντολή του προγράμματος. Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη θύρα C αντί για την A και ως κατεύθυνση κίνησης (Direction) επιλέξτε το σήμα του στοπ.



- Επαναλάβετε την ίδια διαδικασία ακόμη μία φορά. Στον πίνακα ρυθμίσεων αυτή τη φορά επιλέξτε, ως θύρα τη Β.



Το πρόγραμμα σας πρέπει τώρα να είναι ίδιο με το παραπάνω.

Αποθήκευση του προγράμματος

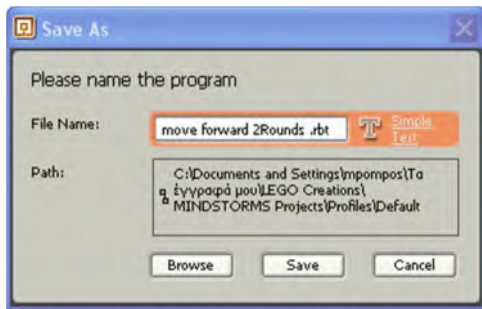
Δεν πρέπει να ξεχνάτε να αποθηκεύουμε την εργασία που κάνουμε, ανά τακτά χρονικά διαστήματα και πάντοτε πριν να στείλουμε ένα πρόγραμμα στο NXT για να το εκτελέσει.

- Για να αποθηκεύσετε το πρόγραμμα που δημιουργήσατε επιλέξτε το *Save As...* από το μενού *File* του παραθύρου σχεδίασης.

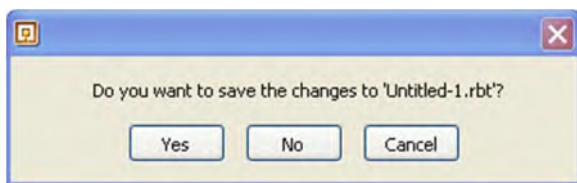
Στο παράθυρο που εμφανίζεται πληκτρολογήστε την πρόταση **κίνηση μπροστά για 2 περιστροφές** ως όνομα για την εργασία μας, καθορίστε το φάκελο **nxt programs** της επιφάνειας εργασίας ως θέση αποθήκευσης και πατήστε το κουμπί *Αποθήκευση*.

Στην καρτέλα για την αποθήκευση προτείνεται από την εφαρμογή η αποθήκευση των προγραμμάτων μας στο φάκελο που έχει δημιουργηθεί για το λόγο αυτό, από την εφαρμογή κατά την εγκατάσταση της. Για να μετακινηθείτε λοιπόν στο φάκελο **nxt programs** πατήστε το κουμπί *Browse*.

Σενάρια Ρομποτικής με το Mindstorms NXT



Προσοχή : Όταν κλείσουμε το παράθυρο του προγραμματιστικού περιβάλλοντος ενώ προηγουμένως έχουμε προχωρήσει σε κάποιες αλλαγές στην εργασία μας, εμφανίζεται το παρακάτω παράθυρο το οποίο μας προτρέπει να σώσουμε τις αλλαγές που έχουμε κάνει, στο πρόγραμμα που επεξεργαζόμαστε.



Κατέβασμα του προγράμματος στο NXT και εκτέλεση του

Για να κατεβάσουμε ένα πρόγραμμα στο όχημα μας πρέπει πρώτα να σιγουρευτούμε ότι το τούβλο NXT είναι σε λειτουργία και συνδεδεμένο με τον Η/Υ μας.

- Συνδέστε το τούβλο NXT με τον προσωπικό σας υπολογιστή χρησιμοποιώντας το καλώδιο USB. Εναλλακτικά μπορούμε να πραγματοποιήσουμε μία ασύρματη σύνδεση Bluetooth ανάμεσα στις δύο συσκευές.

Πατήστε το πορτοκαλί πλήκτρο **enter** στο NXT για να το θέσετε σε λειτουργία.



- Πατήστε το κουμπί Download στο πεδίο ελέγχου, που βρίσκεται στην κάτω δεξιά γωνία της περιοχής εργασίας, για να κατεβάσετε το πρόγραμμα που δημιουργήσατε στο NXT.

Το πρόγραμμα θα αρχίσει να μεταφέρεται και όταν το πρόγραμμά σας φορτωθεί επιτυχώς στο NXT, αυτό θα σφυρίξει.



Όταν κατεβάσετε ένα πρόγραμμα σας στο NXT, αυτό αποθηκεύεται ως αρχείο στη μνήμη του με το όνομα που του έχουμε δώσει. Για να εκτελέσουμε το πρόγραμμα αυτό στο ρομπότ-όχημα μας, πρέπει να πάμε στο φάκελο *Software Files* και να βρούμε το συγκεκριμένο αρχείο.



- Από το κυρίως μενού πατήστε το πορτοκαλί πλήκτρο για να μεταβείτε στο μενού *My Files* και ακόμη μία φορά για να μπούμε στο μενού *Software Files*. Βρείτε το πρόγραμμα σας - αν αυτό δεν εμφανίζεται ήδη στην οθόνη - με τη βοήθεια των πλήκτρων αριστερό και δεξιό βελάκι.
- Όταν το πρόγραμμα σας εμφανιστεί στην οθόνη του NXT πατήστε μία φορά το πορτοκαλί πλήκτρο *enter* για να το επιλέξετε, και ακόμη μία φορά για να το τρέξετε.

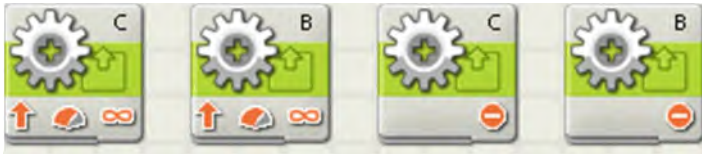
Προσοχή : πριν τρέξετε το πρόγραμμα σας, αν είχατε συνδέσει το NXT στον προσωπικό σας υπολογιστή με το καλώδιο USB, μη ξεχάσετε να αποσυνδέσετε την άκρη του καλωδίου αυτού από το NXT του οχήματος σας.

- Τρέξτε το πρόγραμμα από το NXT πατώντας δύο φορές το πορτοκαλί πλήκτρο *enter*.

Πειραματιστείτε :

Με το δεδομένο ότι όταν ολοκληρωθεί η εκτέλεση ενός προγράμματος, το τουβλάκι NXT διακόπτει αυτόματα την τροφοδοσία ρεύματος και στις τρεις εξόδους του (όπου και έχουμε συνδέσει τους κινητήρες) αν αφαιρέσουμε την εντολή αναμονής τι πιστεύετε ότι θα συμβεί;

Θα κινηθεί καθόλου το όχημα μας, και αν ναι για πόσο διάστημα;



Δοκιμάστε το στην πράξη. Τι ακριβώς συνέβη;
Προσπαθήστε να το εξηγήσετε. (Τεκμηριώστε την άποψη σας)

Συγχαρητήρια ☺
Ολοκληρώσατε την 1^η δραστηριότητα !!!

Δραστηριότητα 2^η : κίνηση προς τα πίσω για 2 περιστροφές (τροποποίηση του προγράμματος)

Για να ξεκινήσουμε τη συγγραφή ενός νέου προγράμματος υπάρχουν όπως είναι γνωστό δύο επιλογές :

1. Να μετονομάσουμε και μετά να επεξεργαστούμε ένα υπάρχον πρόγραμμα.
2. Να ξεκινήσουμε με ένα άδειο φύλλο και να κάνουμε ένα νέο πρόγραμμα από την αρχή.

Θα τροποποιήσουμε το προηγούμενο πρόγραμμα ώστε το όχημα μας να κινηθεί προς τα πίσω για 2 περιστροφές.

Άνοιγμα του προγράμματος

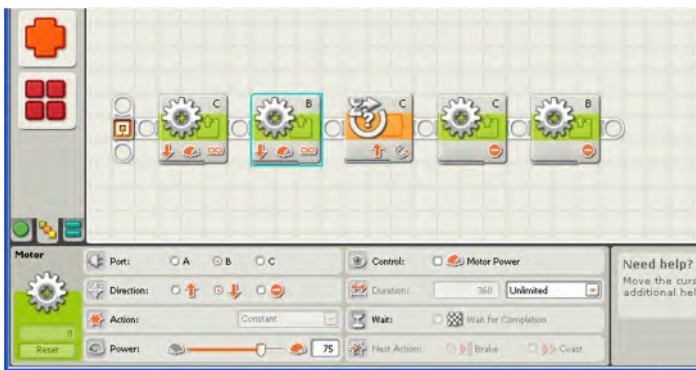
• Για να ανοίξετε ένα υπάρχον πρόγραμμα που έχετε δημιουργήσει, επιλέξτε το *Open...* από το μενού *File* της εφαρμογής.

Στο παράθυρο που εμφανίζεται καθορίστε το φάκελο *nxt programs*, επιλέξτε το αρχείο *κίνηση μπροστά για 2 περιστροφές* και πατήστε το κουμπί *ok*.

Τροποποίηση του προγράμματος

Για να κινηθεί το όχημα-ρομπότ μας προς τα πίσω θα πρέπει να αλλάξουμε τη φορά περιστροφής στις εντολές για την κίνηση των δύο κινητήρων του.

• Επιλέξτε την εντολή του κινητήρα *C* και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (*Direction*) επιλέγοντας το βελάκι που δείχνει προς τα κάτω. Κάντε το ίδιο και με την εντολή του κινητήρα *B*.

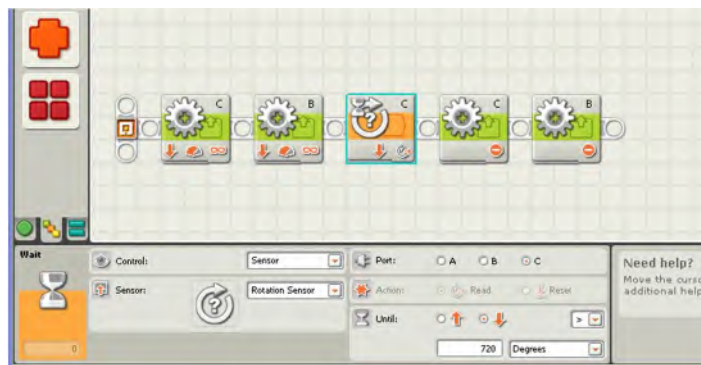


Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το όχημα μας θα κινηθεί προς τα πίσω αλλά δε θα σταματήσει να κινείται μετά από τη συμπλήρωση 2 περιστροφών. Αυτό θα συμβεί επειδή η εντολή της αναμονής για περιστροφή, έτσι όπως τη ρυθμίσαμε προηγουμένως, θα περιμένει μέχρι το όχημα μας να ολοκληρώσει δύο περιστροφές προς τα μπροστά, πράγμα που δεν πρόκειται να συμβεί ποτέ αφού τώρα έχουμε βάλει το όχημα μας να κινείται προς τα πίσω. Άρα για να σταματήσει το όχημα μας μετά από δύο περιστροφές θα

Σενάρια Ρομποτικής με το Mindstorms NXT

πρέπει να αλλάξουμε και τη φορά κίνησης που θα παρακολουθείται από την εντολή αναμονής για περιστροφή.

- Επιλέξτε την εντολή αναμονής για περιστροφή του κινητήρα C και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (*Until*) επιλέγοντας το βελάκι που δείχνει προς τα κάτω.



- Αποθηκεύστε το πρόγραμμα σας ως **κίνηση πίσω για 2 περιστροφές**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Συγχαρητήρια ☺
Ολοκληρώσατε την 2^η δραστηριότητα !!!

Δραστηριότητα 3^η : κίνηση μπροστά για 2 περιστροφές και επιστροφή (επέκταση του προγράμματος)

Θα τροποποιήσουμε το προηγούμενο πρόγραμμα ώστε το όχημα μας να κινηθεί αρχικά προς τα μπροστά για 2 περιστροφές και στη συνέχεια να κινηθεί προς τα πίσω για 2 περιστροφές, ώστε να επιστρέψει στο σημείο από το οποίο ξεκίνησε.

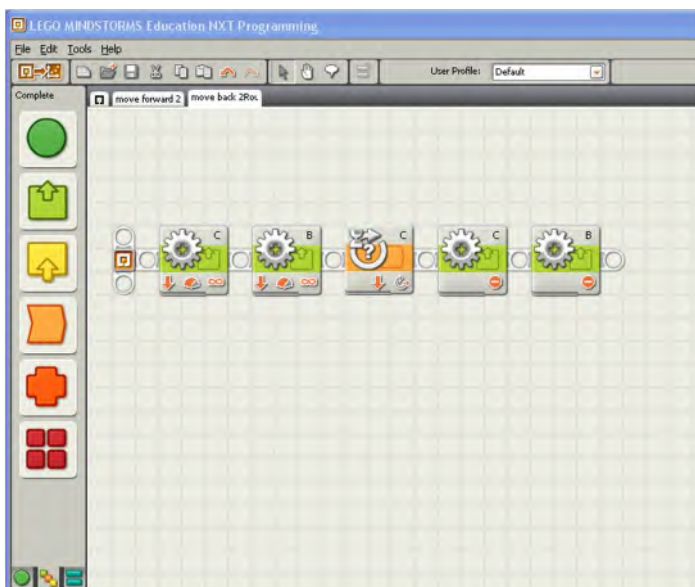
Το πρόγραμμα αυτό αποτελείται από δύο μέρη. Το πρώτο μέρος (κίνηση του οχήματος-ρομπότ προς τα εμπρός για 2 περιστροφές) το έχουμε αναπτύξει ήδη στην 1η άσκηση. Το δεύτερο μέρος (κίνηση του οχήματος-ρομπότ προς τα πίσω για 2 περιστροφές) το έχουμε αναπτύξει στην προηγούμενη εργασία μας. Αρκεί λοιπόν να ανοίξουμε το πρόγραμμα που δημιουργήσαμε στην 1η άσκηση και να το συμπληρώσουμε με το πρόγραμμα που είχαμε δημιουργήσει προηγουμένως στην 2η άσκηση.

Άνοιγμα του προγράμματος

- Ανοίξτε το πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** που δημιουργήσατε στην 1^η άσκηση.

Ανοίξτε επίσης και το πρόγραμμα **κίνηση πίσω για 2 περιστροφές** που δημιουργήσατε προηγουμένως.

Βλέπουμε ότι το λογισμικό NXT-G ανοίγει μία καινούργια καρτέλα για κάθε πρόγραμμα που επεξεργαζόμαστε. Με αυτόν τον τρόπο μπορούμε να έχουμε ανοιχτά πολλαπλά προγράμματα την ίδια στιγμή.



Τροποποίηση του προγράμματος

Για να δημιουργήσουμε το νέο μας πρόγραμμα θα επικολλήσουμε τις εντολές του προγράμματος **κίνηση πίσω για 2 περιστροφές** στο τέλος του προγράμματος **κίνηση μπροστά για 2 περιστροφές**.

- Πατήστε και κρατήστε πατημένο το αριστερό πλήκτρο του ποντικιού σε ένα σημείο της περιοχής εργασίας, πάνω και αριστερά από το πρώτο εικονίδιο του προγράμματος **κίνηση πίσω για 2 περιστροφές**. Τώρα χωρίς να αφήσετε το πλήκτρο του ποντικιού σύρετε το ποντίκι μέχρι ένα σημείο, κάτω και δεξιά από το τελευταίο εικονίδιο του προγράμματος, ώστε να σχηματιστεί ένα ορθογώνιο παραλληλόγραμμο το οποίο να περιέχει όλα τα εικονίδια του προγράμματος. Στο ορθογώνιο παραλληλόγραμμο πρέπει να προσέξουμε ώστε να μη συμπεριλάβουμε και το σημείο εκκίνησης του προγράμματος.

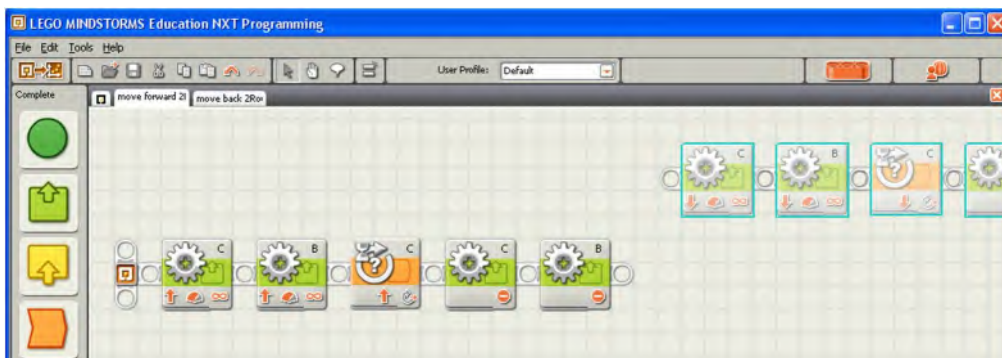


Στο σημείο αυτό όλα τα εικονίδια του προγράμματος πρέπει να περιβάλλονται από ένα μπλε περιθώριο.



- Από το μενού *edit* του λογισμικού NXT-G κάντε κλικ στην επιλογή *copy* και στη συνέχεια μεταβείτε στο πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** κάνοντας κλικ στην αντίστοιχη καρτέλα.

Στην περιοχή εργασίας της καρτέλα **κίνηση μπροστά για 2 περιστροφές** κάντε κλικ σε ένα σημείο δεξιά της τελευταίας εντολής του προγράμματος και από το μενού *edit* του λογισμικού NXT-G κάντε κλικ στην επιλογή *paste*.

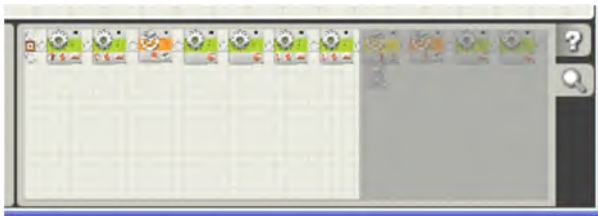


Βλέπουμε μπροστά μας το αντίγραφο που έχει δημιουργηθεί, το οποίο και μεταφέρουμε αμέσως μετά την τελευταία εντολή.

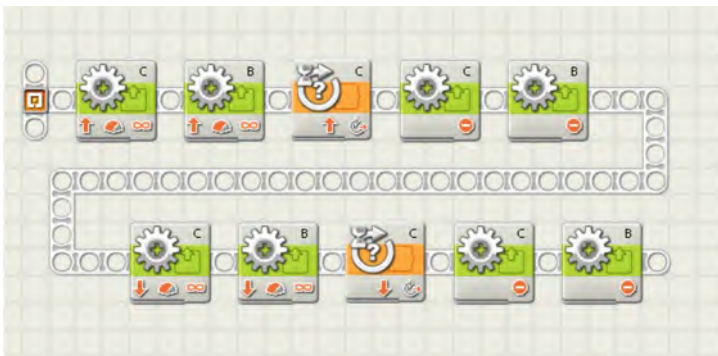


- Για να μετακινήσετε τη νέα ομάδα εντολών που μόλις αντιγράψατε σύρετε το πρώτο εικονίδιο της ομάδας αυτής και τοποθετείστε το στο τέλος της συνδετικής δοκού.

Αν το πρόγραμμα μας δε χωράει ολόκληρο στην ενεργή επιφάνεια της περιοχής εργασίας (αυτή που μπορούμε να δείτε στην οθόνη σας), μπορούμε να χρησιμοποιήσουμε το εργαλείο ολίσθησης (*pan tool*) της μπάρας εργαλείων για να μεταφερθούμε σε διάφορα σημεία του προγράμματος μας. Ένας πιο βολικός τρόπος είναι να μετακινούμε την ενεργή επιφάνεια στο χάρτη του προγράμματος, στο κάτω δεξιό μέρος της εφαρμογής μας.



Εναλλακτικά για να είμαστε σε θέση να βλέπουμε ολόκληρο το πρόγραμμα στην οθόνη μπορείτε να κατεβάσετε μερικές εντολές σε άλλη σειρά.



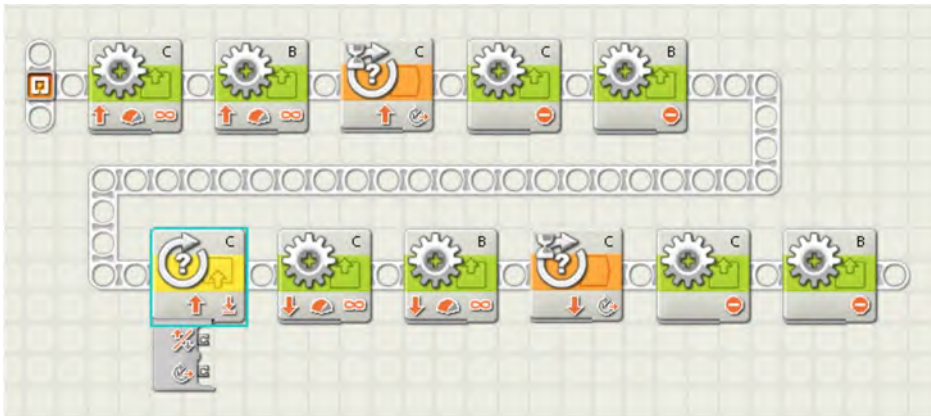
- Αποθηκεύστε το πρόγραμμά σας ως **κίνηση μπροστά για 2 περιστροφές και επιστροφή**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Πειραματιστείτε :

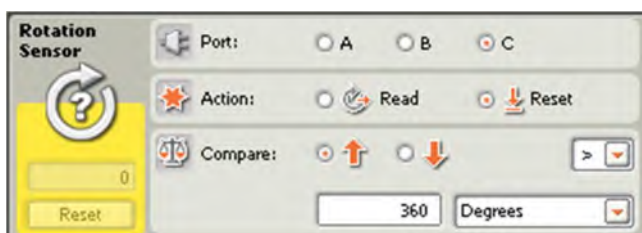
Περιγράψτε τι ακριβώς συνέβη.

Μπορείτε να ερμηνεύσετε τη συμπεριφορά που παρατηρήσατε ;

- Από την υποπαλέτα αισθητήρων **Sensors Action** (έχει ως σχήμα ένα τετράγωνο με ένα βελάκι προς τα κάτω και κίτρινο χρώμα) επιλέξτε το εικονίδιο του αισθητήρα περιστροφής το οποίο έχει ως σχήμα ένα κύκλο με ένα ερωτηματικό, και σύρετε το στην περιοχή εργασίας πριν από το δεύτερο μέρος του προγράμματος.



Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης επιλέξτε ως Θύρα (Port) στην οποία είναι συνδεδεμένος ο κινητήρας μας, τη Θύρα C αντί για την A και ως ενέργεια (Action) την αρχικοποίηση "Reset" αντί για την ανάγνωση "Read".



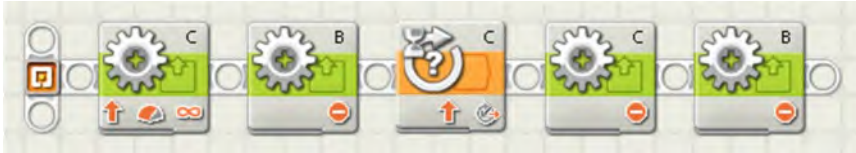
Σενάρια Ρομποτικής με το Mindstorms NXT

- Αποθηκεύστε τις αλλαγές στο πρόγραμμα σας ως **κίνηση μπροστά για 2 περιστροφές και επιστροφή**
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Συγχαρητήρια ☺
Ολοκληρώσατε την 3^η δραστηριότητα !!!

Δραστηριότητα 4^η : Στροφή αριστερά

- Ανοίξτε το πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** που δημιουργήσατε στην άσκηση 1.
- Επιλέξτε την εντολή κινητήρα B και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (*Until*) επιλέγοντας το σήμα του στοπ.



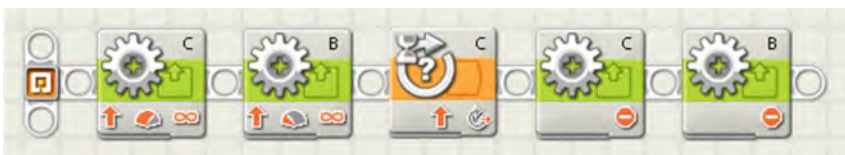
- Αποθηκεύστε το πρόγραμμά σας ως **στροφή αριστερά**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Στο πρόγραμμα γυρίζει η δεξιά ερπύστρια του οχήματος (αφού ξεκινάει ο κινητήρας C) ενώ η αριστερή του ερπύστρια δεν ακολουθεί (ο κινητήρας B είναι φρεναρισμένος) αλλά παραμένει καθηλωμένη στη θέση της. Εκτελώνοντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το όχημά μας θα εκτελέσει μία κλειστή στροφή αριστερά, με κέντρο τη ερπύστρια που είναι ακινητοποιημένη και θα σταματήσει μετά από δύο περιστροφές του κινητήρα C

Σημείωση : Δεν είναι απαραίτητο να τοποθετήσουμε την τελευταία εντολή με την οποία φρενάρουμε τον κινητήρα B.

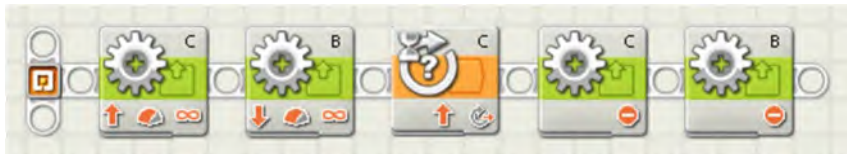
Πειραματιστείτε :

α) Τι θα συμβεί στο όχημά μας στην περίπτωση που αντί να φρενάρουμε τον κινητήρα B, τον ρυθμίσουμε να γυρίζει προς τα μπροστά με μικρή ταχύτητα (αρκετά μικρότερη σε σχέση με τον κινητήρα C) ;

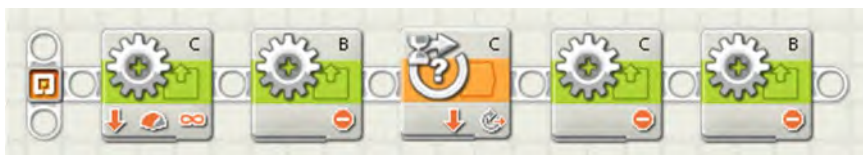


Σενάρια Ρομποτικής με το Mindstorms NXT

β) Τι θα συμβεί στο όχημα μας στην περίπτωση που αντί να φρενάρουμε τον κινητήρα Β, τον ρυθμίσουμε να γυρίζει προς τα πίσω (δηλ. προς την αντίθετη κατεύθυνση) ;



γ) Τι θα συμβεί στο όχημα μας στην περίπτωση που ρυθμίσουμε τον κινητήρα C να γυρίζει προς τα πίσω αντί για μπροστά ;



Συγχαρητήρια ☺
Ολοκληρώσατε την 4^η δραστηριότητα !!!

Δραστηριότητα 5^η : Στροφή δεξιά

- Ανοίξτε το πρόγραμμα **κίνηση μπροστά για 2 περιστροφές** που δημιουργήσαμε στην άσκηση 1.

Θέλουμε το πρόγραμμα να κινήσει το όχημα-ρομπότ μας ώστε να εκτελέσει μία στροφή δεξιά και να σταματήσει μετά από 2 περιστροφές.

- Αυτή τη φορά επιλέξτε την εντολή κινητήρα C και από τον πίνακα ρυθμίσεων που εμφανίζεται, αλλάξτε την επιλογή της κατεύθυνσης (*Until*) επιλέγοντας το σήμα του στοπ.



Πειραματιστείτε :

Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το όχημα μας θα κινηθεί εκτελώντας μία στροφή δεξιά αλλά δε θα σταματήσει να κινείται ποτέ. Μπορείτε να εξηγήσετε για ποιο λόγο συμβαίνει αυτό ;

- Στην εντολή αναμονής για περιστροφή αλλάξτε τη θύρα του κινητήρα που θα παρακολουθείται, ορίζοντας τη θύρα B αντί για τη C.



- Αποθηκεύστε το πρόγραμμά σας ως **στροφή δεξιά**.
- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Συγχαρητήρια ☺
Ολοκληρώσατε την 5^η δραστηριότητα !!!

Φύλλο Αξιολόγησης : Προγραμματιστικό Περιβάλλον NXT-G

Όνομα _____

Ημερομηνία _____

1. Συμπληρώστε παρακάτω ποια είναι η λειτουργία του κάθε κουμπιού.



.....

.....

.....

.....

.....

.....

.....

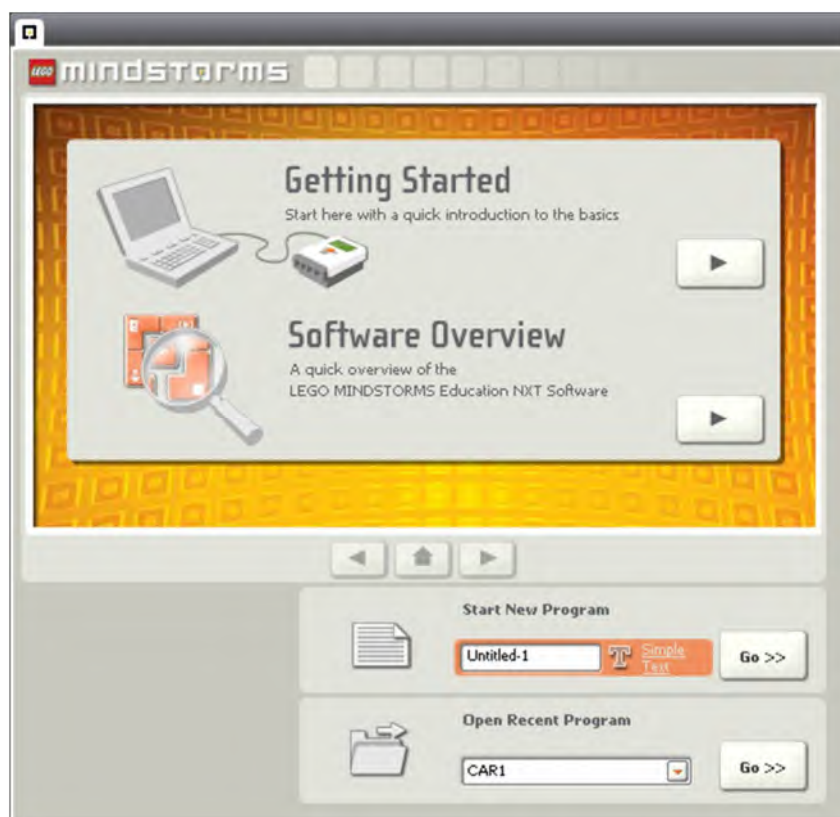
2. Απαντήστε με συντομία στις παρακάτω ερωτήσεις.

Γράψε τις απαντήσεις εδώ



- | | | | |
|-------|-----|---|---|
| _____ | 1. | Πόσες είναι οι θύρες εισόδου του NXT ; | |
| _____ | 2. | Πόσες είναι οι θύρες εξόδου του NXT ; | |
| _____ | 4. | Ανέφερε 2 συσκευές που μπορούν να συνδεθούν στις θύρες εξόδου του NXT | |
| _____ | 5. | Ανέφερε 2 συσκευές που μπορούν να συνδεθούν στις θύρες εξόδου του NXT | |
| _____ | 6. | Τι αντιπροσωπεύει η εικόνα ; |  |
| _____ | 7. | Τι αντιπροσωπεύει η εικόνα ; |  |
| _____ | 8. | Τι αντιπροσωπεύει η εικόνα ; |  |
| _____ | 8. | Τι αντιπροσωπεύει η εικόνα ; |  |
| _____ | 9. | Τι αντιπροσωπεύει η εικόνα ; |  |
| _____ | 10. | Τι αντιπροσωπεύει η εικόνα ; |  |

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασίας 1^ο

Γνωριμία με το προγραμματιστικό περιβάλλον NXT-G

Σημειώσεις Καθηγητή

Σκοπός : να μάθουν να χρησιμοποιούν το περιβάλλον NXT-G, τις εντολές και τις δυνατότητες του.

Εξοικείωση με το NXT-G

- Το NXT-G είναι ένα προγραμματιστικό περιβάλλον το οποίο χρησιμοποιούμε για να πούμε στο ρομπότ μας τι να κάνει.
- Γράφουμε το πρόγραμμα μας στον προσωπικό υπολογιστή μας και μετά το κατεβάζουμε (κάνουμε downloading) στο NXT του ρομπότ μας χρησιμοποιώντας καλωδίωση USB ή εναλλακτικά ασύρματη σύνδεση Bluetooth
- Το NXT-G χρησιμοποιεί εικονίδια (*blocks*) τα οποία αναπαριστούν διαφορετικές εργασίες που πρέπει να εκτελέσει το ρομπότ μας. Για να δημιουργήσουμε μια λογική σειρά από εντολές που θα εκτελέσει το ρομπότ μας, βάζουμε τα εικονίδια αυτά σε μία συγκεκριμένη σειρά και τα συνδέουμε μεταξύ τους.

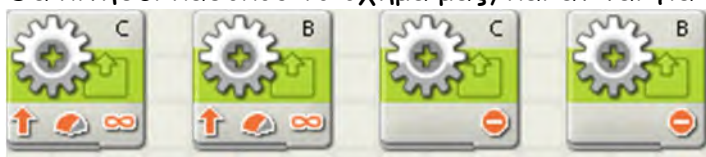
Δραστηριότητα 1 ^η : κίνηση μπροστά για 2 περιστροφές (Ένα απλό πρόγραμμα)

Σημείωση : Ο κινητήρας μόλις συμπληρωθούν οι δύο περιστροφές δε θα σταματήσουν ακαριαία αλλά θα συνεχίσουν να κινούνται για ένα πολύ μικρό χρονικό διάστημα. Το πρόγραμμα αυτό θέτει σε λειτουργία τον κινητήρα C και αμέσως μετά και τον κινητήρα B και στη συνέχεια περιμένει μέχρι ο κινητήρας C να συμπληρώσει δύο περιστροφές. Μόλις συμβεί αυτό δεν υπάρχει καμία εντολή για να πει στους κινητήρες να σταματήσουν να λειτουργούν προτού το πρόγραμμα τερματιστεί. Με τον τερματισμό ενός προγράμματος διακόπτεται η τροφοδοσία σε όλες τις θύρες εξόδου του NXT οπότε οι κινητήρες σταματάνε αλλά όχι ακαριαία όπως θα θέλαμε, παρά μόνο μετά από ολίσθηση μέχρι να σταματήσουν εντελώς λόγω της τριβής. Έτσι αν παρατηρήσετε πιο προσεκτικά το όχημα, θα διαπιστώσετε ότι αυτό θα συνεχίσει να κινείται για λίγο ακόμη αφού ολοκληρωθούν οι δύο περιστροφές που θέσαμε με την εντολή της αναμονής, μέχρι να σταματήσει εντελώς λόγω της τριβής.

Πειραματιστείτε :

Με το δεδομένο ότι όταν ολοκληρωθεί η εκτέλεση ενός προγράμματος, το τουβλάκι NXT διακόπτει αυτόματα την τροφοδοσία ρεύματος και στις τρεις εξόδους του (όπου και έχουμε συνδέσει τους κινητήρες) αν αφαιρέσουμε την εντολή αναμονής τι πιστεύετε ότι θα συμβεί;

Θα κινηθεί καθόλου το όχημα μας, και αν ναι για πόσο διάστημα;



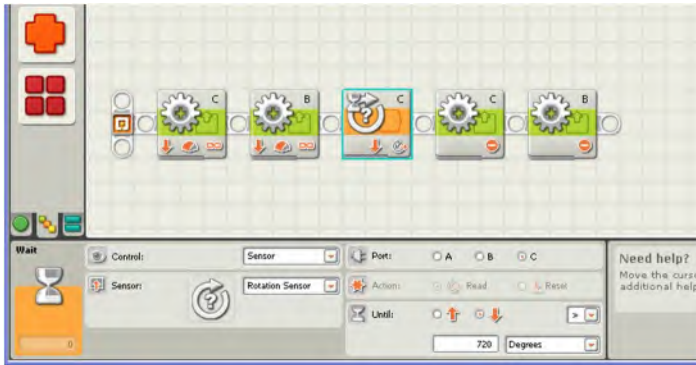
Δοκιμάστε το στην πράξη. Τι ακριβώς συνέβη;

Προσπαθήστε να το εξηγήσετε. (Τεκμηριώστε την άποψη σας)

Απάντηση :

Η εντολή για αναμονή είναι αυτή που ελέγχει το χρονικό διάστημα που θα κινηθούν οι κινητήρες προτού σταματήσουν η λειτουργία τους. Με την εντολή αναμονής το ρομπότ μας θα περάσει στην επόμενη εντολή μετά το χρονικό διάστημα που ορίζεται από την εντολή αναμονής. Μέχρι τότε θα συνεχίσει να περιστρέφει τους κινητήρες του. Αν δεν υπάρχει καμία εντολή για αναμονή θα θέσει σε λειτουργία τους κινητήρες και αμέσως μετά θα τους σταματήσει πριν καν προλάβουν αυτοί να ξεκινήσουν.

Δραστηριότητα 2^η : κίνηση προς τα πίσω για 2 περιστροφές (τροποποίηση του προγράμματος)



Δραστηριότητα 3^η : κίνηση μπροστά για 2 περιστροφές και επιστροφή (επέκταση του προγράμματος)

Πειραματιστείτε :



Περιγράψτε τι ακριβώς συνέβη.

Μπορείτε να ερμηνεύσετε τη συμπεριφορά που παρατηρήσατε ;

Απάντηση :

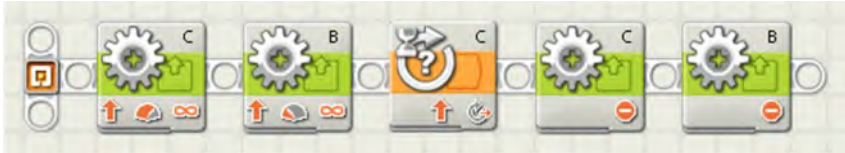
Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το όχημα μας θα κινηθεί προς τα μπροστά για 2 περιστροφές ακριβώς, θα σταματήσει και θα κινηθεί προς τα πίσω για 4 περιστροφές. Δεν θα σταματήσει δηλαδή στο σημείο εκκίνησης αλλά θα κινηθεί επιπλέον για άλλες 720 μοίρες (2 περιστροφές). Το σημείο αναφοράς του αισθητήρα περιστροφής είναι το σημείο που βρίσκεται το όχημα τη στιγμή που ξεκινάει η εκτέλεση του προγράμματος. Θα πρέπει να αλλάξουμε τη συνθήκη αναμονής στο δεύτερη εντολή αναμονής του προγράμματος έτσι ώστε να περιμένει μέχρι να περιστραφεί ο κινητήρας στη θέση 0 μοίρες και όχι 720° πίσω.

Εναλλακτικά μπορούμε πριν από δεύτερο μέρος του προγράμματος, να τοποθετήσουμε μία εντολή για να αναγκάσουμε τον αισθητήρα περιστροφής να αρχίσει να μετράει από την αρχή προτού αρχίσει να κινείται προς τα πίσω. Για το πετύχουμε αυτό χρησιμοποιούμε μία εντολή αισθητήρα περιστροφής.

Δραστηριότητα 4^η : Στροφή αριστερά

Πειραματιστείτε :

α) Τι θα συμβεί στο όχημα μας στην περίπτωση που αντί να φρενάρουμε τον κινητήρα Β, τον ρυθμίσουμε να γυρίζει προς τα μπροστά με μικρή ταχύτητα (αρκετά μικρότερη σε σχέση με τον κινητήρα C) :



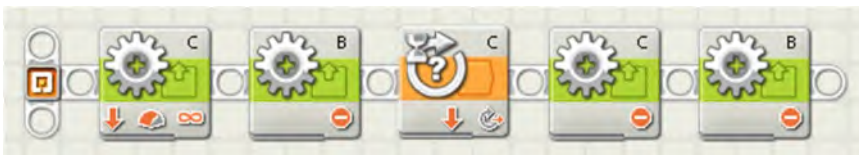
Απάντηση : Το όχημα μας θα κινηθεί εκτελώντας μία ομαλή στροφή αριστερά.

β) Τι θα συμβεί στο όχημα μας στην περίπτωση που αντί να φρενάρουμε τον κινητήρα Β, τον ρυθμίσουμε να γυρίζει προς τα πίσω (δηλ. προς την αντίθετη κατεύθυνση) :



Απάντηση : το όχημα μας θα εκτελέσει μία στροφή αριστερά επιτόπου με κέντρο το μέσο του οχήματος.

γ) Τι θα συμβεί στο όχημα μας στην περίπτωση που ρυθμίσουμε τον κινητήρα C να γυρίζει προς τα πίσω αντί για μπροστά



Απάντηση : το όχημα μας θα εκτελέσει μία στροφή δεξιά με την όπισθεν.

Δραστηριότητα 5^η : Στροφή δεξιά

Πειραματιστείτε :



Εκτελώντας το παραπάνω πρόγραμμα μπορείτε να παρατηρήσετε ότι το όχημα μας θα κινηθεί εκτελώντας μία στροφή δεξιά αλλά δε θα σταματήσει να κινείται ποτέ. Μπορείτε να εξηγήσετε για ποιο λόγο συμβαίνει αυτό ;

Απάντηση :

Στο πρόγραμμα αυτό ξεκινάει η λειτουργία του κινητήρα B και στη συνέχεια η εντολή αναμονής βάζει το πρόγραμμα να περιμένει ώσπου ο κινητήρας C να συμπληρώσει δύο περιστροφές !!! Όμως ο κινητήρας C δε κινείται καθόλου, είναι σταματημένος. Το όχημα μας θα κινείται συνέχεια εκτελώντας κύκλους προς τα δεξιά. Για να λειτουργήσει το πρόγραμμα όπως θα θέλαμε, πρέπει στην εντολή αναμονής για περιστροφή να αλλάξουμε τη θύρα του κινητήρα που θα παρακολουθείται, ορίζοντας τη θύρα B αντί για τη C.



ΓΝΩΣΤΙΚΟ ΑΝΤΙΚΕΙΜΕΝΟ:

**ΠΛΗΡΟΦΟΡΙΚΗ
ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
ΜΕ LEGO MINDSTORMS NXT**

Α) Περιγραφή Διδακτικής Παρέμβασης στην τάξη – με χρήση ΤΠΕ

Δημιουργός σεναρίου: Γεώργιος Κυριακού

Ημ/νίες εφαρμογής:

Τάξεις/Τμήματα: Γ' Γυμνασίου / Α' Λυκείου

Τίτλος διδακτικής παρέμβασης

Εντολές κίνησης και Δομή Επανάληψης

Γνωστικό αντικείμενο και εμπλεκόμενες γνωστικές περιοχές

Πληροφορική Γ' Γυμνασίου. Διδακτική Ενότητα: 1.2 :

Ο Προγραμματισμός στην πράξη

Τάξη εφαρμογής

Γ' Γυμνασίου (Νέα πιλοτικά προγράμματα σπουδών)

Συμβατότητα και συσχέτιση με το Αναλυτικό Πρόγραμμα

Προγραμματίζω τον υπολογιστή - Οι μαθητές σχεδιάζουν και υλοποιούν προγράμματα που απαιτούν χρήση δομής επανάληψης σύμφωνα με τα νέα πιλοτικά προγράμματα σπουδών.

Υλικοτεχνική υποδομή

Εργαστήριο πληροφορικής, βιντεοπροβολέας, πίνακας μαρκαδόρου, πακέτο Lego Mindstorms NXT Education.

Το Λογισμικό

Περιβάλλον προγραμματισμού Lego NXT-G, λογισμικό παρουσιάσεων Powerpoint

Σκοπός

Να αποκτήσει ο μαθητής σχετική ευχέρεια και εμπειρία στη δεξιότητα του δομημένου προγραμματισμού.

Διδακτικοί στόχοι:

Επαφή και πρακτική εξάσκηση με τις δυνατότητες του λογισμικού NXT-G.

- Να γνωρίσουν τις εντολές εξόδου για τους κινητήρες και τις λαμπτήρες
- Να αποκτήσουν ευχέρεια στον προγραμματισμό ενός ρομπότ.
- Να γνωρίσουν την προγραμματιστική δομή της επανάληψης.

Γενικότεροι μαθησιακοί στόχοι

- να αντιλαμβάνονται την αναγκαιότητα και τη χρησιμότητα της δομής επανάληψης
- να εφαρμόζουν τεχνικές ελέγχου και διόρθωσης σφαλμάτων στα προγράμματα που δημιουργούν.

Γενικότεροι εκπαιδευτικοί στόχοι

- συνεργασία των μαθητών για εξαγωγή αποτελεσμάτων και συμπερασμάτων

Στόχοι ως προς την αξιοποίηση της τεχνολογίας

- εξοικείωση των μαθητών με τη χρήση των εντολών εξόδου και της δομής επανάληψης σε περιβάλλον Lego NXT-G

Διδακτική μέθοδος

Αρχικά με διάλεξη περιορισμένου χρόνου και αντίστοιχη επίδειξη
Στη συνέχεια οι μαθητές εργάζονται σε ομάδες (κοινωνιο-επικοινωνιακή μάθηση),
έμμεση διδασκαλία - μαθητοκεντρική

Διδακτικές στρατηγικές / τεχνικές

Μορφή καθοδήγησης (θέτουμε ερωτήματα και οι μαθητές επιλέγουν βήματα και δράσεις σε πλαίσιο καθοδήγησης), πειραματισμός, διερεύνηση, ανακάλυψη, επίλυση προβλήματος.

Οργάνωση διδασκαλίας

Ομάδες των τριών ατόμων. Αναθέτουμε σε ένα μέλος της ομάδας το ρόλο του εκπροσώπου, σε ένα άλλο το ρόλο του γραμματέα ο οποίος θα κρατάει τις σημειώσεις και στο τελευταίο μέλος το ρόλο του προγραμματιστή στο NXT-G. Κάθε ομάδα έχει το δικό της υπολογιστή.

Ρόλος εκπαιδευτικού

Καθοδηγητικός. Καθοδηγούμε τη διαδικασία ώστε να αναδεικνύονται οι προϋπάρχουσες αντιλήψεις των μαθητών, οι στρατηγικές τους, προκαλούμε την αποτελεσματική συζήτηση και την αλληλεπίδραση στις ομάδες και στην ολομέλεια, παροτρύνουμε τους μαθητές που συναντούν δυσκολίες.

Εκτιμώμενη διάρκεια

2 διδακτικές ώρες

2. ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ

Το επιστημονικό περιεχόμενο

Προγραμματισμός → Δομή επανάληψης

Προαπαιτούμενες γνώσεις:

1. Εισαγωγή στη Ρομποτική (βλέπε σημειώσεις: [B1- Ρομποτική](#))
2. Γνωριμία με το Πακέτο Lego Mindstorms (βλέπε σημειώσεις: [B2- Πακέτο Lego Mindstorms NXT](#))
3. Εξοικείωση με το Λογισμικό NXT-G (βλέπε σημειώσεις: [B3- Το προγραμματιστικό περιβάλλον NXT-G](#) και [Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G \(Καθηγητή\)](#))

Περιγραφή δραστηριοτήτων

Αναλυτική παρουσίαση στο έγγραφο “[A2- Πορεία Διδασκαλίας](#)” (Επισυνάπτεται)

1. Παρουσίαση των εντολών του NXT-G για τις προγραμματιστικές δομές της επανάληψης με ταυτόχρονη προβολή της παρουσίασης “[3.1 - Εντολές του NXT-G](#)”.
2. Πρακτική εξάσκηση και εξοικείωση των μαθητών με τις εντολές εξόδου για τους κινητήρες και τη δομή επανάληψης, ακολουθώντας το Φύλλο Εργασιών “[Φύλλο Εργασιών 2 - Κινητήρες και Δομή επανάληψης](#)”

Φύλλα εργασίας

[Φύλλο Εργασιών 2 - Κινητήρες και Δομή επανάληψης](#) (Επισυνάπτεται)

Αξιολόγηση μαθητών

Αξιολόγηση της ικανότητας χρήσης της δομής επανάληψης μέσα από τις δραστηριότητες οι οποίες θα υλοποιηθούν και της συμπλήρωσης του φύλλου εργασίας και αξιολόγησης.

Αξιολόγηση σεναρίου

Αξιολόγηση μέσα από τις ίδιες δραστηριότητες και της ικανότητας συνεργασίας των μαθητών, όπως και από τη συμπλήρωση του φύλλου αξιολόγησης.

B) Επισυναπτόμενο υλικό:

Παρουσίαση :

[3.1 - Εντολές του NXT-G](#)

Φύλλα εργασιών :

[Φύλλο Εργασιών 2 - Κινητήρες και Δομή επανάληψης](#)

Σημειώσεις του καθηγητή :

[A2- Πορεία Διδασκαλίας](#)

[B3- Το προγραμματιστικό περιβάλλον NXT-G](#)

[B4- Οι εντολές του NXT-G](#)

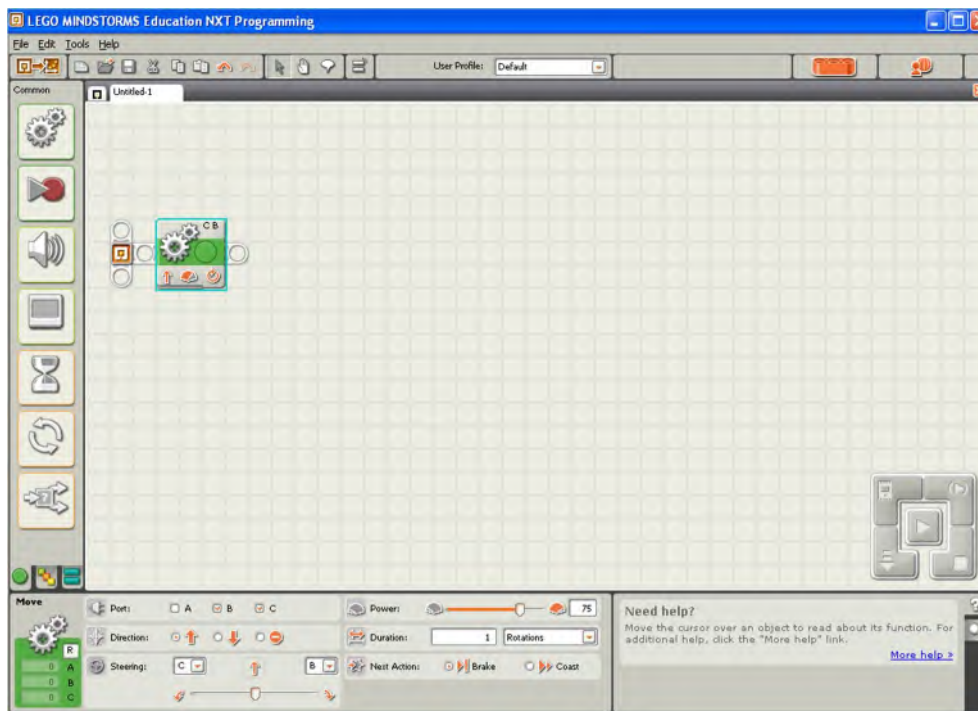
[Φύλλο Εργασιών 2 - Κινητήρες και Δομή επανάληψης \(Καθηγητή\)](#)

Έτοιμα προγράμματα :

[Φάκελος 2 - Κινητήρες και επανάληψη](#)

Γ) Αποτελέσματα –Σχόλια – Προβληματισμοί (Αναστοχασμός)

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 2^ο

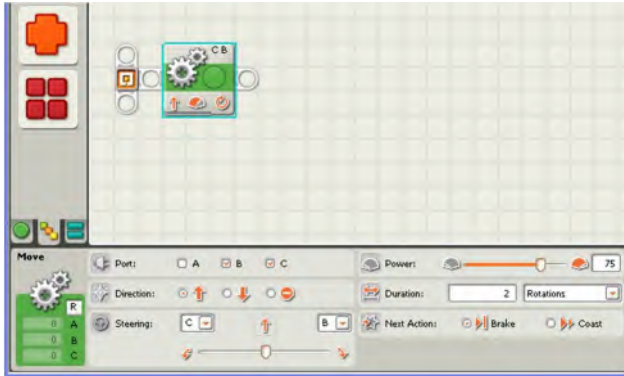
Κινητήρες και Δομή Επανάληψης

Όνομα _____

Ημερομηνία _____

Δραστηριότητα 6^η : κίνηση μπροστά συγχρονισμένα

- Ξεκινήστε ένα νέο πρόγραμμα.
- Από το υπομενού των βασικών ενεργειών **Common** (έχει ως σχήμα ένα πράσινο κύκλο) επιλέξτε το εικονίδιο για την εντολή μετακίνησης, το οποίο έχει ως σχήμα δύο γρανάζια συζευγμένα, και τοποθετείστε το στο χώρο εργασίας δίπλα στο σημείο της αρχής του προγράμματος.



- Στον πίνακα ρυθμίσεων που εμφανίζεται στο κάτω μέρος της οθόνης αλλάζτε τη διάρκεια της μετακίνησης (Duration) σε 2 περιστροφές. Μην πειράξετε καμία άλλη ρύθμιση.



Το πρόγραμμα μας είναι έτοιμο. Δε χρειάζεται να τοποθετήσουμε κάποια εντολή για να σταματήσουμε τους κινητήρες με το πέρας της κίνησης μετά από 2 περιστροφές, αφού όπως έχουμε αναφέρει αυτοί θα φρενάρουν αυτόματα.

- Αποθηκεύστε το πρόγραμμα ως 6.

Πειραματιστείτε :

α) Ποια ρύθμιση πρέπει να αλλάξουμε στο προηγούμενο πρόγραμμα για να μετακινηθεί το όχημα-ρομπότ μας προς τα πίσω για 2 περιστροφές ;

β) Τι πρέπει να αλλάξουμε στο προηγούμενο πρόγραμμα Για να στρίψει το όχημα-ρομπότ μας προς τα αριστερά για δύο περιστροφές ;

γ) Τι πρέπει να κάνουμε για να μετακινηθεί το όχημα-ρομπότ μας προς τα μπροστά για 2 περιστροφές και να επιστρέψει πίσω ;

Δραστηριότητα 7^η : εκτέλεση κίνησης

Εξηγήστε τι κίνηση θα κάνει το όχημα - ρομπότ μας, αν φορτώσουμε το παρακάτω πρόγραμμα :



- Υλοποιήστε το και τρέξτε το. Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;
ΝΑΙ
ΟΧΙ
- Αποθηκεύστε το πρόγραμμα ως 7.

Δραστηριότητα 8^η : τετράγωνο

Προσπαθήστε να δημιουργήσετε ένα πρόγραμμα, ώστε το όχημα-ρομπότ μας να κινηθεί ακολουθώντας μία διαδρομή ή οποία να έχει σχήμα τετραγώνου.

Υπόδειξη : Παρατηρήστε ότι για να αποκτήσει το όχημα μας την αναμενόμενη συμπεριφορά πρέπει να επαναλάβει κάποιες ενέργειες για ένα συγκεκριμένο αριθμό φορές.

Περιγράψτε τις κινήσεις που πρέπει να επαναλάβει το όχημα μας ;

Πόσες φορές θα επαναλάβει τις κινήσεις αυτές ; _____

Υπόδειξη : Για τη δημιουργία του προγράμματος χρησιμοποιήστε τη ενέργεια της αντιγραφής και επικόλλησης.

- Αποθηκεύστε το πρόγραμμα σας ως 8 και τρέξτε το.

Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

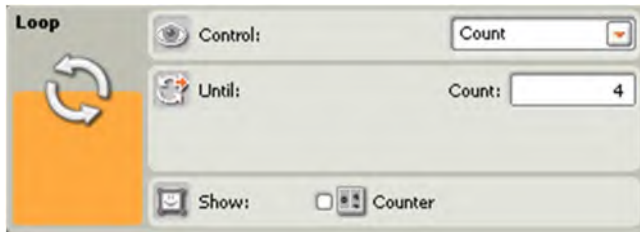
ΟΧΙ

Δραστηριότητα 9^η : τετράγωνο με δομή επανάληψης

Θέλουμε το όχημα-ρομπότ μας να κινηθεί ακολουθώντας μία διαδρομή ή οποία να έχει σχήμα τετραγώνου, όπως και στη δραστηριότητα 8, κάνοντας χρήση αυτή τη φορά, της εντολή επανάληψης για συγκεκριμένες αριθμό επαναλήψεων.

- Ανοίξτε το πρόγραμμα που δημιουργήσατε στην προηγούμενη δραστηριότητα 8 και τροποποιήστε το κατάλληλα χρησιμοποιώντας αυτή τη φορά, μία εντολή επανάληψης.

Υπόδειξη : Στον πίνακα ελέγχου της εντολής επανάληψης πρέπει να αλλάξετε τη επιλογή του ελέγχου της επανάληψης (Control) από "Forever" σε "Count".



- Αποθηκεύστε το πρόγραμμα σας ως **9** και τρέξτε το.

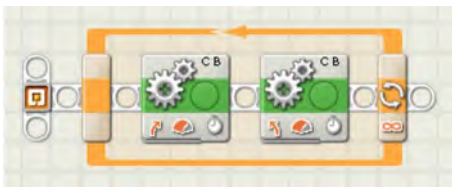
Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

ΟΧΙ

Δραστηριότητα 10 ^η : εκτέλεση συνεχόμενης κίνησης

Εξηγήστε τι κίνηση θα κάνει το όχημα - ρομπότ μας, αν φορτώσουμε το παρακάτω πρόγραμμα :



- Υλοποιήστε το και τρέξτε το. Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

ΟΧΙ

- Αποθηκεύστε το πρόγραμμα σας ως **10**.

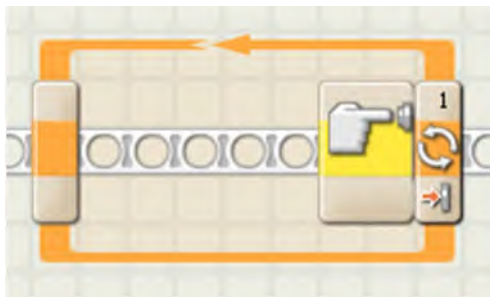
Δραστηριότητα 11 ^η : κίνηση φιδωτά και σταμάτημα με αισθητήρα αφής

Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα αφής.

Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

- Ανοίξτε το πρόγραμμα που δημιουργήσατε στην προηγούμενη δραστηριότητα **10** και τροποποιήστε το κατάλληλα, ώστε το όχημα-ρομπότ μας να κινείται φιδωτά μέχρι να κτυπήσει σε ένα εμπόδιο.

Υπόδειξη : Στον πίνακα ελέγχου της εντολής επανάληψης πρέπει να αλλάξετε τη επιλογή του ελέγχου της επανάληψης (Control) από "Forever" σε "Sensor". Ο αισθητήρας που θέλουμε είναι ο αισθητήρας αφής και άρα δεν πειράζουμε καμία άλλη επιλογή.



- Αποθηκεύστε το πρόγραμμά σας ως **11** και τρέξτε το.

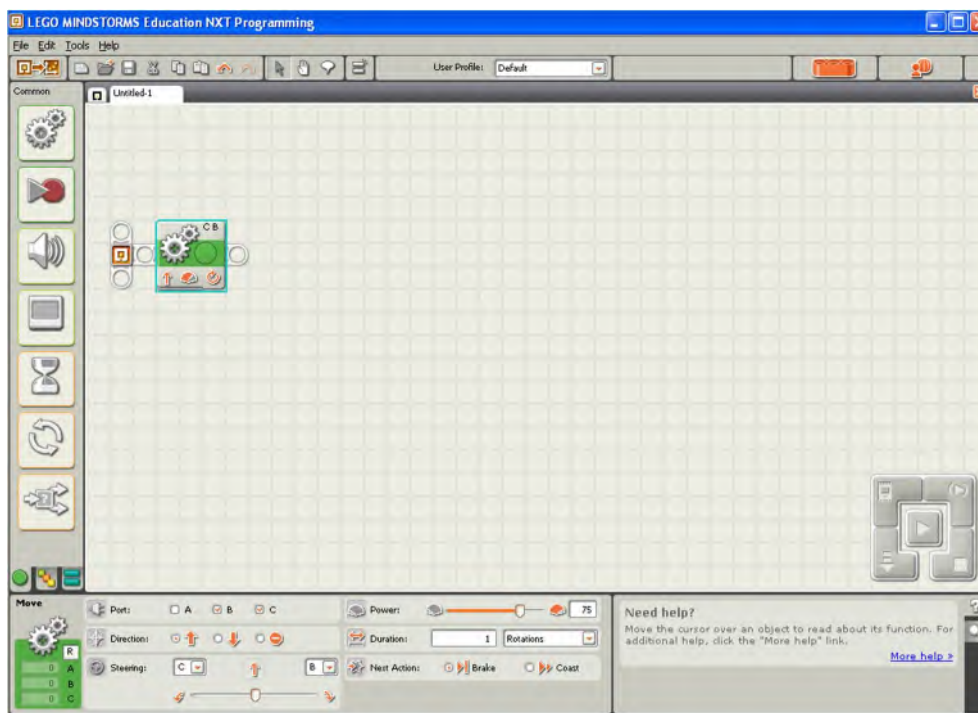
Είχε το όχημά μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

ΟΧΙ

Συγχαρητήρια ☺
Ολοκληρώσατε το 2^ο φύλλο δραστηριοτήτων!!!

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 2^ο

Κινητήρες και Δομή Επανάληψης

Σημειώσεις Καθηγητή

Σενάρια Ρομποτικής με το Mindstorms NXT

Τώρα θα δούμε πως μπορούν να τροποποιηθούν τα προγράμματα που υλοποιήσαμε στις προηγούμενες ασκήσεις με τη χρήση αυτή τη φορά της εντολής μετακίνησης.

Με τις παρακάτω δραστηριότητες οι μαθητές θα γνωρίσουν και με ποιο τρόπο κινείται το όχημα tankbot. Συγκεκριμένα :

- α. Για να προχωρήσει μπροστά ή πίσω σε ευθεία πρέπει να τεθούν και οι δύο κινητήρες ταυτόχρονα με την ίδια ταχύτητα και κατεύθυνση.
- β. Για να πάρει μία ανοικτή στροφή μπροστά ή πίσω πρέπει να τεθούν και οι δύο κινητήρες ταυτόχρονα με τη διαφορετική ταχύτητα αλλά ίδια κατεύθυνση
- γ. Για να στρίψει με άξονα τη μία ερπύστρια του μπροστά ή πίσω πρέπει να κινηθεί μόνο ο ένας κινητήρας ενώ ο άλλος παραμένει φρεναρισμένος..
- α. Για να στρίψει επιτόπου μπροστά ή πίσω πρέπει να τεθούν και οι δύο κινητήρες ταυτόχρονα με την ίδια ταχύτητα αλλά με διαφορετική κατεύθυνση.

Δραστηριότητα 6 ^η : κίνηση μπροστά συγχρονισμένα

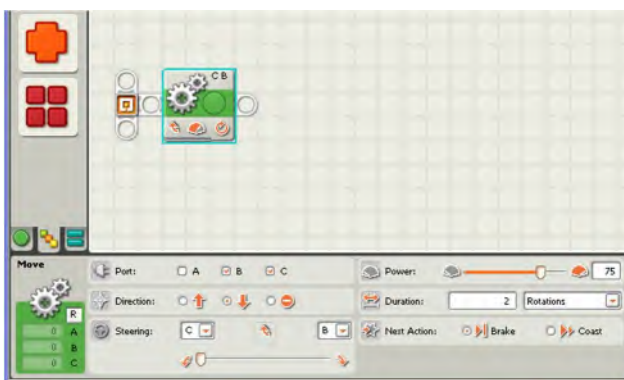
Πειραματιστείτε :

α) Ποια ρύθμιση πρέπει να αλλάξουμε στο προηγούμενο πρόγραμμα για να μετακινηθεί το όχημα-ρομπότ μας προς τα πίσω για 2 περιστροφές ;

Απάντηση : αρκεί να αλλάξουμε την κατεύθυνση (Direction) της κίνησης, στον πίνακα ρυθμίσεων της εντολής μετακίνησης, προς τα πίσω.

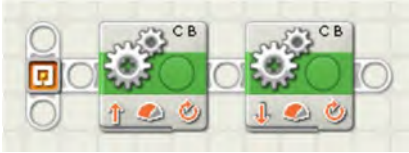
β) Τι πρέπει να αλλάξουμε στο προηγούμενο πρόγραμμα Για να στρίψει το όχημα-ρομπότ μας προς τα αριστερά για δύο περιστροφές ;

Απάντηση : αρκεί να μετακινήσουμε ανάλογα τη μπάρα του τιμονιού (Steering) στον πίνακα ρυθμίσεων της εντολής μετακίνησης.



γ) Τι πρέπει να κάνουμε για να μετακινηθεί το όχημα-ρομπότ μας προς τα μπροστά για 2 περιστροφές και να επιστρέψει πίσω

Απάντηση: αρκεί να τοποθετήσουμε ακόμη μία εντολή μετακίνησης. Στην περίπτωση μάλιστα που χρησιμοποιούμε εντολές μετακίνησης, δε χρειάζεται να τοποθετήσουμε την εντολή του αισθητήρα περιστροφής με την κίτρινη λουρίδα (Rotation Sensor) για να μηδενίζουμε το μετρητή.



Σημείωση: Το ρομπότ μας μπορεί να πάρει μία ομαλή ή πιο απότομη στροφή ανάλογα με τη θέση που θα τοποθετήσουμε το σημάδι στη μπάρα του τιμονιού. Δεν υπάρχει όμως τρόπος να στρίψει το ρομπότ μας επιτόπου με μία μόνο εντολή μετακίνησης.

Δραστηριότητα 7^η : εκτέλεση κίνησης (παρκάρισμα)

Εξηγήστε τι κίνηση θα κάνει το όχημα - ρομπότ μας, αν φορτώσουμε το παρακάτω πρόγραμμα :

Το όχημα-ρομπότ θέλουμε να κινηθεί προς τα μπροστά για 30 cm, να στρίψει 90ο προς τα αριστερά, να σταματήσει και στη συνέχεια να κινηθεί προς τα πίσω για 20 cm και να σταματήσει.

Καθώς πηγαίνει με την όπισθεν να ανάβει μία λάμπα και να ακούγεται ένας ήχος



Υπόδειξη : Για να βρούμε ακριβώς τον αριθμό των περιστροφών που πρέπει να γυρίσει ένας κινητήρας ώστε το όχημα μας να διανύσει συγκεκριμένη απόσταση ακολουθήστε την παρακάτω διαδικασία :

Στο κυρίως μενού του τούβλου NXT βρείτε το μενού εμφάνισης [View] και επιλέξτε το υπομενού με την εικόνα ενός κινητήρα [Motor Rotations] για περιστροφές ή [Motor Degrees] για μοίρες. Στη συνέχεια επιλέξτε τη θύρα που είναι συνδεδεμένος ο κινητήρας. Αν τώρα σπρώξετε με το χέρι σας το ρομπότ μπροστά θα δείτε στην οθόνη του τούβλου NXT πόσες κινήσεις θα κάνει ο κινητήρας για να καλύψει τη συγκεκριμένη απόσταση.

Για να πετύχετε μία στροφή 90ο ακριβώς, στρίψτε με το χέρι το όχημα μας κρατώντας φρεναρισμένο τον ένα κινητήρα και δείτε στην οθόνη του τούβλου NXT τις μοίρες ή τις περιστροφές που χρειάζεται να κινηθεί ο άλλος κινητήρας,

Η εντολή για το άναμμα ενός λαμπτήρα βρίσκεται στο υπομενού ενέργειας Action. Στον πίνακα ρυθμίσεων της, αφήνουμε τη θύρα εξόδου (Port) στο A και την ενέργεια (Action) στο "On".

Για να πούμε στο ρομπότ μας να αναπαράγει έναν ήχο τοποθετούμε την εντολή ήχου ή οποία βρίσκεται και αυτή στο υπομενού ενέργειας Στον πίνακα ρυθμίσεων της επιλέγουμε (File) να γίνει η αναπαραγωγή του αρχείου "attention" και ως λειτουργία (Function) επιλέγουμε ο ήχος να επαναλαμβάνεται "Repeat".

Σενάρια Ρομποτικής με το Mindstorms NXT

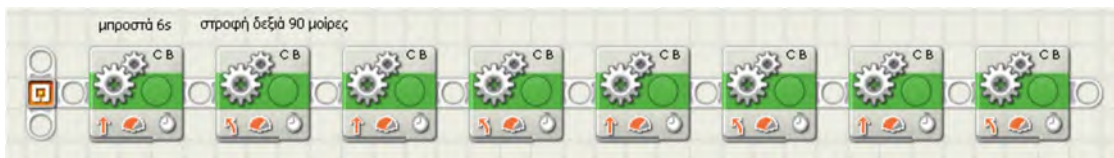
Σημείωση : Με το πέρας του προγράμματος οι κινητήρες θα σταματήσουν να λειτουργούν αυτόματα. Το ίδιο θα συμβεί με το φώς του λαμπτήρα και με την αναπαραγωγή του ήχου που τοποθετήσαμε στο πρόγραμμα μας, αφού με το πέρας ενός προγράμματος διακόπτεται η τροφοδοσία σε όλες τις θύρες εξόδου του NXT.

Δραστηριότητα 8 ^η : τετράγωνο

Το όχημα-ρομπότ μας θέλουμε να κινηθεί ακολουθώντας μία διαδρομή ή οποία να έχει σχήμα τετραγώνου.

Ξεκινάμε βάζοντας το ρομπότ μας να διανύσει μία απόσταση σε ευθεία γραμμή και στη συνέχεια να στρίψει κατά 90ο φρενάροντας τον έναν του κινητήρα.

Στη συνέχεια επαναλαμβάνουμε την παραπάνω διαδικασία άλλες τρεις φορές ώστε το ρομπότ μας να επιστρέψει στο σημείο από όπου ξεκίνησε.



Λίγα λόγια για τη δομή επανάληψης (ν φορές)

Πολλές φορές σε ένα πρόγραμμα θέλουμε μία σειρά ενεργειών να επαναληφθεί ένα συγκεκριμένο αριθμό φορές και όχι να επαναλαμβάνονται συνεχώς έως ότου διακόψουμε εμείς την εκτέλεση του προγράμματος. Στην περίπτωση αυτή γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων που πρέπει να γίνουν.

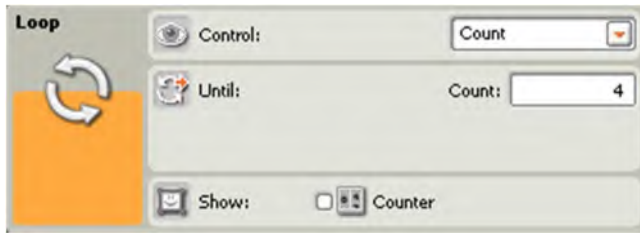
Δραστηριότητα 9 ^η : τετράγωνο με δομή επανάληψης

Θέλουμε το όχημα-ρομπότ μας να κινηθεί ακολουθώντας μία διαδρομή ή οποία να έχει σχήμα τετραγώνου, όπως και στη δραστηριότητα 8, χρησιμοποιώντας αυτή τη φορά τη δομή της επανάληψης.

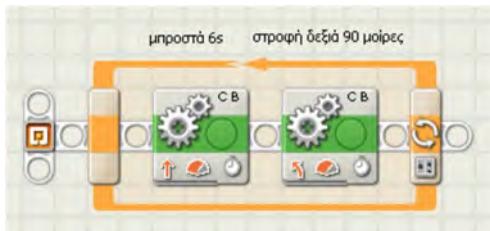
Όπως είχαμε δει στη δραστηριότητα 8 για να κινηθεί το ρομπότ σε ένα τετράγωνο χρειάζεται να επαναλάβει την ίδια διαδικασία (κίνηση μπροστά και στροφή δεξιά) 4 φορές.

Σενάρια Ρομποτικής με το Mindstorms NXT

Ξεκινάμε το πρόγραμμά μας εισάγοντας μία εντολή επανάληψης. Η εντολή αυτή όταν την τοποθετούμε στην περιοχή εργασίας, είναι ρυθμισμένη να επαναλαμβάνει το περιεχόμενο των εντολών που βρίσκονται στο εσωτερικό της για πάντα "Forever". Από τον πίνακα ρυθμίσεων μπορούμε να διαλέξουμε η επανάληψη να γίνει ένα αριθμό φορών αλλάζοντας τη επιλογή του ελέγχου της επανάληψης (Control) σε "Count". Πληκτρολογούμε στο αντίστοιχο πεδίο τον αριθμό 4.



Βλέπουμε λοιπόν ότι η δομή επανάληψης μας επιτρέπει να επαναλάβουμε ένα μέρος του προγράμματος μας, κάποιες συγκεκριμένες φορές. Με αυτό τον τρόπο δεν χρειάζεται να ξαναγράψουμε το ίδιο κομμάτι του προγράμματος μας (βρόχο) ξανά και ξανά.



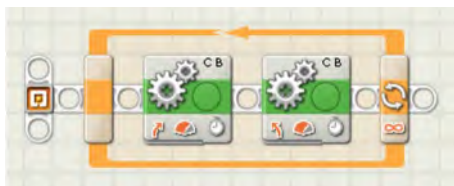
Λίγα λόγια για τη δομή επανάληψης (αένας βρόχος)

Συχνά μέσα στο πρόγραμμα χρειάζεται να πούμε στο ρομπότ μας να επαναλάβει μια συγκεκριμένη διαδικασία περισσότερες από μία φορές. Για να το πετύχουμε αυτό χρησιμοποιούμε μία δομή επανάληψης.



Δραστηριότητα 10^η : εκτέλεση συνεχόμενης κίνησης (κίνηση φιδωτή)

Εξηγήστε τι κίνηση θα κάνει το όχημα - ρομπότ μας, αν φορτώσουμε το παρακάτω πρόγραμμα :



Απάντηση : Το όχημα-ρομπότ κινείται φιδωτά, δηλαδή στρίβει μία αριστερά και μία δεξιά διαγράφοντας καμπύλες γραμμές.

Καταρχήν εισάγουμε το πρόγραμμά μας μία εντολή επανάληψης. Στο εσωτερικό της δομής επανάληψης τοποθετούμε δύο εντολές μετακίνησης. Με την πρώτη, το ρομπότ μας εκτελεί ανοικτή στροφή προς τα δεξιά ενώ με τη δεύτερη ανοικτή στροφή προς τα αριστερά. Στον πίνακα ρυθμίσεων των εντολών αυτών καθορίζουμε τη διάρκεια της κίνησης σε 2 δευτερόλεπτα. Επίσης στο πεδίο για την επόμενη ενέργεια

Σενάρια Ρομποτικής με το Mindstorms NXT

(Next Action) επιλέγουμε την ολίσθηση "Coast" των κινητήρων αντί για το φρενάρισμα τους "Break", ώστε η κίνηση του ρομπότ μας να είναι πιο ομαλή χωρίς διακοπές, αφού εξάλλου στο πρόγραμμα αυτό δεν μας ενδιαφέρει η ακρίβεια των κινήσεων.

Επισημάνση: Με τη δομή επανάληψης προσδίδουμε μόνιμες συμπεριφορές στο ρομπότ μας.

Λίγα λόγια για τη δομή επανάληψης (μέχρι να)

Μία ακόμη περίπτωση επανάληψης έχουμε όταν μία διαδικασία επαναλαμβάνεται συνέχεια όχι όμως για πάντα, αλλά μέχρι να ικανοποιηθεί ένας στόχος ή όσο ισχύει μία συνθήκη. Στην περίπτωση αυτή δε γνωρίζουμε από πιο μπροστά τον αριθμό των επαναλήψεων που θα συμβούν.

Δραστηριότητα 11 ^η: κίνηση μπροστά με σλάλομ και σταμάτημα με αισθητήρα αφής

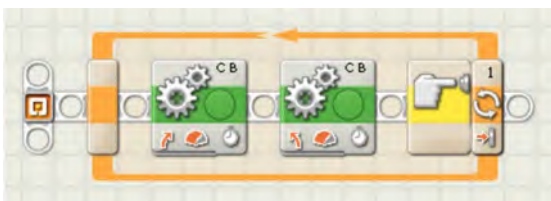
Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα αφής.

Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

Θέλουμε το όχημα-ρομπότ μας να κινείται κάνοντας σλάλομ (δηλαδή στρίβοντας μία δεξιά και μία αριστερά εναλλάξ) μέχρι να κτυπήσει σε ένα εμπόδιο.

- Ανοίξτε το πρόγραμμα που δημιουργήσατε στην προηγούμενη δραστηριότητα **10**, και τροποποιήστε το κατάλληλα, ώστε το όχημα-ρομπότ μας να κινείται φιδωτά μέχρι να κτυπήσει σε ένα εμπόδιο.

Από τον πίνακα ρυθμίσεων μπορούμε να διαλέξουμε η επανάληψη να γίνεται μέχρι να ικανοποιηθεί μία συγκεκριμένη συνθήκη αλλάζοντας τη επιλογή του ελέγχου της επανάληψης (Control) σε "Sensor". Στην περίπτωση αυτή έχουμε τη δυνατότητα να επιλέξουμε οποιοδήποτε αισθητήρα. Εμείς εδώ χρειαζόμαστε τον αισθητήρα αφής, οπότε δεν πειράζουμε καμία άλλη ρύθμιση.



ΓΝΩΣΤΙΚΟ ΑΝΤΙΚΕΙΜΕΝΟ:

**ΠΛΗΡΟΦΟΡΙΚΗ
ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
ΜΕ LEGO MINDSTORMS NXT**

Α) Περιγραφή Διδακτικής Παρέμβασης στην τάξη – με χρήση ΤΠΕ

Δημιουργός σεναρίου: Γεώργιος Κυριακού

Ημ/νίες εφαρμογής:

Τάξεις/Τμήματα: Γ' Γυμνασίου / Α' Λυκείου

Τίτλος διδακτικής παρέμβασης

Εντολές αναμονής και Δομή Επιλογής

Γνωστικό αντικείμενο και εμπλεκόμενες γνωστικές περιοχές

Πληροφορική Γ' Γυμνασίου. Διδακτική Ενότητα: 1.2 :

Ο Προγραμματισμός στην πράξη

Τάξη εφαρμογής

Γ' Γυμνασίου (Νέα πιλοτικά προγράμματα σπουδών)

Συμβατότητα και συσχέτιση με το Αναλυτικό Πρόγραμμα

Προγραμματίζω τον υπολογιστή - Οι μαθητές σχεδιάζουν και υλοποιούν προγράμματα που απαιτούν χρήση δομής επανάληψης σύμφωνα με τα νέα πιλοτικά προγράμματα σπουδών.

Υλικοτεχνική υποδομή

Εργαστήριο πληροφορικής, βιντεοπροβολέας, πίνακας μαρκαδόρου, πακέτο Lego Mindstorms NXT Education.

Το Λογισμικό

Περιβάλλον προγραμματισμού Lego NXT-G, λογισμικό παρουσιάσεων Powerpoint

Σκοπός

Να αποκτήσει ο μαθητής σχετική ευχέρεια και εμπειρία στη δεξιότητα του δομημένου προγραμματισμού.

Διδακτικοί στόχοι:

Επαφή και πρακτική εξάσκηση με τις δυνατότητες του λογισμικού NXT-G.

- Να γνωρίσουν τις εντολές εισόδου για τον έλεγχο της κατασκευής με τη χρήση αισθητήρων.
- Να αποκτήσουν ευχέρεια στον προγραμματισμό ενός ρομπότ.
- Να γνωρίσουν την προγραμματιστική δομή της επιλογής.

Γενικότεροι μαθησιακοί στόχοι

- να αντιλαμβάνονται την αναγκαιότητα και τη χρησιμότητα της δομής επιλογής
- να εφαρμόζουν τεχνικές ελέγχου και διόρθωσης σφαλμάτων στα προγράμματα που δημιουργούν.

Γενικότεροι εκπαιδευτικοί στόχοι

- συνεργασία των μαθητών για εξαγωγή αποτελεσμάτων και συμπερασμάτων

Στόχοι ως προς την αξιοποίηση της τεχνολογίας

- εξοικείωση των μαθητών με τη χρήση των εντολών εισόδου από αισθητήρες και της δομής επιλογής σε περιβάλλον Lego NXT-G

Διδακτική μέθοδος

Αρχικά με διάλεξη περιορισμένου χρόνου και αντίστοιχη επίδειξη
Στη συνέχεια οι μαθητές εργάζονται σε ομάδες (κοινωνιο-επικοινωνιακή μάθηση),
έμμεση διδασκαλία – μαθητοκεντρική

Διδακτικές στρατηγικές / τεχνικές

Μορφή καθοδήγησης (θέτουμε ερωτήματα και οι μαθητές επιλέγουν βήματα και δράσεις σε πλαίσιο καθοδήγησης), πειραματισμός, διερεύνηση, ανακάλυψη, επίλυση προβλήματος.

Οργάνωση διδασκαλίας

Ομάδες των τριών ατόμων. Αναθέτουμε σε ένα μέλος της ομάδας το ρόλο του εκπροσώπου, σε ένα άλλο το ρόλο του γραμματέα ο οποίος θα κρατάει τις σημειώσεις και στο τελευταίο μέλος το ρόλο του προγραμματιστή στο NXT-G. Κάθε ομάδα έχει το δικό της υπολογιστή.

Ρόλος εκπαιδευτικού

Καθοδηγητικός. Καθοδηγούμε τη διαδικασία ώστε να αναδεικνύονται οι προϋπάρχουσες αντιλήψεις των μαθητών, οι στρατηγικές τους, προκαλούμε την αποτελεσματική συζήτηση και την αλληλεπίδραση στις ομάδες και στην ολομέλεια, παροτρύνουμε τους μαθητές που συναντούν δυσκολίες.

Εκτιμώμενη διάρκεια

2 διδακτικές ώρες

2. ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ

Το επιστημονικό περιεχόμενο

Προγραμματισμός → Δομή επιλογής

Προαπαιτούμενες γνώσεις:

1. Εισαγωγή στη Ρομποτική (βλέπε σημειώσεις: [B1- Ρομποτική](#))
2. Γνωριμία με το Πακέτο Lego Mindstorms (βλέπε σημειώσεις: [B2- Πακέτο Lego Mindstorms NXT](#))
3. Εξοικείωση με το Λογισμικό NXT-G (βλέπε σημειώσεις: [B3- Το προγραμματιστικό περιβάλλον NXT-G](#) και [Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G \(Καθηγητή\)](#))
4. Ευχέρεια στις Εντολές κίνησης και τη Δομή Επανάληψης (βλέπε σημειώσεις: [B4- Οι εντολές του NXT-G](#) και [Φύλλο Εργασιών 2 - Κινητήρες και Δομή επανάληψης \(Καθηγητή\)](#))

Περιγραφή δραστηριοτήτων

Αναλυτική παρουσίαση στο έγγραφο “[A2- Πορεία Διδασκαλίας](#)” (Επισυνάπτεται)

1. Παρουσίαση των εντολών του NXT-G για τις προγραμματιστικές δομές της επανάληψης με ταυτόχρονη προβολή της παρουσίασης “[3.1 - Εντολές του NXT-G](#)”.
2. Πρακτική εξάσκηση και εξοικείωση των μαθητών με τις εντολές αναμονής για τους αισθητήρες και τη δομή επιλογής, ακολουθώντας το Φύλλο Εργασιών “[Φύλλο Εργασιών 3 - Αισθητήρες και Δομή επιλογής](#)”

Φύλλα εργασίας

[Φύλλο Εργασιών 3 - Αισθητήρες και Δομή επιλογής](#)

Αξιολόγηση μαθητών

Αξιολόγηση της ικανότητας χρήσης της δομής επιλογής μέσα από τις δραστηριότητες οι οποίες θα υλοποιηθούν και της συμπλήρωσης του φύλλου εργασίας και αξιολόγησης.

Σενάρια Ρομποτικής με το Mindstorms NXT

Αξιολόγηση σεναρίου

Αξιολόγηση μέσα από τις ίδιες δραστηριότητες και της ικανότητας συνεργασίας των μαθητών, όπως και από τη συμπλήρωση του φύλλου αξιολόγησης.

B) Επισυναπτόμενο υλικό:

Παρουσίαση :

[3.1 - Εντολές του NXT-G](#)

Φύλλα εργασιών :

[Φύλλο Εργασιών 3 - Αισθητήρες και Δομή επιλογής](#)

Σημειώσεις του καθηγητή :

[A2- Πορεία Διδασκαλίας](#)

[B3- Το προγραμματιστικό περιβάλλον NXT-G](#)

[B4- Οι εντολές του NXT-G](#)

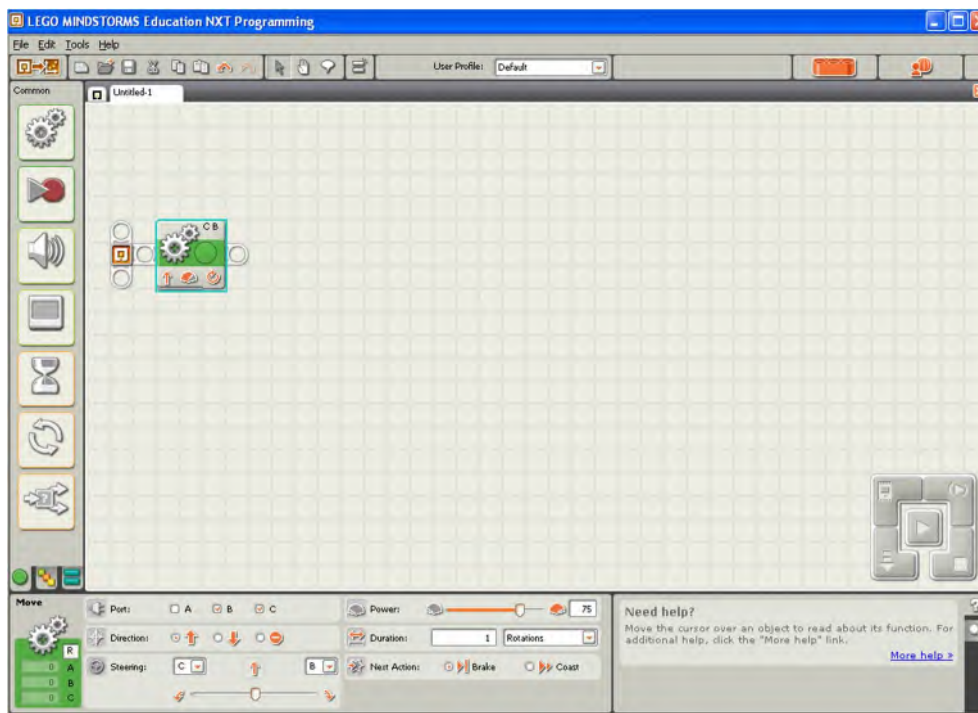
[Φύλλο Εργασιών 3 - Αισθητήρες και Δομή επιλογής \(Καθηγητή\)](#)

Έτοιμα προγράμματα :

[Φάκελος 3 - Αισθητήρες και επιλογή](#)

Γ) Αποτελέσματα –Σχόλια – Προβληματισμοί (Αναστοχασμός)

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 3^ο

Αισθητήρες και Δομή Επιλογής

Όνομα _____

Ημερομηνία _____

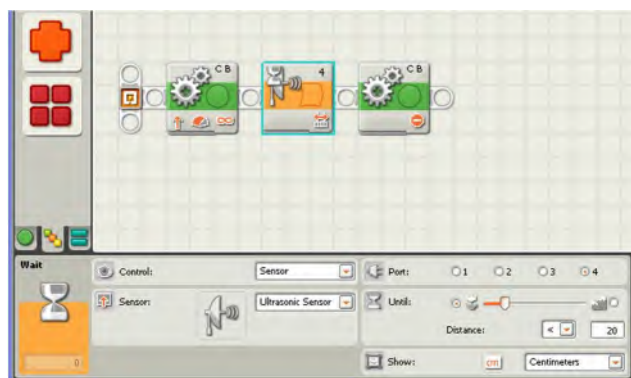
Δραστηριότητα 12 ^η : σταμάτημα με αισθητήρα υπερήχων

Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα υπερήχων.
Συνδέστε τον αισθητήρα υπερήχων στη θύρα εισόδου 4 του NXT.

Προσπαθήστε να δημιουργήσετε ένα πρόγραμμα, ώστε το όχημα-ρομπότ μας να κινείται συνέχεια μέχρι να συναντήσει κάποιο εμπόδιο στα 20 cm μπροστά του οπότε και να σταματήσει.

Για το σκοπό αυτό θα δώσουμε πρώτα στο ρομπότ μας την εντολή να ξεκινήσει να κινείται επ'άοριστο "Unlimited" αφού δεν ξέρουμε εκ των προτέρων πότε θα συναντήσει κάποιο εμπόδιο. Στη συνέχεια για να είναι σε θέση το ρομπότ μας να αντιληφθεί ένα εμπόδιο μπροστά του, θα τοποθετήσουμε στο πρόγραμμα μας μία εντολή αναμονής με αισθητήρα υπερήχων.

Υπόδειξη : Η εντολή αναμονής με αισθητήρα υπερήχων βρίσκεται στην υποπαλέτα ροής **flow**. Στον πίνακα ρυθμίσεων της, αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα υπερήχων "Ultrasonic Sensor" από αισθητήρα αφής "Touch Sensor".



- Αποθηκεύστε το πρόγραμμά σας ως **12** και τρέξτε το.

Είχε το όχημά μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

ΟΧΙ

Δραστηριότητα 13 ^η : εκκίνηση με παλαμάκια και σταμάτημα με αισθητήρα υπερήχων

Θέλουμε στο πρόγραμμα που δημιουργήσαμε στην προηγούμενη δραστηριότητα **12**, το όχημα-ρομπότ μας να μην ξεκινάει αμέσως μόλις τρέξουμε το πρόγραμμα αλλά να περιμένει πρώτα από εμάς να κτυπήσουμε παλαμάκια.

Ποια εντολή αναμονής πρέπει να χρησιμοποιήσετε ώστε το όχημα-ρομπότ να καταλάβει πότε θα κτυπήσετε παλαμάκια ;

Τοποθετείστε στο όχημα - ρομπότ και έναν αισθητήρα ήχου.
Συνδέστε τον αισθητήρα ήχου στη θύρα εισόδου 2 του NXT.

- Ανοίξτε το πρόγραμμα που δημιουργήσατε στην προηγούμενη δραστηριότητα **12** και τροποποιήστε το κατάλληλα, προσθέτοντας στην αρχή την κατάλληλη εντολή.
- Αποθηκεύστε το πρόγραμμα σας ως **13** και τρέξτε το.

Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

ΟΧΙ

Δραστηριότητα 14 ^η : ξεκίνημα με αισθητήρα ήχου και σταμάτημα με αισθητήρα υπερήχων και επανάληψη

Στη δραστηριότητα **13** προγραμματίσαμε το ρομπότ μας να ξεκινήσει όταν χτυπήσουμε παλαμάκια και να σταματήσει όταν πλησιάσει σε ένα εμπόδιο (πχ βάλουμε το χέρι μας μπροστά από τον αισθητήρα υπερήχων). Τις ενέργειες αυτές το ρομπότ θα τις εκτελέσει μία φορά και το πρόγραμμα θα τερματιστεί.

Τι πρέπει να κάνουμε για να υιοθετήσει το ρομπότ μας τη συγκεκριμένη συμπεριφορά μόνιμα, ώστε να αντιδρά κάθε φορά στα ερεθίσματα που θα δέχεται με τον κατάλληλο τρόπο και όχι μόνο μία και μοναδική φορά ;

Σενάρια Ρομποτικής με το Mindstorms NXT

- Ανοίξτε το πρόγραμμα που δημιουργήσατε στην προηγούμενη δραστηριότητα **13** και τροποποιήστε το κατάλληλα.
- Αποθηκεύστε το πρόγραμμα σας ως **14** και τρέξτε το.

Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

ΟΧΙ

Δραστηριότητα 15 ^η : ξεκίνημα και σταμάτημα με αισθητήρα αφής και επανάληψη

Τοποθετείστε στο όχημα - ρομπότ έναν αισθητήρα αφής.

Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

Θέλουμε στο πρόγραμμα που δημιουργήσαμε στην προηγούμενη δραστηριότητα **14**, το όχημα-ρομπότ μας να κινείται προς τα εμπρός όταν πατιέται ο διακόπτης του αισθητήρα αφής και να σταματάει όταν ο διακόπτης του αισθητήρα αφής απελευθερώνεται.

- Ανοίξτε το πρόγραμμα που δημιουργήσατε στην προηγούμενη δραστηριότητα **14** και τροποποιήστε το κατάλληλα.
- Αποθηκεύστε το πρόγραμμα σας ως **15** και τρέξτε το.

Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

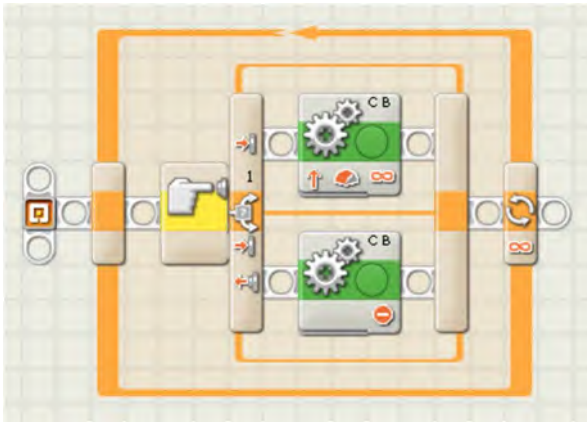
ΟΧΙ

Δραστηριότητα 16 ^η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη

Τοποθετείστε στο όχημα - ρομπότ έναν αισθητήρα αφής.

Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

Ποια νομίζετε ότι θα είναι η συμπεριφορά που θα αποκτήσει το όχημα - ρομπότ μας, αν φορτώσουμε το παρακάτω πρόγραμμα :



- Υλοποιήστε το και τρέξτε το. Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;
ΝΑΙ
ΟΧΙ
- Αποθηκεύστε το πρόγραμμα σας ως **16**.

Δραστηριότητα 17 ^η : άναμμα φώτων κάθε φορά που σκοτεινιάζει

Τοποθετείστε στο όχημα - ρομπότ ένα φωτάκι μπροστά.
Συνδέστε το φωτάκι στη θύρα εξόδου A του NXT.

Θέλουμε το όχημα-ρομπότ μας να ανάβει τα φώτα κάθε φορά που σκοτεινιάζει και να τα σβήνει μόλις φωτίζει.

Ποια εντολή αναμονής πρέπει να χρησιμοποιήσετε ώστε το όχημα-ρομπότ να καταλαβαίνει τότε σκοτεινιάζει ;

Σενάρια Ρομποτικής με το Mindstorms NXT

Τοποθετείστε στο όχημα - ρομπότ έναν αισθητήρα φωτός.

Συνδέστε τον αισθητήρα φωτός στη θύρα εισόδου 3 του NXT.

- Δημιουργήστε το κατάλληλο πρόγραμμα από την αρχή ώστε το όχημα μας να αποκτήσει την επιθυμητή συμπεριφορά.

- Αποθηκεύστε το πρόγραμμα σας ως **17** και τρέξτε το.

Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

ΝΑΙ

ΟΧΙ

Σημείωση : Στο πρόγραμμα αυτό, μας ενδιαφέρει το επίπεδο του φωτός από το περιβάλλον. Για το λόγο αυτό στη ρύθμιση της λειτουργίας (*Function*) του αισθητήρα φωτός απενεργοποιούμε την επιλογή για την παραγωγή υπέρυθρης ακτινοβολίας "*Generate light* " αφού δε χρειαζόμαστε ο αισθητήρας φωτός να εκπέμπει τη δική του υπέρυθρη ακτινοβολία.

Δραστηριότητα 18 ^η : *άναμμα φώτων κάθε φορά που σκοτεινιάζει με διακλάδωση αισθητήρα φωτός*

- Υλοποιείτε τη συμπεριφορά του οχήματος - ρομπότ της προηγούμενης δραστηριότητας **17**, χρησιμοποιώντας αντί για εντολές αναμονής αισθητήρα φωτός, μία εντολή διακλάδωσης αισθητήρα φωτός.

- Αποθηκεύστε το πρόγραμμα σας ως **18** και τρέξτε το.

Είχε το όχημα μας τη συμπεριφορά που περιμένατε ;

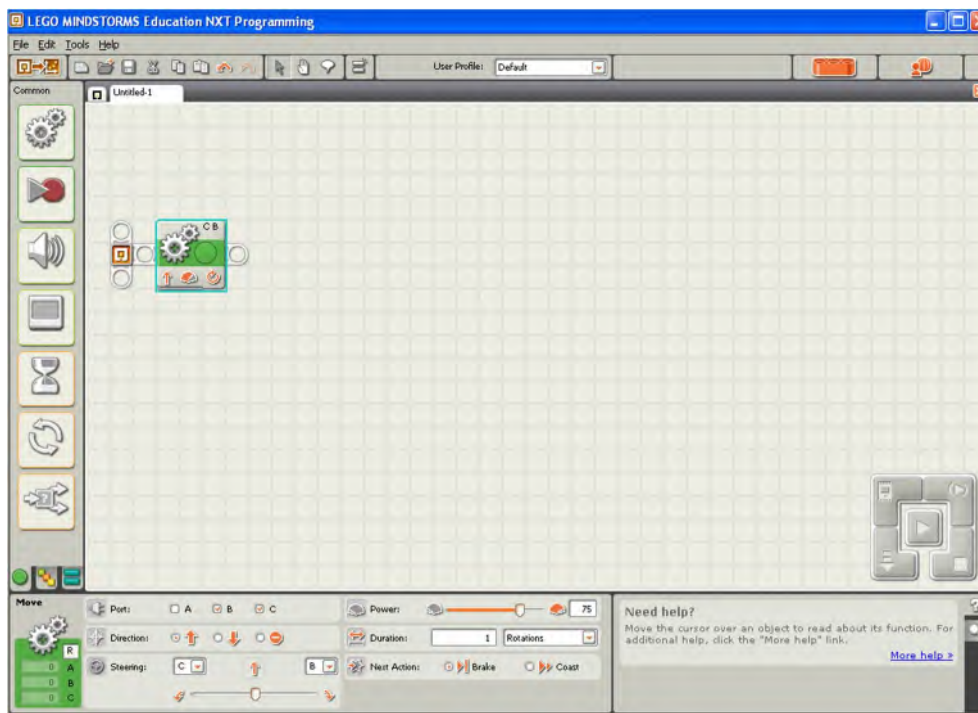
ΝΑΙ

ΟΧΙ

Συγχαρητήρια ☺

Ολοκληρώσατε το 3^ο φύλλο δραστηριοτήτων!!!

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 3^ο Αισθητήρες και Δομή Επιλογής

Σημειώσεις Καθηγητή

Δραστηριότητα 12 ^η : σταμάτημα με αισθητήρα υπερήχων

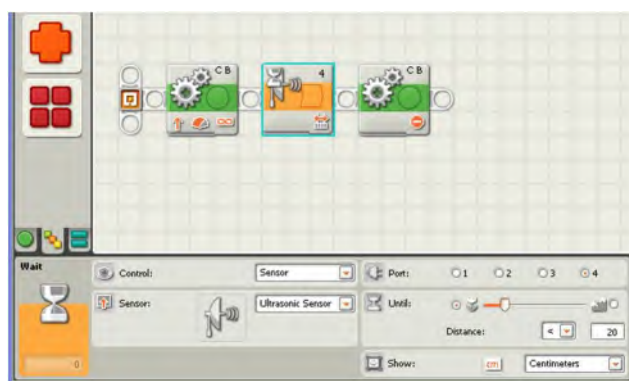
Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα υπερήχων. Συνδέστε τον αισθητήρα υπερήχων στη θύρα εισόδου 4 του NXT.

Θέλουμε το όχημα-ρομπότ μας να κινείται μέχρι να συναντήσει κάποιο εμπόδιο στα 20 cm μπροστά του οπότε και να σταματήσει.

Για το σκοπό αυτό θα δώσουμε πρώτα στο ρομπότ μας την εντολή να ξεκινήσει να κινείται επ'άοριστο "Unlimited" αφού δεν ξέρουμε εκ των προτέρων πότε θα συναντήσει κάποιο εμπόδιο. Στη συνέχεια για να είναι σε θέση το ρομπότ μας να αντιληφθεί ένα εμπόδιο μπροστά του, θα τοποθετήσουμε στο πρόγραμμα μας μία εντολή αναμονής με αισθητήρα υπερήχων.

Υπόδειξη : Η εντολή αναμονής με αισθητήρα υπερήχων βρίσκεται στην υποπαλέτα ροής **flow**. Στον πίνακα ρυθμίσεων της, αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα υπερήχων "Ultrasonic Sensor" από αισθητήρα αφής "Touch Sensor".

Μπορούμε να παρατηρήσουμε ότι το πρόγραμμα θεωρεί ότι ο αισθητήρας μας είναι συνδεδεμένος στη θύρα εισόδου 4, στην οποία ενδείκνυται να τοποθετούμε έναν αισθητήρα υπερήχων για να μην μπερδευόμαστε. Αν για κάποιο λόγο χρειαστεί να συνδέσουμε τον αισθητήρα αυτόν σε μία άλλη θύρα, θα πρέπει να τροποποιήσουμε τη ρύθμιση αυτή αντίστοιχα. Στην επιλογή (Until) για τη συνθήκη που θα πρέπει εκπληρωθεί ώστε να τερματιστεί η αναμονή από το ρομπότ μας, πληκτρολογούμε την αριθμητική τιμή 20, αφού πρώτα έχουμε επιλέξει ως μονάδα μέτρησης (Show) τα εκατοστά του μέτρου "Centimeters" αντί για τις ίντσες "Inches". Τέλος για να πούμε στο ρομπότ να σταματήσει μόλις ικανοποιηθεί η συνθήκη που καθορίσαμε προηγουμένως, τοποθετούμε μία εντολή μετακίνησης με επιλεγμένη τη ρύθμιση στοπ στον πίνακα ρυθμίσεων της.



Σημείωση : Αν δεν τοποθετήσουμε πριν από το τέλος του προγράμματος μας την εντολή για το σταμάτημα των κινητήρων, το ρομπότ μας θα σταματήσει να κινείται ούτως ή άλλως, αφού όπως έχουμε αναφέρει με τον τερματισμό ενός προγράμματος το NXT διακόπτει την τροφοδοσία σε όλες τις θύρες εξόδου του. Το ρομπότ μας όμως δε θα σταματήσει ακαριαία αλλά θα συνεχίσει για ένα πολύ μικρό διάστημα να ολισθαίνει.

Δραστηριότητα 13 ^η : εκκίνηση με αισθητήρα ήχου (παλαμάκια) και σταμάτημα με αισθητήρα υπερήχων

Τοποθετείστε στο όχημα - ρομπότ και έναν αισθητήρα ήχου.

Συνδέστε τον αισθητήρα ήχου στη θύρα εισόδου 2 του NXT.

Θέλουμε στο πρόγραμμα που δημιουργήσαμε στην προηγούμενη δραστηριότητα 12, το όχημα-ρομπότ μας να μην ξεκινάει αμέσως μόλις τρέξουμε το πρόγραμμα αλλά να περιμένει πρώτα από εμάς, να κτυπήσουμε παλαμάκια.

Ποια εντολή αναμονής πρέπει να χρησιμοποιήσετε ώστε το όχημα-ρομπότ να καταλάβει πότε θα κτυπήσετε παλαμάκια :

• Τροποποιήστε κατάλληλα το πρόγραμμα που δημιουργήσατε στην προηγούμενη δραστηριότητα 12, προσθέτοντας στην αρχή του την κατάλληλη εντολή.

Απάντηση :

Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε μία εντολή αναμονής με αισθητήρα ήχου. Την εντολή αυτή την τοποθετούμε στην αρχή του προγράμματος έτσι ώστε το ρομπότ να περιμένει από εμάς να κτυπήσουμε παλαμάκια προτού εκτελέσει το υπόλοιπο μέρος του προγράμματος.

Ο αισθητήρας ήχου ανιχνεύει τη στάθμη του ήχου που παράγεται κάθε χρονική στιγμή στο περιβάλλον που δραστηριοποιείται και τη μετατρέπει σε μία αντίστοιχη αριθμητική τιμή, την οποία και στέλνει στο NXT μέσω της θύρας εισόδου που είναι συνδεδεμένος.

Υπόδειξη : Πριν χρησιμοποιήσουμε έναν αισθητήρα ήχου (τα ίδια ισχύουν και για τον αισθητήρα φωτός όπως θα δούμε παρακάτω) πρέπει να κάνουμε μία σειρά από μετρήσεις με σκοπό να πληροφορηθούμε σε ποιες αριθμητικές τιμές αντιστοιχούν οι διάφοροι ήχοι που θα συναντήσει το ρομπότ μας στο συγκεκριμένο περιβάλλον που θα δραστηριοποιηθεί.

Στη συνέχεια πρέπει να υπολογίσουμε τη στάθμη του ήχου (κατώφλι) πάνω ή κάτω από την οποία μας ενδιαφέρει το ρομπότ μας να αλλάζει συμπεριφορά.

Μπορούμε να λάβουμε μία εκτίμηση του σημείου αυτού μετρώντας τις τιμές σε δύο ακραίες καταστάσεις και υπολογίζοντας το μέσο όρο τους. Έτσι αν θέλουμε να το ρομπότ μας να διακρίνει έναν απαλό από έναν δυνατό ήχο θα πρέπει να κάνουμε μία μέτρηση όταν επικρατεί ησυχία και μία άλλη μέτρηση όταν έχουμε αυξημένο θόρυβο. Για να προσδιορίσουμε στη συνέχεια την τιμή για το κατώφλι μας στη συγκεκριμένη περίπτωση, υπολογίζουμε το μέσο όρο αυτών των τιμών. Ήχος κάτω από την τιμή αυτή θα εκλαμβάνεται από το ρομπότ μας ως απαλός ήχος και πάνω από την τιμή αυτή ως δυνατός.

Ενδεικτικά αναφέρουμε τη στάθμη που κατά προσέγγιση παράγουν οι παρακάτω δραστηριότητες :

Ησυχία 0%

Θόρυβος σε δωμάτιο 6%

Ομιλίες 45%

Χειροκρότημα 70%

Δυνατός θόρυβος 100%

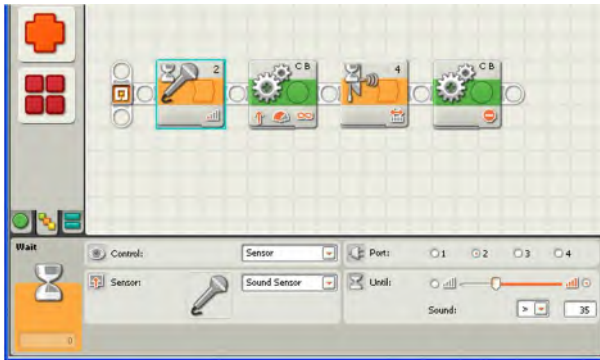
Για να κάνουμε μία μέτρηση για τη στάθμη ενός ήχου χρησιμοποιούμε την επιλογή **View** από το μενού του επεξεργαστή NXT.

Στη δική μας περίπτωση, μας ενδιαφέρει το ρομπότ μας να διακρίνει πότε θα ακουστεί ένας ήχος με τη στάθμη που παράγεται, όταν κτυπήσουμε παλαμάκια με τα χέρια μας. Στο παράδειγμα μας εκτελέσαμε μία μέτρηση χωρίς να κάνουμε κάποιο θόρυβο και μία μέτρηση τη στιγμή που χτυπήσαμε παλαμάκια. Από τις μετρήσεις αυτές λάβαμε τις αριθμητικές τιμές 5 και 65. Στο πρόγραμμα μας λοιπόν θα

Σενάρια Ρομποτικής με το Mindstorms NXT

χρησιμοποιήσουμε ως κατώφλι το μέσο όρο $5+65/2=35$. Οποιοσδήποτε θόρυβος της τάξης του 35% και άνω θα εκλαμβάνεται από το ρομπότ μας ως χειροκρότημα.

Η εντολή για την αναμονή με αισθητήρα ήχου βρίσκεται στην υποπαλέτα ροής **flow**. Την εντολή αυτήν την τοποθετούμε στην αρχή του προγράμματος της δραστηριότητας **12**.



Στον πίνακα ρυθμίσεων της αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα ήχου "Sound Sensor" από αισθητήρα αφής "Touch Sensor". Μπορούμε να παρατηρήσουμε ότι το πρόγραμμα θεωρεί ότι ο αισθητήρας μας είναι συνδεδεμένος στη θύρα εισόδου 2, στην οποία ενδείκνυται να τοποθετούμε έναν αισθητήρα ήχου για να μην μπερδευόμαστε. Στην επιλογή (Until) για τη συνθήκη που θα πρέπει εκπληρωθεί ώστε να τερματιστεί η αναμονή από το ρομπότ μας, πληκτρολογούμε την αριθμητική τιμή 35.

Υπόδειξη : Αν θέλουμε το ρομπότ μας να ξεκινήσει όταν θα έχουν ολοκληρωθεί τα παλαμάκια, πρέπει να τοποθετήσουμε ακόμη μία εντολή αναμονής για ήχο αμέσως μετά την πρώτη, στην οποία ορίζουμε ως κατώφλι μία πολύ χαμηλή τιμή.



Σημείωση : Προτού χρησιμοποιήσουμε στο πρόγραμμα μας ένα αισθητήρα ήχου (το ίδιο ισχύει και για τον αισθητήρα φωτός) πρέπει να κάνουμε βαθμονόμηση του αισθητήρα ώστε να τον προσαρμόζουμε στις συνθήκες που επικρατούν στο συγκεκριμένο περιβάλλον που το ρομπότ μας πρόκειται να δραστηριοποιηθεί.

Η προσαρμογή αυτή γίνεται με τη χρήση δύο εντολών καλιμπραρίσματος. Ένα ενδεικτικό πρόγραμμα για την εργασία αυτή είναι το παρακάτω :



Πρέπει να κατεβάσετε το πρόγραμμα αυτό στο NXT και να το τρέξετε.

Για το NXT ο χαμηλότερος θόρυβος ισούται τώρα με 0 και η φασαρία με 100.

Δραστηριότητα 14 ^η : ξεκίνημα με αισθητήρα ήχου και σταμάτημα με αισθητήρα υπερήχων και επανάληψη

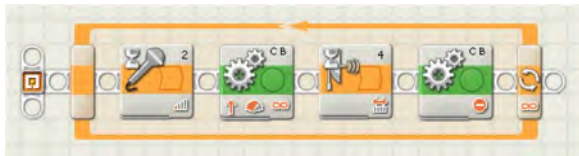
Στη δραστηριότητα 13 προγραμματίσαμε το ρομπότ μας να ξεκινήσει όταν χτυπήσουμε παλαμάκια και να σταματήσει όταν πλησιάσει σε ένα εμπόδιο (πχ βάλουμε το χέρι μας μπροστά από τον αισθητήρα υπερήχων). Τις ενέργειες αυτές το ρομπότ θα τις εκτελέσει μία φορά και το πρόγραμμα θα τερματιστεί.

Τι πρέπει να κάνουμε για να υιοθετήσει το ρομπότ μας τη συγκεκριμένη συμπεριφορά μόνιμα, ώστε να αντιδρά κάθε φορά στα ερεθίσματα που θα δέχεται με τον κατάλληλο τρόπο και όχι μόνο μία και μοναδική φορά ;

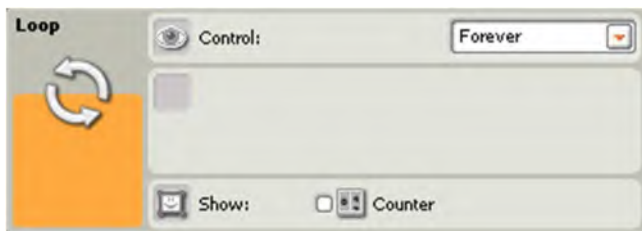
Απάντηση : Πρέπει να τοποθετήσουμε τις ενέργειες που επιθυμούμε να επαναληφθούν στο εσωτερικό μιας εντολής επανάληψης

Το εικονίδιο για τη δομή επανάληψης βρίσκεται στο υπομενού για δομές **Flow** και έχει ως εικόνα δύο βέλη τα οποία σχηματίζουν ένα κύκλο. Μέσα στο εικονίδιο της εντολής επανάληψης τοποθετούμε τις εντολές που είχαμε χρησιμοποιήσει δραστηριότητα 13.

Υπόδειξη : Ένας βολικός τρόπος είναι να ανοίξουμε τη δραστηριότητα 13 και αφού εισάγουμε μία εντολή επανάληψης στην αρχή του προγράμματος να επιλέξουμε όλα τα εικονίδια των εντολών και να τα σύρουμε μέσα στο εσωτερικό του εικονιδίου της εντολής επανάληψης.



Στον πίνακα ρυθμίσεων της εντολής αυτής δεν χρειάζεται να πειράζουμε καμία ρύθμιση.



Με το πρόγραμμα αυτό κάθε φορά που το ρομπότ μας είναι ακίνητο και κτυπάμε παλαμάκια θα ξεκινάει να κινείται και κάθε φορά που κινείται και τοποθετούμε το χέρι μας μπροστά στον αισθητήρα υπερήχων θα σταματάει. Η εκτέλεση του προγράμματος αυτού δε θα σταματήσει παρά μόνο όταν εμείς θελήσουμε να διακόψουμε το πρόγραμμα αυτό.

Δραστηριότητα 15^η : ξεκίνημα και σταμάτημα με αισθητήρα αφής και επανάληψη

Τοποθετείστε στο όχημα - ρομπότ έναν αισθητήρα αφής.
Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

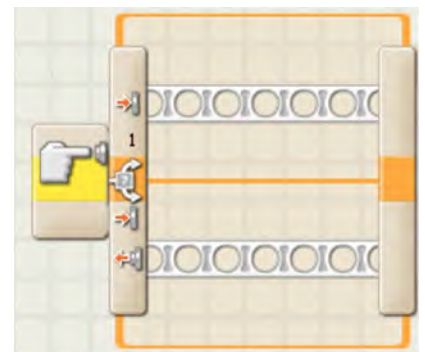
Θέλουμε στο πρόγραμμα που δημιουργήσαμε στην προηγούμενη δραστηριότητα **14**, το όχημα-ρομπότ μας να κινείται προς τα εμπρός όταν πατιέται ο διακόπτης του αισθητήρα αφής και να σταματάει όταν ο διακόπτης του αισθητήρα αφής απελευθερώνεται.



Σημείωση : Το όχημα μας κινείται μόνο κάθε φορά που πατάμε το διακόπτη του αισθητήρα αφής και για όσο διάστημα τον κρατάμε πατημένο.

Λίγα λόγια για τη δομή επιλογής

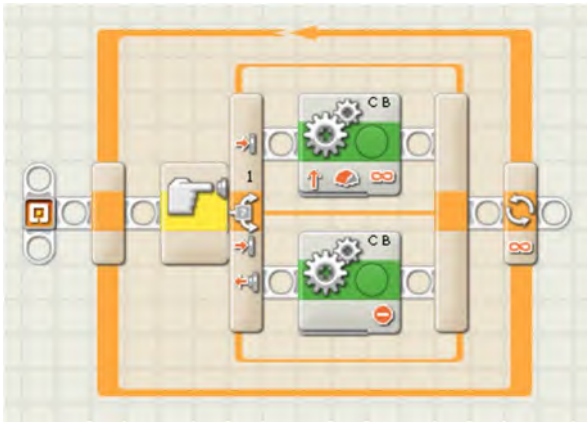
Συχνά μέσα σε ένα πρόγραμμα χρειάζεται να πούμε στο ρομπότ μας τι να κάνει όταν συναντήσει μία μελλοντική κατάσταση, την οποία δεν μπορούμε να ξέρουμε από πριν αν θα τη συναντήσει και πότε. Πχ να πέσει πάνω σε ένα εμπόδιο. Σε αυτήν την περίπτωση θα πρέπει να έχουμε προγραμματίσει το ρομπότ μας κατάλληλα, ώστε να είναι σε θέση τη στιγμή εκείνη να πάρει μια απόφαση για τι θα κάνει, στηριζόμενο στην τιμή κάποιας παραμέτρου, η οποία μπορεί να είναι η τιμή ενός αισθητήρα ή ένα χρονικό διάστημα.



Δραστηριότητα 16^η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη

Τοποθετείστε στο όχημα - ρομπότ έναν αισθητήρα αφής.
Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

Εναλλακτικά για το πρόβλημα της δραστηριότητας **15** μπορούμε να χρησιμοποιήσουμε αντί για δύο εντολές αναμονής με αισθητήρα αφής, μία εντολή διακλάδωσης αισθητήρα αφής.



Δραστηριότητα 17^η : άναμμα φώτων κάθε φορά που σκοτεινιάζει

Τοποθετείστε στο όχημα - ρομπότ ένα φωτάκι μπροστά.

Συνδέστε το φωτάκι στη θύρα εξόδου Α του NXT.

Τοποθετείστε στο όχημα - ρομπότ έναν αισθητήρα φωτός.

Συνδέστε τον αισθητήρα φωτός στη θύρα εισόδου 3 του NXT.

Θέλουμε το όχημα-ρομπότ μας να ανάβει τα φώτα κάθε φορά που σκοτεινιάζει και να τα σβήνει μόλις φωτίζει.

Ποια εντολή αναμονής πρέπει να χρησιμοποιήσετε ώστε το όχημα-ρομπότ να καταλάβει πότε έχει σκοτεινιάσει ;

Απάντηση :

Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε μία εντολή αναμονής με αισθητήρα φωτός.

Ο αισθητήρας φωτός ανιχνεύει την ποσότητα του περιβάλλοντος φωτός ή εναλλακτικά το φως που ανακλάται από μία επιφάνεια. Αναφέρει αυτή την ποσότητα στο NXT με μία αριθμητική τιμή από το 0 (απόλυτο σκοτάδι) ως το 100 (άπλετο φως). Μπορούμε να τον χρησιμοποιήσουμε για να ξεκινάμε και να σταματάμε έναν κινητήρα ή ένα λαμπτήρα.

Υπόδειξη : Πριν χρησιμοποιήσουμε έναν αισθητήρα φωτός πρέπει να κάνουμε μία σειρά από μετρήσεις με σκοπό να πληροφορηθούμε σε ποιες αριθμητικές τιμές αντιστοιχούν τα διάφορα επίπεδα φωτός που θα συναντήσει το ρομπότ μας στο συγκεκριμένο περιβάλλον που θα δραστηριοποιηθεί.

Στη συνέχεια πρέπει να υπολογίσουμε τη στάθμη του φωτός (κατώφλι) πάνω ή κάτω από την οποία μας ενδιαφέρει το ρομπότ μας να αλλάζει συμπεριφορά.

Μπορούμε να λάβουμε μία εκτίμηση του σημείου αυτού μετρώντας τις τιμές σε δύο ακραίες καταστάσεις και υπολογίζοντας το μέσο όρο τους. Έτσι αν θέλουμε να το ρομπότ μας να διακρίνει πότε υπάρχει επαρκές φως και πότε όχι θα πρέπει να κάνουμε δύο αντίστοιχες μετρήσεις στο χώρο που θα δραστηριοποιηθεί το ρομπότ μας και στις περιοχές με επαρκές και ελλιπές φως. Για να προσδιορίσουμε στη συνέχεια την τιμή για το κατώφλι μας, υπολογίζουμε το μέσο όρο αυτών των τιμών. Στάθμη φωτός πάνω από την τιμή αυτή θα εκλαμβάνεται από το ρομπότ μας ως κανονικό φως και κάτω από την τιμή αυτή ως σκοτάδι.

Σενάρια Ρομποτικής με το Mindstorms NXT

Για να κάνουμε μία μέτρηση για τη στάθμη ενός ήχου χρησιμοποιούμε την επιλογή View από το μενού του επεξεργαστή NXT.

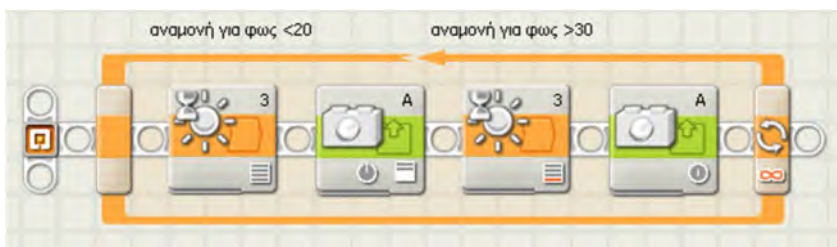
Στην περίπτωση τη δική μας, μας ενδιαφέρει το ρομπότ μας να διακρίνει πότε θα σκοτεινιάσει. Στο παράδειγμα μας εκτελέσαμε μία μέτρηση όταν έχουμε κανονικό φως δωματίου και μία άλλη όταν σκοτεινιάσει. Από τις μετρήσεις αυτές λάβαμε τις αριθμητικές τιμές 55 και 35. Στο πρόγραμμα μας λοιπόν θα χρησιμοποιήσουμε ως κατώφλι για το σκοτάδι το $55+35/2=40$. Οποιοσδήποτε τιμή της τάξης του 40 και κάτω θα εκλαμβάνεται από το ρομπότ μας ως σκοτάδι.

Η εντολή για την αναμονή με αισθητήρα φωτός βρίσκεται στην υποπαλέτα ροής **flow**. Την εντολή αυτήν την τοποθετούμε μετά από μία εντολή μετακίνησης του ρομπότ μας.

Στον πίνακα ρυθμίσεων της αλλάζουμε την επιλογή του αισθητήρα (Sensor) σε αισθητήρα φωτός "Light Sensor" από αισθητήρα αφής "Touch Sensor". Μπορούμε να παρατηρήσουμε ότι το πρόγραμμα θεωρεί ότι ο αισθητήρας μας είναι συνδεδεμένος στη θύρα εισόδου 3, στην οποία ενδείκνυται να τοποθετούμε έναν αισθητήρα φωτός για να μην μπερδευόμαστε.

Σημείωση: Στο πρόγραμμα αυτό, μας ενδιαφέρει το επίπεδο του φωτός από το περιβάλλον. Για το λόγο αυτό στη ρύθμιση της λειτουργίας (Function) του αισθητήρα φωτός απενεργοποιούμε την επιλογή για την παραγωγή υπέρυθρης ακτινοβολίας "Generate light" αφού δε χρειαζόμαστε ο αισθητήρας φωτός να εκπέμπει τη δική του υπέρυθρη ακτινοβολία.

Στην επιλογή (Until) για τη συνθήκη που θα πρέπει να εκπληρωθεί ώστε να τερματιστεί η αναμονή από το ρομπότ μας πληκτρολογούμε την αριθμητική τιμή 40 και αλλάζουμε την κατεύθυνση της ανισότητας σε μικρότερο από "<" αντί του μεγαλύτερο από ">". Στη συνέχεια αφού τοποθετήσουμε μία εντολή λαμπτήρα για το άναμμα των φώτων, για να προσδιορίσουμε για πόσο διάστημα θα μείνουν τα φώτα αναμμένα, τοποθετούμε άλλη μία εντολή αναμονής με αισθητήρα φωτός στον πίνακα ρυθμίσεων της οποίας αυτή τη φορά ρυθμίζουμε τη συνθήκη για φως μεγαλύτερο από ">" 40. Με αυτό τον τρόπο διασφαλίζουμε ότι όσο το ρομπότ μας βρίσκεται μέσα στο τούνελ το φως θα παραμένει ανοικτό. Αμέσως μετά την εντολή της αναμονής, τοποθετούμε μία εντολή λαμπτήρα στον πίνακα ρυθμίσεων της οποίας ορίζουμε ως ενέργεια (Action) το "Off" για το σβήσιμο των φώτων.



Σημείωση: Προτού χρησιμοποιήσουμε στο πρόγραμμα μας ένα αισθητήρα φωτός (το ίδιο ισχύει και για τον αισθητήρα ήχου) πρέπει να κάνουμε βαθμονόμηση του αισθητήρα ώστε να τον προσαρμόζουμε στις συνθήκες που επικρατούν στο συγκεκριμένο περιβάλλον που το ρομπότ μας πρόκειται να δραστηριοποιηθεί.

Η προσαρμογή αυτή γίνεται με τη χρήση δύο εντολών καλιμπραρίσματος. Ένα ενδεικτικό πρόγραμμα για την εργασία αυτή είναι το παρακάτω :



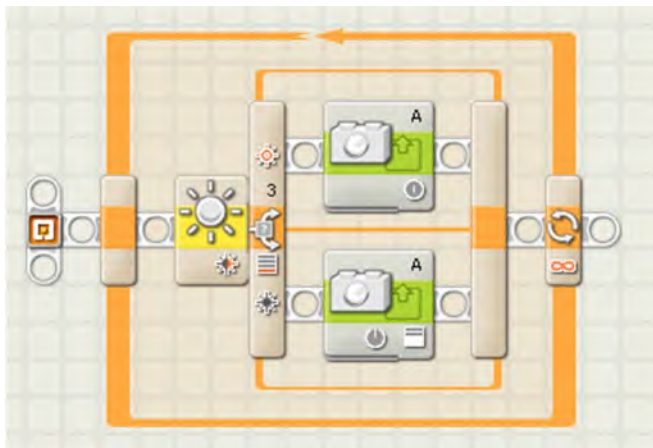
Σενάρια Ρομποτικής με το Mindstorms NXT

Πρέπει να κατεβάσετε το πρόγραμμα αυτό στο NXT και να το τρέξετε.

Για το NXT το πιο σκοτεινό σημείο ισούται τώρα με 0 και το πιο φωτεινό με 100.

Δραστηριότητα 18^η : *άναμμα φώτων κάθε φορά που σκοτεινιάζει με διακλάδωση αισθητήρα φωτός*

Εναλλακτικά για το πρόβλημα της δραστηριότητας **17** μπορούμε να χρησιμοποιήσουμε αντί για δύο εντολές αναμονής αισθητήρα φωτός, μία εντολή διακλάδωσης αισθητήρα φωτός.



ΓΝΩΣΤΙΚΟ ΑΝΤΙΚΕΙΜΕΝΟ: ΠΛΗΡΟΦΟΡΙΚΗ
**ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
ΜΕ LEGO MINDSTORMS NXT**

A) Περιγραφή Διδακτικής Παρέμβασης στην τάξη – με χρήση ΤΠΕ

Δημιουργός σεναρίου: Γεώργιος Κυριακού

Ημ/νίες εφαρμογής:

Τάξεις/Τμήματα: Γ' Γυμνασίου / Α' Λυκείου

Τίτλος διδακτικής παρέμβασης

Σύνθετη δραστηριότητα "έξοδος από σπηλιά"

Γνωστικό αντικείμενο και εμπλεκόμενες γνωστικές περιοχές

Πληροφορική Γ' Γυμνασίου. Διδακτική Ενότητα: 1.2 :

Ο Προγραμματισμός στην πράξη

Τάξη εφαρμογής

Γ' Γυμνασίου (Νέα πιλοτικά προγράμματα σπουδών)

Συμβατότητα και συσχέτιση με το Αναλυτικό Πρόγραμμα

Προγραμματίζω τον υπολογιστή - Οι μαθητές σχεδιάζουν και υλοποιούν προγράμματα για να επιλύσουν πραγματικά προβλήματα σύμφωνα με τα νέα πιλοτικά προγράμματα σπουδών.

Υλικοτεχνική υποδομή

Εργαστήριο πληροφορικής, πίνακας μαρκαδόρου, πακέτο Lego Mindstorms NXT Education.

Το Λογισμικό

Περιβάλλον προγραμματισμού Lego NXT-G.

Σκοπός

Να αποκτήσει ο μαθητής σχετική ευχέρεια και εμπειρία στη δεξιότητα του δομημένου προγραμματισμού.

Διδακτικοί στόχοι:

- Να ολοκληρώσουν μία σειρά από προκλήσεις, σταδιακά αυξανόμενης δυσκολίας, οι οποίες απαιτούν γνώσεις μηχανικής, μαθηματικών και προγραμματιστικές δεξιότητες.
- Να κατανοήσουν το πλεονέκτημα της εργασίας με ομαδικό πνεύμα το οποίο περιλαμβάνει καταιγισμό ιδεών, σχεδίαση, υλοποίηση, δοκιμή και συνεννόηση για το τελικό αποτέλεσμα.
- Να αφομοιώσουν τις συστάσεις και την ανατροφοδότηση που δίνεται από τον καθοδηγητή ώστε να είναι σε θέση να αντιμετωπίσουν πιο πολύπλοκα θέματα.

Γενικότεροι μαθησιακοί στόχοι

- να αναπτύξουν την αναλυτική και κριτική και συνθετική ικανότητα.
- να εφαρμόζουν τεχνικές ελέγχου και διόρθωσης σφαλμάτων στα προγράμματα που δημιουργούν.

Γενικότεροι εκπαιδευτικοί στόχοι

- συνεργασία των μαθητών για εξαγωγή αποτελεσμάτων και συμπερασμάτων

Στόχοι ως προς την αξιοποίηση της τεχνολογίας

- αντίληψη των μαθητών σχετικά με τη ρομποτική.

Διδακτική μέθοδος

Οι μαθητές εργάζονται σε ομάδες (κοινωνιο-επικοινωνιακή μάθηση), έμμεση διδασκαλία - μαθητοκεντρική

Διδακτικές στρατηγικές / τεχνικές

Μορφή καθοδήγησης (θέτουμε ερωτήματα και οι μαθητές επιλέγουν βήματα και δράσεις σε πλαίσιο καθοδήγησης), πειραματισμός, διερεύνηση, ανακάλυψη, επίλυση προβλήματος.

Οργάνωση διδασκαλίας

Ομάδες των τριών ατόμων. Αναθέτουμε σε ένα μέλος της ομάδας το ρόλο του εκπροσώπου, σε ένα άλλο το ρόλο του γραμματέα ο οποίος θα κρατάει τις σημειώσεις και στο τελευταίο μέλος το ρόλο του προγραμματιστή στο NXT-G. Κάθε ομάδα έχει το δικό της υπολογιστή. Μία από τις ομάδες θα επιλεγεί για να παίξει το παιχνίδι ρόλων .

Ρόλος εκπαιδευτικού

Συμβουλευτικός παρά καθοδηγητικός. Τονίζει συνέχεια στους μαθητές τη σημαντικότητα της ομαδικής δουλειάς και της συνεργασίας, ενισχύοντας την ευγενή άμιλλα μεταξύ των μαθητών και προσεγγίζει τα προβλήματα που δημιουργούνται ανάμεσα στα μέλη των ομάδων με κατάλληλο τρόπο ώστε να ανατροφοδοτούν την τάξη, βοηθώντας με αυτό τον τρόπο τη συνεργασία μεταξύ των παιδιών

Εκτιμώμενη διάρκεια

2 διδακτικές ώρες

2. ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ

Το επιστημονικό περιεχόμενο

Προγραμματισμός → Σύνθετα προβλήματα

Προαπαιτούμενες γνώσεις:

1. Εισαγωγή στη Ρομποτική (βλέπε σημειώσεις: [B1- Ρομποτική](#))
2. Γνωριμία με το Πακέτο Lego Mindstorms (βλέπε σημειώσεις: [B2- Πακέτο Lego Mindstorms NXT](#))
3. Εξοικείωση με το Λογισμικό NXT-G (βλέπε σημειώσεις: [B3- Το προγραμματιστικό περιβάλλον NXT-G](#) και [Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G \(Καθηγητή\)](#))
4. Ευχέρεια στις Εντολές κίνησης και τη Δομή Επανάληψης (βλέπε σημειώσεις: [B4- Οι εντολές του NXT-G](#) και [Φύλλο Εργασιών 2 - Κινητήρες και Δομή επανάληψης \(Καθηγητή\)](#))
5. Ευχέρεια στις Εντολές ελέγχου και τη Δομή Επιλογής (βλέπε σημειώσεις: [B4- Οι εντολές του NXT-G](#) και [Φύλλο Εργασιών 3 - Αισθητήρες και Δομή επιλογής \(Καθηγητή\)](#))

Περιγραφή δραστηριοτήτων

Αναλυτική παρουσίαση στο έγγραφο “[A2- Πορεία Διδασκαλίας](#)” (Επισυνάπτεται)

1. Δραστηριότητα, βασισμένη σε σχέδιο εργασίας ακολουθώντας τις οδηγίες και καταγράφοντας την πορεία του “[Φύλλο Εργασιών 4 - Πρόκληση με αισθητήρες αφής](#)”.

Φύλλα εργασίας

[Φύλλο Εργασιών 4 - Πρόκληση με αισθητήρες αφής](#) (Επισυνάπτεται)

Αξιολόγηση μαθητών

Αξιολόγηση της ικανότητας επίλυσης προβλημάτων και υλοποίησης των αντίστοιχων προγραμμάτων μέσα από τις δραστηριότητες οι οποίες θα υλοποιηθούν.

Αξιολόγηση σεναρίου

Αξιολόγηση μέσα από τις ίδιες δραστηριότητες και της ικανότητας συνεργασίας των μαθητών, όπως και από τη συμπλήρωση των φύλλων αξιολόγησης της επόμενης διδακτικής παρέμβασης Νο 6.

B) Επισυναπτόμενο υλικό:

Φύλλα εργασιών :

[Φύλλο Εργασιών 4 - Πρόκληση με αισθητήρες αφής](#)

Σημειώσεις του καθηγητή :

[A2- Πορεία Διδασκαλίας](#)

[B3- Το προγραμματιστικό περιβάλλον NXT-G](#)

[B4- Οι εντολές του NXT-G](#)

[Φύλλο Εργασιών 4 - Πρόκληση με αισθητήρες αφής \(Καθηγητή\)](#)

Φύλλο Αξιολόγησης μαθητών :

Στο τέλος και της επόμενης διδακτικής παρέμβασης Νο 6

Έτοιμα προγράμματα :

[Φάκελος 4 - Πρόκληση με αισθητήρες αφής](#)

Γ) Αποτελέσματα –Σχόλια – Προβληματισμοί (Αναστοχασμός

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 4^ο

Πρόκληση με αισθητήρες αφής

Όνομα _____

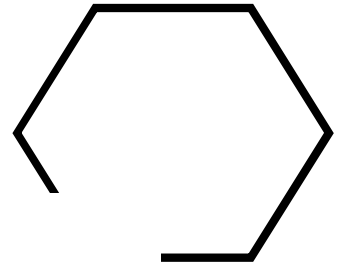
Ημερομηνία _____

Έξοδος από σπηλιά

Θα επιδιώξουμε να προγραμματίσουμε το όχημα - ρομπότ μας ώστε να είναι σε θέση να βρει την έξοδο από μία σπηλιά και να βγει από αυτήν.

Θα νικήσει η ομάδα που θα τα καταφέρει με τον καλύτερο δυνατό τρόπο.

Για να το πετύχουμε κάτι τέτοιο πρέπει να εκπαιδεύσουμε το όχημα ρομπότ μας να αποφεύγει τα εμπόδια



Δραστηριότητα 1^η : αποφυγή εμποδίου με 1 αισθητήρα αφής

Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα αφής.

Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

Στη δραστηριότητα αυτή θέλουμε να προγραμματίσουμε το όχημα - ρομπότ, ώστε να είναι σε θέση να παρακάμπτει τα εμπόδια που θα συναντήσει στην πορεία του.

Γράψτε τι ώρα είναι τώρα : _____

Ανάλυση προβλήματος :

Το όχημα - ρομπότ μας κινείται μπροστά, σε ευθύγραμμη πορεία. Με αυτό τον τρόπο κάποια στιγμή θα συναντήσει κάποιο εμπόδιο και θα παραμείνει κολλημένο στο σημείο εκείνο.

Τι θα κάνετε εσείς, αν είχατε κλειστά τα μάτια σας και έπρεπε να κινηθείτε σε ένα άγνωστο χώρο, έχοντας μόνο το ένα χέρι σας, μπροστά σας, τεντωμένο και ακίνητο ;

Λαμβάνοντας υπόψη ότι το όχημα μας διαθέτει αισθητήρα αφής στο μπροστινό του μέρος, 'είναι σε θέση να καταλάβει' ότι έχει συναντήσει ένα εμπόδιο στην πορεία του.

Σενάρια Ρομποτικής με το Mindstorms NXT

Σκεφτείτε και περιγράψτε τον τρόπο, με τον οποίο το ρομπότ μας θα μπορέσει να προσπεράσει το εμπόδιο αυτό και να συνεχίσει την πορεία του.

Συμπληρώστε τον αλγόριθμο για να μπορεί το ρομπότ μας να παρακάμπτει τα εμπόδια που θα συναντήσει στην πορεία του.

Κίνηση κινητήρα Β και C μπροστά (Ξεκίνα να κινείσαι μπροστά)
Αν διακόπτης αφής = 1 Τότε (Αν πέσεις πάνω σε ένα εμπόδιο τότε)

Γράψτε τι ώρα είναι τώρα : _____

- Ανοίξτε το προγραμματιστικό περιβάλλον NXT-G και δημιουργήστε το πρόγραμμα για τον προηγούμενο αλγόριθμο.
- Αποθηκεύστε το πρόγραμμα σας με το όνομα **pt1** στο φάκελο **ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ**, στην επιφάνεια εργασίας.
- Κατεβάστε το πρόγραμμα στο NXT και τρέξτε το.

Απέκτησε το ρομπότ σας την αναμενόμενη συμπεριφορά ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας. Τώρα σκεφτείτε μήπως υπάρχει περίπτωση να βελτιώσετε το πρόγραμμά σας.

Αν όχι, τι νομίζετε ότι δεν πήγε καλά.

• Προσπαθήστε να διορθώσετε το λάθος σας και ξαναδοκιμάστε.

Λειτουργήσε τελικά το πρόγραμμα σας ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας.

Αν όχι, μην απογοητεύεστε. Θα τα καταφέρετε την επόμενη φορά.

Δραστηριότητα 2^η : αποφυγή εμποδίου με 2 αισθητήρες αφής

Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του,
το βραχίονα με τους δύο αισθητήρες αφής.

Συνδέστε τον αριστερό αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

Συνδέστε το δεξιό αισθητήρα αφής στη θύρα εισόδου 3 του NXT.

Στη δραστηριότητα αυτή θέλουμε να προγραμματίσουμε το όχημα - ρομπότ ώστε να είναι σε θέση να παρακάμπτει, με πιο αποτελεσματικό τρόπο, τα εμπόδια που θα συναντήσει στην πορεία του.

Αυτό που θα επιδιώξουμε, είναι να μπορεί το ρομπότ μας να βγει από μία σπηλιά.

Γράψτε τι ώρα είναι τώρα : _____

Ανάλυση προβλήματος :

Το όχημα - ρομπότ μας κινείται μπροστά, σε ευθύγραμμη πορεία. Με αυτό τον τρόπο κάποια στιγμή θα συναντήσει κάποιο εμπόδιο και θα παραμείνει κολλημένο στο σημείο εκείνο.

Τι θα κάνατε εσείς αν είχατε κλειστά τα μάτια σας και έπρεπε να κινηθείτε σε ένα άγνωστο χώρο έχοντας και τα δύο χέρια σας, μπροστά σας, τεντωμένα και ακίνητα;

Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει δύο αισθητήρες αφής, δεξιά και αριστερά, στο μπροστινό του μέρος "είναι σε θέση να καταλάβει", όχι μόνο ότι έχει συναντήσει ένα εμπόδιο στην πορεία του, αλλά και σε ποιά πλευρά του.

Σκεφτείτε πόσες διαφορετικές περιπτώσεις υπάρχουν, όταν το όχημα μας συγκρουστεί με ένα εμπόδιο.

πχ. Να πέσει με την αριστερή πλευρά, οπότε να πατηθεί ο αριστερός διακόπτης αφής 1.

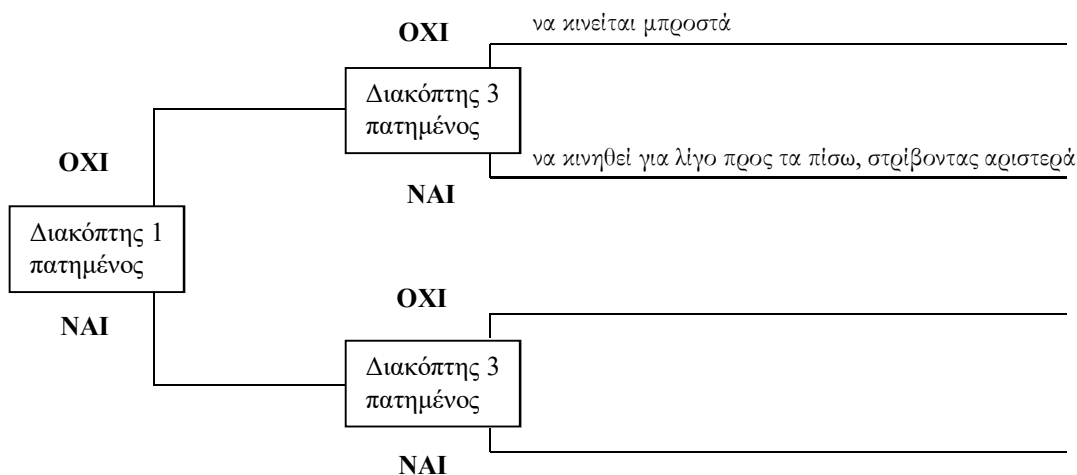
Απαριθμήστε τις επιθυμητές συμπεριφορές που θέλετε να εκτελεί το όχημα μας, για να αποφύγει το εμπόδιο σε κάθε περίπτωση.

πχ. Αν πέσει με την αριστερή πλευρά στο εμπόδιο να κινηθεί για λίγο προς τα πίσω, στρίβοντας αριστερά.

Συμπληρώστε τον παρακάτω πίνακα αντιστοίχισης :

Αριστερός Διακόπτης 1 πατημένος	Δεξιός Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	να κινείται μπροστά
OXI	NAI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας αριστερά

Σχεδιάστε τις παραπάνω συμπεριφορές, όπως αυτές αντιστοιχίζονται με τους δυνατούς συνδυασμούς των πλήκτρων, με ένα λογικό διάγραμμα ροής.



Γράψτε τι ώρα είναι τώρα : _____

- Ανοίξτε το προγραμματιστικό περιβάλλον NXT-G και δημιουργήστε το πρόγραμμα από το προηγούμενο διάγραμμα ροής, χρησιμοποιώντας εντολή διακλάδωσης μέσα σε εντολή διακλάδωσης.
- Αποθηκεύστε το πρόγραμμα σας με το όνομα pt2 στο φάκελο ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ, στην επιφάνειας εργασίας
- Κατεβάστε το πρόγραμμα στο NXT.
- Τοποθετείστε το ρομπότ σας στη σπηλιά που έχουμε δημιουργήσει και θέστε το σε λειτουργία.

Γράψτε τι ώρα είναι τώρα : _____

Σενάρια Ρομποτικής με το Mindstorms NXT

Απέκτησε το ρομπότ σας την αναμενόμενη συμπεριφορά ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας. Τώρα σκεφτείτε μήπως υπάρχει περίπτωση να βελτιώσετε το πρόγραμμα σας.

Αν όχι, τι νομίζετε ότι δεν πήγε καλά.

• Προσπαθήστε να διορθώσετε το λάθος σας και ξαναδοκιμάστε.

Λειτουργήσε τελικά το πρόγραμμα σας ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας.

Αν όχι, μην απογοητεύεστε. Θα τα καταφέρετε την επόμενη φορά.

Δραστηριότητα 3^η : αποφυγή εμποδίου με αισθητήρες υπερήχων

Σκεφτείτε την περίπτωση που για την επίλυση του παραπάνω προβλήματος χρησιμοποιήσουμε αισθητήρες υπερήχων στη θέση των αισθητήρων αφής.

Θεωρείται ότι το πρόβλημα μας θα αντιμετωπιστεί πιο αποτελεσματικά ή όχι και γιατί ;

Συγχαρητήρια ☺

Ολοκληρώσατε το 4^ο φύλλο δραστηριοτήτων!!!

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 4^ο

Πρόκληση με αισθητήρες αφής

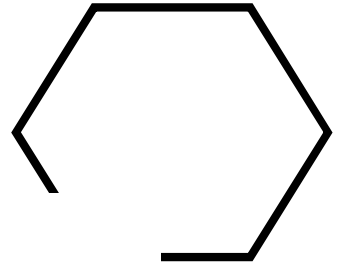
Σημειώσεις Καθηγητή

Έξοδος από σπηλιά

Θα επιδιώξουμε να προγραμματίσουμε το όχημα - ρομπότ μας ώστε να είναι σε θέση να βρει την έξοδο από μία σπηλιά και να βγει από αυτήν.

Θα νικήσει η ομάδα που θα τα καταφέρει με τον καλύτερο δυνατό τρόπο.

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να παρακάμπτει τα εμπόδια που θα συναντήσει στην πορεία του.



Δραστηριότητα 1^η : αποφυγή εμποδίου με 1 αισθητήρα αφής

Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας αφής.

Τον αισθητήρα αφής αυτόν τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Στη δραστηριότητα αυτή θέλουμε να προγραμματίσουμε το όχημα - ρομπότ, ώστε να είναι σε θέση να παρακάμπτει τα εμπόδια που θα συναντήσει στην πορεία του.

Ανάλυση προβλήματος :

Το όχημα - ρομπότ μας κινείται μπροστά, σε ευθύγραμμη πορεία. Με αυτό τον τρόπο κάποια στιγμή θα συναντήσει κάποιο εμπόδιο και θα παραμείνει κολλημένο στο σημείο εκείνο.

Παιχνίδι Ρόλων: Τι θα κάνατε εσείς, αν είχατε κλειστά τα μάτια σας και έπρεπε να κινηθείτε σε ένα άγνωστο χώρο, έχοντας μόνο το ένα χέρι σας, μπροστά σας, τεντωμένο και ακίνητο ;

Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει αισθητήρα αφής στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι έχει συναντήσει ένα εμπόδιο στην πορεία του.

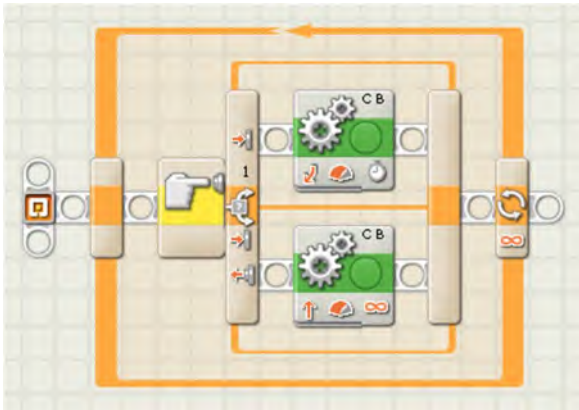
Σκεφτείτε και περιγράψτε τον τρόπο, με τον οποίο το ρομπότ μας θα μπορέσει να προσπεράσει το εμπόδιο αυτό και να συνεχίσει την πορεία του.

Απάντηση :

α) Για να είναι σε θέση το ρομπότ μας να προσπεράσει το εμπόδιο αυτό και να συνεχίσει την πορεία του μπορούμε να του πούμε να πάει πίσω για λίγο, εκτελώντας στροφή προς μία κατεύθυνση πχ δεξιά. Στη συνέχεια να συνεχίσει ευθεία και να επαναλάβει την ίδια διαδικασία για κάθε εμπόδιο.



β) Το ίδιο πρόβλημα με διακλάδωση αφής :



Δραστηριότητα 2^η : αποφυγή εμποδίου με 2 αισθητήρες αφής

Αυτή τη φορά στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκονται τοποθετημένοι δύο αισθητήρες αφής. Οι αισθητήρες αφής είναι τοποθετημένοι ο ένας στην αριστερή και ο άλλος στη δεξιά μεριά του προφυλακτήρα. Τον αριστερό αισθητήρα αφής τον συνδέουμε στη θύρα εισόδου 1 ενώ το δεξιό στη θύρα εισόδου 3 του NXT.

Τοποθετώντας δύο αισθητήρες αφής το όχημα είναι σε θέση να παρακάμπτει, με πιο αποτελεσματικό τρόπο, τα εμπόδια που θα συναντήσει στην πορεία του.

Στη δραστηριότητα αυτή θέλουμε να προγραμματίσουμε το όχημα - ρομπότ ώστε να είναι σε θέση να παρακάμπτει, με πιο αποτελεσματικό τρόπο, τα εμπόδια που θα συναντήσει στην πορεία του.

Αυτό που θα επιδιώξουμε, είναι να μπορεί το ρομπότ μας να βγει από μία σπηλιά.

Ανάλυση προβλήματος :

Παιχνίδι Ρόλων: Τι θα κάνατε εσείς αν είχατε κλειστά τα μάτια σας και έπρεπε να κινηθείτε σε ένα άγνωστο χώρο έχοντας και τα δύο χέρια σας, μπροστά σας, τεντωμένα και ακίνητα;

Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει δύο αισθητήρες αφής, δεξιά και αριστερά, στο μπροστινό του μέρος "είναι σε θέση να καταλάβει", όχι μόνο ότι έχει συναντήσει ένα εμπόδιο στην πορεία του, αλλά και σε ποιά πλευρά του.

Σκεφτείτε και περιγράψτε τον τρόπο, με τον οποίο το ρομπότ μας θα μπορέσει να προσπεράσει το εμπόδιο αυτό και να συνεχίσει την πορεία του.

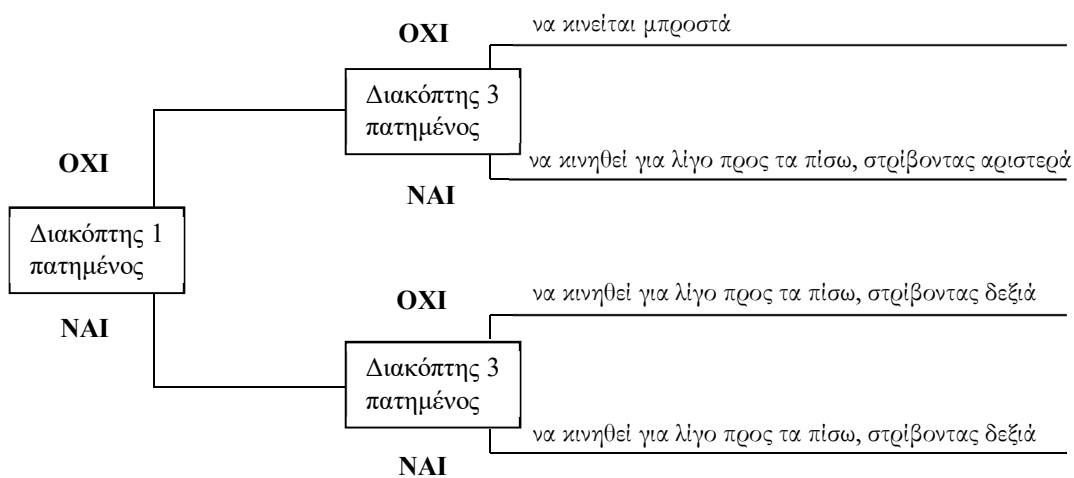
Απάντηση :

Μπορούμε τώρα να του πούμε να πάει πίσω για λίγο και να στρίψει ελαφρώς ανάλογα με τον αισθητήρα που πατήθηκε. Στη συνέχεια να συνεχίσει ευθεία και να επαναλάβει την ίδια διαδικασία για κάθε εμπόδιο.

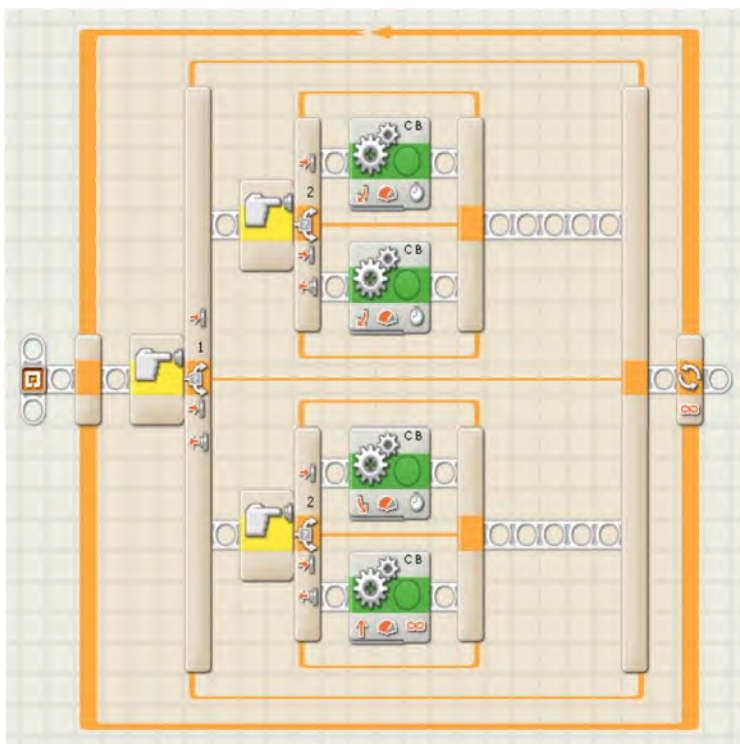
Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

Αριστερός Διακόπτης 1 πατημένος	Δεξιός Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	να κινείται μπροστά
OXI	NAI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας αριστερά
NAI	OXI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας δεξιά
NAI	NAI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας πχ δεξιά

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.

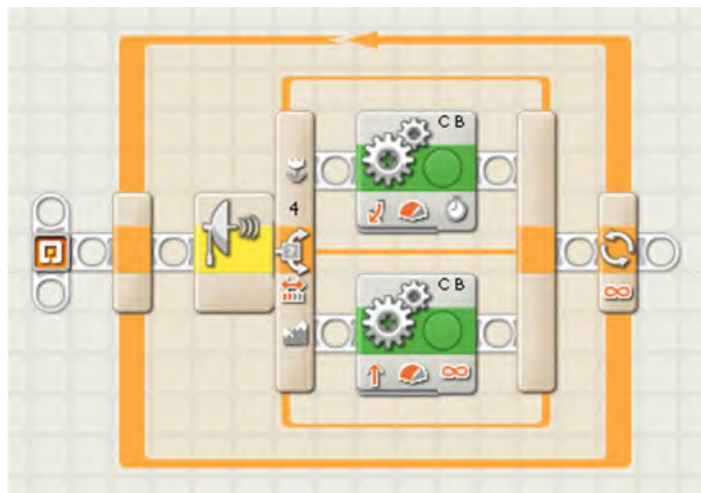


Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



Δραστηριότητα 3^η : αποφυγή εμποδίου με αισθητήρα υπερήχων

Στην περίπτωση που αντί για τον αισθητήρα αφής χρησιμοποιήσουμε έναν αισθητήρα υπερήχων τα πράγματα γίνονται πιο απλά, αφού τώρα το όχημα μας δεν χρειάζεται πλέον να πέσει πάνω στο εμπόδιο για να το αντιληφθεί. Μπορεί να εντοπίσει ένα εμπόδιο το οποίο βρίσκεται στην πορεία της κίνησης του, από αρκετά μεγάλη απόσταση και να το παρακάμψει εκτελώντας μία μικρή στροφή και συνεχίζοντας πάλι μπροστά.



ΓΝΩΣΤΙΚΟ ΑΝΤΙΚΕΙΜΕΝΟ: ΠΛΗΡΟΦΟΡΙΚΗ
**ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
ΜΕ LEGO MINDSTORMS NXT**

A) Περιγραφή Διδακτικής Παρέμβασης στην τάξη – με χρήση ΤΠΕ

Δημιουργός σεναρίου: Γεώργιος Κυριακού

Ημ/νίες εφαρμογής:

Τάξεις/Τμήματα: Γ' Γυμνασίου / Α' Λυκείου

Τίτλος διδακτικής παρέμβασης

Σύνθετη δραστηριότητα "ακολουθώντας τη γραμμή"

Γνωστικό αντικείμενο και εμπλεκόμενες γνωστικές περιοχές

Πληροφορική Γ' Γυμνασίου. Διδακτική Ενότητα: 1.2 :

Ο Προγραμματισμός στην πράξη

Τάξη εφαρμογής

Γ' Γυμνασίου (Νέα πιλοτικά προγράμματα σπουδών)

Συμβατότητα και συσχέτιση με το Αναλυτικό Πρόγραμμα

Προγραμματίζω τον υπολογιστή - Οι μαθητές σχεδιάζουν και υλοποιούν προγράμματα για να επιλύσουν πραγματικά προβλήματα σύμφωνα με τα νέα πιλοτικά προγράμματα σπουδών.

Υλικοτεχνική υποδομή

Εργαστήριο πληροφορικής, πίνακας μαρκαδόρου, πακέτο Lego Mindstorms NXT Education.

Το Λογισμικό

Περιβάλλον προγραμματισμού Lego NXT-G.

Σκοπός

Να αποκτήσει ο μαθητής σχετική ευχέρεια και εμπειρία στη δεξιότητα του δομημένου προγραμματισμού.

Διδακτικοί στόχοι:

- Να ολοκληρώσουν μία σειρά από προκλήσεις, σταδιακά αυξανόμενης δυσκολίας, οι οποίες απαιτούν γνώσεις μηχανικής, μαθηματικών και προγραμματιστικές δεξιότητες.
- Να κατανοήσουν το πλεονέκτημα της εργασίας με ομαδικό πνεύμα το οποίο περιλαμβάνει καταιγισμό ιδεών, σχεδίαση, υλοποίηση, δοκιμή και συνεννόηση για το τελικό αποτέλεσμα.
- Να αφομοιώσουν τις συστάσεις και την ανατροφοδότηση που δίνεται από τον καθοδηγητή ώστε να είναι σε θέση να αντιμετωπίσουν πιο πολύπλοκα θέματα.

Γενικότεροι μαθησιακοί στόχοι

- να αναπτύξουν την αναλυτική και κριτική και συνθετική ικανότητα.
- να εφαρμόζουν τεχνικές ελέγχου και διόρθωσης σφαλμάτων στα προγράμματα που δημιουργούν.

Γενικότεροι εκπαιδευτικοί στόχοι

- συνεργασία των μαθητών για εξαγωγή αποτελεσμάτων και συμπερασμάτων

Στόχοι ως προς την αξιοποίηση της τεχνολογίας

- αντίληψη των μαθητών σχετικά με τη ρομποτική.

Διδακτική μέθοδος

Οι μαθητές εργάζονται σε ομάδες (κοινωνιο-επικοινωνιακή μάθηση), έμμεση διδασκαλία - μαθητοκεντρική

Διδακτικές στρατηγικές / τεχνικές

Μορφή καθοδήγησης (θέτουμε ερωτήματα και οι μαθητές επιλέγουν βήματα και δράσεις σε πλαίσιο καθοδήγησης), πειραματισμός, διερεύνηση, ανακάλυψη, επίλυση προβλήματος.

Οργάνωση διδασκαλίας

Ομάδες των τριών ατόμων. Αναθέτουμε σε ένα μέλος της ομάδας το ρόλο του εκπροσώπου, σε ένα άλλο το ρόλο του γραμματέα ο οποίος θα κρατάει τις σημειώσεις και στο τελευταίο μέλος το ρόλο του προγραμματιστή στο NXT-G. Κάθε ομάδα έχει το δικό της υπολογιστή. Μία από τις ομάδες θα επιλεγεί για να παίξει το παιχνίδι ρόλων .

Ρόλος εκπαιδευτικού

Συμβουλευτικός παρά καθοδηγητικός. Τονίζει συνέχεια στους μαθητές τη σημαντικότητα της ομαδικής δουλειάς και της συνεργασίας, ενισχύοντας την ευγενή άμιλλα μεταξύ των μαθητών και προσεγγίζει τα προβλήματα που δημιουργούνται ανάμεσα στα μέλη των ομάδων με κατάλληλο τρόπο ώστε να ανατροφοδοτούν την τάξη, βοηθώντας με αυτό τον τρόπο τη συνεργασία μεταξύ των παιδιών

Εκτιμώμενη διάρκεια

2 διδακτικές ώρες

2. ΑΝΑΠΤΥΞΗ ΣΕΝΑΡΙΟΥ

Το επιστημονικό περιεχόμενο

Προγραμματισμός → Σύνθετα προβλήματα

Προαπαιτούμενες γνώσεις:

1. Εισαγωγή στη Ρομποτική (βλέπε σημειώσεις: [B1- Ρομποτική](#))
2. Γνωριμία με το Πακέτο Lego Mindstorms (βλέπε σημειώσεις: [B2- Πακέτο Lego Mindstorms NXT](#))
3. Εξοικείωση με το Λογισμικό NXT-G (βλέπε σημειώσεις: [B3- Το προγραμματιστικό περιβάλλον NXT-G](#) και [Φύλλο Εργασιών 1 - Γνωριμία με το Λογισμικό NXT-G \(Καθηγητή\)](#))
4. Ευχέρεια στις Εντολές κίνησης και τη Δομή Επανάληψης (βλέπε σημειώσεις: [B4- Οι εντολές του NXT-G](#) και [Φύλλο Εργασιών 2 - Κινητήρες και Δομή επανάληψης \(Καθηγητή\)](#))
5. Ευχέρεια στις Εντολές ελέγχου και τη Δομή Επιλογής (βλέπε σημειώσεις: [B4- Οι εντολές του NXT-G](#) και [Φύλλο Εργασιών 3 - Αισθητήρες και Δομή επιλογής \(Καθηγητή\)](#))

Περιγραφή δραστηριοτήτων

Αναλυτική παρουσίαση στο έγγραφο “[A2- Πορεία Διδασκαλίας](#)” (Επισυνάπτεται)

1. Δραστηριότητα, βασισμένη σε σχέδιο εργασίας ακολουθώντας τις οδηγίες και καταγράφοντας την πορεία του “[Φύλλο Εργασιών 5 - Πρόκληση με αισθητήρες φωτός](#)”.
2. Αξιολόγηση των μαθητών με τη συμπλήρωση εργασιών για το σπίτι των Φύλλων τελικής Αξιολόγησης
“[6.1 Δομές Προγραμματισμού](#)”
“[6.2 εντυπώσεις και συμπεράσματα](#)”

Φύλλα εργασίας

[Φύλλο Εργασιών 5 - Πρόκληση με αισθητήρες φωτός](#) (Επισυνάπτεται)

Σενάρια Ρομποτικής με το Mindstorms NXT

Αξιολόγηση μαθητών

Αξιολόγηση της ικανότητας επίλυσης προβλημάτων και υλοποίησης των αντίστοιχων προγραμμάτων μέσα από τις δραστηριότητες οι οποίες θα υλοποιηθούν.

Αξιολόγηση σεναρίου

Αξιολόγηση μέσα από τις ίδιες δραστηριότητες και της ικανότητας συνεργασίας των μαθητών, όπως και από τη συμπλήρωση των φύλλων αξιολόγησης.

B) Επισυναπτόμενο υλικό:

Φύλλα εργασιών :

[Φύλλο Εργασιών 5 - Πρόκληση με αισθητήρες φωτός](#)

Σημειώσεις του καθηγητή :

[A2- Πορεία Διδασκαλίας](#)

[B3- Το προγραμματιστικό περιβάλλον NXT-G](#)

[B4- Οι εντολές του NXT-G](#)

[Φύλλο Εργασιών 5 - Πρόκληση με αισθητήρες φωτός \(Καθηγητή\)](#)

Φύλλο Αξιολόγησης μαθητών :

Φύλλα Τελικής αξιολόγησης : [6.1 Δομές Προγραμματισμού](#)
[6.2 εντυπώσεις και συμπεράσματα](#)

Έτοιμα προγράμματα :

[Φάκελος 5 - Πρόκληση με αισθητήρες φωτός](#)

Γ) Αποτελέσματα –Σχόλια – Προβληματισμοί (Αναστοχασμός)

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 5^ο

Πρόκληση με αισθητήρες φωτός

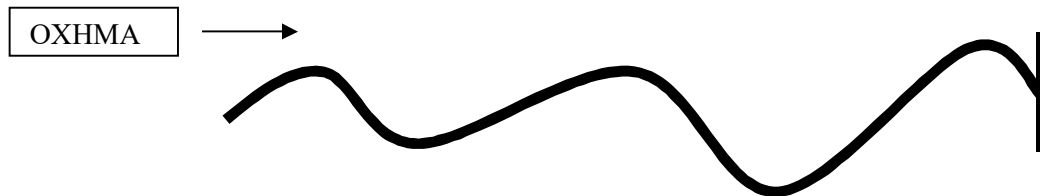
Όνομα _____

Ημερομηνία _____

Ακολουθώντας τη γραμμή (Line follower)

Θα επιδιώξουμε να προγραμματίσουμε το όχημα - ρομπότ μας, ώστε να είναι σε θέση να προχωράει κατά μήκος μίας μαύρης καμπύλης γραμμής, σχεδιασμένης σε μία λευκή επιφάνεια.

Θα νικήσει η ομάδα που θα τα καταφέρει με τον καλύτερο δυνατό τρόπο.



Δραστηριότητα 1^η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι

Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα φωτός.

Συνδέστε τον αισθητήρα φωτός στη θύρα εισόδου 3 του NXT.

Στη δραστηριότητα αυτή θέλουμε να προγραμματίσουμε το όχημα - ρομπότ, ώστε να είναι σε θέση να προχωράει κατά μήκος της μαύρης καμπύλης που είναι σχεδιασμένη στο πάτωμα.

- Ανοίξτε το τούβλο NXT και βρείτε το ήδη φορτωμένο πρόγραμμα **pl1c**.
- Τοποθετήστε το όχημα-ρομπότ στην αρχή της μαύρης γραμμής από την αριστερή πλευρά και τρέξτε το.

Παρατηρήστε και περιγράψτε αναλυτικά τις κινήσεις που εκτελεί το ρομπότ μας.

Στο παραπάνω πρόγραμμα, χρησιμοποιούμε την εντολή αναμονής για σκοτάδι. Τι μπορούμε να πετύχουμε με αυτή την εντολή ;

Αν τοποθετήσουμε το ρομπότ από την άλλη πλευρά της γραμμής, θα μπορέσει το ρομπότ μας να ακολουθήσει και πάλι τη μαύρη γραμμή ; Τι πιστεύετε ;

Εξηγήστε τη συμπεριφορά που παρουσίασε το όχημα-ρομπότ μας.

Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει αισθητήρα φωτός στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι στην πορεία του έχει συναντήσει τη μία μαύρη γραμμή.

Θεωρήστε ότι υπάρχει μία αόρατη σε εσάς γραμμή χαραγμένη στο πάτωμα, την οποία αντιλαμβάνεστε μόνο τη στιγμή που θα τη συναντήσετε. (πχ μία γραμμή από αόρατη μελάνη την οποία δεν βλέπετε, αλλά μόλις πατήσετε πάνω της ειδοποιείστε από ένα σφύριγμα.)

Αρχικά είστε τοποθετημένοι δίπλα στη γραμμή αυτή. Σας ενημερώνουν επίσης, ότι η γραμμή βρίσκεται στα **δεξιά** σας.

Τι θα κάνετε σε αυτή την περίπτωση, ώστε να προχωράτε στην κατεύθυνση της γραμμής αυτής; (Γράψτε την απάντηση σας σε μορφή οδηγιών για ένα ρομπότ.)

Δώστε τις οδηγίες από τον αλγόριθμο που δημιουργήσατε σε κάποιον άλλο, ο οποίος και θα παριστάνει το ρομπότ.

Σενάρια Ρομποτικής με το Mindstorms NXT

Μπόρεσε ο συμμαθητής σας να ακολουθήσει σωστά τη γραμμή με τις οδηγίες που του δώσατε ;

ΝΑΙ

ΟΧΙ

Αν όχι διορθώστε τις οδηγίες και ξαναπροσπαθήστε

Δημιουργήστε τον αλγόριθμο για να είναι σε θέση το όχημα-ρομπότ μας να ακολουθεί τη μαύρη γραμμή.

- Ξεκινήστε το προγραμματιστικό περιβάλλον NXT-G και δημιουργήστε το πρόγραμμα για τον προηγούμενο αλγόριθμο.
- Αποθηκεύστε το νέο πρόγραμμα με το όνομα **pl1** στο φάκελο ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ, στην επιφάνειας εργασίας.
- Κατεβάστε το πρόγραμμα στο NXT και τρέξτε το.

Γράψτε τι ώρα είναι τώρα : _____

Απέκτησε το ρομπότ σας την αναμενόμενη συμπεριφορά ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας. Τώρα σκεφτείτε μήπως υπάρχει περίπτωση να βελτιώσετε το πρόγραμμα σας.

Αν όχι, τι νομίζετε ότι δεν πήγε καλά.

- Προσπαθήστε να διορθώσετε το λάθος σας και ξαναδοκιμάστε.

Λειτουργήσε τελικά το πρόγραμμά σας ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας.

Αν όχι, μην απογοητεύεστε. Θα τα καταφέρετε την επόμενη φορά.

Δραστηριότητα 2^η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως

Στην προηγούμενη δραστηριότητα όταν μειώνουμε το χρόνο, στον οποίο το ρομπότ μας απομακρύνεται από τη μαύρη γραμμή, τότε το ρομπότ μας, καθώς ακολουθάει τη γραμμή, κινείται πιο ομαλά.



Μπορούμε να μειώσουμε ή να αυξήσουμε το χρόνο αυτόν, όσο θέλουμε ή υπάρχει κάποιος περιορισμός ;

Πως νομίζεται ότι μπορούμε να βελτιώσουμε το προηγούμενο πρόγραμμα μας; Με ποια εντολή μπορούμε να αντικαταστήσουμε την εντολή 'αναμονή για χρόνο' για να έχουμε καλύτερο αποτέλεσμα ;

- Ανοίξτε το πρόγραμμα **pl1c** της προηγούμενης δραστηριότητας στο προγραμματιστικό περιβάλλον NXT-G
- Αντικαταστήστε την εντολή 'αναμονή για χρόνο' με την εντολή 'αναμονή για φως'.
- Αποθηκεύστε το πρόγραμμα σας με το όνομα **pl2** στο φάκελο ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ' στην επιφάνεια εργασίας
- Κατεβάστε το πρόγραμμα στο NXT και τρέξτε το.

Γράψτε τι ώρα είναι τώρα : _____

Μπορείτε να παρατηρήσετε τη διαφορά στον τρόπο που το ρομπότ σας ακολουθεί τη μαύρη γραμμή.

Το ρομπότ κινείται τώρα λιγότερο ή περισσότερο ομαλά ;

Δραστηριότητα 3^η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με διακλάδωση

- Ανοίξτε το πρόγραμμα **pl2c** της προηγούμενης δραστηριότητας στο προγραμματιστικό περιβάλλον NXT-G
- Τροποποιήστε το πρόγραμμα, χρησιμοποιώντας αντί για τις εντολές 'αναμονή για σκοτάδι' και 'αναμονή για φως', την εντολή 'διακλάδωση φωτός'.
- Αποθηκεύστε το πρόγραμμα σας με το όνομα **pl3** στο φάκελο ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ, στην επιφάνεια εργασίας
- Κατεβάστε το πρόγραμμα στο NXT και τρέξτε το.

Γράψτε τι ώρα είναι τώρα : _____

Λειτουργεί το νέο πρόγραμμα σας με τον ίδιο τρόπο με τον προηγούμενο ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας. Τώρα σκεφτείτε μήπως υπάρχει περίπτωση να βελτιώσετε το πρόγραμμα σας.

Αν όχι, τι νομίζετε ότι δεν πήγε καλά.

- Προσπαθήστε να διορθώσετε το λάθος σας και ξαναδοκιμάστε.

Λειτουργήσε τελικά το πρόγραμμα σας ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας.

Αν όχι, μην απογοητεύεστε. Θα τα καταφέρετε την επόμενη φορά.

Δραστηριότητα 4^η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση μέσα σε διακλάδωση.

Για καλύτερο αποτέλεσμα, μπορούμε να χρησιμοποιήσουμε δύο αισθητήρες φωτός, τοποθετημένους έτσι ώστε να βρίσκονται δεξιά και αριστερά από τη γραμμή.

Τοποθετείστε στο όχημα - ρομπότ στο μπροστινό μέρος του, το βραχίονα με τους δύο αισθητήρες φωτός.

Συνδέστε τον αριστερό αισθητήρα φωτός στη θύρα εισόδου 1 του NXT.

Συνδέστε το δεξιό αισθητήρα φωτός στη θύρα εισόδου 3 του NXT.

Γράψτε τι ώρα είναι τώρα : _____

Θεωρήστε ότι όπως και προηγουμένως υπάρχει μία αόρατη σε εσάς γραμμή χαραγμένη στο πάτωμα, την οποία αντιλαμβάνεστε μόνο τη στιγμή που θα τη συναντήσετε. (πχ μία γραμμή από αόρατη μελάνη την οποία δεν βλέπετε, αλλά μόλις πατήσετε πάνω της ειδοποιείστε από ένα σφύριγμα.) Εσείς τώρα βρίσκεστε τοποθετημένοι πάνω από τη γραμμή η οποία βρίσκεται ανάμεσα στα δύο πόδια σας.

Τι θα κάνετε σε αυτή την περίπτωση, ώστε να προχωράτε στην κατεύθυνση της γραμμής αυτής; (Σκεφτείτε την απάντηση σας σε μορφή οδηγιών για ένα ρομπότ.)

• Τοποθετήστε το όχημα - ρομπότ πάνω στη μαύρη γραμμή, ώστε οι αισθητήρες φωτός να βρίσκονται δεξιά και αριστερά της μαύρης γραμμής.

Ποιες είναι οι πιθανές καταστάσεις που μπορούν να ανιχνεύσουν οι δύο αισθητήρες φωτός κατά την κίνηση του ρομπότ μας :

πχ. και οι δύο αισθητήρες φωτός να ανιχνεύουν το λευκό δηλ να βρίσκονται εκατέρωθεν της μαύρης γραμμής.

Σενάρια Ρομποτικής με το Mindstorms NXT

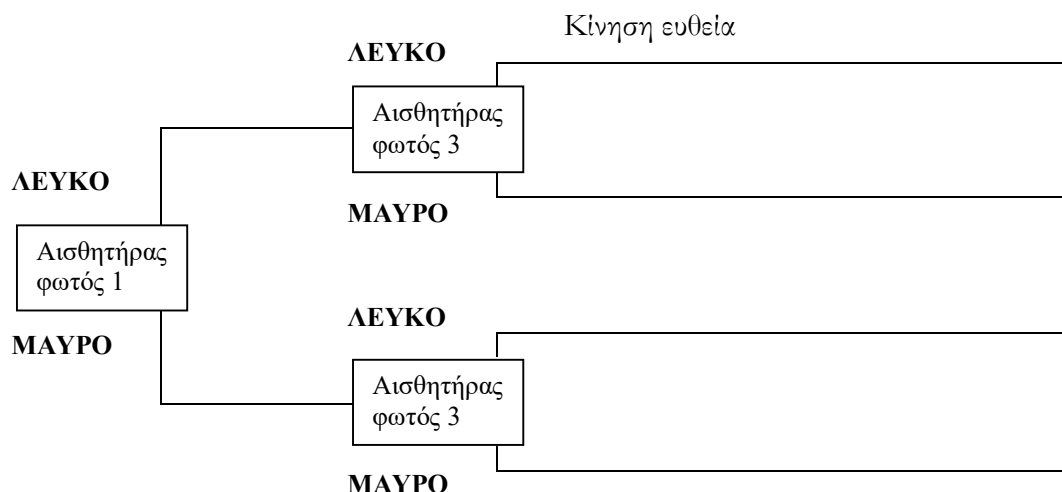
Τι εντολή πρέπει να δώσουμε σε κάθε μία από τις προηγούμενες καταστάσεις, ώστε το ρομπότ μας να συνεχίσει να κινείται πάνω στη γραμμή ;

πχ. Όταν και δύο αισθητήρες φωτός ανιχνεύουν το λευκό το όχημα να κινείται ευθεία.

Αντιστοιχίστε τις παραπάνω συμπεριφορές κίνησης του ρομπότ μας, με τους δυνατούς συνδυασμούς των αισθητήρων φωτός, δημιουργώντας έναν πίνακα αντιστοίχισης.

Αισθητήρας φωτός 1	Αισθητήρας φωτός 3	Συμπεριφορά κίνησης
ΛΕΥΚΟ	ΛΕΥΚΟ	Κίνηση ευθεία

Σχεδιάστε τις παραπάνω συμπεριφορές, όπως αυτές αντιστοιχίζονται με τους δυνατούς συνδυασμούς των αισθητήρων φωτός, με ένα λογικό διάγραμμα ροής.



- Δημιουργείστε στο προγραμματιστικό περιβάλλον NXT-G το πρόγραμμα από το προηγούμενο διάγραμμα ροής χρησιμοποιώντας εντολή διακλάδωσης μέσα σε εντολή διακλάδωσης.

Σενάρια Ρομποτικής με το Mindstorms NXT

- Αποθηκεύστε το πρόγραμμα σας με το όνομα **pl4** στο φάκελο ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ, στην επιφάνειας εργασίας
- Κατεβάστε το πρόγραμμα στο NXT και τρέξτε το.

Γράψτε τι ώρα είναι τώρα : _____

Λειτουργεί το νέο πρόγραμμα σας με τον ίδιο τρόπο με τον προηγούμενο ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας. Τώρα σκεφτείτε μήπως υπάρχει περίπτωση να βελτιώσετε το πρόγραμμα σας.

Αν όχι, τι νομίζετε ότι δεν πήγε καλά.

• Προσπαθήστε να διορθώσετε το λάθος σας και ξαναδοκιμάστε.

Λειτουργήσε τελικά το πρόγραμμα σας ;

ΝΑΙ

ΟΧΙ

Αν ναι, Μπράβο σας.

Αν όχι, μην απογοητεύεστε. Θα τα καταφέρετε την επόμενη φορά.

Συγχαρητήρια ☺

Ολοκληρώσατε το 5^ο φύλλο δραστηριοτήτων!!!

ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ ΡΟΜΠΟΤΙΚΗΣ LEGO MINDSTORMS NXT



Φύλλο Εργασιών 5^ο

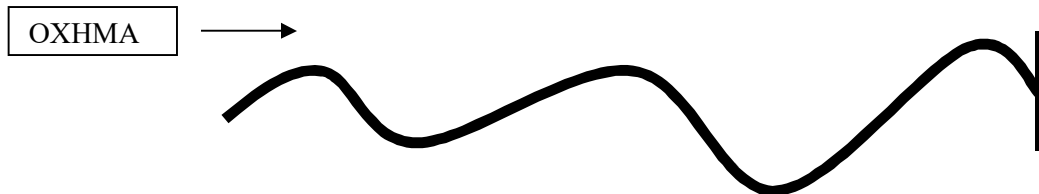
Πρόκληση με αισθητήρες φωτός

Σημειώσεις Καθηγητή

Ακολουθώντας τη γραμμή (Line follower)

Θα επιδιώξουμε να προγραμματίσουμε το όχημα - ρομπότ μας, ώστε να είναι σε θέση να προχωράει κατά μήκος μίας μαύρης καμπύλης γραμμής, σχεδιασμένης σε μία λευκή επιφάνεια.

Θα νικήσει η ομάδα που θα τα καταφέρει με τον καλύτερο δυνατό τρόπο.



Δραστηριότητα 1^η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι

Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας φωτός.

Τον αισθητήρα φωτός αυτόν, τον συνδέουμε στη θύρα εισόδου 3 του NXT.

Στη δραστηριότητα αυτή θέλουμε να προγραμματίσουμε το όχημα - ρομπότ, ώστε να είναι σε θέση να προχωράει κατά μήκος της μαύρης καμπύλης που είναι σχεδιασμένη στο πάτωμα.

Ανάλυση προβλήματος :

Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει αισθητήρα φωτός στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι στην πορεία του έχει συναντήσει τη μία μαύρη γραμμή.

Παιχνίδι ρόλων: Θεωρήστε ότι υπάρχει μία αόρατη σε εσάς γραμμή χαραγμένη στο πάτωμα, την οποία αντιλαμβάνεστε μόνο τη στιγμή που θα τη συναντήσετε. (πχ μία γραμμή από αόρατη μελάνη την οποία δεν βλέπετε, αλλά μόλις πατήσετε πάνω της ειδοποιείστε από ένα σφύριγμα.) Αρχικά είστε τοποθετημένοι δίπλα στη γραμμή αυτή. Σας ενημερώνουν επίσης, ότι η γραμμή βρίσκεται στα **δεξιά** σας.

Τοποθετούμε το όχημα στην **αριστερή** πλευρά της γραμμής.

Σενάρια Ρομποτικής με το Mindstorms NXT

Ξεκινάμε κινώντας μόνο τον έναν κινητήρα ώστε να στρίψει προς την κατεύθυνση που βρίσκεται η γραμμή.

Μόλις τη συναντήσει (ο αισθητήρας φωτός ανιχνεύσει σκοτάδι) του δίνουμε εντολή να απομακρυνθεί από τη γραμμή κινώντας μόνο τον άλλο κινητήρα για συγκεκριμένο χρονικό διάστημα, και μετά να ξαναεπιστρέψει προς αυτήν.



Υποδείξεις :

Με το πρόγραμμα αυτό προκύπτει μία σειρά προβλημάτων:

Αν ο χρόνος όπου το ρομπότ απομακρύνεται από τη μαύρη γραμμή είναι αρκετά μεγάλος υπάρχει το ενδεχόμενο στην περίπτωση που αυτή στρίβει απότομα μακριά από το ρομπότ μας, να απομακρυνθεί το ρομπότ μας μακριά από αυτήν και τη χάσει εντελώς. Όταν μειώνουμε το χρόνο, στον οποίο το ρομπότ μας απομακρύνεται από τη μαύρη γραμμή, τότε το ρομπότ μας κινείται πιο ομαλά ακολουθώντας τη γραμμή. Από την άλλη όμως αν μειώσουμε πάρα πολύ το χρόνο που κινείται το ρομπότ όταν απομακρύνεται από τη μαύρη γραμμή στην περίπτωση που αυτή στρίβει απότομα προς το ρομπότ, το ρομπότ θα συνεχίσει να βρίσκεται μέσα στη γραμμή και μετά την απομάκρυνση του από αυτήν με συνέπεια να αργοπορήσει ή και να περάσει από την άλλη μεριά με αποτέλεσμα να τη χάσει εντελώς.

Δραστηριότητα 2 ^η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως

Στην προηγούμενη δραστηριότητα όταν μειώνουμε το χρόνο, στον οποίο το ρομπότ μας απομακρύνεται από τη μαύρη γραμμή, τότε το ρομπότ μας, καθώς ακολουθάει τη γραμμή, κινείται πιο ομαλά.

Μπορούμε να βελτιώσουμε το προηγούμενο πρόβλημα αν αντικαταστήσουμε την εντολή αναμονής για χρόνο με μία εντολή αναμονής για φως ώστε να απομακρύνεται από τη μαύρη γραμμή μέχρι να βρεθεί στη λευκή επιφάνεια.

Στην περίπτωση αυτήν το ρομπότ μας θα κινηθεί περισσότερο ομαλά.



Δραστηριότητα 3^η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με διακλάδωση

Εναλλακτικά το προηγούμενο πρόγραμμα μπορεί να απλοποιηθεί χρησιμοποιώντας την εντολή διακλάδωσης αισθητήρα φωτός.

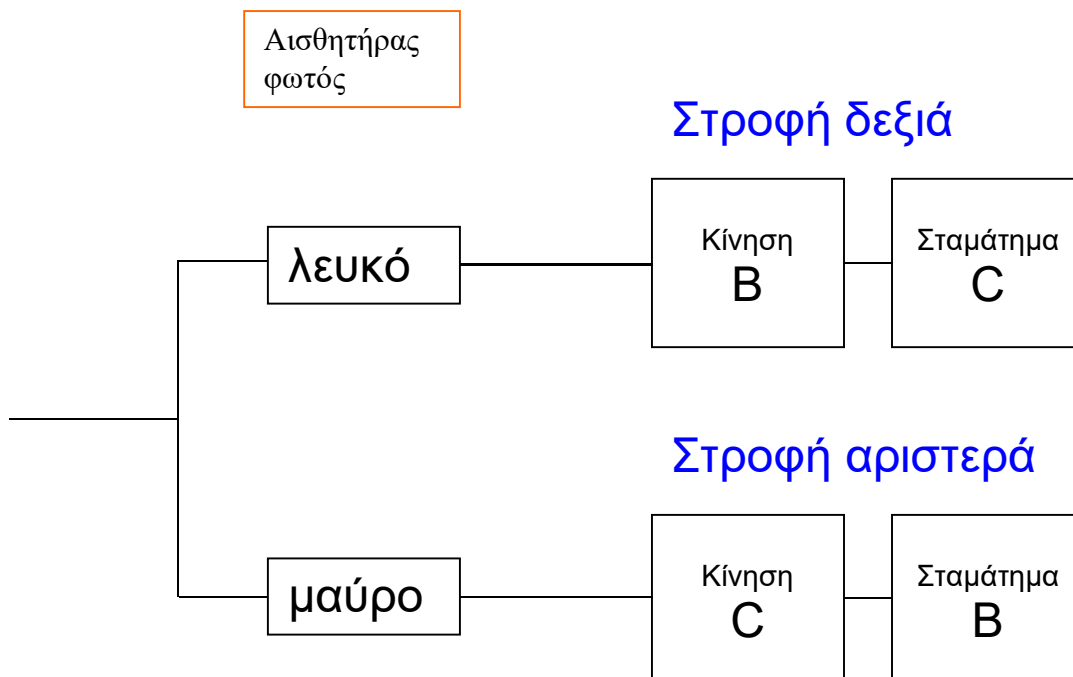
Ανάλυση προβλήματος

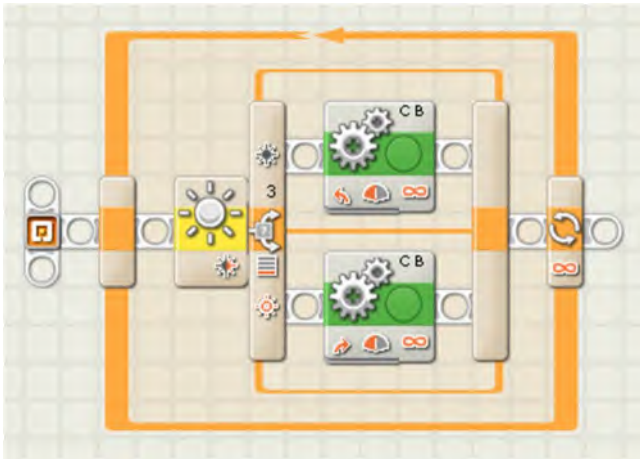
Κάνουμε τη σύμβαση το όχημα να βρίσκεται **αριστερά** από τη γραμμή.

Αυτό που θέλουμε είναι να δώσουμε εντολή στο όχημα ρομπότ μας, να στρίβει δεξιά (κίνηση κινητήρα B, σταμάτημα κινητήρα C) όταν ο αισθητήρας φωτός ανιχνεύει φωτεινό χρώμα και να στρίβει αριστερά (κίνηση κινητήρα C, σταμάτημα κινητήρα B) όταν ο αισθητήρας φωτός ανιχνεύει σκοτεινό χρώμα.

Το ρομπότ διαβάζει την τιμή από τον αισθητήρα φωτός στη θύρα 1 και τη συγκρίνει με την αριθμητική τιμή 45 που έχουμε ορίσει ως κατώφλι. Αν η τιμή είναι πάνω από το κατώφλι, τότε ο αισθητήρας φωτός βρίσκεται έξω από τη μαύρη γραμμή. Το πρόγραμμα κινεί τον αριστερό κινητήρα και σταματάει το δεξιό κινητήρα, στρίβοντας το ρομπότ μας δεξιά προς το αριστερό σύνορο της μαύρης γραμμής. Ομοίως αν η τιμή είναι κάτω από το κατώφλι το ρομπότ στρίβει αριστερά και απομακρύνεται από το αριστερό σύνορο της μαύρης γραμμής. Ο κύκλος αυτός επαναλαμβάνεται συνεχώς μέχρι να σταματήσουμε το πρόγραμμα.

Το παρακάτω λογικό διάγραμμα περιγράφει την επιθυμητή διαδικασία:





Δραστηριότητα 4^η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση μέσα σε διακλάδωση.

Για καλύτερο αποτέλεσμα, μπορούμε να χρησιμοποιήσουμε δύο αισθητήρες φωτός, τοποθετημένους έτσι ώστε να βρίσκονται δεξιά και αριστερά από τη γραμμή.

Παιχνίδι ρόλων: Θεωρήστε ότι όπως και προηγουμένως υπάρχει μία αόρατη σε εσάς γραμμή χαραγμένη στο πάτωμα, την οποία αντιλαμβάνεστε μόνο τη στιγμή που θα τη συναντήσετε. (πχ μία γραμμή από αόρατη μελάνη την οποία δεν βλέπετε, αλλά μόλις πατήσετε πάνω της ειδοποιείτε από ένα σφύριγμα.) Εσείς τώρα βρίσκεστε τοποθετημένοι πάνω από τη γραμμή η οποία βρίσκεται ανάμεσα στα δύο πόδια σας.

Τι θα κάνετε σε αυτή την περίπτωση, ώστε να προχωράτε στην κατεύθυνση της γραμμής αυτής; (Σκεφτείτε την απάντηση σας σε μορφή οδηγιών για ένα ρομπότ.)

Είναι πιο βολικό για να έχουμε καλύτερο έλεγχο των συνδυασμών που προκύπτουν από δύο αισθητήρες φωτός να χρησιμοποιήσουμε μία εντολή διακλάδωσης αισθητήρα φωτός μέσα σε μία άλλη εντολή διακλάδωση φωτός.

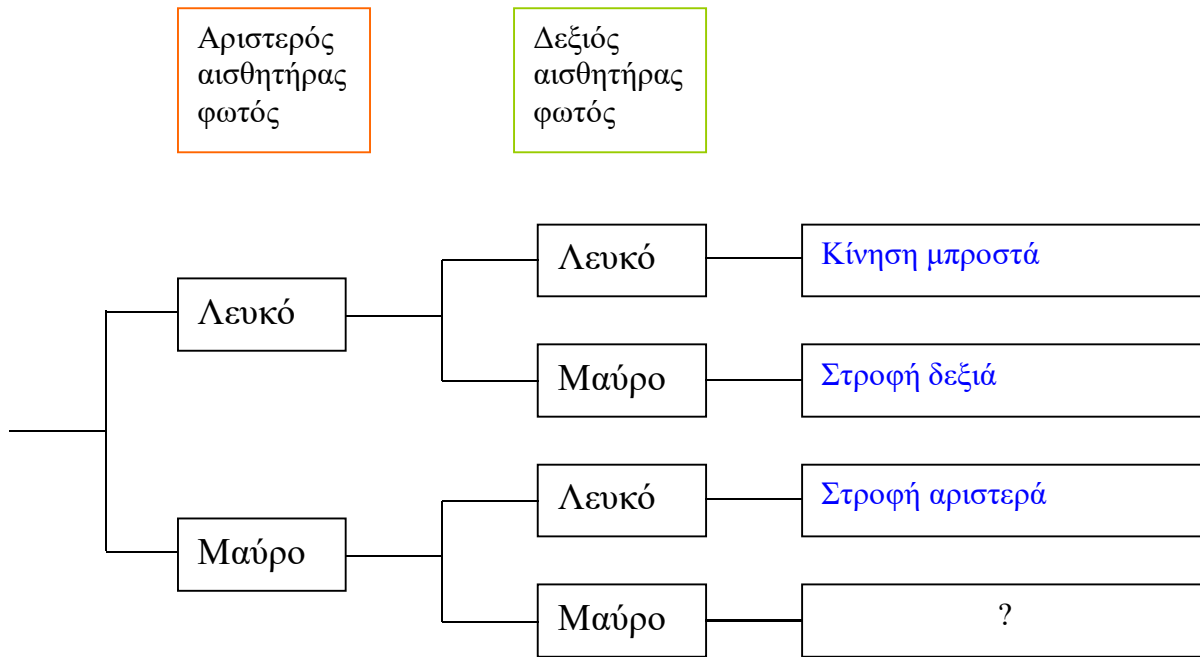
Ανάλυση προβλήματος :

Τοποθετούμε το όχημα - ρομπότ πάνω στη μαύρη γραμμή, ώστε οι αισθητήρες φωτός να βρίσκονται δεξιά και αριστερά της μαύρης γραμμής.

Αυτό που θέλουμε είναι να δώσουμε εντολή στο όχημα - ρομπότ μας κινείται ευθεία όταν και οι δύο αισθητήρες ανιχνεύουν φωτεινό χρώμα, να στρίβει αριστερά (κίνηση κινητήρα C, σταμάτημα κινητήρα A) όταν ο αισθητήρας φωτός 1 ανιχνεύει σκοτεινό χρώμα και να στρίβει δεξιά (κίνηση κινητήρα A, σταμάτημα κινητήρα C) όταν ο αισθητήρας φωτός 2 ανιχνεύει σκοτεινό χρώμα. Μπορούμε επίσης να το κάνουμε να σταματάει όταν και οι δύο αισθητήρες ανιχνεύουν σκοτεινό χρώμα.

Σενάρια Ρομποτικής με το Mindstorms NXT

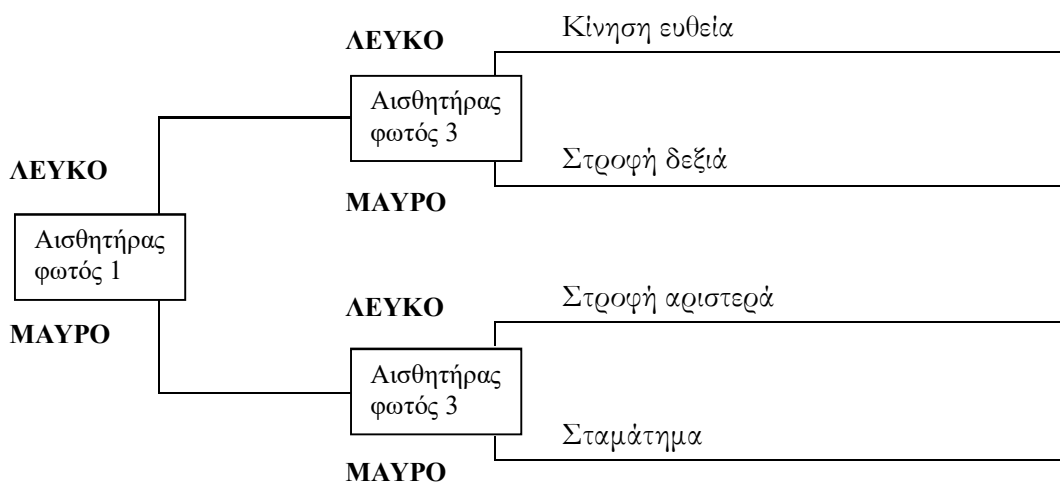
Το παρακάτω λογικό διάγραμμα περιγράφει την επιθυμητή διαδικασία:



Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

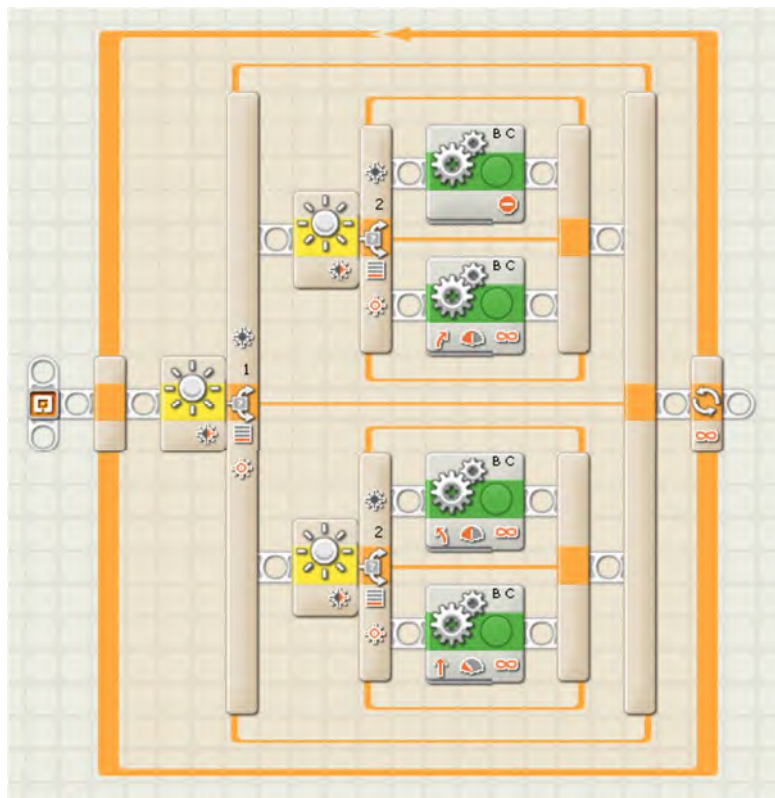
Αισθητήρας φωτός 1	Αισθητήρας φωτός 3	Συμπεριφορά κίνησης
ΛΕΥΚΟ	ΛΕΥΚΟ	Κίνηση ευθεία
ΛΕΥΚΟ	ΜΑΥΡΟ	Στροφή δεξιά
ΜΑΥΡΟ	ΛΕΥΚΟ	Στροφή αριστερά
ΜΑΥΡΟ	ΜΑΥΡΟ	Σταμάτημα

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.



Σενάρια Ρομποτικής με το Mindstorms NXT

Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



Φύλλο Τελικής αξιολόγησης : Δομές Προγραμματισμού

Όνομα _____

Ημερομηνία _____

1. Τι πετυχαίνουμε με μία δομή επιλογής ;
Πότε θα χρησιμοποιήσουμε μία δομή επιλογής ;



2. Ποιο είναι το πλεονέκτημα της χρήσης βρόχου επανάληψης ;
Πότε θα χρησιμοποιήσουμε μία δομή επιλογής ;



Ποια είναι η διαφορά ανάμεσα στις δύο παρακάτω εντολές ;





3. Ποια ήταν η δραστηριότητα που σου φάνηκε περισσότερο διασκεδαστική ;
Τι θα έκανες για να γίνει αυτή η δραστηριότητα πιο συναρπαστική ;

4. Ποια ήταν η δραστηριότητα στην οποία τα κατάφερες καλύτερα ;
Τι θα έκανες για να γίνει αυτή η δραστηριότητα πιο ενδιαφέρουσα ;

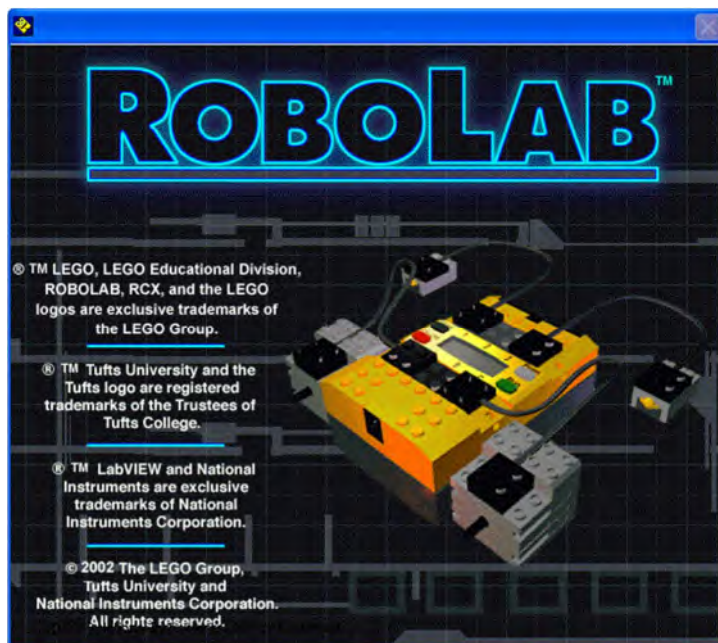
5. Στις δραστηριότητες αυτές, δούλεψες μέσα στην τάξη ως μέλος μιας ομάδας.
Υπήρχε η συνεργασία που περίμενες ;
Θεωρείς ότι ήταν καλή εμπειρία ή θα προτιμούσες να εργαζόσουν μόνος σου ;



Ενότητα 8η

Προγραμματιστικό περιβάλλον ROBOLAB 2.9

Το προγραμματιστικό περιβάλλον Robolab



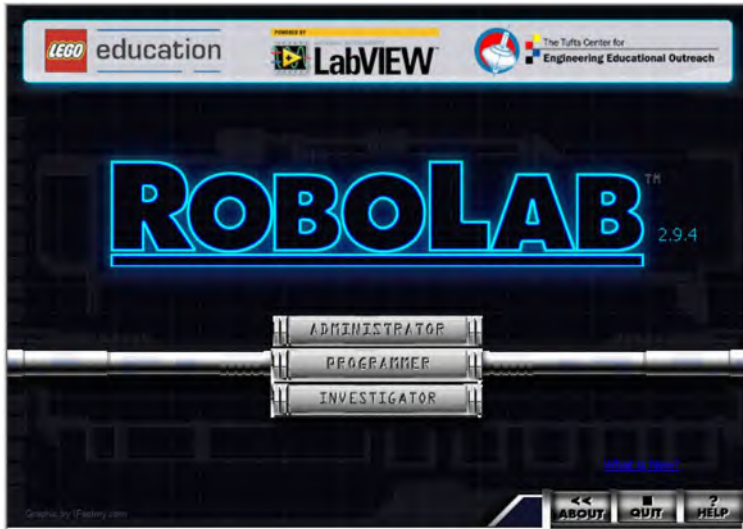
Το Robolab είναι μια γλώσσα προγραμματισμού που χρησιμοποιούμε για να πούμε στο ρομπότ μας τι να κάνει. Γράφουμε το πρόγραμμα μας στον προσωπικό υπολογιστή και μετά το κατεβάζουμε στον επεξεργαστή NXT μέσω σύνδεσης USB ή υπέρυθρης ακτινοβολίας. Το Robolab χρησιμοποιεί εικονίδια, για να αναπαραστήσει τις διαφορετικές λειτουργίες που είναι σε θέση να εκτελέσει ένα ρομπότ εφοδιασμένο με τον επεξεργαστή NXT. Τοποθετώντας αυτά τα εικονίδια με μια συγκεκριμένη σειρά και ενώνοντας τα μεταξύ τους, έχουμε το επιθυμητό πρόγραμμα.

Το λογισμικό Robolab μπορείτε να το κατεβάσετε από την ιστοσελίδα :
<http://www.legoengineering.com/robolab-for-labview/>

Εκκίνηση του προγράμματος

Το πρόγραμμα Robolab είναι ένα οπτικό περιβάλλον προγραμματισμού στο οποίο οι εντολές έχουν τη μορφή εικονιδίων.

Όταν εκτελέσουμε το πρόγραμμα Robolab εμφανίζεται η αρχική καρτέλα του προγράμματος :



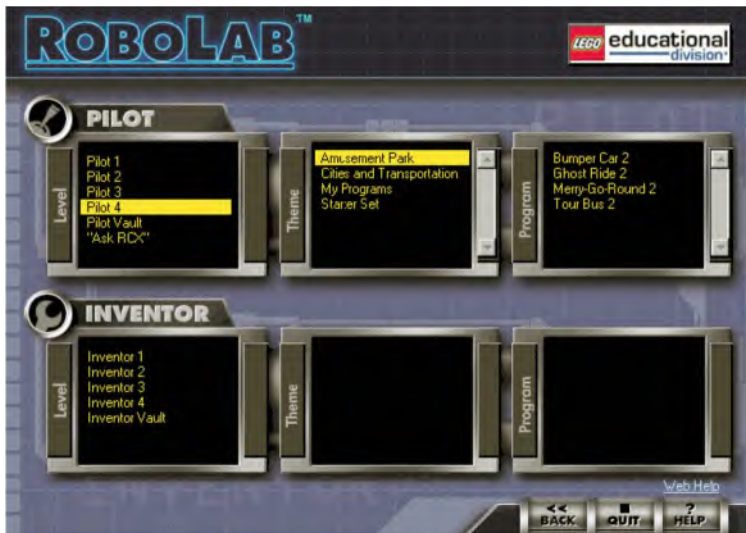
Υπάρχουν οι παρακάτω επιλογές :

Administrator: Περιλαμβάνει τα εργαλεία του διαχειριστή. Μπορούμε να κάνουμε μια σειρά ρυθμίσεων.

Programmer : Μας πηγαίνει στη οθόνη του Αρχικού Μενού για να δημιουργήσουμε το δικό μας πρόγραμμα, με το οποίο θα καθοδηγήσουμε το NXT.

Investigator : Μας δίνει τη δυνατότητα να κάνουμε μια δική μας ερευνητική εργασία ως μικροί επιστήμονες. Μπορούμε να προγραμματίσουμε το NXT, έτσι ώστε πέρα από το να ελέγχουμε τη συμπεριφορά του, να του δώσουμε οδηγίες για να συλλέξει μια σειρά από δεδομένα, τα οποία στη συνέχεια μπορούμε να επεξεργαστούμε κατάλληλα και να καταλήξουμε σε συγκεκριμένα συμπεράσματα για ένα φυσικό φαινόμενο.

Πατώντας το πλήκτρο *Programmer* ανοίγει η καρτέλα του Αρχικού Μενού του προγραμματιστικού περιβάλλοντος Robolab :



Το πρόγραμμα μας δίνει δύο **επιλογές (τρόπους) προγραμματισμού** για να ελέγξουμε τη συμπεριφορά του NXT :

Pilot : απευθύνεται σε αρχάριους χρήστες και μπορεί να χρησιμοποιηθεί από παιδιά που παρακολουθούν την πρωτοβάθμια εκπαίδευση.

Inventor : μας δίνει μεγαλύτερη δυνατότητα ελευθερίας και αξιοποίησης των δυνατοτήτων του NXT.

Και οι δύο τρόποι προγραμματισμού περιλαμβάνουν 4 επίπεδα προγραμματισμού (Level), το καθένα από τα οποία μας δίνει ολοένα και περισσότερες δυνατότητες προγραμματισμού ως προς το προηγούμενο. Το επίπεδο 1 είναι το πιο απλό ενώ το 4 μας δίνει πλήρη ευελιξία και απεριόριστες δυνατότητες.

Κάνοντας κλικ σε ένα από τα επίπεδα αυτά, το πρόγραμμα μας παρέχει τις **κατηγορίες – θέματα (Theme)** των ήδη αποθηκευμένων αρχείων με οδηγίες για το NXT. Υπάρχουν κάποιες ήδη έτοιμες κατηγορίες, αλλά μπορούμε και εμείς να δημιουργήσουμε νέες ή να τροποποιήσουμε αυτές που υπάρχουν.

Κάνοντας κλικ σε ένα από τα θέματα αυτά, μας εμφανίζονται τα αποθηκευμένα αρχεία προγραμμάτων για το NXT αυτής της κατηγορίας.

Τέλος κάνοντας διπλό κλικ σε κάποιο από τα αρχεία αυτά το ανοίγουμε ώστε να μπορούμε να το επεξεργαστούμε.

Κάνοντας διπλό κλικ κατευθείαν σε ένα επίπεδο προγραμματισμού ανοίγει το προγραμματιστικό περιβάλλον για να δημιουργήσουμε ένα νέο πρόγραμμα για το NXT.

Στο κάτω μέρος και δεξιά από τις αρχικές καρτέλες του προγράμματος εμφανίζονται τα πλήκτρα *Back*, *Quit* και *Help*.

Πατώντας το πλήκτρο *Back* επιστρέφουμε στην προηγούμενη καρτέλα. Πατώντας το πλήκτρο *Help* εμφανίζεται ένα παράθυρο στο οποίο μας δίνεται βοήθεια για κάποιο σημείο που θα υποδείξουμε με το ποντίκι.



Σενάρια Ρομποτικής με το Mindstorms NXT

Για να κλείσουμε το πρόγραμμα Robolab αφού κλείσουμε τα παράθυρα με τα ανοικτά προγράμματα που επεξεργαζόμαστε, κάνουμε κλικ στο κουμπί *Quit* και στη νέα καρτέλα που θα εμφανιστεί επιλέγουμε πάλι το πλήκτρο *Quit*. Αν έχουμε μετανιώσει μπορούμε να επιστρέψουμε πίσω πατώντας το πλήκτρο *Back*.

Ρυθμίσεις στο πρόγραμμα Robolab

Για να ρυθμίσουμε το πρόγραμμα Robolab επιλέγουμε το πλήκτρο *Administratort* στην αρχική καρτέλα του προγράμματος. Αρχικά εμφανίζεται το παρακάτω παράθυρο στο οποίο πρέπει να επιλέξουμε τον τύπο του μικροεπεξεργαστή που χρησιμοποιούμε:



Η έκδοση 2.9 του προγράμματος Robolab (Η προηγούμενη έκδοση 2.5 υποστηρίζει μόνο τους μικροεπεξεργαστές NXT) είναι σχεδιασμένη να υποστηρίζει τους μικροεπεξεργαστές από όλες τις σειρές της Lego Mindstorms : Scout, NXT και NXT.

Στην περίπτωση το πρόγραμμα Robolab δεν μπορέσει να βρει το μικροπολογιστή NXT για κάποιο λόγο, εμφανίζεται μία καρτέλα με ένα μήνυμα λάθους στην οποία αναφέρεται το πρόβλημα που υπάρχει και προτείνονται κάποιες πιθανές λύσεις.

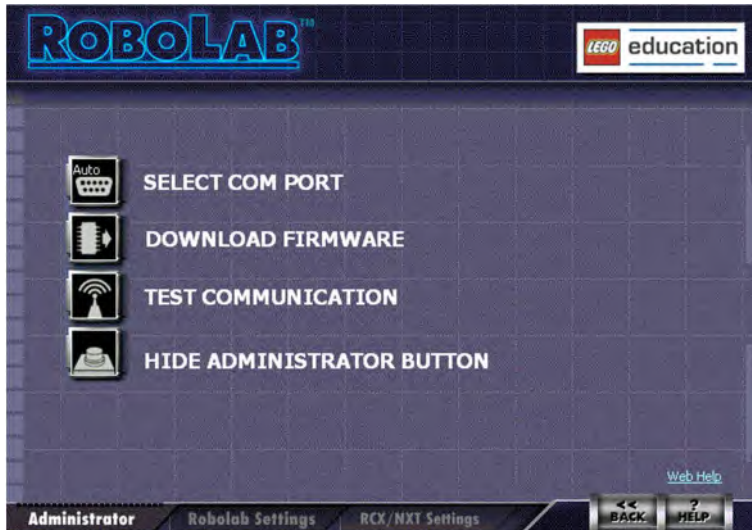


Σενάρια Ρομποτικής με το Mindstorms NXT

Αφού επιλέξουμε τον τύπο του μικροεπεξεργαστή που θέλουμε να προγραμματίσουμε αμέσως εμφανίζεται ένα ταμπλό με τρεις διαθέσιμες καρτέλες :

1. **Administrator**
2. **Robolab Settings**
3. **NXT/NXT Settings**

1. Στην καρτέλα *Administrator* εμφανίζονται οι παρακάτω επιλογές :



Select Com port : επιλέγουμε τη θύρα USB στην οποία έχουμε συνδέσει το μικροϋπολογιστή NXT. Είναι συνήθως βολικό να τοποθετήσουμε τη ρύθμιση αυτή στην αυτόματη ανίχνευση *autodetect*. Αν όμως γνωρίζουμε με βεβαιότητα σε ποια θύρα είναι συνδεδεμένος ο μικροϋπολογιστής μας δηλώνουμε το όνομα της θύρας αυτής.



Download Firmware : μπορούμε να κατεβάσουμε το λογισμικό firmware στο NXT μας για την έκδοση του Robolab που χρησιμοποιούμε. Το firmware είναι ένα μικρό σε μέγεθος ειδικό πρόγραμμα το οποίο είναι υπεύθυνο για τη λειτουργία του μικροϋπολογιστή NXT και το οποίο αποκωδικοποιεί τα προγράμματα που δημιουργούμε. Για κάθε διαφορετικό προγραμματιστικό περιβάλλον απαιτείται να κατεβάσουμε και το αντίστοιχο firmware. Η εργασία αυτή χρειάζεται να γίνει μόνο την πρώτη φορά που θα χρησιμοποιήσουμε το μικροεπεξεργαστή NXT με ένα συγκεκριμένο προγραμματιστικό περιβάλλον. Θα τη χρειαστούμε επίσης και όταν ο μικροϋπολογιστής NXT έχει μείνει για μεγάλο διάστημα χωρίς μπαταρίες οπότε υπάρχει περίπτωση το firmware να σβηστεί ή σε σπάνια περίπτωση όταν ο μικροεπεξεργαστής NXT μπλοκάρει και συμβεί το firmware να καταστραφεί.

Χρειάζεται ιδιαίτερη προσοχή, όταν κατεβάζουμε ένα firmware να μη διακόψουμε τη διαδικασία μεταφόρτωσης κατά λάθος, ώστε αυτή να μπορέσει να ολοκληρωθεί απρόσκοπτα.

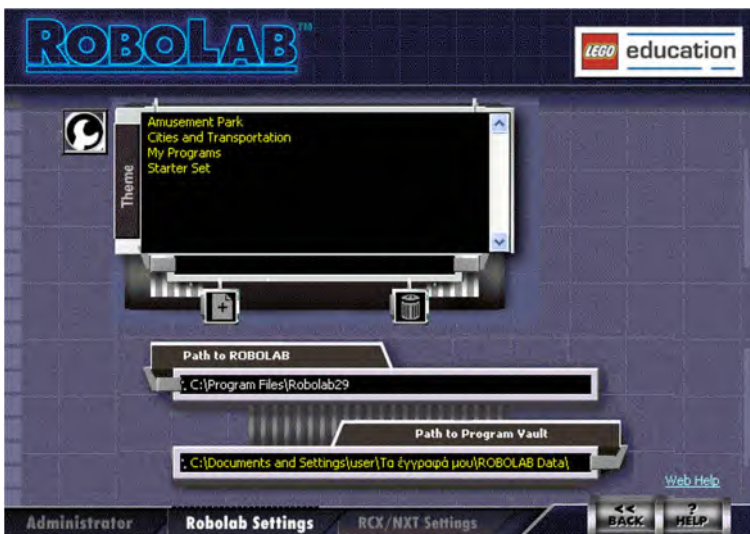
Test Communication : μπορούμε να εξακριβώσουμε ότι ο προσωπικός υπολογιστής στον οποίο τρέχουμε το Robolab μπορεί να επικοινωνήσει χωρίς πρόβλημα με το NXT. Σε διαφορετική περίπτωση θα εμφανιστεί ένα μήνυμα όπου θα περιγράφεται το πρόβλημα που εφίσταται.

Σενάρια Ρομποτικής με το Mindstorms NXT

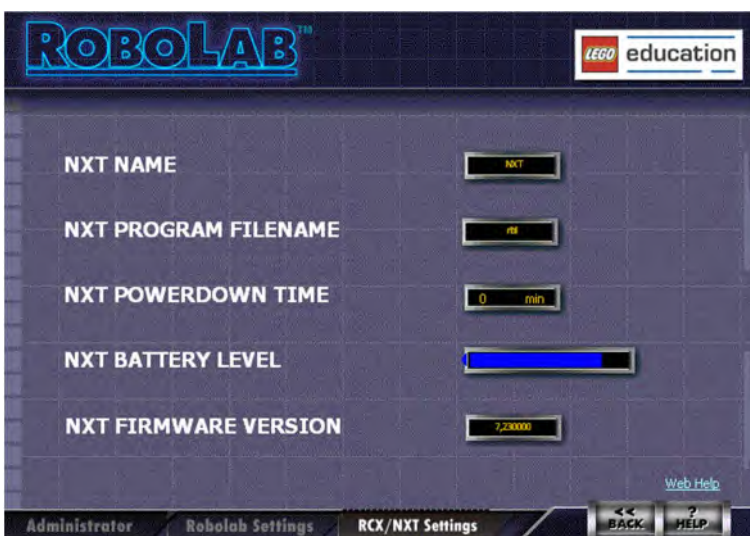
Hide Administrator Button : αφού ολοκληρώσουμε τις ρυθμίσεις που επιθυμούμε μπορούμε να επιλέξουμε να μην εμφανίζεται το πλήκτρο *Administrator* στην αρχική καρτέλα του προγράμματος ώστε να μην έχει πρόσβαση κάποιος τρίτος και πειράξει τις ρυθμίσεις αυτές. Είναι χρήσιμη επιλογή για την περίπτωση που το πρόγραμμα Robolab χρησιμοποιείται και από κάποιον αρχάριο χρήστη. Συνιστάται στην περίπτωση εκπαιδευτικών δραστηριοτήτων.

Για να επανεμφανίσουμε την επιλογή *Administrator* (ή και για να την αποκρύψουμε) χρειάζεται να πατήσουμε το ειδικό πλήκτρο F5 από το πληκτρολόγιο.

2. Από την καρτέλα *Robolab Settings* μπορούμε να διαχειριστούμε τους φακέλους που θα χρησιμοποιεί το πρόγραμμά μας, καθώς και τις κατηγορίες με τα προγράμματα μας ή αλλιώς τα θέματα. Από εδώ μπορούμε να δημιουργήσουμε νέα θέματα ή να τροποποιήσουμε αυτά που ήδη υπάρχουν.



3. Στην καρτέλα *NXT/NXT Settings* εμφανίζονται οι παρακάτω ρυθμίσεις για το τούβλο NXT καθώς και πληροφορίες για την κατάσταση του :



Σενάρια Ρομποτικής με το Mindstorms NXT

NXT Name : Μπορούμε να δούμε και να αλλάξουμε το όνομα του NXT.

NXT Program Filename : Μπορούμε να δούμε και να αλλάξουμε το όνομα με το οποίο θα αποθηκευτεί το πρόγραμμα μας στο NXT.

NXT Powerdown Time : Για λόγους εξοικονόμησης ενέργειας μπορούμε να ρυθμίσουμε το χρόνο, μετά την πάροδο του οποίου και αφού ο μικροϋπολογιστής NXT έχει παραμείνει αδρανής, να διακόπτεται αυτόματα η τροφοδοσία του.

NXT Battery Level : Εμφανίζεται το επίπεδο ισχύος των μπαταριών. Όταν έχει πέσει αρκετά θα πρέπει να φροντίσουμε για την αντικατάστασή τους.

NXT Firmware Version : Εμφανίζεται η έκδοση του προγράμματος firmware που διαθέτει ο μικροϋπολογιστής NXT.

Γνωριμία με το πρόγραμμα Robolab

Κάνοντας διπλό κλικ στο επίπεδο προγραμματισμού Inventor 4 ανοίγει το προγραμματιστικό περιβάλλον για να δημιουργήσουμε ένα νέο πρόγραμμα για το μικροϋπολογιστή NXT :

Παράθυρο κεντρικού πίνακα - Panel Window

Όλες οι λειτουργίες του μπορούν να πραγματοποιηθούν και από το παράθυρο Σχεδίασης. Κλείνοντας αυτό το παράθυρο κλείνουν και τα παράθυρα σχεδίασης και παλέτας εντολών και επιστρέφουμε στο κεντρικό μενού.

Κουμπί εκτέλεσης του προγράμματος μας - Run Button

Ελέγχει αν υπάρχει κάποιο συντακτικό λάθος στο πρόγραμμα μας και στη συνέχεια το στέλνει στο NXT μέσω του πύργου υπερύθρων.

Παράθυρο Σχεδίασης - Diagram Window

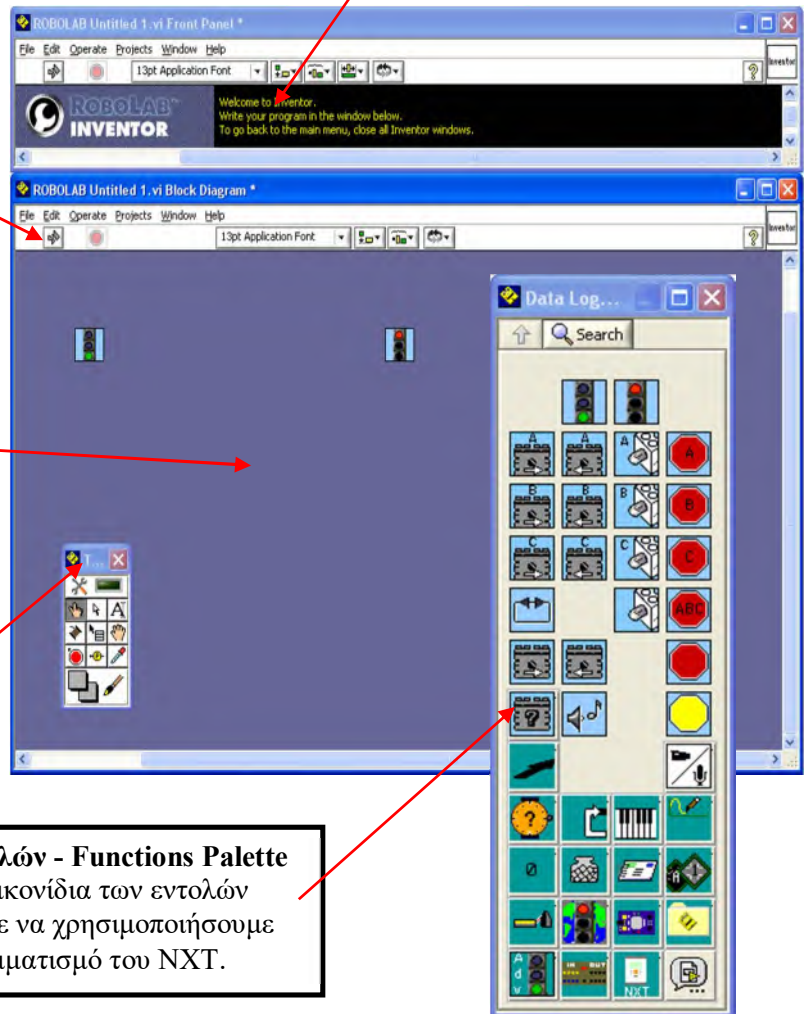
Στο παράθυρο αυτό δημιουργούμε το πρόγραμμα μας

Παλέτα εργαλείων - Tools Palette

Περιέχει τα εργαλεία για να δημιουργήσουμε το πρόγραμμα μας.

Παλέτα εντολών - Functions Palette

Περιέχει τα εικονίδια των εντολών που μπορούμε να χρησιμοποιήσουμε στον προγραμματισμό του NXT.



Ο προγραμματισμός στο Inventor γίνεται μετακινώντας μια σειρά από εικονίδια - τα οποία αντιστοιχούν σε κάποια εντολή - από την παλέτα εντολών μέσα στο παράθυρο Σχεδίασης.

Καλωδιώνοντας (ενώνοντας με γραμμές δηλ.) όλα τα εικονίδια μεταξύ τους, δημιουργούμε το δικό μας πρόγραμμα.

Δραστηριότητες

(για εξοικείωση με το ROBOLAB)

Κάντε κλικ οπουδήποτε στο παράθυρο κεντρικού πίνακα. Η παλέτα Εντολών εξαφανίζεται. Για να ξαναεμφανιστεί η παλέτα Εντολών, κάντε κλικ οπουδήποτε στο παράθυρο Σχεδίασης.

Κλείστε την παλέτα Εντολών. Για να ξαναεμφανιστεί τώρα η παλέτα Εντολών επιλέξτε το μενού *Windows* από το παράθυρο Σχεδίασης και κάντε κλικ στην επιλογή *Show Function Palette*. Εναλλακτικά κάντε δεξί κλικ σε ένα σημείο του παραθύρου Σχεδίασης και κλικ στην καρφίτσα του παραθύρου που θα εμφανιστεί.

Για να εμφανίσετε την παλέτα Εργαλείων επιλέξτε το μενού *Windows* από το παράθυρο Σχεδίασης και κάντε κλικ στην επιλογή *Show Tools Palette*.

Παλέτα Εργαλείων

Για να μπορέσουμε να κάνουμε τις εργασίες μας στο Robolab χρησιμοποιούμε τα εργαλεία της παλέτας εργαλείων :



Εργαλείο Απόθεσης : Χρησιμοποιείτε για να μεταφέρουμε τα εικονίδια εντολών από την παλέτα εντολών στο παράθυρο Σχεδίασης.



Εργαλείο Καλωδίωσης : Χρησιμοποιείτε για να συνδέσουμε (καλωδιώσουμε) τα εικονίδια εντολών το ένα με το άλλο.



Εργαλείο Επιλογής : Χρησιμοποιείτε για να επιλέξουμε και να μετακινήσουμε τα εικονίδια εντολών.



Εργαλείο Κειμένου : Χρησιμοποιείτε για να προσθέσουμε και να επεξεργαστούμε κείμενο ή επεξηγηματικά σχόλια μέσα στο πρόγραμμα.



Εργαλείο αλλαγής τιμής : Χρησιμοποιείτε για να αλλάξουμε αριθμητικές τιμές.

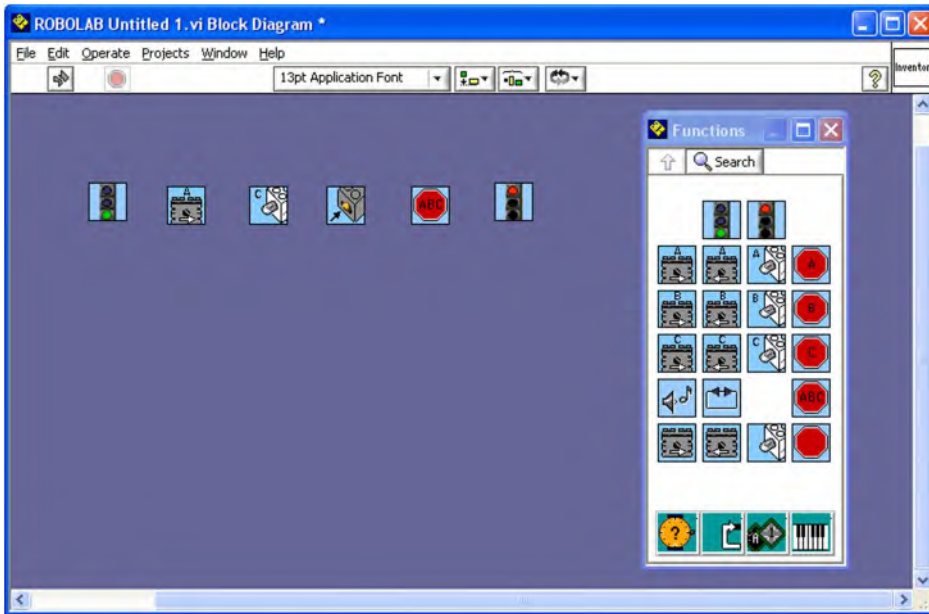
Δραστηριότητες

Πατώντας το πλήκτρο *space bar* από το πληκτρολόγιο μας, γίνεται εναλλαγή ανάμεσα στο εργαλείο επιλογής και καλωδίωσης.

Πατώντας το πλήκτρο *tab* από το πληκτρολόγιο μας, γίνεται εναλλαγή ανάμεσα στα εργαλεία επιλογής, αλλαγής αριθμητικής τιμής, κειμένου και καλωδίωσης.

Διαδικασία για τη δημιουργία ενός προγράμματος στο inventor :

1. Επιλέγουμε από την παλέτα εντολών τα εικονίδια που χρειαζόμαστε και τα τοποθετούμε στο παράθυρο σχεδίασης με τη σειρά που μας εξυπηρετεί.



2. Τακτοποιούμε τα εικονίδια που τοποθετήσαμε στο παράθυρο σχεδίασης.

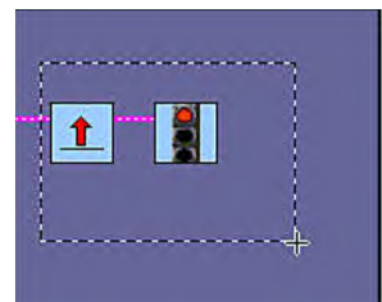
3. Απομακρύνουμε οποιαδήποτε εικονίδια εντολών τυχόν δε χρειαζόμαστε.

Δραστηριότητες

Με το **εργαλείο επιλογής** μπορούμε να κάνουμε τις παρακάτω εργασίες :

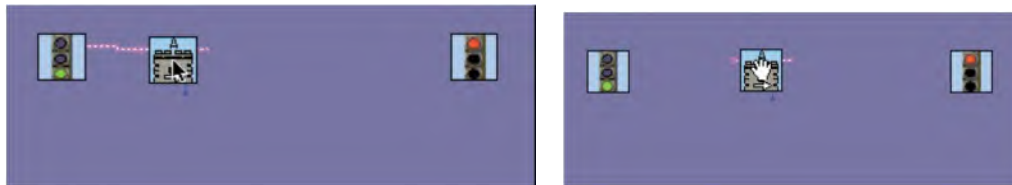
- Να μετακινήσουμε ένα εικονίδιο σε άλλη θέση. Επιλέγουμε και σέρνουμε το εικονίδιο στη θέση που επιθυμούμε
- Να αντικαταστήσουμε ένα εικονίδιο με ένα άλλο. Κάνουμε δεξί κλικ πάνω στο εικονίδιο και επιλέγουμε *Replace*.
- Να διαγράψουμε ένα εικονίδιο. Επιλέγουμε το εικονίδιο και πατάμε το πλήκτρο *Delete* από το πληκτρολόγιο μας.
- Να αντιγράψουμε ένα εικονίδιο σε άλλη θέση. Επιλέγουμε το εικονίδιο και κάνουμε αντιγραφή και επικόλληση από το μενού *Edit*.

Σημείωση: Με το εργαλείο επιλογής μπορούμε να επιλέξουμε και μια ολόκληρη περιοχή η οποία να περιλαμβάνει αρκετά εικονίδια.



4. Καλωδιώνουμε τα εικονίδια εντολών μεταξύ τους.

Όταν τοποθετούμε ένα εικονίδιο εντολών στο παράθυρο σχεδίασης δίπλα σε ένα άλλο, που είχαμε τοποθετήσει κάποια προηγούμενη στιγμή, το εικονίδιο καλωδιώνεται (ενώνεται με το προηγούμενο του) αυτόματα.



Αν όμως αυτό δε συμβεί, τότε χρησιμοποιούμε το εργαλείο καλωδίωσης. Ένας εναλλακτικός τρόπος καλωδίωσης είναι να επιλέξουμε πρώτα το εικονίδιο και μετά να πατήσουμε ταυτόχρονα το πλήκτρο *space bar* από το πληκτρολόγιο μας.

Τα εικονίδια εντολών έχουν συγκεκριμένα σημεία σύνδεσης στα οποία πρέπει να συνδέσουμε την καλωδίωση μας. Για να ενώσουμε δύο εικονίδια ξεκινάμε από το σημείο σύνδεσης στη δεξιά πλευρά του πρώτου εικονιδίου και τερματίζουμε στο σημείο σύνδεσης στην αριστερή πλευρά του άλλου εικονιδίου.



Για να κάνουμε την καλωδίωση να πάρει μια στροφή 90 μοιρών, ενώ τη δημιουργούμε, κάνουμε κλικ στο σημείο του παραθύρου σχεδίασης στο οποίο θέλουμε να στρίψει η καλωδίωση και συνεχίζουμε με το επόμενο σημείο **σύνδεσης**.



Για να διαγράψουμε μία λαθεμένη καλωδίωση, πρώτα την επιλέγουμε με το εργαλείο επιλογής και στη συνέχεια πατάμε το πλήκτρο *Delete* από το πληκτρολόγιο μας. Αν γίνει σωστά η καλωδίωση δύο εικονιδίων εμφανίζεται μια κόκκινη γραμμή, διαφορετικά σε περίπτωση εσφαλμένης καλωδίωσης η γραμμή εμφανίζεται ως διακεκομμένη με μαύρο χρώμα.



Για να απομακρύνουμε τυχόν υπολείμματα λανθασμένων καλωδιώσεων ανοίγουμε το μενού *Edit* από το παράθυρο Σχεδίασης και κάνουμε κλικ στην επιλογή *Remove Broken Wires*.

Μέχρι να ολοκληρωθεί η καλωδίωση σωστά για όλα τα εικονίδια, το κουμπί της εκτέλεσης του προγράμματος (το οποίο στέλνει το πρόγραμμα στο NXT) εμφανίζεται ως ένα σπασμένο βέλος :



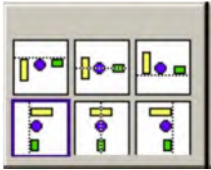
5. Ελέγχουμε το πρόγραμμα για συντακτικά λάθη και αν υπάρχουν τα διορθώνουμε.

6. Στοιχίζουμε τα εικονίδια ώστε να έχουμε ένα καλαίσθητο αποτέλεσμα (προαιρετικά)

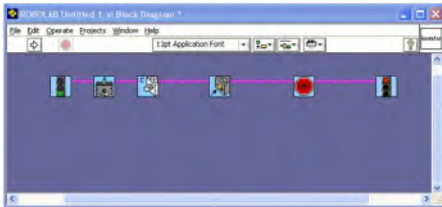
Για να στοιχίσουμε τα εικονίδια εφαρμόζουμε την ευθυγράμμιση και την ομοιόμορφη κατανομή τους :

- Ευθυγράμμιση (align)

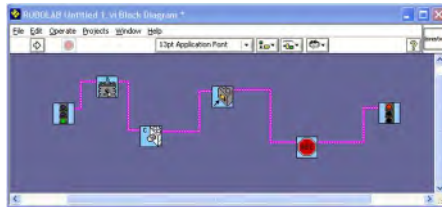
Για να ευθυγραμμίσουμε τα εικονίδια μας πατάμε στο πλήκτρο  και επιλέγουμε έναν από τους 6 προκαθορισμένους τρόπους ευθυγράμμισης :



Το αποτέλεσμα πριν και μετά την οριζόντια ευθυγράμμιση φαίνεται στις παρακάτω εικόνες :



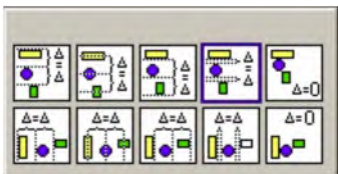
Πριν την ευθυγράμμιση



Μετά την ευθυγράμμιση

- Κατανομή (distribute)

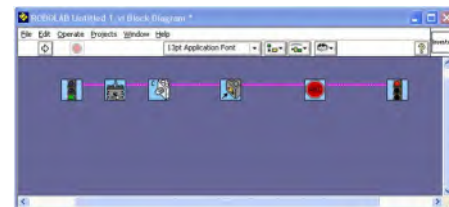
Για να κατανείμουμε ομοιόμορφα τα εικονίδια μας πατάμε στο πλήκτρο  και επιλέγουμε έναν από τους 10 προκαθορισμένους τρόπους κατανομής :



Το αποτέλεσμα πριν και μετά την οριζόντια κατανομή φαίνεται στις παρακάτω εικόνες :



Πριν την εφαρμογή κατανομής

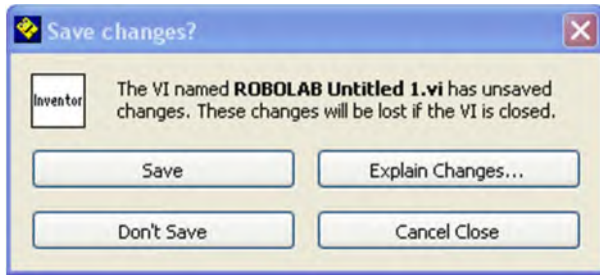


Μετά την εφαρμογή κατανομής

7. Σώζουμε το πρόγραμμα που επεξεργαζόμαστε.

Για να σώσουμε μία εργασία μας ανοίγουμε το μενού *File* από το παράθυρο Σχεδίασης και κάνουμε κλικ στην επιλογή *Save As....* Στο παράθυρο που εμφανίζεται γράφουμε το όνομα, που επιθυμούμε για την εργασία μας και καθορίζουμε τη θέση στη μονάδα αποθήκευσης μας, στην οποία θέλουμε να αποθηκευτεί αυτή.

Όταν κλείσουμε το παράθυρο του προγραμματιστικού περιβάλλοντος inventor, ενώ προηγουμένως είχαμε προχωρήσει σε κάποιες αλλαγές στην εργασία μας, εμφανίζεται το παρακάτω παράθυρο, το οποίο μας προτρέπει να σώσουμε τις αλλαγές που έχουμε κάνει στο πρόγραμμα που επεξεργαζόμαστε.



8. Κατεβάζουμε το πρόγραμμα στο NXT.

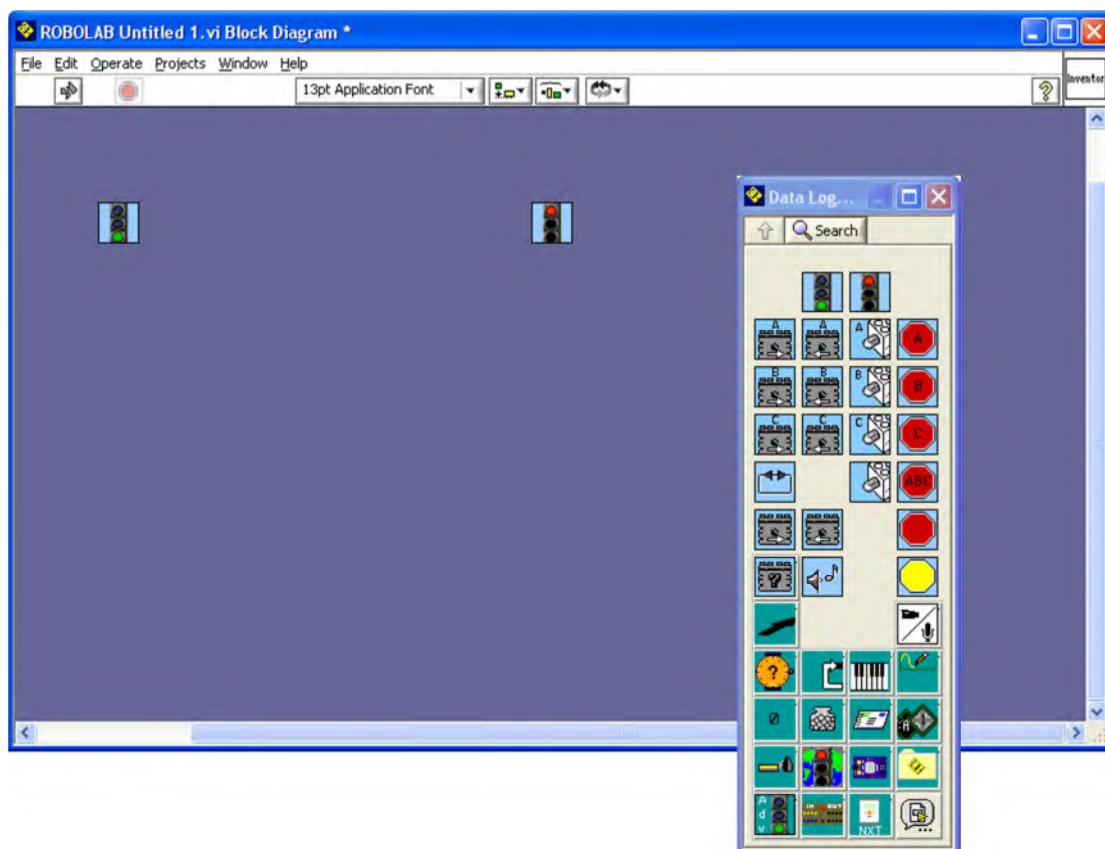
Πατάμε το πλήκτρο 

9. Τρέχουμε το πρόγραμμα από το NXT.



Παλέτες εντολών του Inventor

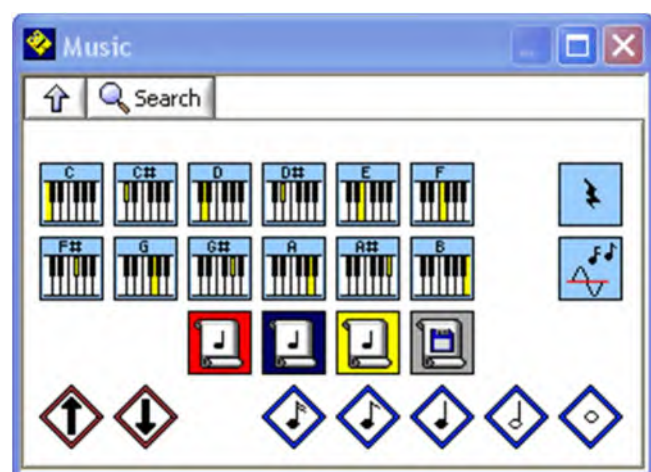
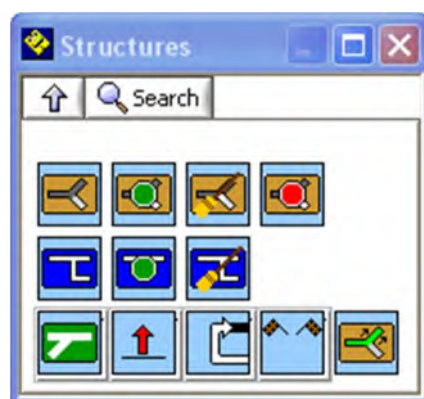
Όταν ανοίγουμε το Robolab στο περιβάλλον προγραμματισμού Inventor 4, εμφανίζεται η παρακάτω εικόνα :

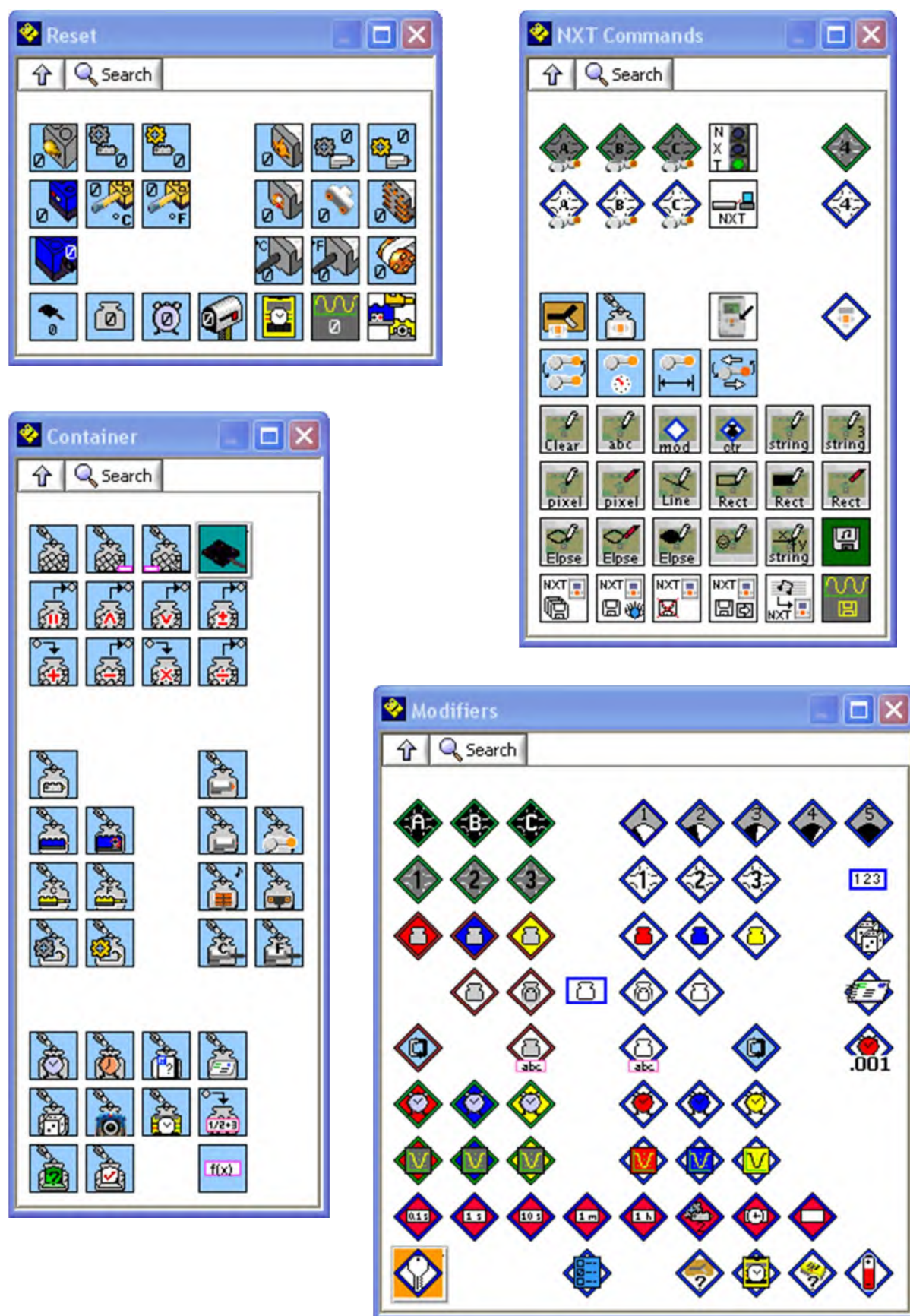


Στην παλέτα εντολών του Inventor 4, υπάρχουν διάφορα υπομενού, τα οποία ανοίγουν όταν επιλέξουμε ένα από τα κουμπιά που βρίσκονται στις τελευταίες σειρές:

- **Αναμονή για (Wait for)** : περιλαμβάνει τα εικονίδια εντολών αναμονής για χρόνο, πίεσης του αισθητήρα αφής και μεταβολής στη φωτεινότητα του αισθητήρα φωτός.
- **Δομών προγραμματισμού (Structures)** : περιλαμβάνει τα εικονίδια εντολών για την υλοποίηση δομών επιλογής, επανάληψης, άλματος και παράλληλων διεργασιών.
- **Μουσικής (Music).**
- **Έρευνας (Investigator)** : περιλαμβάνει τα εικονίδια εντολών για την εκτέλεση πειραματικών μετρήσεων.
- **Μηδενισμού (Reset)** : για το μηδενισμό μιας σειράς παραμέτρων
- **Μεταβλητών (Modifiers)** : περιλαμβάνει τα εικονίδια εντολών που αφορούν στις μεταβλητές.
- **Τροποποιητών (Modifiers)** : περιλαμβάνει τα εικονίδια των τροποποιητών των θυρών εισόδου-εξόδου και της ισχύος.
- **Εντολές για το NXT (NXT Commands)** : περιλαμβάνει τα εικονίδια για τις εντολές που αφορούν το μικροεπεξεργαστή NXT.

κ.α.





Για να επανέλθουμε στη αρχική παλέτα εντολών, κάνουμε κλικ στο κουμπί πάνω αριστερά με το όρθιο βελάκι, στη θυγατρική παλέτα που έχουμε ανοίξει.

Εξοικείωση με το Robolab

Το **RoboLab** είναι ένα προγραμματιστικό περιβάλλον το οποίο το χρησιμοποιούμε για να πούμε στο ρομπότ μας τι να κάνει.

Γράφουμε το πρόγραμμα μας στον προσωπικό υπολογιστή μας και μετά το κατεβάζουμε (κάνουμε downloading) στο μικροεπεξεργαστή NXT του ρομπότ μας, χρησιμοποιώντας τον πύργο υπερύθρων.

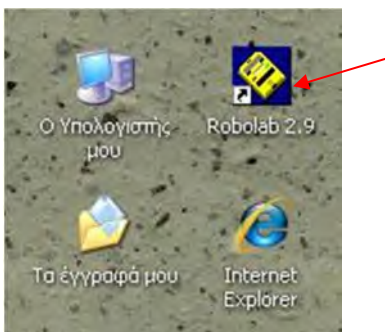
Το RoboLab χρησιμοποιεί εικονίδια (**icons**), τα οποία αναπαριστούν διαφορετικές εργασίες που πρέπει να εκτελέσει το ρομπότ μας. Για να δημιουργήσουμε μια λογική σειρά από εντολές που θα εκτελέσει το ρομπότ μας, βάζουμε τα εικονίδια αυτά σε μία συγκεκριμένη σειρά και τα συνδέουμε μεταξύ τους.

Στις παρακάτω δραστηριότητες θα χρησιμοποιήσουμε το όχημα - ρομπότ tankbot, που έχουμε ήδη κατασκευάσει.

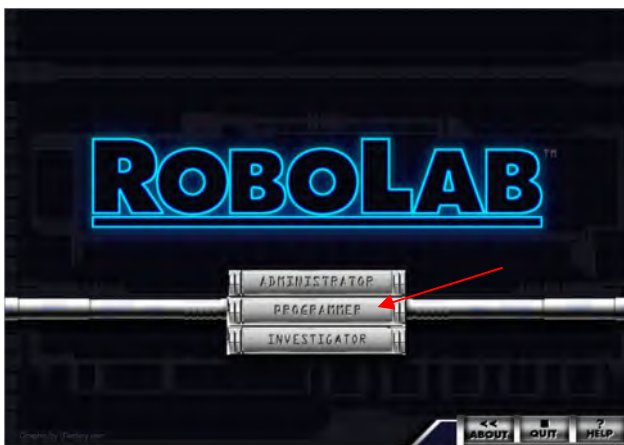
Το όχημα αυτό διαθέτει δύο κινητήρες :

Ο αριστερός κινητήρας κινεί την αριστερή ρόδα και είναι συνδεδεμένος στη θύρα εξόδου A του NXT, ενώ ο δεξιός κινητήρας κινεί τη δεξιά ρόδα και είναι συνδεδεμένος στη θύρα εξόδου C.

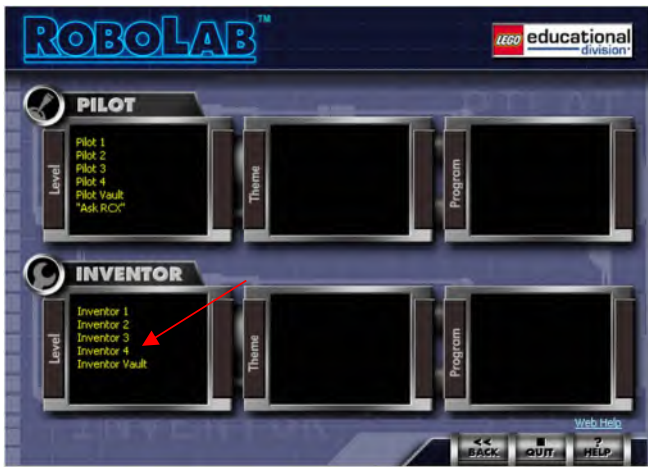
- Ανοίξτε την εφαρμογή RoboLab κάνοντας διπλό κλικ πάνω στη συντόμευση του RoboLab 2.9



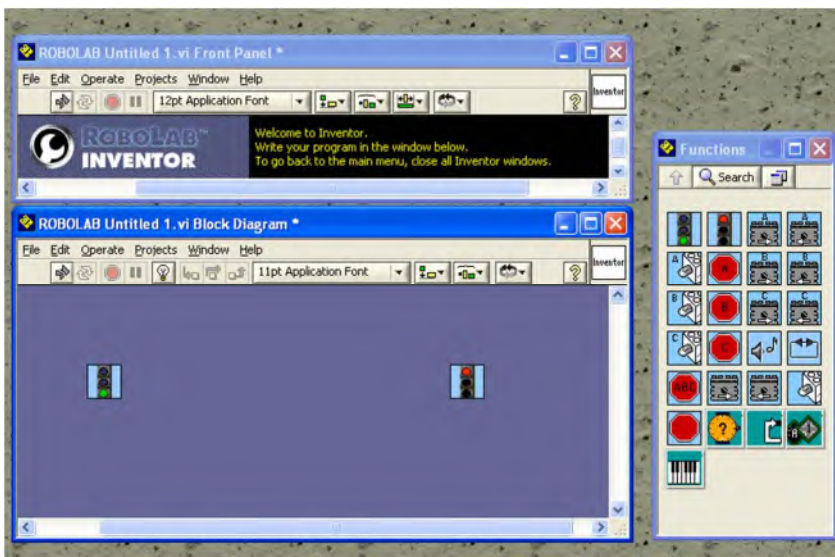
- Κάντε κλικ στο κουμπί *Programmer*.



- Κάντε διπλό κλικ στην επιλογή *Inventor 3* για να ξεκινήσετε ένα νέο πρόγραμμα για το μικροεπεξεργαστή NXT.



Ένα νέο παράθυρο σχεδίασης μοιάζει με την παρακάτω εικόνα. Σε αυτό υπάρχουν ήδη τοποθετημένα ένα εικονίδιο με πράσινο σηματοδότη - το εικονίδιο "**εκκίνηση του προγράμματος**" καθώς και ένα εικονίδιο με κόκκινο σηματοδότη - το εικονίδιο "**τερματισμός του προγράμματος**".



Η παλέτα εντολών περιέχει όλα τα εικονίδια που χρειαζόμαστε για να γράψουμε το δικό μας πρόγραμμα.

Ας εξοικειωθούμε λίγο με το ROBOLAB

Κάντε κλικ οπουδήποτε στο παράθυρο του κεντρικού πίνακα. Η παλέτα Εντολών εξαφανίζεται. Για να ξαναεμφανιστεί η παλέτα Εντολών, κάντε κλικ οπουδήποτε στο παράθυρο Σχεδίασης.

Κλείστε την παλέτα Εντολών. Για να ξαναεμφανιστεί τώρα η παλέτα Εντολών επιλέξτε το μενού *Windows* από το παράθυρο Σχεδίασης και κάντε κλικ στην επιλογή *Show Function Palette*. Εναλλακτικά κάντε δεξί κλικ σε ένα σημείο του παραθύρου Σχεδίασης και κατόπιν κλικ στην εικόνα της καρφίτσας στο παράθυρο που θα εμφανιστεί.

Για να εμφανίσετε την παλέτα Εργαλείων επιλέξτε το μενού *Windows* από το παράθυρο Σχεδίασης και κάντε κλικ στην επιλογή *Show Tools Palette*.

Χρήσιμες συμβουλές

Πατώντας το πλήκτρο *space bar* από το πληκτρολόγιο μας, γίνεται εναλλαγή ανάμεσα στο εργαλείο επιλογής και καλωδίωσης.

Πατώντας το πλήκτρο *tab* από το πληκτρολόγιο μας, γίνεται εναλλαγή ανάμεσα στα εργαλεία επιλογής, αλλαγής αριθμητικής τιμής, κειμένου και καλωδίωσης.

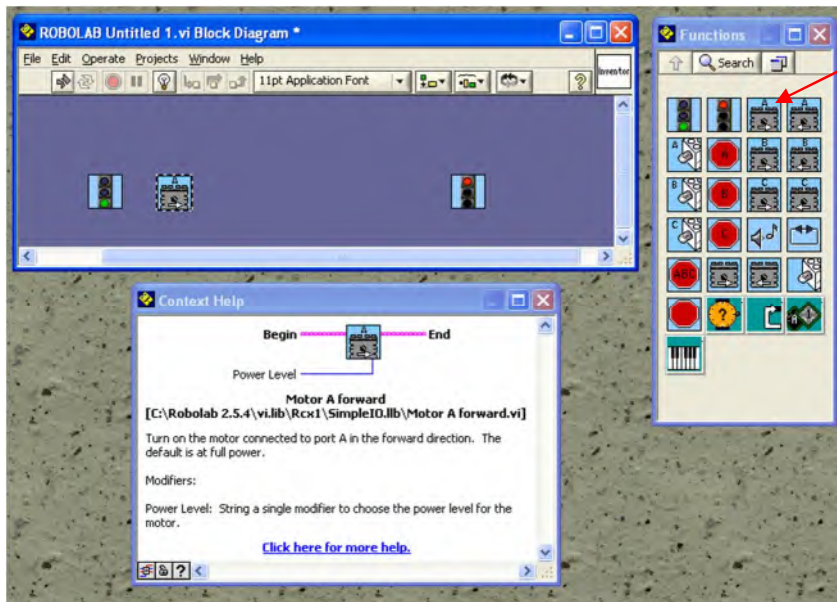
Δραστηριότητα 1η : κίνηση δεξιά για 2s

Προσπαθήστε να καταλάβετε τι αποτέλεσμα θα έχει το παρακάτω πρόγραμμα :



Ας ξεκινήσουμε δημιουργώντας ένα απλό πρόγραμμα. Αυτό το πρόγραμμα θα ξεκινήσει έναν κινητήρα που είναι συνδεδεμένος στη θύρα *A* για 2 δευτερόλεπτα και μετά θα τον σταματήσει. Το πρώτο εικονίδιο που θα χρειαστούμε είναι αυτό που ξεκινάει έναν κινητήρα. Ο κινητήρας μας είναι συνδεδεμένος στη θύρα *A* πάνω στο NXT. Το εικονίδιο που μας χρειάζεται, έχει την εικόνα ενός κινητήρα με το γράμμα *A* πάνω της και ένα λευκό βέλος από κάτω της. Το λευκό βέλος προσδιορίζει προς ποια κατεύθυνση θα γυρίσει ο κινητήρας. Στη δική μας περίπτωση ο κινητήρας θέλουμε να γυρίσει προς τα μπροστά, οπότε θα επιλέξουμε το εικονίδιο με το βελάκι που δείχνει προς τα δεξιά.

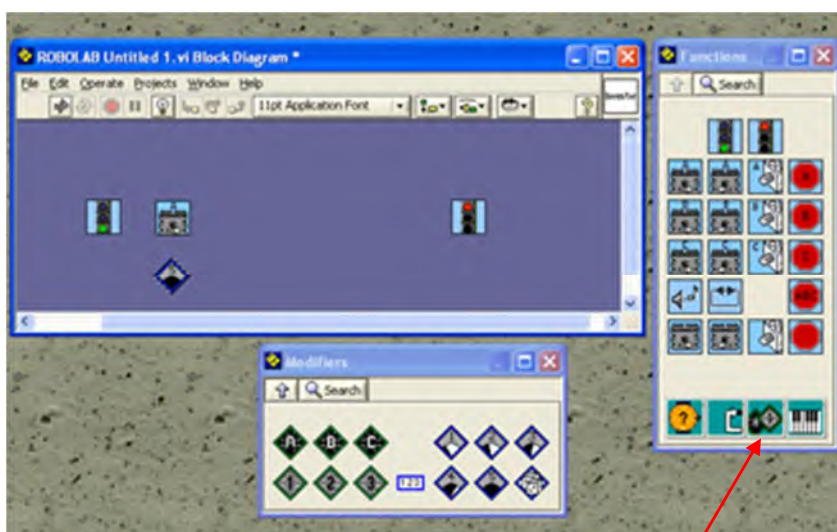
Σενάρια Ρομποτικής με το Mindstorms NXT



- Για να τοποθετήσετε το εικονίδιο μέσα στο πρόγραμμα μας κάντε κλικ στο κατάλληλο εικονίδιο κινητήρα της παλέτας εντολών, μετακινήστε το εικονίδιο στο παράθυρο σχεδίασης του προγράμματός μας, και τοποθετήστε το δίπλα στο εικονίδιο με τον πράσινο σηματοδότη κάνοντας κλικ ξανά.

- Είναι πολύ χρήσιμο να συμβουλευέστε το παράθυρο βοήθειας. Για να εμφανιστεί αυτό κάντε κλικ στο μενού *Help* και επιλέξτε *Show Context Help*. Στη συνέχεια κάντε κλικ στο εικονίδιο για το οποίο θέλετε να μάθετε περισσότερα.

Οι κινητήρες μπορούν να προγραμματιστούν ώστε να περιστρέφονται με διαφορετικές ταχύτητες - **επίπεδο ισχύος (power levels)**, από το επίπεδο 1 (μικρότερη ισχύς) ως το επίπεδο 5 (πλήρης ισχύς). Το εικονίδιο "**επίπεδο ισχύος**" μπορείτε να το βρείτε στην υποπαλέτα **τροποποιητές (modifiers)**, της παλέτας εντολών :



Υποπαλέτα Τροποποιητών

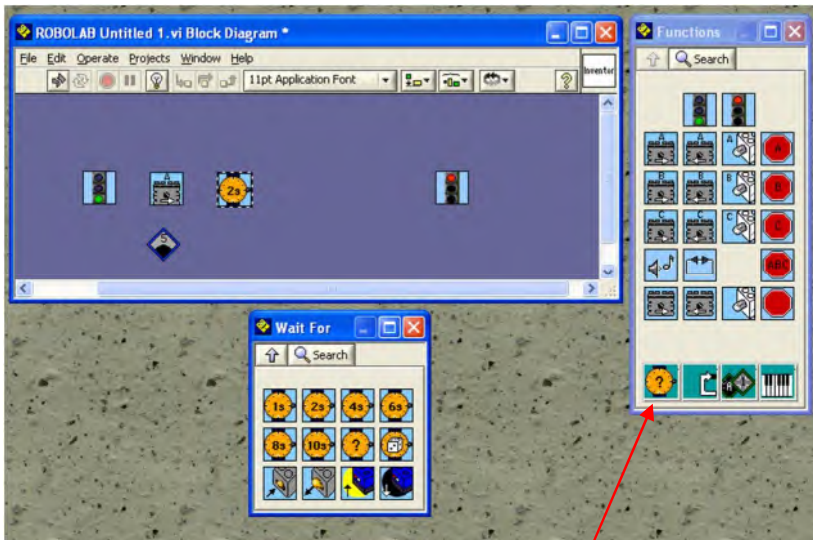
Σενάρια Ρομποτικής με το Mindstorms NXT

- Επιλέξτε το επίπεδο ισχύος 5 και μετακινήστε το εικονίδιο αυτό, στο παράθυρο σχεδίασης κάτω από το εικονίδιο του κινητήρα.

Το επόμενο εικονίδιο που θα χρειαστούμε είναι το εικονίδιο "**αναμονή για χρόνο 2 δευτερολέπτων**", το οποίο βάζει το πρόγραμμα να περιμένει για 2 δευτερόλεπτα προτού συνεχίσει με την εκτέλεση της επόμενης εντολής.

Αυτό το εικονίδιο βρίσκεται στην υποπαλέτα **αναμονής (wait for)** :

- Κάντε κλικ στο εικονίδιο του ρολογιού που γράφει 2s και μετακινήστε το εικονίδιο αυτό στο παράθυρο σχεδίασης, δίπλα από το εικονίδιο του κινητήρα.

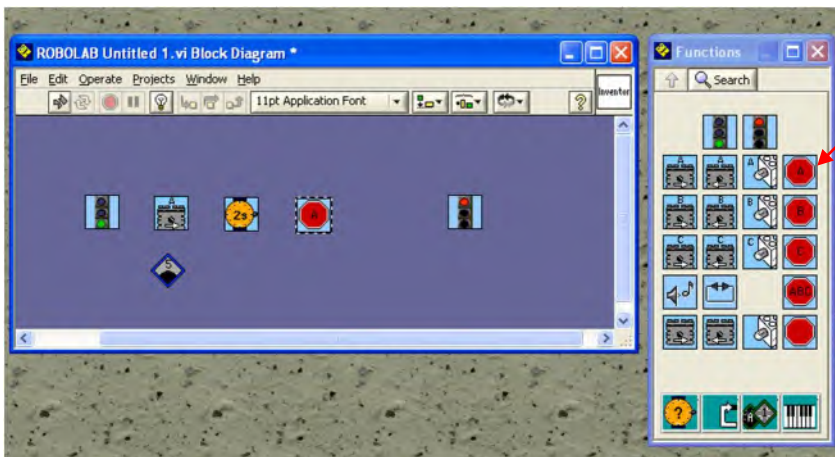


Υποπαλέτα Εντολών Αναμονής

Στο Robolab οι κινητήρες και οι λαμπτήρες δεν σταματούν να λειτουργούν όταν τερματιστεί η εκτέλεση ενός προγράμματος, αλλά συνεχίζουν να τροφοδοτούνται με ισχύ μέχρι να κλείσουμε εμείς, την τροφοδοσία του NXT.

Το τελευταίο εικονίδιο που χρειαζόμαστε λοιπόν είναι αυτό που σταματάει τη λειτουργία ενός κινητήρα. Αυτό το εικονίδιο έχει ως εικόνα το σήμα στοπ :

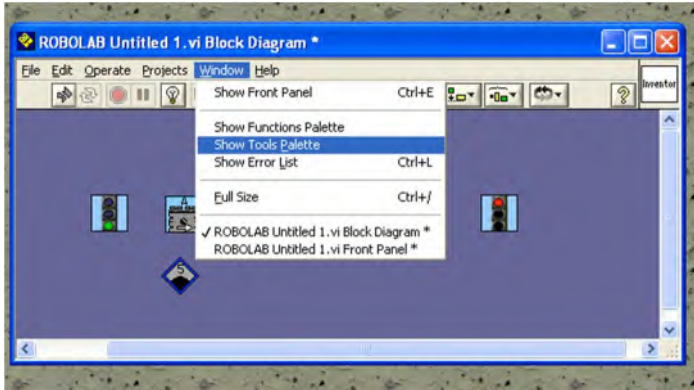
- Κάντε κλικ και μετακινήστε το εικονίδιο με το σήμα στοπ με το γράμμα A πάνω του, στο παράθυρο σχεδίασης, δίπλα στο εικονίδιο της αναμονής.



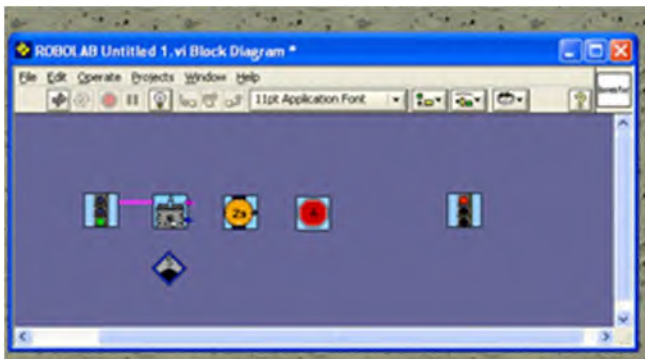
Σενάρια Ρομποτικής με το Mindstorms NXT

Το επόμενο βήμα είναι να ενώσουμε κατάλληλα όλα τα εικονίδια μεταξύ τους. Για να το κάνουμε αυτό χρειαζόμαστε το εργαλείο καλωδίωσης (wire tool) από την παλέτα **εργαλείων (Tools palette)** :

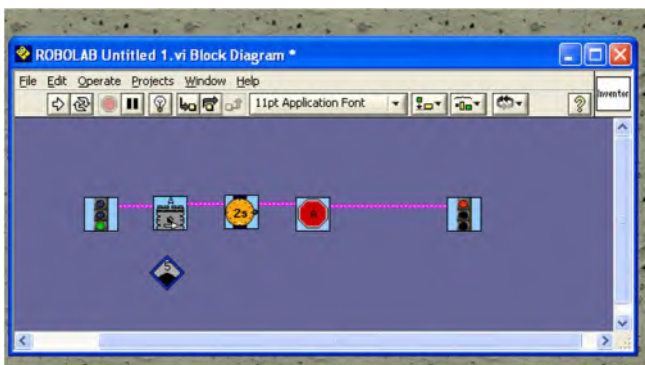
- Για να ενεργοποιήσετε την παλέτα εργαλείων επιλέξτε το *Show Tools Palette* από το μενού *Windows* του παραθύρου σχεδίασης.



- Κατόπιν στην παλέτα εργαλείων που θα εμφανιστεί, κάντε κλικ στο εργαλείο καλωδίωσης.
- Για να ενώσετε τα εικονίδια μεταξύ τους κάντε κλικ στην πάνω δεξιά γωνία του πρώτου εικονιδίου και στη συνέχεια στην πάνω αριστερή γωνία του επόμενου εικονιδίου που θέλετε να συνδέσετε. Ξεκινήστε από το εικονίδιο του πράσινου σηματοδότη και καλωδιώστε το, με το εικονίδιο του κινητήρα.

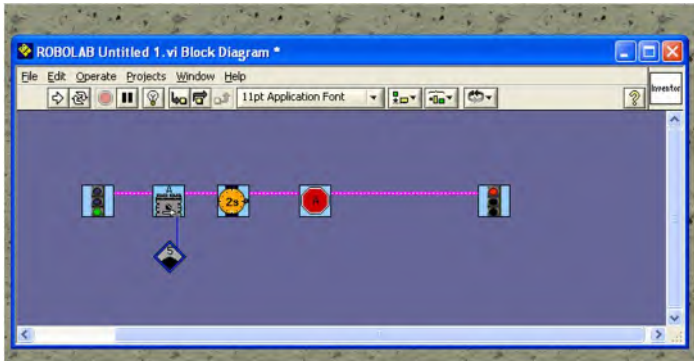


Συνεχίστε με τον ίδιο τρόπο μέχρι και το εικονίδιο του κόκκινου σηματοδότη.



Επίσης χρειάζεται να ενώσουμε τον τροποποιητή του επιπέδου ισχύος με το εικονίδιο του κινητήρα :

- Κάντε κλικ με το εργαλείο της καλωδίωσης στην κάτω δεξιά γωνία του εικονιδίου του κινητήρα και στη συνέχεια στο εικονίδιο του τροποποιητή ισχύος.

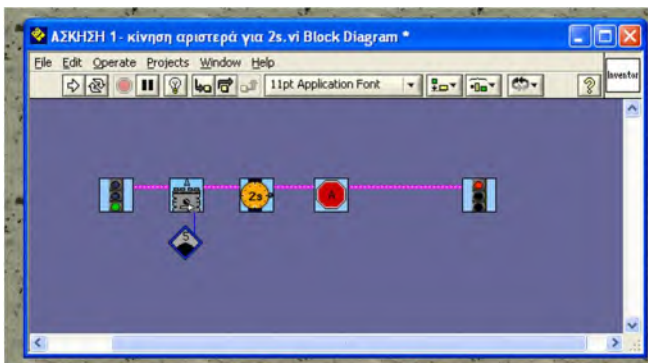


Αποθήκευση του προγράμματος

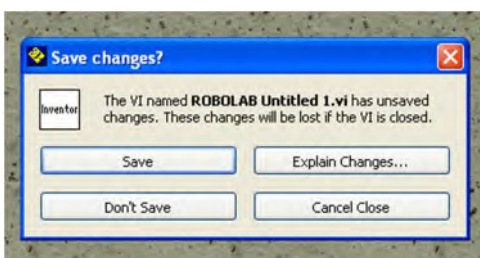
Δεν πρέπει να ξεχνάμε να αποθηκεύουμε την εργασία που κάνουμε ανά τακτά χρονικά διαστήματα και οπωσδήποτε πριν να στείλουμε το πρόγραμμα στο NXT για να το εκτελέσει.

- Για να αποθηκεύσετε το πρόγραμμα που δημιουργήσατε, επιλέξτε το *Save As...* από το μενού *File* του παραθύρου σχεδίασης.

Στο παράθυρο που εμφανίζεται πληκτρολογήστε την πρόταση **κίνηση δεξιά για 2s** ως όνομα για την εργασία σας, καθορίστε το φάκελο **ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ** της επιφάνειας εργασίας ως θέση αποθήκευσης, και πατήστε το κουμπί *Αποθήκευση*.



Προσοχή: Όταν κλείσουμε το παράθυρο του προγραμματιστικού περιβάλλοντος inventor ενώ προηγουμένως είχαμε προχωρήσει σε κάποιες αλλαγές στην εργασία μας, εμφανίζεται το παρακάτω παράθυρο το οποίο μας προτρέπει να σώσουμε τις αλλαγές που έχουμε κάνει στο πρόγραμμα που επεξεργαζόμαστε.

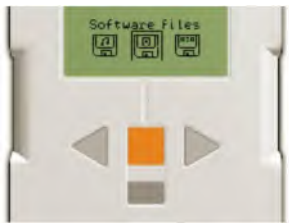


Κατέβασμα του προγράμματος στο NXT και εκτέλεση του

- Συνδέστε το NXT από το ρομπότ με ένα καλώδιο USB στον προσωπικό υπολογιστή.
 - Πατήστε το πορτοκαλί πλήκτρο **On** στο NXT για να το θέσετε σε λειτουργία.
 - Πατήστε το κουμπί με το βέλος , που βρίσκεται στην πάνω δεξιά γωνία του παραθύρου σχεδίασης, για να κατεβάσετε το πρόγραμμα που δημιουργήσατε στο NXT.
- Όταν το πρόγραμμα σας φορτωθεί επιτυχώς στο NXT, αυτό θα σφυρίξει.



- Τρέξτε το πρόγραμμα από το NXT πατώντας το πορτοκαλί πλήκτρο **Enter**



Τι παρατηρείτε ; Πως συμπεριφέρεται το ρομπότ μας ;

Απάντηση :

Το ρομπότ μας εκτελεί μία στροφή δεξιά για 2 δευτερόλεπτα.

Αν θέλουμε το ρομπότ μας να κινηθεί ευθεία μπροστά τι πρέπει να κάνουμε ;

Ανάλυση Προβλήματος

Αυτό το πρόγραμμα λέει στο ρομπότ μας να κινήσει τον κινητήρα Α προς τα μπροστά για 2 δευτερόλεπτα και μετά να σταματήσει τον κινητήρα Α. Από τη στιγμή που το tankbot παίρνει κίνηση από δύο κινητήρες (Α και C) αυτό το πρόγραμμα θα έχει ως αποτέλεσμα, το ρομπότ μας να κινηθεί προς τα δεξιά για 2 δευτερόλεπτα.

Σε ένα όχημα – ρομπότ όπως το tankbot, για να κινηθεί με τον τρόπο που επιθυμούμε, πρέπει να δώσουμε τις κατάλληλες εντολές στον αριστερό και στο δεξιό κινητήρα. Οι κινητήρες μπορούν να οδηγηθούν με διαφορετικές κατευθύνσεις (φορά περιστροφής), με διαφορετική ισχύ (ταχύτητα περιστροφής) και για διαφορετική χρονική διάρκεια.

Για να κινηθεί ένα όχημα σε ευθεία μπροστά ή πίσω, πρέπει να δώσουμε εντολή να κινηθούν ταυτόχρονα και οι δύο κινητήρες μας με την ίδια ισχύ και για ίσο χρονικό διάστημα.

Αν ο ένας κινητήρας (πχ ο αριστερός) περιστραφεί ενώ ο άλλος (πχ ο δεξιός) μείνει ακίνητος, το όχημα μας θα στρίψει (δεξιά στην περίπτωση μας).

Αν ο ένας κινητήρας (πχ ο αριστερός) περιστραφεί με κάποια ισχύ (πχ. 5) και ο άλλος (πχ ο δεξιός) περιστραφεί με μία μικρότερη ισχύ (πχ 2), το όχημα μας θα διαγράψει την πορεία ενός τόξου (προς τα δεξιά στην περίπτωση μας).

Αν ο ένας κινητήρας (πχ ο αριστερός) περιστραφεί με κάποια ισχύ ενώ ο άλλος (πχ ο δεξιός) περιστραφεί με την ίδια ισχύ προς την ανάποδη φορά, το όχημα μας θα κάνει στροφή επί τόπου. (προς τα δεξιά στην περίπτωση μας).

Συγχαρητήρια !!!

Ολοκληρώσατε την 1η δραστηριότητα

Δραστηριότητα 2η : κίνηση μπροστά για 6s

(τροποποίηση προγράμματος)

Για να ξεκινήσουμε τη συγγραφή ενός νέου προγράμματος υπάρχουν όπως είναι γνωστό δύο επιλογές :

1. Να μετονομάσουμε και μετά να επεξεργαστούμε ένα υπάρχον πρόγραμμα.
2. Να ξεκινήσουμε με ένα άδειο φύλλο και να κάνουμε ένα νέο πρόγραμμα από την αρχή.

Στη δραστηριότητα αυτή θα τροποποιήσουμε το προηγούμενο πρόγραμμα ώστε να περιλαμβάνει δύο κινητήρες, οι οποίοι θα περιστρέφονται με την ίδια ισχύ προς τα εμπρός για χρονικό διάστημα 6 δευτερολέπτων.

Άνοιγμα του προγράμματος

Για να ανοίξουμε ένα υπάρχον πρόγραμμα που έχουμε δημιουργήσει, επιλέγουμε το *Open...* από το μενού *File* του παραθύρου σχεδίασης. Στο παράθυρο που εμφανίζεται, καθορίζουμε το φάκελο στον οποίο έχουμε τοποθετήσει το πρόγραμμα, επιλέγουμε το αρχείο του προγράμματος και πατάμε το κουμπί *ok*.

- Ανοίξτε το πρόγραμμα **κίνηση δεξιά για 2s** που δημιουργήσατε προηγουμένως.

Τροποποίηση του προγράμματος

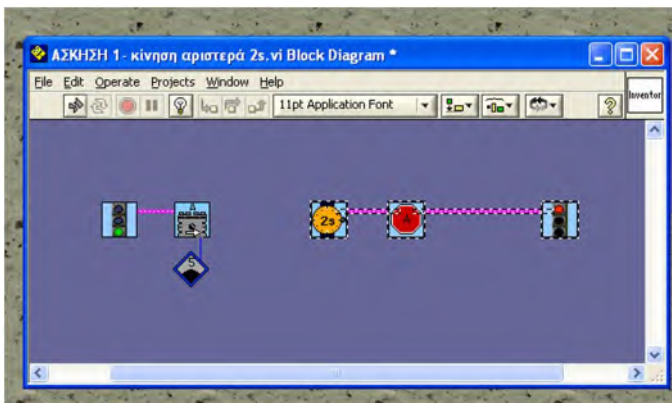
Για να προσθέσουμε ένα επιπλέον εικονίδιο μέσα σε ένα έτοιμο πρόγραμμα πρέπει να δημιουργήσουμε τον απαραίτητο χώρο :

Σενάρια Ρομποτικής με το Mindstorms NXT

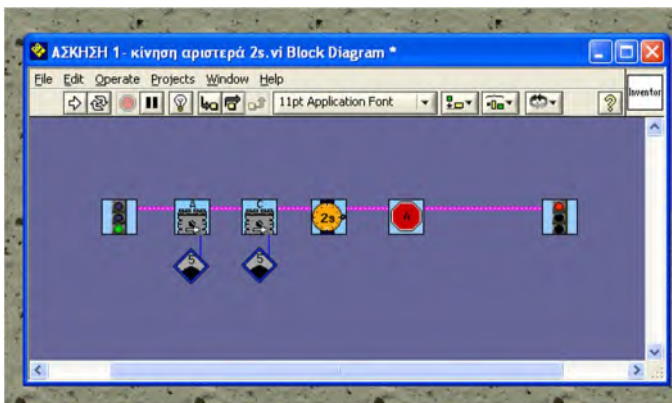
- Διαγράψτε την καλωδίωση ανάμεσα στο εικονίδιο του κινητήρα A και στο εικονίδιο της αναμονής, για να μπορέσετε να τοποθετήσετε ανάμεσα το νέο εικονίδιο. Για να το κάνετε αυτό, επιλέξτε την καλωδίωση και πατήστε το πλήκτρο *delete* από το πληκτρολόγιο.

Στη συνέχεια πρέπει να μεγαλώσουμε το διάστημα ανάμεσα στα εικονίδια, όπου θα τοποθετήσουμε το νέο μας εικονίδιο.

- Με το εργαλείο επιλογής, επιλέξτε τα εικονίδια μετά το εικονίδιο του κινητήρα A μέχρι και το εικονίδιο του κόκκινου σηματοδότη, σέρνοντας το βελάκι του ποντικιού, από ένα σημείο πάνω αριστερά του εικονιδίου της αναμονής ως ένα σημείο κάτω δεξιά του εικονιδίου του κόκκινου σηματοδότη. Μετακινήστε την επιλεγμένη περιοχή προς τα δεξιά ώστε να δημιουργηθεί ο απαραίτητος χώρος για να τοποθετήσετε άλλο ένα εικονίδιο.

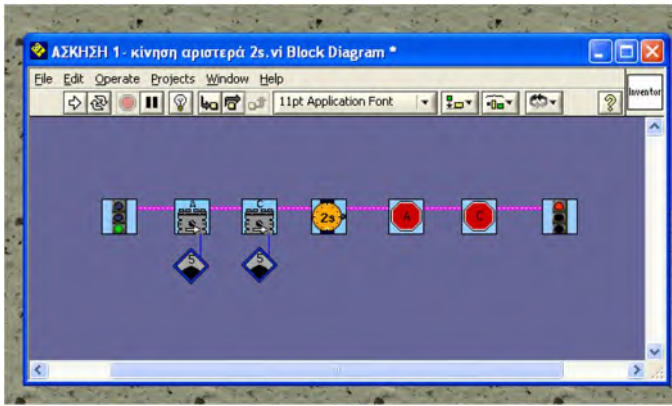


- Επιλέξτε από την παλέτα εντολών το εικονίδιο του κινητήρα C και τοποθετήστε το μέσα στο παράθυρο σχεδίασης, στο κενό χώρο που δημιουργήσατε. Στη συνέχεια τοποθετήστε κάτω από το εικονίδιο του κινητήρα C, το εικονίδιο του επιπέδου ισχύος 5.



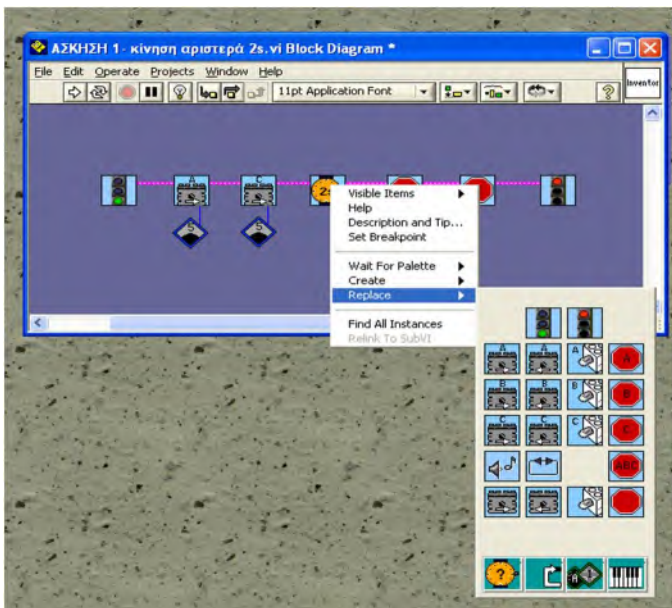
- Ενώστε με καλωδίωση τα εικονίδια που τοποθετήσατε.
- Κάντε ακριβώς τα ίδια βήματα για να τοποθετήσετε πριν από το εικονίδιο του κόκκινου σηματοδότη, το εικονίδιο που σταματάει τον κινητήρα C.

Σενάρια Ρομποτικής με το Mindstorms NXT



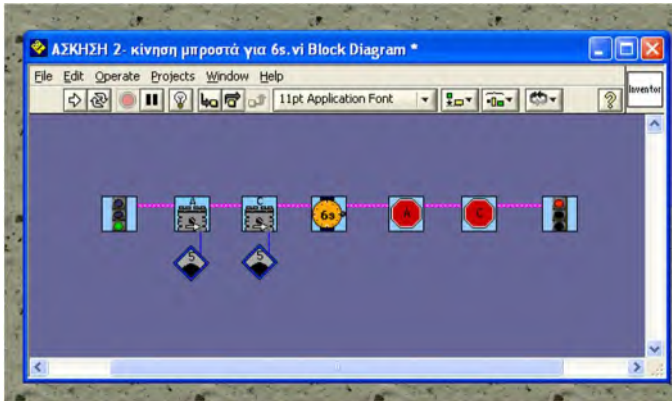
Υπάρχει και μια τελευταία αλλαγή που πρέπει να κάνουμε στο πρόγραμμα μας. Η αναμονή μετά τους κινητήρες θέλουμε να αλλάξει από 2 σε 6 δευτερόλεπτα. Για να το κατορθώσουμε αυτό, δε χρειάζεται να διαγράψουμε το εικονίδιο της αναμονής για 2 δευτερόλεπτα και στη θέση του να τοποθετήσουμε το αντίστοιχο εικονίδιο για αναμονή 6 δευτερολέπτων. Αρκεί να το αντικαταστήσουμε απλώς, με το εικονίδιο που θέλουμε:

- Με το εργαλείο επιλογής κάντε δεξί κλικ πάνω στο εικονίδιο της αναμονής για 2s . Στο μενού που εμφανίζεται, σύρετε το βελάκι του ποντικιού σας στη επιλογή *Replace*. Στο υπομενού που αναδύεται κάντε κλικ στο εικονίδιο της αναμονής για 6s.



Σενάρια Ρομποτικής με το Mindstorms NXT

- Αποθηκεύστε το πρόγραμμα σας ως **κίνηση μπροστά για 6s**.



- Μεταφορτώστε το πρόγραμμα που δημιουργήσατε στο NXT και τρέξτε το.

Συγχαρητήρια !!!

Ολοκληρώσατε την 2η δραστηριότητα

Δραστηριότητα 3η : κίνηση μπροστά για 6s και επιστροφή

(επέκταση προγράμματος)

Το πρόγραμμα στη δραστηριότητα αυτή αποτελείται από δύο μέρη. Το πρώτο μέρος (Κίνηση του οχήματος - ρομπότ προς τα εμπρός για 6 δευτερόλεπτα) το έχουμε αναπτύξει ήδη στην προηγούμενη εργασία. Αρκεί να ανοίξουμε το πρόγραμμα που έχουμε ήδη δημιουργήσει και να το συνεχίσουμε προσθέτοντας το δεύτερο μέρος : Να κινήσουμε το όχημα - ρομπότ στην αντίθετη κατεύθυνση για 6 δευτερόλεπτα.

Το πρώτο μέρος του προγράμματος που ήδη υλοποιήσαμε, μπορούμε να το διατυπώσουμε με τη φράση ' 'Κίνηση του οχήματος - ρομπότ προς τα εμπρός για 6 δευτερόλεπτα' '.

Αυτή η φράση μεταφράζεται στον παρακάτω ψευδοκώδικα :

Κίνηση κινητήρα A μπροστά.

Κίνηση κινητήρα C μπροστά.

Αναμονή 6s.

Σταμάτημα κινητήρα A.

Σταμάτημα κινητήρα C.

Ο ψευδοκώδικας για το δεύτερο μέρος του προγράμματος μας ‘ ‘και επιστροφή’ ’ είναι ο παρακάτω :

Κίνηση κινητήρα Α πίσω.

Κίνηση κινητήρα C πίσω.

Αναμονή 6s.

Σταμάτημα κινητήρα Α.

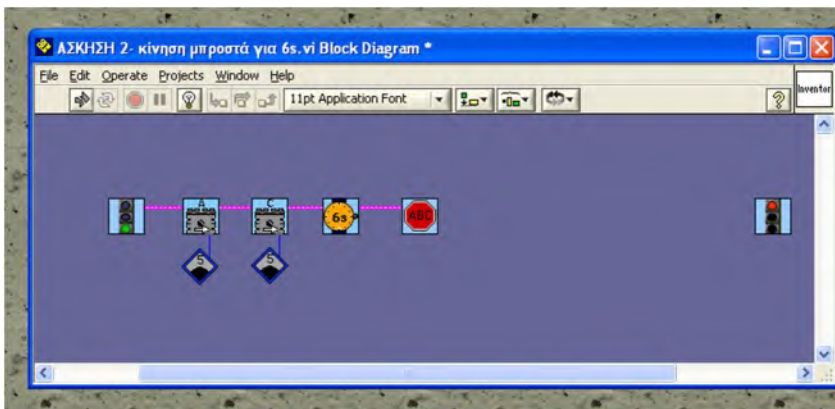
Σταμάτημα κινητήρα C.

Άνοιγμα του προγράμματος

- Ανοίξτε το πρόγραμμα **κίνηση μπροστά για 6s** που δημιουργήσατε προηγουμένως.

Τροποποίηση του προγράμματος

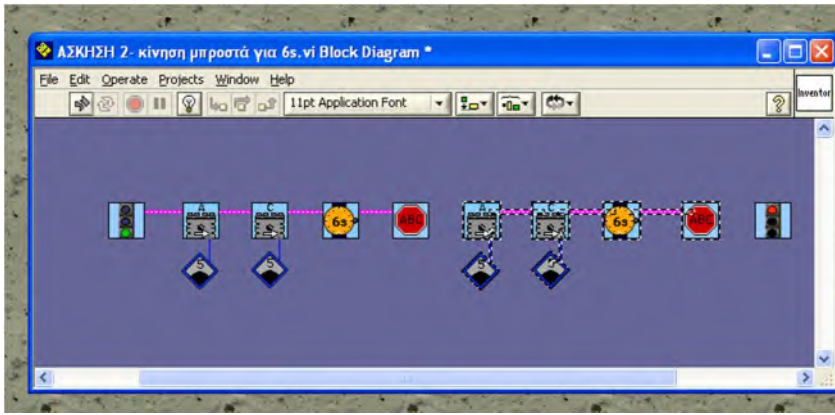
- Διαγράψτε την καλωδίωση πριν από το εικονίδιο του κόκκινου σηματοδότη για να μπορέσετε να τοποθετήσετε τα νέα εικονίδια.
- Μεταφέρετε προς τα δεξιά το εικονίδιο του κόκκινου σηματοδότη.



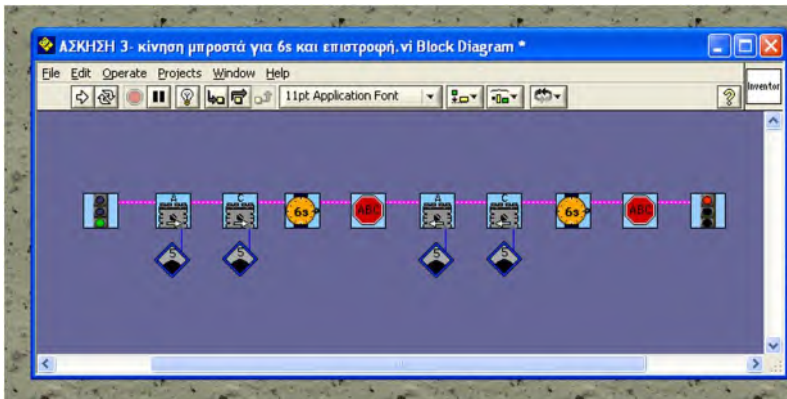
Για λόγους οικονομίας αντικαταστήστε τα δύο εικονίδια για το σταμάτημα των κινητήρων Α και C, με το γενικό εικονίδιο για το σταμάτημα όλων των κινητήρων.

Σε αυτό το σημείο, από τη στιγμή που τα βήματα στο δεύτερο μέρος του προγράμματος μας είναι παρόμοια με το πρώτο, αντί να τοποθετήσουμε από την αρχή τα νέα μας εικονίδια, θα χρησιμοποιήσουμε την ενέργεια της αντιγραφής και επικόλλησης :

- Επιλέξτε όλα τα εικονίδια ανάμεσα στους δύο σηματοδότες σέρνοντας το δείκτη του ποντικιού, από την πάνω αριστερή γωνία του εικονιδίου του κινητήρα Α μέχρι την κάτω δεξιά γωνία του εικονιδίου για το σταμάτημα όλων των κινητήρων. Από το μενού *edit* επιλέξτε διαδοχικά την επιλογή *copy* και στη συνέχεια την *paste*. Θα εμφανιστεί μπροστά σας το αντίγραφο που έχει δημιουργηθεί, το οποίο και μεταφέρετε το, αμέσως μετά το εικονίδιο για το σταμάτημα όλων των κινητήρων.



- Αλλάξτε την κατεύθυνση κίνησης των δύο κινητήρων μας στο δεύτερο μέρος του προγράμματος μας, χρησιμοποιώντας την εντολή *replace*.
- Τοποθετήστε τις καλωδιώσεις που χρειάζεται.
- Αποθηκεύστε το πρόγραμμα σας ως *κίνηση μπροστά για 6s και επιστροφή*.

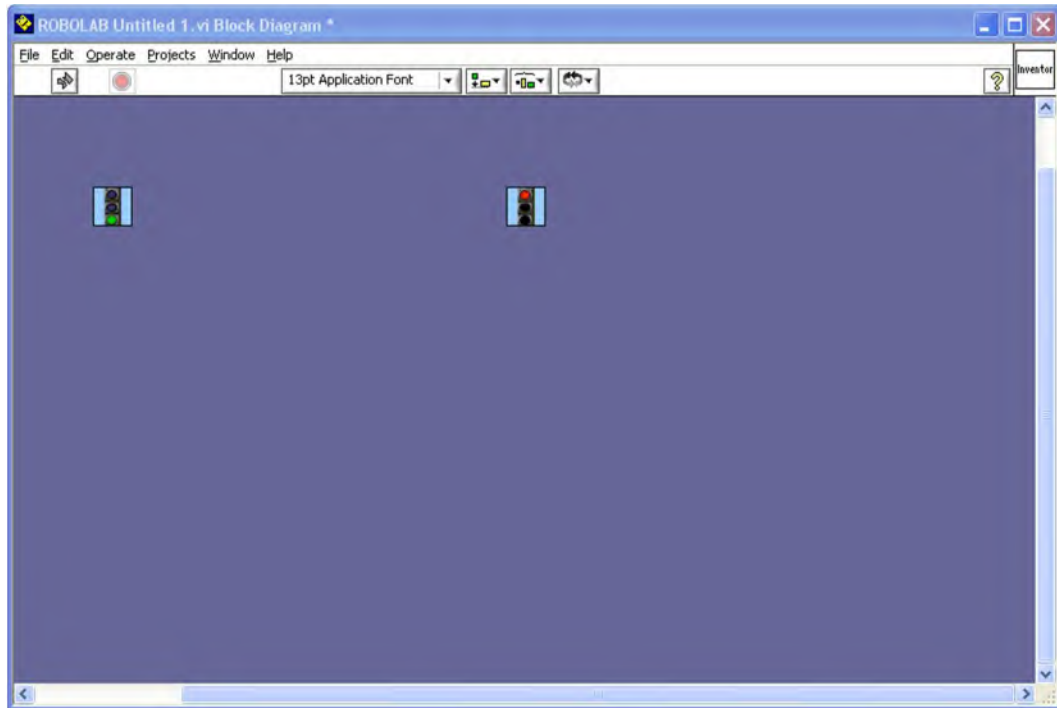


Συγχαρητήρια !!!
Ολοκληρώσατε την 3η δραστηριότητα

Τον κώδικα του RoboLab v2.9 των δραστηριοτήτων εξοικείωσης θα τον βρείτε στο σύνδεσμο :

[Εξοικείωση με το RoboLab](#)

Για περισσότερες πληροφορίες για τις εντολές του προγραμματιστικού περιβάλλοντος RoboLab ανατρέξτε στο ΠΑΡΑΡΤΗΜΑ II



Ενότητα 9η

Δραστηριότητες στο περιβάλλον του ROBOLAB

Οι παρακάτω δραστηριότητες έχουν σχεδιαστεί για το όχημα - ρομπότ NXTtank.

Οι Εικόνες και τα Logo είναι Copyright © της education.lego.com

Σενάρια Ρομποτικής με το Mindstorms NXT

Το όχημα αυτό διαθέτει δύο κινητήρες : Ο αριστερός κινητήρας κινεί την αριστερή ρόδα και είναι συνδεδεμένος στη θύρα εξόδου A του NXT, ενώ ο δεξιός κινητήρας κινεί τη δεξιά ρόδα και είναι συνδεδεμένος στη θύρα εξόδου C.

Λίγα λόγια για τους κινητήρες

Οι κινητήρες μπορούν να προγραμματιστούν να ξεκινούν και να σταματούν τη στιγμή που θέλουμε.



Αυτό το εικονίδιο ξεκινάει τον κινητήρα που είναι συνδεδεμένος στη θύρα A, προς τα μπροστά.

Για να σταματήσουμε έναν κινητήρα χρησιμοποιούμε το εικονίδιο με το σήμα στοπ.



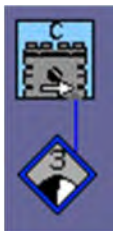
Αυτό το εικονίδιο σταματάει τον κινητήρα που είναι συνδεδεμένος στη θύρα A

Ένας άλλος τρόπος για να καθορίσουμε τη θύρα στην οποία έχουμε συνδέσει έναν κινητήρα είναι να κάνουμε χρήση του εικονιδίου του τροποποιητή θυρών (A,B,C). Ο εναλλακτικός αυτός τρόπος είναι βολικός στην περίπτωση που θέλουμε να κάνουμε αλλαγές σε ένα πρόγραμμα αρκετά συχνά.



Το πρώτο ζευγάρι εικονιδίων ξεκινάει τον κινητήρα που είναι συνδεδεμένος στη θύρα A προς τα μπροστά και αντίστοιχα, το δεύτερο ζευγάρι εικονιδίων σταματάει τον κινητήρα που είναι συνδεδεμένος στη θύρα A

Οι κινητήρες μπορούν να προγραμματιστούν να περιστρέφονται με διαφορετικές ταχύτητες περιστροφής. Για να το πετύχουμε αυτό, τοποθετούμε το εικονίδιο του τροποποιητή ισχύος. Με τον τροποποιητή αυτόν μπορούμε να καθορίσουμε διαφορετικά επίπεδα ισχύος, από το 1 ως το 5.

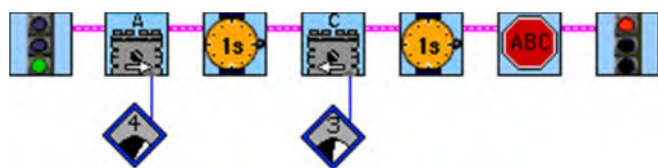


Αυτό το ζευγάρι εικονιδίων ξεκινάει τον κινητήρα C με επίπεδο ισχύος 5.

- Οι κινητήρες μπορούν να προγραμματιστούν να λειτουργούν για ένα συγκεκριμένο χρονικό διάστημα. Για να καθορίσουμε το χρονικό αυτό διάστημα κάνουμε χρήση του εικονιδίου της αναμονής για χρόνο.
- Τους κινητήρες μπορούμε να τους ελέγξουμε και με βάση την τιμή εισόδου από κάποιον αισθητήρα, όπως θα δούμε παρακάτω.

Παράδειγμα :

Το παρακάτω πρόγραμμα



1. ξεκινάει τον κινητήρα A με ισχύ 4 προς τα μπροστά
2. περιμένει να περάσει 1 δευτερόλεπτο
3. στη συνέχεια ξεκινάει τον κινητήρα C με ισχύ 3 προς τα πίσω
4. περιμένει να περάσει 1 δευτερόλεπτο (ο κινητήρας A εξακολουθεί να γυρίζει)
5. και τέλος σταματάει και τους δύο κινητήρες.

(ο κινητήρας A έχει λειτουργήσει συνολικά για 2 δευτερόλεπτα προτού σταματήσει.)

Δραστηριότητα 3η : κίνηση μπροστά για 4s στάση και επιστροφή (β' έκδοση)

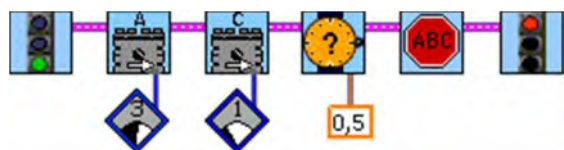
Το όχημα-ρομπότ κινείται προς τα μπροστά με επίπεδο ισχύος 4 για 4 δευτερόλεπτα, σταματάει για 2 δευτερόλεπτα και μετά κινείται προς τα πίσω με επίπεδο ισχύος 2 για 4 δευτερόλεπτα και σταματάει.



Δραστηριότητα 4η : κίνηση σε καμπύλη δεξιά

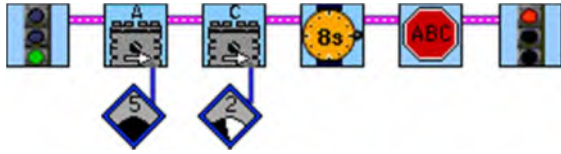
Το όχημα-ρομπότ κινείται διαγράφοντας μία καμπύλη δεξιά για 0.5 δευτερόλεπτα και μετά σταματάει.

Για να στρίψει το ρομπότ διαγράφοντας μία καμπύλη προς τα δεξιά πρέπει να κινηθεί ο κινητήρας A προς τα μπροστά με επίπεδο ισχύος πχ 3 και ο κινητήρας C πάλι προς τα μπροστά με μικρότερο επίπεδο ισχύος σε σχέση με τον A πχ 1.



Δραστηριότητα 4η : κίνηση σε καμπύλη (β' έκδοση)

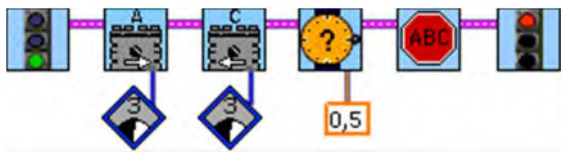
Το όχημα-ρομπότ κινείται με τον αριστερό κινητήρα προς τα μπροστά με επίπεδο ισχύος 5 και με το δεξιό κινητήρα προς τα μπροστά με επίπεδο ισχύος 2, για 8 δευτερόλεπτα και μετά σταματάει.



Δραστηριότητα 5η : στροφή επιτόπου δεξιά

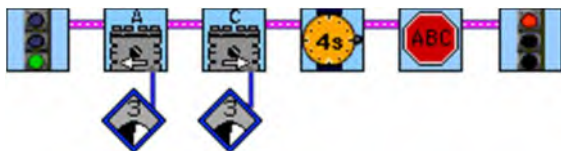
Το όχημα-ρομπότ στρίβει επιτόπου προς τα δεξιά για 0.5 δευτερόλεπτα και μετά σταματάει.

Για να στρίψει το ρομπότ επιτόπου προς τα δεξιά πρέπει να κινηθεί ο κινητήρας A προς τα μπροστά και ο κινητήρας C προς τα πίσω.



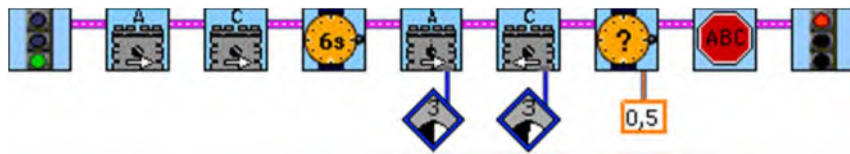
Δραστηριότητα 5η : στροφή επιτόπου αριστερά (β' έκδοση)

Το όχημα-ρομπότ στρίβει επιτόπου προς τα αριστερά για 4 δευτερόλεπτα και μετά σταματάει.



Δραστηριότητα 6η : κίνηση μπροστά για 6s και στροφή επιτόπου

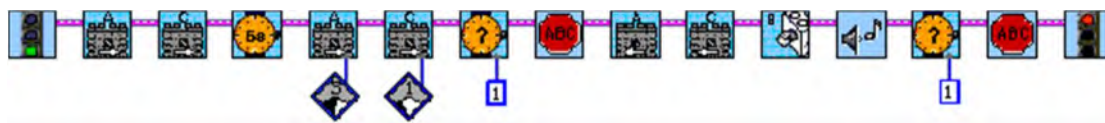
Το όχημα-ρομπότ κινείται προς τα μπροστά για 6 δευτερόλεπτα και μετά στρίβει επιτόπου προς τα δεξιά για 0.5 δευτερόλεπτα και σταματάει.



Δραστηριότητα 7η : παρκάρισμα

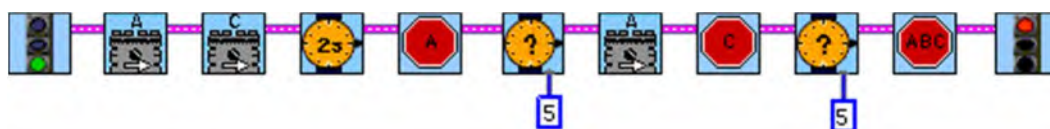
Το όχημα-ρομπότ κινείται προς τα μπροστά για 6 δευτερόλεπτα, στρίβει προς τα δεξιά για 1 δευτερόλεπτο, σταματάει και στη συνέχεια κινείται προς τα πίσω για 1 δευτερόλεπτο και σταματάει.

Καθώς πηγαίνει με την όπισθεν ανάβει μία λάμπα και ακούγεται ήχος



Δραστηριότητα 7η : κίνηση και ελιγμοί (β' έκδοση)

Το όχημα-ρομπότ κινείται προς τα μπροστά για 2 δευτερόλεπτα, εκτελεί αριστερή στροφή για 5 δευτερόλεπτα, κατόπιν δεξιά στροφή για άλλα 5 δευτερόλεπτα, και σταματάει.



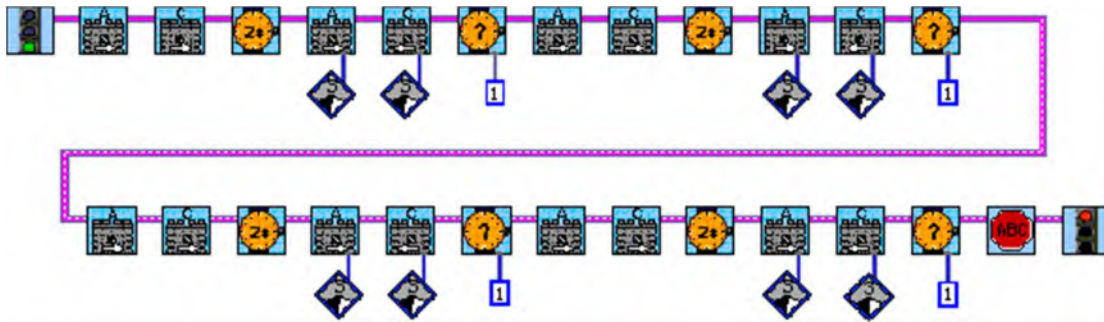
Δραστηριότητα 8η : κίνηση σε τετράγωνο

Το όχημα-ρομπότ κινείται έτσι ώστε να ακολουθήσει μία διαδρομή ή οποία να έχει σχήμα τετραγώνου.

Για να κινηθεί το ρομπότ μας σε τετράγωνο πρέπει :

1. να κινηθεί μπροστά για κάποια απόσταση.
2. να στρίψει κατά 90° προς μία κατεύθυνση.
3. να κινηθεί μπροστά ίδια απόσταση όπως πριν.
4. να στρίψει κατά 90° προς την ίδια κατεύθυνση όπως πριν.
5. να κινηθεί μπροστά ίδια απόσταση όπως πριν.
6. να στρίψει κατά 90° προς την ίδια κατεύθυνση όπως πριν.
7. να κινηθεί μπροστά ίδια απόσταση όπως πριν.
8. να στρίψει κατά 90° προς την ίδια κατεύθυνση όπως πριν.

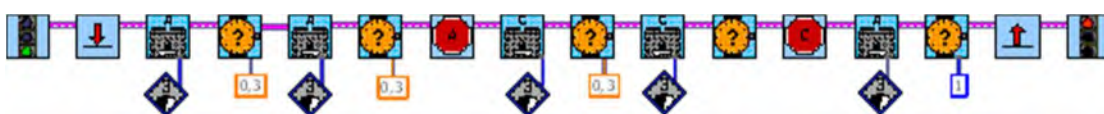
Παρατηρούμε ότι επαναλαμβάνουμε τα ίδια δύο πρώτα βήματα 4 φορές.



Δραστηριότητα 9η : χορός Salsa

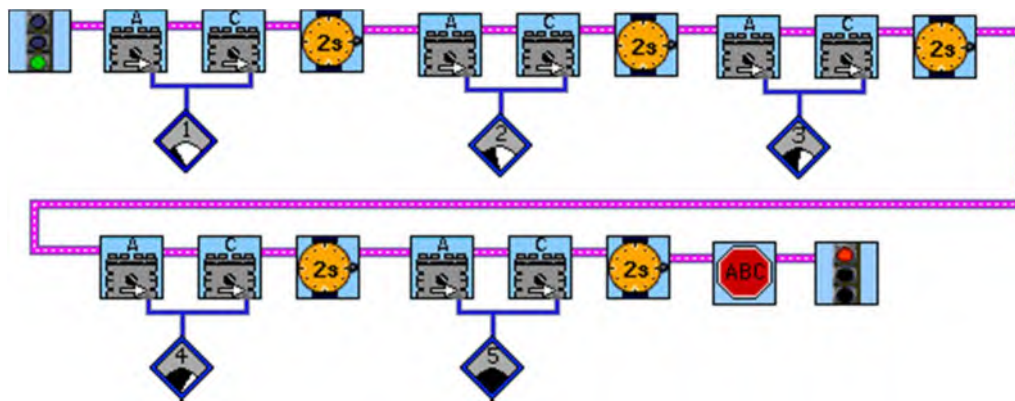
Το όχημα-ρομπότ κινείται ακολουθώντας τα βήματα του παρακάτω αλγόριθμου :

1. Κίνηση κινητήρα A πίσω για 1s
2. Κίνηση κινητήρα A μπροστά για 1s και σταμάτημα
3. Κίνηση κινητήρα C μπροστά για 1s
4. Κίνηση κινητήρα C πίσω για 1s και σταμάτημα
5. Κίνηση κινητήρα A μπροστά για 4s
6. και πάλι από την αρχή



Δραστηριότητα 10η : ξεκίνημα με επιτάχυνση

Το όχημα-ρομπότ κινείται προς τα μπροστά με επίπεδο ισχύος 1 για 2 δευτερόλεπτα, και μετά επιταχύνει σταδιακά κάθε 2 δευτερόλεπτα - αυξάνει δηλαδή το επίπεδο ισχύος στους κινητήρες μέχρι αυτό να φτάσει το επίπεδο ισχύος 5 (δηλαδή κινείται στη συνέχεια με επίπεδο ισχύος 2 για άλλα 2 δευτερόλεπτα, μετά με επίπεδο ισχύος 3 για άλλα 2 δευτερόλεπτα, μετά με επίπεδο ισχύος 4 για άλλα 2 δευτερόλεπτα και μετά με επίπεδο ισχύος 5 για άλλα 2 δευτερόλεπτα και σταματάει).

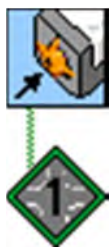


Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τους κινητήρες θα τον βρείτε στο σύνδεσμο :

[Κινητήρες](#)

Λίγα λόγια για τον αισθητήρα αφής

Οι αισθητήρες αφής μπορούν να προγραμματιστούν ώστε να δίνουν την πληροφορία, αν ο διακόπτης τους είναι πατημένος ή όχι σε μια συγκεκριμένη χρονική στιγμή. Πρέπει να καθορίσουμε σε ποια θύρα είναι συνδεδεμένος ένας αισθητήρας αφής.



Αυτό το ζευγάρι εικονιδίων έχει ως αποτέλεσμα, το ρομπότ μας να περιμένει συνέχεια, μέχρι να πατηθεί ο διακόπτης στον αισθητήρα αφής που είναι συνδεδεμένος στη θύρα εισόδου 1.

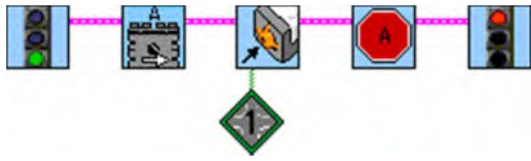


Αυτό το εικονίδιο βάζει το πρόγραμμα μας σε αναμονή, μέχρι να απελευθερωθεί ο διακόπτης στον αισθητήρα αφής που είναι συνδεδεμένος στη θύρα εισόδου 1.

- Οι αισθητήρες αφής μπορούν να χρησιμοποιηθούν για να ελέγξουμε τους κινητήρες.

Παραδείγματα :

α) Το παρακάτω πρόγραμμα



1. ξεκινάει τον κινητήρα A με ισχύ 5 προς τα μπροστά
2. και μετά περιμένει μέχρι να πατηθεί ο διακόπτης του αισθητήρα αφής που είναι συνδεδεμένος στη θύρα 1.
3. Όταν αυτό συμβεί σταματάει τον κινητήρα A.

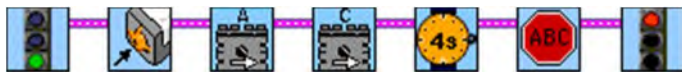
β) Το παρακάτω πρόγραμμα



1. περιμένει μέχρι να πατηθεί ο διακόπτης του αισθητήρα αφής που είναι συνδεδεμένος στη θύρα
2. μόλις αυτός πατηθεί ξεκινάει τον κινητήρα A με ισχύ 5 προς τα μπροστά
3. και μετά περιμένει μέχρι να απελευθερωθεί ο διακόπτης του αισθητήρα αφής που είναι συνδεδεμένος στη θύρα 1.
4. Όταν αυτό συμβεί σταματάει τον κινητήρα A.

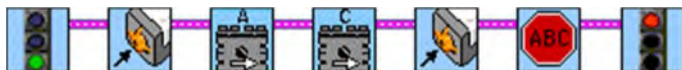
Δραστηριότητα 11η : ξεκίνημα με αισθητήρα αφής

Το όχημα-ρομπότ ξεκινάει να κινείται προς τα μπροστά όταν πατηθεί ο διακόπτης του αισθητήρα αφής και σταματάει μετά από 4 δευτερόλεπτα.



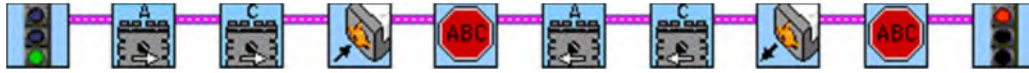
Δραστηριότητα 12η : ξεκίνημα και σταμάτημα με αισθητήρα αφής

Το όχημα-ρομπότ ξεκινάει να κινείται προς τα μπροστά όταν πατηθεί ο διακόπτης του αισθητήρα αφής και σταματάει όταν ο διακόπτης του αισθητήρα αφής πατηθεί ξανά.



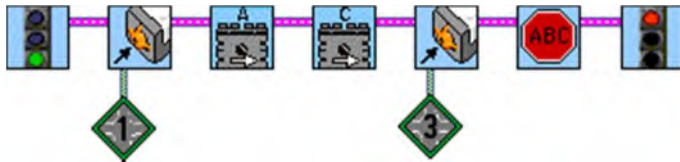
Δραστηριότητα 13η : ξεκίνημα, αλλαγή κατεύθυνσης και σταμάτημα με αισθητήρα αφής

Το όχημα-ρομπότ αρχικά κινείται προς τα μπροστά. Όταν πατηθεί ο διακόπτης του αισθητήρα αφής και κρατηθεί αλλάζει τη φορά της κίνησης. Όταν ο διακόπτης του αισθητήρα αφής απελευθερωθεί σταματάει.



Δραστηριότητα 14η : ξεκίνημα και σταμάτημα με δύο αισθητήρες αφής

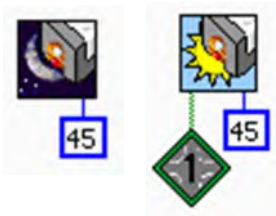
Το όχημα-ρομπότ ξεκινάει να κινείται προς τα μπροστά όταν πατηθεί ο διακόπτης του αισθητήρα αφής 1 και σταματάει μόλις πατηθεί ο διακόπτης του αισθητήρα αφής 3.



Λίγα λόγια για τον αισθητήρα φωτός

Οι αισθητήρες φωτός μπορούν να προγραμματιστούν να ανιχνεύουν την ποσότητα του φωτός η οποία ανακλάται από μία επιφάνεια που υπάρχει μπροστά τους, σε μία συγκεκριμένη χρονική στιγμή. Το επίπεδο του φωτός που ανιχνεύουν το μετατρέπουν σε μία αριθμητική τιμή από το 0 = απόλυτο σκοτάδι ως το 100 = έντονο φως.

Πρέπει να καθορίσουμε σε ποια θύρα είναι συνδεδεμένος ένας αισθητήρας φωτός.



Αυτά τα εικονίδια έχουν ως αποτέλεσμα το ρομπότ μας να περιμένει συνέχεια, μέχρι η τιμή του φωτός που θα μετρηθεί από τον αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα 1, να πέσει κάτω από την τιμή 45 ή να ξεπεράσει την τιμή 45 αντίστοιχα

Αυτό τα εικονίδια έχουν ως αποτέλεσμα το ρομπότ μας να περιμένει συνέχεια, μέχρι η τιμή του φωτός που θα μετρηθεί από τον αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα 1, να αυξηθεί ή να μειωθεί αντίστοιχα κατά μία συγκεκριμένη τιμή από ότι προηγουμένως.

Στα εικονίδια αυτά τοποθετούμε έναν τροποποιητή αριθμητικής τιμής.



- Οι αισθητήρες φωτός μπορούν να χρησιμοποιηθούν για να ελέγξουμε τους κινητήρες.

Παράδειγμα :

Το παρακάτω πρόγραμμα

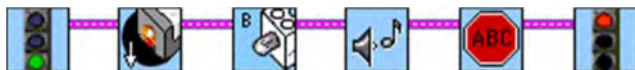


1. ξεκινάει τους κινητήρες A και C με ισχύ 3 προς τα μπροστά
2. και στη συνέχεια περιμένει μέχρι η ποσότητα φωτός στον αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα 1 να πέσει κάτω από τη αριθμητική τιμή 55
3. Όταν συμβεί αυτό σταματάει τους δύο κινητήρες.

Προσοχή : Για να γίνει σωστή μέτρηση από τον αισθητήρα φωτός πρέπει το ρομπότ μας να κινείται με αργή ταχύτητα. Χρησιμοποιούμε λοιπόν τροποποιητή με χαμηλό επίπεδο ισχύος.

Δραστηριότητα 15η : υλοποίηση συναγερμού

Όταν περάσει ένα αντικείμενο μπροστά από τον αισθητήρα φωτός ή όταν σκοτεινιάσει, το NXT ανάβει ένα λαμπάκι και παίζει έναν ήχο.



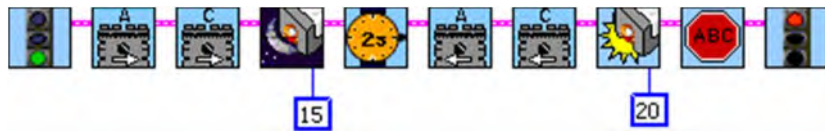
Δραστηριότητα 16η : άναμμα φωτός σε τούνελ

Το όχημα - ρομπότ ανάβει το φως όταν περάσει μέσα από ένα σκοτεινό τούνελ.



Δραστηριότητα 16η : κίνηση μπροστά μέχρι να σκοτεινιάσει (β' έκδοση)

Το όχημα - ρομπότ κινείται προς τα μπροστά μέχρι να σκοτεινιάσει, οπότε και περιμένει για 2 δευτερόλεπτα. Στη συνέχεια κινείται προς τα πίσω μέχρι να βγει από το σκοτάδι οπότε και σταματάει.



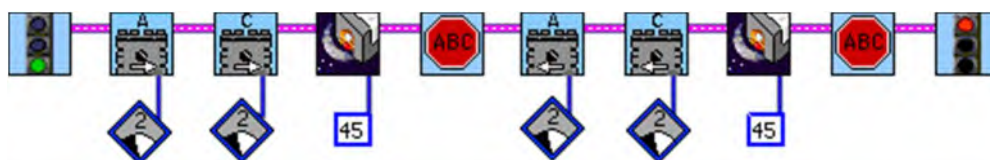
Δραστηριότητα 17η : κίνηση μπροστά μέχρι τη μαύρη γραμμή

Το όχημα - ρομπότ κινείται προς τα μπροστά μέχρι να περάσει πάνω από μία μαύρη γραμμή οπότε και σταματάει.



Δραστηριότητα 18η : κίνηση μπροστά μέχρι τη μαύρη γραμμή και πίσω

Το όχημα-ρομπότ κινείται προς τα μπροστά μέχρι να περάσει πάνω από μία μαύρη γραμμή. Στη συνέχεια κινείται προς τα πίσω μέχρι να συναντήσει μια άλλη μαύρη γραμμή οπότε και σταματάει.

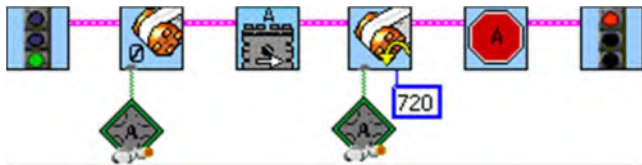


Λίγα λόγια για τον αισθητήρα περιστροφής

Οι αισθητήρες περιστροφής ενός άξονα μπορούν να προγραμματιστούν να μετράνε τον αριθμό των περιστροφών που έχουν πραγματοποιηθεί σε έναν άξονα μέχρι μία συγκεκριμένη χρονική στιγμή. Μια πλήρης περιστροφή του άξονα αντιστοιχεί στην αριθμητική τιμή 360.

Παράδειγμα :

Αυτό το πρόγραμμα



1. μηδενίζει την αριθμητική τιμή που είναι αποθηκευμένη εκείνη τη στιγμή για τον αισθητήρα περιστροφής του κινητήρα που είναι συνδεδεμένος στη θύρα εξόδου A
2. ξεκινάει τον κινητήρα A με ισχύ 5 προς τα μπροστά
3. και στη συνέχεια περιμένει μέχρι ο αισθητήρα περιστροφής να εκτελέσει $720/360 = 2$ περιστροφές.
4. οπότε και σταματάει τον κινητήρα A.

Προσοχή : Δεν πρέπει να ξεχνάμε να μηδενίζουμε την προϋπάρχουσα τιμή του αισθητήρα περιστροφής πριν να τον χρησιμοποιήσουμε.

Δραστηριότητα 19η : κίνηση μπροστά για 1 περιστροφή

Το όχημα - ρομπότ κινείται προς τα μπροστά μέχρι οι ρόδες του να συμπληρώσουν 1 πλήρη περιστροφή



Τον κώδικα του RoboLab v2.9 των δραστηριοτήτων για τους αισθητήρες θα τον βρείτε στο σύνδεσμο : [Αισθητήρες](#)

Λίγα λόγια για τη δομή επιλογής

Συχνά μέσα στο πρόγραμμα χρειάζεται να πούμε στο ρομπότ μας τι να κάνει όταν συναντήσει μία μελλοντική κατάσταση, την οποία δεν μπορούμε να ξέρουμε από πριν αν θα τη συναντήσει και πότε. Πχ να πέσει πάνω σε ένα εμπόδιο. Σε αυτήν την περίπτωση θα πρέπει να έχουμε προγραμματίσει το ρομπότ μας κατάλληλα, ώστε να είναι σε θέση τη στιγμή εκείνη, να πάρει μια απόφαση για τι θα κάνει, στηριζόμενο στην τιμή κάποιας παραμέτρου, η οποία μπορεί να είναι η τιμή ενός αισθητήρα ή ένα χρονικό διάστημα.

Για να το πετύχουμε αυτό χρησιμοποιούμε μία δομή επιλογής. Στο Robolab η δομή επιλογής υλοποιείται με τη χρήση των εικονιδίων διακλάδωσης.

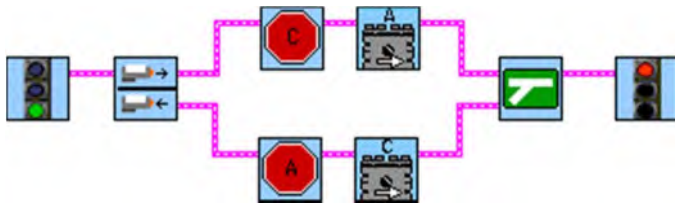


Για παράδειγμα αυτό το εικονίδιο αναπαριστά μία διακλάδωση με χρήση του αισθητήρα αφής.

- Με το εικονίδιο διακλάδωσης αισθητήρα αφής μπορούμε να προγραμματίσουμε το ρομπότ μας να κάνει διαφορετικά πράγματα, ανάλογα με το αν ο διακόπτης του αισθητήρα αφής, τη στιγμή που γίνει ο έλεγχος, θα βρίσκεται πατημένος ή απελευθερωμένος.

Παράδειγμα :

Στο παρακάτω πρόγραμμα τη στιγμή που θα έρθει η σειρά να εκτελεστεί η εντολή διακλάδωσης αισθητήρα αφής



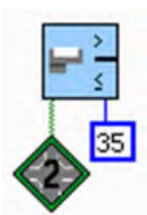
αν ο διακόπτης του αισθητήρα αφής στη θύρα 1 βρεθεί πατημένος **τότε** ο κινητήρας C θα σταματήσει και ο κινητήρας A θα ξεκινήσει.

αλλιώς (αν δηλαδή ο διακόπτης του αισθητήρα αφής στη θύρα 1 βρεθεί απελευθερωμένος **τότε)** ο κινητήρας A θα σταματήσει και ο κινητήρας C θα ξεκινήσει.



Προσοχή: μετά από μια διακλάδωση είναι απαραίτητο να τοποθετούμε το εικονίδιο της συγχώνευσης της διακλάδωσης.

Διακλαδώσεις μπορούμε να υλοποιήσουμε και με τη χρήση των άλλων αισθητήρων.



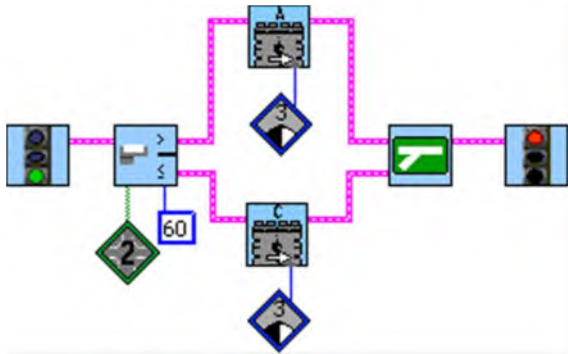
Στην εικόνα αυτή έχουμε μία διακλάδωση με χρήση του αισθητήρα φωτός.

Σενάρια Ρομποτικής με το Mindstorms NXT

- Με το εικονίδιο διακλάδωσης αισθητήρα φωτός μπορούμε να προγραμματίσουμε το ρομπότ μας να κάνει διαφορετικά πράγματα, ανάλογα με το αν η τιμή που θα διαβάσει ο αισθητήρας τη στιγμή που γίνει ο έλεγχος, θα είναι μεγαλύτερη ή όχι από μία συγκεκριμένη τιμή, εδώ 55. Η απόφαση που θα λάβει το ρομπότ μας δηλαδή, εξαρτάται από την ένταση του φωτός που θα μετρηθεί από τον αισθητήρα φωτός, τη στιγμή του ελέγχου.

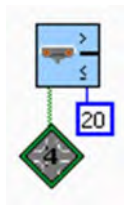
Παράδειγμα :

Στο παρακάτω πρόγραμμα τη στιγμή που θα έρθει η σειρά να εκτελεστεί η εντολή διακλάδωσης αισθητήρα φωτός



αν ο αισθητήρα φωτός στη θύρα 2 ανιχνεύσει μία ποσότητα φωτός πάνω από την αριθμητική τιμή, εδώ 60 **τότε** ο κινητήρας A θα ξεκινήσει.

αλλιώς (αν δηλαδή ο αισθητήρας φωτός στη θύρα 2 ανιχνεύσει μία ποσότητα φωτός ίση ή κάτω από την αριθμητική τιμή 60 **τότε)** ο κινητήρα C θα ξεκινήσει.



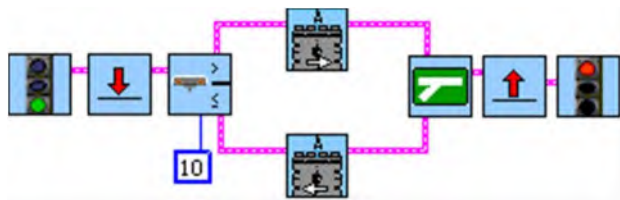
Στην εικόνα αυτή έχουμε μία διακλάδωση με χρήση του αισθητήρα υπερήχων.

- Με το εικονίδιο διακλάδωσης αισθητήρα υπερήχων μπορούμε να προγραμματίσουμε το ρομπότ μας να κάνει διαφορετικά πράγματα, ανάλογα με το αν η απόσταση που θα ανιχνεύσει ο αισθητήρας τη στιγμή που γίνει ο έλεγχος, θα είναι μεγαλύτερη ή όχι από μία συγκεκριμένη τιμή, εδώ 20. Η απόφαση που θα λάβει το ρομπότ μας δηλαδή, εξαρτάται από την απόσταση που θα μετρηθεί από τον αισθητήρα υπερήχων, τη στιγμή του ελέγχου.

Παράδειγμα :

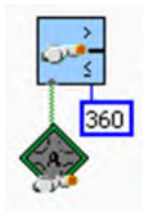
Στο παρακάτω πρόγραμμα τη στιγμή που θα έρθει η σειρά να εκτελεστεί η εντολή διακλάδωσης αισθητήρα υπερήχων.

Σενάρια Ρομποτικής με το Mindstorms NXT



αν ο αισθητήρα υπερήχων στη θύρα 1 ανιχνεύσει μία απόσταση πάνω από την αριθμητική τιμή 10 **τότε** ο κινητήρας A θα γυρίσει προς τα μπροστά.

αλλιώς (αν δηλαδή ο αισθητήρα υπερήχων στη θύρα 1 ανιχνεύσει μία απόσταση ίση ή κάτω από την αριθμητική τιμή 10 **τότε)** ο κινητήρας A θα γυρίσει προς τα πίσω.

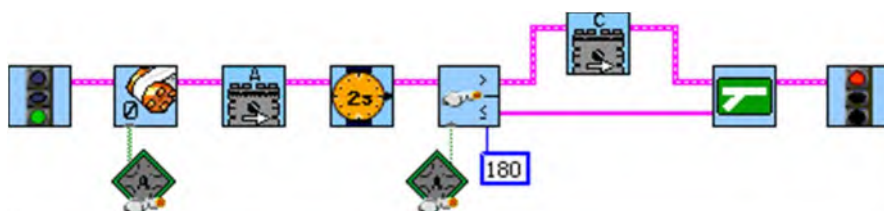


Στην εικόνα αυτή έχουμε μία διακλάδωση με χρήση του αισθητήρα περιστροφής.

- Με το εικονίδιο διακλάδωσης αισθητήρα περιστροφής μπορούμε να προγραμματίσουμε το ρομπότ μας να κάνει διαφορετικά πράγματα, ανάλογα με το αν η τιμή που θα διαβάσει ο αισθητήρας, τη στιγμή που θα γίνει ο έλεγχος θα είναι μεγαλύτερη ή όχι από μία συγκεκριμένη τιμή, εδώ 360 μοίρες. Η απόφαση που θα λάβει το ρομπότ μας δηλαδή εξαρτάται από τον αριθμό των περιστροφών που θα έχουν πραγματοποιηθεί μέχρι τη στιγμή του ελέγχου.

Παράδειγμα :

Στο παρακάτω πρόγραμμα τη στιγμή που θα έρθει η σειρά να εκτελεστεί η εντολή διακλάδωσης αισθητήρα περιστροφής.

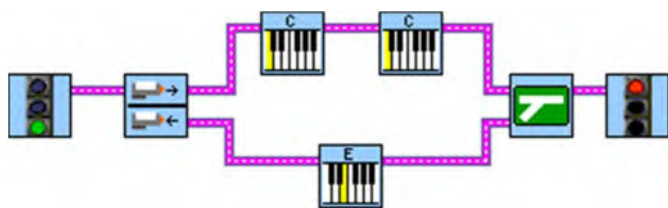


αν ο αισθητήρας περιστροφής του κινητήρα στη θύρα A έχει μετρήσει μία περιστροφή πάνω από την αριθμητική τιμή 180 **τότε** ο κινητήρας C θα ξεκινήσει.

αλλιώς (αν δηλαδή ο αισθητήρας περιστροφής στη θύρα A έχει μετρήσει μία περιστροφή ίση ή κάτω από την αριθμητική τιμή 180 **τότε)** δε θα συμβεί τίποτα.

Δραστηριότητα 20η : νότες με διακλάδωση αφής

Αν ο διακόπτης του αισθητήρα αφής βρεθεί πατημένος, το NXT παίζει μία νότα Ε, διαφορετικά, αν αυτός βρεθεί απελευθερωμένος δηλαδή, παίζει δύο νότες C.

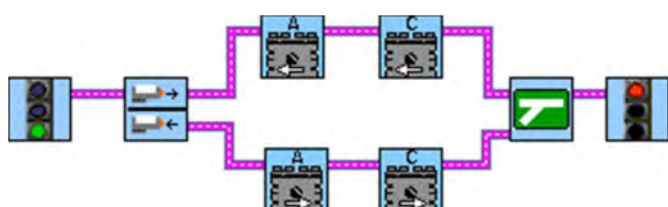


Δραστηριότητα 20η : κίνηση με διακλάδωση αφής (β' έκδοση)

Τοποθετείστε στο όχημα - ρομπότ tankbot στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα αφής.

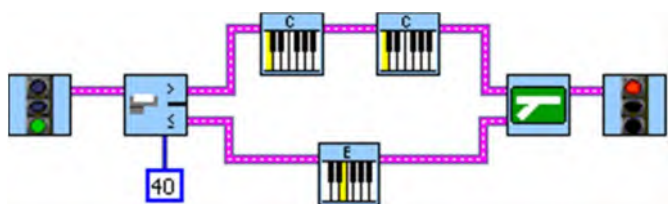
Συνδέστε τον αισθητήρα αφής στη θύρα εισόδου 1 του NXT.

Αν βρεθεί πατημένος ο διακόπτης του αισθητήρα αφής το όχημα-ρομπότ κινείται προς τα μπροστά, διαφορετικά, αν αυτός βρεθεί απελευθερωμένος δηλαδή, το όχημα-ρομπότ κινείται προς τα πίσω.



Δραστηριότητα 21η : νότες με διακλάδωση φωτός

Αν η ποσότητα φωτός που ανιχνεύσει ο αισθητήρας φωτός είναι πάνω από την τιμή 40, το NXT παίζει δύο νότες C, διαφορετικά, αν αυτός ανιχνεύσει ποσότητα φωτός κάτω ή ίση με 40 δηλαδή, παίζει μία νότα Ε.

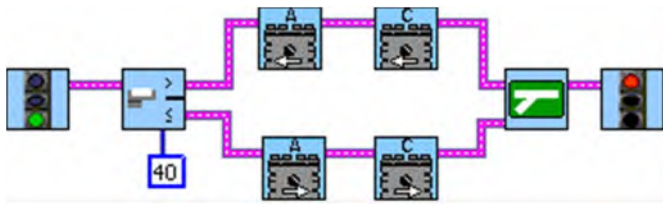


Δραστηριότητα 21η : κίνηση με διακλάδωση φωτός (β' έκδοση)

Τοποθετείστε στο όχημα - ρομπότ *tankbot* στο μπροστινό μέρος του, το βραχίονα με τον έναν αισθητήρα φωτός.

Συνδέστε τον αισθητήρα φωτός στη θύρα εισόδου 1 του NXT.

Αν η ποσότητα φωτός που ανιχνεύσει ο αισθητήρας φωτός είναι πάνω από την τιμή 40 το όχημα-ρομπότ κινείται προς τα πίσω, διαφορετικά, αν αυτός ανιχνεύσει ποσότητα φωτός κάτω ή ίση με 40 δηλαδή, το όχημα-ρομπότ κινείται προς τα μπροστά.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τις δομές επιλογής θα τον βρείτε στο σύνδεσμο :

[Δομές Επιλογής](#)

Λίγα λόγια για τη δομή επανάληψης (αέναο θρόχο)

Στις προηγούμενες δραστηριότητες μπορείτε να παρατηρήσετε ότι το ρομπότ μας περνάει από τη διακλάδωση μόνο μία φορά. Βάζοντας το ρομπότ μας να επαναλαμβάνει συνεχώς την εκτέλεση μίας δομής επιλογής, του προσδίδουμε μία μόνιμη συμπεριφορά, ώστε κάθε φορά που συναντάει μία συγκεκριμένη κατάσταση να είναι σε θέση να αντιδράσει με τον κατάλληλο τρόπο.

Άλματα

Τα άλματα είναι ένα ζευγάρι εικονιδίων τα οποία μας επιτρέπουν να ελέγξουμε τη ροή (την πορεία δηλαδή που θα ακολουθηθεί) του προγράμματος μας.



Όταν το πρόγραμμα συναντήσει αυτό το εικονίδιο άλματος, θα σταματήσει να εκτελείται σειριακά δηλ. να εκτελέσει την επόμενη στη σειρά εντολή, και θα μεταβεί στο σημείο που βρίσκεται στο αντίστοιχο εικονίδιο προσγείωσης.



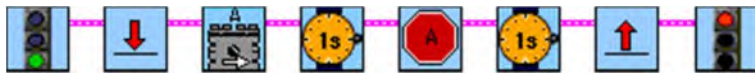
Το εικονίδιο προσγείωσης μπορεί να βρίσκεται μετά ή και πριν από το εικονίδιο άλματος.

Υπάρχουν 5 διαφορετικά ζευγάρια εικονιδίων άλματος με διαφορετικό χρώμα το καθένα.

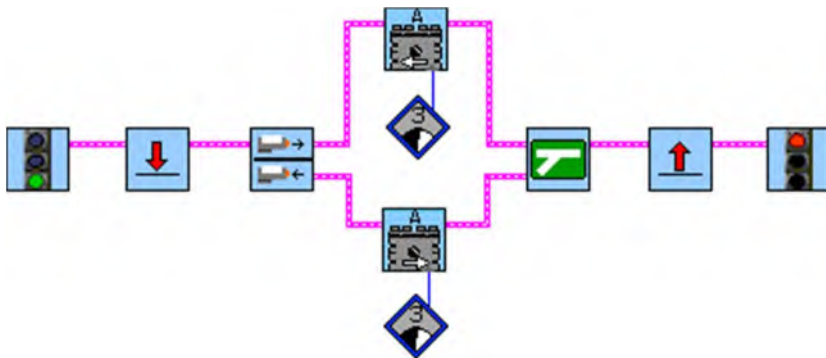
- Ένα ζευγάρι εικονιδίων άλματος μπορεί να χρησιμοποιηθεί για να επαναλάβουμε ένα μέρος του προγράμματος μας, συνέχεια.

Παραδείγματα :

α) Στο παρακάτω πρόγραμμα χρησιμοποιείται το κόκκινο ζευγάρι άλματος για το μέρος του προγράμματος που ξεκινάει τον κινητήρα A για χρόνο 1 δευτερόλεπτο και τον σταματάει για χρόνο 1 δευτερόλεπτο. Αυτή η διαδικασία - σε αντίθεση με την περίπτωση που θα χρησιμοποιούσαμε το ζευγάρι εικονιδίων επανάληψης - θα συμβαίνει συνέχεια, μέχρι να διακοπεί η εκτέλεση του προγράμματος από εμάς τους ίδιους.



β) Στο παρακάτω πρόγραμμα



αν ο διακόπτης του αισθητήρα αφής στη θύρα 1 είναι πατημένος **τότε** ο κινητήρας A θα κινείται προς τα μπροστά.

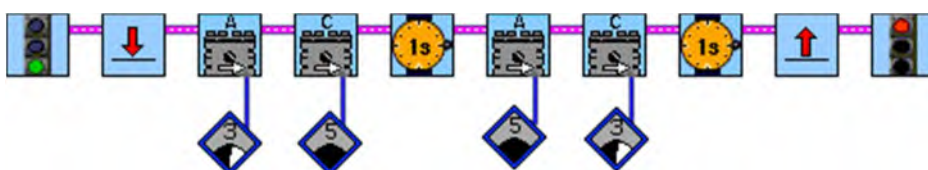
αλλιώς (αν δηλαδή ο διακόπτης του αισθητήρα αφής στη θύρα 1 είναι απελευθερωμένος **τότε)** ο κινητήρας A θα κινείται προς τα πίσω.

Τα παραπάνω θα επαναλαμβάνονται συνεχώς, μέχρι να διακόψουμε εμείς, την εκτέλεση του προγράμματος.

Δραστηριότητα 22η : κίνηση φιδωτή

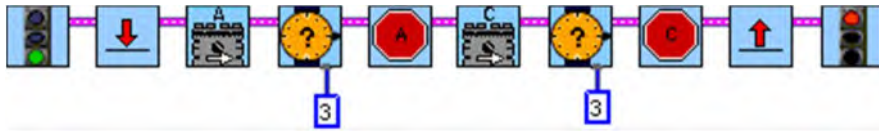
Το όχημα-ρομπότ κινείται φιδωτά, δηλαδή στρίβει μία αριστερά και μία δεξιά διαγράφοντας καμπύλες γραμμές.

Μπορούμε να αλλάξουμε το τόξο της στροφής τροποποιώντας την ισχύ των κινητήρων ή το χρονικό διάστημα της κίνησης.



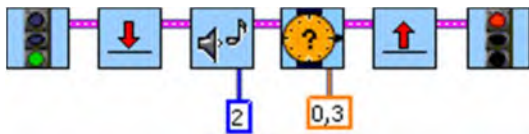
Δραστηριότητα 22η : κίνηση σύρσιμο (6' έκδοση)

Το όχημα-ρομπότ κινείται σαν να σέρνεται. Κινεί τον αριστερό κινητήρα Α για 3 δευτερόλεπτα και τον σταματάει. Στη συνέχεια κινεί το δεξιό κινητήρα C για 3 δευτερόλεπτα και τον σταματάει. Επαναλαμβάνει αυτή τη διαδικασία συνεχώς.



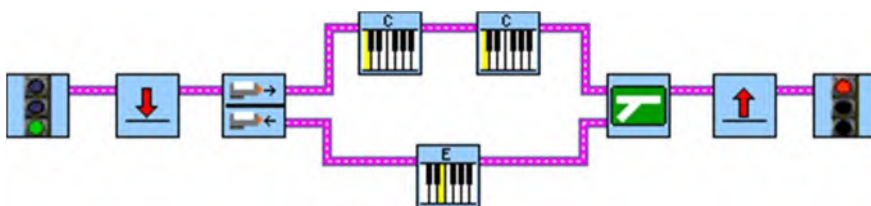
Δραστηριότητα 23η : ήχος συνέχεια

Το NXT παίζει έναν ήχο συνέχεια.



Δραστηριότητα 24η : νότες με διακλάδωση αφής και επανάληψη με άλμα

Όσο είναι πατημένος ο διακόπτης του αισθητήρα αφής το NXT παίζει συνέχεια δύο νότες C, διαφορετικά, αν αυτός είναι απελευθερωμένος δηλαδή, παίζει συνέχεια μία νότα E.



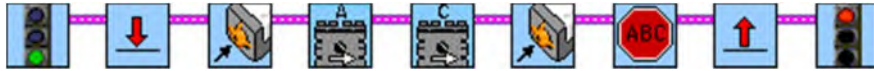
Δραστηριότητα 25η : έλεγχος λάμπας με αισθητήρα αφής

Το NXT ανάβει μία λάμπα όσο είναι πατημένος ο διακόπτης του αισθητήρα αφής.



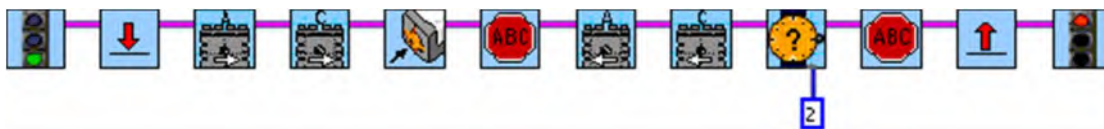
Δραστηριότητα 26η : ξεκίνημα και σταμάτημα με αισθητήρα αφής και επανάληψη με άλμα

Το όχημα-ρομπότ κινείται προς τα μπροστά όταν πατιέται ο διακόπτης του αισθητήρα αφής και σταματάει όταν ο διακόπτης του αισθητήρα αφής πατιέται ξανά.



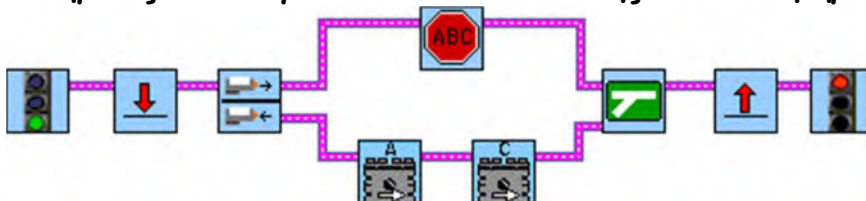
Δραστηριότητα 27η : εμπόδιο με αισθητήρα αφής

Το όχημα-ρομπότ κινείται προς τα μπροστά μέχρι να χτυπήσει σε ένα εμπόδιο, οπότε και κινείται προς τα πίσω για 2 δευτερόλεπτα. Επαναλαμβάνει αυτή τη διαδικασία συνεχώς.



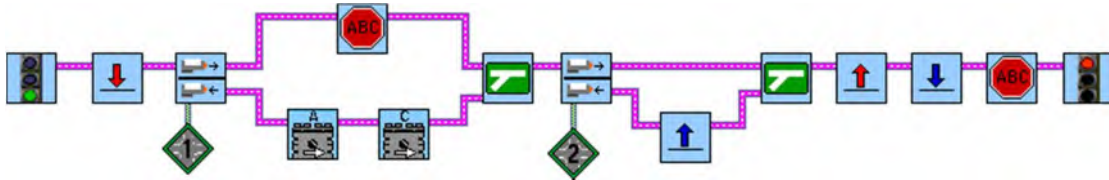
Δραστηριότητα 28η : ξεκίνημα και σταμάτημα με διακλάδωση αισθητήρα αφής και επανάληψη με άλμα

Το όχημα-ρομπότ κινείται προς τα μπροστά όταν ο διακόπτης του αισθητήρα αφής είναι πατημένος και σταματάει όταν ο διακόπτης του αισθητήρα αφής είναι απελευθερωμένος.



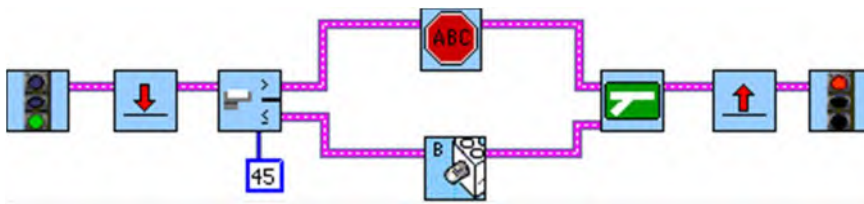
Δραστηριότητα 28η : ξεκίνημα και σταμάτημα με δυνατότητας τερματισμού με διακλάδωση αισθητήρα αφής και επανάληψη με άλμα (β' έκδοση)

Το όχημα-ρομπότ κινείται προς τα μπροστά όταν ο διακόπτης του αισθητήρα αφής 1 είναι πατημένος και σταματάει όταν ο διακόπτης του αισθητήρα αφής 1 είναι απελευθερωμένος. Το πρόγραμμα αυτό τερματίζεται με το πάτημα του διακόπτη του αισθητήρα αφής 2



Δραστηριότητα 29η : υλοποίηση συναγερμού με διακλάδωση αισθητήρα φωτός και επανάληψη με άλμα

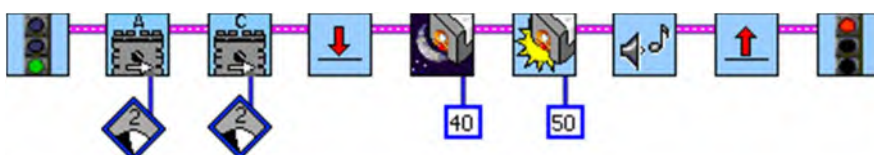
Κάθε φορά που η ένταση του φωτός που ανιχνεύεται από έναν αισθητήρα φωτός πέσει κάτω από την τιμή 45, το NXT ανάβει ένα λαμπάκι.



Δραστηριότητα 30η : κίνηση μπροστά και ήχος για κάθε μαύρη γραμμή

Το όχημα - ρομπότ κινείται προς τα μπροστά και κάθε φορά που περνάει από μία μαύρη γραμμή παίζει ένα ήχο.

Το όχημα κινείται αργά για να προλάβει ο αισθητήρας φωτός να ανιχνεύσει τη μαύρη γραμμή. Μετά την εντολή αναμονής για σκοτάδι έχουμε προσθέσει και μία εντολή αναμονής για φως για να είμαστε σίγουροι ότι το όχημα έχει προσπεράσει μία γραμμή ώστε να μη μετρήσει την ίδια γραμμή πάνω από μία φορά.



Λίγα λόγια για τη δομή επανάληψης (επανάλαβε μέχρι)

Υπάρχουν δύο διαφορετικές μορφές επανάληψης. Στην πρώτη μορφή γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων που πρέπει αν γίνουν, ενώ στη δεύτερη επαναλαμβάνεται μία διαδικασία μέχρι να ικανοποιηθεί ένας στόχος ή όσο ικανοποιείται μία συνθήκη.

Επανάληψη n φορές

Συχνά μέσα στο πρόγραμμα χρειάζεται να πούμε στο ρομπότ μας να επαναλάβει τη μια συγκεκριμένη διαδικασία περισσότερες από μία φορές. Για να το πετύχουμε αυτό χρησιμοποιούμε μία δομή επανάληψης.

Το ζευγάρι εικονιδίων της επανάληψης μας επιτρέπει να επαναλάβουμε ένα μέρος του προγράμματος μας, κάποιες συγκεκριμένες φορές. Με αυτό τον τρόπο δεν χρειάζεται να ξαναγράψουμε το ίδιο κομμάτι του προγράμματος μας (βρόχο) ξανά και ξανά.

Για να το πετύχουμε αυτό στο Robolab τοποθετούμε



το εικονίδιο αρχή επανάληψης στην αρχή του βρόχου



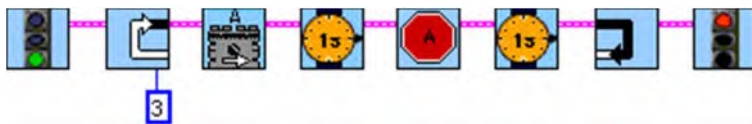
και το εικονίδιο τέλος επανάληψης στο τέλος του βρόχου μας.

Σε αυτήν την περίπτωση πρέπει να χρησιμοποιήσουμε και έναν τροποποιητή αριθμητικής τιμής από την υποπαλέτα τροποποιητών, για να καθορίσουμε τον αριθμό των επαναλήψεων που επιθυμούμε να συμβούν.



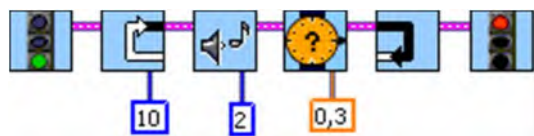
Παράδειγμα :

Στο παρακάτω πρόγραμμα χρησιμοποιείται η δομή επανάληψης για το μέρος του προγράμματος που ξεκινάει τον κινητήρα A για χρόνο 1 δευτερόλεπτο και τον σταματάει για χρόνο 1 δευτερόλεπτο. Αυτό θα συμβεί 3 φορές. Αν δεν χρησιμοποιούσαμε δομή επανάληψης θα έπρεπε να γράψουμε το ίδιο κομμάτι του προγράμματος 3 ξεχωριστές φορές.



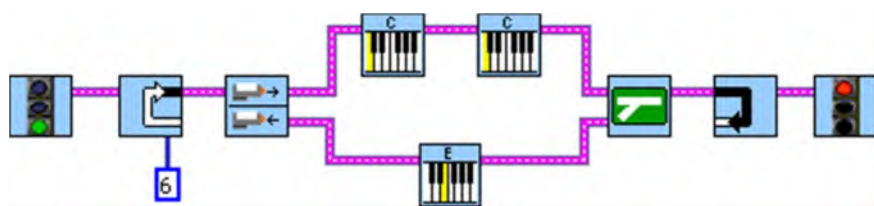
Δραστηριότητα 31η : ήχος 10 φορές

Το NXT παίζει έναν ήχο 10 φορές.



Δραστηριότητα 32η : νότες με διακλάδωση αφής και επανάληψη 6 φορές

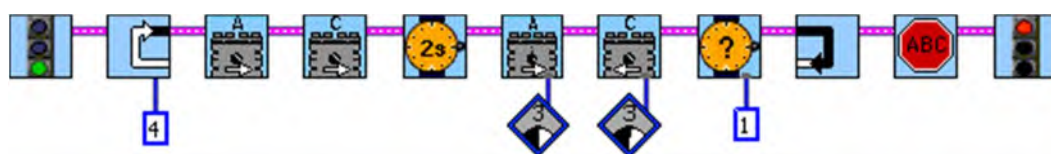
Όταν είναι πατημένος ο διακόπτης του αισθητήρα αφής το NXT παίζει δύο νότες C, διαφορετικά, αν αυτός είναι απελευθερωμένος δηλαδή, παίζει μία νότα E. Τα παραπάνω βήματα επαναλαμβάνονται 6 φορές.



Δραστηριότητα 33η : κίνηση σε τετράγωνο με επανάληψη

Το όχημα-ρομπότ κινείται έτσι ώστε να ακολουθήσει μία διαδρομή ή οποία έχει σχήμα τετραγώνου.

Για να κάνουμε το όχημα - ρομπότ μας να κινηθεί σε ένα τετράγωνο αρκεί να επαναλάβουμε τα δύο πρώτα βήματα 4 φορές.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τις δομές επανάληψης θα τον βρείτε στο σύνδεσμο : [Δομές Επανάληψης](http://kyrgeo.sites.sch.gr)

Λίγα λόγια για τις παράλληλες διεργασίες

Μέχρι τώρα τα προγράμματα που είδαμε περιείχαν μία μόνο διεργασία. Όταν λέμε διεργασία εννοούμε ένα σύνολο από εικονίδια τα οποία είναι ενωμένα μεταξύ τους σε μία σειρά συνεχόμενα. Μπορούμε όμως μέσα σε ένα πρόγραμμα να έχουμε περισσότερες από μία διεργασίες, οι οποίες θα εκτελεστούν παράλληλα. Τοποθετώντας ένα εικονίδιο παράλληλων διεργασιών σε ένα πρόγραμμα, το ρομπότ μας θα εκτελέσει 2 διαφορετικές εργασίες ταυτόχρονα.

Για να το πετύχουμε αυτό στο Robolab τοποθετούμε



το εικονίδιο της παράλληλης διεργασίας.



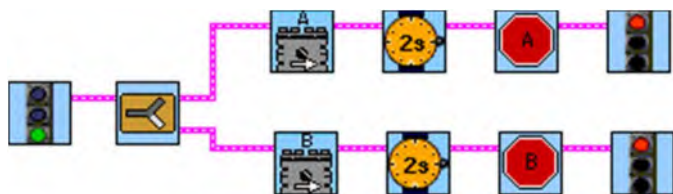
Προσοχή: Σε αυτήν την περίπτωση πρέπει να τοποθετήσουμε στο τέλος κάθε ξεχωριστής διεργασίας το εικονίδιο του τέλους προγράμματος,.

Μπορούμε να έχουμε ως και 10 παράλληλες διεργασίες μέσα σε ένα πρόγραμμα.

Προσοχή: Όταν σε ένα πρόγραμμα μας χρησιμοποιήσουμε παράλληλες διεργασίες πρέπει να προσέχουμε ώστε να μην τοποθετούμε τον ίδιο κινητήρα ή λαμπτήρα σε δύο διεργασίες. Θα συμβεί σύγκρουση αντιφατικών εντολών.

Παράδειγμα :

Στο παρακάτω πρόγραμμα χρησιμοποιείται το εικονίδιο της παράλληλης διεργασίας. Σε αυτό το πρόγραμμα ο κινητήρας A και ο κινητήρας B ξεκινούν να λειτουργούν ταυτόχρονα.

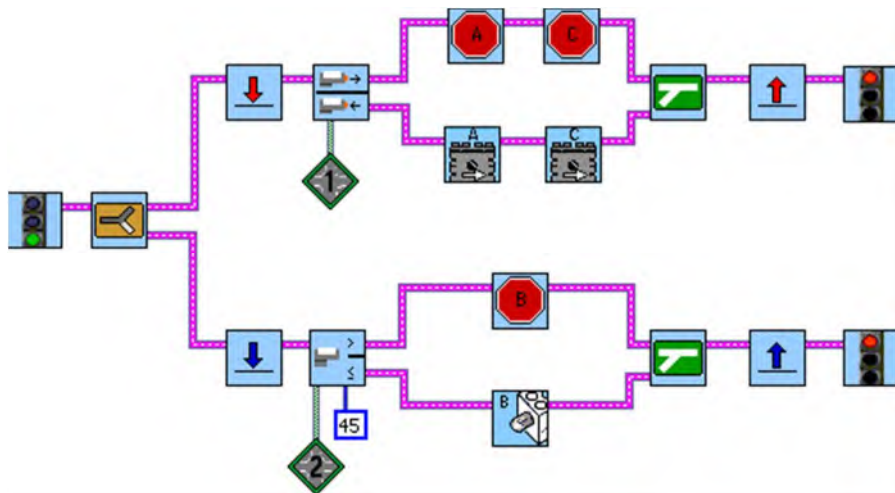


Δραστηριότητα 34η : παράλληλες διεργασίες

Το όχημα-ρομπότ εκτελεί 2 διεργασίες παράλληλα :

Στην πρώτη διεργασία το όχημα-ρομπότ κινείται προς τα μπροστά αν ο διακόπτης του αισθητήρα αφής είναι πατημένος και σταματάει αν ο διακόπτης του αισθητήρα αφής είναι απελευθερωμένος.

Στη δεύτερη διεργασία αν η ποσότητα φωτός που ανιχνεύεται από έναν αισθητήρα φωτός πέσει κάτω από την τιμή 45, ανάβει ένα λαμπάκι.



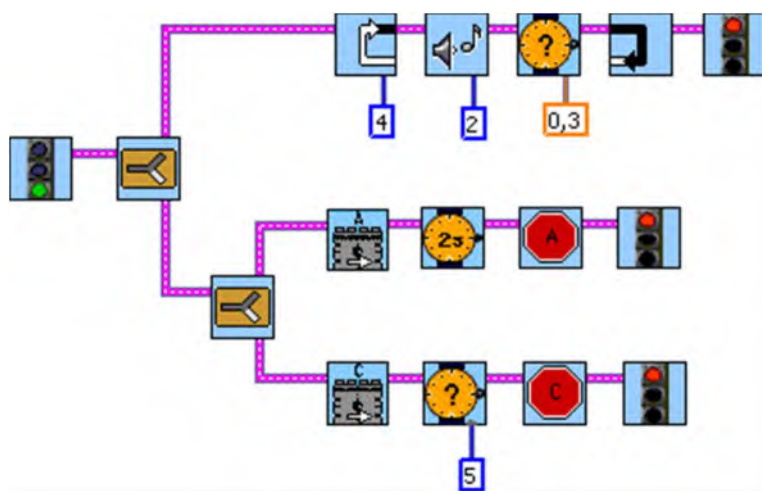
Δραστηριότητα 35η : τρεις παράλληλες διεργασίες

Το όχημα-ρομπότ εκτελεί 3 διεργασίες παράλληλα :

Στην πρώτη διεργασία παίζει ένα ήχο 4 φορές.

Στη δεύτερη διεργασία κινεί τον κινητήρα A προς τα μπροστά για 2 δευτερόλεπτα και μετά τον σταματάει.

Στην τρίτη διεργασία κινεί τον κινητήρα C προς τα πίσω για 5 δευτερόλεπτα και μετά τον σταματάει.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τις παράλληλες διεργασίες θα τον βρείτε στο σύνδεσμο : [Παράλληλες Διεργασίες](#)

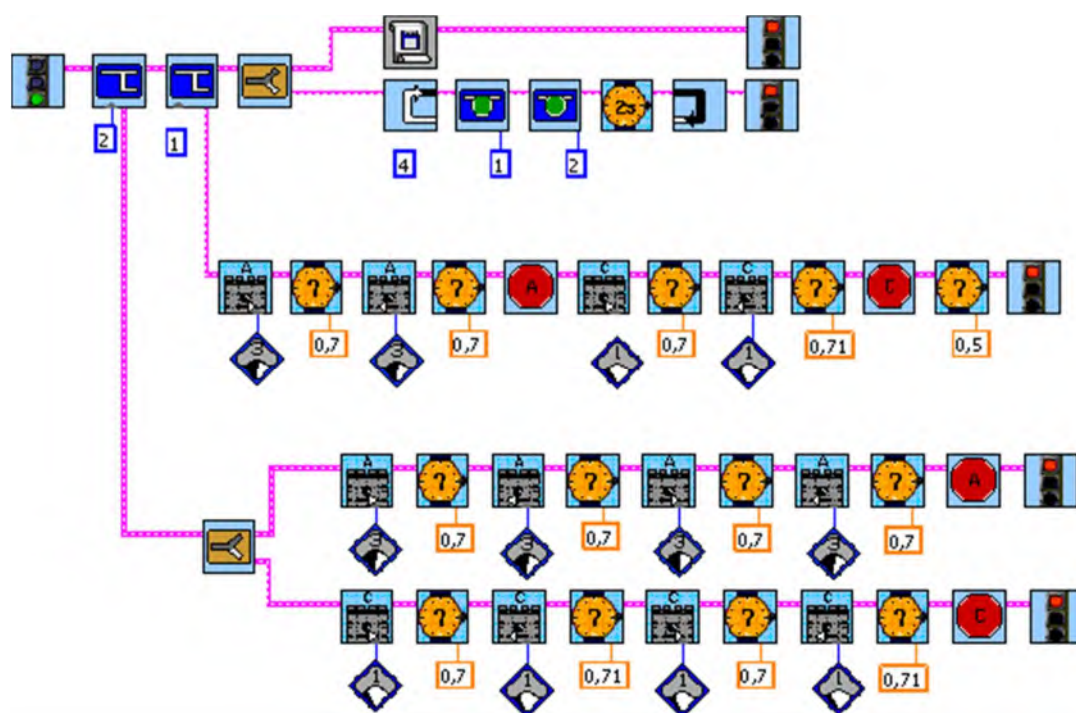


Αν θέλουμε σε ένα πρόγραμμα να δηλώσουμε πάνω από μία υπορουτίνα τοποθετούμε σε κάθε εντολή δημιουργίας υπορουτίνας, από ένα μετατροπέα σταθερής αριθμητικής τιμής.



Δραστηριότητα 36η : χορευτικές κινήσεις

Το όχημα-ρομπότ εκτελεί χορευτικές κινήσεις



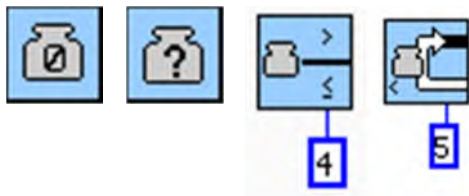
Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τις υπορουτίνες θα τον βρείτε στο σύνδεσμο :

Υπορουτίνες

Λίγα λόγια για τις μεταβλητές

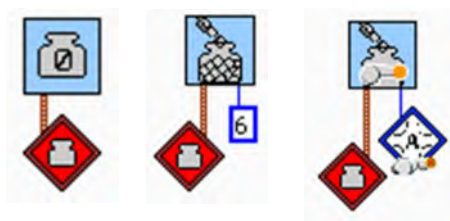
Μια μεταβλητή (Container) περιέχει έναν αριθμό ο οποίος μεταβάλλεται συνεχώς και τον οποίο δε γνωρίζουμε μέχρι τη στιγμή που θα εκτελεστεί το πρόγραμμα μας

Τα παρακάτω εικονίδια μας δίνουν τη δυνατότητα να χειριστούμε τις μεταβλητές μέσα στο πρόγραμμα μας.



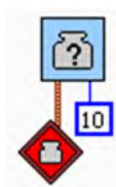
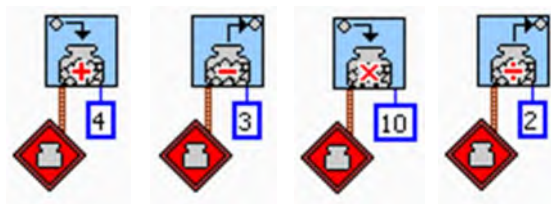
Υπάρχουν 3 διαφορετικές μεταβλητές : η κόκκινη, η μπλε και η κίτρινη.

Για να καθορίσουμε σε ποια μεταβλητή αναφερόμαστε χρησιμοποιούμε τα παρακάτω εικονίδια τροποποιητών μεταβλητής

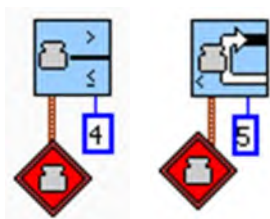


Μπορούμε να μηδενίσουμε μία μεταβλητή, να τοποθετήσουμε εμείς κάποια τιμή ή να την προγραμματίσουμε να πάρει την τιμή που θα έχει εκείνη τη στιγμή ένας αισθητήρας, όπως πχ ένας αισθητήρας περιστροφής που είναι συνδεδεμένος στη θύρα 2.

Υπάρχει η δυνατότητα να υλοποιήσουμε απλές μαθηματικές πράξεις με το περιεχόμενο μιας μεταβλητής. Τα παρακάτω εικονίδια χρησιμοποιούνται για να προσθέσουμε, να αφαιρέσουμε, να πολλαπλασιάσουμε ή να διαιρέσουμε τις τιμές που περιέχουν οι μεταβλητές με άλλους αριθμούς.

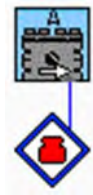


Η αναμονή με τη χρήση μιας μεταβλητής βάζει το ρομπότ μας σε κατάσταση αναμονής μέχρι να ικανοποιηθεί η συνθήκη για τη μεταβλητή.



Η διακλάδωση με μεταβλητή δίνει τη δυνατότητα λήψης απόφασης, βασισμένης στην τιμή μιας μεταβλητής ενώ η επανάληψη με μεταβλητή επαναλαμβάνει ένα μέρος του προγράμματος μέχρι να ικανοποιηθεί μία συνθήκη για τη μεταβλητή.

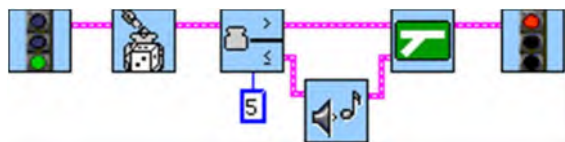
Για να αποκτήσουμε πρόσβαση στο περιεχόμενο μιας μεταβλητής, χρησιμοποιούμε το εικονίδιο του τροποποιητή της τιμής της μεταβλητής.



Παράδειγμα αυτό το ζευγάρι των εικονιδίων θέτει το επίπεδο ισχύος του κινητήρα A στην τιμή που έχει εκείνη τη στιγμή η κόκκινη μεταβλητή.

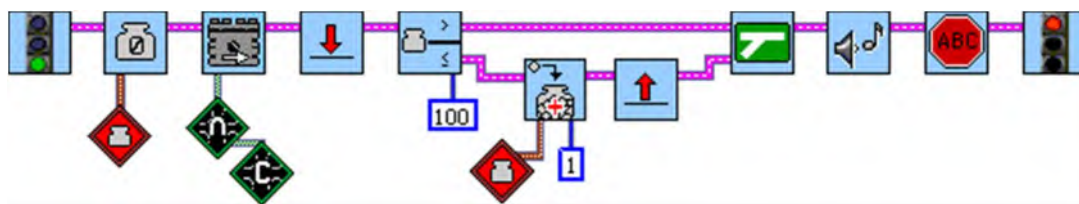
Παράδειγμα :

Στο παρακάτω πρόγραμμα η κόκκινη μεταβλητή παίρνει μία τυχαία τιμή και στη συνέχεια ανάλογα με την τιμή αυτή γίνεται αναπαραγωγή ενός ήχου.



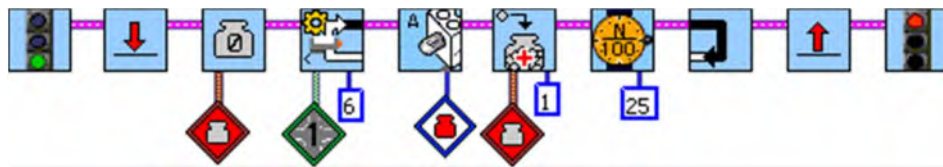
Δραστηριότητα 37η : αναμονή για μετρητή

Το όχημα-ρομπότ ξεκινάει να κινείται μπροστά και στη συνέχεια παίζει έναν ήχο και σταματάει όταν ένας μετρητής φτάσει από το 0 στο 100.



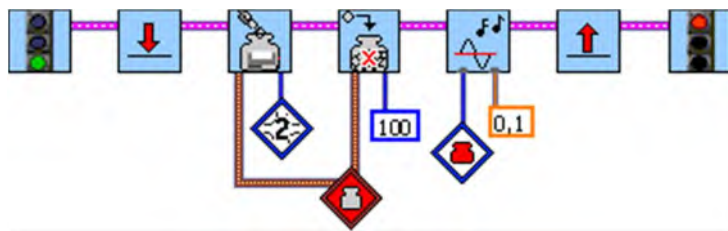
Δραστηριότητα 38η : αλλαγή φωτεινότητας λάμπας

Το NXT αυξάνει τη φωτεινότητα μίας λάμπας κάθε φορά που πατιέται ο διακόπτης ενός αισθητήρα αφής μέχρι η ένταση της λάμπας να φτάσει στην πλήρη ισχύ.



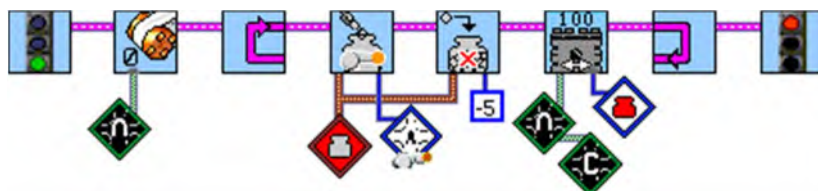
Δραστηριότητα 39η : μετατροπή φωτεινότητας σε νότες (sequencer)

Το NXT αναπαράγει μουσικές νότες μετατρέποντας τις τιμές που διαβάζει ένας αισθητήρας φωτός σε ηχητικές συχνότητες.



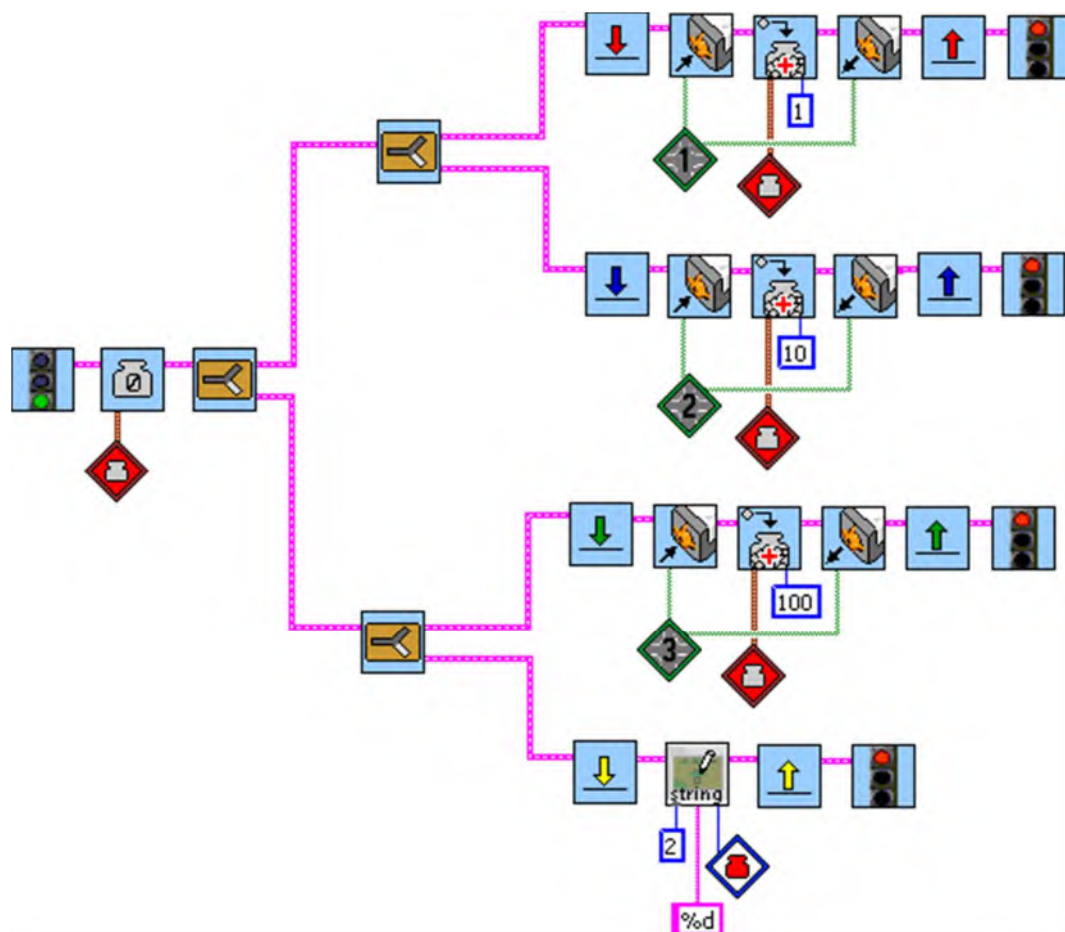
Δραστηριότητα 40η : αντίσταση στη μετακίνηση

Όταν πάμε να μετακινήσουμε με τα χέρια μας το όχημα-ρομπότ αυτό αντιστέκεται και παραμένει στην αρχική του θέση.



Δραστηριότητα 41η : μέτρηση πατημάτων

Το NXT υπολογίζει έναν αριθμό ως εξής : κάθε φορά που πατιέται ο διακόπτης του αισθητήρα αφής 1 προστίθεται μία μονάδα, κάθε φορά που πατιέται ο διακόπτης του αισθητήρα αφής 2 προστίθεται μία δεκάδα και κάθε φορά που πατιέται ο διακόπτης του αισθητήρα αφής 3 προστίθεται μία εκατοντάδα. Ο αριθμός αυτός εμφανίζεται στην οθόνη του NXT.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τις μεταβλητές θα τον βρείτε στο σύνδεσμο :

[Μεταβλητές](#)

Λίγα λόγια για τους χρονομετρητές

Το εσωτερικό ρολόι του NXT μπορεί να χρησιμοποιηθεί ως ακόμα μία είσοδο για να ελέγξουμε το πρόγραμμα μας.

Οι χρονομετρητές (Timers) είναι ένας τρόπος να ελέγξουμε τη ροή του προγράμματος χρησιμοποιώντας την παράμετρο του χρόνου που πέρασε.

Τα παρακάτω εικονίδια μας δίνουν τη δυνατότητα να ελέγξουμε τη ροή του προγράμματος μας με βάση το χρόνο που τρέχει.

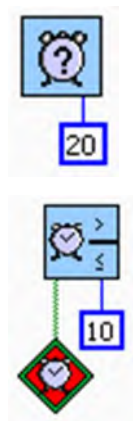


Υπάρχουν 3 διαφορετικοί χρονομετρητές : η κόκκινη, η μπλε και η κίτρινη.

Για να καθορίσουμε σε ποιο χρονομετρητή αναφερόμαστε χρησιμοποιούμε τα παρακάτω εικονίδια τροποποιητών χρονομετρητή.



Τους χρονομετρητές μπορούμε να τους χρησιμοποιήσουμε για να βάλουμε το πρόγραμμα να περιμένει μέχρι να περάσει ένα καθορισμένο χρονικό διάστημα, προτού συνεχίσει με την εκτέλεση της επόμενης εντολής. Είναι σημαντικό να μηδενίζουμε ένα χρονομετρητή πριν να ξεκινήσουμε. Το χρονικό διάστημα το αναγράφουμε σε δέκατα του δευτερολέπτου, έτσι τα 2 δευτερόλεπτα πρέπει να αναγραφούν ως την αριθμητική τιμή 20.



Η αναμονή με τη χρήση χρονομετρητή βάζει το ρομπότ μας σε κατάσταση αναμονής μέχρι ο χρονομετρητής να φτάσει σε μία συγκεκριμένη χρονική τιμή.

Η διακλάδωση με χρονομετρητή επιτρέπει στο ρομπότ μας να λάβει μία απόφαση με βάση το χρόνο που έχει διανυθεί μέχρι τη στιγμή του ελέγχου.

Μπορούμε να έχουμε πρόσβαση στην αριθμητική τιμή ενός χρονομετρητή χρησιμοποιώντας το εικονίδιο του τροποποιητή της τιμής του χρονομετρητή. Αυτό είναι χρήσιμο για να αποθηκεύουμε την τιμή από έναν χρονομετρητή σε μία μεταβλητή.

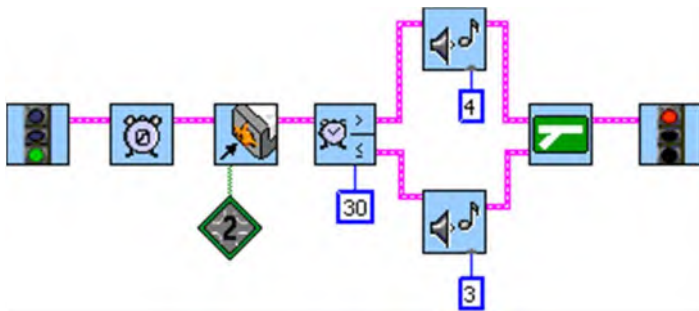


Παραδείγματα :

α) Αυτό το πρόγραμμα μηδενίζει στην αρχή τον κόκκινο χρονομετρητή, και στη συνέχεια περιμένει να περάσουν 2 δευτερόλεπτα οπότε και ξεκινάει τον κινητήρα A.



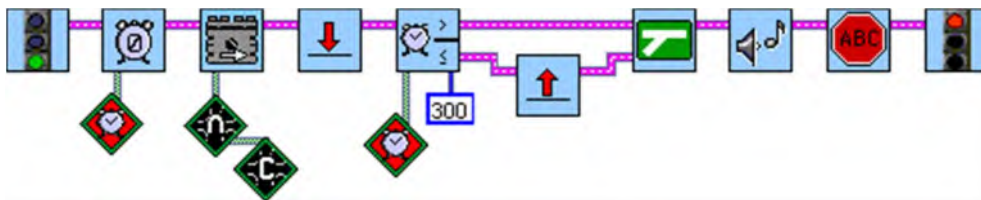
β) Αυτό το πρόγραμμα.



αφού στην αρχή μηδενίζει τον κόκκινο χρονομετρητή, περιμένει μέχρι να πατηθεί ο διακόπτης του αισθητήρα αφής. Αν ο διακόπτης του αισθητήρα αφής πατηθεί στα 3 πρώτα δευτερόλεπτα ή και πιο νωρίς από τη στιγμή που θα ξεκινήσει η εκτέλεση του προγράμματος (τα 3 δευτερόλεπτα αναγράφονται ως την αριθμητική τιμή 30), τότε το πρόγραμμα θα αναπαράγει τον ήχο descending sweep. Αν έχουν περάσει πάνω από τρία δευτερόλεπτα τότε θα αναπαράγει τον ήχο rising sweep.

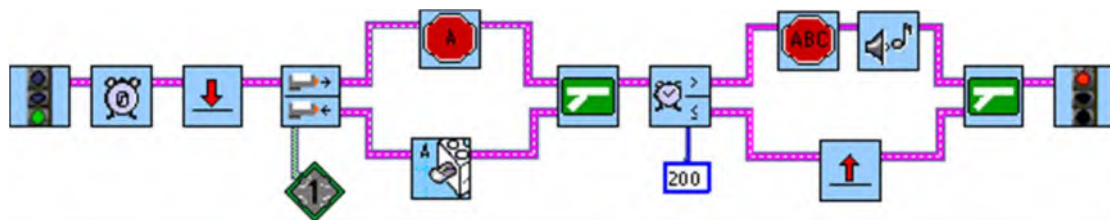
Δραστηριότητα 42η : αναμονή αριθμού δευτερολέπτων

Το όχημα-ρομπότ ξεκινάει να κινείται προς τα μπροστά και όταν περάσουν 30 δευτερόλεπτα παίζει έναν ήχο και σταματάει.



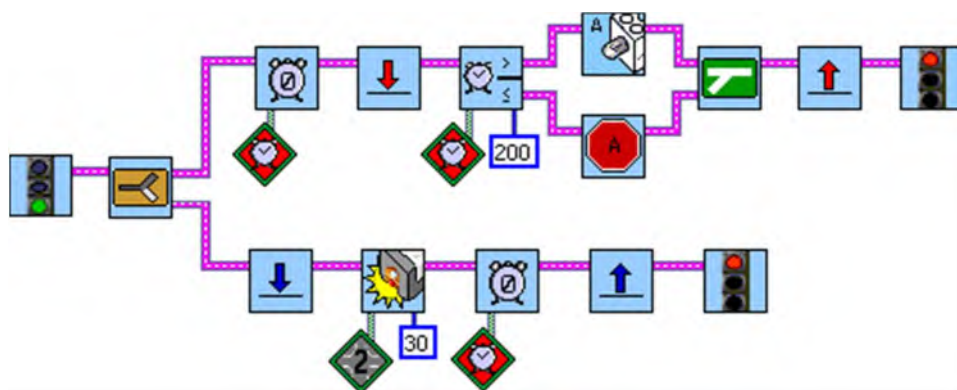
Δραστηριότητα 43η : έλεγχος λάμπας

Το NXT ελέγχει ένα λαμπάκι Α με τη βοήθεια ενός αισθητήρα αφής, όσο το χρονικό διάστημα είναι κάτω από 20 δευτερόλεπτα.



Δραστηριότητα 44η : έλεγχος λάμπας με χρονοκαθυστέρηση

Το NXT ανάβει ένα λαμπάκι Α όταν επικρατεί σκοτάδι για πάνω από 2 δευτερόλεπτα.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τους χρονομετρητές θα τον βρείτε στο σύνδεσμο:

[Χρονομετρητές](#)

Λίγα λόγια για την αποστολή και λήψη μηνυμάτων

Οι μικροεπεξεργαστές NXT έχουν τη δυνατότητα να επικοινωνήσουν μεταξύ τους και να ανταλλάξουν μηνύματα τα οποία μπορεί να περιέχουν κάποιο κείμενο, μία αριθμητική τιμή ή μία λογική τιμή. Η αποστολή και η λήψη των μηνυμάτων γίνεται μέσω της υπέρυθρης διασύνδεσης από ένα τούβλο NXT σε οποιοδήποτε άλλο τούβλο NXT που θα βρεθεί μέσα στην εμβέλεια του σήματος



Για να στείλουμε ένα μήνυμα από ένα μικροϋπολογιστή NXT σε έναν άλλο χρησιμοποιούμε το εικονίδιο της αποστολής μηνύματος.



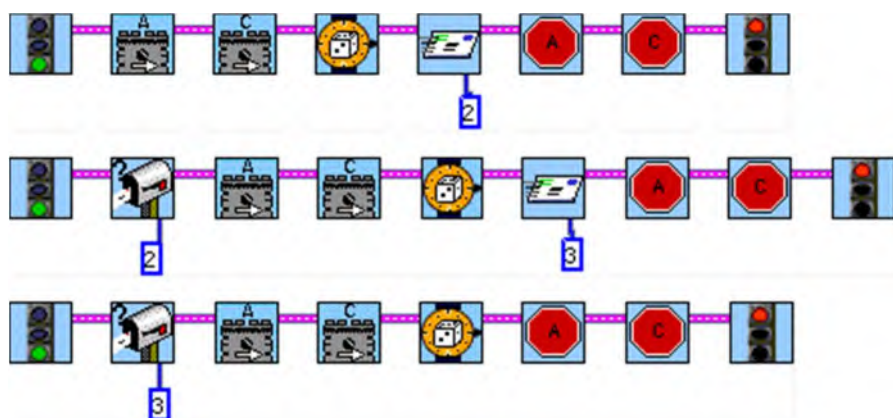
Για να λάβει ένας μικροϋπολογιστής NXT ένα μήνυμα από έναν άλλο μικροϋπολογιστή NXT χρησιμοποιούμε το εικονίδιο της λήψης μηνύματος.

Με τη δυνατότητα αποστολής και λήψης μηνυμάτων μπορούμε ελέγχουμε τη συμπεριφορά ενός ρομπότ ασύρματα στέλνοντας μηνύματα με διάφορες τιμές στις οποίες το ρομπότ θα ανταποκρίνεται αναλόγως.

Επίσης ένα ρομπότ μπορεί να ειδοποιήσει ένα άλλο για την ολοκλήρωση μίας ενέργειας, ώστε να αναλάβει με τη σειρά του κάποια δράση - για την οποία έπρεπε πρώτα να περιμένει την ολοκλήρωση μίας άλλης ενέργειας.

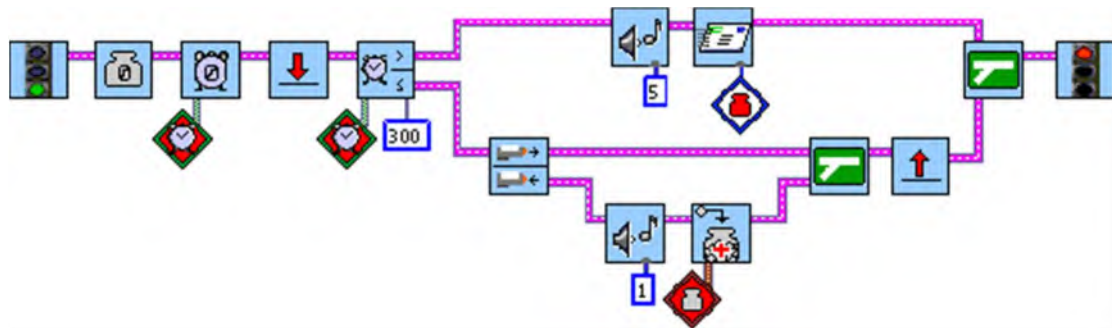
Δραστηριότητα 45η : σκυταλοδρομία

Έχουμε 3 οχήματα- ρομπότ τα οποία κινούνται με τη σειρά, το ένα μετά το άλλο, για ένα τυχαίο χρονικό διάστημα και στη συνέχεια ειδοποιούν το επόμενο όχημα να ξεκινήσει με τη σειρά του.



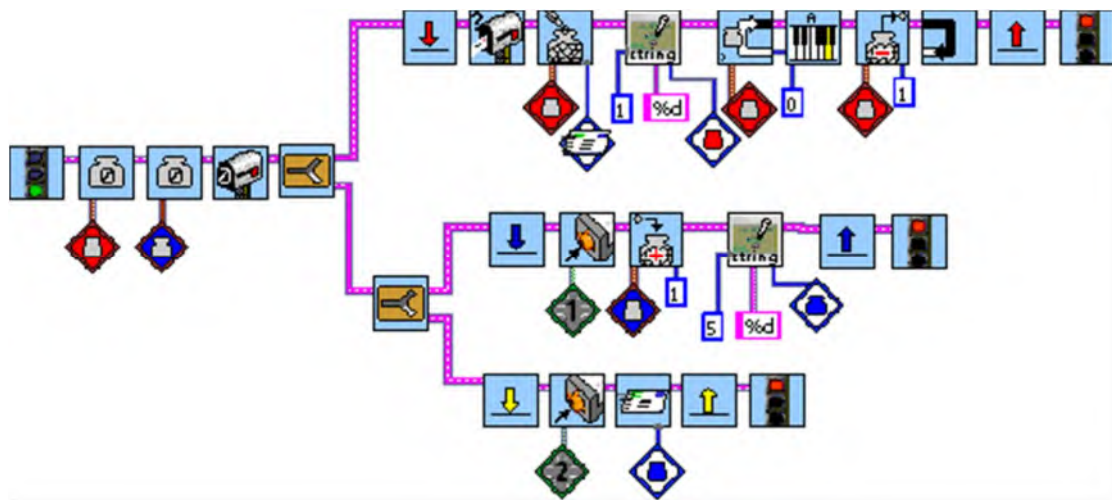
Δραστηριότητα 46η : δημιουργία και αποστολή μηνύματος

Το NXT στέλνει ένα μήνυμα το οποίο προκύπτει από τον αριθμό των πατημάτων του διακόπτη ενός αισθητήρα αφής μέσα σε 30 δευτερόλεπτα.



Δραστηριότητα 47η : αναπαραγωγή νότας σύμφωνα με εισερχόμενο μήνυμα

Το NXT παίζει μία νότα τόσες φορές όσες και ο αριθμός που λαμβάνει ως μήνυμα από ένα άλλο NXT. Ο αριθμός αυτός προκύπτει από τον αριθμό των πατημάτων του διακόπτη του αισθητήρα αφής 1 και αποστέλλεται όταν πατηθεί ο διακόπτης του αισθητήρα αφής 2



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για την αποστολή και λήψη μηνυμάτων θα τον βρείτε στο σύνδεσμο : [Μηνύματα](#)

Για περισσότερες πληροφορίες για τις εντολές του προγραμματιστικού περιβάλλοντος NXT-G ανατρέξτε στο ΠΑΡΑΡΤΗΜΑ Ι.



Ενότητα 10η

Σύνθετες Δραστηριότητες για το Robolab

Α. Αποφυγή εμποδίων

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να παρακάμπτει τα εμπόδια που θα συναντήσει στην πορεία του.

Δραστηριότητα 1η : αποφυγή εμποδίου με 1 αισθητήρα αφής

Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας αφής.

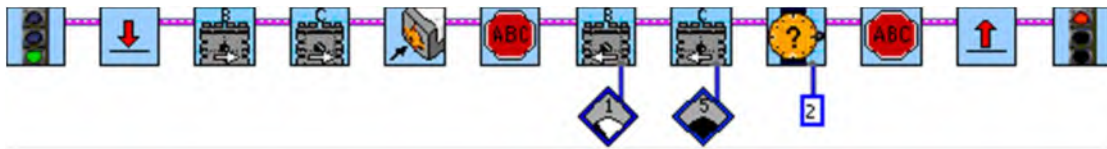
Τον αισθητήρα αφής αυτόν τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Ανάλυση προβλήματος :

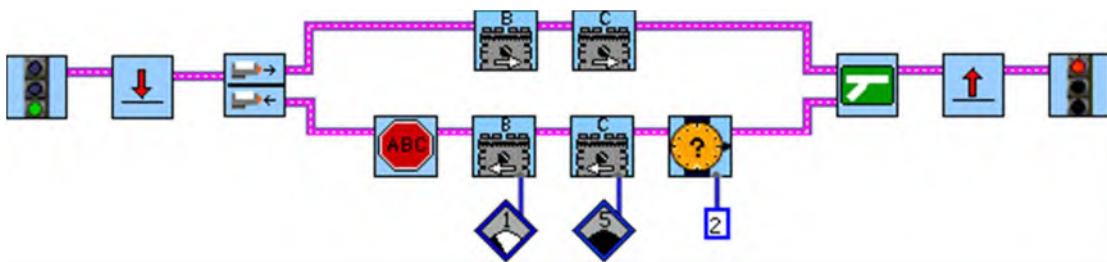
Το όχημα - ρομπότ μας κινείται μπροστά, σε ευθύγραμμη πορεία. Με αυτό τον τρόπο κάποια στιγμή θα συναντήσει κάποιο εμπόδιο και θα παραμείνει κολλημένο στο σημείο εκείνο.

α) Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει αισθητήρα αφής στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι έχει συναντήσει ένα εμπόδιο στην πορεία του.

Για να είναι σε θέση το ρομπότ μας να προσπεράσει το εμπόδιο αυτό και να συνεχίσει την πορεία του μπορούμε να του πούμε να πάει πίσω για λίγο, εκτελώντας στροφή προς μία κατεύθυνση πχ δεξιά. Στη συνέχεια να συνεχίσει ευθεία και να επαναλάβει την ίδια διαδικασία για κάθε εμπόδιο.



β. Το ίδιο πρόβλημα με διακλάδωση αφής :



Δραστηριότητα 2η : αποφυγή εμποδίου με 2 αισθητήρες αφής

Αυτή τη φορά στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκονται τοποθετημένοι δύο αισθητήρες αφής. Οι αισθητήρες αφής είναι τοποθετημένοι ο ένας στην αριστερή και ο άλλος στη δεξιά μεριά του προφυλακτήρα.

Τον αριστερό αισθητήρα αφής τον συνδέουμε στη θύρα εισόδου 1 ενώ το δεξιό στη θύρα εισόδου 3 του NXT.

Τοποθετώντας δύο αισθητήρες αφής το όχημα είναι σε θέση να παρακάμπτει, με πιο αποτελεσματικό τρόπο, τα εμπόδια που θα συναντήσει στην πορεία του.

Ανάλυση προβλήματος :

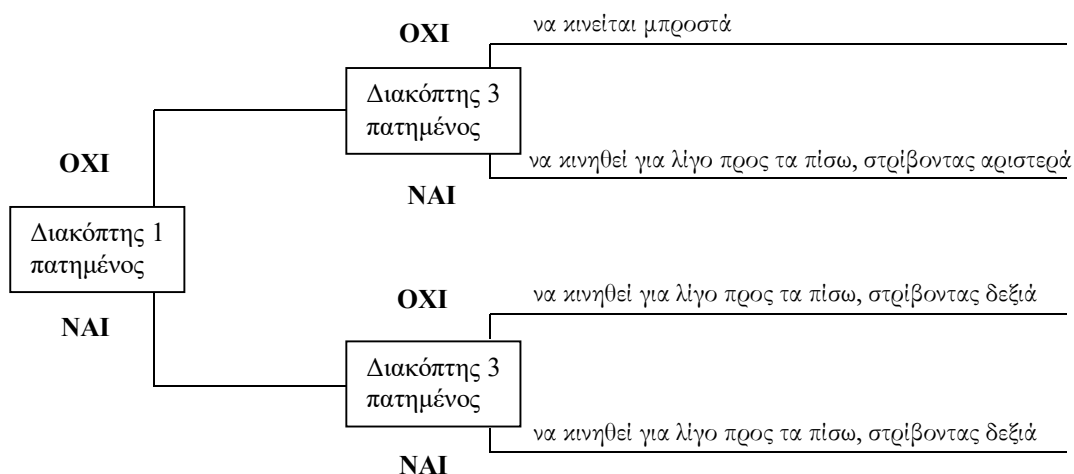
Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει δύο αισθητήρες αφής, δεξιά και αριστερά, στο μπροστινό του μέρος "είναι σε θέση να καταλάβει", όχι μόνο ότι έχει συναντήσει ένα εμπόδιο στην πορεία του, αλλά και σε ποιά πλευρά του.

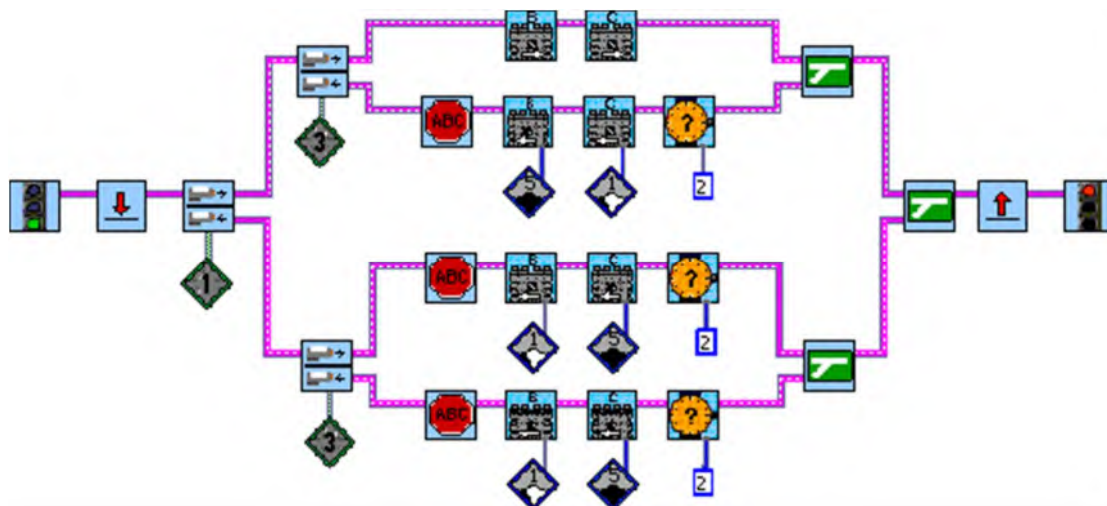
Μπορούμε τώρα να του πούμε να πάει πίσω για λίγο και να στρίψει ελαφρώς ανάλογα με τον αισθητήρα που πατήθηκε. Στη συνέχεια να συνεχίσει ευθεία και να επαναλάβει την ίδια διαδικασία για κάθε εμπόδιο.

Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

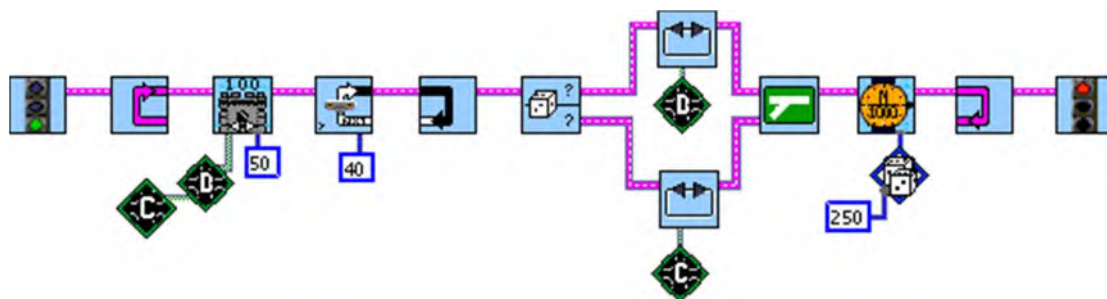
Αριστερός Διακόπτης 1 πατημένος	Δεξιός Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	να κινείται μπροστά
OXI	NAI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας αριστερά
NAI	OXI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας δεξιά
NAI	NAI	να κινηθεί για λίγο προς τα πίσω, στρίβοντας πχ δεξιά

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής



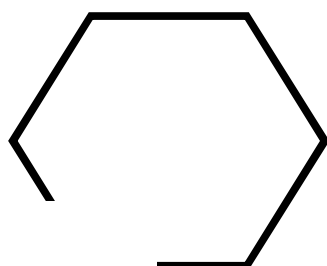


Στην περίπτωση που αντί για τον αισθητήρα αφής χρησιμοποιήσουμε έναν αισθητήρα υπερήχων τα πράγματα γίνονται πιο απλά, αφού τώρα το όχημα μας δεν χρειάζεται πλέον να πέσει πάνω στο εμπόδιο για να το αντιληφθεί. Μπορεί να εντοπίσει ένα εμπόδιο το οποίο βρίσκεται στην πορεία της κίνησης του, από αρκετά μεγάλη απόσταση και να το παρακάμψει εκτελώντας μία μικρή στροφή και συνεχίζοντας πάλι μπροστά. Μπορούμε μάλιστα να προγραμματίσουμε το όχημα μας, να στρίβει δεξιά ή αριστερά με τυχαίο τρόπο και για τυχαίο χρονικό διάστημα.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για την αποφυγή εμποδίων θα τον βρείτε στο σύνδεσμο: [Αποφυγή εμποδίων](#)

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να βρει την έξοδο από μία σπηλιά και να βγει από αυτήν.



Τα παραπάνω προγράμματα με κατάλληλες τροποποιήσεις στις παραμέτρους τους, μπορούμε να τα χρησιμοποιήσουμε και στην περίπτωση που θέλουμε το όχημα-ρομπότ μας να βγει από έναν κλειστό χώρο. Το χώρο αυτό μπορούμε να τον κατασκευάσουμε, φράζοντας τον με τοιχίο ύψους τουλάχιστον 5 εκατοστά. Στο χώρο αυτόν αφήνουμε μία έξοδο τουλάχιστον 30 εκατοστά.

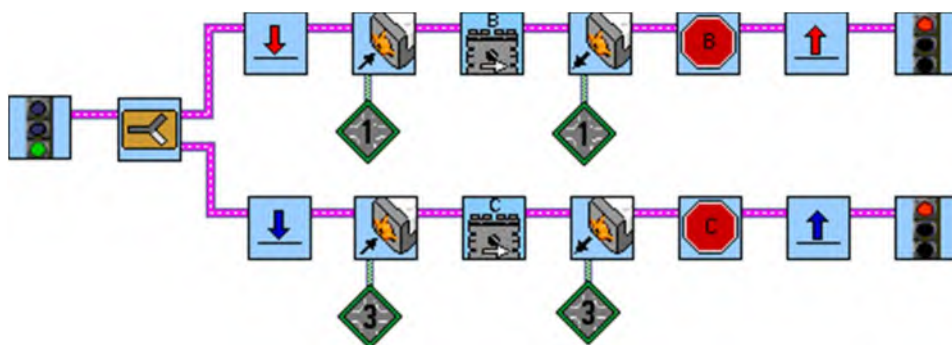
Β. Έλεγχος κίνησης με αισθητήρες αφής

Θέλουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να μπορούμε να το οδηγήσουμε με χειριστήριο να ακολουθεί μία διαδρομή που επιθυμούμε.

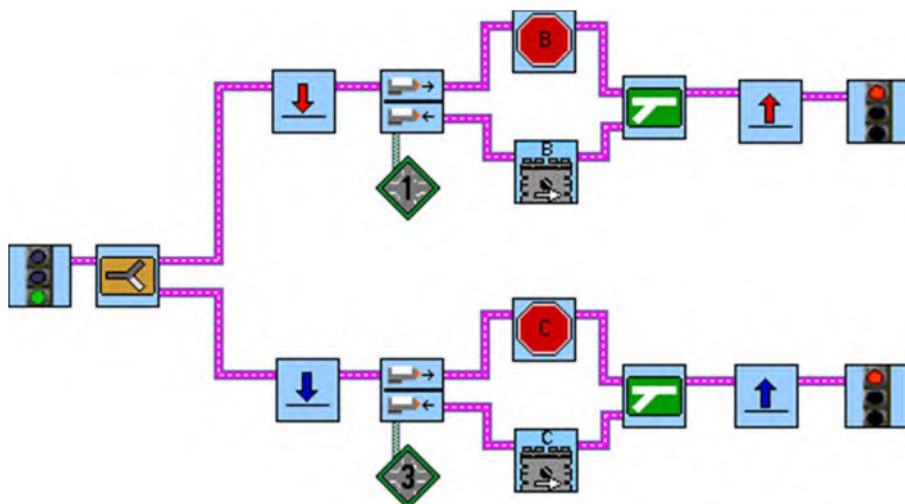
Δραστηριότητα 4η : έλεγχος κίνησης με 2 αισθητήρες αφής με παράλληλες διεργασίες

Συνδέουμε στο όχημα 2 αισθητήρες αφής με μακρύ καλώδιο. Τους αισθητήρες αφής τους συνδέουμε στις θύρες εισόδου 1 και 3 του NXT με τη βοήθεια καλωδίων με μεγάλο μήκος ώστε να υπάρχει μία σχετική ελευθερία στην κίνηση.

α) Το όχημα μας μπορεί να κινηθεί μόνο μπροστά ευθεία ή εκτελώντας στροφή. Στο παρακάτω πρόγραμμα κάθε αισθητήρας αφής ελέγχει έναν κινητήρα. Χρησιμοποιούμε δύο παράλληλες διεργασίες. Στην πρώτη ο αισθητήρας αφής στη θύρα 1 ελέγχει τον κινητήρα Β και στη δεύτερη ο αισθητήρας αφής στη θύρα 3 ελέγχει τον κινητήρα C.

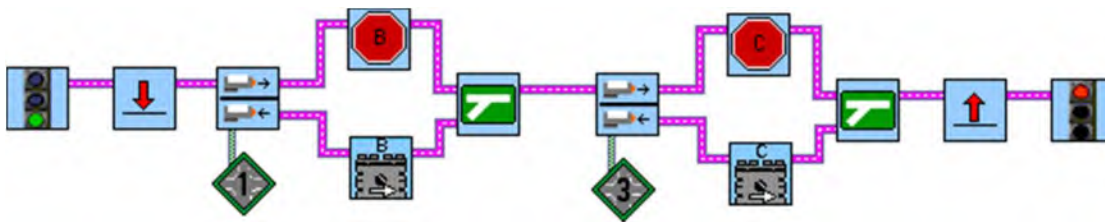


β. Το ίδιο πρόβλημα με διακλαδώσεις αισθητήρα αφής:



Δραστηριότητα 5η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση

Μπορούμε αντί για παράλληλες διεργασίες να χρησιμοποιήσουμε κάλλιστα μόνο μία διεργασία.



Δραστηριότητα 6η : έλεγχος κίνησης με 2 αισθητήρες αφής με διακλάδωση μέσα σε διακλάδωση

Είναι πιο βολικό για να έχουμε καλύτερο έλεγχο των συνδυασμών που προκύπτουν από δύο αισθητήρες αφής να χρησιμοποιήσουμε μία εντολή διακλάδωσης αισθητήρα αφής μέσα σε μία άλλη εντολή διακλάδωσης αφής.

Ανάλυση προβλήματος :

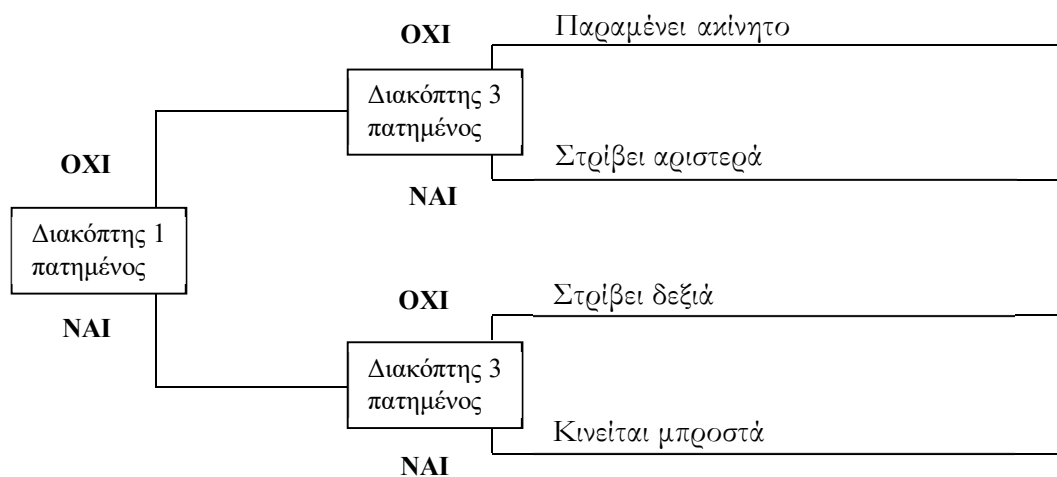
Θέλουμε να μπορούμε να οδηγούμε το ρομπότ μας, χρησιμοποιώντας το χειριστήριο ως τιμόνι και ταχύτητες, έτσι ώστε να μπορούμε να δίνουμε εντολή στο ρομπότ να κινείται μπροστά και να στρίβει δεξιά και αριστερά, ως τηλεκατευθυνόμενο.

Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

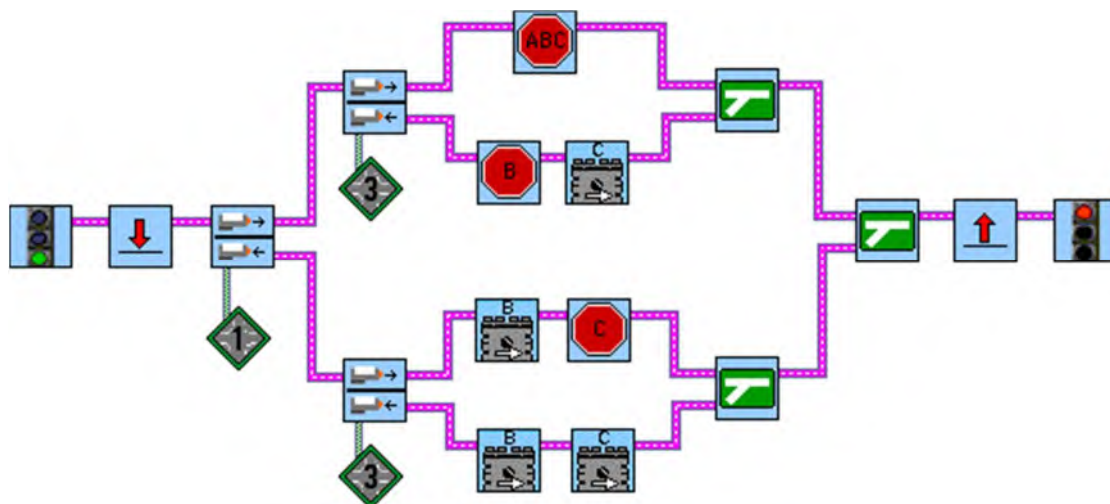
Διακόπτης 1 πατημένος	Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	Παραμένει ακίνητο
OXI	NAI	Στρίβει αριστερά
NAI	OXI	Στρίβει δεξιά
NAI	NAI	Κινείται μπροστά

Σενάρια Ρομποτικής με το Mindstorms NXT

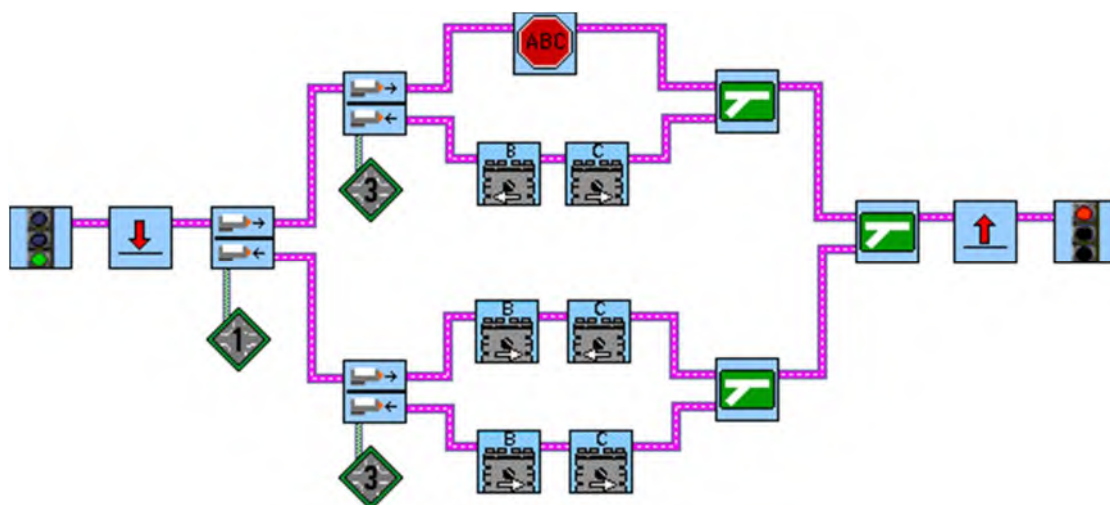
Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.



Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



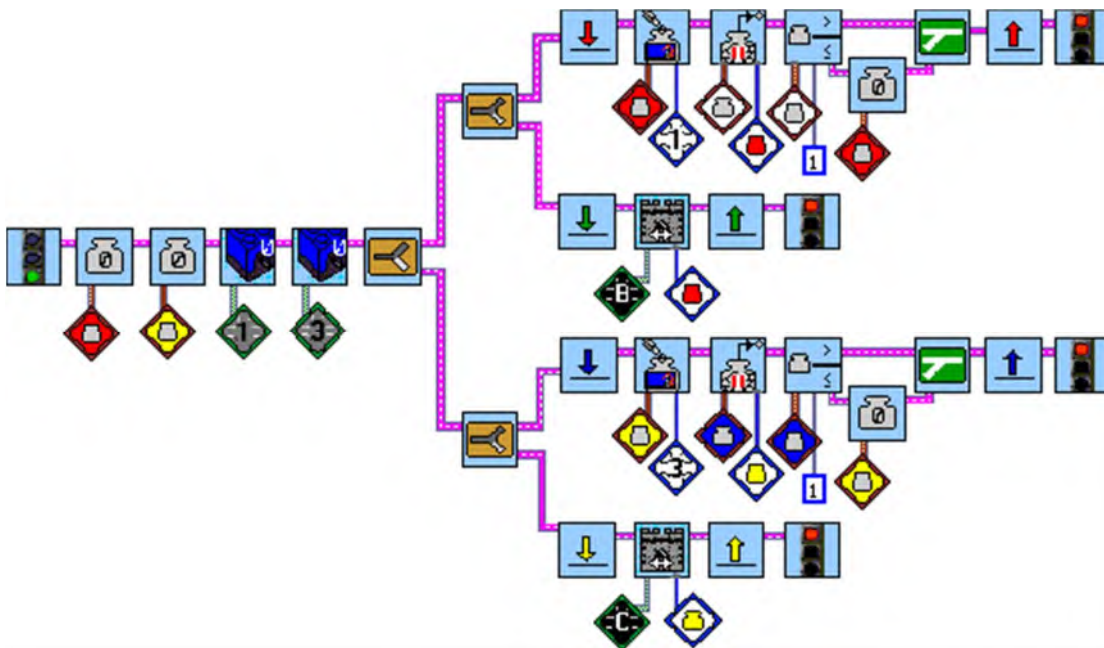
β. Το ίδιο πρόβλημα με δυνατότητα στροφής επιτόπου



Δραστηριότητα 7η : έλεγχος κίνησης με 2 αισθητήρες περιστροφής

Μπορούμε αντί για τους αισθητήρες αφής να χρησιμοποιήσουμε δύο αισθητήρες περιστροφής από το προηγούμενο πακέτο της σειράς NXT. Με τους αισθητήρες περιστροφής μπορούμε να κατευθύνουμε το όχημα μας μπροστά και πίσω με πιο βολικό τρόπο.

Ένα πρόβλημα με τον έλεγχο κίνησης με αισθητήρες περιστροφής είναι ότι λόγω της μεγάλης ευαισθησίας που τους χαρακτηρίζει, οι κινητήρες δεν θα σταματούν να κινούνται με την παραμικρή περιστροφή στους αισθητήρες περιστροφής. Για να διασφαλίσουμε ότι οι μικροκινήσεις των αισθητήρων περιστροφής θα αγνοούνται, καταφεύγουμε στο μηδενισμό της τιμής των μεταβλητών για μικρές τιμές περιστροφής του άξονα των αισθητήρων ανάμεσα από το -1 έως 1.



Δραστηριότητα 8η : έλεγχος κίνησης με 3 αισθητήρες αφής

Αυτή τη φορά συνδέουμε στο όχημα 3 αισθητήρες αφής με μακρύ καλώδιο. Τους αισθητήρες αφής τους συνδέουμε στις θύρες εισόδου 1, 2 και 3 του NXT με τη βοήθεια καλωδίων με μεγάλο μήκος ώστε να υπάρχει μία σχετική ελευθερία στην κίνηση.

Το πλεονέκτημα χρησιμοποιώντας 3 αισθητήρες αφής για να ελέγξουμε ένα όχημα - ρομπότ είναι ότι τώρα μπορούμε να ελέγξουμε 8 συνδυασμούς συμπεριφοράς του οχήματος μας σε σύγκριση με τους 4 συνδυασμούς που προκύπτουν αν χρησιμοποιήσουμε 2 αισθητήρες αφής. Τώρα έχουμε τη δυνατότητα να κατευθύνουμε και προς τα πίσω.

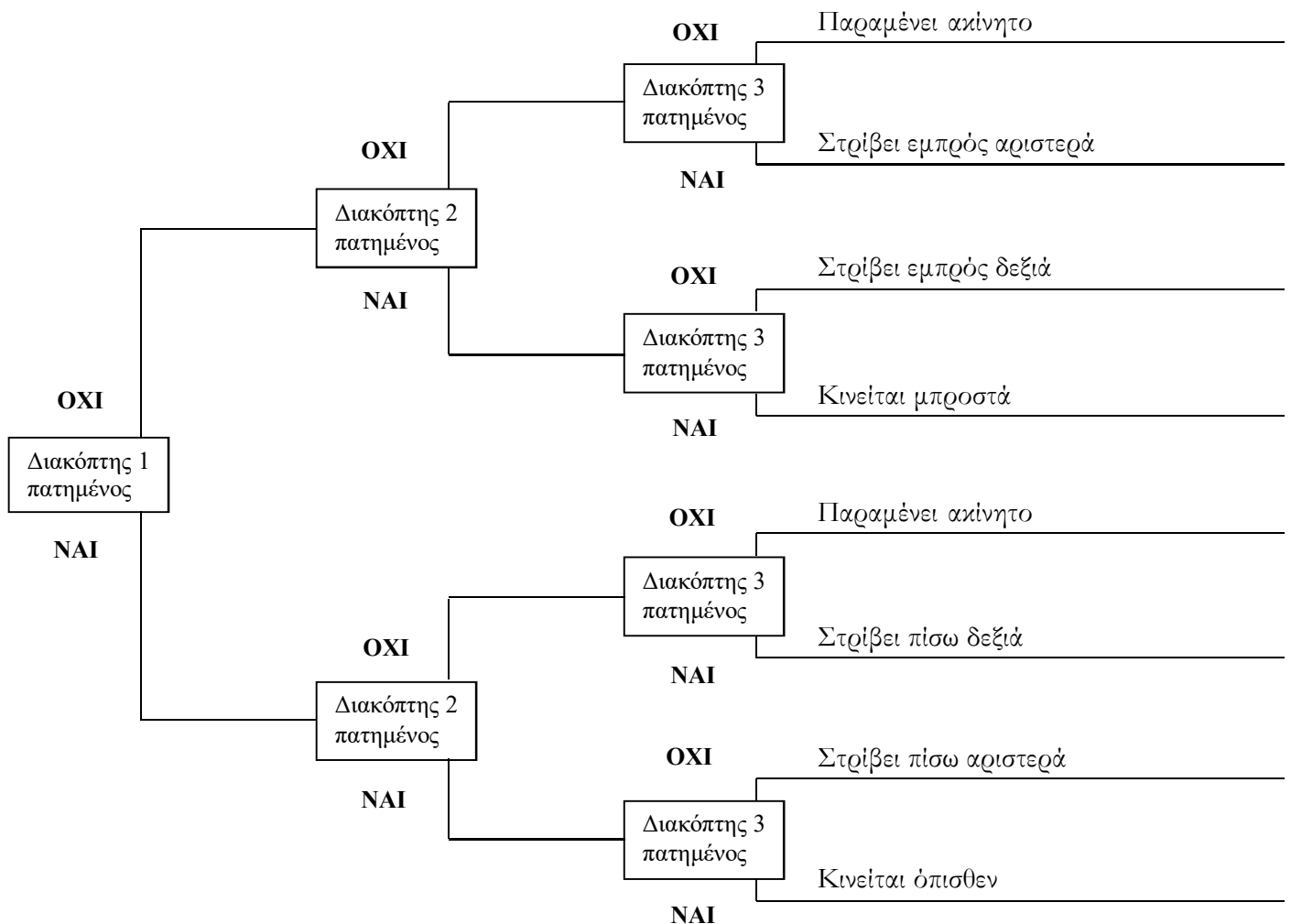
Έτσι μπορούν να προκύψουν αρκετά εναλλακτικά σενάρια.

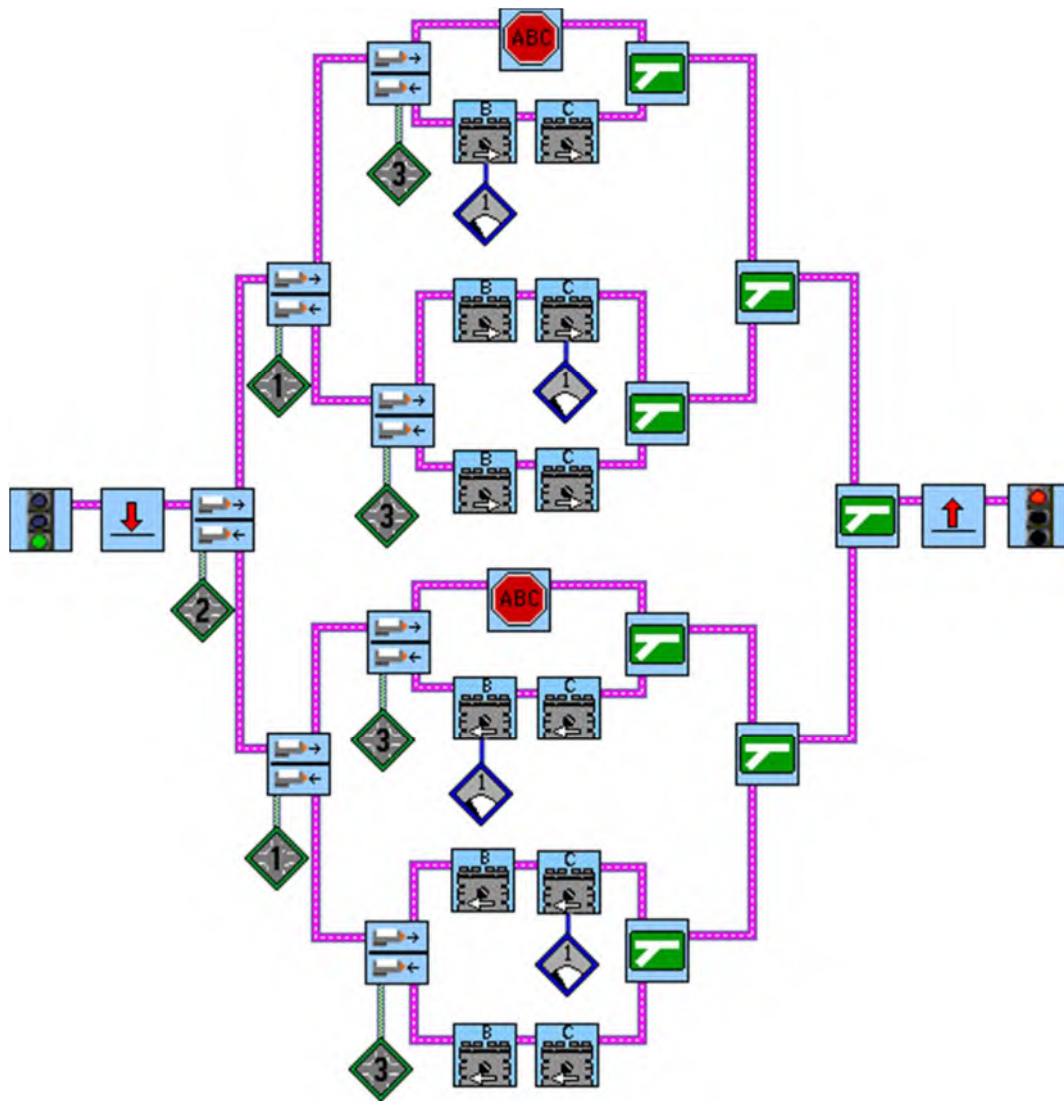
Σενάρια Ρομποτικής με το Mindstorms NXT

Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

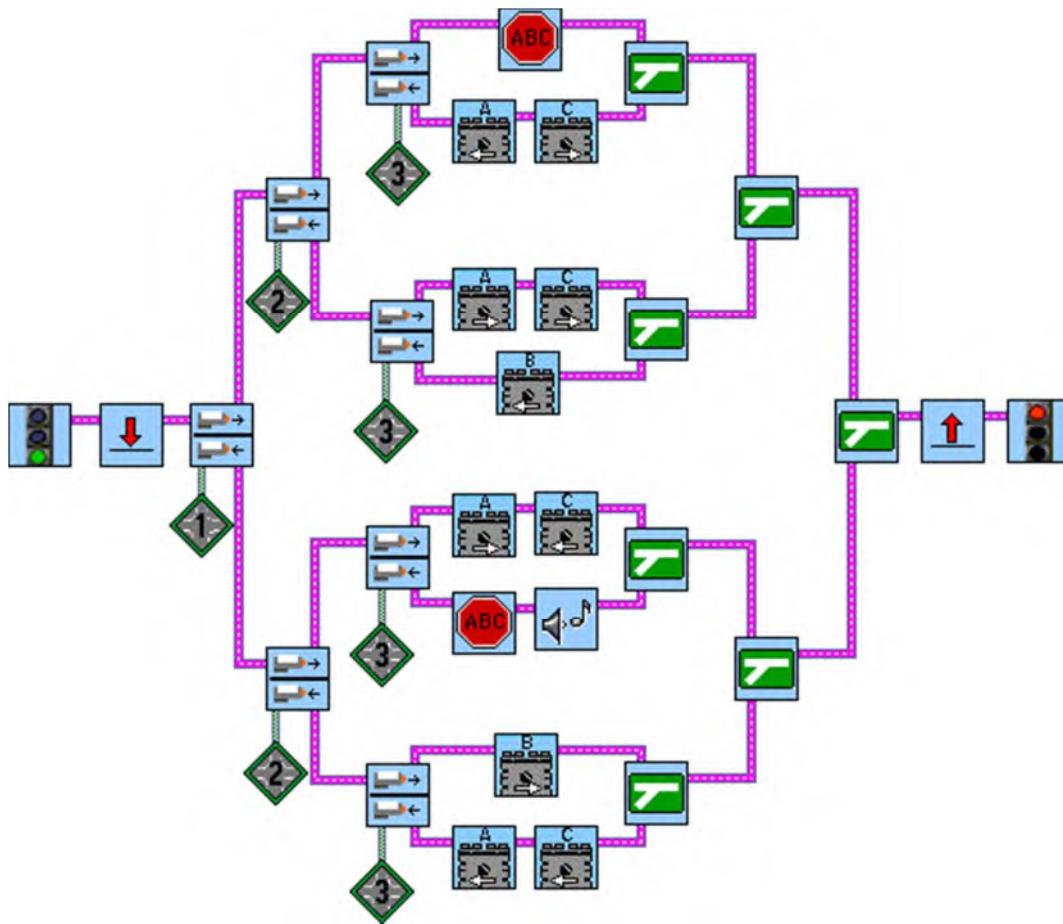
Διακόπτης 1 πατημένος	Διακόπτης 2 πατημένος	Διακόπτης 3 πατημένος	Συμπεριφορά κίνησης
OXI	OXI	OXI	Παραμένει ακίνητο
OXI	OXI	NAI	Στρίβει εμπρός αριστερά
OXI	NAI	OXI	Στρίβει εμπρός δεξιά
OXI	NAI	NAI	Κινείται μπροστά
NAI	OXI	OXI	Παραμένει ακίνητο
NAI	OXI	NAI	Στρίβει πίσω δεξιά
NAI	NAI	OXI	Στρίβει πίσω αριστερά
NAI	NAI	NAI	Κινείται όπισθεν

Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.





Από τη στιγμή που με τρεις αισθητήρες αφής προκύπτουν 8 διαφορετικού συνδυασμοί κίνησης μπορούμε να ελέγξουμε και έναν τρίτο κινητήρα. Ένα πιθανό σενάριο υλοποιείται με το παρακάτω πρόγραμμα :



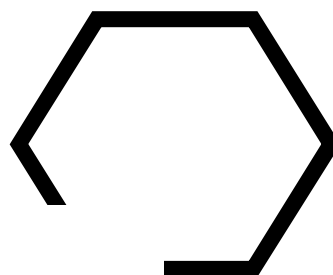
Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τον έλεγχο της κίνησης θα τον βρείτε στο σύνδεσμο: [Έλεγχος κίνησης](#)

Γ. Έξοδος από κυκλικό σχήμα

Θα επιδιώξουμε να προγραμματίσουμε το ρομπότ μας, ώστε να είναι σε θέση να βρει την έξοδο από ένα κλειστό χώρο, σχεδιασμένο στο πάτωμα με μια μαύρη γραμμή και να βγει από αυτό.

Δραστηριότητα 9η : έξοδος από κυκλικό σχήμα

Σχεδιάζουμε ένα κλειστό σχήμα με μαύρο χρώμα σε μία λευκή επιφάνεια. Στο σχήμα αυτό αφήνουμε μία έξοδο τουλάχιστον 30 εκατοστά.



Στο μπροστινό μέρος από το όχημα ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας φωτός. Τον αισθητήρα φωτός αυτόν, τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Προετοιμασία

Ένας αισθητήρας φωτός, δε μπορεί να αναγνωρίσει διαφορετικά χρώματα ή αποχρώσεις του μαύρου. Αυτό που μας δίνει είναι το ποσοστό (0-100%) του φωτός που ανακλάται από μία επιφάνεια.

Όταν έχουμε να κάνουμε με έναν αισθητήρα φωτός πριν προχωρήσουμε στη δημιουργία ενός προγράμματος, πρέπει να δούμε πως αντιλαμβάνεται ο αισθητήρας φωτός τις διαφορετικές επιφάνειες που θα συναντήσει στο περιβάλλον που θα δραστηριοποιηθεί. Το γεγονός αυτό επηρεάζεται από τη χρωματική απόχρωση που έχει μία επιφάνεια αλλά και από τις συνθήκες φωτισμού που επικρατούν στο περιβάλλοντα χώρο. Για να βρούμε τις τιμές που θα αναγνωρίσει ο αισθητήρα φωτός τον τοποθετούμε πάνω από τις διαφορετικές επιφάνειες που μας ενδιαφέρουν και εκτελούμε μία σειρά μετρήσεων.

Στην περίπτωση μας έχουμε μία μαύρη γραμμή την οποία θέλουμε να διακρίνουμε από την υπόλοιπη λευκή επιφάνεια εκτελούμε μία μέτρηση της έντασης του φωτός που αντανακλάται για τη μαύρη γραμμή και μία για τη λευκή επιφάνεια. Στη συνέχεια υπολογίζουμε τη μέση τιμή των δύο αυτών μετρήσεων την οποία και χρησιμοποιούμε ως τιμή κατωφλίου για να διακρίνει το ρομπότ μας τις δύο διαφορετικές επιφάνειες.

Για παράδειγμα αν το μαύρο διαβαστεί ως 30% και το λευκό ως 60%

$$30 + 60 = 90$$

$$90/2 = 45$$

τότε το 45, θα είναι η τιμή που θα χρησιμοποιήσουμε στη συνθήκη για την εντολή με αισθητήρα φωτός.

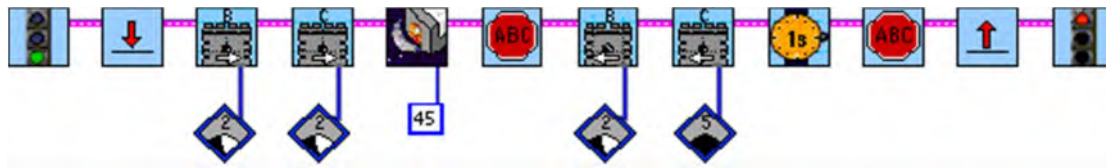
Οδηγίες

Στα προγράμματα στα οποία ένα όχημα-ρομπότ πρέπει να ξεχωρίσει δύο επιφάνειες διαφορετικής απόχρωσης, χρησιμοποιούμε τον αισθητήρα φωτός με ενεργοποιημένη την εκπομπή της υπέρυθρης ακτινοβολίας. Επίσης για να μπορέσει ένα όχημα ρομπότ να αναγνωρίσει σωστά τις επιφάνειες θα πρέπει ο αισθητήρας φωτός να βρίσκεται σε μικρή απόσταση από την επιφάνεια του δαπέδου και το όχημα να κινείται με μικρή ταχύτητα.

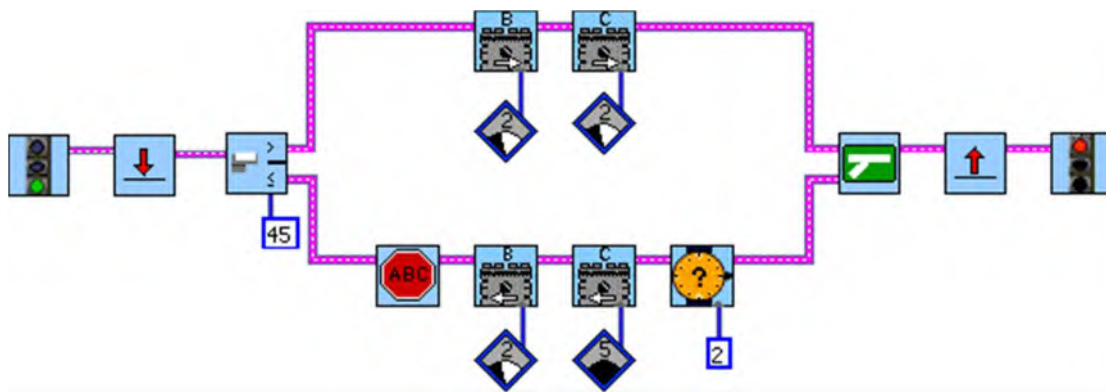
Ανάλυση προβλήματος :

Το όχημα - ρομπότ μας κινείται μπροστά σε ευθύγραμμη πορεία. Με αυτό τον τρόπο κάποια στιγμή θα συναντήσει τη μαύρη γραμμή.

α) Λαμβάνοντας υπόψη, ότι το ρομπότ μας διαθέτει αισθητήρα φωτός στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι έχει συναντήσει μία μαύρη γραμμή στην πορεία του. Μόλις συναντήσει τη μαύρη γραμμή θα κινηθεί λίγο προς τα πίσω, στρίβοντας ελαφρώς. Στη συνέχεια θα επαναλάβει τα βήματα αυτά μέχρι να βρει την έξοδο.



β. Το ίδιο πρόβλημα με διακλάδωση φωτός :

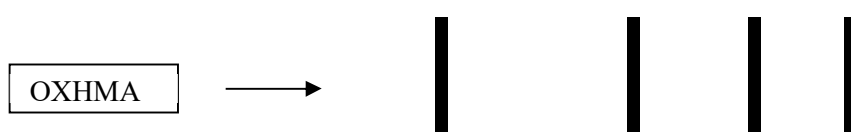


Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για την έξοδο από σπηλιά θα τον βρείτε στο σύνδεσμο: [Έξοδος από σπηλιά](#)

Δ. Μέτρηση μαύρων γραμμών

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να μετρήσει τις μαύρες γραμμές που θα συναντήσει στην πορεία του.

Σχεδιάζουμε μία σειρά από μαύρες γραμμές με μαύρο χρώμα σε μία λευκή επιφάνεια. Οι γραμμές αυτές βρίσκονται μπροστά από την πορεία του οχήματος μας και είναι παράλληλες μεταξύ τους.



Δραστηριότητα 10η : μέτρημα μαύρων γραμμών

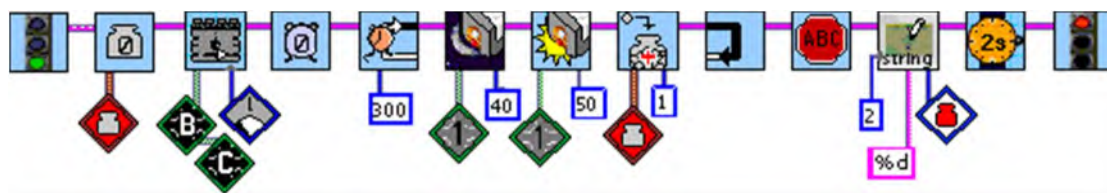
Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας φωτός.

Τον αισθητήρα φωτός αυτόν, τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Ανάλυση προβλήματος :

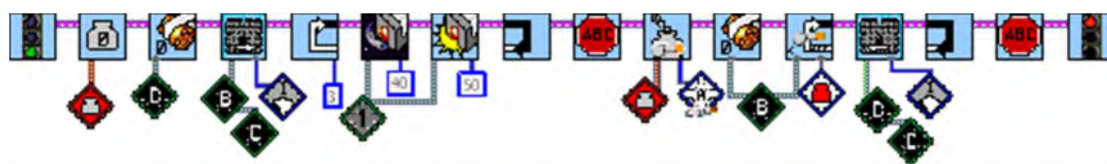
Το όχημα - ρομπότ μας κινείται μπροστά σε ευθύγραμμη πορεία για κάποιο χρονικό διάστημα. Στην πορεία θα συναντήσει έναν αριθμό από μαύρες γραμμές. Για να μπορέσει να μετρήσει τον αριθμό των γραμμών αυτών χρησιμοποιούμε έναν μετρητή.

Μόλις συναντήσει μία μαύρη γραμμή αυξάνει το μετρητή αυτόν κατά μία μονάδα. Η δεύτερη εντολή του αισθητήρα φωτός για αναμονή για φως είναι απαραίτητη για είμαστε σίγουροι ότι το όχημα έχει διασχίσει μία μαύρη γραμμή ώστε να μην ξαναμετρήσει κάποια χοντρή γραμμή πάνω από μία φορά.



Δραστηριότητα 11η : μπροστά πάνω από 3 μαύρες γραμμές και επιστροφή

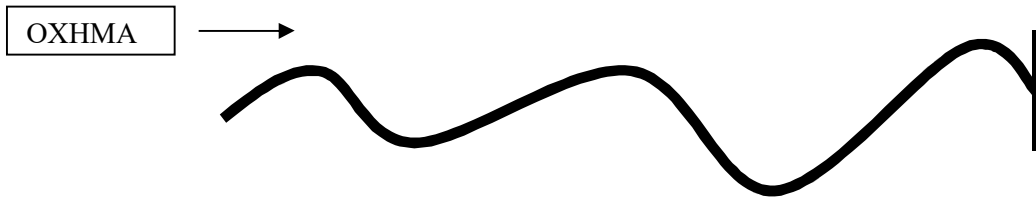
Στο παρακάτω πρόγραμμα το όχημα ρομπότ το οποίο είναι εφοδιασμένο με βραχίονα με έναν αισθητήρα φωτός, προχωράει προς τα μπροστά μέχρι να διασχίσει 3 μαύρες γραμμές και στη συνέχεια επιστρέφει ξανά πίσω στην αρχική του θέση.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για τη μέτρηση γραμμών θα τον βρείτε στο σύνδεσμο: [Μέτρηση γραμμών](#)

E. Ακολουθώντας τη γραμμή (Line follower)

Θα επιδιώξουμε να προγραμματίσουμε το ρομπότ μας, ώστε να είναι σε θέση να προχωράει κατά μήκος μίας μαύρης καμπύλης γραμμής, σχεδιασμένης στο πάτωμα.



Πως μπορούμε να κατορθώσουμε ένα όχημα-ρομπότ με δύο κινητήρες να ακολουθεί μία μαύρη γραμμή χρησιμοποιώντας έναν αισθητήρα φωτός.

Στο μπροστινό μέρος από το όχημα-ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας φωτός.

Τον αισθητήρα φωτός αυτόν, τον συνδέουμε στη θύρα εισόδου 1 του NXT.

Δραστηριότητα 12η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι

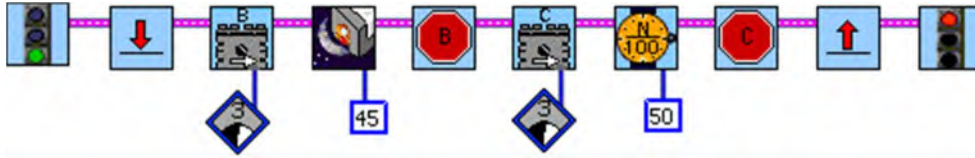
Ανάλυση προβλήματος :

Λαμβάνοντας υπόψη ότι το ρομπότ μας διαθέτει αισθητήρα φωτός στο μπροστινό του μέρος, ' ' είναι σε θέση να καταλάβει ' ' ότι στην πορεία του έχει συναντήσει τη μία μαύρη γραμμή. Θεωρήστε ότι υπάρχει μία αόρατη σε εσάς γραμμή χαραγμένη στο πάτωμα, την οποία αντιλαμβάνεστε μόνο τη στιγμή που θα τη συναντήσετε. (πχ μία γραμμή από αόρατη μελάνη την οποία δεν βλέπετε, αλλά μόλις πατήσετε πάνω της ειδοποιείστε από ένα σφύριγμα.) Αρχικά είστε τοποθετημένοι δίπλα στη γραμμή αυτή. Σας ενημερώνουν επίσης, ότι η γραμμή βρίσκεται στα **δεξιά** σας.

Τοποθετούμε το όχημα στην **αριστερή** πλευρά της γραμμής.

Ξεκινάμε κινώντας μόνο τον έναν κινητήρα ώστε να στρίψει προς την κατεύθυνση που βρίσκεται η γραμμή.

Μόλις τη συναντήσει (ο αισθητήρας φωτός ανιχνεύσει σκοτάδι) του δίνουμε εντολή να απομακρυνθεί από τη γραμμή κινώντας μόνο τον άλλο κινητήρα και μετά να ξαναεπιστρέψει προς αυτήν.



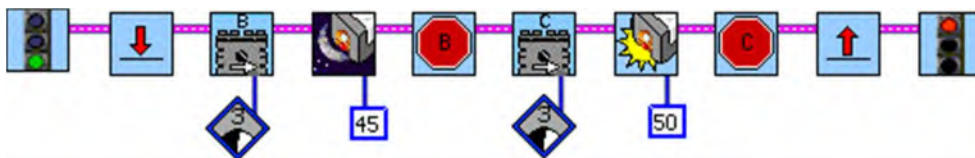
Με το πρόγραμμα αυτό προκύπτει μία σειρά προβλημάτων:

Αν ο χρόνος όπου το ρομπότ απομακρύνεται από τη μαύρη γραμμή είναι αρκετά μεγάλος υπάρχει το ενδεχόμενο στην περίπτωση που αυτή στρίβει απότομα μακριά από το ρομπότ μας, να απομακρυνθεί το ρομπότ μας μακριά από αυτήν και τη χάσει εντελώς. Όταν μειώνουμε το χρόνο, στον οποίο το ρομπότ μας απομακρύνεται από τη μαύρη γραμμή, τότε το ρομπότ μας κινείται πιο ομαλά ακολουθώντας τη γραμμή. Από την άλλη όμως αν μειώσουμε πάρα πολύ το χρόνο που κινείται το ρομπότ όταν απομακρύνεται από τη μαύρη γραμμή στην περίπτωση που αυτή στρίβει απότομα προς το ρομπότ, το ρομπότ θα συνεχίσει να βρίσκεται μέσα στη γραμμή και μετά την απομάκρυνση του από αυτήν με συνέπεια να αργοπορήσει ή και να περάσει από την άλλη μεριά με αποτέλεσμα να τη χάσει εντελώς.

Δραστηριότητα 13η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως

Μπορούμε να βελτιώσουμε το προηγούμενο πρόβλημα αν αντικαταστήσουμε την εντολή αναμονής για χρόνο με μία εντολή αναμονής για φως ώστε να απομακρύνεται από τη μαύρη γραμμή μέχρι να βρεθεί στη λευκή επιφάνεια.

Στην περίπτωση αυτήν το ρομπότ μας θα κινηθεί περισσότερο ομαλά.



Δραστηριότητα 14η : Ακολουθώντας τη γραμμή (Line follower) με 1 αισθητήρα φωτός με διακλάδωση

Το παραπάνω πρόγραμμα μπορεί να απλοποιηθεί χρησιμοποιώντας την εντολή διακλάδωσης αισθητήρα φωτός.

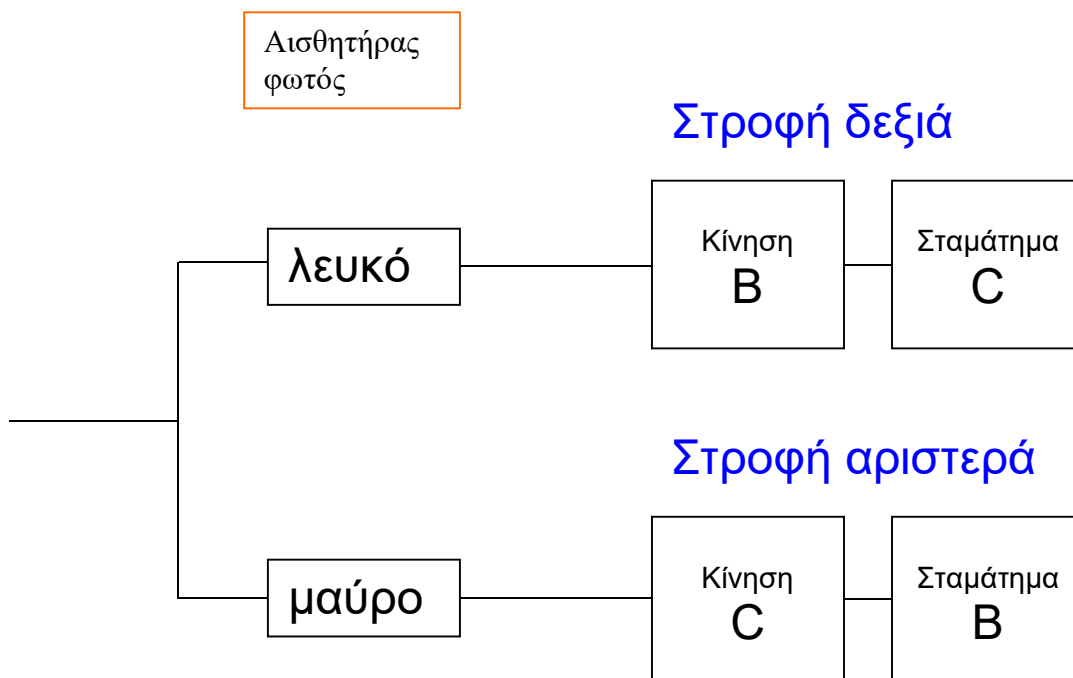
Ανάλυση προβλήματος :

Κάνουμε τη σύμβαση το όχημα να βρίσκεται **αριστερά** από τη γραμμή.

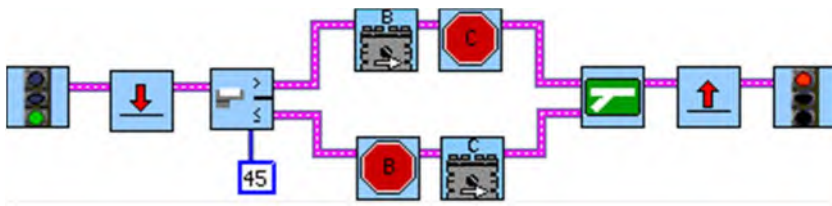
Αυτό που θέλουμε είναι να δώσουμε εντολή στο όχημα ρομπότ μας, να στρίβει δεξιά (κίνηση κινητήρα Β, σταμάτημα κινητήρα C) όταν ο αισθητήρας φωτός ανιχνεύει φωτεινό χρώμα και να στρίβει αριστερά (κίνηση κινητήρα C, σταμάτημα κινητήρα Β) όταν ο αισθητήρας φωτός ανιχνεύει σκοτεινό χρώμα.

Το ρομπότ διαβάζει την τιμή από τον αισθητήρα φωτός στη θύρα 1 και τη συγκρίνει με την αριθμητική τιμή 45 που έχουμε ορίσει ως κατώφλι. Αν η τιμή είναι πάνω από το κατώφλι, τότε ο αισθητήρας φωτός βρίσκεται έξω από τη μαύρη γραμμή. Το πρόγραμμα κινεί τον αριστερό κινητήρα και σταματάει το δεξιό κινητήρα, στρίβοντας το ρομπότ μας δεξιά προς το αριστερό σύνορο της μαύρης γραμμής. Ομοίως αν η τιμή είναι κάτω από το κατώφλι το ρομπότ στρίβει αριστερά και απομακρύνεται από το αριστερό σύνορο της μαύρης γραμμής. Ο κύκλος αυτός επαναλαμβάνεται συνεχώς μέχρι να σταματήσουμε το πρόγραμμα.

Το παρακάτω λογικό διάγραμμα περιγράφει την επιθυμητή διαδικασία:

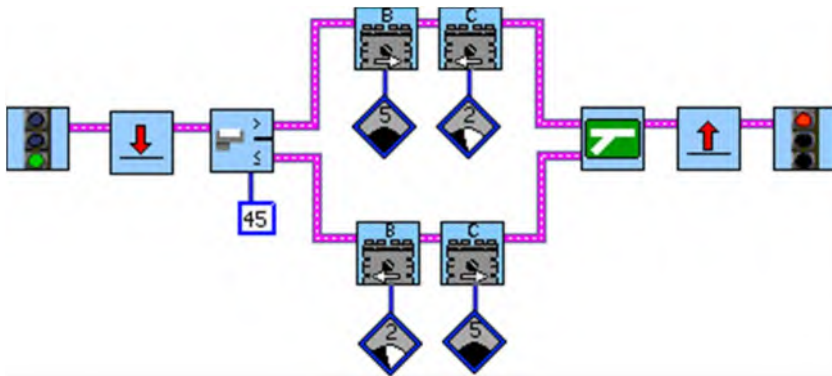


Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



β. Άλλες υλοποιήσεις

Για πιο καλό αποτέλεσμα αντί να σταματήσουμε τον ένα κινητήρα το βάζουμε να κινείται προς την αντίθετη κατεύθυνση με επίπεδο ισχύος 2. (υλοποίηση wiggler)



Πως μπορούμε να κατορθώσουμε ένα όχημα-ρομπότ με δύο κινητήρες να ακολουθεί μία μαύρη γραμμή χρησιμοποιώντας έναν αισθητήρα φωτός.

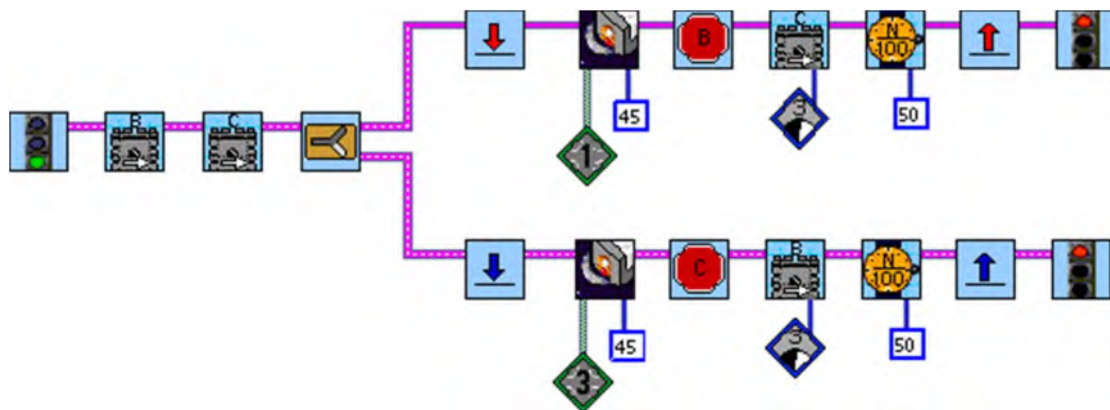
Για καλύτερο αποτέλεσμα μπορούμε να χρησιμοποιήσουμε δύο αισθητήρες φωτός τοποθετημένους έτσι ώστε να βρίσκονται δεξιά και αριστερά από τη γραμμή.

Αυτή τη φορά στο μπροστινό μέρος από το όχημα ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένοι δύο αισθητήρες φωτός. Οι αισθητήρες φωτός είναι τοποθετημένοι ο ένας στην αριστερή και ο άλλος στη δεξιά μεριά του προφυλακτήρα.

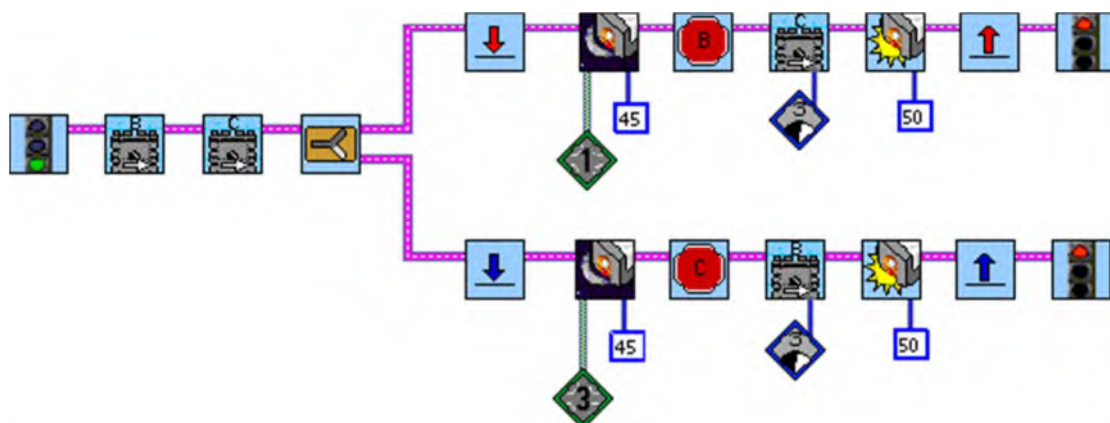
Τον αριστερό αισθητήρα φωτός τον συνδέουμε στη θύρα εισόδου 1 ενώ το δεξιό στη θύρα εισόδου 3 του NXT.

Τοποθετούμε το όχημα - ρομπότ πάνω στη μαύρη γραμμή, ώστε οι αισθητήρες φωτός να βρίσκονται δεξιά και αριστερά της μαύρης γραμμής.

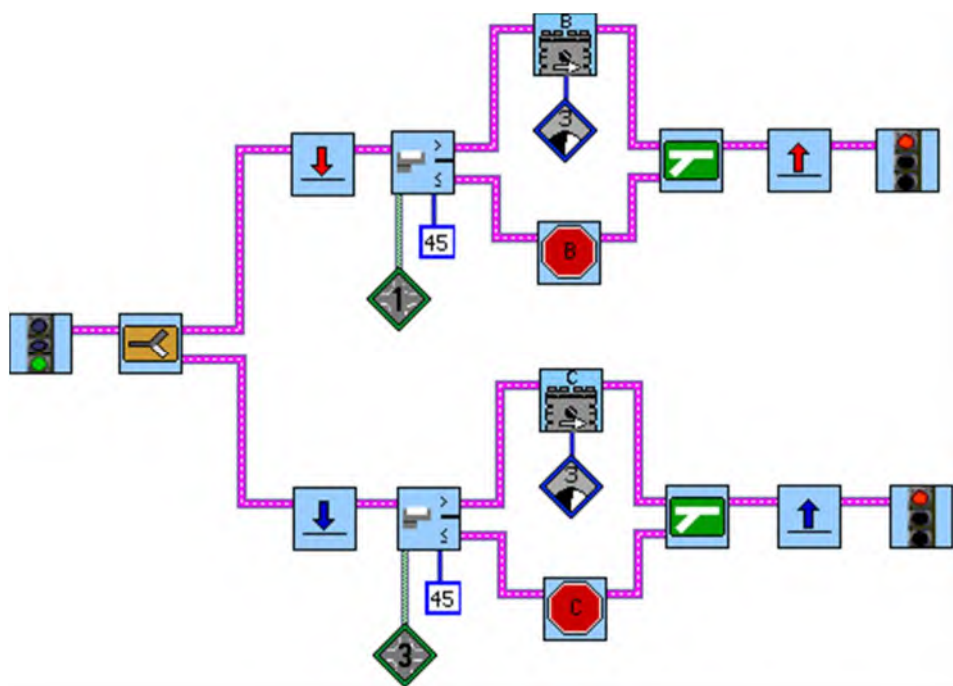
**Δραστηριότητα 15η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός
με εντολή αναμονής για σκοτάδι**



**Δραστηριότητα 16η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός
με εντολή αναμονής για σκοτάδι και εντολή αναμονής για φως**

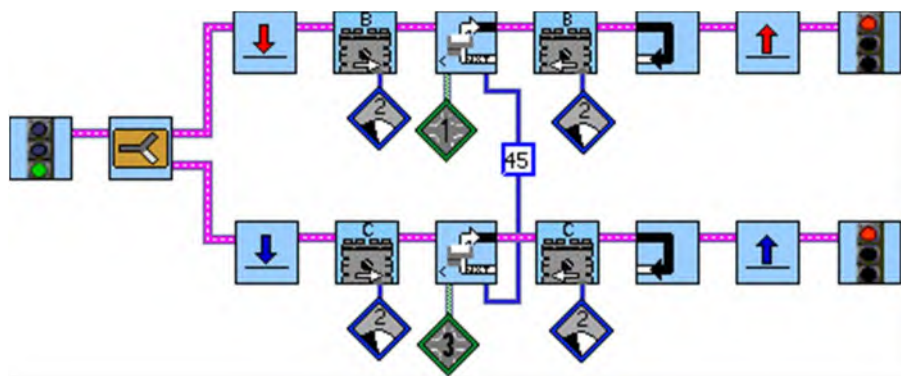


Δραστηριότητα 17η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με διακλάδωση και παράλληλες διεργασίες



Δραστηριότητα 17η : Ακολουθώντας τη γραμμή (Line follower) με 2 αισθητήρες φωτός με επανάληψη και παράλληλες διεργασίες (β' έκδοση)

Το ίδιο πρόβλημα με εντολή επανάληψης αισθητήρα φωτός :



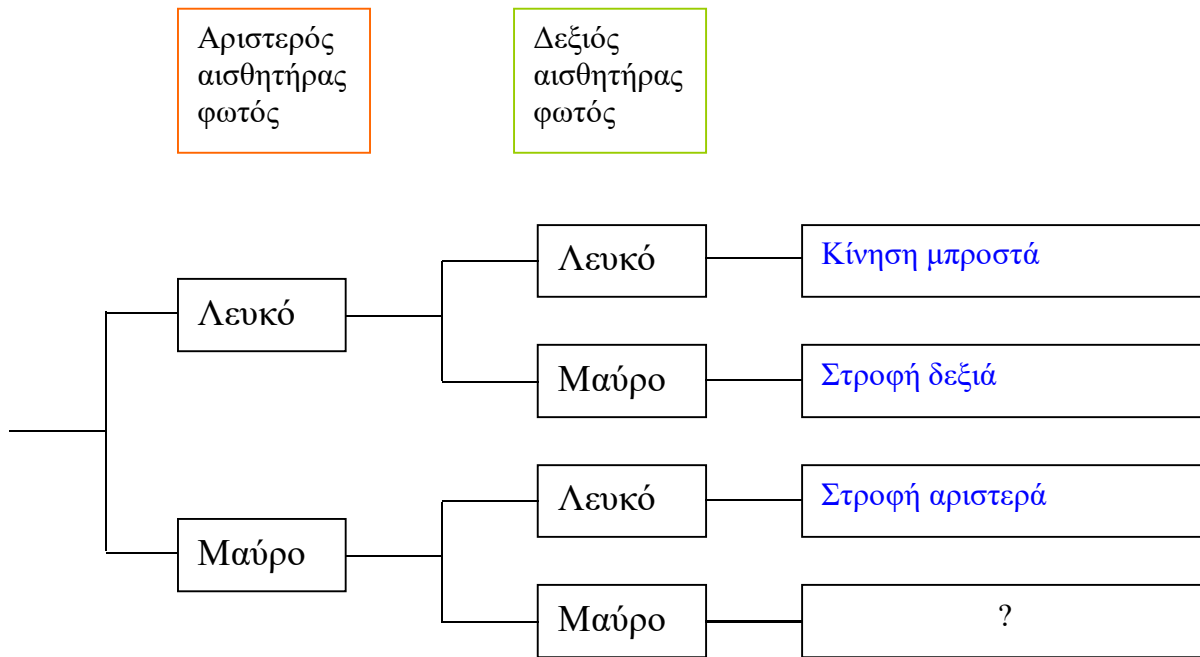
Είναι πιο βολικό για να έχουμε καλύτερο έλεγχο των συνδυασμών που προκύπτουν από δύο αισθητήρες φωτός να χρησιμοποιήσουμε μία εντολή διακλάδωσης αισθητήρα φωτός μέσα σε μία άλλη εντολή διακλάδωση φωτός.

Τοποθετούμε το όχημα - ρομπότ πάνω στη μαύρη γραμμή, ώστε οι αισθητήρες φωτός να βρίσκονται δεξιά και αριστερά της μαύρης γραμμής.

Μπορούμε επίσης να το κάνουμε να σταματάει όταν και οι δύο αισθητήρες ανιχνεύουν σκοτεινό χρώμα.

Σενάρια Ρομποτικής με το Mindstorms NXT

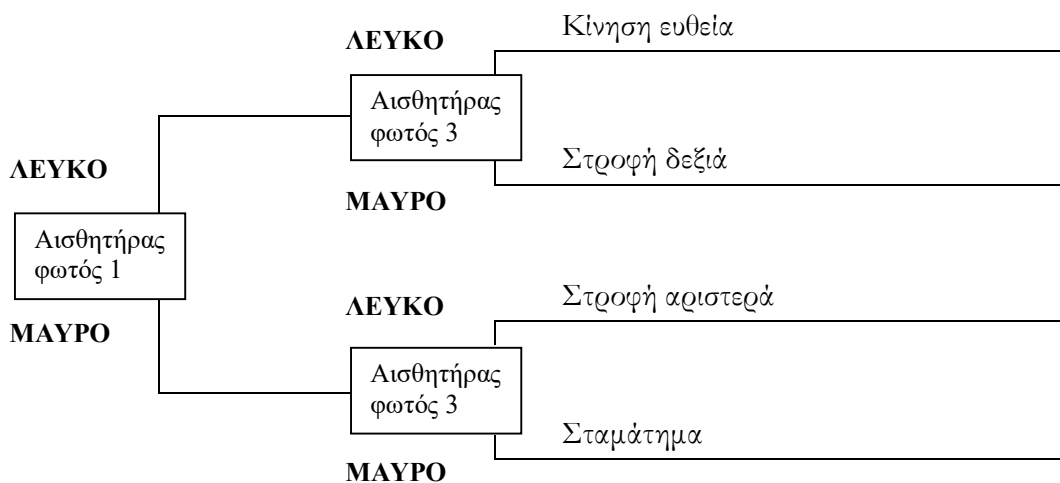
Το παρακάτω λογικό διάγραμμα περιγράφει την επιθυμητή διαδικασία:



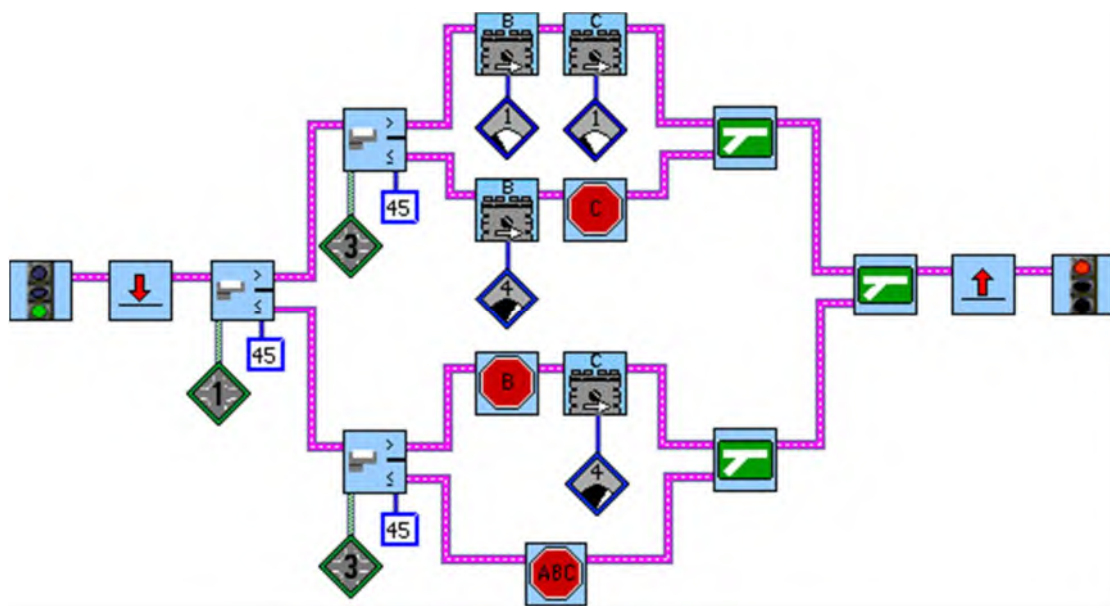
Στον παρακάτω πίνακα περιγράφεται η επιθυμητή συμπεριφορά του οχήματος μας για τους διαφορετικούς συνδυασμούς ανάμεσα στους διακόπτες αφής.

Αισθητήρας φωτός 1	Αισθητήρας φωτός 3	Συμπεριφορά κίνησης
ΛΕΥΚΟ	ΛΕΥΚΟ	Κίνηση ευθεία
ΛΕΥΚΟ	ΜΑΥΡΟ	Στροφή δεξιά
ΜΑΥΡΟ	ΛΕΥΚΟ	Στροφή αριστερά
ΜΑΥΡΟ	ΜΑΥΡΟ	Σταμάτημα

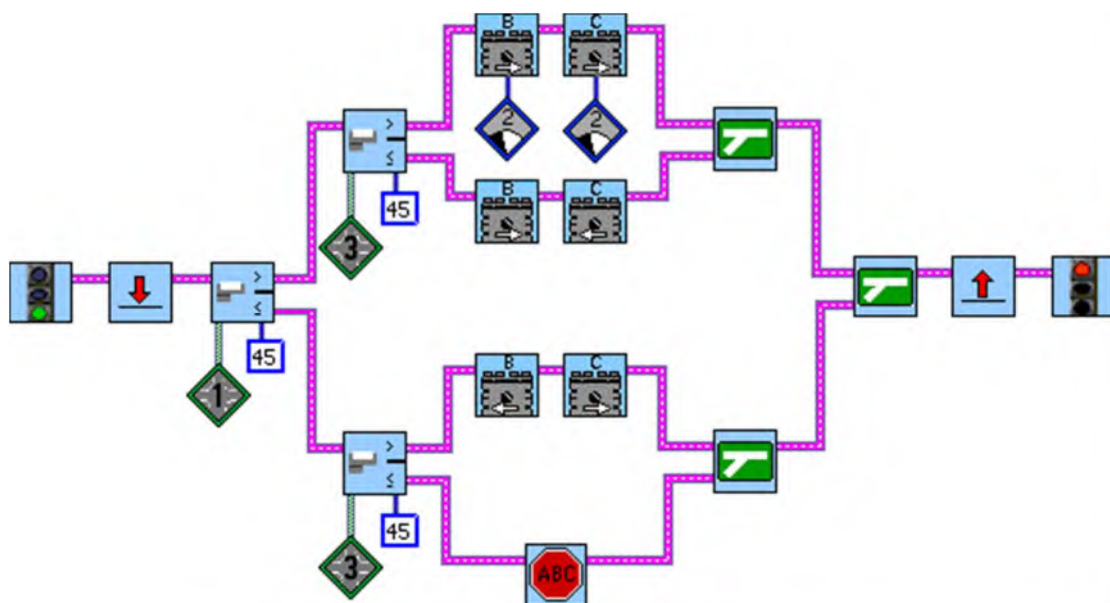
Από τον πίνακα αυτόν προκύπτει το παρακάτω λογικό διάγραμμα ροής.



Το διάγραμμα αυτό αντιστοιχεί στο παρακάτω πρόγραμμα :



β. και για καλύτερο αποτέλεσμα (ώστε να μπορέσει το ρομπότ μας να ακολουθήσει γραμμές με αρκετά απότομες στροφές) μπορούμε να βάλουμε το ρομπότ μας να στρίβει επιτόπου.



Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για να ακολουθήσει το ρομπότ μας μία γραμμή θα τον βρείτε στο σύνδεσμο : [Ακολουθώντας τη γραμμή](#)

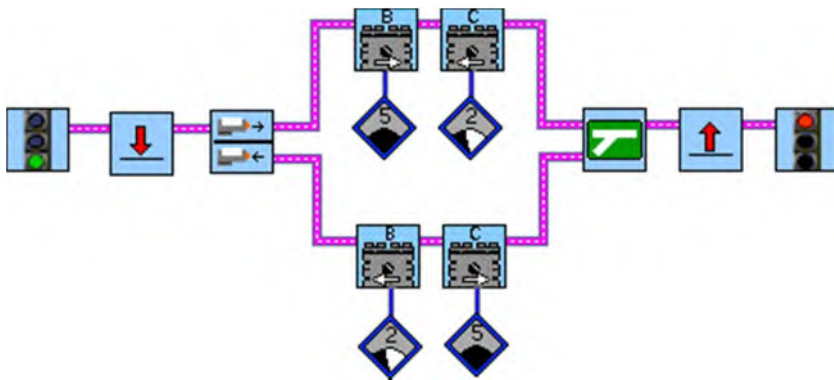
Ζ. Ακολουθώντας τον τοίχο

Θα επιδιώξουμε να προγραμματίσουμε το όχημα-ρομπότ μας ώστε να είναι σε θέση να ακολουθεί μία πορεία παράλληλη με έναν τοίχο.

Δραστηριότητα 20η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα αφής

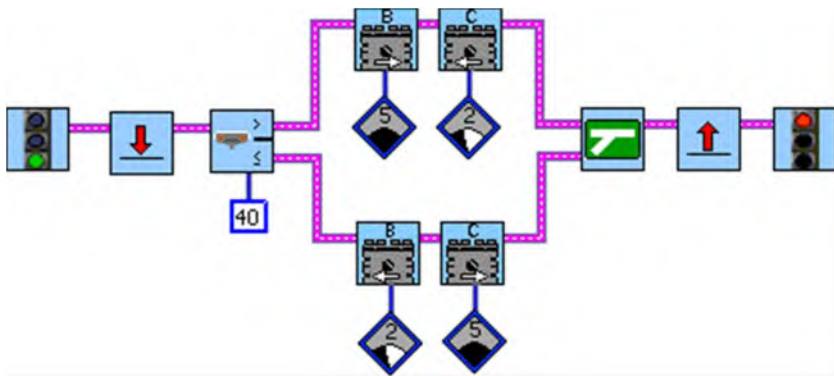
Στο μπροστινό μέρος από το όχημα ρομπότ μας εφαρμόζουμε έναν κατάλληλα προσαρμοσμένο προφυλακτήρα στον οποίο βρίσκεται τοποθετημένος ένας αισθητήρας αφής.

Τον αισθητήρα αφής αυτόν τον συνδέουμε στη θύρα εισόδου 1 του NXT.



Δραστηριότητα 21η : Ακολουθώντας τον τοίχο (Wall follower) με αισθητήρα υπερήχων

Στην περίπτωση που αντί για τον αισθητήρα αφής χρησιμοποιήσουμε έναν αισθητήρα υπερήχων τα πράγματα γίνονται πιο απλά, αφού τώρα το όχημα μας δεν χρειάζεται πλέον να πέσει πάνω στον τοίχο για να τον αντιληφθεί. Μπορεί να εντοπίσει τον τοίχο, από αρκετά μεγάλη απόσταση.

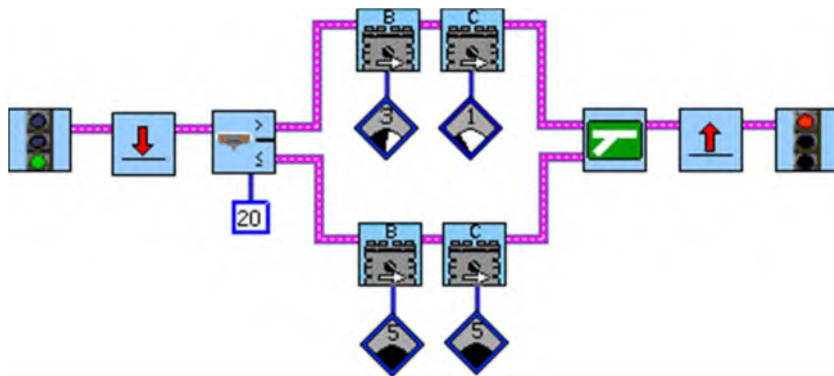


Τον κώδικα του Robolab v2.9 των δραστηριοτήτων για να ακολουθήσει το ρομπότ μας έναν τοίχο θα τον βρείτε στο σύνδεσμο : [Ακολουθώντας τον τοίχο](#)

Η. Άλλα προγράμματα

Δραστηριότητα 22η : φύλαξη με αισθητήρα υπερήχων

Στο παρακάτω πρόγραμμα το όχημα-ρομπότ εποπτεύει ένα χώρο κινούμενο με μικρή ταχύτητα εκτελώντας κυκλική πορεία. Αν κάποιος εισβολέας βρεθεί στο οπτικό του πεδίο τρέχει με δύναμη προς το μέρος του.



Δραστηριότητα 23η : Σχεδιαστήριο

Με το παρακάτω πρόγραμμα μπορούμε να σχεδιάζουμε μία εικόνα στην οθόνη του NXT ελέγχοντας τη θέση του μολυβιού μας με τη βοήθεια δύο αισθητήρων περιστροφής. Όταν ο διακόπτης του αισθητήρα αφής βρίσκεται απελευθερωμένος ο χρήστης μπορεί να σχεδιάσει στην οθόνη του NXT χρησιμοποιώντας ως χειριστήρια τους αισθητήρες περιστροφής B και C. Οι τιμές από τους αισθητήρες περιστροφής πολλαπλασιάζονται με τον παράγοντα 20 για να γίνει προσαρμογή τους στην κλίμακα της οθόνης του NXT. Όταν ο διακόπτης του αισθητήρα αφής πατηθεί, η οθόνη του NXT καθαρίζεται και γίνεται αρχικοποίηση των αισθητήρων περιστροφής ώστε να ξεκινήσουμε ένα νέο σχέδιο.



Τον κώδικα του Robolab v2.9 αυτών των δραστηριοτήτων θα τον βρείτε στο σύνδεσμο :

[Άλλα προγράμματα](#)

Για περισσότερες πληροφορίες για τις εντολές του προγραμματιστικού περιβάλλοντος Robolab v2.9 ανατρέξτε στο ΠΑΡΑΡΤΗΜΑ II.

ΠΑΡΑΡΤΗΜΑ Ι

Παρουσίαση των εντολών του NXT-G



Ένα πρόγραμμα που δημιουργούμε με το λογισμικό NXT-G είναι μία σειρά από εντολές, οι οποίες καθορίζουν πως θα συμπεριφερθεί το ρομπότ μας στα συμβάντα - γεγονότα που θα συναντήσει μέσα στο περιβάλλον που δραστηριοποιείται. Αναλυτικότερα, το ρομπότ δέχεται πληροφορίες για το περιβάλλον γύρω του μέσω των αισθητήρων (*sensors*). Στη συνέχεια ανάλογα με το πρόγραμμα που έχουμε δημιουργήσει και του έχουμε δώσει να εκτελέσει (το έχουμε κατεβάσει – *download*) το ρομπότ μας "παίρνει αποφάσεις" για το πώς πρόκειται να κινηθεί. Αισθάνεται και αντιδρά περίπου όπως και εμείς.

Οι αισθητήρες εφοδιάζουν το μικροεπεξεργαστή (τον ονομάζουμε και εγκέφαλο) NXT με πληροφορίες για το κοντινό του περιβάλλον. Ένας αισθητήρας αφής (*touch sensor*) ενημερώνει το NXT για την κατάσταση του διακόπτη του (αν είναι πατημένος ή όχι), πληροφορώντας το για το αν έχει πέσει ή όχι πάνω σε ένα εμπόδιο. Ο αισθητήρας φωτός (*light sensor*) στέλνει στο NXT έναν αριθμό ανά χρονική στιγμή ο οποίος αντιστοιχεί στην ένταση του φωτός τη στιγμή εκείνη. Ο αισθητήρας περιστροφής (*rotation sensor*) στέλνει στο NXT έναν αριθμό ο οποίος αντιστοιχεί στις περιστροφές ενός άξονα. Ο αισθητήρας θερμοκρασίας (*temperature sensor*) στέλνει στο NXT έναν αριθμό ανά χρονική στιγμή ο οποίος αντιστοιχεί στη θερμοκρασία που ανιχνεύει τη στιγμή εκείνη. Ένας άλλος αισθητήρας, ο χρονομετρητής (*timer*), - ο οποίος είναι ενσωματωμένος στα κυκλώματα του NXT - μπορεί να πληροφορήσει τον NXT για το χρονικό διάστημα που έχει περάσει. Τα προγράμματα μπορούν να διαβάσουν τις τιμές που λαμβάνουν από τους αισθητήρες και έτσι να "πάρουν μια απόφαση" για τι πρέπει να κάνουν. Με αυτό τον τρόπο κατορθώνουμε το ρομπότ μας να αποκτήσει την επιθυμητή συμπεριφορά.

Ένα πρόγραμμα στο NXT-G δημιουργείται από έναν αριθμό εικονιδίων (Block) τα οποία τοποθετούμε στη σειρά πάνω σε μία συνδετική δοκό (sequence beam). Καθένα από τα εικονίδια αναπαριστά μία απλή εντολή, και οι εντολές εκτελούνται με τη σειρά με την οποία είναι τοποθετημένες, συνιστώντας μια ακολουθία εντολών, η οποία στη συνέχεια εκτελείται από το ρομπότ μας.

Ένα πρόγραμμα διαβάζεται από αριστερά προς τα δεξιά και είναι μία **ακολουθία εικονιδίων**. Με το πέρας της ακολουθίας αυτής -αφού δηλαδή το ρομπότ μας εκτελέσει και την τελευταία εντολή του προγράμματος μας- αυτόματα διακόπτεται η τροφοδοσία σε όλες τις εξόδους.

Στην πλειοψηφία των εικονιδίων αυτών αν κάνουμε κλικ πάνω στην εσοχή που διακρίνεται στο κάτω μέρος τους εμφανίζεται μία προέκταση τους η οποία είναι γνωστή με το όνομα **κατανεμητής δεδομένων (Data Hubs)**.

Ο κατανεμητής δεδομένων μας επιτρέπει να ενώσουμε με καλωδίωση (*data wires*) δύο εντολές χρησιμοποιώντας τα αντίστοιχα σημεία ένωσης του (*data plugs*). Για να ενώσουμε δύο εντολές μεταξύ τους, ξεκινάμε μία καλωδίωση από ένα σημείο ένωσης στον κατανεμητή δεδομένων της πρώτης εντολής και την τερματίζουμε σε ένα αντίστοιχο σημείο ένωσης στον κατανεμητή δεδομένων της δεύτερης εντολής. Μέσω αυτής της καλωδίωσης οι εντολές μπορούν να ανταλλάξουν πληροφορίες όπως κείμενο, αριθμούς και διάφορες άλλες τιμές.

Ένας αρχάριος χρήστης μπορεί να δημιουργήσει ένα πρόγραμμα χωρίς να χρησιμοποιήσει καθόλου τους κατανεμητές δεδομένων. Στους κατανεμητές όμως αυτούς, κρύβεται η πραγματική δυναμική του προγραμματιστικού περιβάλλοντος NXT-G, με τη βοήθεια των οποίων μπορούμε να προσδώσουμε στο ρομπότ μας πολύπλοκες συμπεριφορές

Όταν κατασκευάζουμε μία ρομποτική κατασκευή είναι βολικό να συνδέσουμε τους κινητήρες και τους αισθητήρες μας σε συγκεκριμένες θύρες όπως αυτές βρίσκονται στις αρχικές ρυθμίσεις των παραμέτρων των εντολών του προγραμματιστικού περιβάλλοντος NXT-G. Με αυτό τον τρόπο δε θα χρειαστεί να κάνουμε πολλές αλλαγές στις παραμέτρους αυτές. Πιο συγκεκριμένα στις εντολές έχει γίνει ή σύμβαση ότι :

Ο αισθητήρας αφής είναι συνδεδεμένος τη θύρα εισόδου 1.

Ο αισθητήρας ήχου είναι συνδεδεμένος στη θύρα εισόδου 2.

Ο αισθητήρας φωτός είναι συνδεδεμένος στη θύρα εισόδου 3.

Ο αισθητήρας υπερήχων είναι συνδεδεμένος στη θύρα εισόδου 4.

Ο αριστερός (κοιτώντας από μπροστά) κινητήρας είναι συνδεδεμένος στη θύρα εξόδου B.

Ο δεξιός (κοιτώντας από μπροστά) κινητήρας είναι συνδεδεμένος στη θύρα εξόδου C.

Στην περίπτωση που η διάταξη αυτή δε μας βολεύει μπορούμε οποιαδήποτε στιγμή θέλουμε να τροποποιήσουμε τις παραμέτρους αυτές μέσα από τον πίνακα ρυθμίσεων των εντολών.

Παρακάτω παρατίθεται μία σύντομη περιγραφή όλων των εντολών που είναι διαθέσιμες στο προγραμματιστικό περιβάλλον NXT-G :

(Για περισσότερες πληροφορίες για κάθε μία εντολή ξεχωριστά, το λογισμικό NXT-G περιλαμβάνει μία ενσωματωμένη υπηρεσία βοήθειας. Για να την προσπελάσουμε κάνουμε κλικ στο σύνδεσμο [More help>>](#) στο κάτω δεξιό μέρος της εφαρμογής μας ή εναλλακτικά επιλέγουμε το *context and index...* από το μενού *help*.)

Στους πίνακες ρυθμίσεων που παραθέτονται παρακάτω μπορούμε να διακρίνουμε τις εξ'ορισμού ρυθμίσεις κάθε εντολής, τις οποίες βέβαια και μπορούμε να τροποποιήσουμε.



Βασική Παλέτα (Common Palette)



Μπλοκ μετακίνησης
Move Block



Με αυτήν την εντολή μπορούμε να πούμε στο ρομπότ μας να κινηθεί μπροστά ή πίσω σε ευθεία γραμμή ή να στρίψει ακολουθώντας μία καμπύλη γραμμή. Επίσης χρησιμοποιώντας την ίδια εντολή μπορούμε να πούμε στο ρομπότ μας να σταματήσει να κινείται.

Συνήθως αυτή την εντολή τη χρησιμοποιούμε στις περιπτώσεις που έχουμε κατασκευάσει ένα ρομπότ το οποίο είναι εφοδιασμένο με δύο τουλάχιστον κινητήρες, υπεύθυνους για την κίνηση του.

Σημείωση : Η εντολή αυτή εμπεριέχεται και ένα είδος 'ευφυΐας' αφού παρακολουθεί συνεχώς την περιστροφή των κινητήρων με τη βοήθεια των ενσωματωμένων αισθητήρων περιστροφής και μέσω ενός αλγορίθμου επιτυγχάνεται ο συγχρονισμός των κινητήρων και η μετακίνηση του ρομπότ μας με μεγάλη ακρίβεια.

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων μπορούμε να ρυθμίσουμε τις θύρες εξόδου (*Port*) στις οποίες θέλουμε να ελέγξουμε τους κινητήρες, την κατεύθυνση της κίνησης (*Direction*) ή εναλλακτικά να πούμε στο ρομπότ μας να σταματήσει να κινείται, και την ισχύ (*Power*) με την οποία θα τροφοδοτηθούν οι κινητήρες (κλίμακα 0-100). Όσο μεγαλύτερη ισχύ καθορίσουμε, τόσο πιο γρήγορα θα κινείται το ρομπότ μας.

Μπορούμε επίσης να καθορίσουμε αν η κίνηση θα γίνει σε ευθεία γραμμή ή σε καμπύλη (*Steering*), οπότε το ρομπότ μας θα εκτελέσει μία στροφή.

Τέλος μπορούμε να καθορίσουμε τη διάρκεια που θα έχει η κίνηση (*Duration*) η οποία επηρεάζει την απόσταση που θα διανύσει το ρομπότ μας :

Αν επιλέξουμε το ρομπότ μας να κινηθεί για ένα συγκεκριμένο διάστημα (σε πλήρεις περιστροφές "*Rotations*", μοίρες περιστροφής "*Degrees*" ή χρονικό διάστημα "*Seconds*") το πρόγραμμα θα περιμένει να ολοκληρωθεί αυτή η κίνηση, προτού προχωρήσει στην εκτέλεση της επόμενης εντολής. Στην περίπτωση αυτή μπορούμε να καθορίσουμε και αν με το πέρας της κίνησης οι κινητήρες θα φρενάρουν αστραπιαία "*Brake*" ή θα συνεχίσουν να ολισθαίνουν "*Coast*" μέχρι να σταματήσουν, λόγω τριβής, ενώ θα έχει αρχίσει η εκτέλεση της επόμενης εντολής (*Next Action*).

Σενάρια Ρομποτικής με το Mindstorms NXT

Αν επιλέξουμε το ρομπότ μας να κινείται συνεχώς *''Unlimited''* το πρόγραμμα θα θέσει σε κίνηση το ρομπότ και θα περάσει αμέσως στην εκτέλεση της επόμενης εντολής. Στην περίπτωση αυτή χρησιμοποιούμε κάποια άλλη εντολή (συνήθως αναμονής από έναν αισθητήρα) αργότερα μέσα στο πρόγραμμα μας, για να σταματήσουμε το ρομπότ μας.

Επεξήγηση συμβόλων

1. Τα γράμματα πάνω δεξιά δηλώνουν τις θύρες εξόδου στις οποίες θέλουμε να ελέγξουμε τους κινητήρες.
2. Η εικόνα κάτω αριστερά εμφανίζει την κατεύθυνση προς την οποία θα κινηθεί το ρομπότ.
3. Η εικόνα κάτω στη μέση εμφανίζει την ισχύ των κινητήρων.
4. Η εικόνα κάτω δεξιά εμφανίζει τη ρύθμιση της διάρκειας της κίνησης : αν έχει οριστεί σε συνεχόμενη *''Unlimited''* [σύμβολο ∞], μοίρες *''Degrees''*, περιστροφές *''Rotations''* ή δευτερόλεπτα *''Seconds''*.



Αν επιλέξουμε ως ενέργεια τη διακοπή της κίνησης των κινητήρων τότε στο κάτω μέρος της εντολής αυτής θα εμφανιστεί μόνο η εικόνα της διακοπής, στα δεξιά.



**Μπλοκ Εγγραφής/Αναπαραγωγής
Record/Play Block**



Με αυτή την εντολή μπορούμε να πούμε στον επεξεργαστή NXT να απομνημονεύσει μία σειρά ενεργειών κίνησης, τις οποίες καθορίζουμε εμείς οι ίδιοι μετακινώντας το ρομπότ μας χειροκίνητα. Σε δεύτερη φάση μπορούμε να ζητήσουμε από το ρομπότ μας να αναπαραγάγει αυτούσια αυτή τη σειρά των ενεργειών.

Σημείωση : Οι ενέργειες κίνησης που εγγράφουμε ενδέχεται να μην είναι ακριβώς ίδιες με τις ενέργειες που αναπαράγονται από το ρομπότ μας

Πίνακας Ρυθμίσεων



Για να κάνουμε εγγραφή μιας σειράς ενεργειών, πρώτα ορίζουμε ένα όνομα (*Name*) για αυτήν και στη συνέχεια καθορίζουμε τη χρονική διάρκεια της εγγραφής (*Time*), αφού υπολογίσουμε το χρονικό διάστημα που θα διαρκέσει αυτή η σειρά των ενεργειών μας. Το χρονικό διάστημα μπορούμε να το καθορίσουμε είτε μετά από εκτίμηση, είτε αφού πρώτα εκτελέσουμε τη σειρά των ενεργειών που θέλουμε να εγγράψουμε, και μετρήσουμε το χρόνο που χρειάζεται για να ολοκληρωθούν αυτές.

Σενάρια Ρομποτικής με το Mindstorms NXT

Επίσης καθορίζουμε και τις θύρες εξόδου (*Recording*) στις οποίες θα γίνει η καταγραφή των κινήσεων. Στη συνέχεια, αφού πρώτα έχουμε μεταφορτώσει το πρόγραμμα με το μπλοκ εγγραφής, όταν είμαστε έτοιμοι τρέχουμε το πρόγραμμα και εκτελούμε τις ενέργειες κίνησης που επιθυμούμε, μετακινώντας το ρομπότ μας με τα χέρια μας. Όταν ολοκληρωθεί το χρονικό διάστημα το οποίο είχαμε καθορίσει, οι ενέργειες που εμείς επιβάλλαμε χειρωνακτικά, σώζονται στον επεξεργαστή NXT, ως αρχείο με το όνομα που είχαμε καθορίσει.

Για να γίνει η αναπαραγωγή των ενεργειών κίνησης αλλάζουμε στον πίνακα ρυθμίσεων της εντολής εγγραφής / αναπαραγωγής, την αντίστοιχη ρύθμιση της ενέργειας (*Action*) από "*Record*" σε "*Play*", πληκτρολογούμε το όνομα που είχαμε δώσει κατά την εγγραφή της σειράς των ενεργειών κίνησης, και αφού μεταφορτώσουμε το νέο πρόγραμμα μας, το τρέχουμε. Η εντολή αυτή στο νέο πρόγραμμα μας, θα έχει ως αποτέλεσμα να γίνει αναπαραγωγή της σειράς των ενεργειών κίνησης πλέον χωρίς τη δική μας παρέμβαση, όσο πιο πιστά είναι αυτό δυνατόν.



Επεξήγηση συμβόλων

1. Η εικόνα κάτω δεξιά εμφανίζει την κατάσταση της εντολή : αν γίνεται εγγραφή ή αναπαραγωγή ενεργειών.



**Μπλοκ ήχου
Sound Block**



Με αυτήν την εντολή μπορούμε να πούμε στο ρομπότ μας να παίξει ένα συγκεκριμένο αρχείο ήχου ή έναν απλό τόνο.

Για να συνθέσουμε μια μελωδία πρέπει να τοποθετήσουμε αρκετές εντολές ήχου στη σειρά με κάθε μία από αυτές να παίζει ένα διαφορετικό τόνο.

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων μπορούμε να ρυθμίσουμε την ενέργεια (*Action*) για να καθορίσουμε αν θα γίνει αναπαραγωγή ενός αρχείου ήχου "*Sound File*" ή ενός τόνου, "*Tone*" το οποίο και επιλέγουμε από τη δεξιά πλευρά του πίνακα.

καθώς και την ένταση (*Volume*) της αναπαραγωγής.

Αν ενεργοποιήσουμε την παράμετρο της επανάληψης "*Repeat*" της ρύθμισης της λειτουργίας (*Function*), ο τόνος ή το αρχείο ήχου που επιλέξαμε θα αναπαράγεται ξανά και ξανά.

Σενάρια Ρομποτικής με το Mindstorms NXT

Αν ενεργοποιήσουμε την παράμετρο της αναμονής για ολοκλήρωση *“Wait for completion”* της ρύθμισης της αναμονής (*Wait*) το πρόγραμμα θα περάσει στην εκτέλεση της επόμενης εντολής, αφού πρώτα ολοκληρωθεί ο τόνος ή το αρχείο ήχου που επιλέξαμε να αναπαραχθεί. Με αυτήν την επιλογή ανενεργή ο τόνος ή το αρχείο ήχου θα συνεχίσει να παίζει ενώ θα εκτελείται ταυτόχρονα η επόμενη εντολή του προγράμματος μας.

Για να σταματήσουμε την αναπαραγωγή των ήχων που συνεχίζουν να παίζουν επιλέγουμε στη ρύθμιση του ελέγχου (*Control*) την παράμετρο της διακοπής *“Stop”*.

Επεξήγηση συμβόλων

1. Η εικόνα κάτω αριστερά εμφανίζει την είδος του ήχου : αρχείο ήχου *“Sound File”* ή απλός τόνος *“Tone”*.
2. Η εικόνα κάτω στη μέση εμφανίζει την κατάσταση : έναρξη ή διακοπή της αναπαραγωγής του ήχου.
3. Η εικόνα κάτω δεξιά εμφανίζει τη ρύθμιση της έντασης (*Volume*).



**Μπλοκ εμφάνισης
Display Block**



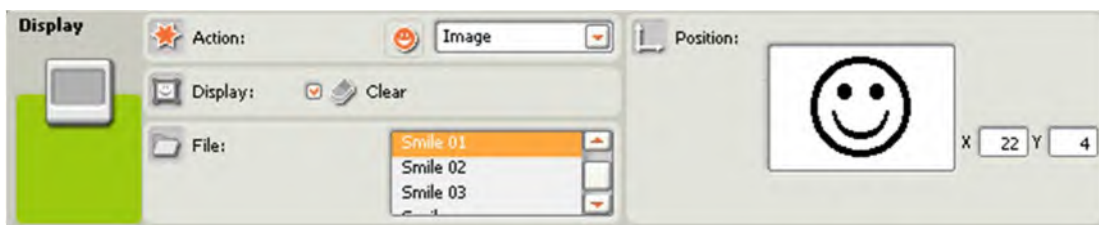
Με αυτήν την εντολή μπορούμε να πούμε στο NXT του ρομπότ μας να εμφανίσει στην οθόνη του μία εικόνα, κάποιο κείμενο ή ένα συγκεκριμένο σχέδιο το οποίο σχεδιάζουμε εμείς.

Μπορούμε να δημιουργήσουμε πολύπλοκα σχέδια τοποθετώντας αρκετά μπλοκ εμφάνισης στη σειρά με κάθε ένα από αυτά να προσθέτει κάποιο γράφημα, κείμενο ή συγκεκριμένο σχήμα.

Η οθόνη του NXT έχει ανάλυση 100 x 64 εικονοστοιχεία.

Σημείωση : Για να προλάβουμε να δούμε την εικόνα που επιλέξαμε στην οθόνη του NXT θα πρέπει η εντολή αυτή να ακολουθείται από μία εντολή αναμονής για χρόνο, αλλιώς η εικόνα αυτή θα εμφανιστεί για κλάσμα του δευτερολέπτου.

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων επιλέγουμε τον τύπο του γραφήματος από τη ρύθμιση της ενέργειας (*Action*) ανάμεσα στην εικόνα *“Image”*, κάποιο κείμενο *“Text”* ή σχέδιο *“Design”* το οποίο έχουμε

Σενάρια Ρομποτικής με το Mindstorms NXT

σχεδιάσει εμείς. Αν επιλέξουμε την παράμετρο της επαναφοράς *''Reset''* θα εμφανιστεί το αρχικό εξ' ορισμού κείμενο *''LEGO MINDSTORMS''*

Αν θέλουμε να σβήσουμε το περιεχόμενο της οθόνης του NXT προτού εμφανιστεί το νέο μας γράφημα ενεργοποιούμε στη ρύθμιση της εμφάνισης (*Display*) την παράμετρο του καθαρισμού *''Clear''*.

Επεξήγηση συμβόλων

1. Η εικόνα κάτω δεξιά εμφανίζει τον τύπο του γραφήματος : εικόνα κείμενο *''text''* ή σχέδιο *''drawing''* ή αν θα γίνει επαναφορά *''reset''* σχεδίου στην οθόνη.



του *''image''*,
αρχικού



Μπλοκ Αναμονής
Wait Block



Με αυτές τις εντολές μπορούμε να πούμε στο ρομπότ μας να περιμένει μέχρι να συμβεί κάποιο συγκεκριμένο γεγονός (να ικανοποιηθεί κάποια συγκεκριμένη συνθήκη).

Οι εντολές αυτές δίνουν τη δυνατότητα στο ρομπότ μας να αντιλαμβάνεται το περιβάλλον στο οποίο δραστηριοποιείται ως προς μία συγκεκριμένη κατάσταση προτού συνεχίσει με την επόμενη εντολή.

Μέσω της ρύθμισης (*Control*) του πίνακα ρυθμίσεων μπορούμε να καθορίσουμε αν το ρομπότ μας θα περιμένει να περάσει κάποιο χρονικό διάστημα *''Time''* ή αν θα περιμένει μέχρι να συμβεί ένα συγκεκριμένο γεγονός σε κάποιο συγκεκριμένο αισθητήρα *''Sensor''*, τον οποίο δηλώνουμε για την περίπτωση αυτή στη ρύθμιση του αισθητήρα (*Sensor*).

Ανάλογα με τις επιλογές μας στον πίνακα ρυθμίσεων η εντολή αυτή μπορεί να πάρει τις παρακάτω μορφές :



Μπλοκ Αναμονής για χρόνο

Με αυτή την εντολή το NXT θα περιμένει μέχρι να περάσει ένα συγκεκριμένο χρονικό διάστημα.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε το χρονικό διάστημα "Until" σε δευτερόλεπτα.

Επεξήγηση συμβόλων

1. Η εικόνα κάτω στο κέντρο εμφανίζει το χρονικό διάστημα μετά από την πάροδο του οποίου η εντολή θα επιτρέψει το πρόγραμμα να προχωρήσει στην εκτέλεση της επόμενης εντολής.



Μπλοκ Αναμονής αισθητήρα αφής

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι να πατηθεί/απελευθερωθεί το κουμπί σε έναν αισθητήρα αφής (touch sensor).

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (Port) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου no 1.

Επίσης δηλώνουμε ποια ενέργεια (Action) στον αισθητήρα αφής μας ενδιαφέρει : "Pressed", απελευθέρωση "Released" ή πάτημα και απελευθέρωση "Bumped".

Σενάρια Ρομποτικής με το Mindstorms NXT

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την κατάσταση του διακόπτη του αισθητήρα αφής. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

2. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
3. Η εικόνα κάτω δεξιά εμφανίζει την κατάσταση για την οποία η εντολή θα επιτρέψει το πρόγραμμα να προχωρήσει στην εκτέλεση της επόμενης εντολής.



Μπλοκ Αναμονής αισθητήρα φωτός

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση της φωτεινότητας σε ένα αισθητήρα φωτός «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι).

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (*Port*) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου no 3.

Επίσης καθορίζουμε την τιμή της στάθμης του φωτός που θέλουμε το πρόγραμμα να περιμένει ώσπου να επιτευχθεί σέρνοντας τον αντίστοιχο σύρτη (*Until*) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Αν αφήσουμε ενεργή την παράμετρο της παραγωγής φωτός "*Generate light*" της ρύθμισης της λειτουργίας (*Function*) τότε ο αισθητήρας φωτός εκπέμπει τη δική του υπέρυθη ακτινοβολία. Διαφορετικά, αν απενεργοποιήσουμε δηλαδή, τη συγκεκριμένη παράμετρο τότε ο αισθητήρας θα ανιχνεύει την ένταση του φωτός από το περιβάλλον.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα στάθμη του φωτός. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω στη μέση δηλώνει αν είναι ενεργοποιημένη ή παραγωγή υπέρυθρης ακτινοβολίας "Generated Light" στον αισθητήρα..
3. Η εικόνα κάτω δεξιά εμφανίζει το σημείο κατωφλίου στο οποίο η εντολή θα επιτρέψει το πρόγραμμα να προχωρήσει στην εκτέλεση της επόμενης εντολής.



Μπλοκ Αναμονής αισθητήρα ήχου

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση του ήχου σε ένα αισθητήρα ήχου «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι).

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (*Port*) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου νο 2.

Επίσης καθορίζουμε την τιμή της στάθμης του ήχου που θέλουμε το πρόγραμμα να περιμένει ώπου να επιτευχθεί σέρνοντας τον αντίστοιχο σύρτη (*Until*) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα στάθμη του ήχου. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά εμφανίζει το σημείο κατωφλίου στο οποίο η εντολή θα επιτρέψει το πρόγραμμα να προχωρήσει στην εκτέλεση της επόμενης εντολής.





Μπλοκ Αναμονής αισθητήρα υπέρηχων

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η απόσταση από κάποιο εμπόδιο που θα ανιχνευτεί από ένα αισθητήρα υπέρηχων «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι).

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (*Port*) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου νο 4.

Επίσης καθορίζουμε την τιμή της απόστασης που θέλουμε το πρόγραμμα να περιμένει ώσπου να επιτευχθεί σέρνοντας τον αντίστοιχο σύρτη (*Until*) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα απόσταση από κάποιο εμπόδιο. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά εμφανίζει το σημείο κατωφλίου στο οποίο η εντολή θα επιτρέψει το πρόγραμμα να προχωρήσει στην εκτέλεση της επόμενης εντολής.





Μπλοκ Αναμονής χρονομετρητή

Ο μικροεπεξεργαστής NXT διαθέτει 3 εσωτερικούς χρονομετρητές. Όταν ένα πρόγραμμα ξεκινάει, αυτοί οι χρονομετρητές ξεκινάνε αυτόματα να μετράνε το χρόνο που περνάει.

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι ο τιμή του χρόνου σε ένα χρονομετρητή «να ξεπεράσει / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι).

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων ορίζουμε πρώτα το χρονομετρητή (*Timer*) που θέλουμε να παρατηρήσουμε και καθορίζουμε το χρόνο που θέλουμε το πρόγραμμα να περιμένει ώσπου να επιτευχτεί πληκτρολογώντας τον στο αντίστοιχο πεδίο (*Until*) Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου).

Για να μηδενίσουμε έναν χρονομετρητή επιλέγουμε στη ρύθμιση της ενέργειας (*Action*) αντί για την παράμετρο *Read* τη *Reset*.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή το χρόνο που έχει διανυθεί. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Η εικόνα κάτω στη μέση δηλώνει την ενέργεια : παρακολούθηση των χρόνου (read) ή μηδενισμός του χρονομετρητή (reset).
2. Η εικόνα κάτω δεξιά εμφανίζει τον αριθμό του χρονομετρητή που παρακολουθείται.



Μπλοκ Αναμονής αισθητήρα περιστροφής

Σε κάθε κινητήρα του πακέτου NXT υπάρχει και ένας ενσωματωμένος αισθητήρας περιστροφής. Αντίστοιχα ο μικρουπολογιστής NXT διαθέτει 3 εσωτερικούς μετρητές για την περιστροφή κάθε κινητήρα στις θύρες εξόδου του.

Σενάρια Ρομποτικής με το Mindstorms NXT

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η περιστροφή που πραγματοποιείται σε έναν αισθητήρα περιστροφής «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι).

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων επιλέγουμε πρώτα τον κινητήρα που θέλουμε να ελέγξουμε (*Port*).

Επίσης καθορίζουμε τη φορά περιστροφής που μας ενδιαφέρει και την τιμή της περιστροφής που θέλουμε το πρόγραμμα να περιμένει ώσπου να επιτευχθεί, πληκτρολογώντας τη στο αντίστοιχο πεδίο (*Until*). Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου).

Για να μηδενίσουμε το μετρητή για την περιστροφή επιλέγουμε στη ρύθμιση της ενέργειας (*Action*) αντί για την παράμετρο "*Read*" τη "*Reset*". Έτσι αν θέλουμε να ξεκινήσουμε από την αρχή μία μέτρηση για την περιστροφή που θα πραγματοποιήσει κάποιος κινητήρας, στο πρόγραμμα που θα δημιουργήσουμε θα πρέπει να τοποθετήσουμε πιο μπροστά μία εντολή αισθητήρα περιστροφής με επιλεγμένη την παράμετρο "*Reset*", για να μηδενίσουμε τον αντίστοιχο μετρητή περιστροφής.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα περιστροφή που έχει πραγματοποιηθεί. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Το γράμμα πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω στη μέση δηλώνει την κατεύθυνση προς στην οποία θα πραγματοποιηθεί η μέτρηση.
3. Η εικόνα κάτω δεξιά δηλώνει την ενέργεια : παρακολούθηση των περιστροφών "*Read*" ή μηδενισμός του μετρητή "*Reset*".



Μπλοκ Αναμονής για λήψη μηνύματος

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι να λάβουμε ένα συγκεκριμένο μήνυμα ασύρματα.

Για να μπορέσει να γίνει αποστολή και λήψη μηνυμάτων θα πρέπει οι μικροϋπολογιστές NXT που θα συμμετέχουν στην ανταλλαγή μηνυμάτων να έχουν ρυθμιστεί κατάλληλα για ασύρματη επικοινωνία οπότε και να τους έχει αποδοθεί ένας αριθμός σύνδεσης (connection number).

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων δηλώνουμε το είδος του μηνύματος (*Message*) (κείμενο *Text*, αριθμό *Number* ή λογικό αριθμό *Logic*) που περιμένουμε να αποσταλεί στο NXT μας, και να καθορίζουμε το ακριβές περιεχόμενο του προς σύγκριση. Στη συνέχεια επιλέγουμε τον αριθμό της θυρίδας ταχυδρομείου (*Mailbox*) όπου το μήνυμα θα αποθηκευτεί στο NXT μας.

Το πρόγραμμά μας θα περιμένει μέχρι να έρθει ένα εισερχόμενο μήνυμα στο οποίο υπάρχει το συγκεκριμένο περιεχόμενο που έχουμε καθορίσει.

Επεξήγηση συμβόλων

1. Η εικόνα κάτω στη μέση δηλώνει το είδος του μηνύματος : κείμενο *Text*, αριθμός *Number* ή λογικός αριθμός *Logic*.
2. Η εικόνα κάτω δεξιά εμφανίζει την τον αριθμό της θυρίδας.



Μπλοκ Αναμονής πλήκτρου NXT

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι να πατηθεί/απελευθερωθεί κάποιο από τα πλήκτρα που βρίσκονται πάνω στο NXT.

Την εντολή αυτή μπορούμε να την τοποθετήσουμε μετά από μία εντολή εμφάνισης έτσι ώστε να μπορεί κάποιος να διαβάσει τα στοιχεία που εμφανίζονται στην οθόνη χωρίς να υπάρχει χρονικό περιθώριο, οπότε και μόνο τότε να πατήσει κάποιο πλήκτρο του NXT για να συνεχιστεί η εκτέλεση του προγράμματος.

Πίνακας Ρυθμίσεων



Σενάρια Ρομποτικής με το Mindstorms NXT

Στον πίνακα ρυθμίσεων δηλώνουμε το πλήκτρο (*Button*) που επιθυμούμε (ανάμεσα στα enter δεξί και αριστερό πλήκτρο).

Επίσης δηλώνουμε ποια ενέργεια (*Action*) στον αισθητήρα αφής μας ενδιαφέρει : *''Pressed''*, απελευθέρωση *''Released''* ή πάτημα και απελευθέρωση *''Bumped''*.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την κατάσταση του επιλεγμένου πλήκτρου. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Η εικόνα στη μέση δεξιά εμφανίζει το πλήκτρο που παρακολουθείται : πορτοκαλί (enter) ή γκρι αριστερό (left) ή δεξιό βελάκι (right)
2. Η εικόνα κάτω δεξιά δηλώνει την ενέργεια που έχει επιλεγεί : πάτημα διακόπτη, απελευθέρωση ή πάτημα και απελευθέρωση.



**Μπλοκ Επανάληψης
Loop Block**



Με αυτήν την εντολή μπορούμε να πούμε στο ρομπότ μας να επαναλάβει μία σειρά εντολών. Μπορούμε να καθορίσουμε η επανάληψη αυτή να συμβεί για κάποιο συγκεκριμένο αριθμό φορών ή για πάντα (αέναςος βρόχος) είτε να τερματιστεί όταν ικανοποιηθεί κάποια συνθήκη : πχ να ικανοποιηθεί μία λογική συνθήκη, να λήξει ένα συγκεκριμένο χρονικό διάστημα, να συμβεί κάποιο γεγονός σε κάποιο συγκεκριμένο αισθητήρα.

Μέσω της ρύθμισης (*Control*) του πίνακα ρυθμίσεων μπορούμε να καθορίσουμε αν το ρομπότ μας θα επαναλάβει μία σειρά εντολών συνέχεια *''Forever''*, για κάποιο χρονικό διάστημα *''Time''*, για έναν ορισμένο αριθμό φορών *''Count''*, μέχρι να επαληθευτεί μία λογική πράξη *''Logic''* ή μέχρι να συμβεί ένα συγκεκριμένο γεγονός σε κάποιο συγκεκριμένο αισθητήρα *''Sensor''*, τον οποίο δηλώνουμε για την περίπτωση αυτή στη ρύθμιση του αισθητήρα (*Sensor*).

Ανάλογα με τις επιλογές μας στον πίνακα ρυθμίσεων η εντολή αυτή μπορεί να πάρει τις παρακάτω μορφές :



Επανάληψη παντοτινή

Με αυτήν την εντολή, το τμήμα του προγράμματος εντός του βρόχου επανάληψης, θα επαναλαμβάνεται συνεχώς *''Forever''* μέχρι εμείς να διακόψουμε την εκτέλεση του προγράμματος.

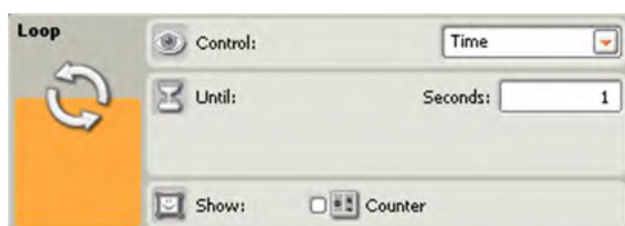
Πίνακας Ρυθμίσεων



Επανάληψη για χρόνο

Με αυτήν την εντολή, το τμήμα του προγράμματος εντός του βρόχου επανάληψης, θα επαναλαμβάνεται μέχρι να περάσει ο χρόνος σε δευτερόλεπτα *''Seconds''* που έχουμε καθορίσει στο πίνακα ρύθμισης της.

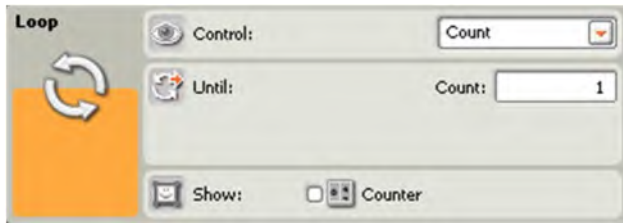
Πίνακας Ρυθμίσεων



Επανάληψη για n φορές

Με αυτήν την εντολή, το τμήμα του προγράμματος εντός του βρόχου επανάληψης, θα επαναληφθεί τόσες φορές *''Count''*, όσες έχουμε καθορίσει στον πίνακα ρύθμισης της.

Πίνακας Ρυθμίσεων



Επανάληψη μέχρι να

Με αυτήν την εντολή, το τμήμα του προγράμματος εντός του βρόχου επανάληψης, θα επαναλαμβάνεται μέχρι να επαληθευτεί μία λογική συνθήκη *''Logic''* που έχουμε καθορίσει στον πίνακα ρύθμισης της.

Στην περίπτωση αυτή εμφανίζεται ένα σημείο εισόδου λογικού αριθμού. Θα πρέπει να καλωδιώσουμε το σημείο αυτό, από κάποιο αντίστοιχο σημείο εξόδου του κατανεμητή δεδομένων (data hub) μιας άλλης εντολής, για να εισάγουμε ένα λογικό όρισμα προς σύγκριση.

Πίνακας Ρυθμίσεων



Επανάληψη μέχρι να από αισθητήρα

Με αυτήν την εντολή, το τμήμα του προγράμματος εντός του βρόχου επανάληψης, θα επαναλαμβάνεται μέχρι να συμβεί ένα συγκεκριμένο γεγονός σε ένα συγκεκριμένο αισθητήρα *''Sensor''* που έχουμε καθορίσει στον πίνακα ρύθμισης της.

Σενάρια Ρομποτικής με το Mindstorms NXT

Με μια εντολή επανάληψης με αισθητήρα αφής, για παράδειγμα, μπορούμε να καθορίσουμε το ρομπότ μας να επαναλαμβάνει μία σειρά εντολών μέχρι να πατηθεί ο διακόπτης αφής που προσδιορίσαμε.

Πίνακας Ρυθμίσεων



Σημείωση: Για τη χρήση των αισθητήρων στην εντολή της επανάληψης βλ. εντολές αναμονής.

Επεξήγηση συμβόλων

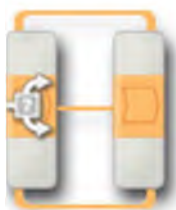
1. Η εικόνα κάτω δεξιά εμφανίζει τον αριθμό των επαναλήψεων: συνέχεια [σύμβολο ∞], μέχρι να συμβεί κάποιο γεγονός το οποίο θα ανιχνεύσει ένας αισθητήρας (sensor), μέχρι να περάσει ένα συγκεκριμένο χρονικό διάστημα (time), συγκεκριμένος αριθμός (count), ή μέχρι να ισχύσει μία συγκεκριμένη λογική πρόταση (logic).

Αν επιλέξουμε ο αριθμός των επαναλήψεων να ελέγχεται από κάποιον αισθητήρα τότε το εικονίδιο της εντολής αυτής θα επεκταθεί, συμπεριλαμβάνοντας και το σύμβολο του αντίστοιχου αισθητήρα, ενώ στον πίνακα ρυθμίσεων θα εμφανιστούν και οι ανάλογες επιλογές.

2. Αν καθορίσουμε ο έλεγχος της επανάληψης να γίνει από κάποια λογική τιμή *“Logic”* και όχι από κάποιον αισθητήρα θα εμφανιστεί ένα σημείο εισόδου μέσω του οποίου μπορούμε να στείλουμε μία τιμή ως είσοδο από κάποιο αντίστοιχο σημείο εξόδου του κατανεμητή δεδομένων (data hub) μιας άλλης εντολής.

3. να αυτή εμφανίζεται ένα σημείο εισόδου λογικού αριθμού. Θα πρέπει να καλωδιώσουμε το σημείο αυτό από κάποιο αντίστοιχο σημείο εξόδου του κατανεμητή δεδομένων (data hub) μιας άλλης εντολής, για να εισάγουμε ένα λογικό όρισμα προς σύγκριση.

4. Αν τσεκάρουμε την επιλογή εμφάνισης μετρητή *“Show Counter”*, θα εμφανιστεί ένα σημείο εξόδου μέσω του οποίου μπορούμε να στείλουμε την τιμή του αριθμού των επαναλήψεων που έχουν γίνει μέχρι τότε ως είσοδο προς κάποιο αντίστοιχο σημείο εισόδου του κατανεμητή δεδομένων (data hub) μιας άλλης εντολής.



**Μπλοκ Επιλογής
Switch Block**



Με μία εντολή επιλογής δημιουργούνται μέσα στο πρόγραμμα δύο πιθανές διαδρομές (μονοπάτια) τις οποίες μπορεί ακολουθήσει το πρόγραμμά μας. Από αυτά τα δύο μονοπάτια το πρόγραμμα θα ακολουθήσει μόνο το ένα, ανάλογα με την κατάσταση που θα έχει διαμορφωθεί εκείνη τη στιγμή, μετά

Σενάρια Ρομποτικής με το Mindstorms NXT

από την πραγματοποίηση ενός ελέγχου. Με αυτήν την εντολή το ρομπότ μας θα ακολουθήσει μία σειρά εντολών ανάμεσα από δύο πιθανές διαδρομές, ανάλογα με τη συνθήκη που θα επικρατεί τη στιγμή εκείνη.

Ανάλογα με τις επιλογές μας στον πίνακα ρυθμίσεων η εντολή αυτή μπορεί να πάρει τις παρακάτω μορφές :



Επιλογή από τιμή

Με την εντολή αυτή αν τη στιγμή που γίνει ο έλεγχος μία μεταβλητή *“Value”* βρεθεί αληθής (true), το πρόγραμμα θα ακολουθήσει τη σειρά των εντολών που βρίσκεται στην πάνω πλευρά, διαφορετικά αν η μεταβλητή βρεθεί ψευδής (false) θα ακολουθηθεί η σειρά των εντολών που βρίσκεται στην κάτω πλευρά.

Στην περίπτωση αυτή εμφανίζεται ένα σημείο εισόδου λογικού αριθμού. Θα πρέπει να καλωδιώσουμε το σημείο αυτό, από κάποιο αντίστοιχο σημείο εξόδου του κατανεμητή δεδομένων (data hub) μιας άλλης εντολής, για να εισάγουμε ένα λογικό όρισμα προς σύγκριση.

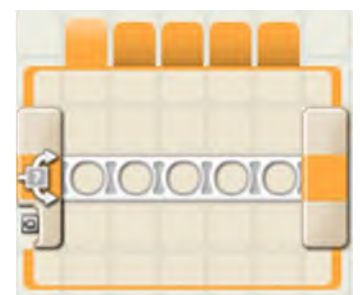
Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων δηλώνουμε τον τύπο (Type) που θέλουμε να έχει η μεταβλητή μας ανάμεσα σε λογική τιμή *“Logic”*, αριθμό *“Number”* και κείμενο *“Text”*.



Στην περίπτωση που δηλώσουμε μία μεταβλητής τύπου αριθμού ή κειμένου μπορούμε να αλλάξουμε και τις τιμές που θα δέχεται αυτή στη συνθήκη σύγκρισης (Conditions). Αν μάλιστα απενεργοποιήσουμε την επιλογή εμφάνισης *“Flat view”* τότε μπορούμε να προσθέσουμε και επιπλέον τιμές για τη συνθήκη σύγκρισης πατώντας το πλήκτρο +, κάτι που θα έχει ως συνέπεια τη δημιουργία επιπλέον πιθανών διαδρομών. Για να τοποθετήσουμε τις κατάλληλες εντολές σε κάθε ένα από τα μονοπάτια αυτά, μπορούμε να συμβουλευτούμε τα καρτελάκια στην πάνω πλευρά της εντολής αυτής, πάνω από τα οποία εμφανίζεται το όνομα κάθε διαδρομής.



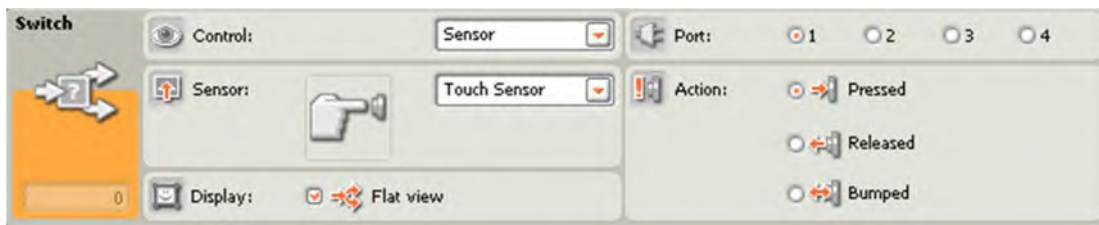


Επιλογή από αισθητήρα

Με την εντολή αυτή αν τη στιγμή που γίνει ο έλεγχος συμβεί ένα συγκεκριμένο γεγονός σε ένα συγκεκριμένο αισθητήρα *''Sensor''*, το πρόγραμμα θα ακολουθήσει τη σειρά των εντολών που βρίσκεται στην πάνω πλευρά, διαφορετικά, αν δεν ικανοποιηθεί η συνθήκη που θέσαμε για το συγκεκριμένο αισθητήρα, θα ακολουθηθεί η σειρά των εντολών που βρίσκεται στην κάτω πλευρά.

Με μια εντολή επιλογής με αισθητήρα αφής, για παράδειγμα, μπορούμε να καθορίσουμε το ρομπότ μας να εκτελέσει τη μία σειρά εντολών (αυτή που βρίσκεται στην πάνω πλευρά) στην περίπτωση που ο διακόπτης του αισθητήρα βρεθεί πατημένος τη στιγμή του ελέγχου, διαφορετικά (αν ο διακόπτης του αισθητήρα βρεθεί απελευθερωμένος τη στιγμή του ελέγχου) να εκτελέσει την άλλη σειρά των εντολών (αυτή που βρίσκεται στην κάτω πλευρά).

Πίνακας Ρυθμίσεων



Επεξήγηση συμβόλων

1. Η εικόνα στην αρχή της εντολής αυτής δηλώνει το είδος του αισθητήρα ή κάποια άλλη τιμή που ευθύνεται για τη λήψη της απόφασης ως προς το ποια από τις δύο διακλαδώσεις του προγράμματος θα ακολουθηθεί.
2. Αν καθορίσουμε ο έλεγχος για τη λήψη της απόφασης (control) να γίνει από κάποια τιμή *''Value''* και όχι από κάποιον αισθητήρα θα εμφανιστεί ένα σημείο εισόδου μέσω του οποίου μπορούμε να στείλουμε μία τιμή ως είσοδο από κάποιο αντίστοιχο σημείο εξόδου του κατανεμητή δεδομένων (data hub) μιας άλλης εντολής.
3. Αν απενεργοποιήσουμε την επιλογή εμφάνισης *''Flat view''*, θα αλλάξει η μορφή εμφάνισης της εντολής αυτής. Στο πάνω μέρος της τώρα εμφανίζονται δύο καρτελάκια με τα οποία μπορούμε να δούμε και την άλλη διακλάδωση του προγράμματος.



Πλήρης παλέτα (Complete Palette)



Μπλοκ Ενεργειών Action Blocks



Μπλοκ Κινητήρα Motor Block

Με αυτήν την εντολή μπορούμε να πούμε στο ρομπότ μας να ξεκινήσει ή να σταματήσει τον κινητήρα που ορίζεται από τη θύρα εξόδου στον πίνακα ρυθμίσεων.

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων μπορούμε να ρυθμίσουμε τη θύρα εξόδου (*Port*) στην οποία θέλουμε να ελέγξουμε τον κινητήρα, την κατεύθυνση της περιστροφής (*Direction*) ή εναλλακτικά να πούμε στον κινητήρα να σταματήσει να κινείται, και την ισχύ (*Power*) με την οποία θα τροφοδοτηθεί ο κινητήρας (κλίμακα 0-100). Όσο μεγαλύτερη ισχύ καθορίσουμε, τόσο πιο γρήγορα θα περιστρέφεται ο κινητήρας αυτός.

Τέλος μπορούμε να καθορίσουμε τη διάρκεια που θα έχει η κίνηση (*Duration*) :

Αν επιλέξουμε ο κινητήρας να περιστραφεί για ένα συγκεκριμένο διάστημα (σε πλήρεις περιστροφές "*Rotations*", μοίρες περιστροφής "*Degrees*" ή χρονικό διάστημα "*Seconds*") το πρόγραμμα θα περιμένει να ολοκληρωθεί αυτή η κίνηση, προτού προχωρήσει στην εκτέλεση της επόμενης εντολής. Στην περίπτωση αυτή μπορούμε να καθορίσουμε και αν με το πέρας της περιστροφής ο κινητήρας

Σενάρια Ρομποτικής με το Mindstorms NXT

αυτός θα φρενάρει αστραπιαία *''Brake''* ή θα συνεχίσει να ολισθαίνει *''Coast''* μέχρι να σταματήσει, λόγω τριβής, ενώ θα έχει αρχίσει η εκτέλεση της επόμενης εντολής (*Next Action*).

Αν επιλέξουμε ο κινητήρας να περιστρέφεται συνεχώς *''Unlimited''* το πρόγραμμα θα θέσει σε κίνηση τον κινητήρα και θα περάσει αμέσως στην εκτέλεση της επόμενης εντολής. Στην περίπτωση αυτή χρησιμοποιούμε κάποια άλλη εντολή (συνήθως αναμονής από έναν αισθητήρα) αργότερα μέσα στο πρόγραμμα μας, για να σταματήσουμε τον κινητήρα αυτόν.

Επεξήγηση συμβόλων

1. Το γράμμα πάνω δεξιά δηλώνει τη θύρα εξόδου στην οποία θέλουμε να ελέγξουμε τον κινητήρα.
2. Η εικόνα κάτω αριστερά εμφανίζει την κατεύθυνση προς την οποία θα κινηθεί ο κινητήρας.
3. Η εικόνα κάτω στη μέση εμφανίζει την ισχύ του κινητήρα.
4. Η εικόνα κάτω δεξιά εμφανίζει τη ρύθμιση της διάρκειας της κίνησης : αν έχει οριστεί σε συνεχόμενη *''Unlimited''* [σύμβολο ∞], μοίρες *''Degrees''*, περιστροφές *''Rotations''* ή δευτερόλεπτα *''Seconds''*.

Αν επιλέξουμε ως ενέργεια τη διακοπή της κίνησης του κινητήρα τότε στο κάτω μέρος της εντολής αυτής θα εμφανιστεί μόνο η εικόνα της διακοπής, στα δεξιά.



**Μπλοκ κινητήρα NXT
Motor* Block (old)**

Αυτή η εντολή μας επιτρέπει να ελέγξουμε τους παλιούς κινητήρες από τη σειρά NXT της Lego. Με αυτήν την εντολή μπορούμε να πούμε στο ρομπότ μας να ξεκινήσει ή να σταματήσει έναν κινητήρα.

Σημείωση : Για να συνδέσουμε ένα κινητήρα αυτού του τύπου στο NXT πρέπει να χρησιμοποιήσουμε το αντίστοιχο καλώδιο προσαρμογής.

Επεξήγηση συμβόλων

1. Το γράμμα πάνω δεξιά δηλώνει τη θύρα εξόδου στην οποία θέλουμε να ελέγξουμε τον κινητήρα.
2. Η εικόνα κάτω αριστερά εμφανίζει την κατεύθυνση προς την οποία θα κινηθεί ο κινητήρας.
3. Η εικόνα κάτω στη μέση εμφανίζει την ισχύ του κινητήρα.
4. Η εικόνα κάτω δεξιά εμφανίζει τη ρύθμιση της διάρκειας της κίνησης : αν έχει οριστεί σε συνεχόμενη *''Unlimited''* [σύμβολο ∞], μοίρες *''Degrees''*, περιστροφές *''Rotations''* ή δευτερόλεπτα *''Seconds''*.

Αν επιλέξουμε ως ενέργεια τη διακοπή της κίνησης του κινητήρα τότε στο κάτω μέρος της εντολής αυτής θα εμφανιστεί μόνο η εικόνα της διακοπής, στα δεξιά.



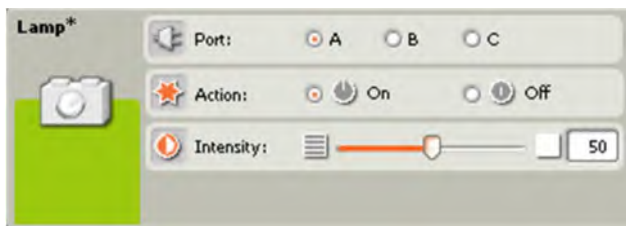


Μπλοκ λαμπτήρα NXT Lamp* Block

Αυτή η εντολή μας επιτρέπει να ελέγχουμε τους παλιούς λαμπτήρες από τη σειρά NXT της Lego. Με αυτήν την εντολή μπορούμε να πούμε στο ρομπότ μας να ανάψει ή να σβήσει ένα λαμπτήρα.

Σημείωση : Για να συνδέσουμε ένα λαμπτήρα αυτού του τύπου στο NXT πρέπει να χρησιμοποιήσουμε το αντίστοιχο καλώδιο προσαρμογής.

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων μπορούμε να ρυθμίσουμε τη θύρα εξόδου (*Port*) στην οποία θέλουμε να ελέγχουμε το λαμπτήρα, και την ισχύ (*Intensity*) με την οποία θα τροφοδοτηθεί αυτός (κλίμακα 0-100). Όσο μεγαλύτερη ισχύ καθορίσουμε, τόσο πιο έντονο θα είναι το φως του λαμπτήρα.

Για να κλείσουμε ένα λαμπτήρα επιλέγουμε στη ρύθμιση της ενέργειας (*Action*) την παράμετρο του σβήσιματος "Off".

Επεξήγηση συμβόλων

1. Το γράμμα πάνω δεξιά δηλώνει τη θύρα εξόδου στην οποία θέλουμε να ελέγχουμε το λαμπτήρα.
2. Η εικόνα κάτω στη μέση δηλώνει την ενέργεια : άναμμα "On" ή σβήσιμο "Off"
3. Η εικόνα κάτω δεξιά εμφανίζει την ένταση του λαμπτήρα.



Μπλοκ ήχου Sound Block



Μπλοκ εμφάνισης Display Block

Βλέπε στην περιγραφή των εντολών στη βασική παλέτα



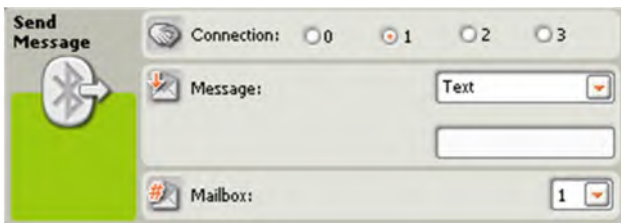
Μπλοκ αποστολής μηνύματος Send Message Block

Με αυτήν την εντολή μπορούμε να πούμε στο NXT του ρομπότ μας να στείλει ένα μήνυμα ασύρματα σε ένα άλλο NXT.

Τη δυνατότητα αποστολής και λήψης μηνυμάτων μπορούμε να τη χρησιμοποιήσουμε όταν θέλουμε το ρομπότ μας να είναι σε θέση να επικοινωνήσει με ένα άλλο NXT ρομπότ και να ανταλλάξει πληροφορίες μαζί του. Μια άλλη εφαρμογή της ανταλλαγής μηνυμάτων για την περίπτωση που έχουμε και ένα δεύτερο τούβλο NXT είναι να το χρησιμοποιήσουμε ως συσκευή απομακρυσμένης διαχείρισης (remote control) μέσω του οποίου μπορούμε να στέλνουμε εντολές στο ρομπότ μας.

Για να μπορέσει να γίνει αποστολή και λήψη μηνυμάτων θα πρέπει οι μικροϋπολογιστές NXT που θα συμμετέχουν στην ανταλλαγή μηνυμάτων να έχουν ρυθμιστεί κατάλληλα για ασύρματη επικοινωνία οπότε και να τους έχει αποδοθεί ένας αριθμός σύνδεσης (connection number).

Πίνακας Ρυθμίσεων



Για να στείλουμε ένα μήνυμα πρέπει πρώτα να δηλώσουμε τον αριθμό σύνδεσης (*Connection*) του NXT, στο οποίο θέλουμε να απευθυνθούμε (δέκτη) και στη συνέχεια να καθορίσουμε το είδος του μηνύματος (*Message*) ανάμεσα στο κείμενο *''text''*, αριθμό *''number''* ή λογικό αριθμό *''logic''*. Στη συνέχεια αφού γράψουμε στο μήνυμα μας στο πεδίο *Message* πρέπει να επιλέξουμε τον αριθμό της θυρίδας ταχυδρομείου (*mailbox*) όπου το μήνυμα θα αποθηκευτεί στο NXT δέκτη.

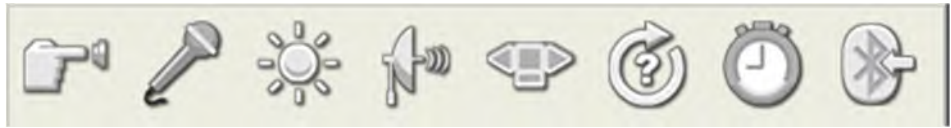
Επεξήγηση συμβόλων

1. Η εικόνα κάτω αριστερά δηλώνει το είδος του μηνύματος : κείμενο *''text''*, αριθμός *''number''* ή λογικός αριθμός *''logic''*.
2. Η εικόνα κάτω στη μέση εμφανίζει τον αριθμό της σύνδεσης.
3. Η εικόνα κάτω δεξιά εμφανίζει τον αριθμό της θυρίδας.





Μπλοκ Αισθητήρων Sensor Blocks



Μπλοκ αισθητήρα αφής Touch Sensor Block

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για την κατάσταση ενός αισθητήρα αφής σε ένα συγκεκριμένο σημείο του προγράμματος μας. Αν είχε προηγηθεί κάποια στιγμή προηγουμένως, η ενέργεια που έχουμε δηλώσει στον πίνακα ρυθμίσεων της εντολής, η εντολή αυτή θα μας επιστρέφει το λογικό σήμα 1 (true).

Σημείωση : Για να μπορέσουμε να αξιοποιήσουμε την πληροφορία από αυτήν την εντολή πρέπει να καλωδιώσουμε το σημείο της λογικής εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (Port) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου νο 1.

Επίσης δηλώνουμε ποια ενέργεια (Action) στον αισθητήρα αφής μας ενδιαφέρει : *''Pressed''*, απελευθέρωση *''Released''* ή πάτημα και απελευθέρωση *''Bumped''*.

Η εντολή αυτή επιστρέφει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων.

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά δηλώνει την ενέργεια που θα αποστείλει το σήμα ενεργοποίησης *true* : πάτημα διακόπτη *''Pressed''*, απελευθέρωση *''Released''* ή πάτημα και απελευθέρωση *''Bumped''*.
3. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.





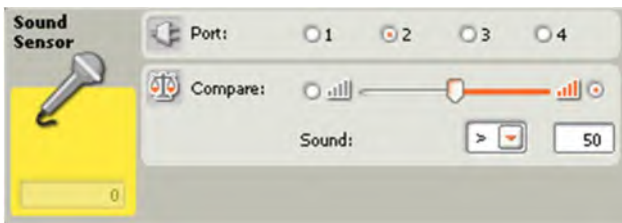
Μπλοκ αισθητήρα ήχου Sound Sensor Block

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για τη στάθμη του ήχου που ανιχνεύει ένας αισθητήρας ήχου σε ένα συγκεκριμένο σημείο του προγράμματος μας. Η εντολή αυτή λειτουργεί σαν ένας ανιχνευτής ήχου.

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της στάθμης του ήχου σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η στάθμη αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (Port) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου νο 2.

Επίσης καθορίζουμε την τιμή της στάθμης του ήχου που θέλουμε να συγκρίνουμε σέρνοντας τον αντίστοιχο σύρτη (Compare) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η στάθμη του ήχου βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα *true* για επίπεδα στάθμης ήχου πάνω από 50%.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα στάθμη του ήχου. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά δηλώνει τη στάθμη του ήχου που χρησιμοποιείται ως σημείο κατωφλίου.
3. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.





Μπλοκ αισθητήρα φωτός Light Sensor Block

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για τη στάθμη του φωτός που ανιχνεύει ένας αισθητήρας φωτός σε ένα συγκεκριμένο σημείο του προγράμματος μας.

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της στάθμης του φωτός σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η στάθμη αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (*Port*) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου no 3.

Επίσης καθορίζουμε την τιμή της στάθμης του φωτός που θέλουμε να συγκρίνουμε σέρνοντας τον αντίστοιχο σύρτη (*Compare*) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η στάθμη του φωτός βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν αφήσουμε ενεργή την παράμετρο της παραγωγής φωτός "*Generate light*" της ρύθμισης της λειτουργίας (*Function*) τότε ο αισθητήρας φωτός εκπέμπει τη δική του υπέρυθη ακτινοβολία. Διαφορετικά, αν απενεργοποιήσουμε δηλαδή, τη συγκεκριμένη παράμετρο τότε ο αισθητήρας θα ανιχνεύει την ένταση του φωτός από το περιβάλλον. Στη δεύτερη περίπτωση η εντολή λειτουργεί σαν ένας ανιχνευτής της φωτεινότητας που υπάρχει στο περιβάλλον που δραστηριοποιείται το ρομπότ μας. Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα *true* για επίπεδα στάθμης φωτός πάνω από 50%.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα στάθμη του φωτός. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά δηλώνει το επίπεδο του φωτός που χρησιμοποιείται ως σημείο κατωφλίου.
3. Η εικόνα κάτω στη μέση δηλώνει αν είναι ενεργοποιημένη ή παραγωγή υπέρυθρης ακτινοβολίας (επιλογή Generated Light) στον αισθητήρα.
4. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



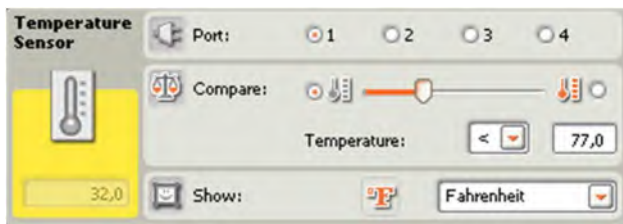
**Μπλοκ αισθητήρα θερμοκρασίας
Temperature Sensor Block**

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για τη θερμοκρασία που ανιχνεύει ένας αισθητήρας θερμοκρασίας σε ένα συγκεκριμένο σημείο του προγράμματος μας.

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της θερμοκρασίας σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η τιμή αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (*Port*) στην οποία είναι συνδεδεμένος ο αισθητήρας. Επίσης καθορίζουμε την τιμή της θερμοκρασίας που θέλουμε να συγκρίνουμε σέρνοντας τον αντίστοιχο σύρτη (*Compare*) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Σενάρια Ρομποτικής με το Mindstorms NXT

Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η θερμοκρασία βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα *true* για επίπεδα θερμοκρασίας κάτω 25° C.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα στάθμη της θερμοκρασίας. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά δηλώνει το επίπεδο της θερμοκρασίας που χρησιμοποιείται ως σημείο κατωφλίου.
3. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ αισθητήρα υπερήχων
Ultrasonic Sensor Block

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για την απόσταση στην οποία βρίσκεται κάποιο εμπόδιο, τοποθετημένο μπροστά από το σόναρ (αισθητήρα υπερήχων) του ρομπότ μας. Η μέγιστη απόσταση στην οποία μπορεί να λειτουργήσει το σόναρ είναι 250 cm (100 inches)

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της απόστασης από ένα εμπόδιο σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η στάθμη αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (*Port*) στην οποία είναι συνδεδεμένος ο αισθητήρας. Εξ' ορισμού ελέγχεται η θύρα εισόδου no 4.

Σενάρια Ρομποτικής με το Mindstorms NXT

Επίσης καθορίζουμε την τιμή της απόστασης που θέλουμε να συγκρίνουμε σέρνοντας τον αντίστοιχο σύρτη (*Compare*) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η απόσταση από ένα εμπόδιο βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα *true* όταν η απόσταση από ένα εμπόδιο είναι μικρότερη από 50 ίντσες (127 cm).

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα απόσταση από κάποιο εμπόδιο. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

3. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
4. Η εικόνα κάτω δεξιά δηλώνει την απόσταση που χρησιμοποιείται ως σημείο κατωφλίου.
5. Ο κατανεμητής δεδομένων (*data hub*) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.

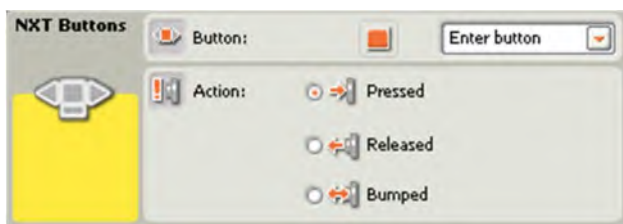


Μπλοκ NXT πλήκτρων
NXT Buttons Block

Αυτήν η εντολή στέλνει ένα λογικό σήμα *true* μέσω της καλωδίωσης δεδομένων όταν δράσουμε κατάλληλα πάνω σε κάποιο από τα πλήκτρα που βρίσκονται πάνω στο NXT. Αν είχε προηγηθεί κάποια στιγμή προηγουμένως, η ενέργεια που έχουμε δηλώσει στον πίνακα ρυθμίσεων της εντολής για ένα συγκεκριμένο πλήκτρο του τούβλου NXT, η εντολή αυτή θα μας επιστρέφει το λογικό σήμα 1 (*true*).

Σημείωση : Για να μπορέσουμε να αξιοποιήσουμε την πληροφορία από αυτήν την εντολή πρέπει να καλωδιώσουμε το σημείο της λογικής εξόδου του κατανεμητή δεδομένων (*data hub*) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής.

Πίνακας Ρυθμίσεων



Σενάρια Ρομποτικής με το Mindstorms NXT

Στον πίνακα ρυθμίσεων δηλώνουμε το πλήκτρο (*Button*) που επιθυμούμε (ανάμεσα στα enter δεξί και αριστερό πλήκτρο).

Επίσης δηλώνουμε ποια ενέργεια (*Action*) στον αισθητήρα αφής μας ενδιαφέρει : *''Pressed''*, απελευθέρωση *''Released''* ή πάτημα και απελευθέρωση *''Bumped''*.

Η εντολή αυτή επιστρέφει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*).

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την κατάσταση του επιλεγμένου πλήκτρου. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

3. Η εικόνα στη μέση δεξιά εμφανίζει το πλήκτρο που παρακολουθείται : πορτοκαλί (enter) ή γκρι αριστερό (left) ή δεξιό βελάκι (right)

4. Η εικόνα κάτω δεξιά δηλώνει την ενέργεια που θα αποστείλει το σήμα ενεργοποίησης *true* : πάτημα διακόπτη, απελευθέρωση ή πάτημα και απελευθέρωση.

5. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



**Μπλοκ αισθητήρα περιστροφής
Rotation Sensor Block**

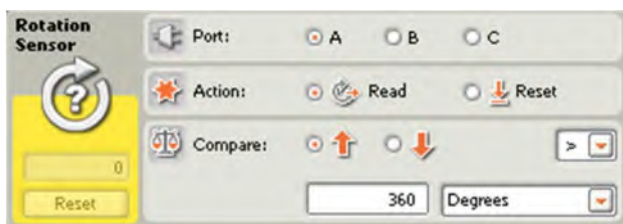
Σε κάθε κινητήρα του πακέτου NXT υπάρχει και ένας ενσωματωμένος αισθητήρας περιστροφής. Αντίστοιχα ο μικρουπολογιστής NXT διαθέτει 3 εσωτερικούς μετρητές για την περιστροφή κάθε κινητήρα στις θύρες εξόδου του.

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για τη γωνιακή μετατόπιση (περιστροφή) που έχει πραγματοποιήσει ένας κινητήρας σε μοίρες degrees (μία πλήρη περιστροφή είναι 360°) ή σε πλήρεις περιστροφές.

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της περιστροφής σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η τιμή αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Πίνακας Ρυθμίσεων



Σενάρια Ρομποτικής με το Mindstorms NXT

Στον πίνακα ρυθμίσεων επιλέγουμε πρώτα τον κινητήρα που θέλουμε να ελέγξουμε (*Port*).

Επίσης καθορίζουμε τη φορά περιστροφής που μας ενδιαφέρει και την τιμή της περιστροφής που θέλουμε να συγκρίνουμε, πληκτρολογώντας τη στο αντίστοιχο πεδίο (*Compare*). Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η περιστροφή βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα *true* όταν η περιστροφή που έχει πραγματοποιηθεί είναι μεγαλύτερη από 360° δηλ μία πλήρη περιστροφή.

Για να μηδενίσουμε το μετρητή για την περιστροφή επιλέγουμε στη ρύθμιση της ενέργειας (*Action*) αντί για την παράμετρο "*Read*" τη "*Reset*". Έτσι αν θέλουμε να ξεκινήσουμε από την αρχή μία μέτρηση για την περιστροφή που θα πραγματοποιήσει κάποιος κινητήρας, στο πρόγραμμα που θα δημιουργήσουμε θα πρέπει να τοποθετήσουμε πιο μπροστά μία εντολή αισθητήρα περιστροφής με επιλεγμένη την παράμετρο "*Reset*", για να μηδενίσουμε τον αντίστοιχο μετρητή περιστροφής.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα περιστροφή που έχει πραγματοποιηθεί. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

4. Το γράμμα πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
5. Η εικόνα κάτω στη μέση δηλώνει την κατεύθυνση προς στην οποία θα πραγματοποιηθεί η μέτρηση.
6. Η εικόνα κάτω δεξιά δηλώνει την ενέργεια : παρακολούθηση των περιστροφών "*Read*" ή μηδενισμός του μετρητή "*Reset*".
7. Ο κατανεμητής δεδομένων (*data hub*) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ χρονομετρητή
Timer Block

Ο μικροεπεξεργαστής NXT διαθέτει 3 εσωτερικούς χρονομετρητές. Όταν ένα πρόγραμμα ξεκινάει, αυτοί οι χρονομετρητές ξεκινάνε αυτόματα να μετράνε το χρόνο που περνάει.

Με αυτήν την εντολή μπορούμε να επιλέξουμε είτε να διαβάσουμε την τρέχουσα τιμή του χρόνου ή να μηδενίσουμε τον αντίστοιχο χρονομετρητή ώστε να ξεκινήσουμε μία νέα μέτρηση από το μηδέν.

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή του χρονομετρητή σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η τιμή αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σενάρια Ρομποτικής με το Mindstorms NXT

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων ορίζουμε πρώτα το χρονομετρητή (*Timer*) που θέλουμε να παρατηρήσουμε και καθορίζουμε το χρόνο που θέλουμε να συγκρίνουμε πληκτρολογώντας τον στο αντίστοιχο πεδίο (*Compare*). Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν ο χρόνος βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Μπορούμε να προμηθεύσουμε την τιμή κατωφλίου δυναμικά καλωδιώνοντας την αντίστοιχη είσοδο του κατανεμητή δεδομένων της εντολής αυτής και έτσι δίνοντας την τιμή κατωφλίου για το χρόνο από κάποιο αντίστοιχο σημείο εξόδου μιας άλλης εντολής.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα *true* όταν ο χρόνος που έχει περάσει είναι μεγαλύτερος από 5 δευτερόλεπτα.

Για να μηδενίσουμε έναν χρονομετρητή επιλέγουμε στη ρύθμιση της ενέργειας (*Action*) αντί για την παράμετρο "*Read*" τη "*Reset*".

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή το χρόνο που έχει διανυθεί. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

3. Η εικόνα κάτω στη μέση δηλώνει την ενέργεια : παρακολούθηση των χρόνου (*read*) ή μηδενισμός του χρονομετρητή (*reset*).
4. Η εικόνα κάτω δεξιά εμφανίζει τον αριθμό του χρονομετρητή που παρακολουθείται.
5. Ο κατανεμητής δεδομένων (*data hub*) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.





Μπλοκ λήψης μηνύματος Receive Message Block

Με αυτήν την εντολή μπορούμε να διαβάσουμε ένα μήνυμα το οποίο έχει ληφθεί ασύρματα.

Για να μπορέσει να γίνει αποστολή και λήψη μηνυμάτων θα πρέπει οι μικροϋπολογιστές NXT που θα συμμετέχουν στην ανταλλαγή μηνυμάτων να έχουν ρυθμιστεί κατάλληλα για ασύρματη επικοινωνία οπότε και να τους έχει αποδοθεί ένας αριθμός σύνδεσης (connection number).

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Πίνακας Ρυθμίσεων



Για να λάβουμε ένα μήνυμα πρέπει πρώτα να δηλώσουμε το είδος του μηνύματος (*Message*) ανάμεσα στο κείμενο *''Text''*, αριθμό *''Number''* ή λογικό αριθμό *''Logic''* που περιμένουμε να αποσταλεί στο NXT μας, και να καθορίσουμε το ακριβές περιεχόμενο του προς σύγκριση. Στη συνέχεια πρέπει να επιλέξουμε τον αριθμό της θυρίδας ταχυδρομείου (*Mailbox*) όπου το μήνυμα θα αποθηκευτεί στο NXT μας.

Αν το εισερχόμενο μήνυμα ταιριάζει με το περιεχόμενο με το οποίο καθορίσαμε να γίνει η σύγκριση του τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα true.

Επεξήγηση συμβόλων

3. Η εικόνα κάτω στη μέση δηλώνει το είδος του μηνύματος : κείμενο *''Text''*, αριθμός *''Number''* ή λογικός αριθμός *''Logic''*.
4. Η εικόνα κάτω δεξιά εμφανίζει την τον αριθμό της θυρίδας.
5. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Σημείωση : Για πληροφορίες για την αποστολή μηνυμάτων ασύρματα βλέπε στο Μπλοκ αποστολή μηνύματος



Μπλοκ αισθητήρα αφής NXT Touch* Sensor Block (old)

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για την κατάσταση ενός αισθητήρα αφής παλιού τύπου, από τη σειρά NXT της Lego, σε ένα συγκεκριμένο σημείο του προγράμματος μας.

Σημείωση : Για να μπορέσουμε να αξιοποιήσουμε την πληροφορία από αυτήν την εντολή πρέπει να καλωδιώσουμε το σημείο της λογικής εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής.

Σημείωση : Για να συνδέσουμε ένα αισθητήρα αυτού του τύπου στο NXT πρέπει να χρησιμοποιήσουμε το αντίστοιχο καλώδιο προσαρμογής.

Πίνακας Ρυθμίσεων

Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (Port) στην οποία είναι συνδεδεμένος ο αισθητήρας.

Επίσης δηλώνουμε ποια ενέργεια (Action) στον αισθητήρα αφής μας ενδιαφέρει : *''Pressed''*, απελευθέρωση *''Released''* ή πάτημα και απελευθέρωση *''Bumped''*. Η εντολή αυτή επιστρέφει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων.

Επεξήγηση συμβόλων

4. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.

5. Η εικόνα κάτω δεξιά δηλώνει την ενέργεια που θα αποστείλει το σήμα ενεργοποίησης *true* : πάτημα διακόπτη *''Pressed''*, απελευθέρωση *''Released''* ή πάτημα και απελευθέρωση *''Bumped''*.

6. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ αισθητήρα φωτός NXT Light* Sensor Block (old)

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για τη στάθμη του φωτός που ανιχνεύει ένας αισθητήρας φωτός παλιού τύπου, από τη σειρά NXT της Lego, σε ένα συγκεκριμένο σημείο του προγράμματος μας.

Σε αυτό τον τύπο του αισθητήρα φωτός ο πομπός υπέρυθρης ακτινοβολίας είναι συνέχεια ενεργοποιημένος.

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της στάθμης του φωτός σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η στάθμη αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σενάρια Ρομποτικής με το Mindstorms NXT

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Σημείωση : Για να συνδέσουμε ένα αισθητήρα αυτού του τύπου στο NXT πρέπει να χρησιμοποιήσουμε το αντίστοιχο καλώδιο προσαρμογής.

Πίνακας Ρυθμίσεων

Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (Port) στην οποία είναι συνδεδεμένος ο αισθητήρας.

Επίσης καθορίζουμε την τιμή της στάθμης του φωτός που θέλουμε να συγκρίνουμε σέρνοντας τον αντίστοιχο σύρτη (Compare) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η στάθμη του φωτός βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα true για επίπεδα στάθμης φωτός πάνω από 50%.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα στάθμη του φωτός. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά δηλώνει το επίπεδο του φωτός που χρησιμοποιείται ως σημείο κατωφλίου.
3. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



**Μπλοκ αισθητήρα περιστροφής NXT
Rotation* Sensor Block (old)**

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για τη γωνιακή μετατόπιση (περιστροφή) ενός αισθητήρα περιστροφής παλιού τύπου, από τη σειρά NXT της Lego, σε βήματα του 1/16 μιας πλήρους περιστροφής (μία πλήρη περιστροφή είναι 16/16).

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της περιστροφής σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η τιμή αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σενάρια Ρομποτικής με το Mindstorms NXT

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Σημείωση : Για να συνδέσουμε ένα αισθητήρα αυτού του τύπου στο NXT πρέπει να χρησιμοποιήσουμε το αντίστοιχο καλώδιο προσαρμογής.

Πίνακας Ρυθμίσεων

Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (Port) στην οποία είναι συνδεδεμένος ο αισθητήρας.

Επίσης καθορίζουμε τη φορά περιστροφής που μας ενδιαφέρει και την τιμή της περιστροφής που θέλουμε να συγκρίνουμε, πληκτρολογώντας τη στο αντίστοιχο πεδίο (Compare). Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (true ή false) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η περιστροφή βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα true όταν η περιστροφή που έχει πραγματοποιηθεί είναι μεγαλύτερη από 16 δηλ μία πλήρη περιστροφή.

Για να μηδενίσουμε το μετρητή για την περιστροφή επιλέγουμε στη ρύθμιση της ενέργειας (Action) αντί για την παράμετρο "Read" τη "Reset". Έτσι αν θέλουμε να ξεκινήσουμε από την αρχή μία μέτρηση για την περιστροφή που θα πραγματοποιήσει κάποιος κινητήρας, στο πρόγραμμα που θα δημιουργήσουμε θα πρέπει να τοποθετήσουμε πιο μπροστά μία εντολή αισθητήρα περιστροφής με επιλεγμένη την παράμετρο "Reset", για να μηδενίσουμε τον αντίστοιχο μετρητή περιστροφής.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα περιστροφή που έχει πραγματοποιηθεί. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω στη μέση δηλώνει την κατεύθυνση προς στην οποία θα πραγματοποιηθεί η μέτρηση.
3. Η εικόνα κάτω δεξιά δηλώνει την ενέργεια : παρακολούθηση των περιστροφών "Read" ή μηδενισμός του μετρητή "Reset".
4. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



**Μπλοκ αισθητήρα θερμοκρασίας NXT
Temperature* Sensor Block (old)**

Με αυτήν την εντολή μπορούμε να πάρουμε πληροφορίες για τη θερμοκρασία που ανιχνεύει ένας αισθητήρας θερμοκρασίας παλιού τύπου, από τη σειρά NXT της Lego, σε ένα συγκεκριμένο σημείο του προγράμματος μας.

Σενάρια Ρομποτικής με το Mindstorms NXT

Χρησιμοποιώντας καλωδίωση δεδομένων μπορούμε να στείλουμε την τρέχουσα τιμή της θερμοκρασίας σε ένα άλλο μπλοκ καθώς και ένα λογικό αριθμό βασισμένο στο γεγονός εάν η τιμή αυτή είναι μεγαλύτερη ή μικρότερη από μία συγκεκριμένη αριθμητική τιμή που καθορίζουμε εμείς ως σημείο κατωφλίου.

Σημείωση : Πρέπει να καλωδιώσουμε ένα τουλάχιστον σημείο εξόδου του κατανεμητή δεδομένων (data hub) της εντολής αυτής, προς κάποιο αντίστοιχο σημείο εισόδου μιας άλλης εντολής για να πάρουμε πληροφορίες από αυτή την εντολή.

Σημείωση : Για να συνδέσουμε ένα αισθητήρα αυτού του τύπου στο NXT πρέπει να χρησιμοποιήσουμε το αντίστοιχο καλώδιο προσαρμογής.

Πίνακας Ρυθμίσεων

Στον πίνακα ρυθμίσεων καθορίζουμε τη θύρα (Port) στην οποία είναι συνδεδεμένος ο αισθητήρας.

Επίσης καθορίζουμε την τιμή της θερμοκρασίας που θέλουμε να συγκρίνουμε σέρνοντας τον αντίστοιχο σύρτη (Compare) είτε πληκτρολογώντας τη στο αντίστοιχο πεδίο. Στη συνέχεια καθορίζουμε την περιοχή που μας ενδιαφέρει (πάνω ή κάτω από την τιμή κατωφλίου). Η περιοχή αυτή εμφανίζεται χρωματισμένη κόκκινη.

Η εντολή αυτή θα μας επιστρέψει ως αποτέλεσμα ένα λογικό αριθμό 1=ναι ή 0=όχι (*true* ή *false*) μέσω της καλωδίωσης δεδομένων, ανάλογα με το αν η θερμοκρασία βρίσκεται εντός της οριζόμενης περιοχής ή όχι.

Αν δεν πειράξουμε καμία ρύθμιση του πίνακα ρυθμίσεων τότε η εντολή αυτή θα μας επιστρέψει το λογικό σήμα true για επίπεδα θερμοκρασίας κάτω 25° C.

Στο πλαίσιο ανατροφοδότησης στην αριστερή πλευρά του πίνακα ρυθμίσεων μπορούμε να βλέπουμε ανά πάσα στιγμή την τρέχουσα στάθμη της θερμοκρασίας. (με την προϋπόθεση βεβαίως ότι η επικοινωνία του NXT με τον υπολογιστή μας είναι ενεργοποιημένη).

Επεξήγηση συμβόλων

1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που παρακολουθείται - είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω δεξιά δηλώνει το επίπεδο της θερμοκρασίας που χρησιμοποιείται ως σημείο κατωφλίου.
3. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας





Μπλοκ Ροής - Δομές Flow Blocks



Βλέπε στην περιγραφή των εντολών στη βασική παλέτα



Μπλοκ Σταματήματος Stop Block

Το ρομπότ πάντοτε σταματάει όταν φτάσει στο τέλος του προγράμματος εκτός αν έχουμε κάπου, κάποια εντολής επανάληψης η οποία να βάζει το πρόγραμμα μας ή κάποιο μέρος του να επαναλαμβάνεται συνεχώς. Στην περίπτωση που το πρόγραμμα επαναλαμβάνεται συνεχώς, μπορούμε να το σταματήσουμε είτε πατώντας το πλήκτρο ακύρωσης στο τούβλο NXT είτε τοποθετώντας κάπου μέσα στο βρόγχο μία εντολή σταματήματος εκτέλεσης του προγράμματος.

Αυτή η εντολή σταματάει την εκτέλεση του προγράμματος μας αμέσως, καθώς και σταματάει όλους τους κινητήρες, σβήνει όλους τους λαμπτήρες και διακόπτει όλους τους παραγόμενους ήχους. Οι κινητήρες θα σταματήσουν μετά από ολίσθηση και όχι με απότομο σταμάτημα –φρένο.

Μπορούμε να στείλουμε μία λογική τιμή true/false ώστε να ελέγξουμε αυτήν την εντολή δυναμικά καλωδιώνοντας την αντίστοιχη είσοδο του κατανεμητή δεδομένων της εντολής αυτής από κάποιο αντίστοιχο σημείο εξόδου μιας άλλης εντολής. Σε αυτήν την περίπτωση η εντολή αυτή θα δράσει μόνο αν το σήμα που θα φθάσει εκείνη τη στιγμή στο σημείο της λογικής εισόδου του κατανεμητή είναι 1 (true), σε διαφορετική περίπτωση η εντολή θα αγνοηθεί.





Μπλοκ Δεδομένων Data Blocks

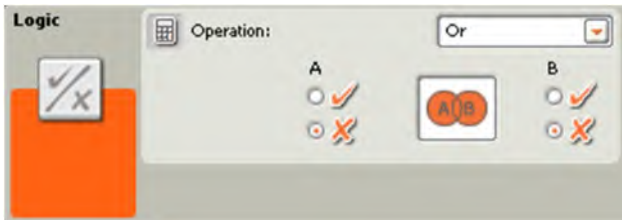


Μπλοκ λογικών πράξεων Logic Block

Αυτή η εντολή εκτελεί μία λογική πράξη ανάμεσα σε δύο λογικά δεδομένα και δίνει το αποτέλεσμα της πράξης αυτής μέσω της καλωδίωσης δεδομένων.

Τα δεδομένα που λαμβάνουν μέρος στην πράξη αυτή, καθώς και το αποτέλεσμα είναι λογικοί αριθμοί της μορφής 1=ναι ή 0=όχι (*true* ή *false*). Οι λογικές πράξεις που μπορεί να εκτελέσει αυτή η εντολή είναι οι OR, AND, XOR και NOT και οι οποίες μας επιτρέπουν να εκτελούμε μία σειρά από συγκρίσεις.

Πίνακας Ρυθμίσεων



Τα δεδομένα που επεξεργάζεται η συγκεκριμένη εντολή μπορούν να δοθούν από τον πίνακα ρύθμισης της εντολής, από όπου μπορούμε να επιλέξουμε τις τιμές για τα ορίσματα A και B. Για να αξιοποιήσουμε όμως κατάλληλα την εντολή αυτή, ένα τουλάχιστον όρισμα επιβάλλεται να δοθεί δυναμικά μέσω καλωδίωσης των σημείων εισόδου δεδομένων A και B του κατανεμητή δεδομένων (data hub).

Μια εντολή λογικών πράξεων που τοποθετείται στην περιοχή εργασίας εμφανίζεται με αναδιπλωμένο τον κατανεμητή δεδομένων με δύο εισόδους A και B στην αριστερή του πλευρά για τα δεδομένα εισόδου και τρεις εξόδους A, B και $\sqrt{x}$ στη δεξιά του πλευρά. Στην έξοδο $\sqrt{x}$ στέλνεται το αποτέλεσμα της πράξης ενώ από τις εξόδους A και B, αν είναι απαραίτητο, μπορούμε να περάσουμε τα δεδομένα εισόδου της εντολής αυτής απευθείας στη είσοδο κάποιας άλλης εντολής.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Επεξήγηση συμβόλων

1. Το σύμβολο κάτω δεξιά δηλώνει τον τύπο της λογικής πράξης : OR, AND, XOR και NOT.
2. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



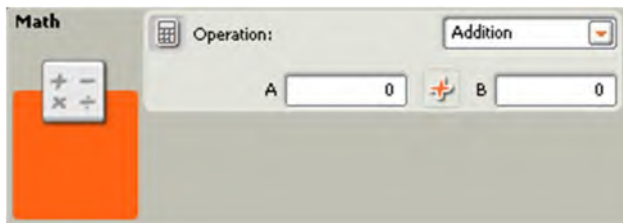
**Μπλοκ Μαθηματικών πράξεων
Math Block**

Αυτή η εντολή εκτελεί μία απλή μαθηματική πράξη ανάμεσα σε δύο πραγματικούς αριθμούς και δίνει το αποτέλεσμα της πράξης αυτής μέσω της καλωδίωσης δεδομένων.

Οι μαθηματικές πράξεις που μπορεί να εκτελέσει αυτή η εντολή είναι οι :

- πρόσθεση Addition (+) (εξ ορισμού)
- αφαίρεση Subtraction (-)
- πολλαπλασιασμός Multiplication (x)
- διαίρεση Division (/)
- απόλυτο αριθμός Absolute Value ([x])
- τετραγωνική ρίζα Square Root ($\sqrt{x}$)

Πίνακας Ρυθμίσεων



Τα δεδομένα που επεξεργάζεται η συγκεκριμένη εντολή μπορούν να δοθούν από τον πίνακα ρύθμισης της εντολής, από όπου μπορούμε να επιλέξουμε τις τιμές για τα ορίσματα A και B. Για να αξιοποιήσουμε όμως κατάλληλα την εντολή αυτή, ένα τουλάχιστον όρισμα επιβάλλεται να δοθεί δυναμικά μέσω των καλωδιώσεων δεδομένων εισόδου A και B.

Ένα μπλοκ μαθηματικών πράξεων που τοποθετείται στην περιοχή εργασίας εμφανίζεται με αναδιπλωμένο τον κατανεμητή δεδομένων με δύο εισόδους A και B στην αριστερή του πλευρά για τα δεδομένα εισόδου και τρεις εξόδους A, B και # στη δεξιά του πλευρά. Στην έξοδο # στέλνεται το αποτέλεσμα της πράξης ενώ από τις εξόδους A και B, αν είναι απαραίτητο, μπορούμε να περάσουμε τα δεδομένα εισόδου της εντολής απευθείας στη είσοδο κάποιας άλλης εντολής.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Επεξήγηση συμβόλων

1. Το σύμβολο κάτω δεξιά δηλώνει τον τύπο της μαθηματικής πράξης : +, -, x, /, [x] και √x.
2. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ Σύγκρισης
Compare Block

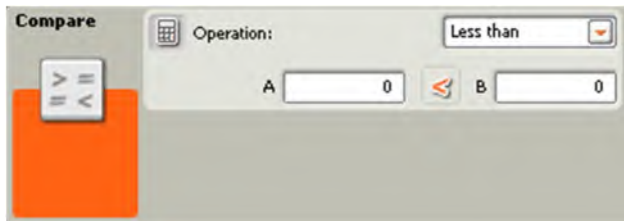
Με αυτή την εντολή εκτελεί συγκρίσεις ανάμεσα σε δύο αριθμούς και δίνει το αποτέλεσμα της σύγκρισης αυτής μέσω της καλωδίωσης δεδομένων.

Η εντολή αυτή συγκρίνει αν ένας αριθμός είναι μεγαλύτερος, μικρότερος ή ίσος με κάποιο άλλο και το αποτέλεσμα της σύγκρισης είναι ένας λογικός αριθμός της μορφής 1=ναι ή 0=όχι (*true ή false*).

Οι συγκρίσεις που μπορεί να εκτελέσει αυτή η εντολή είναι οι :

- Μεγαλύτερο από Greater Than (>)
- Μικρότερο από Less Than (<)
- ίσο με Equal To (=)

Πίνακας Ρυθμίσεων



Τα δεδομένα που επεξεργάζεται η συγκεκριμένη εντολή μπορούν να δοθούν από τον πίνακα ρύθμισης της εντολής, από όπου μπορούμε να επιλέξουμε τις τιμές για τα ορίσματα A και B. Για να αξιοποιήσουμε όμως κατάλληλα την εντολή αυτή, ένα τουλάχιστον όρισμα επιβάλλεται να δοθεί δυναμικά μέσω των καλωδιώσεων δεδομένων εισόδου A και B.

Ένα μπλοκ σύγκρισης που τοποθετείται στην περιοχή εργασίας εμφανίζεται με αναδιπλωμένο τον κατανεμητή δεδομένων με δύο εισόδους A και B στην αριστερή του πλευρά για τα δεδομένα εισόδου και τρεις εξόδους A, B και √/χ στη δεξιά του πλευρά. Στην έξοδο √/χ στέλνεται το αποτέλεσμα της πράξης ενώ από τις εξόδους A και B, αν είναι απαραίτητο, μπορούμε να περάσουμε τα δεδομένα εισόδου της εντολής αυτής απευθείας στη είσοδο κάποιας άλλης εντολής.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Επεξήγηση συμβόλων

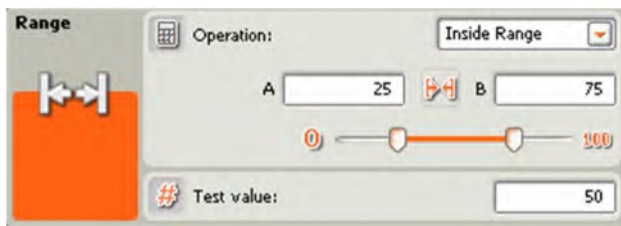
1. Το σύμβολο κάτω δεξιά δηλώνει τον τύπο της σύγκρισης : <, > και =
2. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ εύρους αριθμών
Range Block

Η εντολή αυτή ελέγχει αν ένας αριθμός βρίσκεται εντός ή εκτός μιας περιοχής αριθμών και δίνει το αποτέλεσμα της σύγκρισης αυτής μέσω της καλωδίωσης δεδομένων. Το αποτέλεσμα του ελέγχου αυτού είναι ένας λογικός αριθμός της μορφής 1=ναι ή 0=όχι (*true* ή *false*).

Πίνακας Ρυθμίσεων



Τα δεδομένα που επεξεργάζεται η συγκεκριμένη εντολή μπορούν να δοθούν από τον πίνακα ρύθμισης της εντολής από όπου μπορούμε να επιλέξουμε μία τιμή για το όρισμα που επιθυμούμε να ελέγξουμε (# *Test value*) και δύο άλλες τιμές A και B για να καθορίσουμε το εύρος των αριθμών με το οποίο θα γίνει ο έλεγχος. Για να αξιοποιήσουμε όμως κατάλληλα την εντολή αυτή, ένα τουλάχιστον όρισμα επιβάλλεται να δοθεί δυναμικά μέσω των καλωδιώσεων δεδομένων εισόδου A, B και #.

Ένα μπλοκ εύρους αριθμών που τοποθετείται στην περιοχή εργασίας εμφανίζεται με αναδιπλωμένο τον κατανεμητή δεδομένων με μία είσοδο # στην αριστερή του πλευρά για το δεδομένο εισόδου και δύο εξόδους # και √/χ στη δεξιά του πλευρά. Στην έξοδο √/χ στέλνεται το αποτέλεσμα της πράξης ενώ από την έξοδο #, αν είναι απαραίτητο, μπορούμε να περάσουμε το δεδομένο εισόδου της εντολής αυτής απευθείας στη είσοδο κάποιας άλλης εντολής.

Αν κάνουμε κλικ στο πάνω μέρος του κατανεμητή δεδομένων θα εμφανιστούν και οι είσοδοι/έξοδοι A και B.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Επεξήγηση συμβόλων

1. Το σύμβολο κάτω δεξιά δηλώνει ποια περιοχή ανάμεσα στους δύο αριθμούς A και B εξετάζουμε : την εσωτερική (Inside Range) ([]) ή την εξωτερική Outside Range ([I])
2. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ τυχαίου αριθμού Random Block

Η εντολή αυτή παράγει έναν τυχαίο αριθμό ο οποίος βρίσκεται εντός μιας περιοχής αριθμών που επιλέγουμε. Μπορούμε να χρησιμοποιήσουμε τυχαίους αριθμούς για να προσδώσουμε απρόβλεπτες συμπεριφορές στο ρομπότ μας, όπως συμβαίνει και στο φυσικό κόσμο μας.

Πίνακας Ρυθμίσεων



Τα δεδομένα που επεξεργάζεται η συγκεκριμένη εντολή μπορούν να δοθούν από τον πίνακα ρύθμισης της εντολής από όπου μπορούμε να επιλέξουμε τις δύο τιμές, την ελάχιστη *“Minimum”* A και τη μέγιστη *“Maximum”* B, για να καθορίσουμε το εύρος των αριθμών στο οποίο θα ανήκει ο τυχαία παραγόμενος αριθμός.

Ένα μπλοκ εύρους αριθμών που τοποθετείται στην περιοχή εργασίας εμφανίζεται με αναδιπλωμένο τον κατανεμητή δεδομένων με μία έξοδο # στη δεξιά του πλευρά. Αν κάνουμε κλικ στο πάνω μέρος του κατανεμητή δεδομένων θα εμφανιστούν και οι εισοδοί/έξοδοι A και B.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.



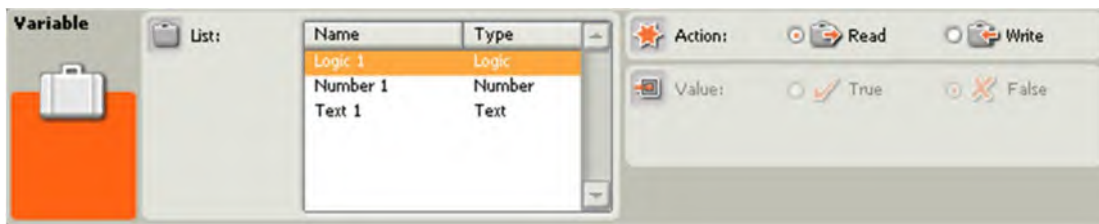


Μπλοκ μεταβλητής Variable Block

Μπορούμε να θεωρήσουμε ότι η μεταβλητή είναι ένα μέρος στο οποίο μπορούμε να αποθηκεύσουμε διάφορες τιμές οι οποίες τροποποιούνται διαρκώς. Το μέρος αυτό βρίσκεται μέσα στη μνήμη του μικροϋπολογιστή NXT.

Η τιμή μιας μεταβλητής μπορεί να μεταβάλλεται οποιαδήποτε στιγμή ως αποτέλεσμα της εξόδου άλλων εντολών, μέσω καλωδίωσης δεδομένων. Κάποιες άλλες εντολές μπορούν να διαβάσουν την τρέχουσα τιμή μιας μεταβλητής ως είσοδο τους, μέσω καλωδίωσης δεδομένων.

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων της εντολής αυτής επιλέγουμε τη μεταβλητή από μία λίστα (*List*) που επιθυμούμε να χρησιμοποιήσουμε λαμβάνοντας υπόψη το είδος των δεδομένων που θα δέχεται η μεταβλητή μας ανάμεσα κείμενο "*Text*", αριθμό "*Number*" ή λογικό αριθμό "*Logic*". Επίσης ορίζουμε και την ενέργεια (*Action*) που θα συμβεί: Αν θέλουμε να διαβάσουμε μία τιμή επιλέγουμε ανάγνωση (*Read*) ενώ αν θέλουμε να δώσουμε μία νέα τιμή στη μεταβλητή μας επιλέγουμε εγγραφή (*Write*). Στη δεύτερη περίπτωση μπορούμε να δώσουμε μία τιμή (*Value*) μέσα από τον πίνακα ρυθμίσεων ή εναλλακτικά, να περάσουμε την τιμή από το σημείο εισόδου # που εμφανίζεται στην αριστερή του πλευρά του κατανεμητή δεδομένων.

Αν θέλουμε να χρησιμοποιήσουμε περισσότερες από μία μεταβλητές για ένα είδος δεδομένων ή πρέπει να τις δηλώσουμε πρώτα μέσα από την επιλογή *Define Variables* του μενού *Edit* του προγραμματιστικού περιβάλλοντος NXT-G

Επεξήγηση συμβόλων

1. Το κείμενο κάτω αριστερά εμφανίζει το όνομα της μεταβλητής.
2. Η εικόνα κάτω δεξιά εμφανίζει την ενέργεια: ανάγνωση (*read*) ή εγγραφή (*write*).
3. Ο κατανεμητής δεδομένων (*data hub*) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.





Μπλοκ σταθερής τιμής Constant Block

Μία σταθερά τη χρειαζόμαστε όταν υπάρχει ανάγκη να χρησιμοποιήσουμε στο πρόγραμμα μας μία συγκεκριμένη σταθερή τιμή πάρα πολλές φορές.

Η τιμή μιας σταθεράς παραμένει η ίδια σε όλη τη διάρκεια που τρέχει το πρόγραμμα μας και μπορεί μόνο να διαβαστεί από τις άλλες εντολές ως είσοδο τους μέσω καλωδίωσης δεδομένων.

Πίνακας Ρυθμίσεων

Από τον πίνακα ρυθμίσεων της εντολής αυτής δίνουμε ένα συμβολικό όνομα (*Name*) για να ξεχωρίζουμε αυτή τη σταθερά από άλλες και καθορίζουμε το είδος των δεδομένων (*Data Type*) που θα δέχεται η σταθερά μας (κείμενο, αριθμός ή λογικός αριθμός) καθώς και την τιμή (*Value*) που θα έχει αυτή.

Αν χρησιμοποιούμε αρκετά συχνά στα προγράμματα που δημιουργούμε συγκεκριμένες σταθερές τιμές τότε μπορούμε να τις δηλώσουμε μέσα από την επιλογή *Define Constants* του μενού *Edit* του προγραμματιστικού περιβάλλοντος NXT-G. Σε αυτή την περίπτωση για να χρησιμοποιήσουμε κάποια από αυτές θα πρέπει να επιλέξουμε στη ρύθμιση της ενέργειας (*Action*) την παράμετρο της επιλογής από τη λίστα "*Choose from list*".



Μπλοκ για προχωρημένους Advanced Blocks



Μπλοκ κειμένου Text Block

Με αυτήν την εντολή μπορούμε να ομαδοποιήσουμε ένα αριθμό χαρακτήρων ως μία οντότητα κειμένου. Οι χαρακτήρες αυτοί μπορεί να είναι γράμματα, ειδικοί χαρακτήρες όπως σύμβολα και σημεία στίξης ή και αριθμοί. Το κείμενο αυτό στη συνέχεια μπορούμε να το στείλουμε να προβληθεί στην οθόνη του NXT.

Πίνακας Ρυθμίσεων



Η εντολή αυτή δέχεται ως ορίσματα τρεις ομάδες χαρακτήρων A, B και C. Τα ορίσματα αυτά μπορούν να δοθούν ως είσοδοι και μέσω καλωδίωσης από άλλες εντολές. Το σχηματιζόμενο κείμενο μπορεί να αξιοποιηθεί μέσω της καλωδίωσης εξόδου T.

Αν θέλουμε να σχηματίσουμε μία φράση πρέπει να προσθέσουμε και τα κενά στα σημεία που χρειάζεται. Για το σχηματισμό μεγάλων προτάσεων μπορούμε να χρησιμοποιήσουμε πολλές εντολές κειμένου στη σειρά.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Επεξήγηση συμβόλων

1. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.





Μπλοκ αριθμού σε κείμενο Number to Text Block

Οι αριθμοί που προκύπτουν από το μικροεπεξεργαστή NXT καθώς τρέχει ένα πρόγραμμα δεν μπορούν να αξιοποιηθούν από τις εντολές του προγράμματος που διαχειρίζονται κείμενο και έτσι πχ δεν μπορούν να προβληθούν απευθείας στην οθόνη του.

Με αυτήν την εντολή μπορούμε να μετατρέψουμε ένα αριθμό σε κείμενο το οποίο θα μπορεί να προβληθεί στην οθόνη του NXT.

Πίνακας Ρυθμίσεων



Τον αριθμό μπορούμε να τον πληκτρολογήσουμε στο αντίστοιχο πεδίο του πίνακα ρυθμίσεων ή να δοθεί ως είσοδος και μέσω καλωδίωσης από άλλες εντολές. Το σχηματιζόμενο κείμενο μπορεί να αξιοποιηθεί μέσω της καλωδίωσης εξόδου T.

Αν αποφασίσουμε να δώσουμε τον αριθμό συνδέοντας με καλώδιο το σημείο εισόδου # στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού (*Number*) στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Σε συνδυασμό με την εντολή κειμένου μπορούμε να δημιουργήσουμε προτάσεις όπως: “The motor turned 6 times” ή “The light sensor reading is 35%”

Επεξήγηση συμβόλων

1. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ επαγρύπνησης Keep Alive Block

Ο μικροεπεξεργαστής NXT όταν δεν χρησιμοποιηθεί για ένα συγκεκριμένο χρονικό διάστημα - το οποίο μπορούμε να ρυθμίσουμε μέσω του μενού *NXT settings* – περνάει στην κατάσταση αδράνειας (sleep mode) για λόγους εξοικονόμησης ενέργειας. Παρόλο αυτά πολλές φορές μέσα στο πρόγραμμα μας,

Σενάρια Ρομποτικής με το Mindstorms NXT

επιβάλλεται το NXT να περιμένει επιπλέον χρόνο. Στις περιπτώσεις αυτές χρησιμοποιώντας την εντολή αυτή, μπορούμε να παρατείνουμε το χρόνο πριν το τούβλο NXT περάσει στην κατάσταση αδράνειας,.

Επεξήγηση συμβόλων

1. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ πρόσβασης αρχείου File Access Block

Με αυτήν την εντολή μπορούμε να αποθηκεύσουμε διάφορα δεδομένα και πληροφορίες από το ρομπότ μας σε ένα αρχείο στη μνήμη του μικροεπεξεργαστή NXT.

Για να αξιοποιήσουμε κατάλληλα την εντολή αυτή, το όνομα του αρχείου και τα δεδομένα του αρχείου μπορούν να δοθούν δυναμικά μέσω των καλωδιώσεων δεδομένων εισόδου.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (data hub) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων δηλώνουμε ένα όνομα (*Name*) για το αρχείο.

Τα δεδομένα μπορεί να είναι της μορφής (*Type*) κειμένου "*Text*" ή αριθμοί "*Number*". Αφού αποθηκεύσουμε "*Write*" ένα αρχείο, χρειαζόμαστε μία άλλη εντολή πρόσβασης αρχείου για να διαβάσουμε "*Read*" τα δεδομένα του αρχείου αυτού. Πριν όμως διαβάσουμε ένα αρχείο πρέπει πρώτα να το κλείσουμε "*Close*" με μία ακόμη εντολή πρόσβασης αρχείου. Επίσης με την εντολή αυτή, μπορούμε να διαγράψουμε "*Delete*" ένα υπάρχον αρχείο.

Κάνοντας εγγραφή σε ένα ήδη υπάρχον αρχείο τα νέα δεδομένα θα προστεθούν στο τέλος αυτού του αρχείου.

Επεξήγηση συμβόλων

1. Η εικόνα κάτω στη μέση εμφανίζει τον τύπο των δεδομένων : κείμενο *''Text''* ή αριθμοί *''Number''*.
2. Η εικόνα κάτω δεξιά εμφανίζει την ενέργεια : ανάγνωση *''Read''*, εγγραφή *''Write''*, κλείσιμο *''Close''* ή διαγραφή *''Delete''*.
3. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



**Μπλοκ Ρεγουλαρίσματος
Calibration Block**

Με αυτή την εντολή μπορούμε να ρυθμίσουμε τη ελάχιστη *''Minimum''* (0%) και τη μέγιστη *''Maximum''* (100%) τιμή (*Value*) σε έναν αισθητήρα ήχου ή φωτός ώστε να προσαρμόσουμε τη μέτρηση που θα λαμβάνουν στο συγκεκριμένο περιβάλλον που θα τους χρησιμοποιήσουμε.

Χρειαζόμαστε δύο εντολές ρεγουλαρίσματος για να γίνει πλήρες καλιμπράρισμα :

Μία για να ρυθμίσουμε την ελάχιστη τιμή που θα μετριέται ως 0% και μία για τη μέγιστη τιμή ως 100%.

Ένας τρόπος για να ρεγουλάρουμε πλήρως έναν αισθητήρα ήχου ή φωτός είναι να χρησιμοποιήσουμε δύο εντολές ρεγουλαρίσματος (μία για τη ελάχιστη και μία για τη μέγιστη τιμή) με αντίστοιχες εντολές αναμονής από αισθητήρας αφής μετά από κάθε μία από αυτές, στην αρχή κάποιου προγράμματος. Με αυτό τον τρόπο μόλις τρέξουμε το πρόγραμμα μας (αν πχ πρόκειται για αισθητήρα φωτός), τοποθετούμε το ρομπότ μας πρώτα στο πιο σκοτεινό σημείο αν πρόκειται να προσδιορίσουμε την ελάχιστη τιμή, και αφού πατήσουμε το διακόπτη του αισθητήρα αφής τοποθετούμε το ρομπότ μας στη συνέχεια στο πιο φωτεινό σημείο και ξαναπατάμε το διακόπτη του αισθητήρα αφής για να συνεχιστεί η εκτέλεση του προγράμματος μας.

Ένας άλλος τρόπος (πχ αν πάλι πρόκειται για αισθητήρα φωτός) είναι να τοποθετήσουμε μία εντολή ρεγουλαρίσματος σε συγκεκριμένο σημείο του προγράμματος, αν γνωρίζουμε ότι τη στιγμή εκείνη το ρομπότ μας θα βρίσκεται στο πιο σκοτεινό ή αντίστοιχα φωτεινό σημείο.

Σημείωση : Το καλιμπράρισμα πραγματοποιείται ανά αισθητήρα και δε παίζει ρόλο η θύρα εισόδου στην οποία είναι συνδεδεμένος ένας αισθητήρας. Αφού καλιμπράρουμε πχ έναν αισθητήρα φωτός που είναι συνδεδεμένος στη θύρα εισόδου 3 οι ελάχιστη και η μέγιστη τιμή για τις οποίες ρυθμίσαμε τον παραπάνω αισθητήρα θα ισχύουν και για όλους τους άλλους αισθητήρες φωτός που πιθανόν να είναι συνδεδεμένοι στο ρομπότ μας, ανεξαρτήτως της θύρα εισόδου που είναι συνδεδεμένοι.

Σημαντικό : Για να καλιμπράρουμε τους αισθητήρες ήχου και φωτός πριν από την εκτέλεση του προγράμματος μας, μπορούμε επίσης να χρησιμοποιήσουμε και την επιλογή *Calibrate Sensors* από το μενού *Tools* του προγραμματιστικού περιβάλλοντος NXT-G.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων δηλώνουμε τη θύρα (*Port*) που είναι συνδεδεμένος ο αισθητήρας μας και το είδος του αισθητήρα (*Sensor*) που θέλουμε να καλιμπράρουμε.

Επίσης καθορίζουμε αν πρόκειται να προσδιορίσουμε την ελάχιστη "*Minimum*" ή τη μέγιστη "*Maximum*" τιμή (*Value*).

Αν θέλουμε αντί για ρεγουλάρισμα απλώς να ακυρώσουμε τις τιμές με τις οποίες είναι ήδη ρεγουλαρισμένος ο αισθητήρας, στη ρύθμιση της ενέργειας (*Action*) επιλέγουμε αντί για την παράμετρο "*Calibrate*" τη "*Delete*".

Επεξήγηση συμβόλων

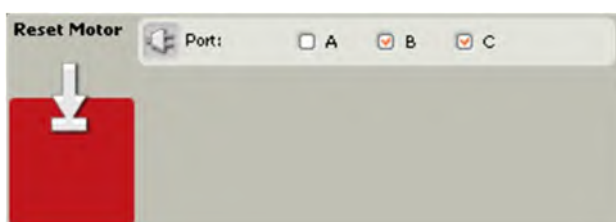
1. Ο αριθμός πάνω δεξιά δηλώνει τη θύρα που είναι συνδεδεμένος ο αισθητήρας μας.
2. Η εικόνα κάτω στη μέση δηλώνει το είδος του αισθητήρα : φωτός ή ήχου.
3. Η εικόνα κάτω δεξιά δηλώνει την τιμή που θα ρεγουλαριστεί : ελάχιστη " ή μέγιστη
4. Ο κατανεμητής δεδομένων (data hub) ανοίγει αυτόματα μόλις τοποθετήσουμε την εντολή στην περιοχή εργασίας.



Μπλοκ ακύρωσης αυτόματου μηχανισμού κινητήρα Reset Motor Block

Οι κινητήρες που συνοδεύουν το σετ NXT της Lego είναι διαδραστικοί και έχουν έναν αυτόματο μηχανισμό συγχρονισμού μεταξύ τους ο οποίος βοηθάει το ρομπότ να κινείται με πολύ μεγάλη ακρίβεια. Παρόλο αυτά μπορεί σε κάποιες δραστηριότητες το συγκεκριμένο χαρακτηριστικό να μη είναι βολικό. Για τις περιπτώσεις αυτές μπορούμε να ακυρώσουμε προσωρινά τον αυτόματο μηχανισμό συγχρονισμού των κινητήρων χρησιμοποιώντας την εντολή αυτή.

Πίνακας Ρυθμίσεων



Στον πίνακα ρυθμίσεων επιλέγουμε τους κινητήρες που θέλουμε να ελέγξουμε (*Port*).

Επεξήγηση συμβόλων

1. Τα γράμματα πάνω δεξιά δηλώνουν τις θύρες εξόδου στις οποίες θέλουμε να ελέγξουμε τους κινητήρες.



Start Datalog

Αυτή την εντολή τη χρησιμοποιούμε για να συλλέξουμε μία σειρά δεδομένων μετά από δειγματοληψία των σημάτων που ανιχνεύονται από τους διάφορους αισθητήρες.

Τα δεδομένα αυτά αποθηκεύονται με την ολοκλήρωση της διαδικασίας, ως ένα αρχείο κειμένου με το όνομα που έχουμε καθορίσει.

Αν αποφασίσουμε να δώσουμε τιμή σε κάποιο όρισμα, συνδέοντας με καλώδιο το σημείο εισόδου του στον κατανεμητή δεδομένων (*data hub*) τότε η αντίστοιχη περιοχή του ορίσματος αυτού στον πίνακα ρυθμίσεων, απενεργοποιείται αυτόματα.

Πίνακας Ρυθμίσεων

Start Datalog			
Name:	myData	None	Port: 1
Duration:	10 Seconds	None	Port: 2
Rate:	5 Samples per Second	None	Port: 3
Wait:	☒ Wait for Completion	None	Port: 4

Από τον πίνακα ρυθμίσεων δηλώνουμε ένα όνομα (*Name*) για το αρχείο.

Στη συνέχεια ορίζουμε τη διάρκεια (*Duration*) της δειγματοληψίας την οποία μπορούμε να τη ρυθμίσουμε να πραγματοποιηθεί για ένα συγκεκριμένο χρονικό σε δευτερόλεπτα "*Seconds*" ή λεπτά "*Minutes*", ή συνεχόμενα "*Unlimited*" ή ακόμα και για ένα μόνο δείγμα. "*Sample Measurement*".

Στην περίπτωση που καθορίσουμε τη διάρκεια της δειγματοληψίας σε συνεχόμενη "*Unlimited*" το πρόγραμμα θα ξεκινήσει τη δειγματοληψία και θα περάσει αμέσως στην εκτέλεση της επόμενης εντολής. Στην περίπτωση αυτή πρέπει να χρησιμοποιήσουμε την εντολή της διακοπής δειγματοληψίας αργότερα μέσα στο πρόγραμμα μας, για να τερματίσουμε τη διαδικασία αυτή.

Στην περίπτωση που καθορίσουμε η δειγματοληψία μας να λάβει μόνο ένα στιγμιότυπο ενός σήματος, τότε το πρόγραμμα θα περιμένει να ολοκληρωθεί η διαδικασία αυτή, προτού περάσει στην επόμενη εντολή.

Σενάρια Ρομποτικής με το Mindstorms NXT

Επίσης πρέπει να καθορίσουμε το ρυθμό Rate με τον οποίο θα πραγματοποιείται η λήψη των δειγμάτων σε κάποιον αριθμό δειγμάτων ανά δευτερόλεπτο *“Samples per Seconds”* ή σε κάποιον αριθμό δευτερόλεπτων ανάμεσα στα δείγματα *“Seconds between Samples”*.

Τέλος δηλώνουμε τους αισθητήρες μέσω των οποίων θα πραγματοποιηθεί η δειγματοληψία (ως 4 ταυτόχρονα) και τις αντίστοιχες θύρες εισόδου που είναι αυτοί συνδεδεμένοι. Για την περίπτωση του αισθητήρα φωτός μπορούμε να επιλέξουμε αν θα είναι ενεργοποιημένη ή εκπομπή υπέρυθρης ακτινοβολίας ή όχι.

Αν ενεργοποιήσουμε την παράμετρο της αναμονής για ολοκλήρωση *“Wait for Completion”* της ρύθμισης της αναμονής (*Wait*) το πρόγραμμα θα περάσει στην εκτέλεση της επόμενης εντολής, αφού πρώτα ολοκληρωθεί η δειγματοληψία που έχουμε ορίσει. Με αυτήν την επιλογή ανενεργή η δειγματοληψία θα ξεκινήσει και θα συνεχίσει να πραγματοποιείται ενώ θα εκτελείται ταυτόχρονα η επόμενη εντολή του προγράμματος μας.

Επεξήγηση συμβόλων

1. Η εικόνα κάτω στη μέση εμφανίζει το όνομα του αρχείου στο οποίο αποθηκεύονται τα δεδομένα μας.
2. Η εικόνα κάτω δεξιά εμφανίζει τη διάρκεια της δειγματοληψίας : δευτερόλεπτα *“Seconds”* ή λεπτά *“Minutes”*, ή συνεχόμενα *“Unlimited”* ή ακόμα και για ένα μόνο δείγμα. *“Sample Measurement”*.



Stop Datalog

Αυτή την εντολή τη χρησιμοποιούμε για να διακόψουμε μία διαδικασία δειγματοληψίας την οποία είχαμε ξεκινήσει πιο μπροστά..

Πίνακας Ρυθμίσεων



Από τον πίνακα ρυθμίσεων δηλώνουμε το όνομα (*Name*) του αρχείου στο οποίο συλλέγονται τα δεδομένα της δειγματοληψίας που θέλουμε να διακόψουμε. Αυτόματα γίνεται και η αποθήκευση των δεδομένων μας.

Επεξήγηση συμβόλων

1. Η εικόνα κάτω στη μέση εμφανίζει το όνομα του.



Προσαρμοσμένη παλέτα (Custom Palette)



Το δικό μου μπλοκ My Block

Πολλές φορές είναι χρήσιμο να δημιουργήσουμε ένα δικό μας μπλοκ εντολής για μία σειρά εντολών που χρησιμοποιούμε συχνά ή θέλουμε να ομαδοποιήσουμε.

Για να το πετύχουμε αυτό αφού δημιουργήσουμε ένα τμήμα προγράμματος στην περιοχή εργασίας, το επιλέγουμε και στη συνέχεια επιλέγουμε *Make a new my block* από το μενού *Edit* του προγραμματιστικού περιβάλλοντος NXT-G. Στο παράθυρο που εμφανίζεται δίνουμε ένα όνομα (και μία περιγραφή προαιρετικά) και ορίζουμε ένα δικό εικονίδιο για αυτό το μπλοκ μας.

Σε αυτή την εντολή βρίσκονται ομαδοποιημένα τα δικά μας μπλοκ, τα οποία μπορούμε να τα χρησιμοποιήσουμε μέσα σε ένα νέο πρόγραμμα όπως και οποιαδήποτε άλλα μπλοκ.

Διαφορές ανάμεσα στις εντολές Move και Motor

Στο προγραμματιστικό περιβάλλον NXT-G, όπως ήδη είπαμε, υπάρχουν δύο εντολές με τις οποίες μπορούμε να δώσουμε κίνηση τη ρομποτική μας κατασκευή :



Οι εντολές **Move**



και **Motor**.

Η εντολή Motor έχει στο εικονίδιο της ένα μεγάλο γρανάζι και μία ανοικτή πράσινη λουρίδα και βρίσκεται στην υποπαλέτα ενεργειών (Action Palette).

Η εντολή Move έχει στο εικονίδιο της δύο γρανάζια και μία σκούρη πράσινη λουρίδα και βρίσκεται στη βασική υποπαλέτα (Basic Palette).

Αν και η λειτουργία τους φαίνεται παρόμοια, παρόλο αυτά υπάρχουν σημαντικές διαφορές ως προς τη συμπεριφορά στην κίνηση. Η πιο μεγάλη τους διαφορά βρίσκεται στο επίπεδο ευφυΐας που ενσωματώνει κάθε μία από αυτές τις εντολές :

Η εντολή Motor έχει σχεδιαστεί ώστε να ξεκινάει και να σταματάει κινητήρες. Μέσα από τον πίνακα ρυθμίσεων της μπορείς να καθορίσεις τη διάρκεια της κίνησης, αλλά η εντολή motor δε μας δίνει τη δυνατότητα να κινήσουμε το ρομπότ μας με μεγάλη ακρίβεια. Έτσι αν χρησιμοποιήσουμε την εντολή αυτή για να βάλουμε το ρομπότ μας να κινηθεί για συγκεκριμένη απόσταση και να σταματάει σε ένα συγκεκριμένο σημείο, το ρομπότ μας δε θα σταματήσει στο σημείο που επιθυμούμε αλλά θα το προσπεράσει και στη συνέχεια θα κάνει τη διόρθωση που χρειάζεται πηγαίνοντας λίγο πίσω. Αυτό συμβαίνει επειδή η εντολή αυτή δε μειώνει σταδιακά την ταχύτητα των κινητήρων λίγο πριν φτάσει στο επιθυμητό σημείο ώστε το ρομπότ μας να σταματήσει με ακρίβεια στο σημείο αυτό, αλλά αφού φτάσει στο επιθυμητό σημείο, και στη συνέχεια προσπαθεί να σταματήσει ακαριαία τους κινητήρες πράγμα που είναι ανέφικτο.



Η εντολή Move, από την άλλη, μπορεί να μας δώσει πλήρη έλεγχο στην κίνηση του ρομπότ μας. Η εντολή αυτή κατά τη διάρκεια της κίνησης του ρομπότ μας παρακολουθεί τους ενσωματωμένους στους κινητήρες, αισθητήρες περιστροφής. Με τον τρόπο αυτό γνωρίζει ανά πάσα στιγμή τη θέση του ρομπότ μας και ελέγχει κατάλληλα τους κινητήρες ώστε να ακολουθηθεί η επιθυμητή διαδρομή που έχουμε ορίσει. Επίσης με τον ίδιο τρόπο, φροντίζει ώστε και οι δύο κινητήρες στο ρομπότ μας να περιστρέφονται συγχρονισμένα έτσι ώστε αν κάποια στιγμή το ρομπότ μας αρχίσει να αποκλίνει από

Σενάρια Ρομποτικής με το Mindstorms NXT

την καθορισμένη πορεία του με την εντολή αυτή το πρόγραμμα φροντίζει να το επαναφέρει στην αρχική του πορεία.



Η εντολή Motor είναι πολύ απλή. Μία εντολή τέτοια ελέγχει μόνο έναν κινητήρα. Επομένως για να κάνουμε ένα πρόγραμμα για ένα τυπικό ρομπότ με δύο κινητήρες χρειαζόμαστε πάνω από μία τέτοια εντολή με αποτέλεσμα το πρόγραμμα μας να δεσμεύει περισσότερο χώρο στη μνήμη του NXT από ότι αν χρησιμοποιήσουμε τη εντολή Move. Επιπρόσθετα, αν επιλέξουμε στην εντολή Motor η διάρκεια της κίνησης να υλοποιηθεί με αριθμό περιστροφών ή μοιρών περιστροφής, τότε θα πρέπει ενδιάμεσα από κάθε χρήση να χρησιμοποιούμε και την εντολή του αισθητήρα περιστροφής με την κίτρινη λουρίδα (Rotation Sensor) για να μηδενίζουμε το μετρητή.

Περιπτώσεις χρήσης

- Αν θέλουμε το ρομπότ μας να αρχίσει να κινείται σε μία ευθεία ή να ακολουθήσει ακριβώς μία καθορισμένη διαδρομή ή να κινηθεί για μία καθορισμένη απόσταση τότε θα χρησιμοποιήσουμε την εντολή Move, η οποία θα αναλάβει την εκπλήρωση της αποστολής με βολικό τρόπο. Στην περίπτωση αυτή μάλιστα, αν πρόκειται και για ένα μεγάλο και αρκετά πολύπλοκο πρόγραμμα πετυχαίνουμε και σημαντική εξοικονόμηση στο χώρο της μνήμης του NXT.
- Αν επιθυμούμε να χρησιμοποιήσουμε το πακέτο NXT για εκπαιδευτικούς σκοπούς τότε ο πιο κατάλληλος τρόπος για να μυήσουμε τους μαθητές μας είναι να κάνουμε επίδειξη και χρήση μόνο της εντολής motor μέχρι να γίνει απολύτως κατανοητός ο τρόπος με τον οποίο ελέγχουμε την κίνηση ενός τυπικού ρομπότ με δύο κινητήρες. Για να αποφευχθεί η περίπτωση σύγχυσης στους μαθητές κρίνεται σωστότερο η διδασκαλία της εντολής move να ακολουθήσει αφού οι μαθητές εξοικειωθούν σε μεγάλο βαθμό με τη λογική της απόδοσης της επιθυμητής συμπεριφοράς στο ρομπότ μας.
- Αν κάποιο πρόγραμμα στο οποίο έχουμε χρησιμοποιήσαμε την εντολή move συμπεριφέρεται απρόσμενα, τότε μπορούμε να εντοπίσουμε πιο εύκολα την αιτία για την παράξενη αυτή συμπεριφορά, ξαναγράφοντας το προβληματικό μέρος του προγράμματος χρησιμοποιώντας την εντολή Motor. Με τη χρήση της εντολής Motor μπορούμε να κατανοήσουμε τη συμπεριφορά που αποκτάει το ρομπότ μας με μεγαλύτερη ευκολία.

ΠΑΡΑΡΤΗΜΑ II

Παρουσίαση των εντολών του Robolab 2.9



Ένα πρόγραμμα που δημιουργούμε με το λογισμικό Robolab είναι μία σειρά από εντολές, οι οποίες καθορίζουν πως θα συμπεριφερθεί το ρομπότ μας στα συμβάντα - γεγονότα που θα συναντήσει μέσα στο περιβάλλον που δραστηριοποιείται. Αναλυτικότερα, το ρομπότ δέχεται πληροφορίες για το περιβάλλον γύρω του μέσω των αισθητήρων (*sensors*). Στη συνέχεια ανάλογα με το πρόγραμμα που έχουμε δημιουργήσει και του έχουμε δώσει να εκτελέσει (το έχουμε κατεβάσει – *download*) το ρομπότ μας "παίρνει αποφάσεις" για το πώς πρόκειται να κινηθεί. Αισθάνεται και αντιδρά περίπου όπως και εμείς.

Οι αισθητήρες εφοδιάζουν το μικροεπεξεργαστή (τον ονομάζουμε και εγκέφαλο) NXT με πληροφορίες για το κοντινό του περιβάλλον. Ένας αισθητήρας αφής (touch sensor) ενημερώνει το NXT για την κατάσταση του διακόπτη του (αν είναι πατημένος ή όχι), πληροφορώντας το π.χ. για το αν έχει πέσει ή όχι πάνω σε ένα εμπόδιο. Ο αισθητήρας φωτός (light sensor) στέλνει στο NXT έναν αριθμό ανά χρονική στιγμή ο οποίος αντιστοιχεί στην ένταση του φωτός τη στιγμή εκείνη. Ο αισθητήρας ήχου (sound sensor) στέλνει στο NXT έναν αριθμό ο οποίος αντιστοιχεί στην ένταση του ήχου τη στιγμή εκείνη. Ο αισθητήρας υπερήχων (ultrasonic sensor) στέλνει στο NXT έναν αριθμό ο οποίος αντιστοιχεί στην απόσταση από ένα αντικείμενο τη στιγμή εκείνη. Ο αισθητήρας περιστροφής (rotation sensor) στέλνει στο NXT έναν αριθμό ο οποίος αντιστοιχεί στην περιστροφή ενός άξονα τη στιγμή εκείνη. Ο αισθητήρας θερμοκρασίας (temperature sensor) στέλνει στο NXT έναν αριθμό ο οποίος αντιστοιχεί στη θερμοκρασία που ανιχνεύει τη στιγμή εκείνη. Ένας άλλος αισθητήρας, ο χρονομετρητής (timer), - ο οποίος είναι ενσωματωμένος στα κυκλώματα του NXT - μπορεί να πληροφορήσει τον NXT για το χρονικό διάστημα που έχει περάσει. Τα προγράμματα μπορούν να διαβάσουν τις τιμές που λαμβάνουν από τους αισθητήρες και έτσι να "πάρουν μια απόφαση" για τι πρέπει να κάνουν. Με αυτό τον τρόπο κατορθώνουμε το ρομπότ μας, να αποκτήσει την επιθυμητή συμπεριφορά.

Ένα πρόγραμμα στο Robolab δημιουργείται από έναν αριθμό εικονιδίων (command boxes) τα οποία ενώνουμε μεταξύ τους με μία γραμμή, η οποία αναπαριστά ένα καλώδιο (wire). Καθένα από τα εικονίδια αναπαριστούν μία απλή εντολή, και οι εντολές εκτελούνται με τη σειρά με την οποία είναι καλωδιωμένες, συνιστώντας μια ακολουθία εντολών, η οποία στη συνέχεια εκτελείται από το ρομπότ μας.

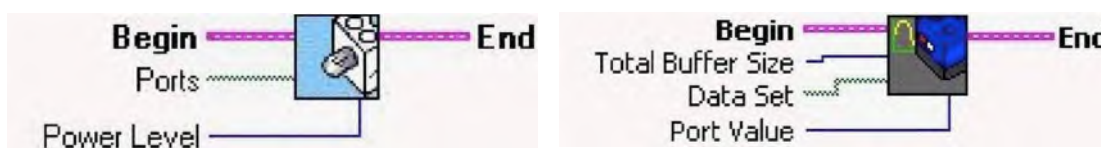
Ένα πρόγραμμα διαβάζεται από αριστερά προς τα δεξιά και είναι μία **ακολουθία εικονιδίων**.



Κάθε εικονίδιο έχει καλωδιώσεις με τις οποίες ενώνεται με το προηγούμενο και το επόμενο.

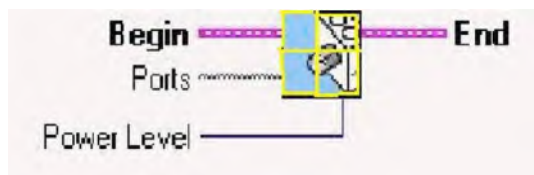


Μερικά εικονίδια έχουν περισσότερες από δύο καλωδιώσεις σύνδεσης. Στο πάνω μέρος κάθε εικονιδίου, στην αριστερή και τη δεξιά γωνία του αντίστοιχα, υπάρχουν οι θέσεις Begin και End που βοηθούν τη συναρμολόγηση – ένωση με προηγούμενα και επόμενα εικονίδια. Οι υπόλοιπες καλωδιώσεις χρειάζονται για δηλώσεις παραμέτρων όπως θύρα σύνδεσης, ισχύς λειτουργίας, είδος δεδομένων, τιμή θύρας.



Σενάρια Ρομποτικής με το Mindstorms NXT

Το κάθε εικονίδιο χωρίζεται σε περιοχές ανάλογα με τις καλωδιώσεις που διαθέτει. Το επόμενο εικονίδιο που αντιπροσωπεύει ένα λαμπτήρα έχει τέσσερις καλωδιώσεις και γι' αυτό χωρίζεται σε τέσσερις περιοχές. Από κάθε περιοχή ξεκινά μία καλωδίωση.



Παρακάτω παραθέτουμε μία σύντομη περιγραφή των βασικότερων εντολών που είναι διαθέσιμες στο προγραμματιστικό περιβάλλον Robolab :

(Για περισσότερες πληροφορίες για κάθε μία εντολή ξεχωριστά, το λογισμικό Robolab περιλαμβάνει ένα ενσωματωμένο μενού βοήθειας. Για να το προσπελάσουμε ανοίγουμε το Robolab στον τρόπο προγραμματισμού *Inventor* και επιλέγουμε το *show context help* από το μενού *help*.)

Εντολές αρχής και τέλους



Εκκίνηση Προγράμματος (Begin Program)

Εκκίνηση ενός προγράμματος. Είναι απαραίτητη ως πρώτη εντολή σε οποιοδήποτε πρόγραμμα του προγραμματιστικού περιβάλλοντος Robolab



Τέλος Προγράμματος (End Program)

Τερματισμός ενός προγράμματος. Είναι απαραίτητη ως τελευταία εντολή σε οποιοδήποτε πρόγραμμα του προγραμματιστικού περιβάλλοντος Robolab

Παράδειγμα





Εντολές εξόδου (Output Commands)



Λαμπτήρας A/B/C Μπροστά (Lamp A/B/C)

Αυτή η εντολή λέει στο NXT να ανάψει το λαμπτήρα στη θύρα εξόδου A ή B ή C.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) έντασης για να ελέγξουμε πόσο έντονα θα φέγγει ο λαμπτήρας. Αν δεν προσθέσουμε τροποποιητή έντασης, ο λαμπτήρας μας θα ανάψει στη μέγιστη ένταση.

Παράδειγμα



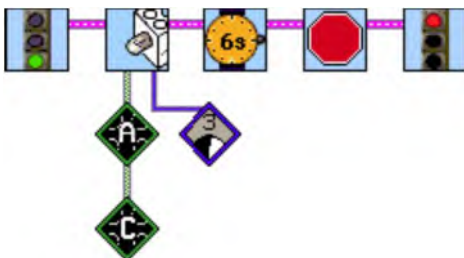
Λαμπτήρας A/B/C Μπροστά (Lamp A/B/C)

Αυτή η εντολή λέει στο NXT να ανάψει το λαμπτήρα στη θύρα εξόδου που ορίζεται από τον τροποποιητή της θύρα εξόδου.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα τροφοδοτηθούν με ρεύμα και οι τρεις θύρες εξόδου.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) έντασης για να ελέγξουμε πόσο έντονα θα φέγγει ο λαμπτήρας. Αν δεν προσθέσουμε τροποποιητή έντασης, ο λαμπτήρας μας θα ανάψει στη μέγιστη ένταση.

Παράδειγμα





Κίνηση Κινητήρα A/B/C Μπροστά (Motor A/B/C Forward)

Αυτή η εντολή λέει στο NXT να ξεκινήσει τον κινητήρα στη θύρα εξόδου A ή B ή C προς τα μπροστά. Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.

Παράδειγμα



Σημείωση : Η κίνηση του κινητήρα προς τα μπροστά δεν σημαίνει ότι και το όχημα-ρομπότ μας θα κινηθεί οπωσδήποτε προς τα μπροστά. Αυτό εξαρτάται από τον τρόπο που έχουμε συνδέσει τον κινητήρα. Χρειάζεται λοιπόν πρώτα να πειραματιστούμε και να κάνουμε την κατάλληλη σύνδεση του ακροδέκτη στον κινητήρα ώστε και το όχημα μας να κινείται προς τα μπροστά. (Αν δε συμβαίνει αυτό αλλάζουμε τη φορά του ακροδέκτη στον κινητήρα ή στο NXT).



Κίνηση Κινητήρα A/B/C Πίσω (Motor A/B/C Backwards)

Αυτή η εντολή λέει στο NXT ξεκινήσει τον κινητήρα A ή B ή C προς την ανάποδη κατεύθυνση. Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.

Παράδειγμα



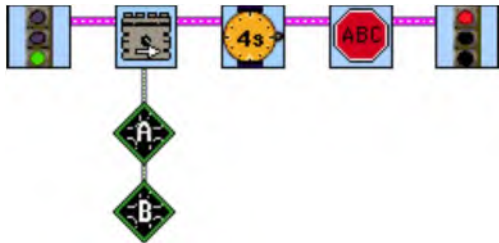
Κίνηση Κινητήρα Μπροστά (Motor Forward)

Αυτή η εντολή λέει στο NXT ξεκινήσει τον κινητήρα που ορίζεται από τον τροποποιητή της θύρας εξόδου, προς τα μπροστά.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα δοθεί σήμα για εκκίνηση (θα τροφοδοτηθούν δηλαδή με ρεύμα) και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.

Παράδειγμα



Κίνηση Κινητήρα Πίσω (Motor Backwards)

Αυτή η εντολή λέει στο NXT ξεκινήσει τον κινητήρα που ορίζεται από τον τροποποιητή της θύρα εξόδου προς την αντίθετη κατεύθυνση.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα δοθεί σήμα για εκκίνηση (θα τροφοδοτηθούν δηλαδή με ρεύμα) και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.



Αλλαγή κατεύθυνσης (Flip direction)

Αυτή η εντολή λέει στο NXT αλλάξει την κατεύθυνση κίνησης ενός κινητήρας στην έξοδο που ορίζεται από τον τροποποιητή της θύρα εξόδου.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε αλλάξει η κατεύθυνση κίνησης για όλους τους συνδεδεμένους κινητήρες.

Παράδειγμα





Σταμάτημα εξόδου A/B/C (Stop A/B/C)

Αυτή η εντολή λέει στο NXT να διακόψει την τροφοδοσία με ρεύμα στην έξοδο A ή B ή C. Χρειάζεται για το σταμάτημα των κινητήρων και των λαμπτήρων.

Αν έχουμε συνδεδεμένο κάποιο κινητήρα στην αντίστοιχη έξοδο αυτός θα σταματήσει να γυρίζει ακαριαία (δηλ θα φρενάρει).

Αν δεν τοποθετήσουμε αυτή την εντολή πριν από το τέλος ενός προγράμματος, τότε ο αντίστοιχος κινητήρας θα συνεχίσει να γυρίζει και μετά το πέρας του προγράμματος μας.



Σταμάτημα ABC (Stop ABC)

Αυτή η εντολή λέει στο NXT να διακόψει την τροφοδοσία με ρεύμα και στις τρεις εξόδους A, B και C.

Τοποθετείται συνήθως πάντοτε πριν από το τέλος ενός προγράμματος για να φρενάρει όλους τους συνδεδεμένους κινητήρες και λαμπτήρες.



Σταμάτημα (Stop)

Αυτή η εντολή λέει στο NXT να διακόψει την τροφοδοσία με ρεύμα στην έξοδο που ορίζεται από τον τροποποιητή της θύρα εξόδου. Χρειάζεται για το φρενάρισμα των κινητήρων και των λαμπτήρων.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα διακοπεί η τροφοδοσία ρεύματος και στις τρεις εξόδους.

Παράδειγμα



Σταμάτημα με ολίσθηση (Float Outputs)

Αυτή η εντολή λέει στο NXT να διακόψει την τροφοδοσία με ρεύμα στην έξοδο που ορίζεται από τον τροποποιητή της θύρα εξόδου. Σε αντίθεση με την εντολή "σταμάτημα", η εντολή αυτή δε φρενάρει τους κινητήρες, με αποτέλεσμα αυτοί να σταματάνε βαθμιαία.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα διακοπεί η τροφοδοσία ρεύματος και στις τρεις εξόδους.

Παράδειγμα





Παίξιμο Ήχου (Play Sound)

Με την εντολή αυτή το NXT θα παίξει ένα ήχο.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής από το 1 έως το 6.

Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 1.

Ο ήχος που θα αναπαραχθεί εξαρτάται από το αριθμό που έχουμε προσθέσει στον τροποποιητή σταθερής αριθμητικής τιμής ως εξής :

1: Key-click, 2: Beep Beep, 3: Descending Sweep, 4: Rising Sweep, 5: Buzz, 6: Fast Rising Sweep.

Συμβουλή :

Χρησιμοποιείτε τους ήχους για να κάνετε ανίχνευση λαθών και έλεγχο της σωστής λειτουργίας του προγράμματος σας.



Προχωρημένες εντολές εξόδου (Advanced Output Commands)

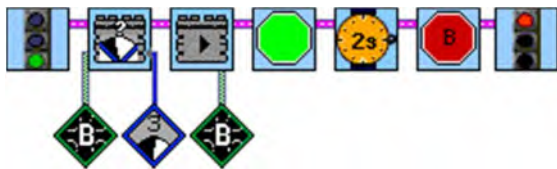


Εφαρμογή ισχύος σε Έξοδο (Turn Output On)

Αυτή η εντολή λέει στο NXT να τροφοδοτήσει με ισχύ την έξοδο που ορίζεται από τον τροποποιητή της θύρα εξόδου. Το επίπεδο ισχύος και η κατεύθυνση μιας εξόδου είτε έχουν οριστεί πιο μπροστά με μία σχετική εντολή, είτε παραμένουν όπως ήταν καθορισμένοι από πριν.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου Α ή Β ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα εφαρμοστεί η τροφοδοσία ρεύματος και στις τρεις εξόδους.

Παράδειγμα



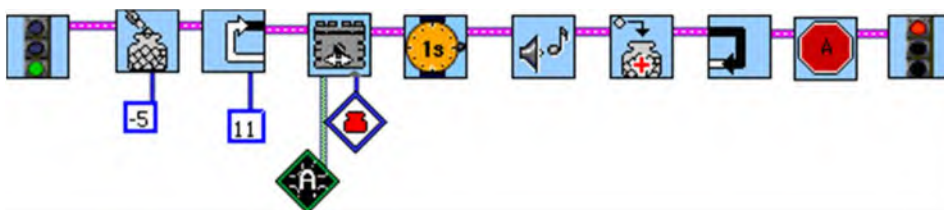
Κίνηση Κινητήρα μπροστά ή πίσω (Motor Forward or Back)

Αυτή η εντολή λέει στο NXT να ξεκινήσει τον κινητήρα προς τα μπροστά ή προς τα πίσω, ανάλογα με την τιμή ενός τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου Α ή Β ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα δοθεί σήμα για εκκίνηση (θα τροφοδοτηθούν δηλαδή με ρεύμα) και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε μία τιμή από το -5 έως το 5 με έναν τροποποιητή σταθερής αριθμητικής τιμής για να καθορίσουμε τη φορά και την ταχύτητα περιστροφής. Αν δεν προσθέσουμε τροποποιητή σταθερής τιμής, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα προς τα μπροστά.

Παράδειγμα





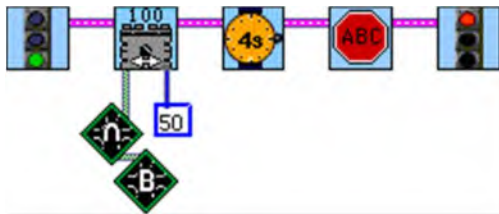
Κίνηση Κινητήρα με ισχύ (Motor Power)

Αυτή η εντολή λέει στο NXT να ξεκινήσει τον κινητήρα προς τα μπροστά ή προς τα πίσω, ανάλογα με την τιμή ενός τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα δοθεί σήμα για εκκίνηση (θα τροφοδοτηθούν δηλαδή με ρεύμα) και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε μία τιμή από το -100 έως το 100 με έναν τροποποιητή σταθερής αριθμητικής τιμής για να καθορίσουμε τη φορά και την ταχύτητα περιστροφής. Αν δεν προσθέσουμε τροποποιητή σταθερής τιμής, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα προς τα μπροστά.

Παράδειγμα



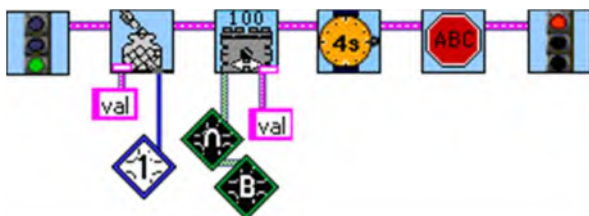
Κίνηση Κινητήρα με ισχύ (Motor Power)

Αυτή η εντολή λέει στο NXT να ξεκινήσει τον κινητήρα προς τα μπροστά ή προς τα πίσω, ανάλογα με την τιμή ενός ονόματος μεταβλητής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα δοθεί σήμα για εκκίνηση (θα τροφοδοτηθούν δηλαδή με ρεύμα) και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε μία τιμή από -100 έως 100 με ένα τροποποιητή ονόματος μεταβλητής για να καθορίσουμε τη φορά και την ταχύτητα περιστροφής.

Παράδειγμα





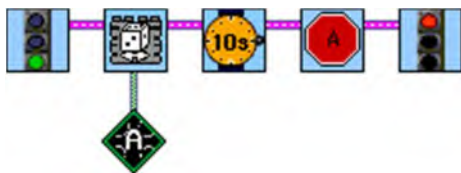
Κίνηση Κινητήρα με τυχαία κατεύθυνση (Motor Random)

Αυτή η εντολή λέει στο NXT να ξεκινήσει έναν κινητήρα προς μία τυχαία κατεύθυνση.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα εφαρμοστεί η τροφοδοσία ρεύματος και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.

Παράδειγμα



Καθορισμός ισχύος κινητήρα (Change Motor Speed)

Αυτή η εντολή λέει στο NXT να θέσει την ισχύ ενός κινητήρα σε μία συγκεκριμένη τιμή. Η εντολή αυτή δεν ξεκινάει κανένα κινητήρα. Πρέπει να χρησιμοποιηθεί σε συνδυασμό με την εντολή εφαρμογής ισχύος εξόδου ή ενώ ήδη ένας κινητήρας είναι σε λειτουργία.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα εφαρμοστεί η τροφοδοσία ρεύματος και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.

Παράδειγμα





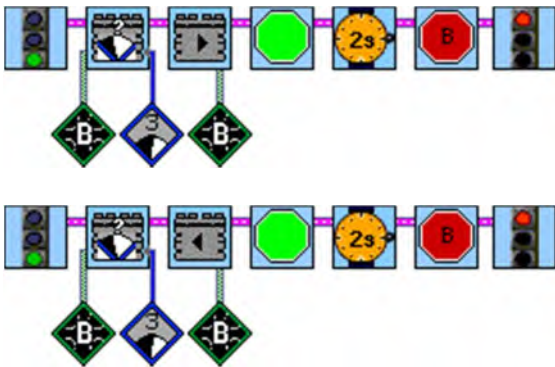
Καθορισμός κατεύθυνσης κίνησης Μπρος / Πίσω (Forward / Reverse)

Αυτή η εντολή λέει στο NXT να θέσει την κατεύθυνση ενός κινητήρα προς τα μπροστά / πίσω. Η εντολή αυτή δεν ξεκινάει κανένα κινητήρα. Πρέπει να χρησιμοποιηθεί σε συνδυασμό με την εντολή εφαρμογής ισχύος εξόδου ή ενώ ήδη ένας κινητήρας είναι σε λειτουργία. Αν ο κινητήρας γυρίζει ήδη προς την ίδια κατεύθυνση η εντολή αυτή δε θα έχει καμία επίπτωση.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα εφαρμοστεί η τροφοδοσία ρεύματος και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.

Παραδείγματα



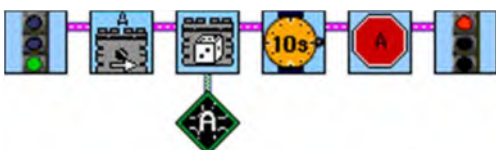
Καθορισμός κατεύθυνσης κίνησης με τυχαίο τρόπο (Random Direction)

Αυτή η εντολή λέει στο NXT να θέσει την κατεύθυνση ενός κινητήρα με τυχαίο τρόπο. Η εντολή αυτή δεν ξεκινάει κανένα κινητήρα. Πρέπει να χρησιμοποιηθεί σε συνδυασμό με την εντολή εφαρμογής ισχύος εξόδου ή ενώ ήδη ένας κινητήρας είναι σε λειτουργία.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή θύρας εξόδου A ή B ή C. Αν δεν προσθέσουμε κανένα τροποποιητή θύρας εξόδου τότε θα εφαρμοστεί η τροφοδοσία ρεύματος και στις τρεις εξόδους.

Σε αυτήν την εντολή μπορούμε να προσθέσουμε έναν τροποποιητή (modifier) ισχύος για να ελέγξουμε το πόσο γρήγορα θα γυρίζει ο αντίστοιχος κινητήρας. Αν δεν προσθέσουμε τροποποιητή ισχύος, ο κινητήρας μας θα γυρίζει με τη μέγιστη ταχύτητα.

Παράδειγμα





Εντολές αναμονής (*Wait for Commands*)



Αναμονή για Χρόνο (*Wait for Time*)

Με αυτή την εντολή το NXT θα περιμένει μέχρι να περάσει το χρονικό διάστημα που καθορίζεται από τον αντίστοιχο τροποποιητή σε δευτερόλεπτα/λεπτά της ώρας.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή σταθερής αριθμητικής τιμής σε δευτερόλεπτα/λεπτά. Αν δεν προσθέσουμε κανένα τροποποιητή τότε το πρόγραμμα θα περιμένει για 1 δευτερόλεπτο/λεπτό της ώρας.

Άλλες εντολές αναμονής χρόνου.



Αναμονή για n δευτερόλεπτα (*Wait n seconds*)

Με αυτή την εντολή το NXT θα περιμένει μέχρι να περάσουν n δευτερόλεπτα.

Παράδειγμα



Αναμονή για n/100 δευτερόλεπτα (*Wait n/100 seconds*)

Με αυτή την εντολή το NXT θα περιμένει μέχρι να περάσουν n εκατοστά του δευτερόλεπτα που καθορίζεται από τον αντίστοιχο τροποποιητή.

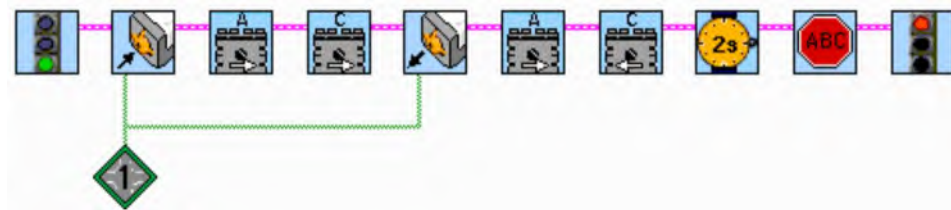
Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τουλάχιστον τροποποιητή σταθερής αριθμητικής τιμής.



NXT Αναμονή μέχρι να πατηθεί / να απελευθερωθεί (*NXT Wait for Push/Letgo*)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι να πατηθεί/απελευθερωθεί το κουμπί σε έναν αισθητήρα αφής (touch sensor). Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα



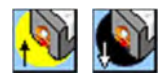
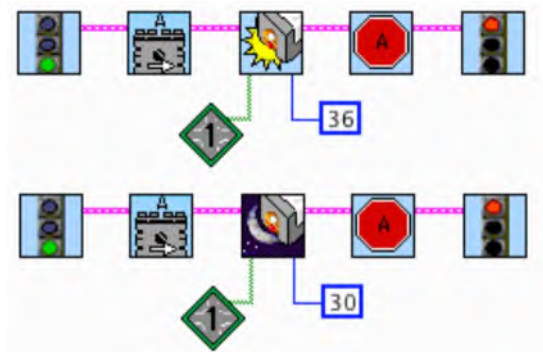
NXT Αναμονή για Φως / Σκοτάδι (NXT Wait for Light/Dark)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση της φωτεινότητας σε ένα αισθητήρα φωτός «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παραδείγματα



NXT Αναμονή για Φωτεινότερο / Σκοτεινότερο (NXT Wait for Brighter/Darker)

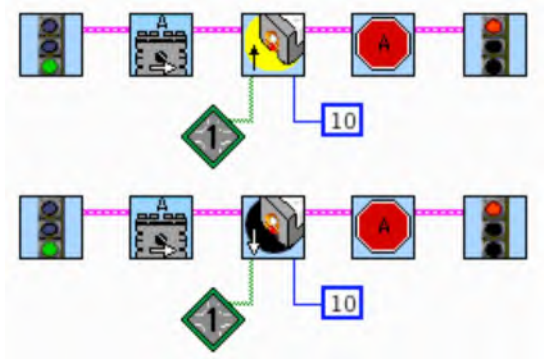
Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση της φωτεινότητας σε ένα αισθητήρα φωτός να αυξηθεί / μειωθεί κατά ένα συγκεκριμένο ποσό το οποίο καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει πόσο μεγάλη πρέπει να είναι η αλλαγή στη φωτεινότητα πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 5.

Σενάρια Ρομποτικής με το Mindstorms NXT

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παραδείγματα



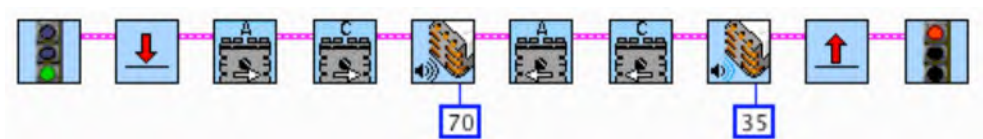
NXT Αναμονή για ήχο Δυνατό/Σιγανό (NXT Wait for Loud/Quiet)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση του ήχου που μετρείται από έναν αισθητήρα ήχου «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 70 dB.

Ο αισθητήρας ήχου ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα



NXT Αναμονή για Δυνατότερο / Σιγανότερο ήχο (NXT Wait for Louder / Quieter)

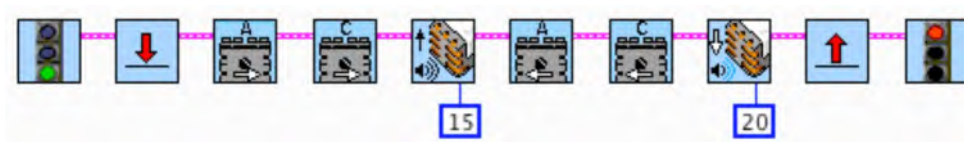
Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση του ήχου σε ένα αισθητήρα ήχου να αυξηθεί / μειωθεί κατά ένα συγκεκριμένο ποσό το οποίο καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σενάρια Ρομποτικής με το Mindstorms NXT

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει πόσο μεγάλη πρέπει να είναι η αλλαγή στη φωτεινότητα πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 30 %.

Ο αισθητήρας ήχου ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα



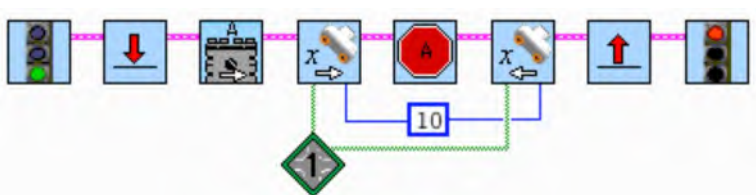
NXT Αναμονή για απόσταση μεγαλύτερη/μικρότερη από X (NXT Forward/Reverse Distance X)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η απόσταση σε cm που μετρείται έναν αισθητήρα υπερήχων, από ένα εμπόδιο «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 20 cm. Οι επιτρεπτές τιμές είναι από 1 έως 200 cm.

Ο αισθητήρας υπερήχων ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα





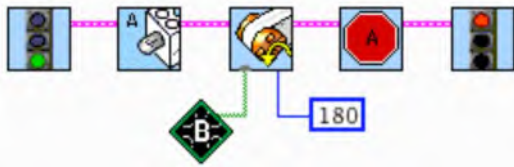
NXT Αναμονή για περιστροφή (NXT Wait for Rotation)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι ο άξονας σε ένα αισθητήρα περιστροφής να περιστραφεί κατά μία συγκεκριμένη τιμή σε μοίρες, η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει την τιμή που πρέπει να περιστραφεί ο άξονας πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 360° (ολόκληρη περιστροφή).

Ο αισθητήρας περιστροφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εξόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εξόδου τότε εξ ορισμού ελέγχεται η θύρα εξόδου A.

Παράδειγμα



NXT Αναμονή για ακριβή περιστροφή (NXT Wait for Location)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι ο άξονας σε ένα αισθητήρα περιστροφής να περιστραφεί κατά μία συγκεκριμένη τιμή σε μοίρες, η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

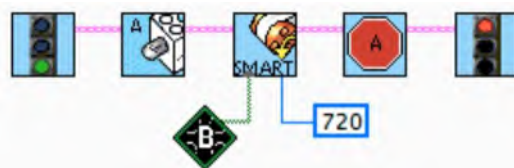
Όταν ολοκληρωθεί η επιθυμητή περιστροφή ο κινητήρας θα φρενάρει απότομα.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει την τιμή που πρέπει να περιστραφεί ο άξονας πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 360° (ολόκληρη περιστροφή).

Ο αισθητήρας περιστροφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εξόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εξόδου τότε εξ ορισμού ελέγχεται η θύρα εξόδου A.

Αν έχουμε πιο μπροστά καθορίσει να υπάρχει συγχρονισμός των κινητήρων τότε χρειάζεται να τοποθετήσουμε τροποποιητή μόνο για τον κινητήρα ελέγχου και όχι και για τους δύο κινητήρες.

Παράδειγμα





NXT Αναμονή για περιστροφή χωρίς αρχικοποίηση (NXT Wait for Rotation w/o Reset)

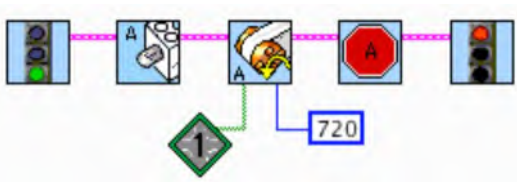
Αυτή η εντολή λέει στο NXT να περιμένει μέχρι ο άξονας σε ένα αισθητήρα περιστροφής να περιστραφεί κατά μία συγκεκριμένη τιμή σε μοίρες, η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Με την εντολή αυτή δεν θα προηγηθεί μηδενισμός του μετρητή περιστροφής του αντίστοιχου αισθητήρα. Μπορεί δηλαδή τη στιγμή που το πρόγραμμα φτάσει στην εντολή αυτή το κριτήριο της περιστροφής να έχει επιτευχθεί.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει την τιμή που πρέπει να περιστραφεί ο άξονας πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 360° (ολόκληρη περιστροφή).

Ο αισθητήρας περιστροφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εξόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εξόδου τότε εξ ορισμού ελέγχεται η θύρα εξόδου A.

Παράδειγμα



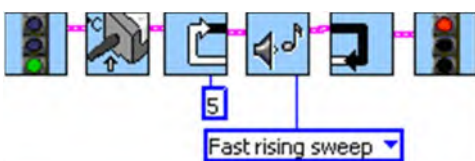
NXT Αναμονή για Θερμότερο / Ψυχρότερο (NXT Wait for Temperature Increase / Decrease)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι θερμοκρασία σε ένα αισθητήρα θερμοκρασίας να αυξηθεί / μειωθεί κατά ένα συγκεκριμένο ποσό το οποίο καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει πόσο μεγάλη πρέπει να είναι η αλλαγή στη φωτεινότητα πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 25. Οι επιτρεπτές τιμές είναι από -20 έως 120 °C.

Ο αισθητήρας θερμοκρασίας ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα

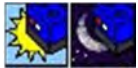
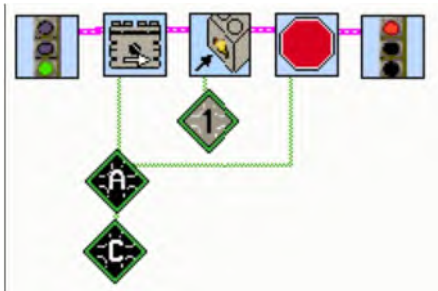




Αναμονή μέχρι να πατηθεί / να απελευθερωθεί (Wait for Push/Letgo)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι να πατηθεί/απελευθερωθεί ο διακόπτης σε έναν αισθητήρα αφής (touch sensor). Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα



Αναμονή για Φως / Σκοτάδι (Wait for Light/Dark)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση της φωτεινότητας σε ένα αισθητήρα φωτός «να αυξηθεί πάνω από / μειωθεί κάτω από» μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα





Αναμονή για Φωτεινότερο / Σκοτεινότερο (Wait for Brighter/Darker)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η ένταση της φωτεινότητας σε ένα αισθητήρα φωτός να αυξηθεί / μειωθεί κατά ένα συγκεκριμένο ποσό το οποίο καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει πόσο μεγάλη πρέπει να είναι η αλλαγή στη φωτεινότητα πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 5.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παραδείγματα



Αναμονή για Γωνία (Wait for Angle)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι ο άξονας σε ένα αισθητήρα περιστροφής να γυρίσει κατά μία γωνία (σε μοίρες) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει την τιμή που πρέπει να περιστραφεί ο άξονας πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 180° (μισή περιστροφή).

Η περιστροφή μπορεί να συμβεί είτε προς τα μπροστά είτε προς τα πίσω.

Παράδειγμα



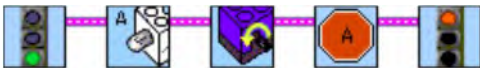


Αναμονή για Περιστροφή (Wait for Rotation)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι ο άξονας σε ένα αισθητήρα περιστροφής να περιστραφεί κατά μία συγκεκριμένη τιμή (σε 1/16) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει την τιμή που πρέπει να περιστραφεί ο άξονας πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 16 (ολόκληρη περιστροφή).

Παράδειγμα



Αναμονή για Μείωση / Αύξηση Θερμοκρασίας (Wait for Decreasing / Increasing Temp)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η θερμοκρασία σε ένα αισθητήρα θερμοκρασίας να μειωθεί κάτω από / αυξηθεί πάνω από μία συγκεκριμένη τιμή η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 30.

Παράδειγμα





Αναμονή για Μεταβλητή (Wait for Container)

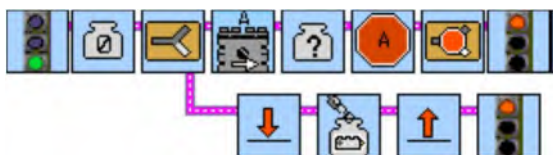
Αυτή η εντολή λέει στο NXT να περιμένει μέχρι η τιμή μιας μεταβλητής γίνει ίση με ένα συγκεκριμένο αριθμό ο οποίος καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει τον αριθμό που πρέπει να αποκτήσει η μεταβλητή πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 1.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής για να καθορίσουμε σε ποια μεταβλητή αναφερόμαστε. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού η κόκκινη μεταβλητή που συμβολίζεται με το κόκκινο δοχείο.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού μεταβλητής.

Παράδειγμα



Αναμονή για Χρόνομετρητή (Wait for Timer 0.1/0.01)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι το εσωτερικό ρολόι σε έναν χρονομετρητή να φτάσει σε κάποιο συγκεκριμένο χρόνο, ο οποίος καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής. Ο χρόνος αυτός αναγράφεται σε δέκατα / εκατοστά του δευτερολέπτου.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει το χρόνο που πρέπει να μετρήσει ο χρονομετρητής πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 1 δευτερόλεπτο.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή χρονομετρητή για να καθορίσουμε σε ποια χρονομετρητή αναφερόμαστε. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού ο κόκκινος χρονομετρητής που συμβολίζεται με το κόκκινο ρολόι.

Πρέπει πάντα πριν από αυτήν την εντολή να χρησιμοποιούμε την εντολή μηδενισμού χρονομετρητή.

Παράδειγμα





Αναμονή για Ρολόι (Wait for Clock)

Αυτή η εντολή λέει στο NXT να περιμένει μέχρι το ρολόι του να φτάσει σε κάποιο συγκεκριμένο χρόνο σε λεπτά, ο οποίος καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ο οποίος καθορίζει το χρόνο που πρέπει να μετρήσει το ρολόι πριν το πρόγραμμα να συνεχίσει με την επόμενη εντολή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 1 λεπτό.

Παράδειγμα



Αναμονή για μήνυμα (Wait for mail)

Αυτή η εντολή κάνει το NXT να περιμένει μέχρι να λάβει κάποιο μήνυμα από ένα άλλο NXT μέσω της θύρας υπερύθρων του.

Παράδειγμα





Δομές (Structures)



Δομές Επιλογής ή Διακλάδωσης (Forks)

Με μία εντολή διακλάδωσης δημιουργούνται μέσα στο πρόγραμμα δύο πιθανές διαδρομές (μονοπάτια) τις οποίες μπορεί να ακολουθήσει το πρόγραμμα μας. Από αυτά τα δύο μονοπάτια το πρόγραμμα θα ακολουθήσει μόνο το ένα, ανάλογα με την κατάσταση που θα έχει διαμορφωθεί εκείνη τη στιγμή, μετά από την πραγματοποίηση ενός ελέγχου. Τα δύο αυτά μονοπάτια τα οποία προκύπτουν από τις εντολές διακλάδωσης πρέπει στο τέλος τους να τα συνενώσουμε με μια εντολή "συγχώνευσης διακλάδωσης" (fork merge)

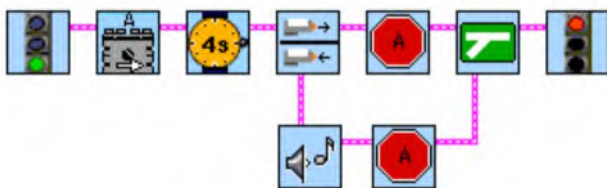


NXT Διακλάδωση αισθητήρα Αφής (NXT Touch sensor Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν τη στιγμή του ελέγχου ο διακόπτης του αισθητήρα αφής βρεθεί πατημένος ή όχι. Αν βρεθεί πατημένος θα ακολουθηθεί το κάτω μονοπάτι ενώ στην αντίθετη περίπτωση το πάνω.

Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου νο 1.

Παράδειγμα



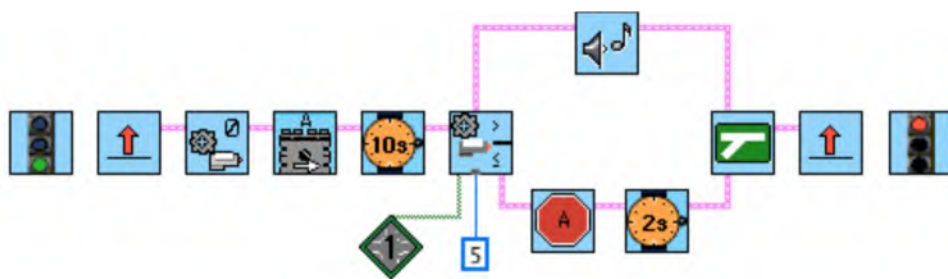


NXT Διακλάδωση Πατήματος (NXT Click Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι μεγαλύτερος ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων του αισθητήρα αφής.

Παράδειγμα

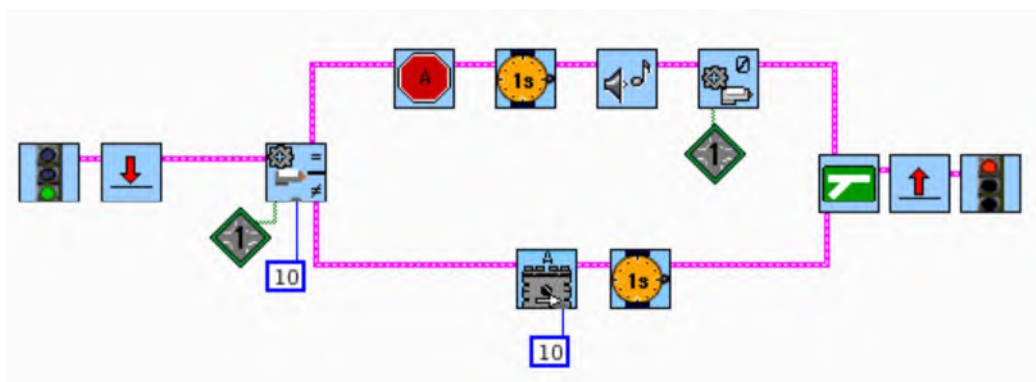


NXT Διακλάδωση Πατήματος με ισότητα (NXT Click Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι ίσος ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων του αισθητήρα αφής.

Παράδειγμα



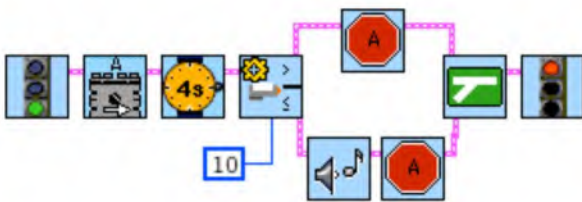


NXT Διακλάδωση Πατήματος και Απελευθέρωσης (NXT Touch and Release Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί και απελευθερωθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι μεγαλύτερος ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων και απελευθερώσεων του αισθητήρα αφής.

Παράδειγμα

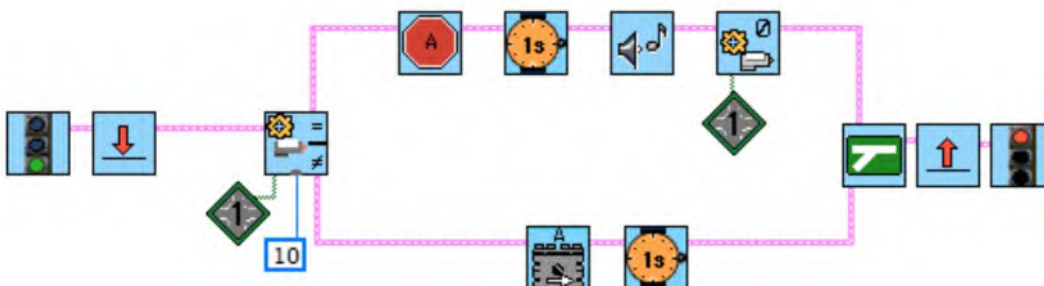


NXT Διακλάδωση Πατήματος και Απελευθέρωσης με Ισότητα (NXT Touch and Release Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί και απελευθερωθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι ίσος ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων και απελευθερώσεων του αισθητήρα αφής.

Παράδειγμα





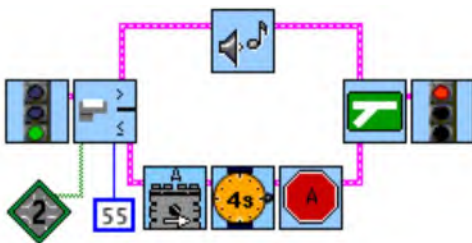
NXT Διακλάδωση αισθητήρα Φωτός (NXT Light sensor Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα φωτός είναι μεγαλύτερη ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς. Αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα φωτός είναι μεγαλύτερη από τη σταθερή αριθμητική τιμή που έχουμε εμείς καθορίσει, θα ακολουθηθεί το πάνω μονοπάτι ενώ στην αντίθετη περίπτωση ισότητας ή μικρότερης από, θα ακολουθεί το κάτω μονοπάτι.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

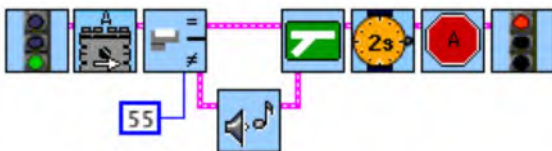
Παράδειγμα



NXT Διακλάδωση αισθητήρα Φωτός με ισότητα (NXT Light sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα φωτός είναι ίση ή όχι, με μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Παράδειγμα





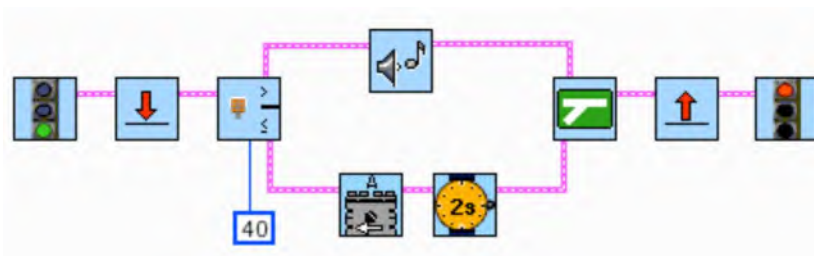
NXT Διακλάδωση αισθητήρα Ήχου (NXT Sound Sensor Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα ήχου είναι μεγαλύτερη ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς. Αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα ήχου είναι μεγαλύτερη από τη σταθερή αριθμητική τιμή που έχουμε εμείς καθορίσει, θα ακολουθηθεί το πάνω μονοπάτι ενώ στην αντίθετη περίπτωση ισότητας ή μικρότερης από, θα ακολουθεί το κάτω μονοπάτι.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας ήχου ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

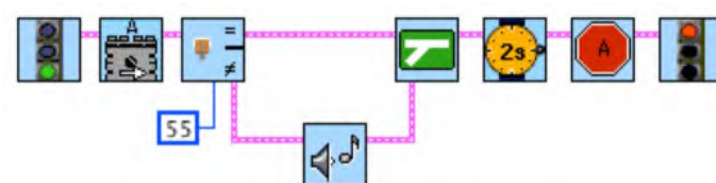
Παράδειγμα



NXT Διακλάδωση αισθητήρα ήχου με ισότητα (NXT Sound Sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα ήχου είναι ίση ή όχι, με μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Παράδειγμα





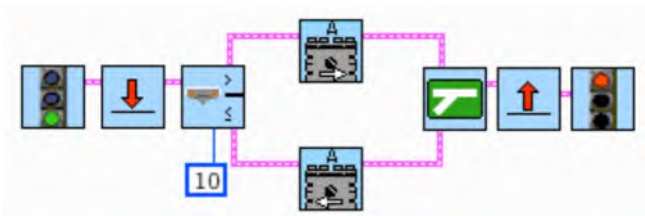
NXT Διακλάδωση αισθητήρα Υπερήχων (NXT Distance Sensor Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η απόσταση που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα υπερήχων είναι μεγαλύτερη ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς. Αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα υπερήχων είναι μεγαλύτερη από τη σταθερή αριθμητική τιμή που έχουμε εμείς καθορίσει, θα ακολουθηθεί το πάνω μονοπάτι ενώ στην αντίθετη περίπτωση ισότητας ή μικρότερης από, θα ακολουθεί το κάτω μονοπάτι.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 20.

Ο αισθητήρας υπερήχων ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

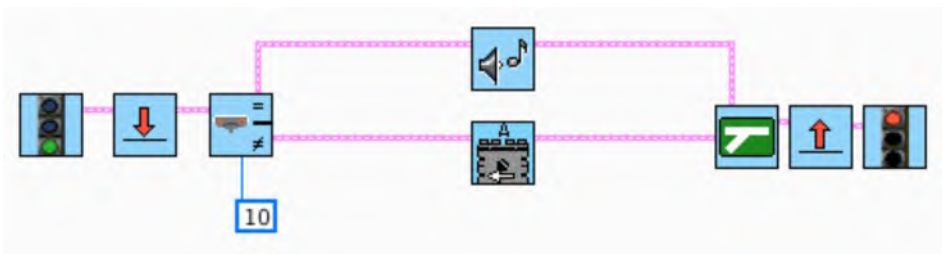
Παράδειγμα



NXT Διακλάδωση αισθητήρα Υπερήχων με ισότητα (NXT Distance Sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η απόσταση που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα υπερήχων είναι ίση ή όχι, με μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Παράδειγμα





NXT Διακλάδωση αισθητήρα Περιστροφής (NXT Rotation sensor Fork)

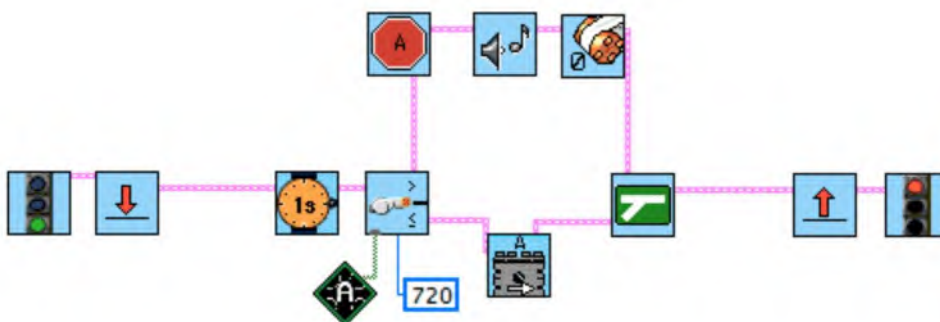
Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα περιστροφής είναι μεγαλύτερη ή όχι, από μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς. Αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα περιστροφής είναι μεγαλύτερη από τη σταθερή αριθμητική τιμή που έχουμε εμείς καθορίσει, θα ακολουθηθεί το πάνω μονοπάτι ενώ στην αντίθετη περίπτωση ισότητας ή μικρότερης από, θα ακολουθεί το κάτω μονοπάτι.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 360° (μία περιστροφή).

Ο αισθητήρας περιστροφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εξόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εξόδου τότε εξ ορισμού ελέγχεται η θύρα εξόδου A.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού αισθητήρα περιστροφής.

παράδειγμα

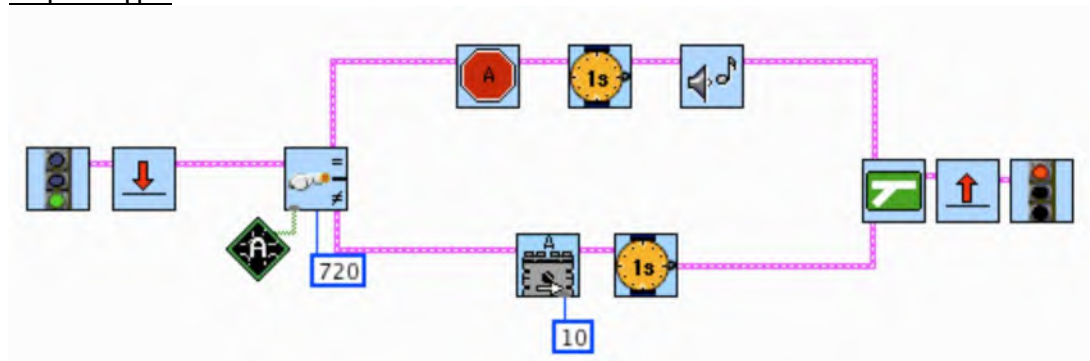


NXT Διακλάδωση αισθητήρα Περιστροφής με ισότητα (NXT Rotation sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα περιστροφής είναι ίση ή όχι, με μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού αισθητήρα περιστροφής.

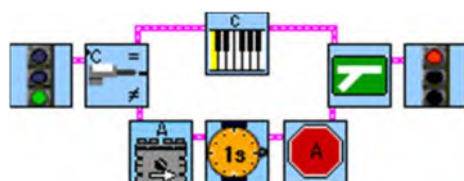
Παράδειγμα



NXT Διακλάδωση αισθητήρα Θερμοκρασίας με ισότητα (NXT Temperature sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα θερμοκρασίας είναι ίση ή όχι, με μία συγκεκριμένη αριθμητική σταθερή τιμή που καθορίζουμε εμείς.

Παράδειγμα

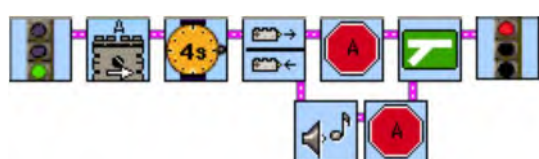


Διακλάδωση αισθητήρα Αφής (Touch sensor Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν τη στιγμή του ελέγχου ο διακόπτης του αισθητήρα αφής βρεθεί πατημένος ή όχι. Αν βρεθεί πατημένος θα ακολουθηθεί το κάτω μονοπάτι ενώ στην αντίθετη περίπτωση το πάνω.

Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα





Διακλάδωση Πατήματος (Click Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι μεγαλύτερος ή όχι, από μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων του αισθητήρα αφής.

Παράδειγμα



Διακλάδωση Πατήματος με ισότητα (Click Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι ίσος ή όχι, από μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων του αισθητήρα αφής.

Παράδειγμα





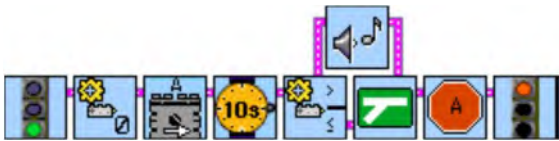
Διακλάδωση Πατήματος και Απελευθέρωσης (Touch and Release Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί και απελευθερωθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι μεγαλύτερος ή όχι, από μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων και απελευθερώσεων του αισθητήρα αφής.

Παράδειγμα



Διακλάδωση Πατήματος και Απελευθέρωσης με ισότητα (Touch and Release Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν ο αριθμός των φορών που έχει πατηθεί και απελευθερωθεί ο διακόπτης του αισθητήρα αφής μέχρι τη στιγμή του ελέγχου, είναι ίσος ή όχι, από μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων και απελευθερώσεων του αισθητήρα αφής.

Παράδειγμα





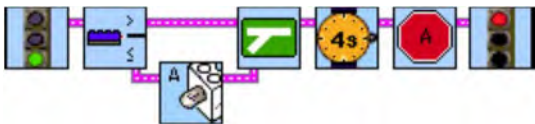
Διακλάδωση αισθητήρα Φωτός (Light sensor Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα φωτός είναι μεγαλύτερη ή όχι, από μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς. Αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα φωτός είναι μεγαλύτερη από τη σταθερή αριθμητική τιμή που έχουμε εμείς καθορίσει, θα ακολουθηθεί το πάνω μονοπάτι ενώ στην αντίθετη περίπτωση ισότητας ή μικρότερης από, θα ακολουθεί το κάτω μονοπάτι.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα



Διακλάδωση αισθητήρα Φωτός με ισότητα (Light sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα φωτός είναι ίση ή όχι, με μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς.

Παράδειγμα



Διακλάδωση αισθητήρα Περιστροφής (Rotation sensor Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα περιστροφής είναι μεγαλύτερη ή όχι, από μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς. Αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα περιστροφής είναι μεγαλύτερη από τη σταθερή αριθμητική τιμή που έχουμε εμείς καθορίσει, θα ακολουθηθεί το πάνω μονοπάτι ενώ στην αντίθετη περίπτωση ισότητας ή μικρότερης από, θα ακολουθεί το κάτω μονοπάτι.

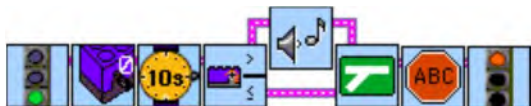
Σενάρια Ρομποτικής με το Mindstorms NXT

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 16 (μία περιστροφή).

Ο αισθητήρας περιστροφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού αισθητήρα περιστροφής.

Παράδειγμα



Διακλάδωση αισθητήρα Περιστροφής με ισότητα (Rotation sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα περιστροφής είναι ίση ή όχι, με μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς. Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού αισθητήρα περιστροφής.

Παράδειγμα



Διακλάδωση αισθητήρα Θερμοκρασίας (Temperature sensor Fork)

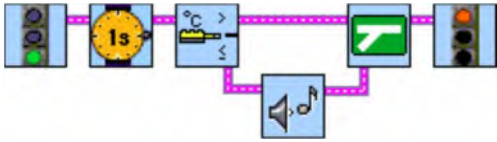
Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα θερμοκρασίας είναι μεγαλύτερη ή όχι, από μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς. Αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα θερμοκρασίας είναι μεγαλύτερη από τη σταθερή αριθμητική τιμή που έχουμε εμείς καθορίσει, θα ακολουθηθεί το πάνω μονοπάτι ενώ στην αντίθετη περίπτωση ισότητας ή μικρότερης από, θα ακολουθεί το κάτω μονοπάτι.

Σενάρια Ρομποτικής με το Mindstorms NXT

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 30.

Ο αισθητήρας θερμοκρασίας ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

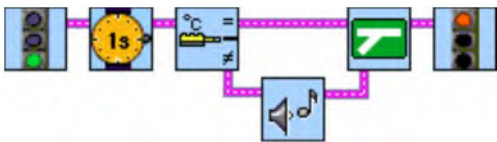
Παράδειγμα



Διακλάδωση αισθητήρα Θερμοκρασίας με ισότητα (Temperature sensor Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με το αν η τιμή που έχει μετρηθεί εκείνη τη στιγμή από τον αισθητήρα θερμοκρασίας είναι ίση ή όχι, με μία συγκεκριμένη σταθερή αριθμητική τιμή που καθορίζουμε εμείς.

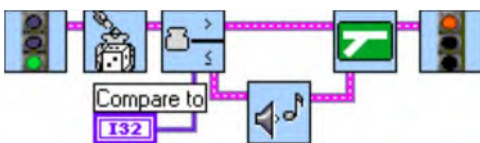
Παράδειγμα



Διακλάδωση Μεταβλητής (Container Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με την τιμή που έχει μια μεταβλητή τη στιγμή του ελέγχου.

Παράδειγμα

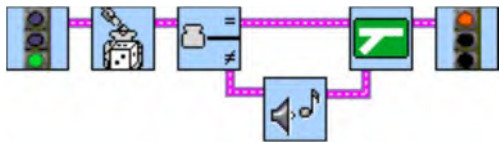




Διακλάδωση Μεταβλητής με ισότητα (Container Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με την τιμή που έχει μια μεταβλητή τη στιγμή του ελέγχου.

Παράδειγμα

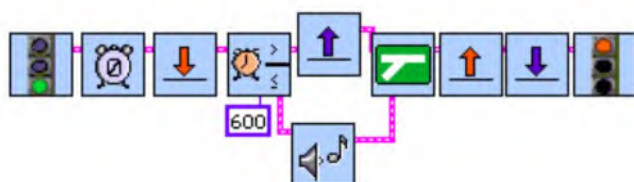


Διακλάδωση Χρονομετρητή (Timer Fork 0.1/0.01)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με την τιμή που έχει ένας χρονομετρητής τη στιγμή του ελέγχου.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού χρονομετρητή.

Παραδείγματα



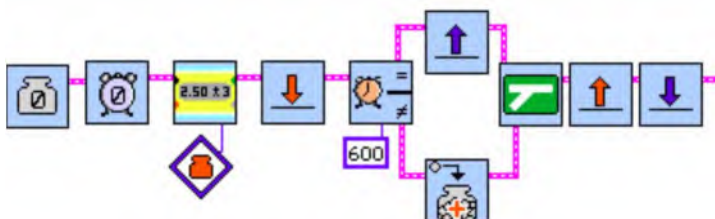
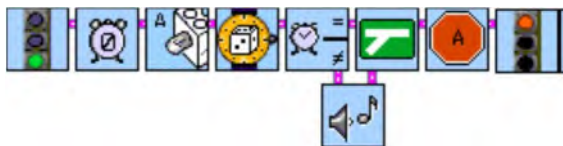


Διακλάδωση Χρονομετρητή με ισότητα (Timer Equal Fork 0.1/0.01)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με την τιμή που έχει ένας χρονομετρητής τη στιγμή του ελέγχου.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού χρονομετρητή.

Παραδείγματα

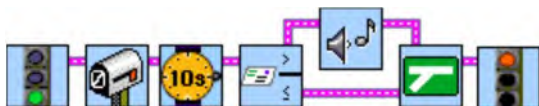


Διακλάδωση Γραμματοθυρίδας (Mailbox Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με την τιμή βρίσκεται στη γραμματοθυρίδα τη στιγμή του ελέγχου.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού γραμματοθυρίδας.

Παράδειγμα





Διακλάδωση Γραμματοθυρίδας με ισότητα (Mailbox Equal Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν, ανάλογα με την τιμή βρίσκεται στη γραμματοθυρίδα τη στιγμή του ελέγχου.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού γραμματοθυρίδας.

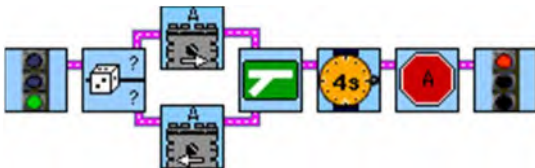
Παράδειγμα



Διακλάδωση με τυχαίο τρόπο (Random Fork)

Αυτή η εντολή έχει ως αποτέλεσμα το NXT να ακολουθήσει ένα από τα δύο μονοπάτια που ακολουθούν με εντελώς τυχαίο τρόπο.

Παράδειγμα



Συγχώνευση διακλάδωσης (Fork Merge)

Η εντολή αυτή συνενώνει τα δύο ξεχωριστά μονοπάτια που προκύπτουν μετά από μία εντολή διακλάδωσης, ώστε να συνεχίσει η εκτέλεση του προγράμματος με τις εντολές που ακολουθούν μετά τη διακλάδωση.

Η εντολή αυτή είναι απαραίτητη μετά από μία οποιαδήποτε εντολή διακλάδωσης.

Δομές Επανάληψης (Loop)



Άλμα χωρίς συνθήκη (Jump/Land)



Προσγείωση (Red/Blue/Yellow/Green/Black Land)



Άλμα (Red/Blue/Yellow/Green/Black Jump)

Οι παρακάτω εντολές τοποθετούνται σε ένα πρόγραμμα κατά δυάδες των αντίστοιχων χρωμάτων (πχ κόκκινη προσγείωση με κόκκινο άλμα).

Όταν μέσα στο πρόγραμμα βρεθεί μια εντολή άλματος, το πρόγραμμα δεν θα συνεχίσει την εκτέλεση του με την επόμενη εντολή που ακολουθεί την εντολή άλματος, αλλά από το σημείο που βρίσκεται η αντίστοιχου χρώματος εντολή προσγείωσης.

Μια εντολή άλματος καταργεί τη σειριακή εκτέλεση των εντολών δίνοντας μας τη δυνατότητα να κάνουμε ένα άλμα σε ένα άλλο σημείο του προγράμματος.

Μια εφαρμογή των εντολών προσγείωσης άλματος είναι να δημιουργήσουμε ένα ατέρμονα βρόγχο, μία επανάληψη δηλαδή μιας συγκεκριμένης σειράς εντολών χωρίς τέλος.



Επανάληψη με Βρόχο

Α. Επανάληψη με μετρητή χωρίς συνθήκη

Με αυτές τις εντολές μπορούμε να προκαλέσουμε την επανάληψη μιας σειράς εντολές κάποιες συγκεκριμένες φορές, καθορισμένες εκ των προτέρων.



Αρχή της επανάληψης (Start of Loop)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν μέχρι την εντολή "τέλους της επανάληψης", τόσες φορές, όσες φορές καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής για να καθορίσουμε τον αριθμό των επαναλήψεων που επιθυμούμε να συμβούν. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 2.

Σενάρια Ρομποτικής με το Mindstorms NXT

Μετά την εντολή αυτή τοποθετούμε μια σειρά εντολών τις οποίες θέλουμε το πρόγραμμα να επαναλάβει. Στη συνέχεια τοποθετείται πάντοτε η εντολή του "τέλους της επανάληψης". Η σειρά των εντολών που βρίσκεται ανάμεσα στην εντολή "αρχή της επανάληψης" και την εντολή "τέλους της επανάληψης" εκτελείται τόσες φορές όσες έχουμε καθορίσουμε με τον τροποποιητή αριθμητικής σταθερής τιμής.



Τέλος της επανάληψης (End of Loop)

Με την εντολή αυτή, το πρόγραμμα επιστρέφει στην εντολή της "αρχής της επανάληψης". Κάθε εντολή "αρχή της επανάληψης" απαιτεί την ύπαρξη μιας εντολής "τέλους της επανάληψης" ώστε να αγκαλιάσουν μαζί τις εντολές που πρόκειται εκτελεστούν περισσότερες από μία φορές.

Παράδειγμα



Αρχή της αέναης επανάληψης (Start of Forever Loop)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της αέναης επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν μέχρι την εντολή "τέλους της αέναης επανάληψης" συνεχώς.

Παράδειγμα



Τέλος της αέναης επανάληψης (End of Forever Loop)

Με την εντολή αυτή, το πρόγραμμα επιστρέφει στην εντολή της "αρχής της αέναης επανάληψης". Κάθε εντολή "αρχή της αέναης επανάληψης" απαιτεί την ύπαρξη μιας εντολής "τέλους αέναης της επανάληψης" ώστε να αγκαλιάσουν μαζί τις εντολές που πρόκειται εκτελεστούν περισσότερες από μία φορές.

Παράδειγμα



Β. Επανάληψη με συνθήκη

Με αυτές τις εντολές μπορούμε να προκαλέσουμε την επανάληψη μιας σειράς εντολές, όχι κάποιες συγκεκριμένες φορές καθορισμένες εκ των προτέρων, αλλά για όσο διάστημα ισχύει μια συνθήκη ελέγχου, υπάρχει δηλαδή μια επιθυμητή κατάσταση.



NXT Επανάλαβε όσο ο αισθητήρας αφής είναι πατημένος (NXT Loop while Touch Sensor is Pushed)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα παραμένει πατημένο το κουμπί ενός αισθητήρα αφής.

Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Μετά την εντολή αυτή τοποθετούμε μια σειρά εντολών τις οποίες θέλουμε το πρόγραμμα να επαναλάβει τις φορές που επιθυμούμε. Στη συνέχεια τοποθετείται πάντοτε η εντολή του "τέλους της επανάληψης". Η σειρά των εντολών που βρίσκεται ανάμεσα στην εντολή "αρχή της επανάληψης" αισθητήρα αφής" και την εντολή "τέλους της επανάληψης" εκτελείται τόσες φορές όσες έχουμε καθορίσουμε με τον τροποποιητή αριθμητικής σταθερής τιμής

Παράδειγμα



NXT Επανάλαβε όσο ο αισθητήρας αφής είναι απελευθερωμένος (NXT Loop while Touch Sensor is Released)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα δεν είναι πατημένο το κουμπί ενός αισθητήρα αφής.

Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Μετά την εντολή αυτή τοποθετούμε μια σειρά εντολών τις οποίες θέλουμε το πρόγραμμα να επαναλάβει τις φορές που επιθυμούμε. Στη συνέχεια τοποθετείται πάντοτε η εντολή του "τέλους της επανάληψης". Η σειρά των εντολών που βρίσκεται ανάμεσα στην εντολή "αρχή της επανάληψης" αισθητήρα αφής" και την εντολή "τέλους της επανάληψης" εκτελείται τόσες φορές όσες έχουμε καθορίσουμε με τον τροποποιητή αριθμητικής σταθερής τιμής.

Παράδειγμα





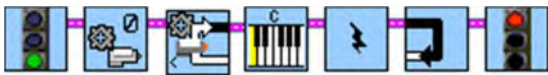
NXT Επανάλαβε όσο ο αριθμός των πατημάτων είναι μικρότερος από
(NXT Loop while Number of Clicks is Less than)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα ο αριθμός των πατημάτων του διακόπτη σε έναν αισθητήρα αφής είναι μικρότερος από μία συγκεκριμένη αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων του αισθητήρα αφής.

Παράδειγμα



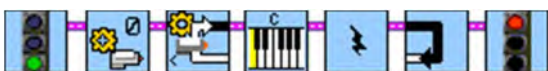
NXT Επανάλαβε όσο ο αριθμός των πατημάτων και των απελευθερώσεων είναι μικρότερος από
(NXT Loop while Number of Touches and Releases is Less than)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα ο αριθμός των πατημάτων και απελευθερώσεων του διακόπτη σε έναν αισθητήρα αφής είναι μικρότερος από μία συγκεκριμένη αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων και απελευθερώσεων του αισθητήρα αφής.

Παράδειγμα





**NXT Επανάλαβε όσο ο αισθητήρας φωτός είναι μικρότερος /
μεγαλύτερος από
(NXT Loop while Light Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η ένταση της φωτεινότητας σε έναν αισθητήρα φωτός είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα



**NXT Επανάλαβε όσο ο αισθητήρας ήχων είναι
μικρότερος / μεγαλύτερος από
(NXT Loop while Sound Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η ένταση του ήχου σε έναν αισθητήρα ήχου είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα





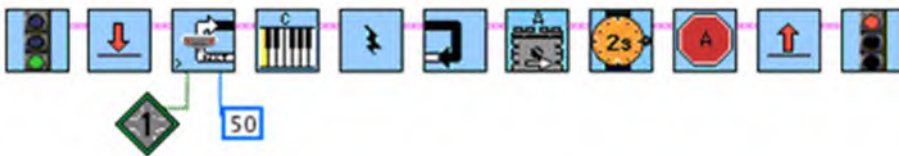
**NXT Επανάλαβε όσο ο αισθητήρας υπερήχων είναι μικρότερος / μεγαλύτερος από
(NXT Loop while Ultrasound Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η απόσταση που μετρίεται σε έναν αισθητήρα υπερήχων είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 20.

Ο αισθητήρας υπερήχων ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα



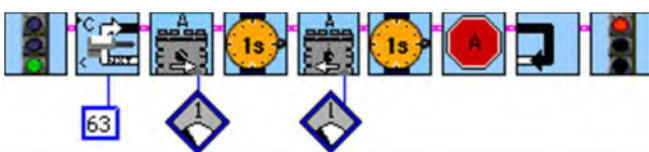
**NXT Επανάλαβε όσο ο αισθητήρας θερμότητας είναι μικρότερος / μεγαλύτερος από
(NXT Loop while Temperature Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η θερμοκρασία που ανιχνεύεται από έναν αισθητήρα θερμοκρασίας είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 25.

Ο αισθητήρας θερμοκρασίας ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα





**NXT Επανάλαβε όσο ο αισθητήρας περιστροφής είναι μικρότερος / μεγαλύτερος από
(NXT Loop while Rotation Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή του τέλους της επανάληψης, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η περιστροφή του άξονα σε έναν αισθητήρα περιστροφής είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 360.

Ο αισθητήρας περιστροφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εξόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εξόδου τότε εξ ορισμού ελέγχεται η θύρα εξόδου A.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού αισθητήρα περιστροφής.

παράδειγμα



**Επανάλαβε όσο ο αισθητήρας αφής είναι πατημένος
(Loop while Touch Sensor is Pushed)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα παραμένει πατημένος ο διακόπτης ενός αισθητήρα αφής.

Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Μετά την εντολή αυτή τοποθετούμε μια σειρά εντολών τις οποίες θέλουμε το πρόγραμμα να επαναλάβει. Στη συνέχεια τοποθετείται πάντοτε η εντολή του "τέλους της επανάληψης". Η σειρά των εντολών που βρίσκεται ανάμεσα στην εντολή "αρχή της επανάληψης αισθητήρα αφής" και την εντολή "τέλους της επανάληψης" εκτελείται τόσες φορές όσες έχουμε καθορίσουμε με τον τροποποιητή αριθμητικής σταθερής τιμής

Παράδειγμα





Επανάλαβε όσο ο αισθητήρας αφής είναι απελευθερωμένος (Loop while Touch Sensor is Released)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα δεν είναι πατημένο το κουμπί ενός αισθητήρα αφής.

Ο αισθητήρας αφής ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Μετά την εντολή αυτή τοποθετούμε μια σειρά εντολών τις οποίες θέλουμε το πρόγραμμα να επαναλάβει. Στη συνέχεια τοποθετείται πάντοτε η εντολή του "τέλους της επανάληψης". Η σειρά των εντολών που βρίσκεται ανάμεσα στην εντολή "αρχή της επανάληψης αισθητήρα αφής" και την εντολή "τέλους της επανάληψης" εκτελείται τόσες φορές όσες έχουμε καθορίσουμε με τον τροποποιητή αριθμητικής σταθερής τιμής.

Παράδειγμα



Επανάλαβε όσο ο αριθμός των πατημάτων είναι μικρότερος από (Loop while Number of Clicks is Less than)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα ο αριθμός των πατημάτων του διακόπτη σε έναν αισθητήρα αφής είναι μικρότερος από μία συγκεκριμένη αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων του αισθητήρα αφής.

Παράδειγμα





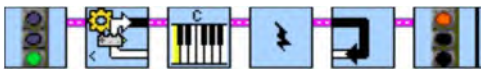
**Επανάλαβε όσο ο αριθμός των πατημάτων
και των απελευθερώσεων είναι μικρότερος από
(Loop while Number of Touches and Releases is Less than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή “τέλους της επανάληψης”, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα ο αριθμός των πατημάτων και απελευθερώσεων του διακόπτη σε έναν αισθητήρα αφής είναι μικρότερος από μία συγκεκριμένη αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 10.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού πατημάτων και απελευθερώσεων του αισθητήρα αφής.

Παράδειγμα



**Επανάλαβε όσο ο αισθητήρας φωτός είναι μικρότερος /
μεγαλύτερος από
(Loop while Light Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή “τέλους της επανάληψης”, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η ένταση της φωτεινότητας σε έναν αισθητήρα φωτός είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 55.

Ο αισθητήρας φωτός ο οποίος θα παρακολουθείται από το πρόγραμμα καθορίζεται από τον αντίστοιχο τροποποιητή θύρας εισόδου τον οποίο προσθέτουμε. Αν δεν έχει προστεθεί κανένας τροποποιητής θύρας εισόδου τότε εξ ορισμού ελέγχεται η θύρα εισόδου 1.

Παράδειγμα





**Επανάλαβε όσο ο αισθητήρας περιστροφής είναι μικρότερος /
μεγαλύτερος από
(Loop while Rotation Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή “τέλους της επανάληψης”, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η περιστροφή του άξονα σε έναν αισθητήρα περιστροφής είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 16.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού αισθητήρα περιστροφής.

Παράδειγμα



**Επανάλαβε όσο ο αισθητήρας θερμότητας είναι μικρότερος /
μεγαλύτερος από
(Loop while Temperature Sensor Is Less/Greater Than)**

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή “τέλους της επανάληψης”, προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η θερμοκρασία που ανιχνεύεται από έναν αισθητήρα θερμοκρασίας είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 30.

Παράδειγμα



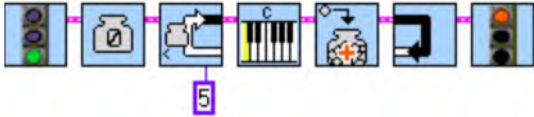


Επανάλαβε όσο η τιμή της μεταβλητής είναι μικρότερη / μεγαλύτερη από (Loop while Container is Less/Greater Than)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η τιμή μιας μεταβλητής είναι μικρότερη / μεγαλύτερη από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής.

Παράδειγμα

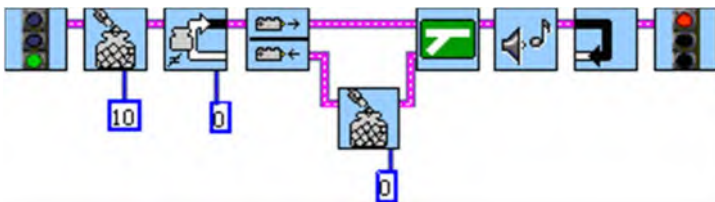


Επανάλαβε όσο η τιμή της μεταβλητής είναι ίση / άνιση από (Loop while Container is / is not Equal Than)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η τιμή μιας μεταβλητής είναι ίση / άνιση από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής.

Παράδειγμα





Επανάλαβε όσο η τιμή του χρονομετρητή είναι μικρότερη / μεγαλύτερη από (Loop while Timer is Less/Greater Than)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα ο χρόνος σε ένα χρονομετρητή είναι μικρότερος / μεγαλύτερος από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 5.

Πριν από αυτήν την εντολή μερικές φορές είναι βολικό να χρησιμοποιήσουμε την εντολή μηδενισμού χρονομετρητή.

Παράδειγμα



Επανάλαβε όσο η τιμή του μηνύματος είναι μικρότερη / μεγαλύτερη από (Loop while Mail is Less/Greater Than)

Η εντολή αυτή, συνδυαζόμενη πάντοτε με την εντολή "τέλους της επανάληψης", προκαλεί την εκτέλεση της σειράς των εντολών που ακολουθούν για όσο χρονικό διάστημα η τιμή ενός μηνύματος είναι μικρότερος / μεγαλύτερος από μία αριθμητική τιμή (κατώφλι) η οποία καθορίζεται από ένα τροποποιητή σταθερής αριθμητικής τιμής. Ο βρόχος τερματίζεται όταν κάποια στιγμή σταλεί το κατάλληλο μήνυμα.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής.

Παράδειγμα



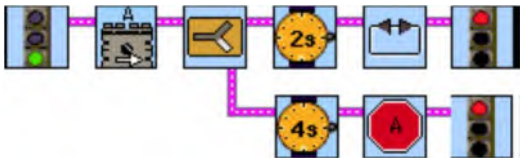
Παράλληλες Διεργασίες



Παράλληλες διεργασίες(Task split)

Αυτή η εντολή δημιουργεί δύο ανεξάρτητες διαδρομές (κλάδους) μέσα στο πρόγραμμα οι οποίοι θα εκτελεστούν ταυτόχρονα (παράλληλα). Κάθε ένας από αυτούς τους δύο κλάδους πρέπει να έχει τη δική του εντολή "Τέλος Προγράμματος"

Παράδειγμα

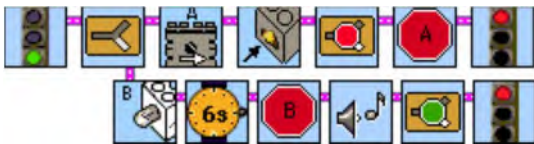


Σταμάτημα παράλληλων διεργασιών (Stop Tasks)

Η εντολή αυτή, ξεκινάει την εκτέλεση των παράλληλων διεργασιών.

Σε αυτή την εντολή μπορούμε να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής για να δηλώσουμε τον αριθμό της εντολής παράλληλων διεργασιών που θα σταματήσουν να εκτελούνται. Αν δεν δηλώσουμε κάποια αριθμητική τιμή θα σταματήσουν όλες οι παράλληλες διεργασίες.

Παράδειγμα

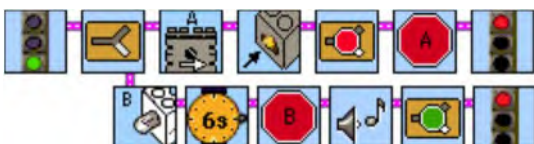


Εκκίνηση παράλληλων διεργασιών (Start Tasks)

Η εντολή αυτή, σταματάει την εκτέλεση των παράλληλων διεργασιών ή εκτέλεση των οποίων είχε σταματήσει.

Σε αυτή την εντολή μπορούμε να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής για να δηλώσουμε τον αριθμό της εντολής παράλληλων διεργασιών που θα ξεκινήσουν να εκτελούνται. Αν δεν δηλώσουμε κάποια αριθμητική τιμή θα ξεκινήσουν όλες οι παράλληλες διεργασίες.

Παράδειγμα



Υπορουτίνες

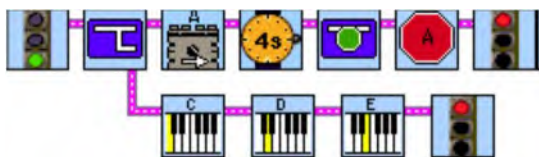


Δημιουργία Υπορουτίνας (Create Subroutine)

Με την εντολή αυτή δηλώνουμε μία υπορουτίνα για το πρόγραμμα μας. Στο τέλος της συγγραφής μιας υπορουτίνας πρέπει να τοποθετήσουμε την εντολή "Τέλος Προγράμματος". Η εντολή αυτή δεν ξεκινάει κάποια υπορουτίνα.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής για να δηλώσουμε τον αριθμό της υπορουτίνας.

Παράδειγμα

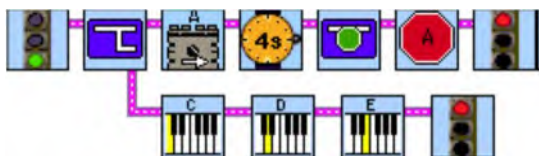


Εκκίνηση Υπορουτίνας (Start Subroutine)

Η εντολή αυτή ξεκινάει την εκτέλεση μίας υπορουτίνας.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής για να δηλώσουμε τον αριθμό της υπορουτίνας που θα ξεκινήσει.

Παράδειγμα





Μουσική



Παίξιμο Νότας (Play Note)

Με την εντολή αυτή το NXT θα παίξει μία νότα. Μπορούμε να ρυθμίσουμε τη διάρκεια της καθώς και την οκτάβα. Η εξ ορισμού τιμή της διάρκειας είναι $\frac{1}{4}$.

Παράδειγμα



Παύση (Rest)

Με την εντολή αυτή το NXT θα εισάγει μία παύση. Μπορούμε να ρυθμίσουμε τη διάρκεια της. Η εξ ορισμού τιμή της διάρκειας της είναι $\frac{1}{4}$.

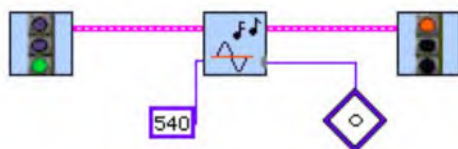
Παράδειγμα



Παίξιμο οποιασδήποτε Νότας (Play Any Note)

Με την εντολή αυτή το NXT θα παίξει μία νότα της οποίας καθορίζουμε τη συχνότητα και τη διάρκεια με αντίστοιχους μετατροπείς σταθερής αριθμητικής τιμής.

Παράδειγμα





Παίξιμο μελωδίας από παρτιτούρα (Play Scroll)

Με την εντολή αυτή το NXT θα παίξει τη μελωδία που βρίσκεται στην αντίστοιχη παρτιτούρα.

Παράδειγμα



Φόρτωση παρτιτούρας από αρχείο (Load scroll from file)

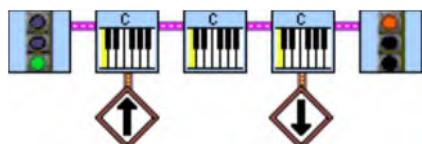
Με την εντολή αυτή το NXT θα φορτώσει τη μελωδία από το αρχείο που θα υποδείξουμε και στη συνέχεια θα παίξει τη μελωδία αυτή.

Παράδειγμα



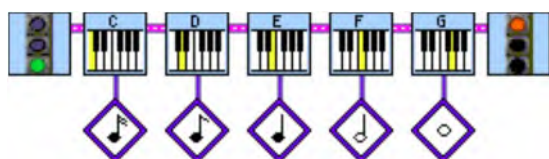
Τροποποιητές καθορισμού οκτάβας για μουσική

Παράδειγμα



Τροποποιητές καθορισμού διάρκειας για μουσική

Παράδειγμα





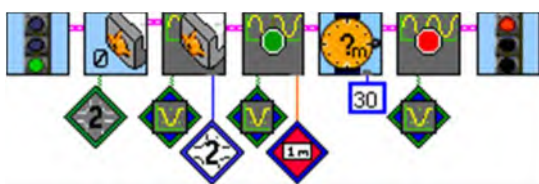
Μηδενισμός (Reset)



NXT Μηδενισμός αισθητήρα αφής (NXT Zero Touch sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα αφής. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

Παράδειγμα



NXT Μηδενισμός Πατημάτων αισθητήρα αφής (NXT Zero Click sensor)

Με την εντολή αυτή το NXT θα μηδενίσει τον αριθμό των πατημάτων που έχει αποθηκευμένο για τον αισθητήρα αφής.

Παράδειγμα



NXT Μηδενισμός Πατημάτων και Απελευθερώσεων αισθητήρα αφής (NXT Zero Touch and Release sensor)

Με την εντολή αυτή το NXT θα μηδενίσει τον αριθμό των πατημάτων και απελευθερώσεων που έχει αποθηκευμένο για τον αισθητήρα αφής.

Παράδειγμα

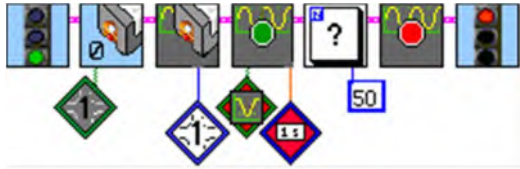




NXT Μηδενισμός αισθητήρα φωτός (NXT Zero Light sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα φωτός. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

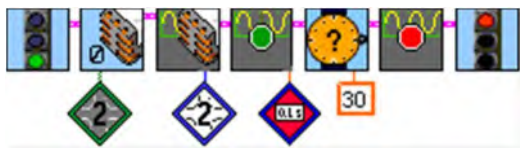
Παράδειγμα



NXT Μηδενισμός αισθητήρα ήχου (NXT Zero Sound sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα ήχου. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

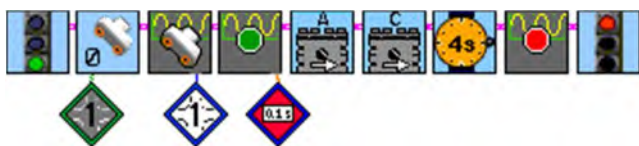
Παράδειγμα



NXT Μηδενισμός αισθητήρα υπερήχων (NXT Zero Ultrasound sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα υπερήχων. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

Παράδειγμα





NXT Μηδενισμός αισθητήρα περιστροφής (NXT Zero Rotation sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα περιστροφής. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

Παράδειγμα



NXT Μηδενισμός αισθητήρα θερμοκρασίας (NXT Zero Temperature sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα θερμοκρασίας. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

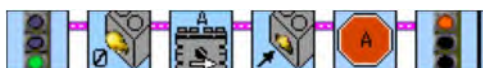
Παράδειγμα



Μηδενισμός αισθητήρα αφής (Zero Touch sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα αφής. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

Παράδειγμα

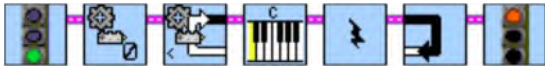




Μηδενισμός Πατημάτων αισθητήρα αφής (Zero Click sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για τον αριθμό των πατημάτων του αισθητήρα αφής.

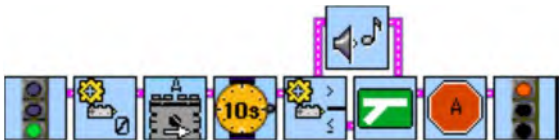
Παράδειγμα



Μηδενισμός Πατημάτων και Απελευθερώσεων αισθητήρα αφής (Zero Touch and Release sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για τον αριθμό των πατημάτων και απελευθερώσεων του αισθητήρα αφής.

Παράδειγμα



Μηδενισμός αισθητήρα φωτός (Zero Light sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα φωτός. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

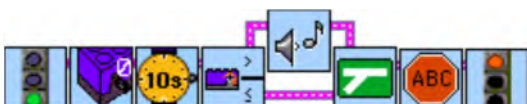
Παράδειγμα



Μηδενισμός αισθητήρα περιστροφής (Zero Rotation sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα περιστροφής. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

Παράδειγμα





Μηδενισμός αισθητήρα θερμοκρασίας (Zero Temperature sensor)

Με την εντολή αυτή το NXT θα μηδενίσει την πληροφορία για την κατάσταση του αισθητήρα θερμοκρασίας. Είναι χρήσιμη όταν θέλουμε να συλλέξουμε δεδομένα.

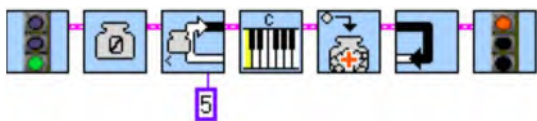
Παράδειγμα



Μηδενισμός μεταβλητής (Zero Container)

Με την εντολή αυτή το NXT θα μηδενίσει την τιμή της αντίστοιχης μεταβλητής. Είναι χρήσιμη όταν θέλουμε να τοποθετήσουμε νέες τιμές σε μία μεταβλητή.

Παράδειγμα



Μηδενισμός χρονομετρητή (Zero Timer)

Με την εντολή αυτή το NXT θα μηδενίσει τον αντίστοιχο χρονομετρητή. Είναι χρήσιμη κάθε φορά που θέλουμε να χρησιμοποιήσουμε ένα χρονομετρητή.

Παράδειγμα





Μηδενισμός γραμματοθυρίδας (Zero Mailbox)

Με την εντολή αυτή το NXT θα αδειάζει το περιεχόμενο της γραμματοθυρίδας. Είναι χρήσιμη κάθε φορά που περιμένουμε νέο μήνυμα.

Παράδειγμα



Μηδενισμός ρολογιού (Zero Clock)

Με την εντολή αυτή το NXT θα μηδενίσει το ρολόι του.

Παράδειγμα





Μεταβλητές (Containers)

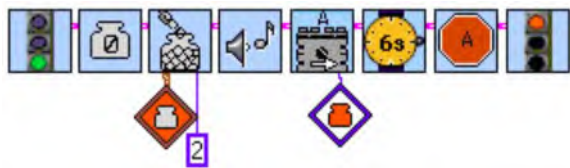


Απόδοση τιμής σε μεταβλητή (Fill Container)

Με την εντολή αυτή το NXT θα αποδώσει μια αριθμητική τιμή η οποία καθορίζεται από έναν τροποποιητή σταθερής αριθμητικής τιμής, σε μία μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 1.

Παράδειγμα

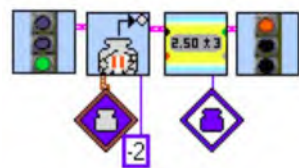


Απόδοση της απόλυτης τιμής σε μεταβλητή (ABS Container)

Με την εντολή αυτή το NXT θα αποδώσει την απόλυτη τιμή μιας αριθμητικής τιμής η οποία καθορίζεται από έναν τροποποιητή σταθερής αριθμητικής τιμής, σε μία μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 0.

Παράδειγμα

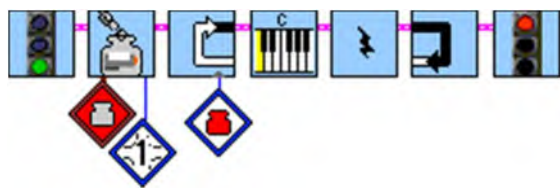


NXT Μεταβλητή αισθητήρα αφής (NXT Touch Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα αφής σε μια μεταβλητή. Αυτή η τιμή θα είναι ή το 0 ή το 1.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

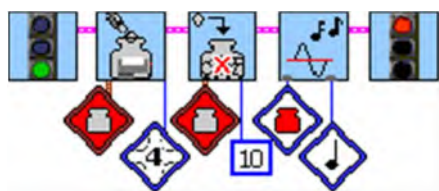
Παράδειγμα



NXT Μεταβλητή αισθητήρα φωτός (NXT Light Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα φωτός σε μια μεταβλητή. Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

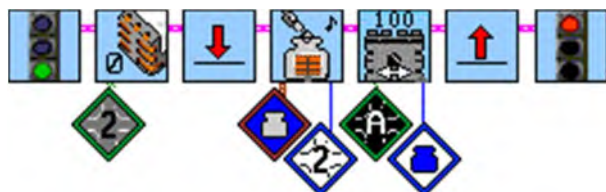
Παράδειγμα



NXT Μεταβλητή αισθητήρα ήχου (NXT Sound Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα ήχου σε μια μεταβλητή. Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα



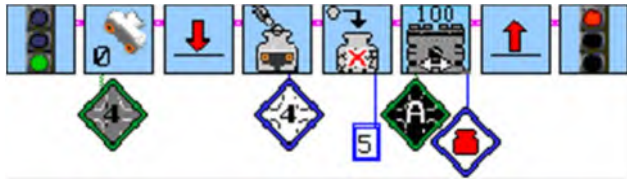


NXT Μεταβλητή αισθητήρα υπερήχων (NXT Ultrasound Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα υπερήχων σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

παράδειγμα



NXT Μεταβλητή αισθητήρα περιστροφής (NXT Rotation Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα περιστροφής σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εξόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εξόδου ή θύρα A και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα

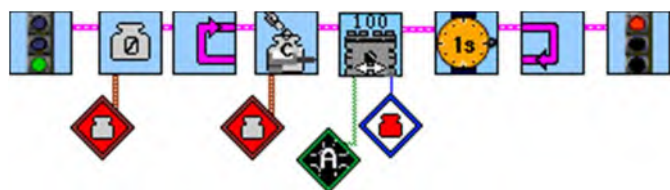


NXT Μεταβλητή αισθητήρα θερμοκρασίας (NXT Temp (C) Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα θερμοκρασίας σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα

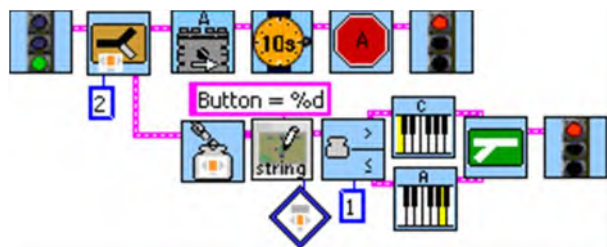


NXT Μεταβλητή των πλήκτρων (NXT Buttons Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα θερμοκρασίας σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα

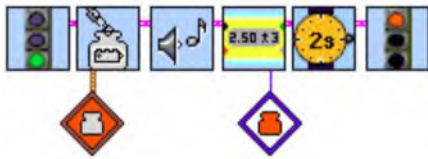


Μεταβλητή αισθητήρα αφής (Touch Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα αφής σε μια μεταβλητή. Αυτή η τιμή θα είναι ή το 0 ή το 1.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα

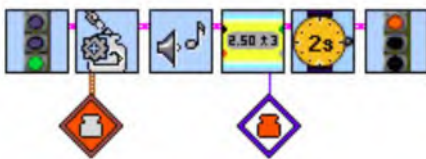


Μεταβλητή Πατημάτων αισθητήρα αφής (Click Container)

Με την εντολή αυτή το NXT θα αποδώσει τον αριθμό των πατημάτων του αισθητήρα αφής σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα

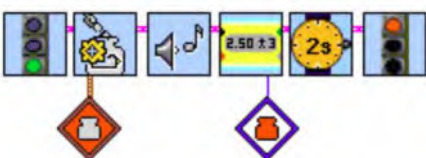


Μεταβλητή Πατημάτων και Απελευθερώσεων αισθητήρα αφής (Touch and Release Container)

Με την εντολή αυτή το NXT θα αποδώσει τον αριθμό των πατημάτων και απελευθερώσεων του αισθητήρα αφής σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα





Μεταβλητή αισθητήρα φωτός (Light Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα αφής σε μια μεταβλητή. Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα

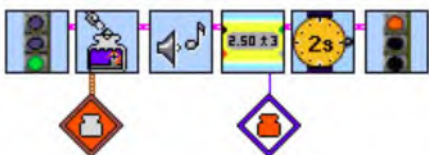


Μεταβλητή αισθητήρα περιστροφής (Rotation Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα περιστροφής σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

Παράδειγμα



Μεταβλητή αισθητήρα θερμοκρασίας (Temp (C) Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του αισθητήρα θερμοκρασίας σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή θύρας εισόδου και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως θύρα εισόδου ή θύρα 1 και ως μεταβλητή η κόκκινη μεταβλητή.

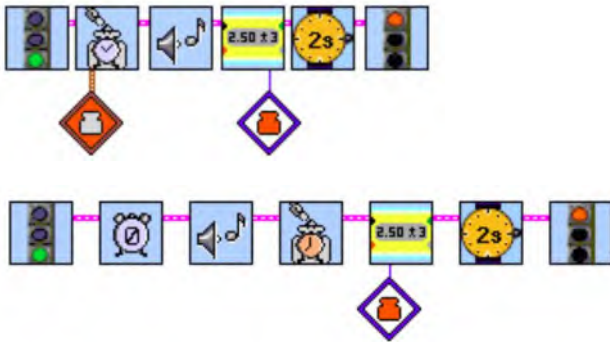
Παράδειγμα



Μεταβλητή Χρονομετρητή (Timer Value Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του χρονομετρητή σε μια μεταβλητή. Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή χρονομετρητή και ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως χρονομετρητής ο κόκκινος χρονομετρητής και ως μεταβλητή η κόκκινη μεταβλητή.

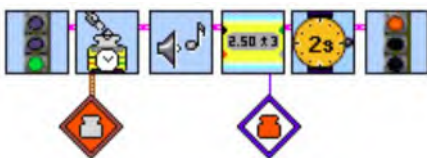
Παραδείγματα



Μεταβλητή Ρολογιού (Clock Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή του ρολογιού του σε μια μεταβλητή. Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται εξ ορισμού η κόκκινη μεταβλητή.

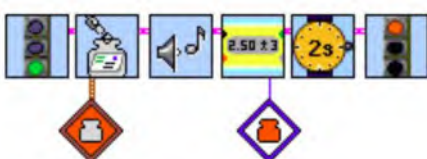
Παράδειγμα



Μεταβλητή Μηνύματος (Mail Container)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή της γραμματοθυρίδας σε μια μεταβλητή. Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται εξ ορισμού η κόκκινη μεταβλητή.

Παράδειγμα



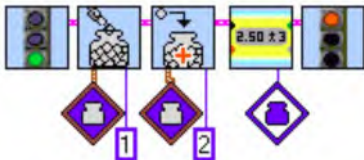


Πρόσθεση σε μεταβλητή (Add to Container)

Με την εντολή αυτή το NXT θα προσθέσει μια αριθμητική τιμή η οποία καθορίζεται από έναν τροποποιητή σταθερής αριθμητικής τιμής, στο περιεχόμενο που υπάρχει ήδη σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 1.

Παράδειγμα

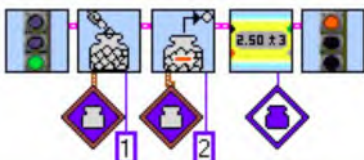


Αφαίρεση από μεταβλητή (Remove from Container)

Με την εντολή αυτή το NXT θα αφαιρέσει μια αριθμητική τιμή η οποία καθορίζεται από έναν τροποποιητή σταθερής αριθμητικής τιμής, από το περιεχόμενο που υπάρχει ήδη σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 1.

Παράδειγμα



Πολλαπλασιασμός σε μεταβλητή (Multiply to Container)

Με την εντολή αυτή το NXT θα πολλαπλασιάσει μια αριθμητική τιμή η οποία καθορίζεται από έναν τροποποιητή σταθερής αριθμητικής τιμής, με το περιεχόμενο που υπάρχει ήδη σε μια μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 2.

Παράδειγμα

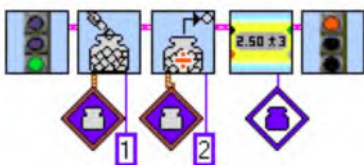


Διαίρεση μεταβλητής (Divide Container)

Με την εντολή αυτή το NXT θα διαιρέσει το περιεχόμενο που υπάρχει ήδη σε μια μεταβλητή με μια αριθμητική τιμή, η οποία καθορίζεται από έναν τροποποιητή σταθερής αριθμητικής τιμής.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 2.

Παράδειγμα

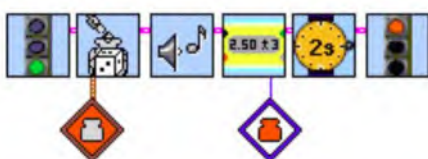


Απόδοση Τυχαίας τιμής σε μεταβλητή (Random Fill Container)

Με την εντολή αυτή το NXT θα αποδώσει μια τυχαία αριθμητική τιμή σε μία μεταβλητή. Η τιμή αυτή θα βρίσκεται ανάμεσα στο 0 και σε μία μέγιστη αριθμητική τιμή που ορίζεται από εμάς.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 8.

Παράδειγμα





Απόδοση τιμής ενός μαθηματικού τύπου σε μεταβλητή (Formula Container)

Με την εντολή αυτή το NXT θα αποδώσει το αποτέλεσμα μιας αριθμητικής παράστασης που ορίζεται από εμάς σε μία μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού η κόκκινη μεταβλητή.

Για να καθορίσουμε τη μαθηματική έκφραση που θέλουμε να υπολογιστεί, κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant), όπου και πληκτρολογούμε την έκφραση. Στο έκφραση αυτή μπορούμε να χρησιμοποιήσουμε τις τιμές των μεταβλητών ως εξής :

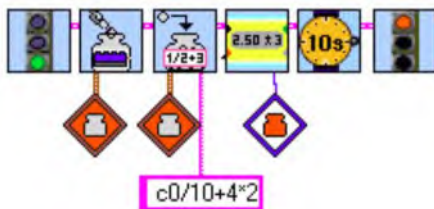
c0 ή red για την κόκκινη μεταβλητή

c1 ή blue για την μπλε μεταβλητή

c2 ή yellow για την κίτρινη μεταβλητή

c3-22 για τις γενικές μεταβλητές

παράδειγμα



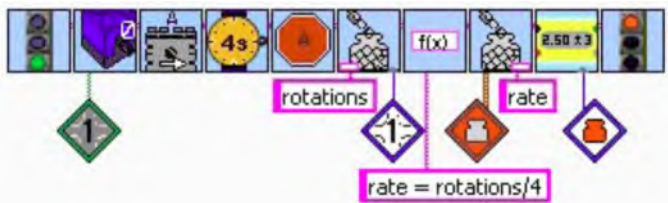
Απόδοση τιμής σε οριζόμενη μεταβλητή (Fill String Container)

Με την εντολή αυτή το NXT θα ορίσει μία νέα μεταβλητή με το όνομα που θα δώσουμε και θα αποδώσει σε αυτήν μια αριθμητική τιμή ή την τιμή από ένα τροποποιητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής ή κάποιον άλλο τροποποιητή.

Για να καθορίσουμε το όνομα της μεταβλητής κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant), όπου και πληκτρολογούμε το όνομα.

Παράδειγμα





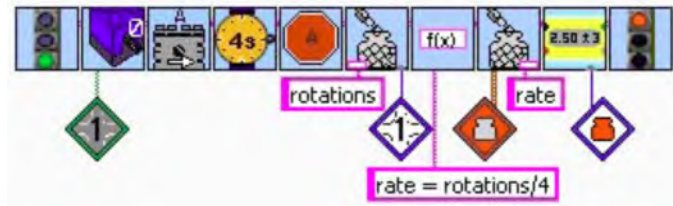
Απόδοση τιμής μεταβλητής σε μεταβλητή (Fill Container with Variable)

Με την εντολή αυτή το NXT θα αποδώσει την αριθμητική τιμή μίας οριζόμενης μεταβλητής σε μία μεταβλητή. Με την εντολή αυτή γίνεται μετατροπή του δεκαδικού αριθμού σε ακέραιο.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής.

Για να καθορίσουμε το όνομα της μεταβλητής κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant), όπου και πληκτρολογούμε το όνομα.

Παράδειγμα

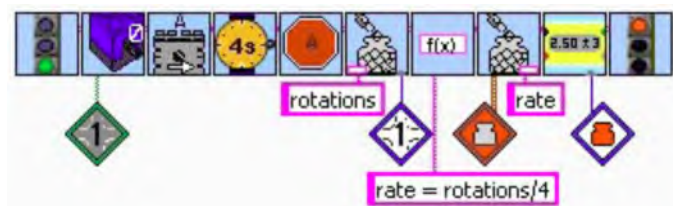


Υπολογισμός Συνάρτησης (Evaluate Expression)

Με την εντολή αυτή το NXT θα ορίσει μία νέα μεταβλητή με το όνομα που θα δώσουμε και θα αποδώσει σε αυτήν το αποτέλεσμα μίας αριθμητικής έκφρασης που θα ορίσουμε πάλι εμείς. Σε αυτήν την εντολή επιτρέπεται και η χρήση δεκαδικών αριθμών.

Για να καθορίσουμε τη μαθηματική έκφραση που θέλουμε να υπολογιστεί κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant), όπου και πληκτρολογούμε την έκφραση. Στο έκφραση αυτή μπορούμε να χρησιμοποιήσουμε τις τιμές των μεταβλητών ως εξής :

Παράδειγμα





Επικοινωνία μεταξύ των NXT (*NXT Communication*)

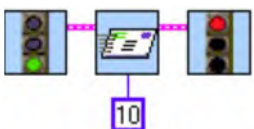


Αποστολή μηνύματος (Send mail)

Με την εντολή αυτή το NXT θα στείλει μια αριθμητική τιμή ως μήνυμα. Η αποστολή αυτή γίνεται μέσω της υπέρυθρης διασύνδεσης σε οποιοδήποτε άλλο μικροϋπολογιστή NXT θα βρεθεί μέσα στην εμβέλεια του σήματος.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται εξ ορισμού ως αριθμητική τιμή το 1.

Παράδειγμα



Αναμονή για μήνυμα (Wait for mail)

Αυτή η εντολή κάνει το NXT να περιμένει μέχρι να λάβει κάποιο μήνυμα από ένα άλλο μικροϋπολογιστή NXT μέσω της θύρας υπέρυθρων του.

Παράδειγμα

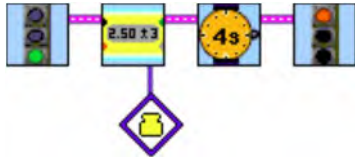


Καθορισμός για εμφάνιση (Set Display)

Με την εντολή αυτή το NXT θα εμφανίσει στην οθόνη του την τιμή ενός τροποποιητή. Η εντολή αυτή είναι χρήσιμη για την ανεύρεση λαθών του προγράμματος μας.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού τροποποιητής η κόκκινη μεταβλητή.

Παράδειγμα



Αλλαγή εμφάνισης (Change View)

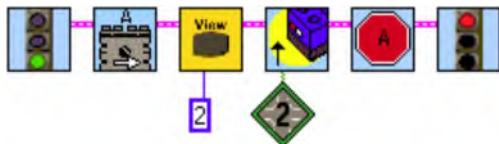
Αυτή η εντολή αλλάζει την παράμετρο που θα εμφανίζεται στην οθόνη του NXT.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής από το 0 έως το 6. Αυτό που θα εμφανίζεται εξαρτάται από το αριθμό που έχουμε προσθέσει στον τροποποιητή σταθερής αριθμητικής τιμής ως εξής :

0 : Εσωτερικό Ρολόι

1,2,3 : Θύρες εισόδου

Παράδειγμα

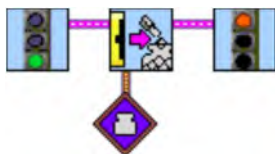


Απόδοση τιμής σε μεταβλητή από απόσταση (Fill Remote Container)

Με την εντολή αυτή το NXT θα αποδώσει μια αριθμητική τιμή η οποία καθορίζεται από έναν τροποποιητή σταθερής αριθμητικής τιμής, σε μία μεταβλητή που ανήκει σε ένα άλλο μικροϋπολογιστή NXT. Αυτή η εντολή είναι χρήσιμη όταν έχουμε να στείλουμε πολλές τιμές σε ένα άλλο μικροϋπολογιστή NXT.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή μεταβλητής και ένα τροποποιητή σταθερής αριθμητικής τιμής. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρούνται εξ ορισμού ως μεταβλητή η κόκκινη μεταβλητή και ως αριθμητική τιμή το 1.

Παράδειγμα





Εντολές NXT



NXT Συγχρονισμός κινητήρων (NXT Synch Motor Control)

Με αυτήν την εντολή μπορούμε να συγχρονίσουμε ένα ζευγάρι κινητήρων. Αυτό σημαίνει ότι και οι δύο κινητήρες θα γυρίζουν με την ίδια ταχύτητα έτσι ώστε το ρομπότ μας να κινείται με ακρίβεια σε ευθεία γραμμή.

Σε αυτή την εντολή πρέπει να προσθέσουμε δύο μετατροπείς θύρας εξόδου για να καθορίσουμε τους κινητήρες που επιθυμούμε να είναι συγχρονισμένοι.

Μέσα στο πρόγραμμα όταν θέλουμε να δώσουμε εντολές κίνησης στο ζευγάρι των κινητήρων για τους οποίους έχουμε καθορίσει να κινούνται συγχρονισμένα χρειάζεται να τοποθετήσουμε την αντίστοιχη εντολή μόνο για τον ένα από τους δύο αυτούς κινητήρες, τον κινητήρα ελέγχου.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής με τον οποίο μπορούμε να ρυθμίσουμε τη σχέση συγχρονισμού των κινητήρων μας ως ένα ποσοστό. Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού αριθμητική τιμή το 100 %.

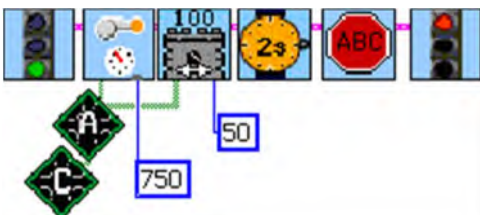
Παράδειγμα



NXT Έλεγχος ταχύτητας κινητήρων (NXT Speed Control)

Με αυτήν την εντολή μπορούμε να καθορίσουμε μία σταθερά τιμή για την ταχύτητα των κινητήρων μας. Η μέγιστη τιμή που μπορούμε να δώσουμε είναι 750 στροφές το δευτερόλεπτο. Στη συνέχεια κάθε φορά που θα χρησιμοποιήσουμε μία εντολή κινητήρα, αντί για το επίπεδο ισχύος του κινητήρα, ορίζουμε ένα ποσοστό της σταθερής τιμής που έχουμε καθορίσει πιο μπροστά με την εντολή αυτήν.

Παράδειγμα

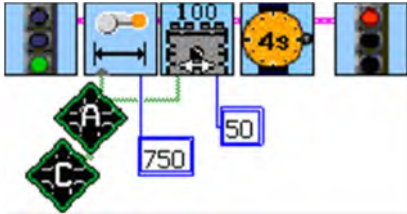




NXT Έλεγχος απόστασης κινητήρα (NXT Distance Control)

Με αυτήν την εντολή μπορούμε να καθορίσουμε μία σταθερά τιμή για τη γωνία περιστροφής των κινητήρων μας σε μοίρες. Στην αρχή του προγράμματος μας μπορούμε να καθορίσουμε διαφορετική απόσταση για κάθε κινητήρα και στη συνέχεια να τους ξεκινήσουμε όλους μαζί.

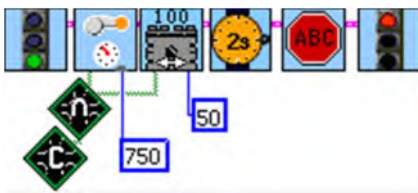
Παράδειγμα



NXT Αλλαγή κατεύθυνσης κινητήρα (NXT Flip Motor Orientation)

Με αυτήν την εντολή μπορούμε να αλλάξουμε τη φορά περιστροφής ενός κινητήρα.

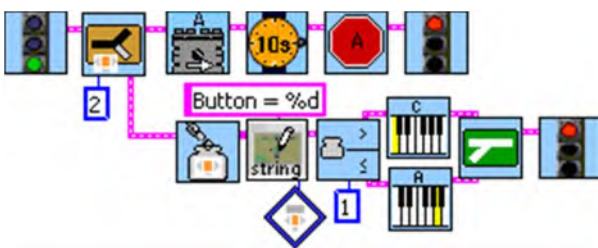
Παράδειγμα



NXT Έλεγχος παράλληλης διεργασίας μέσω των πλήκτρων (NXT Button Control)

Με αυτήν την εντολή μπορούμε να ελέγξουμε να ξεκινήσει η εκτέλεση μίας παράλληλης διεργασίας όταν πατηθεί ένα πλήκτρο του NXT. Σε αυτήν την εντολή μπορούμε να καθορίσουμε και πόσες φορές πρέπει να πατηθεί συνεχόμενα το πλήκτρο ακύρωσης για να ακυρώσουμε την εκτέλεση της παράλληλης διεργασίας.

Παράδειγμα





NXT Αλλαγή εμφάνισης (NXT Change View)

Αυτή η εντολή αλλάζει την παράμετρο που θα εμφανίζεται στην οθόνη του NXT.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής από το 0 έως το 3. Αυτό που θα εμφανίζεται εξαρτάται από το αριθμό που έχουμε προσθέσει στον τροποποιητή σταθερής αριθμητικής τιμής ως εξής :

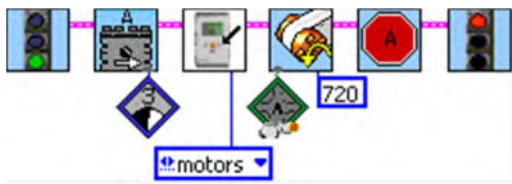
0 : Κανονική προβολή

1 : Τιμές των αισθητήρων

2 : Ταχύτητα των κινητήρων

3 : Γράφημα της δειγματοληψίας

Παράδειγμα



NXT Τιμή από τα πλήκτρα (Value of NXT Buttons)

Καθορίζει ως τιμή μιας εντολής την τιμή που αντιστοιχεί για το πλήκτρο του NXT που πατήθηκε :

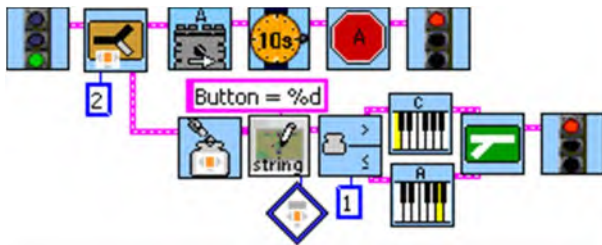
0= γκρι πλήκτρο ακύρωσης,

1= δεξιό πλήκτρο,

2=αριστερό πλήκτρο,

3=πορτοκαλί πλήκτρο enter

Παράδειγμα

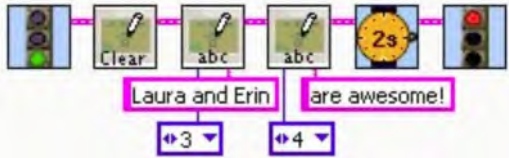




NXT Καθαρισμός οθόνης (NXT Clear Display)

Αυτή η εντολή καθαρίζει την οθόνη του NXT. Χρησιμοποιείται συνήθως πριν από κάποια εντολή εμφάνισης.

Παράδειγμα



NXT Εμφάνιση Εικόνας (NXT Draw Bitmap)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT την εικόνα ενός αρχείου .bmp.

Σε αυτή την εντολή πρέπει να προσθέσουμε τη διαδρομή του αρχείου ως μία σταθερά κάνοντας δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγοντας τη δημιουργία σταθεράς (create→constant).

Παράδειγμα

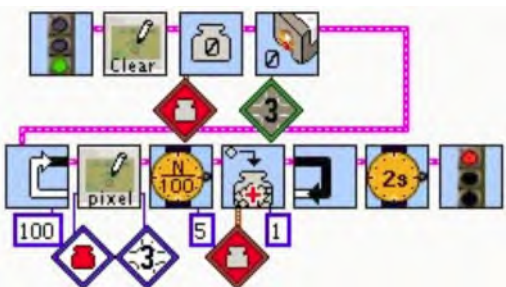


NXT Σχεδίαση τελείας (NXT Draw Pixel)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT μία μαύρη τελεία.

Σε αυτή την εντολή πρέπει να προσθέσουμε δύο μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για το σημείο της οθόνης στο οποίο θα εμφανιστεί η τελεία. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64). Μπορούμε να δώσουμε τις συντεταγμένες και ως τιμή αισθητήρα ή μεταβλητής ή ως σταθερά. Για να δημιουργήσουμε μία σταθερά κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant).

Παράδειγμα





NXT Σβήσιμο τελείας (NXT Clear Pixel)

Αυτή η εντολή σβήνει από την οθόνη του NXT μία μαύρη τελεία.

Σε αυτή την εντολή πρέπει να προσθέσουμε δύο μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για το σημείο της οθόνης στο οποίο θα σβήσουμε την τελεία. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64).

Παράδειγμα

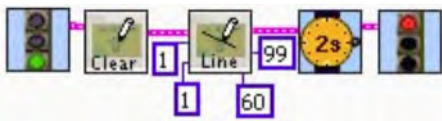


NXT Σχεδίαση Γραμμής (NXT Draw Line)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT μία γραμμή.

Σε αυτή την εντολή πρέπει να προσθέσουμε δύο μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για τα σημεία της οθόνης για την αρχή και το τέλος της γραμμής αυτής. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64). Μπορούμε να δώσουμε τις συντεταγμένες και ως τιμή αισθητήρα ή μεταβλητής ή ως σταθερά. Για να δημιουργήσουμε μία σταθερά κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant).

Παράδειγμα



NXT Σχεδίαση Ορθογώνιου Παραλληλόγραμμου (NXT Draw Rectangle)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT ένα ορθογώνιο παραλληλόγραμμο.

Σε αυτή την εντολή πρέπει να προσθέσουμε τέσσερις μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για τα σημεία της οθόνης για την κάτω αριστερή και την πάνω δεξιά γωνία αντίστοιχα του ορθογωνίου αυτού. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64).



NXT Σχεδίαση Βαμμένου Ορθογώνιου Παραλληλόγραμμου (NXT Draw Filled Rectangle)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT ένα ορθογώνιο παραλληλόγραμμο.

Σε αυτή την εντολή πρέπει να προσθέσουμε τέσσερις μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για τα σημεία της οθόνης για την κάτω αριστερή και την πάνω δεξιά γωνία αντίστοιχα του ορθογώνιου αυτού. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64).

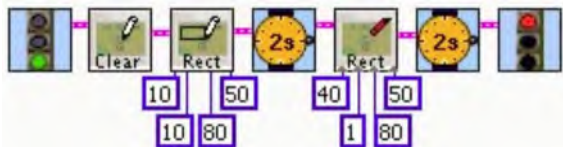


NXT Σβήσιμο Ορθογώνιου Παραλληλόγραμμου (NXT Erase Rectangle)

Αυτή η εντολή σβήνει από την οθόνη του NXT ένα ορθογώνιο παραλληλόγραμμο.

Σε αυτή την εντολή πρέπει να προσθέσουμε τέσσερις μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για τα σημεία της οθόνης για την κάτω αριστερή και την πάνω δεξιά γωνία αντίστοιχα του ορθογώνιου αυτού. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64).

Παράδειγμα



NXT Σχεδίαση Έλλειψης (NXT Draw Ellipse)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT μία έλλειψη.

Σε αυτή την εντολή πρέπει να προσθέσουμε τέσσερις μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για τα σημεία της οθόνης για την κάτω αριστερή και την πάνω δεξιά γωνία αντίστοιχα του ορθογώνιου που περικλείει την έλλειψη. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64).



NXT Σχεδίαση Βαμμένης Έλλειψης (NXT Draw Filled Ellipse)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT μία έλλειψη.

Σε αυτή την εντολή πρέπει να προσθέσουμε τέσσερις μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για τα σημεία της οθόνης για την κάτω αριστερή και την πάνω δεξιά γωνία αντίστοιχα του ορθογωνίου που περικλείει την έλλειψη. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64).

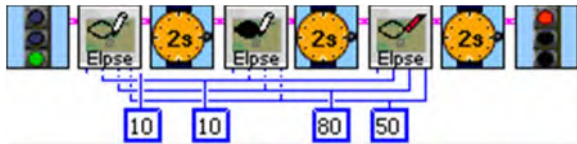


NXT Σβήσιμο Έλλειψης (NXT Erase Ellipse)

Αυτή η εντολή σβήνει από την οθόνη του NXT μία έλλειψη.

Σε αυτή την εντολή πρέπει να προσθέσουμε τέσσερις μετατροπείς σταθερής αριθμητικής τιμής με τους οποίους μπορούμε να καθορίσουμε τις συντεταγμένες (x,y) για τα σημεία της οθόνης για την κάτω αριστερή και την πάνω δεξιά γωνία αντίστοιχα του ορθογωνίου που περικλείει την έλλειψη. Το σημείο (0, 0) αντιστοιχεί στην κάτω αριστερή γωνία της οθόνης, ενώ η πάνω δεξιά γωνία της οθόνης αντιστοιχεί στο σημείο (100, 64).

Παράδειγμα

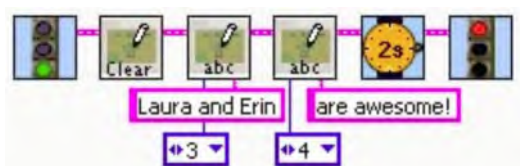


NXT Εμφάνιση κειμένου (NXT Write Text)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT ένα κείμενο. Για να καθορίσουμε το κείμενο που θέλουμε να εμφανιστεί κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant), όπου και πληκτρολογούμε το κείμενο μας.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής τιμής για να καθορίσουμε τη γραμμή από το 0 ως το 7 που θα εμφανιστεί στην οθόνη.

Παράδειγμα





NXT Εμφάνιση μορφοποιημένου κείμενου (NXT Formatted Text)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT ένα μορφοποιημένο κείμενο το οποίο περιέχει και την τιμή μιας μεταβλητής.

Για να καθορίσουμε το κείμενο που θέλουμε να εμφανιστεί κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant), όπου και πληκτρολογούμε το κείμενο μας. Στο κείμενο αυτό παρεμβάλλεται στο σημείο όπου τοποθετούμε τις παρακάτω παραμέτρους :

%d για εμφάνιση ακέραιας τιμής

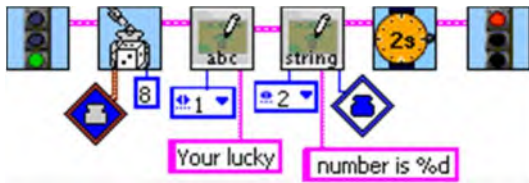
%0.1f για εμφάνιση πραγματικής τιμής με 1 δεκαδικό

%c για εμφάνιση χαρακτήρα,

η τιμή της μεταβλητής που έχουμε καθορίσει.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής τιμής για να καθορίσουμε τη γραμμή από το 0 ως το 7 που θα εμφανιστεί στην οθόνη.

Παράδειγμα



NXT Εμφάνιση μορφοποιημένου κείμενου επεκτεινόμενη (NXT Expanded Formatted Text)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT ένα μορφοποιημένο κείμενο το οποίο περιέχει και τιμές από μέχρι και τρεις μεταβλητές.

Για να καθορίσουμε το κείμενο που θέλουμε να εμφανιστεί κάνουμε δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγουμε τη δημιουργία σταθεράς (create→constant), όπου και πληκτρολογούμε το κείμενο μας. Στο κείμενο αυτό παρεμβάλλονται στα σημεία όπου τοποθετούμε τις παρακάτω παραμέτρους :

%d για εμφάνιση ακέραιας τιμής

%0.1f για εμφάνιση πραγματικής τιμής με 1 δεκαδικό

%c για εμφάνιση χαρακτήρα,

οι τιμές από τις μεταβλητές τις οποίες έχουμε καθορίσει

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής τιμής για να καθορίσουμε τη γραμμή από το 0 ως το 7 που θα εμφανιστεί στην οθόνη.

Παράδειγμα



NXT Εμφάνιση τροποποιητή (NXT Write Modifier)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT τη λίστα με τους τροποποιητές.



NXT Εμφάνιση μεταβλητής (NXT Write Container)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT μία λίστα με τις μεταβλητές.



NXT Αναπαραγωγή Ήχου (NXT Play Sound File)

Αυτή η εντολή αναπαράγει ένα ήχο ενός αρχείου .rso.

Σε αυτή την εντολή πρέπει να προσθέσουμε τη διαδρομή του αρχείου ως μία σταθερά κάνοντας δεξί κλικ στο κάτω δεξιό μέρος του εικονιδίου και επιλέγοντας τη δημιουργία σταθεράς (create→constant).

Παράδειγμα





NXT Λίστα Αρχείου (NXT File Listing)

Αυτή η εντολή εμφανίζει στην οθόνη του NXT μία λίστα με τα αρχεία που βρίσκονται φορτωμένα στη μνήμη του NXT.



NXT Get NXT File

Αυτή η εντολή επιστρέφει τα δεδομένα ενός αρχείου.



NXT Εκτέλεση αρχείου (Run NXT File)

Με αυτή την εντολή τρέχουμε ένα αρχείο προγράμματος.



NXT Σβήσιμο αρχείου (Delete NXT File)

Με αυτή την εντολή σβήνουμε ένα αρχείο.

Παράδειγμα





Τροποποιητές (Modifiers)

Τους τροποποιητές τους προσθέτουμε σε ένα πλήθος εντολών, συνδέοντας τους στην αντίστοιχη θέση των εικονιδίων των εντολών, ώστε να καθορίσουμε τις παραμέτρους των εντολών αυτών.



Έξοδος A/B/C (Output A/B/C)

Καθορίζει την έξοδο του NXT στην οποία θα ενεργήσει μια εντολή εξόδου.



Είσοδος 1/2/3/4 (Input 1/2/3/4)

Καθορίζει την είσοδο του NXT από την οποία μια εντολή θα διαβάσει την εισερχόμενη τιμή.

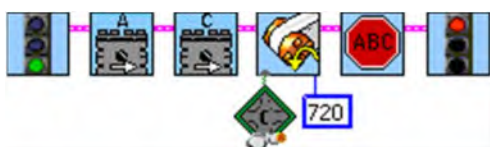
Παράδειγμα



Είσοδος A/B/C (Encoder A/B/C)

Καθορίζει την είσοδο του NXT από την οποία μια εντολή θα διαβάσει την εισερχόμενη τιμή ενός αισθητήρα περιστροφής.

Παράδειγμα

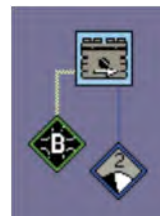




Επίπεδο Ισχύος 1-5 (Power Level 1-5)

Επηρεάζει την ταχύτητα με την οποία θα περιστραφεί ένας κινητήρας.
Καθορίζει την ισχύ με την οποία θα κινηθεί ένας κινητήρας.

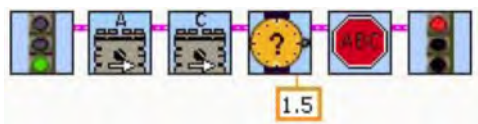
Παράδειγμα



Αριθμητική σταθερή τιμή (Numeric Constant)

Οι τροποποιητές σταθερής αριθμητικής τιμής τοποθετούνται σε αρκετά εικονίδια εντολών στις οποίες χρειάζεται να ορισθεί μια αριθμητική τιμή.

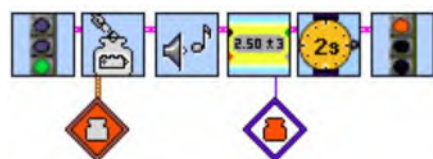
Παράδειγμα



Κόκκινη/Μπλε/Κίτρινη Μεταβλητή (Red/Blue/Yellow Container)

Οι τροποποιητές μεταβλητής τοποθετούνται στα εικονίδια εντολών μεταβλητών και σε άλλες στις οποίες χρειάζεται να ορισθεί μια συγκεκριμένη μεταβλητή.

Παράδειγμα





Γενική Μεταβλητή (Generic Container)

Οι τροποποιητές γενικής μεταβλητής τοποθετείται στα εικονίδια εντολών μεταβλητών και σε άλλες στις οποίες χρειάζεται να ορισθεί μια συγκεκριμένη μεταβλητή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής στον οποίο καθορίζουμε έναν αριθμό ως ταυτότητα της συγκεκριμένης μεταβλητής ως εξής :

0 για την κόκκινη μεταβλητή

1 για την μπλε μεταβλητή

2 για την κίτρινη μεταβλητή

3-22 για τις γενικές μεταβλητές

Το 23-47 αντιστοιχούν σε ειδικές μεταβλητές

Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού, η γενική μεταβλητή 3.

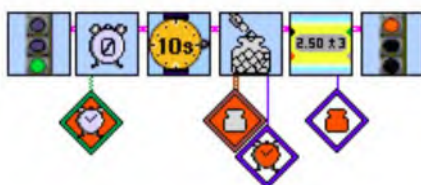
Παράδειγμα



Κόκκινος/Μπλε/Κίτρινος Χρονομετρητή (Red/Blue/Yellow Timer)

Οι τροποποιητές χρονομετρητή τοποθετούνται στα εικονίδια εντολών χρονομετρητή και σε άλλες στις οποίες χρειάζεται να ορισθεί ένας συγκεκριμένος χρονομετρητής.

Παράδειγμα



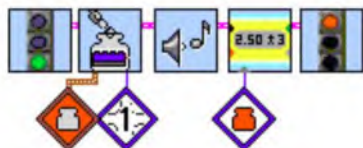
Σενάρια Ρομποτικής με το Mindstorms NXT



Τιμή στην είσοδο 1/2/3/4 (Value on Port 1/2/3/4)

Καθορίζει την τιμή από την είσοδο του NXT από την οποία μια εντολή θα διαβάσει την εισερχόμενη τιμή.

Παράδειγμα



Τιμή στην είσοδο A/B/C (Value on Encoder A/B/C)

Καθορίζει την τιμή από την είσοδο του NXT από την οποία μια εντολή θα διαβάσει την εισερχόμενη τιμή ενός αισθητήρα περιστροφής.

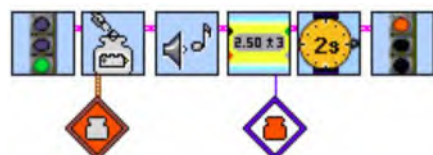
Παράδειγμα



Τιμή στην Κόκκινη/Μπλε/Κίτρινη Μεταβλητή (Value of Red/Blue/Yellow Container)

Καθορίζει τη μεταβλητή της οποίας μια εντολή θα χρησιμοποιήσει την τιμή.

Παράδειγμα





Τιμή στη γενική Μεταβλητή (Value of Generic Container)

Καθορίζει τη μεταβλητή της οποίας μια εντολή θα χρησιμοποιήσει την τιμή.

Σε αυτή την εντολή πρέπει να προσθέσουμε ένα τροποποιητή σταθερής αριθμητικής τιμής στον οποίο καθορίζουμε έναν αριθμό ως ταυτότητα της συγκεκριμένης μεταβλητής ως εξής :

0 για την κόκκινη μεταβλητή

1 για την μπλε μεταβλητή

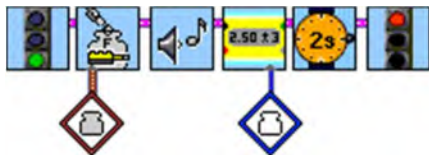
2 για την κίτρινη μεταβλητή

3-22 για τις γενικές μεταβλητές

Το 23-47 αντιστοιχούν σε ειδικές μεταβλητές

Αν δεν προσθέσουμε κανένα τροποποιητή τότε θεωρείται ως εξ ορισμού, η γενική μεταβλητή 3.

Παράδειγμα



Τιμή στην Κόκκινο/Μπλε/Κίτρινο Χρονομετρητή (Value of Red/Blue/Yellow Timer)

Καθορίζει το χρονομετρητή του οποίου μια εντολή θα χρησιμοποιήσει την τιμή.

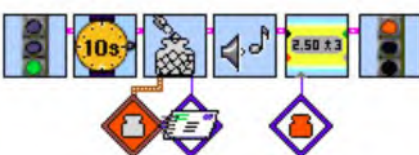
Παράδειγμα



Τιμή μηνύματος (Value of Mail)

Καθορίζει ως τιμή μιας εντολή την τιμή του μηνύματος.

Παράδειγμα

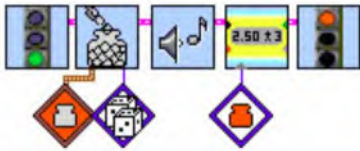




Τυχαία τιμή (Random Value)

Παράγει μία τυχαία τιμή.

Παράδειγμα



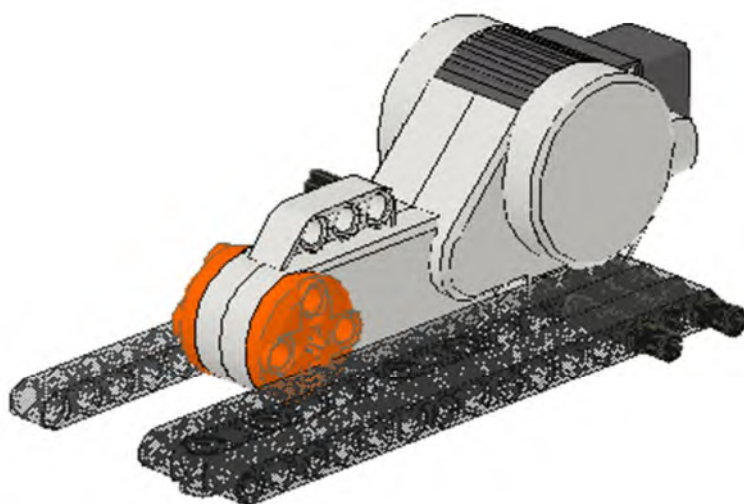
ΠΑΡΑΡΤΗΜΑ III

Κατασκευή οχήματος – ρομπότ TankBot NXT

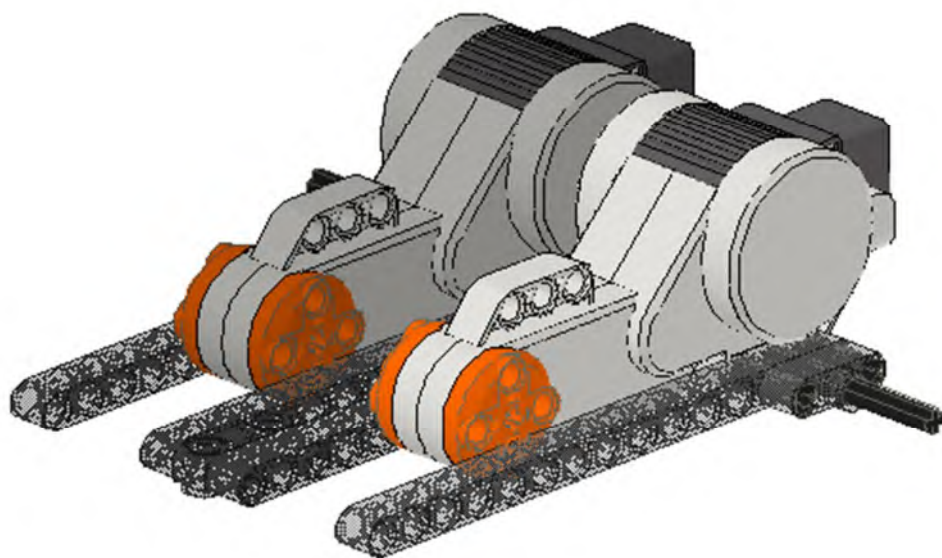
1



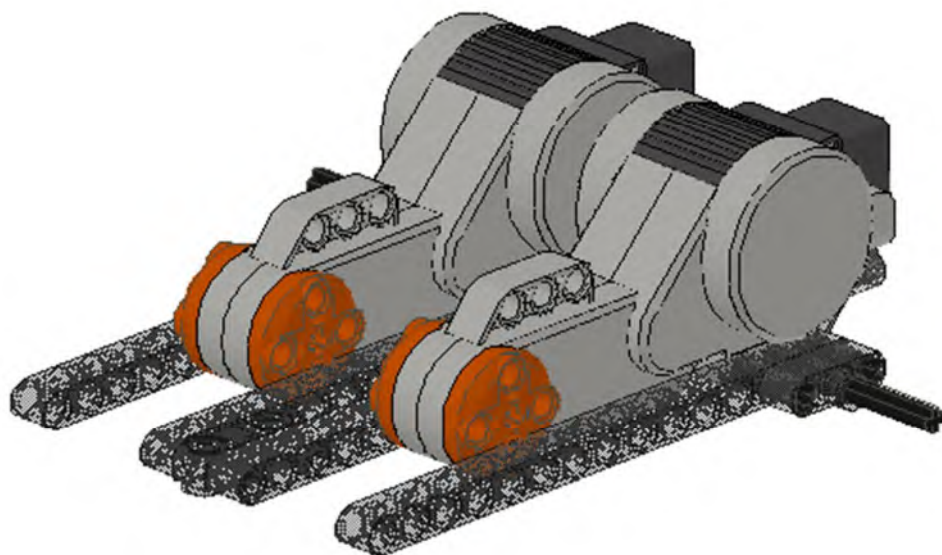
2



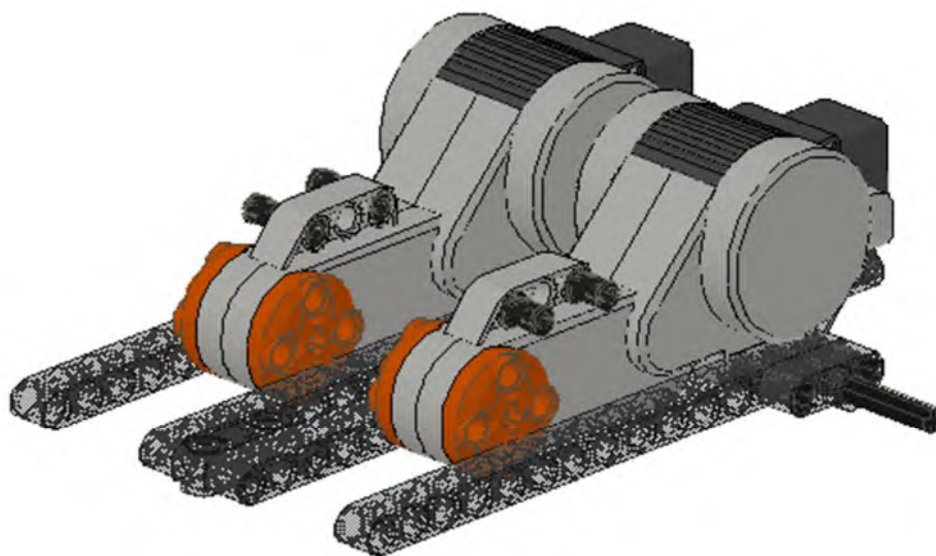
3



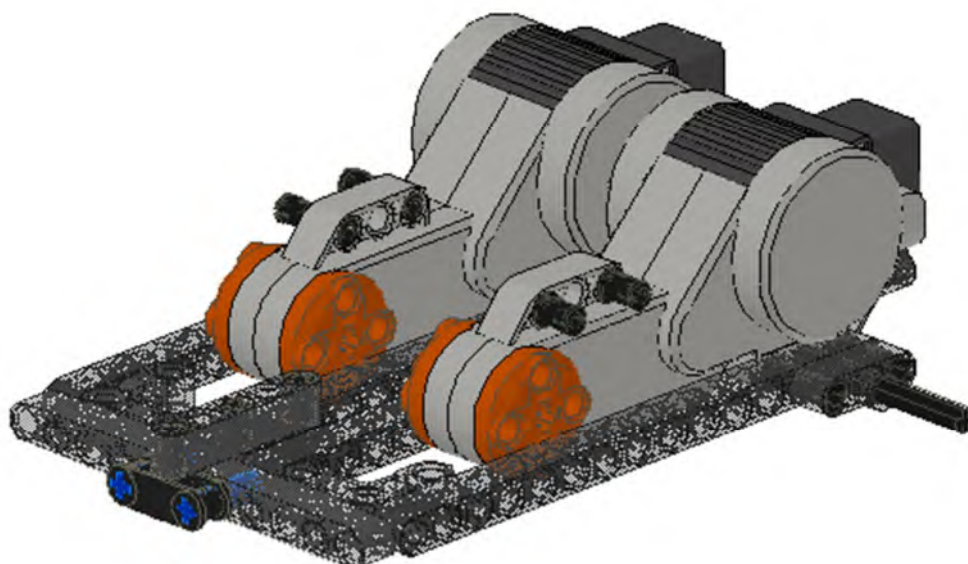
4



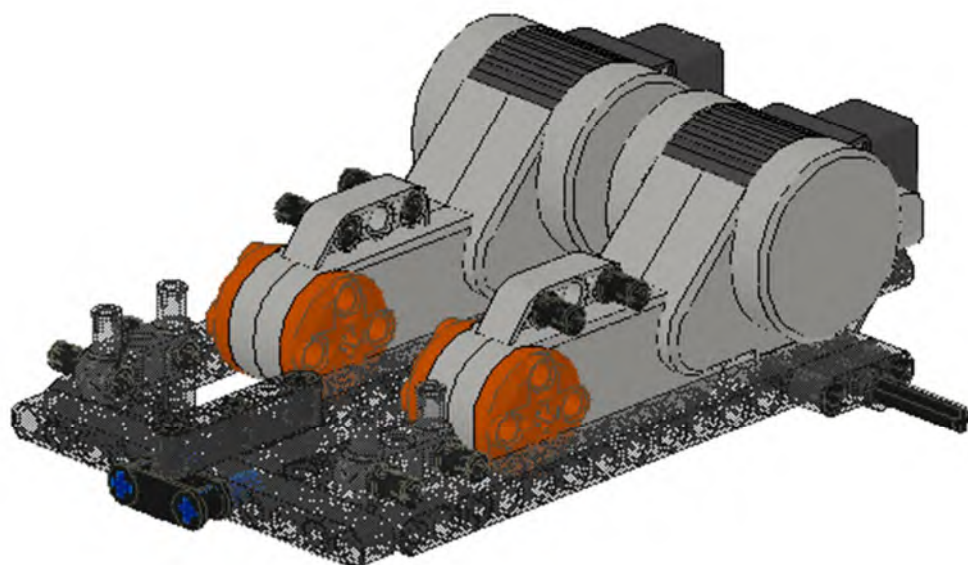
5



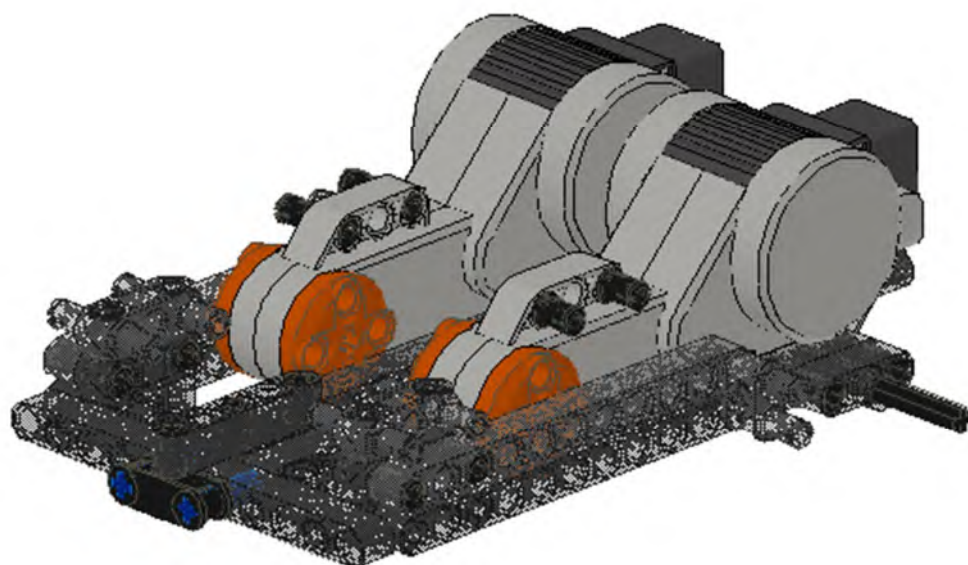
6



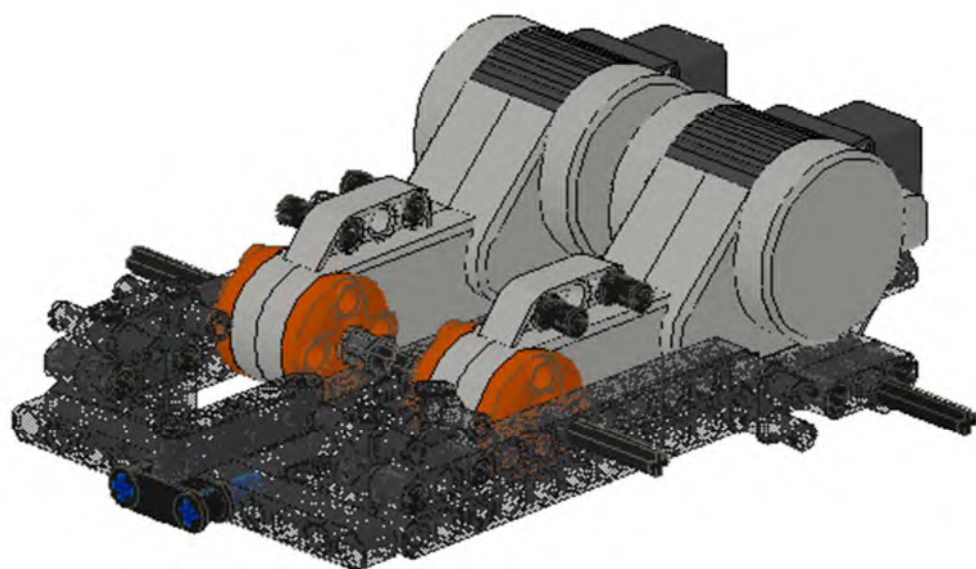
7



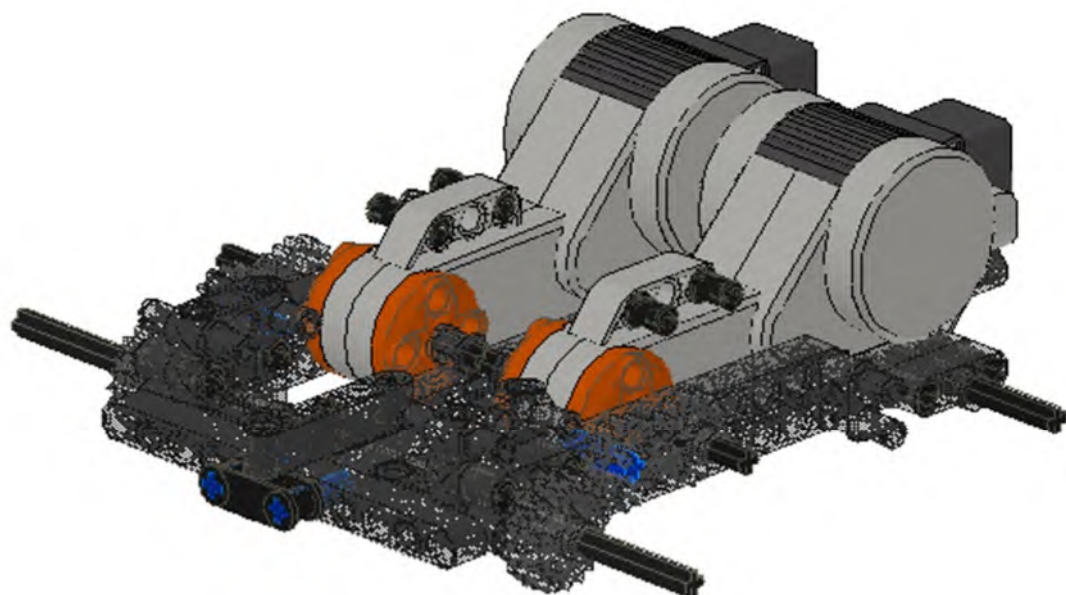
8



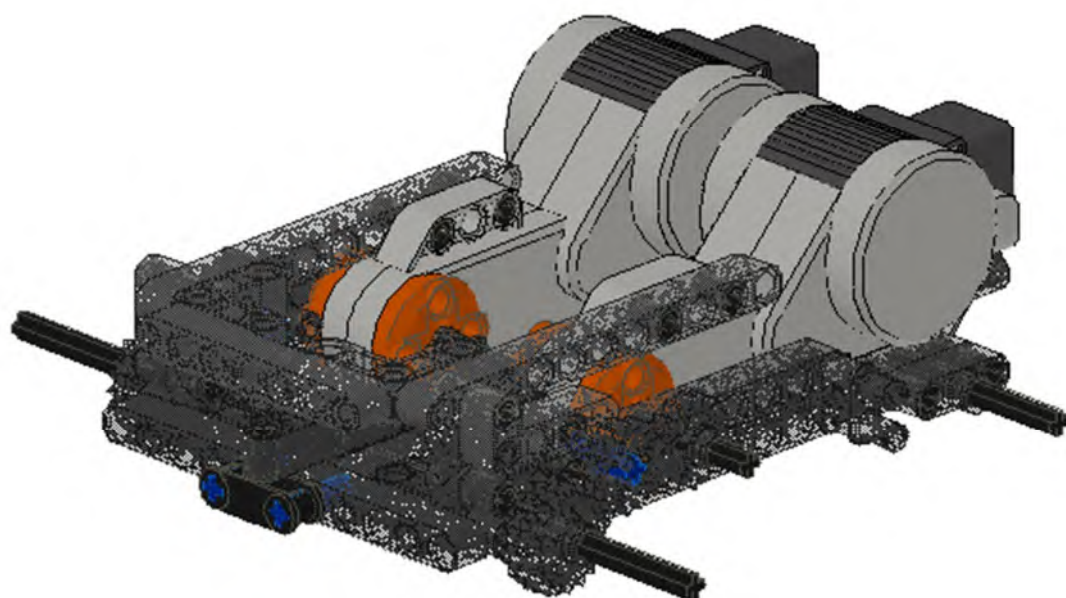
9



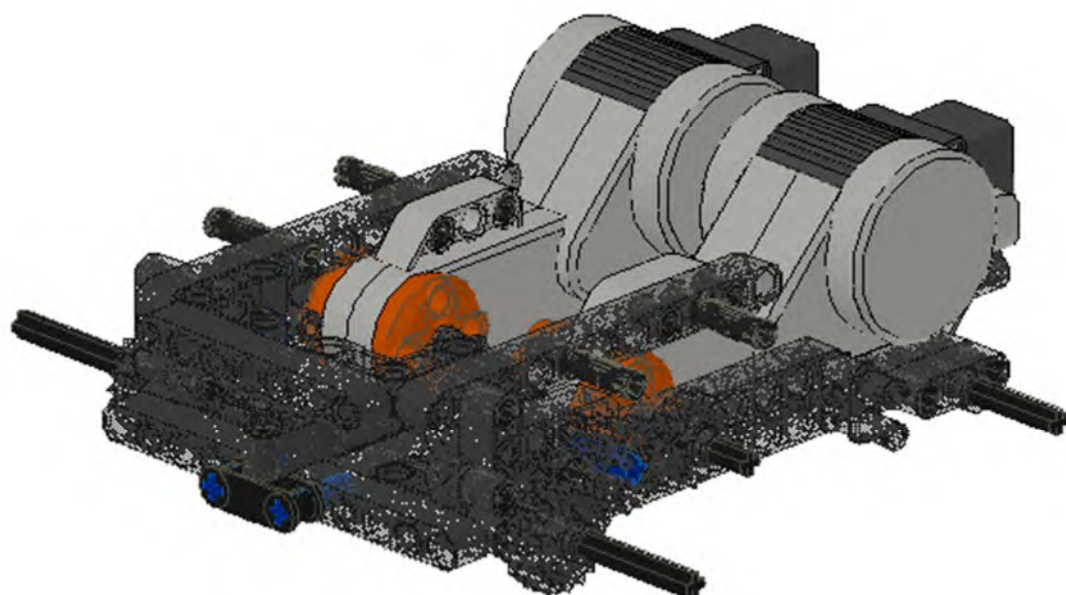
10



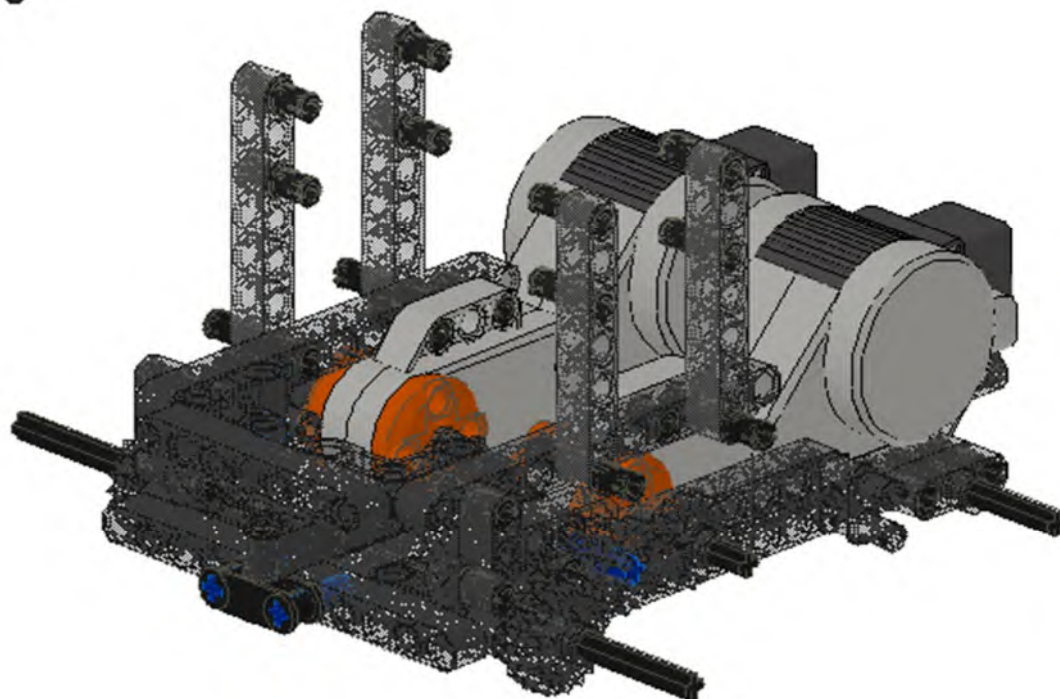
11



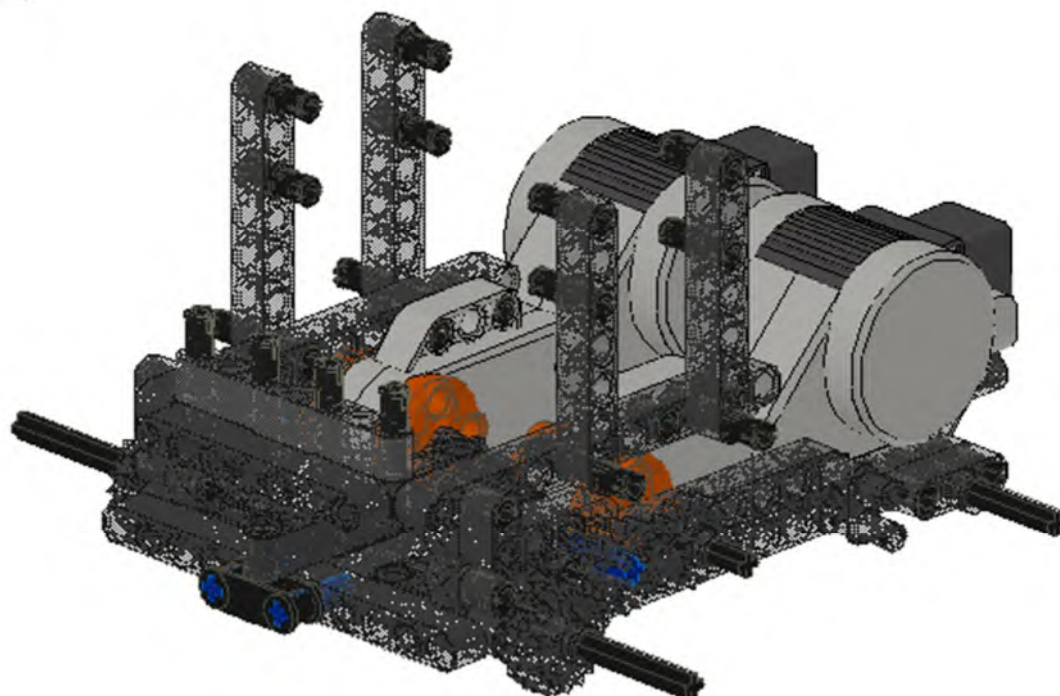
12



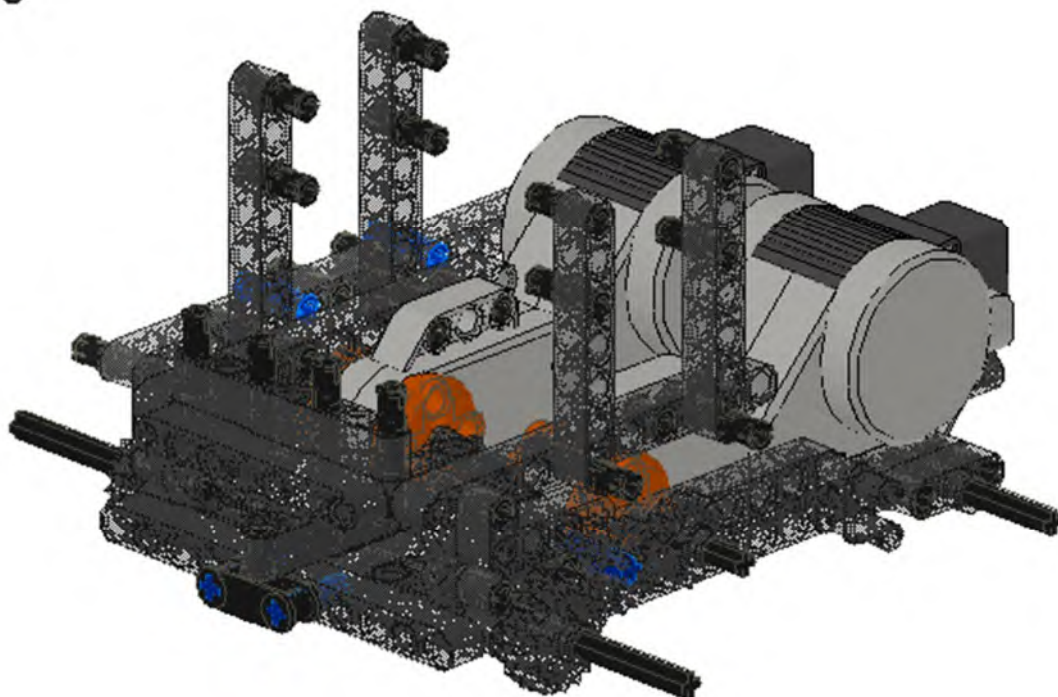
13



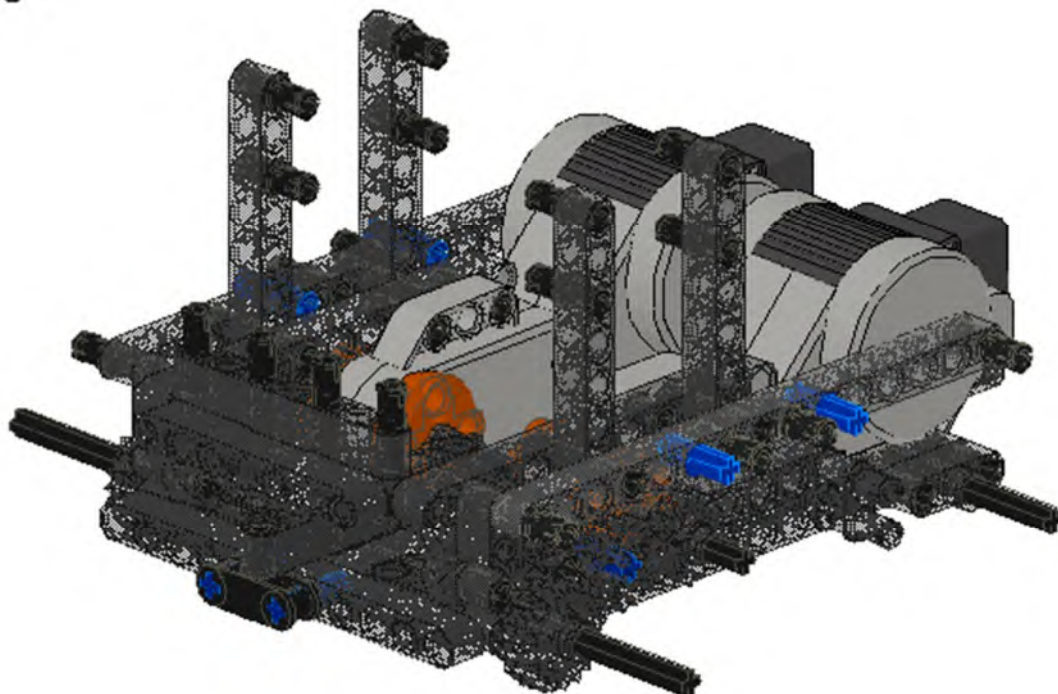
14



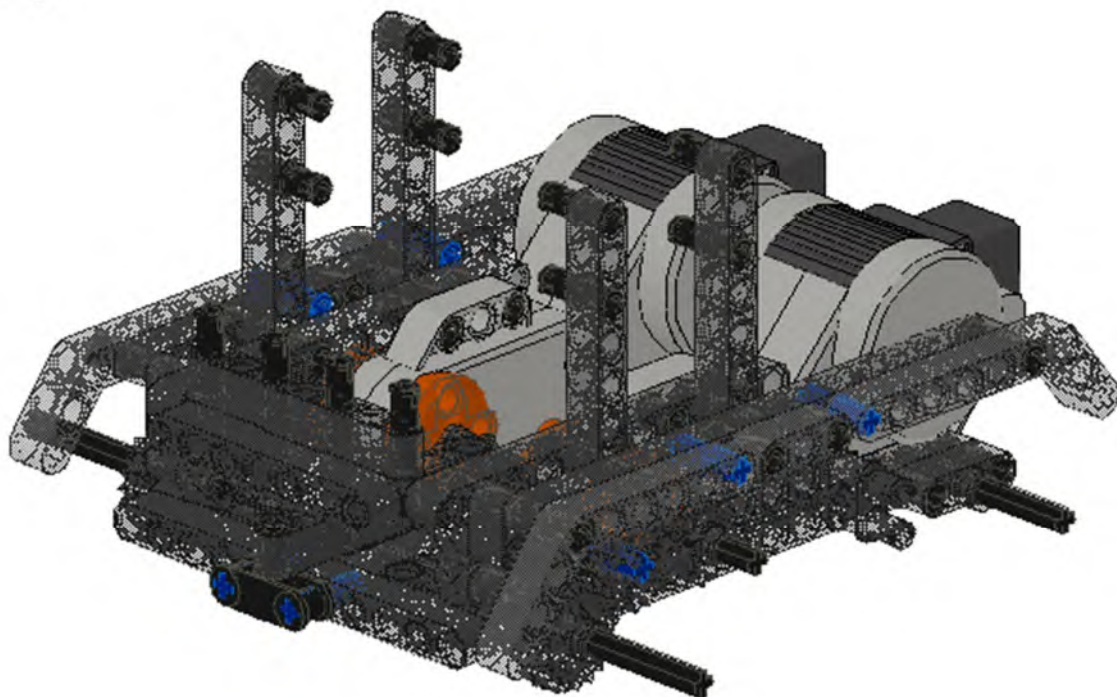
15



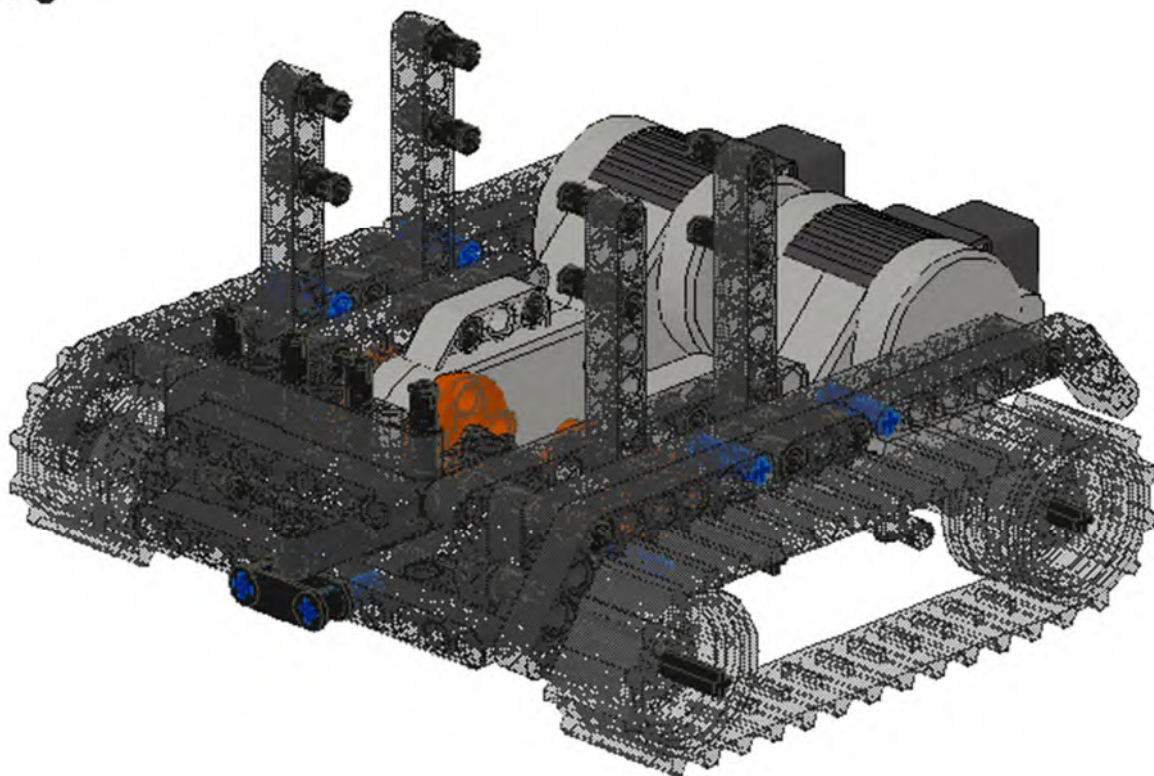
16



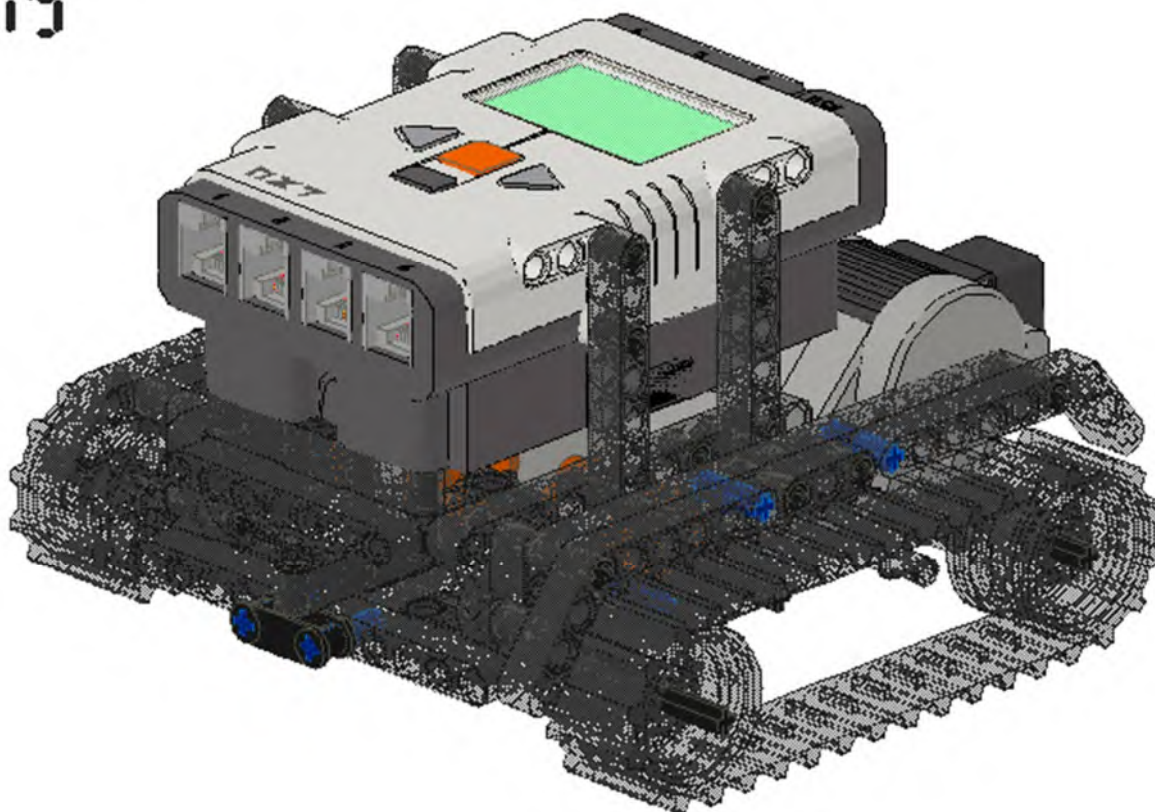
17



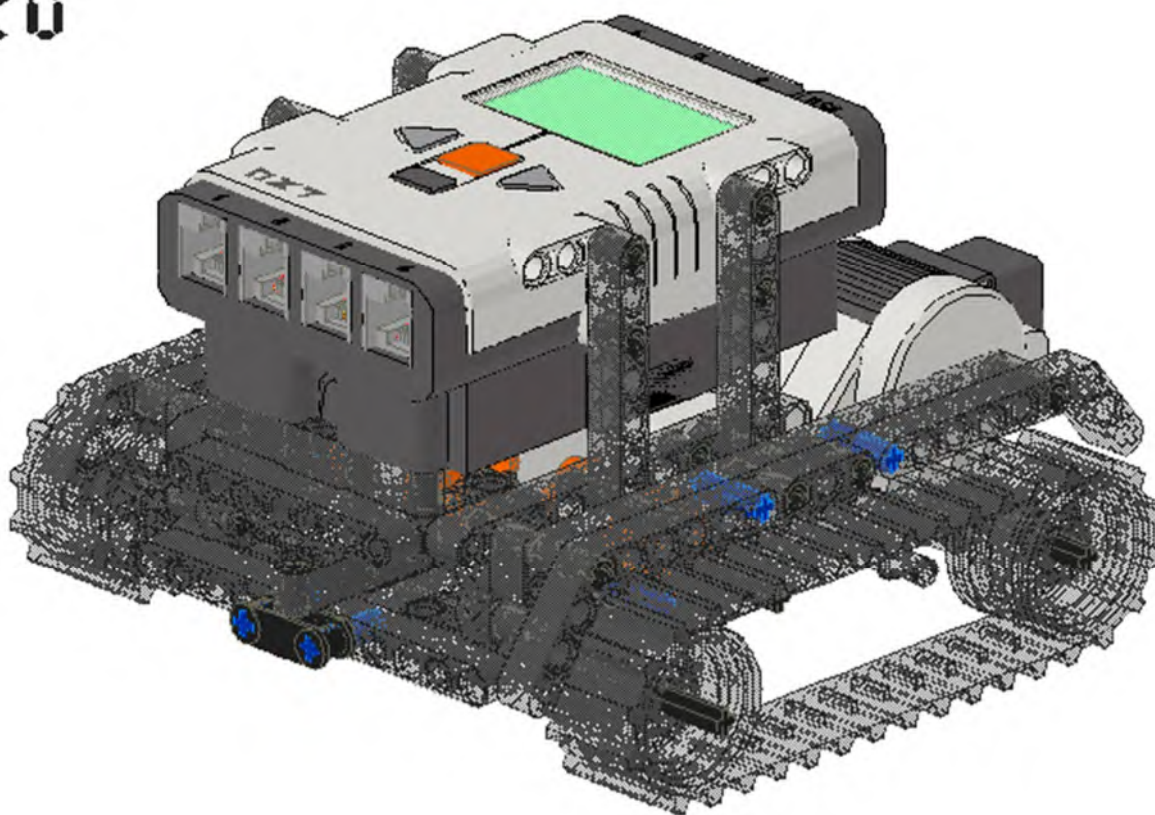
18



19



20



Μπορείτε να κατεβάσετε το αρχείο με το σχέδιο σε μορφή [LDD](#) και σε [Mlcad](#)

Βιβλιογραφία

1. Άρθρα

- Αλιμήσης Δ. Το προγραμματιστικό περιβάλλον Lego Mindstorms ως εργαλείο υποστήριξης εκπαιδευτικών δραστηριοτήτων ρομποτικής. *4ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής*, Πάτρα, Μάρτιος 2008.
- Αναγνωστάκης Σ., Μαργετουσάκη Α., Μιχαηλίδης Π. Γ. Δυνατότητα Εργαστηρίου Εκπαιδευτικής Ρομποτικής στα Σχολεία. *4ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής*, Πάτρα, Μάρτιος 2008.
- Ατματζίδου Σ., Μαρκέλης Η., Δημητριάδης Σ. Χρήση των LEGO Mindstorms στο Δημοτικό και Λύκειο: Το παιχνίδι ως έναυσμα μάθησης. *4ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής*, Πάτρα, Μάρτιος 2008.
- Γρηγοριάδου, Μ., Γόγολου, Α., & Γούλη, Ε. Μαθησιακές Δυσκολίες στις Επαναληπτικές Δομές. *4ο Συνεδρίο ΕΤΠΕ*, Αθήνα, Σεπτέμβριος 2004.
- Δημητρίου Α. και Χατζηκρανιώτης Ε. Η εκπαιδευτική ρομποτική ως εργαλείο ανάπτυξης δεξιοτήτων για τη λύση προβλήματος: Εφαρμογή με το περιβάλλον LegoDacta. *2ο Συνέδριο Εκπαιδευτικών για τις ΤΠΕ στην Εκπαίδευση*, Σύρος, Μάιος 2003
- Καγκάνη, Κ., Δαγδιλέλης, Β., Σατρατζέμη, Μ., Ευαγγελίδης, Γ. Μια μελέτη περίπτωσης της διδασκαλίας του προγραμματισμού στη δευτεροβάθμια Εκπαίδευση με τα LEGO Mindstorms. *3ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής*, Κόρινθος, Οκτώβριος 2005.
- Κάππας, Κ., Μαζέρας, Α., Παπαλεωνίδα, Γ., Λαμπροπούλου, Ε., & Ζήβελδης, Α. Εκπαιδευτική και Παιδαγωγική Προσέγγιση της Αλγοριθμικής και του Προγραμματισμού με τις ΤΠΕ και Λογισμικά. *3ο Συνέδριο ΕΤΠΕ*, Ρόδος, Σεπτέμβριος 2002.
- Καρατράντου Α., Τάχος Ν., Αλιμήσης Δ. Εισαγωγή σε βασικές αρχές και δομές Προγραμματισμού με τις ρομποτικές κατασκευές LEGO Mindstorms. *3ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής*, Κόρινθος, Οκτώβριος 2005.
- Καρατράντου, Α., Παναγιωτακόπουλος, Χ. & Πιερρή, Ε. Οι ρομποτικές κατασκευές Lego Mindstorms στην κατανόηση εννοιών Φυσικής στο Δημοτικό σχολείο: Μια μελέτη περίπτωσης. *5ο Συνέδριο ΕΤΠΕ*, Θεσσαλονίκη, Οκτώβριος 2006.
- Κυνηγός, Χ και Φράγκου, Σ. Πτυχές της παιδαγωγικής αξιοποίησης της Τεχνολογίας Ελέγχου στη σχολική τάξη. *2ο Συνέδριο ΕΤΠΕ*, Πάτρα, Οκτώβριος 2000.
- Κυνηγός, Χ, Γιαννούτσου, Ν., & Φράγκου, Σ. Μετατρέποντας «Μισοψημένους Μικρόκοσμους» σε ηλεκτρονικά παιχνίδια: μια πρόταση για τη διδασκαλία του προγραμματισμού. *5ο Συνέδριο της ΕΤΠΕ*, Θεσσαλονίκη, Οκτώβριος 2006.
- Κυνηγός, Χ. και Φράγκου, Σ. Παιδαγωγική Αξιοποίηση της Τεχνολογίας Ελέγχου στην τάξη. *2ο Συνέδριο ΕΤΠΕ*, Πάτρα, Οκτώβριος 2000.
- Μαργαρίτης, Κ., Δαγδιλέλης, Β., Σατρατζέμη, Μ., Συκοπετρίτης, Α. Μεταβάλλοντας το εκπαιδευτικό πανόραμα: η περίπτωση της Εκπαιδευτικής Ρομποτικής, *Συνέδριο Τεχνολογίας και Κοινωνίας των Πληροφοριών*, Αθήνα 2000.
- Ματάνας, Ν., Παπαβασιλείου, Α., & Παπαμήτσιου, Ζ. Τα προβλήματα της διδασκαλίας της Πληροφορικής στη Δευτεροβάθμια Εκπαίδευση. *2ο Συνέδριο Διδακτικής της Πληροφορικής*, Βόλος, Οκτώβριος 2003.
- Μπέλλου, Ι., & Μικρόπουλος, Α. Τ. Μία Εναλλακτική Πρόταση για την Εισαγωγή στον Προγραμματισμό στο Γυμνάσιο. *3ο Συνέδριο Διδακτικής της Πληροφορικής*, Κόρινθος, Οκτώβριος 2005.

- Νικολός Δ., Καρατράντου Α., Παναγιωτακόπουλος Χ. Αξιοποίηση του MicroWorlds EX Robotics για την κατανόηση βασικών δομών προγραμματισμού. *4ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής*, Πάτρα, Μάρτιος 2008.
- Ξυνόγαλος Σ. Εκπαιδευτική Τεχνολογία: Ένας Διδακτικός Μικρόκοσμος για την Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό, *Διδακτορική Διατριβή, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας*, 2002
- Ξυνόγαλος, Σ., Σατρατζέμη, Μ. & Δαγδιδέλης, Β. Η εισαγωγή στον προγραμματισμό: Διδακτικές Προσεγγίσεις και Εκπαιδευτικά Εργαλεία. *2ο Συνέδριο ΕΤΠΕ*, Πάτρα, Οκτώβριος 2000.
- Παπανικολάου Κυπαρισσία, Φράγκου Στασινή, Αλιμήσης Δημήτρης. Αναπτύσσοντας ένα πλαίσιο σχεδίασης και εφαρμογής δραστηριοτήτων προγραμματιζόμενων ρομποτικών κατασκευών: το έργο TERECoP. *4ο Συνέδριο Σύρου για τις ΤΠΕ στην Εκπαίδευση*, Σύρος, Μάιος 2007
- Τζιμογιάννης, Α. & Κόμης, Β. Επίλυση προβλημάτων σε προγραμματιστικό περιβάλλον: η οικοδόμηση της δομής ελέγχου από τους μαθητές του Ενιαίου Λυκείου. *4ο Συνέδριο Διδακτικής των Μαθηματικών & Πληροφορική στην Εκπαίδευση*, Ρέθυμνο 1999.
- Τζιμογιάννης, Α., & Γεωργίου, Β. Οι δυσκολίες μαθητών δευτεροβάθμιας εκπαίδευσης στην εφαρμογή της δομής ελέγχου για την ανάπτυξη αλγορίθμων. Μία μελέτη περίπτωσης. *1ο Συνέδριο ΕΤΠΕ*, Ιωάννινα, Μάιος 1999.
- Τζιμογιάννης, Α. Η διδασκαλία του προγραμματισμού στο Ενιαίο Λύκειο: Προς ένα ολοκληρωμένο πλαίσιο με στόχο την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων. *2ο Συνέδριο για τις ΤΠΕ στην Εκπαίδευση*, Σύρος, Μάιος 2003.
- Τζιμογιάννης, Α. Προς ένα Παιδαγωγικό Πλαίσιο Διδασκαλίας του Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση. *3ο Συνέδριο Διδακτικής της Πληροφορικής*, Κόρινθος, Οκτώβριος 2005.
- Τσοβόλας Σ., Κόμης Β. Προγραμματισμός ρομποτικών κατασκευών: μελέτη περίπτωσης με μαθητές δημοτικού. *4ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής*, Πάτρα, Μάρτιος 2008.
- Τσοβόλας, Σ., & Κόμης, Β. Διδασκαλία Βασικών Προγραμματιστικών Εννοιών στο Περιβάλλον Οπτικού Προγραμματισμού ROBOLAB. *3ο Συνέδριο Διδακτικής της Πληροφορικής*, Κόρινθος, Οκτώβριος 2005.
- Τσοβόλας, Σ., Αντωνίου, Α. Το ρομπότ και η χελώνα. *3ο Συνέδριο Σύρου για τις ΤΠΕ στην Εκπαίδευση*, Σύρος, Μάιος 2005.
- Χαρίσης Χ. και Μικρόπουλος Τ. Α. Ρομποτική, Οπτικός Προγραμματισμός και Βασικές Προγραμματιστικές Δομές. *4ο Πανελλήνιο Συνέδριο Διδακτική της Πληροφορικής*, Πάτρα, Μάρτιος 2008.

2. Βιβλία

- Papert, S. (1991) Situating Constructionism. In S.Papert and I.Harel (eds.). *Constructionism*, Norwood, NJ, Ablex Publishing Corporation.
- Papert, S. (1991). *Νοητικές Θύελλες-Παιδιά, ηλεκτρονικοί υπολογιστές και δυναμικές ιδέες*, Εκδόσεις Οδυσσέας, Αθήνα.
- Piaget, J. (1974), *To Understand Is To Invent*. N.Y.: Basic Books.
- Κόμης, Β. (2004). *Εισαγωγή στις εκπαιδευτικές εφαρμογές των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών*, Εκδόσεις Νέων Τεχνολογιών Αθήνα.
- Κόμης, Β. (2005). *Εισαγωγή στη Διδακτική της Πληροφορικής*, Εκδόσεις Κλειδάριθμος Αθήνα
- Κόμης, Β. Ι. (2005b). *Παιδαγωγικές Δραστηριότητες με (και για) Υπολογιστές στην Προσχολική και την Πρώτη Σχολική Ηλικία*. Πανεπιστημιακές Παραδόσεις, 2^η Έκδοση, Πάτρα.

3. Ηλεκτρονικές πηγές

https://el.wikipedia.org/wiki/Lego_Mindstorms
<http://edurobotics.weebly.com>

Εγχειρίδια

<http://www.legoengineering.com>
<http://www.ceeo.tufts.edu/roboLABatceeo>
<http://www.education.rec.ri.cmu.edu>

Οδηγίες κατασκευής

<http://www.brickshelf.com/cgi-bin/search.cgi?q=mindstorms&stype=fi>
<http://robotics.benedettelli.com>
<https://robotics.benedettelli.com/category/lego-nxt-projects>
<http://www.legoeducation.us/support>
<http://www.philohome.com/mindstorms.htm>
<http://www.isogawastudio.co.jp/legostudio/index.html>
<http://www.ricquin.net/lego/instructions>
<http://www.ricquin.net/lego>
<http://www.nxtprograms.com>
<http://www.tau.ac.il/~stoledo/lego>
<http://thenxtstep.blogspot.com>
<http://www.legoengineering.com>
<http://www.legoengineering.com/platform/nxt>
<http://www.education.rec.ri.cmu.edu>

Μαθήματα εκμάθησης

<http://www.legoengineering.com>

Κατάλογοι

<http://www.lugnet.com/robotics>

Χρήσιμο Υλικό

<http://www.ceeo.tufts.edu>
<http://www.roboticslearning.com>
<http://www.teamhassenplug.org>
<http://www.legoengineering.com>
<http://www.education.rec.ri.cmu.edu>

Δραστηριότητες

<http://www.legoengineering.com>
http://legoengineering.com/index.php?option=com_community&c=list&task=a&Itemid=154
http://www.legoengineering.com/index.php?option=com_community&Itemid=59&c=list&task=keywords&keywords=79
<http://www.legoeducation.us/activities/section.aspx?CategoryID=159>
<http://usfirst.org/roboticsprograms/fll/content.aspx?id=13056>
<http://www.education.rec.ri.cmu.edu>

Forums

<http://joshesnxtblog.blogspot.com>

<http://mezzanxtprojects.blogspot.com>

<http://www.nxtclub.com>

<http://mindstormsnxt.blogspot.com/>

<http://www.nxtasy.org>

<http://nxtmongersworld.blogspot.com>

http://plinetptol.sch.gr/tiki-robotics/tiki-view_articles.php

<http://robotchallenge.blogspot.com>

<http://thenxtstep.blogspot.com>

Για προχωρημένους

<http://www.multimatica.it/mindstorms/barcodereader.htm>

<http://www.teamhassenplug.org>

<http://www.hitechnic.com>

<http://www.vernier.com/nxt>