



ΕΥΦΥΗ ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ & ΕΦΑΡΜΟΓΕΣ

Ρομποτική Εφαρμογή ΒΟΕ-ΒΟΤ (Πτυχιακή Εργασία)

Επιβλέπων: Γιάννης Λιαπέρδος

Αθανάσιος Κατεργάρης & Κωνσταντίνος Παπαδάκης

Ιούνιος 2014

Πίνακας Περιεχομένων

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	2
1. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ	7
1.1 ΕΙΣΑΓΩΓΗ	9
1.2 ΙΣΤΟΡΙΑ ΤΩΝ ΡΟΜΠΟΤ	11
1.2.1 Τα αρχαία χρόνια.....	11
1.2.2 Από το 200 έως το 1900.....	14
1.2.3 Από το 1901 έως το 1950.....	19
1.2.4 Από το 1951 έως το 1960.....	22
1.2.5 Από το 1961 έως το 1971.....	24
1.2.6 Από το 1972 έως το 1980.....	26
1.2.7 Από το 1981 έως το 1990.....	29
1.2.8 Από το 1991 έως το 2000.....	30
1.2.9 Από το 2001 έως το 2009.....	32
1.2.10 Από το 2010 μέχρι σήμερα	34
1.3 ΕΤΥΜΟΛΟΓΙΑ	36
1.4 ΟΡΙΣΜΟΣ	39
1.5 ΚΑΤΑΛΟΓΟΣ ΤΩΝ ΡΟΜΠΟΤ.....	40
1.6 ΑΥΤΟΝΟΜΑ ΡΟΜΠΟΤ	45
1.6.1 Αυτο-συντήρηση.....	46
1.6.2 Αίσθηση του περιβάλλοντος	47
1.6.3 Επίδοση της εργασίας.....	48
1.6.4 Εσωτερική θέση αίσθησης και πλοήγησης.....	48
1.6.5 Εξωτερική αυτόνομη ανίχνευση θέσεως και πλοήγηση.....	50
1.6.6 Ανοιχτά προβλήματα στην αυτόνομη ρομποτική	52
1.6.7 Ενεργειακή αυτονομία και αναζήτηση τροφής	52
1.7 ΤΟΜΕΙΣ ΠΟΥ ΑΣΧΟΛΟΥΝΤΑΙ	53
1.7.1 Εργοστασιακά Ρομπότ	53
1.7.2 Βρώμικα, επικίνδυνα, αμβλύ η απρόσιτα καθήκοντα.....	55
1.7.3 Σχολεία – Εκπαιδευτικά Ιδρύματα.....	57
1.7.4 Υγειονομική Περίθαλψη	58
1.7.5 Ερευνητικά Ρομπότ.....	61
1.8 ΒΙΟΜΗΧΑΝΙΚΑ ΡΟΜΠΟΤ.....	64
1.8.1 Τεχνική περιγραφή	65
1.8.2 Ρομποτικός Προγραμματισμός και Διεπαφές.....	68

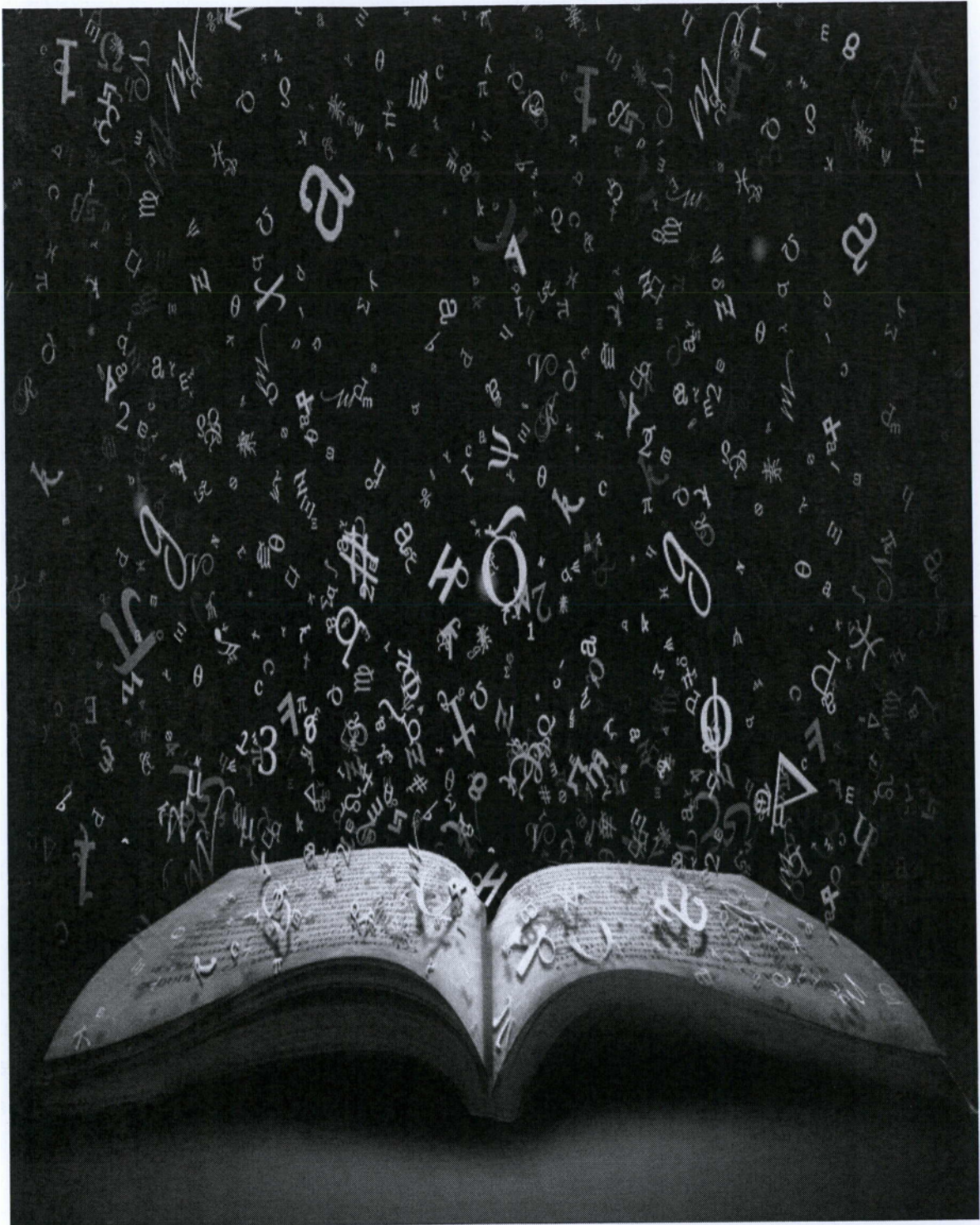
1.8.3 Έλεγχος της κίνησης.....	72
1.8.4 Ανωμαλίες.....	75
1.8.5 Πρόσφατες και μελλοντικές εξελίξεις.....	76
1.8.6 Η διάρθρωση της αγοράς.....	77
1.9 ΝΑΝΟΜΠΟΤΣ - ΝΑΝΟΡΟΜΠΟΤΣ - ΝΑΝΟΡΟΜΠΟΤΙΚΗ.....	80
1.9.1 Νανορομποτική Θεωρία.....	81
1.9.2 Προσεγγίσεις της Νανορομποτικής.....	82
1.9.3 Αγώνας των νανορομπότ.....	84
1.9.4 Πιθανές εφαρμογές στην ναυοϊατρική.....	85
1.10 ΑΝΔΡΟΕΙΔΕΣ (ANDROID).....	88
1.10.1 Τα ανδροειδή στην Ιαπωνία.....	90
1.10.2 Τα ανδροειδή στην Κορέα.....	93
1.10.3 Τα ανδροειδή στις Ηνωμένες Πολιτείες.....	94
1.10.4 Τα ανδροειδή στο Ηνωμένο Βασίλειο.....	95
1.11 CYBORG.....	96
1.11.1 Προέλευση της έννοιας.....	99
1.11.2 Ιστούς Cyborg στη μηχανική.....	100
1.11.3 Ατομικά – Προσωπικά Cyborgs.....	101
1.11.5 Πολλαπλασιασμός Cyborg στην κοινωνία.....	104
1.11.5.1 Στην Ιατρική.....	104
1.11.5.2 Στο Στρατό.....	107
1.11.5.3 Δημιουργία cyborgs για κρίσιμες μελέτες κωφών.....	109
1.11.5.4 Ίδρυμα Cyborg.....	110
1.12 ΔΙΑΣΗΜΕΣ ΚΑΙ ΜΗ ΡΟΜΠΟΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ.....	111
1.12.1 ASIMO.....	111
1.12.2 AIBO.....	115
1.12.3 ATLAS.....	117
1.13 ΤΑ ΡΟΜΠΟΤ ΣΤΗΝ ΚΟΙΝΩΝΙΑ.....	119
1.13.1 Περιφερειακές (χωρικές) προοπτικές.....	119
1.13.2 Αυτονομία και ηθικά ζητήματα.....	120
1.13.3 Στρατιωτικά ρομπότ.....	121
1.13.4 Σχέση με την ανεργία.....	122
1.14 ΤΟ ΜΕΛΛΟΝ ΤΩΝ ΡΟΜΠΟΤ.....	123
1.14.1 Τεχνολογικές τάσεις.....	123
1.14.2 Τεχνολογική ανάπτυξη.....	124
1.14.3 Εφαρμογές.....	126
2. ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ.....	131
2.1 ΒΟΕ SHIELD-BOT.....	133

2.1.1 Λίγα λόγια για το BOE Shield-Bot.....	133
2.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ BOARD OF EDUCATION SHIELD	134
2.3 ΚΑΤΑΣΚΕΥΗ ΚΑΙ ΔΟΚΙΜΗ ΤΩΝ ΕΝΔΕΙΚΤΙΚΩΝ LED	138
2.3.1 Παρουσίαση της Αντίστασης	138
2.3.2 Παρουσίαση του LED	139
2.3.3 Παρουσίαση της Prototyping Area	140
2.3.4 Δοκιμή LED κυκλώματος.....	142
2.3.5 Πως ένα πρόγραμμα ανάβει και σβήνει τα LED	143
2.3.6 Παράδειγμα Προγράμματος: HighLowLed	145
2.3.7 Παρουσίαση του διαγράμματος χρονισμού.....	145
2.4 ΠΑΡΑΚΟΛΟΥΘΗΣΗ LED SERVO ΣΗΜΑΤΩΝ.....	147
2.4.1 Παράδειγμα Προγράμματος: ServoSlowMoCcw	147
2.5 ΣΥΝΔΕΣΗ ΣΕΡΒΟΚΙΝΗΤΗΡΩΝ ΚΑΙ ΜΠΑΤΑΡΙΩΝ	148
2.5.1 Σύνδεση των Servos με το BOE Shield.....	148
2.5.2 Σύνδεση της θήκης των μπαταριών με το BOE Shield	150
2.5.3 Εγκατάσταση της θήκης μπαταριών.....	151
2.5.4 Ζυγοστάθμιση των σερβοκινητήρων.....	152
2.5.5 Δοκιμή των σερβοκινητήρων.....	154
2.5.5.1 Έλεγχος της ταχύτητας και της κατεύθυνσης των Servo.....	156
2.5.5.2 Πώς να ελέγχεται ο χρόνος κίνησης των Servo	158
2.6 ΣΥΝΑΡΜΟΛΟΓΗΣΗ ΤΟΥ BOE-SHIELD-BOT.....	160
2.7 ΛΕΙΤΟΥΡΓΙΑ ΚΑΙ ΕΠΑΝΑΦΟΡΑ ΔΕΙΚΤΗ	169
2.7.1 Κατασκευή του Piezospeaker κυκλώματος.....	170
2.7.2 Προγραμματισμός του Piezospeaker	171
2.7.3 Παράδειγμα προγράμματος StartResetIndicator	172
2.8 ΕΠΑΝΕΛΕΓΧΟΣ ΤΩΝ SERVO.....	173
2.9 ΡΥΘΜΙΣΗ ΕΛΕΓΧΟΥ ΤΑΧΥΤΗΤΑΣ.....	175
2.9.1 Έλεγχος ταχύτητας των Servo	176
2.9.2 Πως λειτουργεί το πρόγραμμα TestServoSpeed	178
2.9.3 Καταγραφή των δικών μας δεδομένων.....	180
2.10 ΠΛΗΘΗΣΗ ΤΟΥ BOE SHIELD-BOT	182
2.10.1 Παράδειγμα προγράμματος F.ThreeSeconds	183
2.10.2 Πως λειτουργεί το πρόγραμμα F.ThreeSeconds	184
2.10.3 Διάφορες κινήσεις του BOE Shield-Bot.....	186
2.10.3.1 Παράδειγμα προγράμματος ForwardLeftRightBackward	186
2.10.3.2 Ρύθμιση της ταχύτητας για βέλτιστη ευθεία	187
2.10.3.3 Συντονισμός των στροφών.....	188
2.10.3.4 Επιτόπια στροφή 90°	189
2.10.4 Υπολογισμός Αποστάσεων.....	191

2.10.5 Σταδιακή λειτουργία ελιγμών.....	194
2.10.5.1 Παράδειγμα προγράμματος - StartAndStopWithRamping	195
2.10.5.2 Προσθήκη Ramping σε άλλους ελιγμούς	196
2.10.6 Απλούστευση πλοήγησης με Συναρτήσεις	197
2.10.6.1 Παράδειγμα προγράμματος - SimpleFunctionCall	198
2.10.6.2 Κλήση συναρτήσεων με παραμέτρους	199
2.10.6.3 Παράδειγμα προγράμματος - FunctionCallWithParameter	200
2.10.6.4 Επέκταση συναρτήσεων με δυο παραμέτρους.....	200
2.10.6.5 Εισαγωγή ελιγμών μέσα σε συναρτήσεις	202
2.10.6.6 Κλήση συναρτήσεων σε βρόχο.....	203
2.10.7 Εξατομικευμένη συνάρτηση ελιγμών	204
2.10.7.1 Παράδειγμα προγράμματος - TestManeuverFunction	205
2.10.8 Ακολουθίες ελιγμών με Πίνακες.....	206
2.10.8.1 Χρησιμοποιώντας στοιχεία του πίνακα	208
2.10.8.2 Παράδειγμα προγράμματος - PlayOneNote	208
2.10.8.3 Παράδειγμα προγράμματος - PlayNotesWithLoop.....	209
2.10.8.4 Χρησιμοποιώντας τη συνάρτηση sizeof	210
2.10.8.5 Πλοήγηση με Πίνακες.....	211
2.10.8.6 Παράδειγμα προγράμματος - ManeuverSequence.....	212
2.10.8.7 Πίνακες χαρακτήρων και switch-case	213
2.10.8.8 Παράδειγμα προγράμματος - ControlWithCharacters.....	214
2.11 ΑΠΤΙΚΗ ΠΛΟΗΓΗΣΗ ΜΕ WHISKERS	217
2.11.1 Κατασκευή και έλεγχος των Whiskers	218
2.11.1.1 Πως λειτουργούν οι διακόπτες των Whiskers	220
2.11.1.2 Ελέγχοντας τα Whiskers	221
2.11.1.3 Παραδειγμα προγραμματος - DisplayWhiskerStates	222
2.11.1.4 Πως λειτουργεί το DisplayWhiskerStates	222
2.11.1.5 Εμφωλευμένες κλήσεις συναρτήσεων.....	223
2.11.2 Επιτόπιος έλεγχος των Whiskers	224
2.11.2.1 Προγραμματισμός του κυκλώματος ελέγχου με LED.....	225
2.11.3 Πλοήγηση με τα Whiskers.....	227
2.11.3.1 Παράδειγμα προγράμματος - RoamingWithWhiskers.....	227
2.11.3.2 Πως λειτουργεί το RoamingWithWhiskers	229
2.11.4 Τεχνητή Νοημοσύνη για αποφυγή γωνιών.....	231
2.11.4.1 Παράδειγμα προγράμματος - EscapingCorners	232
2.11.4.2 Πως λειτουργεί το EscapingCorners	234
2.12 ΦΩΤΟΕΥΑΙΣΘΗΤΗ ΠΛΟΗΓΗΣΗ ΜΕ ΦΩΤΟΤΡΑΝΖΙΣΤΟΡ	236
2.12.1 Παρουσίαση του the Φωτοτρανζίστορ.....	237
2.12.1.1 Κύματα φωτός.....	238
2.12.2 Απλοί αισθητήρες φωτός	238
2.12.2.1 Κατασκευή του φωτο-ανιχνευτή.....	239

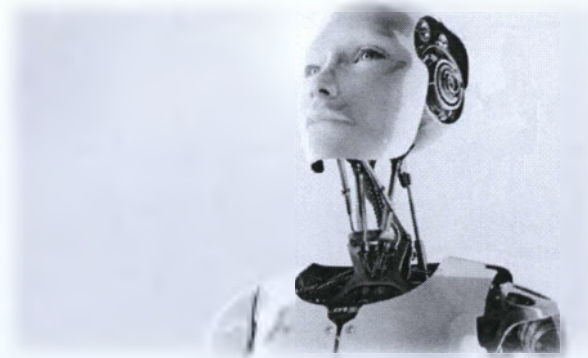
2.12.2.2	Παράδειγμα προγράμματος - PhototransistorVoltage	240
2.12.2.3	Ακίνητοποίηση κάτω από φετινό φως	242
2.12.2.4	Πως λειτουργεί η συνάρτηση Volts	243
2.12.2.5	Πως λειτουργεί το κύκλωμα του φωτοτρανζίστορ	244
2.12.2.6	Νόμος του Ohm	246
2.12.3	Μέτρηση επιπέδων φωτός σε μεγαλύτερο εύρος	247
2.12.3.1	Παρουσίαση του Πυκνωτή	247
2.12.3.2	Κατασκευή των φωτοευαίσθητων ματιών	248
2.12.3.3	Έλεγχος του κυκλώματος του φωτοτρανζίστορ	250
2.12.3.4	rcTime και φθορά της τάσης	252
2.12.4	Μετρήσεις φωτός για τη πλοήγηση	253
2.12.4.1	Παράδειγμα προγράμματος - LightSensorValues	254
2.12.5	Δοκιμή μιας φωτοκίνητης ρουτίνας	256
2.12.5.1	Έλεγχος αποφάσεων πλοήγησης με το Serial Monitor	258
2.12.5.2	Παράδειγμα προγράμματος - Light Seeking Display	259
2.12.6	Πλοήγηση του BOE Shield-Bot με φως	260
2.13	ΠΛΟΗΓΗΣΗ ΜΕ ΥΠΕΡΥΘΡΟΥΣ ΠΡΟΒΟΛΕΙΣ	263
2.13.1	Υπέρυθρα Σήματα	263
2.13.2	Κατασκευή και έλεγχος των ανιχνευτών	265
2.13.2.1	Συναρμολόγηση των IR προβολέων	265
2.13.2.2	Κύκλωμα υπέρυθρης ανίχνευσης	266
2.13.2.3	Δοκιμαστικός κώδικας για ανίχνευση αντικειμένων	267
2.13.2.4	Παράδειγμα προγράμματος - TestLeftIr	268
2.13.3	Δοκιμή στο χώρο	270
2.13.3.1	Εισαγωγή ενδεικτικών LED κυκλωμάτων	270
2.13.3.2	Παράδειγμα προγράμματος - TestBothIrAndIndicators	271
2.13.3.3	Ανίχνευση για υπέρυθρες παρεμβολές	273
2.13.4	Ρυθμίσεις στο εύρος ανίχνευσης	273
2.13.5	Ανίχνευση αντικειμένων και Αποφυγή	275
2.13.5.1	Παράδειγμα προγράμματος - RoamingWithIr	276
2.13.6	Υπέρυθρη πλοήγηση Υψηλής Απόδοσης	278
2.13.6.1	Παράδειγμα προγράμματος - FastIrRoaming	278
2.13.7	Ανίχνευση πτώσεων	280
	ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ-LINKS	284

1. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ



1.1 Εισαγωγή

Ένα ρομπότ είναι ένας μηχανικός ή εικονικός παράγοντας, συνήθως μια ηλεκτρο-μηχανική μηχανή που οδηγείται από ένα υπολογιστικό πρόγραμμα ή από ηλεκτρονικά κυκλώματα. Τα Ρομπότ μπορεί να είναι αυτόνομα ή ημι-αυτόνομα και ταξινομούνται σε



Εικ.1 Ανδοειδές σκίτσο του Benedict Campbell

ανθρωποειδή, όπως το Advanced Step in Innovative Mobility (ASIMO) της Honda και το TOSY Ping Pong Playing Robot (TOPIO) της Tosy, σε βιομηχανικά ρομπότ, σε συλλογικά προγραμματισμένα «σμήνος» ρομπότ, ακόμη και σε μικροσκοπικά ρομπότ νάνο (nano robots). Καθώς μιμείται μια ζωντανή παρουσία ή την αυτοματοποίηση των κινήσεων, ένα ρομπότ μπορεί να αποδώσει μια αίσθηση νοημοσύνης ή σκέψεων από μόνο του.

Η Ρομποτική είναι ο κλάδος της τεχνολογίας που ασχολείται με το σχεδιασμό, την κατασκευή, τη λειτουργία και την εφαρμογή των ρομπότ, (Oxford Dictionaries), καθώς και τα συστήματα πληροφορικής για τον έλεγχό τους, την ανατροφοδότηση αισθητήρων και την επεξεργασία των πληροφοριών. Οι τεχνολογίες αυτές ασχολούνται με αυτοματοποιημένες μηχανές που μπορούν να πάρουν τη θέση των ανθρώπων σε επικίνδυνα περιβάλλοντα ή σε διαδικασίες κατασκευής, ή να μοιάζουν με τους ανθρώπους στην εμφάνιση, τη συμπεριφορά, ή / και τη γνωστική λειτουργία. Πολλά από τα σημερινά ρομπότ είναι εμπνευσμένα από τη φύση συμβάλλοντας στον τομέα της βίο-εμπνευσμένης ρομποτικής (bio-inspired robotics).

Όπως οι μηχανικές τεχνικές αναπτύχθηκαν μέσα από τη βιομηχανική εποχή, οι πιο πρακτικές εφαρμογές προτάθηκαν από τον Nikola Tesla, ο οποίος το 1898 σχεδίασε ένα τηλεκατευθυνόμενο σκάφος. Η Ηλεκτρονική εξελίχθηκε σε κινητήρια δύναμη της ανάπτυξης, με την έλευση των πρώτων ηλεκτρονικών αυτόνομων ρομπότ που δημιουργήθηκαν από τον William Grey Walter στο Μπρίστολ της Αγγλίας το 1948. Το

πρώτο ψηφιακό και προγραμματιζόμενο ρομπότ εφευρέθηκε από τον George Devol το 1954 και ονομάστηκε Unimate. Το οποίο πωλήθηκε στην General Motors το 1961, όπου χρησιμοποιήθηκε για να σηκώνει κομμάτια ζεστού μετάλλου από μηχανές χύτευσης μετάλλου («die casting») στο «Inland Fisher Guide Plant» του τμήματος West Trenton του Ewing Township, στο Νιού Τζέρσεϊ. (PEARCE)

Τα ρομπότ έχουν αντικαταστήσει τους ανθρώπους (Aquino) στη βοήθεια της εκτέλεσης αυτών των επαναλαμβανόμενων και επικίνδυνων εργασιών που οι άνθρωποι προτιμούν να μην κάνουν, ή δεν είναι σε θέση να κάνουν λόγω των περιορισμών μεγέθους, ή ακόμα και αυτές, όπως στο διάστημα ή στο βυθό της θάλασσας, όπου οι άνθρωποι δεν μπορούν να επιβιώσουν στα ακραία περιβάλλοντα.

Υπάρχουν ανησυχίες σχετικά με την αυξανόμενη χρήση των ρομπότ και το ρόλο τους στην κοινωνία, καθώς τα Ρομπότ είναι υπεύθυνα για την άνοδο της ανεργίας ,αφού αντικαθιστούν τους εργαζόμενους σε ορισμένες λειτουργίες. Η χρήση των ρομπότ σε στρατιωτικές μάχες εγείρει ηθικές ανησυχίες, και η δυνατότητα της αυτονομίας των ρομπότ και των πιθανών επιπτώσεων έχει εξεταστεί στη μυθιστοριογραφία και μπορεί να είναι μια ρεαλιστική ανησυχία για το μέλλον.

1.2 Ιστορία των Ρομπότ

Η ιστορία των ρομπότ έχει τις ρίζες της ήδη από τους αρχαίους μύθους και θρύλους. Οι σύγχρονες αντιλήψεις άρχισαν να αναπτύσσονται όταν η βιομηχανική επανάσταση επέτρεψε τη χρήση μιας πιο πολύπλοκης μηχανικής και η επακόλουθη εισαγωγή της ηλεκτρικής ενέργειας έδωσε τη δυνατότητα κατασκευής μηχανημάτων με μικρούς συμπαγής κινητήρες. Μετά τη δεκαετία του 1920, αναπτύχθηκε η σύγχρονη διαμόρφωση της ανθρωποειδούς μηχανής. Σήμερα, μπορούμε να προβλέψουμε ένα ρομπότ ανθρωπίνου μεγέθους με την ικανότητα για ανθρώπινες σκέψεις και κινήσεις, πρόβλεψη που είχε διατυπωθεί χιλιετίες πριν.

Η πρώτη χρήση των σύγχρονων ρομπότ ήταν σε εργοστάσια, όπως τα βιομηχανικά ρομπότ, απλές σταθερές μηχανές ικανές για κατασκευαστικές εργασίες που επέτρεπαν την παραγωγή χωρίς την ανάγκη για ανθρώπινη βοήθεια. Ψηφιακά ελεγχόμενα βιομηχανικά ρομπότ και ρομπότ που χρησιμοποιούν τεχνητή νοημοσύνη έχουν κατασκευαστεί από το 1960.

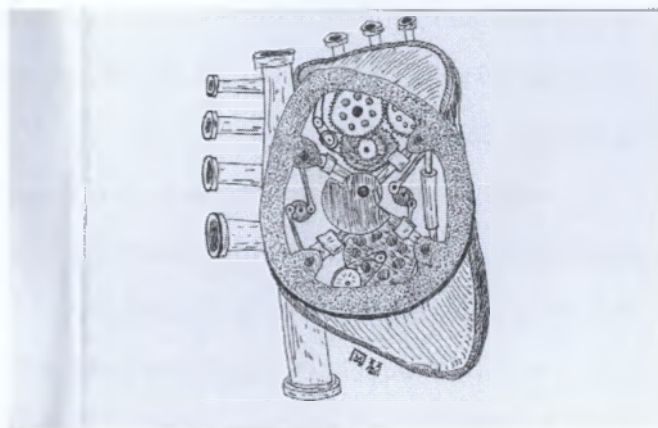
1.2.1 Τα αρχαία χρόνια

Στην αρχαία Κίνα, ένας περίεργος απολογισμός στα αυτόματα βρίσκεται στο κείμενο Lie Zi, γραμμένο τον 3^ο αιώνα π.Χ.. Μέσα σε αυτό υπάρχει μια περιγραφή μιας πολύ νεώτερης συνάντησης μεταξύ του βασιλιά Mu Zhou (976-922 π.Χ.) και ενός μηχανολόγου μηχανικού γνωστού ως Yen Shih, ένας «τεχνίτης»(artificer), ο οποίος παρουσίασε με υπερηφάνεια στο βασιλιά μια πραγματικού μεγέθους ανθρώπινη διαμορφωμένη φιγούρα των μηχανικών εργοχειρών του. Ο βασιλιάς κοίταξε την φιγούρα με κατάπληξη. Αυτό περπάτησε με ταχεία πρόοδο, κουνώντας το κεφάλι του πάνω κάτω, έτσι ώστε ο καθένας θα μπορούσε να το εκλάβει σαν ένα ζωντανό ανθρώπινο ον. Ο τεχνίτης άγγιξε το πηγούνι του, και άρχισε να τραγουδάει, σε τέλεια αρμονία. Άγγιξε το χέρι του, και άρχισε να ποζάρει, διατηρώντας τέλειο χρονισμό.

Καθώς η παράσταση πλησίαζε στο τέλος της, το ρομπότ έκλεισε το μάτι του και προχώρησε προς τις παρούσες κυρίες, πράγμα που εξόργισε το βασιλιά. Από το φόβο

του να μην εκτελεστεί επί τόπου, ο Yen Shih αμέσως κομμάτιασε το ρομπότ για να δήσει στον βασιλιά τι πραγματικά

ήταν. Και πράγματι, αποδείχθηκε ότι ήταν μόνο μια κατασκευή από δέρμα, ξύλο, κόλλα και βερνίκι, ποικιλοτρόπως χρωματισμένα λευκό, μαύρο, κόκκινο και μπλε. Εξετάζοντας το περεταιίρω, ο βασιλιάς βρήκε όλα τα εσωτερικά



Εικ.2 Σκίτσο του τεχνίτη Yen Shih

όργανα πλήρη, συκώτι, χοληδόχους κύστη, καρδιά, πνεύμονες, σπλήνα, νεφρά, στομάχι και έντερα, και πάνω από αυτά και πάλι, οι μύες, τα οστά και τα άκρα με αρθρώσεις, το δέρμα, τα δόντια και τα μαλλιά, όλα αυτά τεχνητά. Ο βασιλιάς δοκίμασε την αφαίρεση της καρδιάς και διαπίστωσε ότι το στόμα δεν μπορούσε πλέον να μιλήσει, πήρε μακριά το συκώτι και τα μάτια δεν μπορούσαν πλέον να δούναι, πήρε μακριά τα νεφρά και τα πόδια έχασαν την δύναμη τους. Ο βασιλιάς ήταν ευχαριστημένος. (Needham)

Η ιδέα του τεχνητού ανθρώπου στη δυτική μυθολογία χρονολογείται τουλάχιστον από τους αρχαίους Ελληνικούς μύθους του Κάδμου (hellasmythology), ο οποίος έσπειρε δόντια δράκου τα οποία μετατράπηκαν σε στρατιώτες, και ο μύθος του Πυγμαλίωνα, του οποίου το άγαλμα της Γαλάτειας ζωντάνεψε (Greece). Ο παραμορφωμένος θεός της μεταλλοτεχνίας Ήφαιστος (ή Vulcan) δημιούργησε μηχανικούς υπηρέτες, που κυμαίνονταν από τις έξυπνες, χρυσές υπηρέτριες που τον βοηθούσαν να στέκεται όρθιος όταν κουραζόταν, σε πιο ωφέλιμα τρίποδα τραπέζια που μπορούσαν να μετακινηθούν με τη δική τους δύναμη (ενέργεια) (greekhistory). Επίσης του αποδίδεται και η κατασκευή του Τάλου, ένα τεράστιο ανθρωπόμορφο, χάλκινο κατασκεύασμα το οποίο ο θεός δώρισε στον βασιλιά Μίνωα

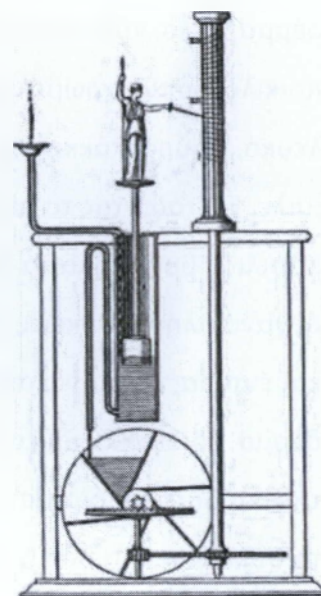


Εικ.3 Τάλος

για να προστατεύει τη Κρήτη. Το τέλος του Τάλω ήρθε όταν συνάντησε τους Αργοναύτες, που με την βοήθεια της Μήδειας, μαγεύοντας τον με τα λόγια της και την υπόσχεση της αθανασίας, ο Ιάσωνας κατάφερε να του αφαιρέσει το καρφί στη φτέρνα

του που έκλεινε τη μια και μοναδική φλέβα (Ταλως). Ακόμα, και ο Ιάσοντας για να πάρει το χρυσόμαλλο δέρας ήταν υποχρεωμένος να δαμάσει δύο ταύρους που ξεφυσούσαν φωτιά με χάλκινες σπλές, αλλά και να νικήσει στρατιώτες που ο Κάδμος έσπειρε με τα δόντια του δράκου.

Πρώρα ρολόγια του νερού είναι μερικές φορές ομαδοποιημένα με τις αρχές της ρομποτικής. Άρχισαν στην Κίνα τον 6ο αιώνα π.Χ. και στον ελληνο-ρωμαϊκό κόσμο τον 4ο π.Χ. αιώνα, όπου η Κλεψύδρα είναι γνωστό ότι είχε χρησιμοποιηθεί ως χρονόμετρο για την επιβολή προθεσμίας στους πελάτες των αθηναϊκών οίκων ανοχής. (waterclock) Ωστόσο, ο Κτησίβιος της Αλεξάνδρειας, ένας φυσικός και εφευρέτης από την πτολεμαϊκή Αίγυπτο, έκανε πολλές καινοτομίες στον τομέα των ποιο μηχανοποιημένων κλεψύδρων και των αυτομάτων, μεταξύ των οποίων μία που υποτίθεται ότι μπορούσε να μιλήσει. (ktisivios)



Εικ.4 Κλεψύδρα ή ελληνικό ρολόι νερού

Έννοιες που μοιάζουν με ρομπότ μπορούν να βρεθούν ήδη από τον 4ο αιώνα π.Χ., ο Έλληνας μαθηματικός Αρχύτας του Τάραντα φημίζεται για την δημιουργία ενός μηχανικού πουλιού γύρω στο 400 π.Χ., που ονόμασε "Το περιστέρι", το οποίο πιθανών προωθούνταν με ατμό και ήταν ικανό να πετάει. (architas)

Λαμβάνοντας μια παλαιότερη αναφορά στην Ομήρου "Ιλιάδα, ο Αριστοτέλης θεώρησε μέσα στο έργο του Πολιτική (περ. 322 π.Χ., βιβλίο 1, μέρος 4) ότι τα αυτόματα θα μπορούσαν κάποια μέρα να επιφέρουν την ανθρώπινη ισότητα, καθιστώντας δυνατή την κατάργηση της δουλείας :

- Υπάρχει μόνο μια κατάσταση κατά την οποία μπορούμε να φανταστούμε διαχειριστές να μην χρειάζονται υφισταμένους, και οι άρχοντες να μην χρειάζονται σκλάβους. Αυτή η κατάσταση θα μπορούσε να είναι ότι κάθε όργανο θα μπορούσε να κάνει την δική του δουλειά, μέσω της εντολής ή με προσδοκία ευφυΐας, όπως τα αγάλματα του Δαίδαλου ή οι τρίποδες από τον Ήφαιστο, από την οποία ο Όμηρος αναφέρει ότι «Από τη δική τους κίνηση (με την δική τους ενέργεια) μπήκαν στο συμβούλιο των Θεοί του Ολύμπου », σαν μια σαΐτα αργαλειού θα πρέπει να υφαίνει από μόνη της, και μια πένα θα πρέπει να παίζει άρπα μόνη της. (politikh)

Οι Εβραϊκές λαϊκές παραδόσεις αναφέρουν τον εβραϊκό μύθο του Γκόλεμ (Golem) , ένα πλάσμα από πηλό κινούμενο με Kabbalistic (Καμπάλα) μαγεία. Ομοίως, στην Prose Edda (Younger Edda) , η Σκανδιναβική μυθολογία αναφέρει ένα πήλινο γίγαντα, τον Mōkkurkálfi ή Mistcalf, κατασκευασμένο για να βοηθήσει τον troll Hrungnir σε μια μονομαχία με τον Thor , τον Θεό της βροντής .

1.2.2 Από το 200 έως το 1900

Η Κοσμική μηχανή, ένας 10μετρος (33 ft) πύργος ρολογιού που χτίστηκε από τον Su Song στην Καϊφένγκ (Κίνα) το 1088, αποτελούταν από μηχανικές κούκλες που σήμαιναν (chimed) τις ώρες, χτυπώντας γκονγκ(gong) ή καμπάνες μεταξύ άλλων συσκευών. (song)



Εικ.5 Η κοσμική μηχανή του Su Song

Ο Al-Jazari(1136-1206), ένας Άραβας μουσουλμάνος εφευρέτης κατά τη διάρκεια της δυναστείας Artuqid , σχεδίασε και κατασκεύασε μια σειρά από αυτόματες μηχανές, συμπεριλαμβανομένων συσκευών κουζίνας, μουσικών αιτημάτων τροφοδοτούμενα από νερό, και το πρώτο προγραμματιζόμενο ανθρωποειδές ρομπότ το 1206. Το Ρομπότ Al-Jazari ήταν ένα σκάφος με τέσσερα μουσικά αυτόματα που επέπλεε σε μια λίμνη για να διασκεδάσει τους επισκέπτες στα βασιλικά πάρτη.



Εικ.6 Το μουσικό αυτόματο του Al-Jazari

Οι μηχανισμοί του είχαν μια προγραμματιζόμενη μηχανή τυμπάνων με έκκεντρα τα οποία προσέκρουαν σε μικρούς μοχλούς για να λειτουργήσουν τα κρουστά. Ο ντράμερ είχε φτιαχτεί με σκοπό να παίζει διαφορετικούς ρυθμούς και διαφορετικά μοτίβα μετακινώντας τα έκκεντρα σε διαφορετικές τοποθεσίες. (al-jazari)

Στον ισλαμικό θρύλο του Rocail , ο νεότερος αδελφός του Seth δημιούργησε μια πόλη από αυτόνομα αγάλματα που ζούσαν τις ζωές των ανθρώπων. (rocail)

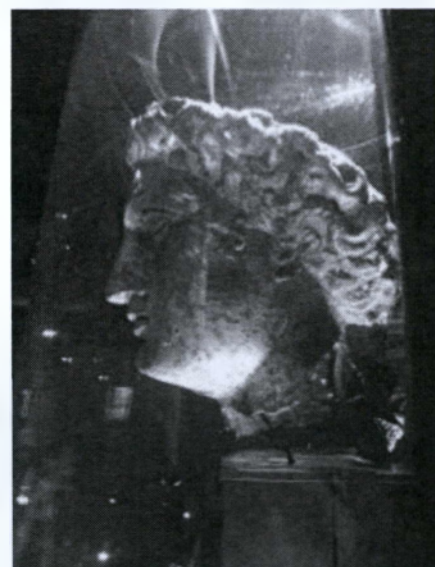
Ο πάπας Sylvester II λέγεται ότι είναι ένας από τους πρώτους χριστιανικούς κληρικούς που έμαθαν από τους Μουσουλμάνους , και είναι υπεύθυνος για την

εισαγωγή και τη χρήση των ρολογιών προς τα δυτικά . Επίσης υποστηρίζεται ότι είχε κατασκευάσει ένα διφορούμενο ορειχάλκινο κεφάλι. (sylvester)

Σύμφωνα με θρύλους στις αρχές της δεκαετίας του 1200 ο εκκλησιαστικός γιατρός Albertus Magnus δημιούργησε έναν άνδρα από ορείχαλκο, ο οποίος μπορούσε να εργαστεί μόνο κάτω από ορισμένους αστερισμούς , και η διαδικασία αυτή του πήρε σχεδόν 30 χρόνια για να ολοκληρωθεί. Αυτός ο ορειχάλκινος άνθρωπος απαντούσε σε περίπλοκα ερωτήματα και εργαζόταν ως οικιακός βοηθός . Η ικανότητα του για ευφυή ομιλία διατάραξε βαθιά τον συνάδελφο του Thomas Aquinas , μαθητή του Magnus, ο οποίος σε μια έκρηξη οργής κατέστρεψε τον ορειχάλκινο άνδρα με ένα σφυρί .

(A.Magnus)

Ένα άλλος διάσημος μεσαιωνικός μύθος αυτόματων είναι του Roger Bacon (Ρότζερ Μπέικον). Στην ιστορία της φυσικής ο Friend λέει ότι ο Ρότζερ Μπέικον κατασκεύαζε αγάλματα που μπορούσαν να κινούνται και να συντάσσουν κατανοητούς ήχους από μια ορειχάλκινη κεφαλή, όχι με μαγεία, αλλά μέσω των φυσικών επιστημών. Αυτή η ιστορία χρησιμοποιήθηκε ως βάση για έναν από τους πιο γνωστούς μύθους της Αγγλίας. Ο Friar Bacon και ο Father Bungy υποσχέθηκαν να περικλείσουν την Αγγλία με ένα τοίχο ώστε να είναι απρόσιτη από τους εισβολείς. Για να τα καταφέρουν, ο διάβολος τους είπε ότι πρέπει να κατασκευάσουν ένα ορειχάλκινο κεφάλι με όλες τις εσωτερικές δομές και τα όργανα ενός ανθρώπινου κεφαλιού, έτσι ώστε να μπορέσει να μάθει να μιλάει και να μπορεί να τους απαντήσει



Εικ.7 Ορειχάλκινη Κεφαλή

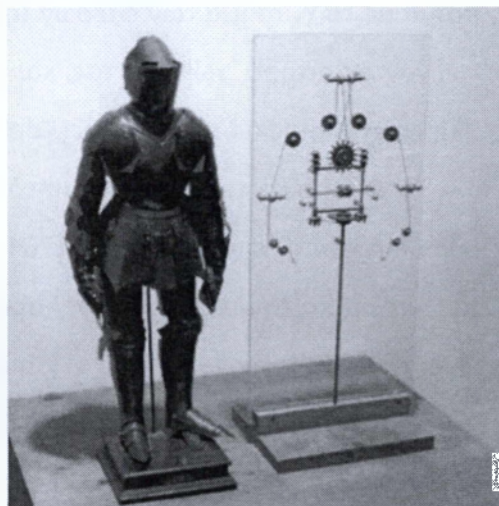
σε οποιαδήποτε ερώτηση είχαν. Οπότε, κατασκεύασαν την κεφαλή, η οποία μετά από λίγο καιρό άρχισε να μιλάει και να βοηθάει σε οποιοδήποτε πρόβλημα προκαλούταν. Ωστόσο, μια νύχτα μετά από 7 χρόνια κατασκευής του τοίχους, το κεφάλι είπε τα εξής : «Ο χρόνος είναι, Ο χρόνος ήταν» και τέλος «Ο χρόνος έχει περάσει» πριν καταστραφεί σε χίλια κομμάτια από έναν κεραυνό. (R.bacon)

Στις αρχές του 13ου αιώνα ο καλλιτέχνης και μηχανικός Villard de Honnecourt σχεδίασε επίσης σχέδια για αρκετά αυτόματα. (V.de.H) Στο τέλος του δέκατου τρίτου

αιώνα , ο Robert II, Count of Artois έχτισε ένα όμορφο κήπο στο κάστρο του στο Hesdin , και ενσωμάτωσε μια σειρά από ανθρωποειδή και ζωικά ρομπότ. (R.2)

Τα αυτόματα βρήκαν επίσης τον δρόμο τους μέσα στο φανταστικό κόσμο της μεσαιωνικής λογοτεχνίας. Για παράδειγμα, το Middle Dutch παραμύθι, Roman van Walewein (The Romance of Walewein , αρχές 13ου αιώνα) που περιγράφει μηχανικά πουλιά και αγγέλους που παράγουν ήχο με τη βοήθεια ενός συστήματος σωλήνων. (R.vanW.)

Ένα από τα πρώτα σχέδια που καταγράφονται για ένα ανθρωποειδές ρομπότ έγινε από τον Leonardo da Vinci (1452-1519) γύρω στο 1495. Σημειωματάρια του Ντα Βίντσι, που ανακαλύφθηκαν τη δεκαετία του 1950, περιλαμβάνουν λεπτομερή σχέδια ενός μηχανικού ιπότη με πανοπλία που ήταν σε θέση να ανασηκωθεί, κουνήσει τα χέρια του και να μετακινήσει το κεφάλι και το σαγόνι του. Το σχέδιο πιθανόν να βασίζεται στην ανατομική έρευνα του, που καταγράφεται στο «Άνθρωπος του Βιτρούβιου», αλλά δεν είναι γνωστό αν προσπάθησε να κατασκευάσει το ρομπότ. (L.daV.)



Εικ.8 Μηχανικός ιπότης, Leonardo da Vinci

Το 1533, ο Johannes Müller von Königsberg δημιούργησε από σίδηρο έναν αετό και μια μύγα αυτόματα, τα οποία μπορούσαν να πετάξουν. (J.M.) Ο John Dee είναι επίσης γνωστός για τη δημιουργία ενός ξύλινου σκαθαριού το οποίο μπορούσε επίσης να πετάει. (J.D.)



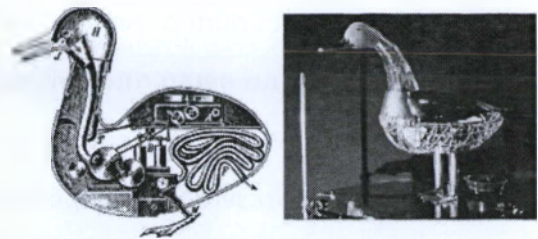
Εικ.9 John Dee, ξύλινο σκαθάρι

Ο Blaise Pascal εφηύρε τη μηχανική αριθμομηχανή το 1642, η οποία μπορούσε να προσθέσει και να αφαιρέσει δύο αριθμούς αυτόματα. (B.P.) Τον ακολούθησε ο Γιονanni Poleni ο οποίος έχτισε τη δεύτερη λειτουργική μηχανική αριθμομηχανή το 1709, ένα υπολογιστικό ρολόι, το οποίο ήταν φτιαγμένο από ξύλο και αφού ρυθμιζόταν μπορούσε να πολλαπλασιάσει δύο αριθμούς αυτόματα. (G.P.) Γύρω από το 1700 κατασκευάστηκαν πολλά αυτόματα , συμπεριλαμβανομένων και εκείνων που είχαν την ικανότητα να υποκρίνονται, να σχεδιάζουν, να πετάνε, και να παίζουν μουσική.

Μερικά από τα πιο διάσημα έργα της περιόδου δημιουργήθηκαν από τον Jacques de Vaucanson το 1737,

συμπεριλαμβανομένου ενός φλαουτίστα αυτόματου, ένα τυμπανιστή, και το πιο

φημισμένο έργο του, "Digesting Duck" (" Η Πάπια που Αφομοιώνει"). Η Πάπια του Vaucanson τροφοδοτούταν με σταθμά και ήταν σε θέση να μιμείται μια πραγματική πάπια με το χτύπημα των φτερών της (πάνω από 400 μέρη σε κάθε ένα από τα φτερά), τρώγοντας σιτηρά, χωνεύοντας, και αφοδεύοντας με το να αποθηκεύει την εκκρινόμενη ύλη σε ένα κρυφό διαμέρισμα. (J.deV.)



Εικ.10 Digesting Duck

Ο John Kay εφεύρε την «Πτάμενη Σαΐτα» (Flying Shuttle) το 1733, και το "Spinning Jenny" επινοήθηκε το 1764 από τον James Hargreaves . Καθ' ένα από αυτά αύξησε ριζικά την ταχύτητα της παραγωγής στις βιομηχανίες ύφανσης και κλωστοβιομηχανίες αντίστοιχα. (Shuttle) Το Spinning Jenny τροφοδοτούταν με το χέρι

και απαιτούσε έμπειρο χειριστή, ενώ το Spinning Mule του Samuel Crompton, που αναπτύχθηκε για πρώτη φορά το 1779, είναι μια πλήρως αυτοματοποιημένη μηχανοκίνητη μηχανή ικανή να υφαίνει εκατοντάδες νήματα με τη μία. (Mule) Ο Richard Arkwright έχτισε μια τροφοδοτούμενη με νερό μηχανή ύφανσης και ένα εργοστάσιο, πάνω στο Spinning Mule το 1781, ως κινητήριο αρχή για την Βιομηχανική Επανάσταση. (R.A.)



Εικ.11 Richard Arkwright, μηχανή ύφανσης

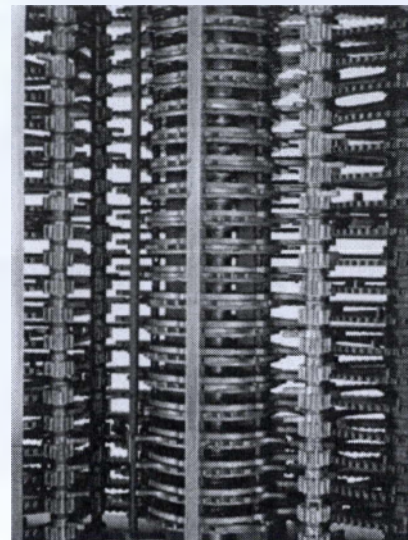
Ο Ιάπωνας τεχνίτης Hisashige Tanaka , που είναι γνωστός ως ο «Edison της Ιαπωνίας», δημιούργησε μια σειρά από εξαιρετικά περίπλοκα μηχανικά παιχνίδια, μερικά από τα οποία ήταν σε θέση να σερβίρουν τσάι, να πετάνε βέλη από μια φαρέτρα, ή ακόμη και να ζωγραφίζουν ένα ιαπωνικό ιδεόγραμμα



Εικ.12 Ρολόι του Tanaka

(kanji) . Το ορόσημο κείμενο Karakuri Zui (Εικονογραφημένα Μηχανήματα), δόθηκε στη δημοσιότητα το 1796. (Tanaka)

Από το 1800, η παραγωγή υφασμάτων ήταν πλήρως αυτοματοποιημένη. Με την έλευση της βιομηχανικής επανάστασης, η ιδέα των αυτόματων άρχισε να εφαρμόζεται στη βιομηχανία, σαν συσκευές εξοικονόμησης κόστους και χρόνου. Βελτιώσεις στην βιομηχανία ύφανσης είχαν οδηγήσει σε μεγάλες ποσότητες αυτοματισμού, και η ιδέα των προγραμματιζόμενων μηχανών έγινε δημοφιλής με την "Analytical Engine" (Αναλυτική Μηχανή) του Charles Babbage . Ο Babbage συνέλαβε την



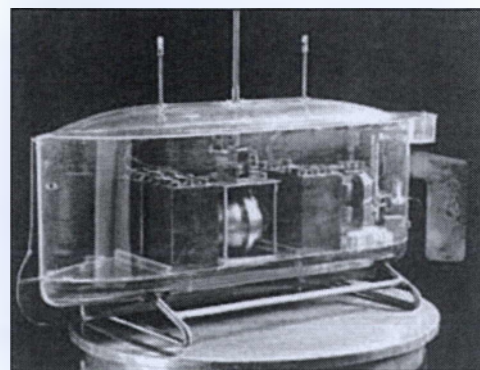
Εικ.13 Αναλυτική Μηχανή, πειραματικό μοντέλο, 1871

Αναλυτική Μηχανή του ως αντικατάσταση της μη ολοκληρωμένης Difference Engine , η οποία ήταν μεγαλύτερη, πιο πολύπλοκη, σε θέση να εκτελεί πολλαπλές λειτουργίες και να λειτουργεί από διατρητές κάρτες. Η κατασκευή της Αναλυτικής Μηχανής είχε ξεκινήσει το 1833 αλλά το έργο δεν ολοκληρώθηκε ποτέ. (engine)Ωστόσο, οι εργασίες της Ada Lovelace για το έργο της είχε ως αποτέλεσμα την αναγνώριση της ως τον πρώτο προγραμματιστή ηλεκτρονικών υπολογιστών. (A.lovelace)

Στη λογοτεχνία, γύρω σε αυτή τη χρονική περίοδο, ο Mary Shelley έγραψε το Φρανκενστάιν, το οποίο περιγράφει ένα τέρας που αποτελείται από νεκρή σάρκα και επανήλθε στη ζωή με τη βοήθεια της ηλεκτρικής ενέργειας. (frankenstein)

Το 1837, η ιστορία του Γκόλεμ της Πράγας εξιστορεί ένα τεχνητής νοημοσύνης ανθρωποειδές που ενεργοποιείται αναγράφοντας γράμματα του Hebrew στο μέτωπό του. (golem)Η ιστορία είναι βασισμένη στον George Boole που εφηύρε ένα νέο είδος συμβολικής λογικής το 1847, η οποία ήταν καθοριστική για τη δημιουργία των ηλεκτρονικών υπολογιστών και των ρομπότ. (G.Boole)

Το 1898 ο Nikola Tesla παρουσίασε δημοσίως ένα τηλεκατευθυνόμενο (teleoperated) σκάφος, παρόμοιο με ένα σύγχρονο ROV . Με βάση τα διπλώματα ευρεσιτεχνίας του U.S. Patent 613,809, U.S.



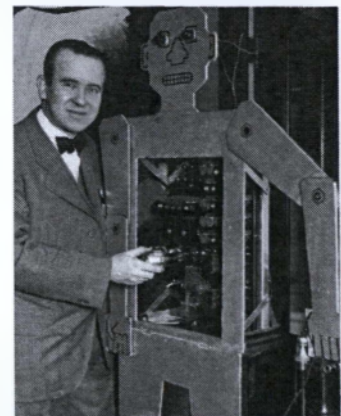
Εικ.14 Nicola Tesla, Τηλεκατευθυνόμενο σκάφος

Patent 723,188 και U.S. Patent 725,605 για το τηλεαυτοματισμό ("teleautomation"), ο Tesla ήλπιζε να αναπτύξει την ασύρματη τορπίλη σε ένα οπλικό σύστημα για το Πολεμικό Ναυτικό των ΗΠΑ (Cheney 1989). (Tesla)

1.2.3 Από το 1901 έως το 1950

Η λέξη ρομπότ διαδόθηκε από τον Τσέχο συγγραφέα Karel Čapek το 1921 στο έργο του RUR (Rossum's Universal Robots) . Σύμφωνα με τον Karel, ο αδελφός του Ιωσήφ ήταν ο πραγματικός εφευρέτης της λέξης "ρομπότ", δημιουργώντας τη λέξη από την Τσέχικη λέξη "robota", που σημαίνει δουλεία. (rur)

Το 1927, το έργο Metropolis (Μητρόπολη) του Fritz Lang παρουσιάζει το Maschinenmensch ("μηχανής-ανθρώπου"), ένα γυνοειδ ανθρωποειδές ρομπότ, που ονομάζεται επίσης και "Parody", "Futura", "Robotrix", ή "Maria impersonator" (που παίχτηκε από τη γερμανίδα ηθοποιό Brigitte Helm), το οποίο ήταν το πρώτο ρομπότ που απεικονίζεται σε ταινία. (Metropolis) Το πρώτο πραγματικό ρομπότ στον κόσμο είναι το ανθρωποειδές που ονομάζεται Televox, το οποίο λειτουργεί μέσω του τηλεφωνικού συστήματος και κατασκευάστηκε στις Ηνωμένες Πολιτείες το 1927. (Westinghouse)

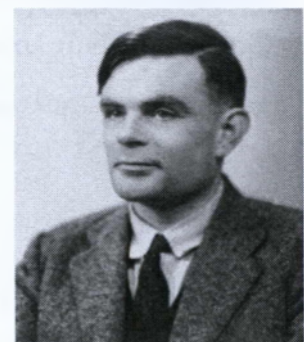


Εικ.15 Ρομπότ Televox

Το 1928, ο βιολόγος Makoto Nishimura σχεδίασε και κατασκεύασε το πρώτο ρομπότ της Ιαπωνίας, εν ονόματι Gakutensoku. (Gakutensoku)

Ο Vannevar Bush δημιούργησε το πρώτο διαφορικό αναλυτή στο Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT) το 1931, έναν υπολογιστή που μπορούσε να λύνει διαφορικές εξισώσεις με ολοκλήρωση. (V.B.)

Στο έγγραφο του το 1936 "On Computable Numbers, With An Application To The Entscheidungsproblem" (που υποβλήθηκε στις 28 Μαΐου 1936), ο Alan Turing αναδιατύπωσε τα αποτελέσματα του Kurt Gödel (1931) για τα όρια της απόδειξης και του υπολογισμού, αντικαθιστώντας την καθολική αριθμητική



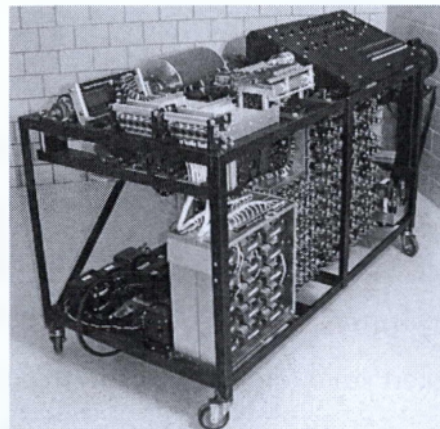
Εικ.16 Alan M. Turing (23 Ιουνίου 1912 – 9 Ιουνίου 1954)

βάση της επίσημης γλώσσας του Gödel με ότι σήμερα ονομάζουμε μηχανές Turing (Turing machine). Απέδειξε ότι μια τέτοια μηχανή θα ήταν σε θέση να εκτελέσει οποιαδήποτε πιθανούς μαθηματικούς υπολογισμούς, αν ήταν αντιπροσωπευτικός ως αλγόριθμος, δημιουργώντας έτσι τη βάση για την επιστήμη που τώρα ονομάζουμε επιστήμη των υπολογιστών. (A.T.1936)

Πολλά ρομπότ είχαν κατασκευαστεί πριν από την αυγή (άνθηση) των computer-controlled servomechanisms (σερβομηχανισμών ελεγχόμενων με υπολογιστή), με σκοπό τις δημόσιες σχέσεις των μεγάλων επιχειρήσεων. Ο Elektro εμφανίστηκε στο περίπτερο της Westinghouse στη Διεθνή Έκθεση της Νέας Υόρκης το 1939.

(Westinghouse) Κάποια δημιουργήθηκαν μεταξύ αυτών των μεγάλων δημόσιων συγκεντρώσεων, όπως ο Garco, που κατασκευάστηκε από την Garrett AiResearch στη δεκαετία του 1950. (Garco) Επρόκειτο κυρίως για μηχανήματα που μπορούσαν να εκτελέσουν μερικές φιγούρες, όπως τα αυτόματα του 18ου αιώνα.

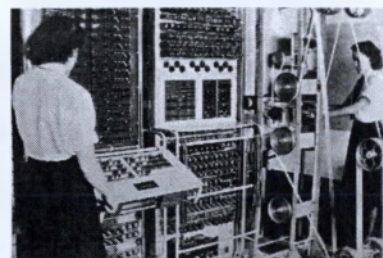
Το 1940 επέφερε την δημιουργία δύο ηλεκτρικών υπολογιστών, τον Atanasoff-Berry Computer (ABC) του John Vincent και τον Atanasoff του Clifford Berry. (abc) Έννοιες του ABC χρησιμοποιήθηκαν αργότερα στο λειτουργικό ENIAC.



Εικ.17 Αντίγραφο του ABC, 1996

Το 1941 και το 1942, ο Isaac Asimov διατύπωσε τους Τρεις Νόμους της Ρομποτικής και πάνω στη διαδικασία της υλοποίησης, έπλασε τη λέξη «ρομποτική». (I.Asimov)

Η μηχανή Heath Robinson είχε σχεδιαστεί για τη βρετανική πολεμική προσπάθεια να σπάσει Αινιγματικά (Enigma) μηνύματα. Αυτό έγινε στο Government Code and Cypher School (GC&CS) στο Bletchley Park, και η νοημοσύνη που χρησιμοποιήθηκε ονομάστηκε Ultra (Robinson). Η Heath Robinson αντικαταστάθηκε από τον Colossus (Κολοσσό), που κατασκευάστηκε το 1943 από τον Tommy Flowers, για να αποκωδικοποιήσει FISH μηνύματα από τον βρετανικό όμιλο Ultra. (Colossus) Ήταν 100 με 1000 φορές γρηγορότερος από του



Εικ.18 Ο Κολοσσός του πάρκου Bletchley, 1944

Robinson, και ήταν ο πρώτος πλήρως ηλεκτρονικός υπολογιστής. (Bletchley) Οι μηχανές Bletchley κρατήθηκαν μυστικές για δεκαετίες, και έτσι δεν εμφανίζονταν στις ιστορίες των υπολογιστών μέχρι πρόσφατα. Μετά τον πόλεμο, ο Tommy Flowers προσχώρησε στην ομάδα που κατασκεύασε τους πρώτους υπολογιστές Μάντσεστερ.

Στη Γερμανία, ο Konrad Zuse κατασκεύασε το πρώτο πλήρως προγραμματιζόμενο ψηφιακό υπολογιστή στον κόσμο (Z3) το 1941, ο οποίος καταστράφηκε σε έναν βομβαρδισμό το 1944 (Z3). Ο Zuse ήταν επίσης γνωστός για την κατασκευή του πρώτου δυαδικού υπολογιστή (Z1) από το 1936 μέχρι το 1938, και για την κατασκευή του Z4, το μόνο μηχάνημα του που επιβίωσε τον Β' Παγκόσμιο Πόλεμο. (Zuse)

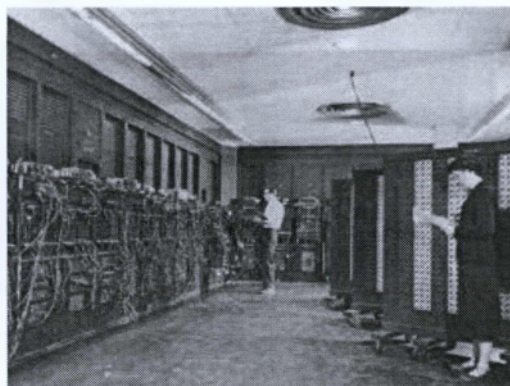


Εικ.19 Υπολογιστής Z4, ETH-Zürich

Ο πρώτος Αμερικάνικος προγραμματιζόμενος υπολογιστής ολοκληρώθηκε το 1944 από τον Howard Aiken και τον Grace Hopper. Ο Mark I (όπως ονομαζόταν) έτρεχε υπολογισμούς για το Πολεμικό Ναυτικό των ΗΠΑ μέχρι το 1959. (MarkI)

Ο ENIAC χτίστηκε το 1946 και κέρδισε τη φήμη λόγω της αξιοπιστίας, της ταχύτητας, και της ευελιξίας. Ο John Presper Eckert και ο John W. Mauchly πέρασαν 3 χρόνια κατασκευάζοντας τον ENIAC, που ζύγιζε πάνω από 60.000 lbs. (Eniac)

Το 1948, ο Norbert Wiener διατύπωσε τις αρχές της κυβερνητικής (cybernetics), δηλαδή, τη βάση της πρακτικής ρομποτικής. (N.Wiener)



Εικ.20 ENIAC

Τα πρώτα αυτόματα ρομπότ χελώνες (Elmo και Elsie) δημιουργήθηκαν από τον πρωτοπόρο της ρομποτικής William Grey Walter το 1949. (Walter)

Ο πρώτος ψηφιακός υπολογιστής προς πώληση ήταν ο Z4 του Zuse στη Γερμανία, το 1949 (Zuse). Δώδεκα μήνες νωρίτερα, τον Σεπτέμβριο του 1949, ο BINAC πωλήθηκε στη Northrop, αλλά ποτέ δεν λειτούργησε με αξιοπιστία, λόγω κακού χειρισμού κατά τη μεταφορά (binac). Τρίτος ήταν ο Ferranti Mark 1 του Ηνωμένου Βασιλείου, που παραδόθηκε τον Φεβρουάριο του 1951. Ήταν ο πρώτος ψηφιακός ηλεκτρονικός υπολογιστής με προγραμματιζόμενο λογισμικό, βασισμένος στο πρώτο

στον κόσμο ψηφιακό ηλεκτρονικό υπολογιστή με προγραμματιζόμενο λογισμικό, τον Μάντσεστερ SSME του 1948. (mosi)

Το 1950, το UNIVAC I (επίσης από τον Eckert και Mauchley) χειρίστηκε τα αποτελέσματα της Απογραφής των ΗΠΑ. Ήταν ο τρίτος στο εμπόριο ηλεκτρονικών υπολογιστών που λειτούργησε κατά την παράδοση (τον Δεκέμβριο του 1951). (Univac)



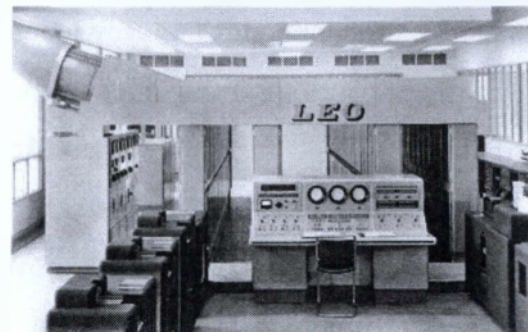
Εικ.21 UNIVAC I

Το Turing τεστ προτάθηκε από τον Alan Turing το 1950 στο έγγραφο του «Computing Machinery and Intelligence», το οποίο ξεκινάει με τις λέξεις: «Εγώ προτείνω να εξεταστεί το θέμα: Μπορούν οι μηχανές να σκέφτονται;». (T.)

1.2.4 Από το 1951 έως το 1960

Το 1951 ο William Shockley εφεύρε το διπολικό τρανζίστορ (bipolar junction transistor, BJT, bipolar transistor), που ανακοινώθηκε σε μια συνέντευξη Τύπου, στις 4 Ιουλίου του 1951 (shock).

Το 1951 ένας υπολογιστής που ονομάζεται LEO άρχισε να λειτουργεί στο Ηνωμένο Βασίλειο. Κατασκευάστηκε από τη J. Lyons & Co για προσωπική τους χρήση, και με την υποστήριξη του σχεδίου Cambridge EDSAC. Ήταν ο πρώτος στον κόσμο ψηφιακός ηλεκτρονικός υπολογιστής με λογισμικό προγραμματισμού για εμπορικές εφαρμογές, αξιοποιώντας την



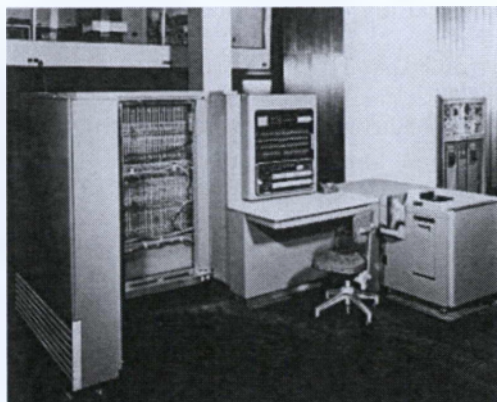
Εικ.22 LEO

ανάπτυξη των ΗΠΑ στη γραμμή καθυστέρησης με μνήμη υδραργύρου. Ο LEO χρησιμοποιούνταν για εμπορικές εργασίες τρέχοντας προγράμματα επιχειρησιακών εφαρμογών, η πρώτη εκ των οποίων πραγματοποιήθηκε στις 5, Σεπτέμβρη του 1951. (LEO)

Ο J. Presper Eckert και ο John Mauchly ολοκλήρωσαν τον EDVAC το 1951, αλλά η λειτουργία του ήταν περιορισμένη μέχρις ότου διορθωθούν όλα τα μικροπροβλήματα του. Ποιο βελτιωμένος από τον ENIAC και τον UNIVAC, ο EDVAC είναι ο πρώτος

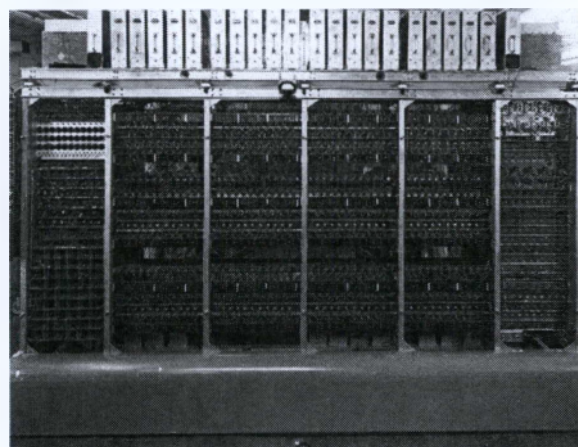
δυναμικός υπολογιστής που αποθηκεύει πρόγραμμα και δεδομένα χρησιμοποιώντας υδραργυρικές γραμμές καθυστέρησης. (edvac) Το 1952, ο UNIVAC προέβλεψε σωστά την εκλογή του Dwight D. Eisenhower ως πρόεδρο, κατά τη διάρκεια μια τηλεοπτικής εκπομπής ειδήσεων. Ο UNIVAC παραδόθηκε στις 14 Ιουνίου του 1951 από τους κατασκευαστές του, Eckert και Mauchly. Από το 1951 μέχρι το 1958 σταδιακά παραδόθηκαν 46 υπολογιστές UNIVAC. (Univac)

Το 1952, η IBM ανακοινώνει τον IBM 701 υπολογιστή που διατίθενται στην αγορά για επιστημονική χρήση, σχεδιασμένο από τον Nathaniel Rochester. Ο IBM 701 ήταν ένας δυναμικός (σε αντίθεση με το δεκαδικό) λογικός υπολογιστής κενού σωλήνα με 36-bit λέξεων. (ibm)



Εικ.23 IBM 701

Το Μάρτιο του 1952 λειτουργεί επιτυχημένα ο υπολογιστής MANIAC I, κατασκευασμένος από μια ομάδα με επικεφαλής τον John von Neumann και τον Nicholas Metropolis. Η πρώτη του δουλειά ήταν η επίλυση των υπολογισμών που απαιτούνται για να κατασκευαστεί το πρωτότυπο της βόμβας υδρογόνου, όπου και τα κατάφερε με επιτυχία. Έπειτα, χρησιμοποιήθηκε το 1956 για να παίξει ένα τροποποιημένο παιχνίδι σκακιού, όπου για πρώτη φορά ένας υπολογιστής νικάει έναν άνθρωπο σε ένα παιχνίδι ευφυΐας. (maniac1)

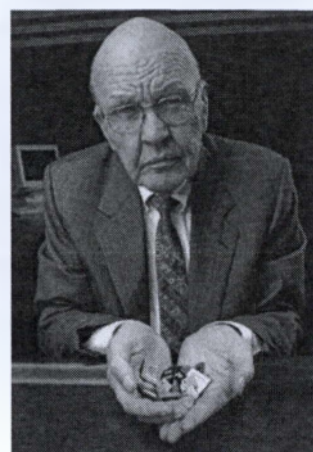


Εικ.24 MANIAC, με ανοιχτά καλύμματα

Ο όρος «Τεχνητή Νοημοσύνη» δημιουργήθηκε σε μια διάσκεψη που πραγματοποιήθηκε στο Dartmouth College το 1956. Ο Allen Newell, ο JC Shaw και ο Herbert A. Simon πρωτοστάτησαν στο νεοσύστατο τμήμα της τεχνητής νοημοσύνης με το Logic Theory Machine (Μηχάνημα λογικής θεωρίας) (1956), και το General Problem Solver (G.P.S.) το 1957. Το 1958, ο John McCarthy και ο Marvin Minsky ξεκίνησαν το MIT Artificial Intelligence lab (εργαστήριο Τεχνητής Νοημοσύνης του MIT) με \$ 50.000. (A.I.mit) Ο John McCarthy επίσης δημιούργησε τη LISP, το καλοκαίρι του 1958,

μια γλώσσα προγραμματισμού που εξακολουθεί να είναι σημαντική στον τομέα της τεχνητής νοημοσύνης. (lisp)

Ο Jack Kilby μαζί με το Robert Noyce εφηύρε το ολοκληρωμένο κύκλωμα ή "τσιπ" το 1959. Οι εφευρέτες εργάστηκαν ανεξάρτητα από κάθε άλλον και κατάφεραν να αναπτύξουν αυτή την επαναστατική εύρεση για του υπολογιστές, που επηρέασε τόσο το μέγεθος όσο και την ταχύτητα. (chip)

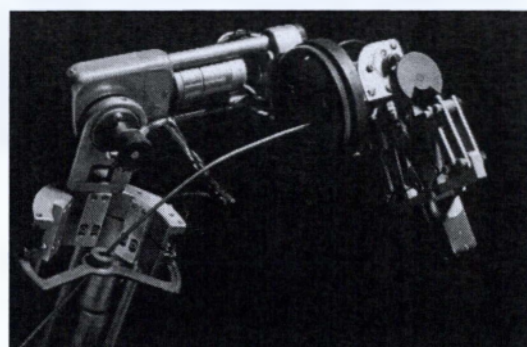


Εικ.25 Jack Kilby

1.2.5 Από το 1961 έως το 1971

Το Unimate, που επινοήθηκε και κατασκευάστηκε το 1954 από τον George Devol, είναι το πρώτο βιομηχανικό ρομπότ που δημιουργήθηκε ποτέ, και άρχισε να συμμετέχει από το 1961 στην γραμμή συναρμολόγησης για την General Motors. (unimate)

Το 1962 ο John McCarthy ίδρυσε το Εργαστήριο Τεχνητής Νοημοσύνης του Στάνφορντ στο Πανεπιστήμιο του Στάνφορντ (Stanford University). Ο βραχίονας Rancho αναπτύχθηκε ως ένα ρομποτικό βραχίονα για να βοηθάει άτομα με ειδικές ανάγκες στο Rancho Los Amigos Νοσοκομείο στο Ντάουνι της Καλιφόρνιας. Αγοράστηκε από το Πανεπιστήμιο του Στάνφορντ το 1963, και κατέχει μια θέση μεταξύ των πρώτων τεχνητών ρομποτικών βραχιόνων που ελέγχονται από έναν υπολογιστή. (ranchoa.)



Εικ.26 Rancho Arm

Η IBM ανακοίνωσε το IBM System/360 το 1964. Το σύστημα είχε αναδειχθεί ως πιο ισχυρό, πιο γρήγορο και πιο ικανό από τους προκατόχους του. (System/360)

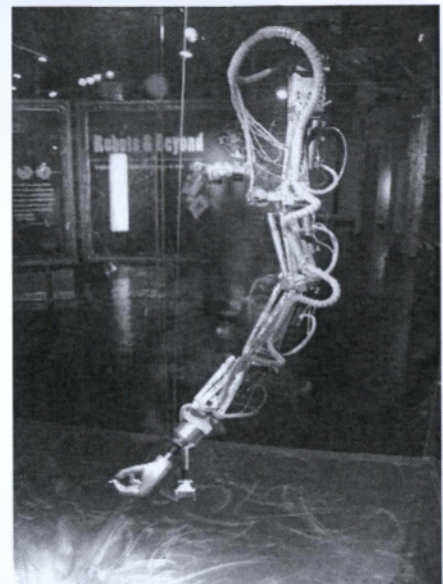
Το 1965, ο Gordon Moore, συνιδρυτής της Intel το 1968, αναπτύσσει αυτό που θα γίνει γνωστό ως ο νόμος του Moore, την ιδέα ότι ο αριθμός των στοιχείων που είναι ικανά να χτίζονται πάνω σε ένα τσιπ θα διπλασιάζεται κάθε δύο χρόνια. (Moore) Το ίδιο έτος ο διδακτορικός φοιτητής Edward Feigenbaum, ο γενετιστής και βιοχημικός Joshua

Lederberg και ο Norman Buchanan (που κατείχε πτυχίο φιλοσοφίας) ξεκίνησαν τις εργασίες πάνω στην DENDRAL, ένα έμπειρο σύστημα σχεδιασμένο για την εργασία στον τομέα της οργανικής χημείας. (Dendral) Επίσης, Ο Φαιγκενμπάουμ (Feigenbaum) ίδρυσε το Heuristic Programming Project (Πρόγραμμα Ευρετικού Προγραμματισμού) το 1965, το οποίο έγινε αργότερα το Εργαστήριο Τεχνητής Νοημοσύνης των Συστημάτων Γνώσης του Στάνφορντ. (Edward F.)

Το πρόγραμμα Mac Hack γράφτηκε το 1966 από τον Richard Greenblatt, το οποίο νίκησε τον κριτικό τεχνητής νοημοσύνης Hubert Dreyfus σε μια παρτίδα σκάκι. (Mac)

Ο Seymour Papert δημιούργησε τη γλώσσα προγραμματισμού Logo το 1967, η οποία είχε σχεδιαστεί ως μια εκπαιδευτική γλώσσα προγραμματισμού. (logo)

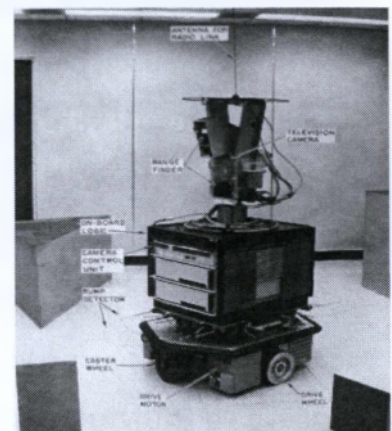
Η ταινία 2001: Οδύσσεια του Διαστήματος (2001: A Space Odyssey) κυκλοφόρησε το 1968, όπου κατέχει εξέχουσα θέση ο HAL 9000, μια κακόβουλη μονάδα τεχνητής νοημοσύνης που ελέγχει ένα διαστημικό σκάφος. (hal9000) Ο Marvin Minsky δημιούργησε τον βραχίονα πλοκάμι το 1968. Ο βραχίονας ελεγχόταν με υπολογιστή και οι 12 αρθρώσεις του τροφοδοτούνταν με υδραυλικά συστήματα. (tentacle)



Εικ.27 Tentacle Arm

Ο Μηχανολόγων Μηχανικών φοιτητής Victor Scheinman δημιούργησε τον βραχίονα του Στάνφορντ το 1969, που αναγνωρίζεται ως το πρώτο ρομποτικό βραχίονα που ελέγχεται με ηλεκτρονικό υπολογιστή (οι Unimate εντολές ήταν αποθηκευμένες σε ένα μαγνητικό τύμπανο). (Stanford.arm)

Η πρώτη δισκέτα κυκλοφόρησε το 1970 με οκτώ εκατοστά σε διάμετρο και μόνο για ανάγνωση. (Floppy.disk) Το πρώτο κινητό ρομπότ ικανό να συλλογιστεί σχετικά με το περιβάλλον του είναι το Shakey το οποίο κατασκευάστηκε, επίσης το 1970 από το Stanford Research Institute (τώρα SRI



Εικ.28 Το ρομπότ Shakey

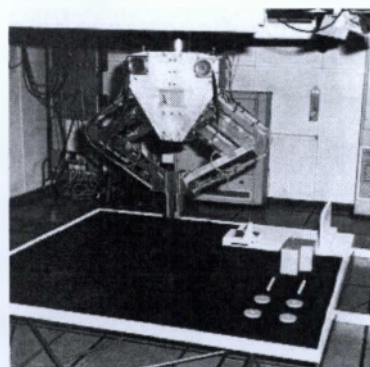
International). Το Shakey συνδύαζε πολλαπλές εισόδους αισθητήρων, συμπεριλαμβανομένου κάμερες τηλεόρασης, αποστασιομετρικά λέιζερ και αισθητήρες σύγκρουσης για να πλοηγηθεί. (Shakey) Το χειμώνα του 1970, η Σοβιετική Ένωση διερεύνησε την επιφάνεια του φεγγαριού με το σεληνιακό όχημα Lunokhod 1, το πρώτο περιπλανώμενο τηλεκατευθυνόμενο ρομπότ που προσγειώθηκε σε άλλο κόσμο. (lunokhod1)

Ο πρώτος μικροεπεξεργαστής, που ονομάζεται 4004, δημιουργήθηκε από τον Ted Hoff στην Intel το 1971, με διαστάσεις 1/8 της ίντσας - 1/16 της ίντσας, το τσιπ ήταν πιο ισχυρό από τον ENIAC. (c4004)

1.2.6 Από το 1972 έως το 1980

Ο κριτικός τεχνητής νοημοσύνης Hubert Dreyfuss δημοσίευσε το σημαντικό βιβλίο του «αυτό που οι υπολογιστές δεν μπορούν ακόμα να κάνουν» (What Computers Still Can't Do) το 1972. (H.Dreyfuss) Το "Silent Running" του Douglas Trumbull κυκλοφόρησε επίσης το 1972, η ταινία ήταν αξιοσημείωτη για τα τρία ρομπότ συμπρωταγωνιστές, που ονομάζονταν Huey, Dewey και Louie. (S.Running) Το 1972 ξεκίνησε το έμπειρο σύστημα MYCIN, το οποίο συντάχθηκε στην Lisp, και αναπτύχθηκε για να μελετήσουν τις αποφάσεις και τις προδιαγραφές που σχετίζονται με λοιμώξεις του αίματος. (mycin)

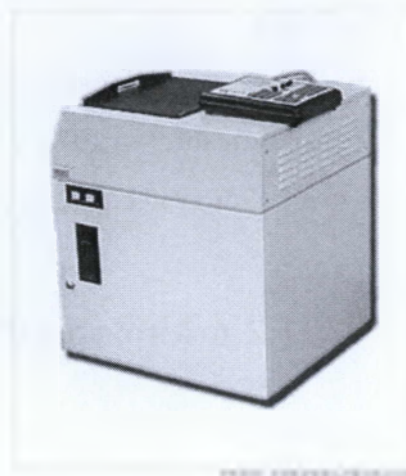
Το 1973 κυκλοφόρησε η βασική λογική γλώσσα προγραμματισμού PROLOG, όπου και καθίσταται σημαντική στον τομέα της τεχνητής νοημοσύνης. (prolog) Οι Freddy και Freddy II, και οι δυο κατασκευασμένοι στο πανεπιστήμιο του Edinburgh (Ηνωμένο Βασίλειο), ήταν ρομπότ ικανά να συναρμολογούν ξύλινα μπλοκ σε μια χρονική περίοδο αρκετών ωρών. (Freddy) Τον ίδιο χρόνο, η εταιρεία KUKA με έδρα την Γερμανία έχτισε το πρώτο βιομηχανικό ρομπότ στον κόσμο με έξι ηλεκτρομηχανικούς άξονες οδήγησης, που είναι γνωστή ως Famulus. (Famulus)



Εικ.29 Freddy II, περίπου το 1973

Το 1974, ο David Silver σχεδίασε το «The Silver Arm» το οποίο ήταν ικανό για κινήσεις ακριβείας αναπαριστώντας τα ανθρώπινα χέρια. Η ανάδραση προμηθευόταν με την αφή και τους αισθητήρες πίεσης, αφού αναλύονταν από έναν υπολογιστή. (TSA) Τον ίδιο χρόνο, ο Marvin Minsky δημοσίευσε το έγγραφο ορόσημο του «Ένα πλαίσιο για την αναπαράσταση της γνώσης» (A Framework for Representing Knowledge) στην τεχνητή νοημοσύνη. (M.M.)

Το Kurzweil Reading Machine (μηχάνημα ανάγνωσης Kurzweil), που εφευρέθηκε από τον Raymond Kurzweil με σκοπό να βοηθήσει τους τυφλούς, κυκλοφόρησε το 1976. Ικανό να αναγνωρίζει τους χαρακτήρες, το μηχάνημα διατύπωνε την ομιλία του με βάση τους προγραμματισμένους κανόνες. (KRM)



Εικ.30 kurzweil reading machine

Με βάση τις μελέτες των ευέλικτων αντικειμένων στη φύση (όπως κορμούς ελέφαντα και τους σπονδύλους των φιδιών), ο Shigeo Hirose σχεδίασε το Soft Gripper το 1976, όπου η λαβή του ήταν ικανή να ανταποκρίνεται σύμφωνα με το αντικείμενο που είχε πιάσει. (SGr)

Το γνωστικό σύστημα Automated Mathematician (AM) (Αυτοματοποιημένη Μαθηματικός) παρουσιάστηκε από τον Douglas Lenat το 1976 ως μέρος της διδακτορικής του διατριβής. Το Automated Mathematician χρησιμοποιεί ένα σύνολο 115 θεμελιωδών αρχών, όπως έννοιες και διασταυρώσεις των συνόλων, και ένα σύνολο 243 αιρετικών (πειραματικών) κανόνων για να τους χειριστεί και να τους οδηγήσει σε κλασικούς μαθηματικούς ορισμούς και συγκυρίες. (A.Math) Το ίδιο έτος, ο Joseph Weizenbaum (δημιουργός του ELIZA, ένα πρόγραμμα ικανό να προσομοιώσει έναν Rogerian ψυχοθεραπευτή) δημοσίευσε το Computer Power and Human Reason (η δύναμη του υπολογιστή και η ανθρώπινη λογική), παρουσιάζοντας ένα επιχειρήμα κατά της δημιουργίας της τεχνητής νοημοσύνης. (J.Weizenbaum)

Ο Prospector είναι ένα computer-based πρόγραμμα διαβούλευσης για την εξερεύνηση ορυκτών, το οποίο δημιουργήθηκε το 1977 και ανακάλυψε ένα άγνωστο κοίτασμα μολυβδαινίου στην πολιτεία της Ουάσινγκτον. Το έμπειρο σύστημα ενημερώνεται σε ετήσια βάση από την ίδρυσή του. (Prospector)

Ο Steven Jobs και ο Stephen Wozniak δημιούργησαν την εταιρία της Apple το 1977, όπου και κυκλοφόρησε ο υπολογιστής Apple II, χωρίς να γνωρίζουν ότι θα καταφέρουν να γίνουν από τους μεγαλύτερους τεχνολογικούς κολοσσούς του αιώνα μας. (AppleII) Στην ταινία του George Lucas «Star Wars», που



Εικ.32 Apple II

κυκλοφόρησε επίσης το 1977, παρουσιάζονται δύο ανδροειδή ρομπότ που ονομάζονται C-3PO και R2-D2. (S.Wars) Οι Voyagers 1 και 2 ξεκίνησαν το 1977 για να διερευνήσει το ηλιακό σύστημα. Τα 30 χρονών ρομποτικά διαστημικά συνεχίζουν να μεταδίδουν δεδομένα πίσω στη γη και πλησιάζουν την heliopause (ηλιόπαυση) και το interstellar medium (διαστρικό μέσο). (Voyagers.1.2)

Το πρώτο ρομπότ SCARA, Selective Compliance Assembly Robot Arm (Επιλεκτική Συνέλευση Συμμόρφωσης ρομποτικού βραχίονα), δημιουργήθηκε το 1978 ως ένα αποτελεσματικό, τεσσάρων αξόνων ρομποτικό βραχίονα. Το SCARA υπερέχει στη «pick and place» λειτουργία, δηλαδή τη μοναδική ικανότητα του να παίρνει βιομηχανικά εξαρτήματα από μία θέση και να τα τοποθετεί τα σε μια άλλη, με ακρίβεια, ταχύτητα και ομαλή κίνηση. (το SCARA εισήχθη σε γραμμές συναρμολόγησης το 1981). (Scara)



Εικ.31 Το πρώτο ρομπότ SCARA

Το XCON κυκλοφόρησε το 1979, ως ένα έμπειρο σύστημα σχεδιασμένο για να προσαρμόζει τις παραγγελίες για τη βιομηχανική χρήση. (Xcon) Τον ίδιο χρόνο, το Stanford Cart διέσχισε με επιτυχία ένα δωμάτιο γεμάτο καρέκλες, χωρίς ανθρώπινη παρέμβαση. Το Stanford Cart (Καλάθι Στάνφορντ) στηρίχθηκε κυρίως στη στερεοσκοπική όραση για να πλοηγηθεί και να καθορίσει τις αποστάσεις. (S.Cart)

Το Ρομποτικό Ινστιτούτο στο Carnegie Mellon University ιδρύθηκε το 1979 από τον Raj Reddy. (Raj.R.)

1.2.7 Από το 1981 έως το 1990

Ο Takeo Kanade δημιούργησε το πρώτο «άμεσης κίνησης βραχίονα» (Direct Drive Arm) το 1981. Οι κινητήρες του βραχίονα περιέχονται στο ίδιο το ρομπότ έτσι ώστε να εξαλείφουν τις πολύχρονες μεταδόσεις. (dda) Η IBM κυκλοφόρησε τον πρώτο προσωπικό υπολογιστή (PC) το 1981, και το όνομα του υπολογιστή ήταν υπεύθυνο για την εκκλαίκευση του όρου "προσωπικός υπολογιστής". (ibmpc)

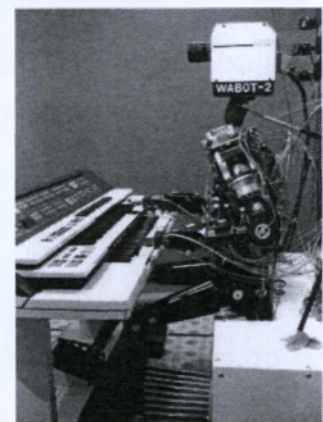


Εικ.33 IBM PC, 1981

Το πρόγραμμα της Πέμπτης γενιάς Σχεδιασμού Υπολογιστικών Συστημάτων (FGCS) (Fifth Generation Computer Systems) ξεκίνησε το 1982 με τη πρωτοβουλία του Ιαπωνικού Υπουργείου Διεθνούς Εμπορίου και Βιομηχανίας. Είχε ως στόχο τη δημιουργία ενός κοσμοϊστορικού υπολογιστή με υπερυπολογιστικές επιδόσεις και να παρέχει μια πλατφόρμα για τις μελλοντικές εξελίξεις στον τομέα της τεχνητής νοημοσύνης. (progr.)

Το πρώτο πρόγραμμα που χρησιμοποιήθηκε για να δημοσιεύσει ένα βιβλίο ήταν το έμπειρο σύστημα Racter (expert system Racter), που έχει προγραμματιστεί από το William Chamberlain και το Thomas Etter, το οποίο έγραψε το βιβλίο «The Policeman's Beard is Half-Constructed »το 1983. (racter)

Ο Douglas leant ξεκίνησε το 1984 ένα έργο για τη δημιουργία μιας βάσης δεδομένων της κοινής λογικής για τη τεχνητή νοημοσύνη, με όνομα cyc. (cyc) Επίσης, το 1984 αποκαλύφθηκε το Wabot-2, ένα ρομπότ με 10 δάχτυλα και 2 πόδια ικανό να επικοινωνεί με ένα άτομο, να διαβάζει παρτιτούρες μουσικής και να παίζει μουσική μέτριας δυσκολίας σε ένα ηλεκτρονικό όργανο. (wabot2)



Εικ.34 Wabot-2

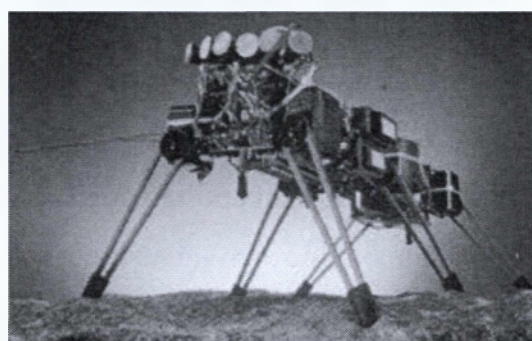
Το 1985,η συμφωνία άδειας χρήσης (license agreement) της «Kawasaki Heavy Industries» με την Unimation τερματίστηκε και η Kawasaki άρχισε να παράγει τα δικά της ρομπότ, με το πρώτο ρομπότ τους να κυκλοφορεί ένα χρόνο αργότερα. Μέχρι το

1986 τα έσοδα της τεχνητής νοημοσύνης ήταν περίπου \$ 1 δισεκατομμύρια δολάρια. Το 1986 η Honda ξεκίνησε την έρευνα ανθρωποειδών και το πρόγραμμα ανάπτυξης για τη δημιουργία ρομπότ ικανών να αλληλεπιδρούν με επιτυχία με τον άνθρωπο. (p3)

Το 1988 ο όμιλος Stäubli αγοράζει την Unimation και τον ίδιο χρόνο κατασκευάζεται από το Daniel Hillis η μηχανή σύνδεσης (Connection Machine), ένας υπερυπολογιστής που χρησιμοποιούσε 64.000 επεξεργαστές ταυτόχρονα. (cm)

Προγράμματα που παίζαν σκάκι όπως το HiTech και το Deep Thought νίκησαν επαγγελματίες σκακιστές το 1989. Και τα δύο αναπτύχθηκαν από το Carnegie Mellon University, όπου η ανάπτυξη του Deep Thought άνοιξε το δρόμο για το Deep Blue.

(chess) Τον ίδιο χρόνο αποκαλύφθηκε από το MIT ένα εξάποδο (hexapodal) ρομπότ που ονομάζεται Genghis(Τζένγκις). Το Τζένγκις ήταν διάσημο γιατί κατασκευαζόταν γρήγορα και φτηνά λόγω των μεθόδων κατασκευής και χρησιμοποιούσε 4



Εικ.35 Genghis

μικροεπεξεργαστές, 22 αισθητήρες και 12

servo κινητήρες. (genghis) Επίσης, εκείνη την περίοδο, ο Rodney Brooks και η Anita M. Flynn δημοσίευσαν το σύγγραμμα με τίτλο: «Γρήγορη, Φθηνή, και Εκτός Ελέγχου: Μια εισβολή ρομπότ στο ηλιακό σύστημα» (Fast, Cheap, and Out of Control: A Robot Invasion of The Solar System). Το σύγγραμμα υποστήριζε τη δημιουργία μικρότερων και φθηνότερων ρομπότ σε μεγαλύτερους αριθμούς για την αύξηση του χρόνου παραγωγής και τη μείωση της δυσκολίας προώθησης των ρομπότ στο διάστημα. (RB.AF)

1.2.8 Από το 1991 έως το 2000

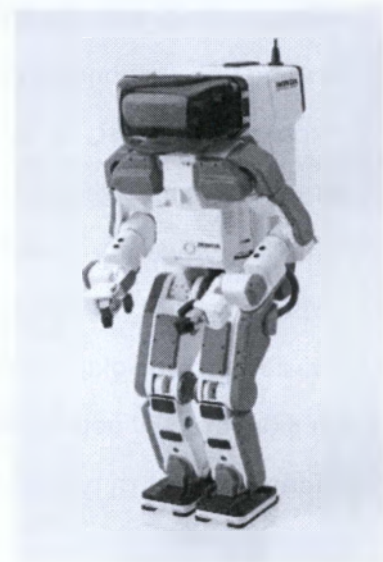
Ενώ ανταγωνιζόταν το 1993 σε έναν διαγωνισμό με χορηγό την NASA, το οκτάποδο ρομπότ Dante του Carnegie Mellon University αποτυγχάνει να συγκεντρώσει αέρια από το Mt. Erebus λόγω ενός σπασμένου καλωδίου οπτικών ινών. Το Dante είχε σχεδιαστεί για να σκαρφαλώνει πλαγιές και να συλλέγει αέρια κοντά στην επιφάνεια του μάγματος, ωστόσο η αποτυχία στο καλώδιο δεν επέτρεψε στο ρομπότ να εισέλθει

στο ενεργό ηφαίστειο. Ένα χρόνο αργότερα (1994) το Dante II εισήχθη στο Mt. Spurr και με επιτυχία μάζεψε δείγματα αέριων μέσα από το ηφαίστειο. (dande-dante2) Τον ίδιο χρόνο εφευρέθηκε από τον Δρ John Adler το CyberKnife, ένα ρομπότ που εκτελεί στερεοτακτική ακτινοχειρουργική, το οποίο πρόσφερε μια εναλλακτική θεραπεία των όγκων με ακρίβεια που ήταν συγκρίσιμη με τη χειρουργική επέμβαση που γίνεται από ανθρώπινους γιατρούς. (cknife)



Εικ.37 Ρομπότ Dante II

Το βιομηχανικό ρομπότ Robotuna κατασκευάστηκε από τον διδακτορικό φοιτητή David Barrett στο Massachusetts Institute of Technology το 1996, για να μελετήσει πώς τα ψάρια κολυμπούν στο νερό. Το Robotuna είχε σχεδιαστεί για να κολυμπάει και να μοιάζει με έναν ερυθρού χρώματος τόνο. (robotuna) Το ανθρωποειδές ρομπότ P2 της Honda παρουσιάστηκε για πρώτη φορά το 1996, αντιπροσωπεύοντας το "Prototype Model 2". Το P2 με πάνω από 6 πόδια ψηλό, ήταν μικρότερο από τους προκατόχους του και φάνηκε να είναι πιο ανθρώπινο στις κινήσεις του. Αυτό το ρομπότ ήταν αναπόσπαστο μέρος της ανθρωποειδούς ανάπτυξης (του έργου) της Honda. (P2)



Εικ.36 P2, ρομπότ της Honda

Αναμένοντας να λειτουργήσει μόνο για επτά ημέρες, το Sojourner rover τελικά διακόπτει την λειτουργία του μετά από 83 ημέρες λειτουργίας το 1997.

Αυτό το μικρό ρομπότ (ζυγίζοντας μόνο 10.45 kgs)

εκτελούσε ημι-αυτόνομες λειτουργίες στην επιφάνεια του Άρη, στο πλαίσιο της αποστολής Mars Pathfinder. Εξοπλισμένο με ένα πρόγραμμα αποφυγή εμποδίων το Sojourner ήταν ικανό να σχεδιάζει και να πλοηγείτε διαδρομές για να μελετήσει την επιφάνεια του πλανήτη. Η ικανότητα του Sojourner να πλοηγείτε με λίγα μόνο στοιχεία σχετικά με το περιβάλλον του και την γύρω περιοχή, επέτρεψε στο ρομπότ να αντιδρά σε μη προγραμματισμένα συμβάντα και αντικείμενα. (Srover)

Επίσης το 1997, το πρόγραμμα Deep Blue της IBM νίκησε τον τότε τρέχων Παγκόσμιο Πρωταθλητή Σκακιού Garry Kasparov, παίζοντας στο «Grandmaster» επίπεδο. Ο σούπερ υπολογιστής ήταν μια εξειδικευμένη έκδοση του πλαισίου (framework) που παράγεται από την IBM, και ήταν ικανό να επεξεργάζεται διπλάσιο αριθμό κινήσεων ανά δευτερόλεπτο, από το πρώτο αγώνα (τον οποίο ο Deep Blue είχε χάσει) που είχε σύμφωνα με πληροφορίες 200.000.000 κινήσεις ανά δευτερόλεπτο. Η εκδήλωση μεταδόθηκε ζωντανά μέσω του Διαδικτύου και έλαβε πάνω από 74 εκατομμύρια επισκέψεις. (Dblue)

Το ανθρωποειδές ρομπότ P3 αποκαλύφθηκε από την Honda το Σεπτέμβριο του 1997 ως μέρος του συνεχιζόμενου ανθρωποειδούς σχεδίου της εταιρείας. (P3)

Το 1999, η Sony παρουσιάζει το Aibo, ένα ρομποτικό σκύλο ικανό να αλληλεπιδρά με τους ανθρώπους, και τα πρώτα μοντέλα που κυκλοφόρησαν στην Ιαπωνία εξαντλήθηκαν μέσα σε 20 λεπτά. (aibo)

Η Honda αποκάλυψε το πιο προηγμένο αποτέλεσμα του ανθρωποειδούς έργου τους το 2000, με το ρομπότ εν' ονόματι ASIMO. Ο ASIMO είναι ικανός να τρέχει, να περιπατάει, να επικοινωνεί με τους ανθρώπους, να αναγνωρίζει πρόσωπα και περιβάλλον, να αναγνωρίζει τη φωνή και τη στάση του σώματος και να αλληλεπιδρά με το περιβάλλον του. (Asimo)

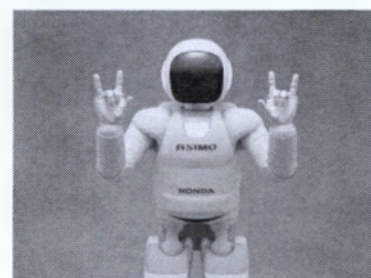
Τον Οκτώβριο του 2000, τα Ηνωμένα Έθνη εκτιμούν ότι υπήρχαν 742.500 βιομηχανικά ρομπότ στον κόσμο, με περισσότερα από τα μισά ρομπότ να χρησιμοποιούνται στην Ιαπωνία.

1.2.9 Από το 2001 έως το 2009

Τον Απρίλιο του 2001, το Canadarm2 τέθηκε σε τροχιά και συνδέθηκε στο Διεθνή Διαστημικό Σταθμό. Το Canadarm2 είναι μια μεγαλύτερη, πιο ικανή έκδοση του



Εικ.38 Aibo, ρομποτικός σκύλος

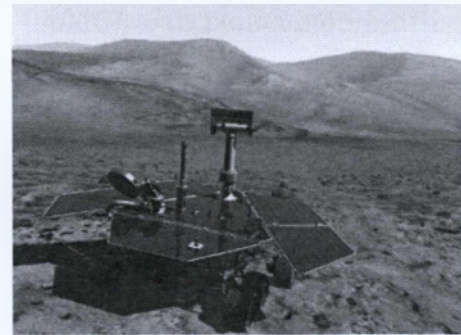


Εικ.39 ASIMO, Honda's ρομπότ

βραχίονα που χρησιμοποιείται από το Διαστημικό Λεωφορείο και χαιρετίστηκε ως «έξυπνο». (Canadarm2) Επίσης, τον ίδιο μήνα, το μη επανδρωμένο εναέριο όχημα Global Hawk έκανε την πρώτη αυτόνομη non-stop (χωρίς στάση) πτήση πάνω από τον Ειρηνικό Ωκεανό, από την Edwards Air Force Base στην Καλιφόρνια μέχρι την RAAF Base Edinburgh στη Νότια Αυστραλία, και η πτήση κράτησε 22 ώρες. (G.H.)

Η δημοφιλής Roomba, μια ρομποτική ηλεκτρική σκούπα, κυκλοφόρησε για πρώτη φορά το 2002 από την εταιρεία iRobot. (Roomba)

Στις 3 και 24 Ιανουαρίου του 2004, τα ρομποτικά ρόβερ (rover) Spirit και Opportunity προσγειώνονται στην επιφάνεια του Άρη, όπου τα δύο ρομπότ θα οδηγήσουν πολλές φορές την απόσταση που αρχικά αναμενόταν, με το Opportunity είναι ακόμα σε λειτουργία. (S.O.Rover)



Εικ.40 Opportunity Rover στον Άρη

Το 2004, το Cornell University αποκαλύπτει ένα ρομπότ ικανό για αυτο-αναπαραγωγή, μια σειρά από κύβους ικανοί να συνδέονται και να αποσυνδέονται, το πρώτο ρομπότ με την ικανότητα να κατασκευάζει αντίγραφα του εαυτού του. (cornell) Τα αυτο-οδηγούμενα αυτοκίνητα (Self-driving cars) είχαν κάνει την εμφάνισή τους από τα μέσα της πρώτης δεκαετίας του 21ου αιώνα, αλλά υπήρχε περιθώριο για βελτίωση. Και οι 15 ομάδες που ανταγωνίζονταν στο 2004 DARPA Grand Challenge απέτυχαν να ολοκληρώσουν την διαδρομή, με κανένα από τα ρομπότ να πιλοτάρετε με επιτυχία πάνω από το πέντε τις εκατό (5%) των 241.40km ανώμαλου δρόμου, αφήνοντας το βραβείο των 1,000,000 δολαρίων αδιεκδίκητο. (Darpa2004)

Στο DARPA Grand Challenge του 2005, πέντε ομάδες ολοκλήρωσαν την ανώμαλη διαδρομή, με το Stanley του Πανεπιστημίου του Στάνφορντ να κατακτά την πρώτη θέση και το έπαθλο των 2 εκατομμυρίων δολαρίων. (Darpa2005) Επίσης, το 2005 η Honda αποκαλύπτει μια νέα έκδοση του ρομπότ ASIMO, ενημερωμένο με νέες συμπεριφορές και δυνατότητες. (N.asimo)

Το 2006, το Cornell University αποκάλυψε το ρομπότ του "Starfish, ένα τετράποδο ρομπότ ικανό να αυτό-μοντελοποιείται και να μαθαίνει να περπατάει, αφού είχε υποστεί ζημιά. (Starfish)

Τον Σεπτέμβριο του 2007, η Google ανακοίνωσε το Lunar X Prize, το οποίο προσφέρει 30 εκατομμύρια δολάρια στην πρώτη ιδιωτική εταιρεία που θα προσγειώσει ένα rover στο φεγγάρι και θα στείλει τις εικόνες του πίσω στη γη. (L.X.P) Επίσης, το 2007 η εταιρία TOMY

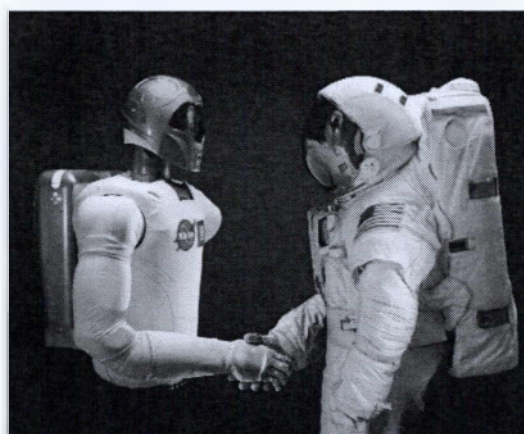


Εικ.41 i-sobot , το ρομπότ της TONY

προώθησε το ρομπότ ψυχαγωγίας i-sobot, το οποίο είναι ένα ανθρωποειδές δίποδο ρομπότ που μπορεί να περπατήσει σαν ανθρώπινο όν και να εκτελεί κλωτσιές , γροθιές , μερικά διασκεδαστικά κόλπα και ειδικές ενέργειες υπό την επιλογή "Special Action Mode". (isobot)

1.2.10 Από το 2010 μέχρι σήμερα

Το Robonaut 2, η τελευταία γενιά των βοηθών για αστροναύτες, ξεκίνησε στον διαστημικό σταθμό επί του Διαστημικού Λεωφορείου Discovery στην STS-133 αποστολή, το 2011. Είναι το πρώτο ανθρωποειδές ρομπότ στο διάστημα, και παρόλο που πρωταρχικό καθήκον του προς το παρόν είναι η διδασκαλία των μηχανικών για το πόσο επιδέξια συμπεριφέρονται τα ρομπότ στο διάστημα, η ελπίδα είναι ότι μέσω αναβαθμίσεων και εξελίξεων θα μπορούσε μια μέρα να επιχειρήσει να βγει έξω από το σταθμό για να βοηθήσει τους spacewalkers να κάνουν επισκευές ή προσθήκες στο σταθμό ή να εκτελέσει επιστημονικές εργασίες. (Robonaut2)



Εικ.42 Robonaut 2 , Nasa

Το τηλεκατευθυνόμενο αυτοκίνητο της Google έγινε διάσημο τη δεκαετία του 2010, καθώς ρομποτικά οχήματα της Google (με τους οδηγούς πίσω από τους τροχούς τους, σε περίπτωση έκτακτης ανάγκης) οδήγησαν σε πολλά μέρη παίρνοντας φωτογραφίες τα τοπία και χαρτογραφώντας πολιτείες και χωριά. (googlecars)

Εμπορικά και βιομηχανικά ρομπότ χρησιμοποιούνται ευρέως σήμερα σε θέσεις εργασίας με χαμηλότερο κόστος, και με μεγαλύτερη ακρίβεια και αξιοπιστία από τους ανθρώπους. Επίσης χρησιμοποιούνται σε εργασίες όπου το περιβάλλον είναι πάρα πολύ βρώμικο ή επικίνδυνο για να είναι κατάλληλο για τους ανθρώπους. Ρομπότ χρησιμοποιούνται ευρέως στην κατασκευή, συναρμολόγηση και συσκευασία, τη μεταφορά, τη γη και την εξερεύνηση του διαστήματος, τη χειρουργική επέμβαση, τα όπλα, την εργαστηριακή έρευνα, και στη μαζική παραγωγή καταναλωτικών και βιομηχανικών προϊόντων.

1.3 Ετυμολογία

Η λέξη ρομπότ εισήχθη στο κοινό από τον Τσέχο μεσοπολεμικό συγγραφέα Karel Čapek στο έργο του R.U.R. (Rossum's Universal Robots), που δημοσιεύθηκε το 1920. (Zunt) Το θεατρικό έργο ξεκινάει σε ένα εργοστάσιο που κατασκευάζει τεχνητούς ανθρώπους που ονομάζονται ρομπότ, αν και είναι πιο κοντά στις



Εικ.43 R.U.R. (1922)

σύγχρονες ιδέες των androids, πλάσματα που μπορούν να εκληφθούν για ανθρώπους. Μπορούν σκέφτονται για τον εαυτό τους ευκρινώς, αν και φαίνονται πρόθυμα να εξυπηρετούν. Το ζήτημα είναι κατά πόσο τα ρομπότ αποτελούν αντικείμενο εκμετάλλευσης και οι συνέπειες της μεταχείρισής τους.



Εικ.45 Karel Čapek

Ο Karel Čapek δεν έπλασε τη λέξη ο ίδιος. Έγραψε μια σύντομη επιστολή σε σχέση με την ετυμολογία στο αγγλικό λεξικό της Οξφόρδης το οποίο ονόμασε τον αδελφό του, τον ζωγράφο και συγγραφέα Josef Čapek, ως πραγματικό δημιουργό της. (Zunt)

Σε ένα άρθρο στην Εφημερίδα της Τσεχίας Lidové Noviny το 1933, εξήγησε ότι αρχικά ήθελε να καλέσει τα πλάσματα laboři ("εργαζόμενοι", από τη Λατινική λέξη labor). Ωστόσο, ο ίδιος δεν του άρεσε η λέξη, και ζήτησε συμβουλές από τον αδελφό του Josef, ο οποίος πρότεινε

"roboti". Η προέλευση της λέξης είναι η Παλαιά Εκκλησιαστική Σλαβική (αρχαία Βουλγάρικα) rabota "δουλεία" ("εργασία" στα σύγχρονα Βουλγάρικα και Ρώσικα), η οποία με τη σειρά της προέρχεται από την ινδοευρωπαϊκή ρίζα «orbh-». Η λέξη robota σημαίνει κυριολεκτικά "αγγαρεία", "δουλοπάροικος", και μεταφορικά "ελεεινή



Εικ.44 WPA αφίσα
κουκλοθέατρου για το R.U.R.

εργασία" ή "σκληρή δουλειά" στην Τσεχική γλώσσα και (γενικότερα) "έργο", "εργασία" σε πολλές Σλαβικές γλώσσες (π.χ.: Βουλγάρικα, Ρώσικα, Σέρβικα, Σλοβάκικα, Πολωνικά, Μακεδονικά, Ουκρανικά, αρχαία Τσέχικα). Παραδοσιακά, η *robot* ήταν η περίοδος εργασίας ενός δουλοπάροικου (αγγαρεία) που έπρεπε να δώσει για τον Κύριό του, συνήθως 6 μήνες του έτους. Το Robot είναι συγγενής με τη γερμανική ρίζα Arbeit (εργασία). (Online Etymology Dictionary) (Bartleby)



Εικ.46 Isaac Asimov

Η λέξη "ρομποτική" ("robotics"), που χρησιμοποιείται για να περιγράψει αυτό το πεδίο της μελέτης, επινοήθηκε από τον συγγραφέα επιστημονικής φαντασίας Isaac Asimov. Ο Asimov δημιούργησε τους "Τρεις Νόμους της Ρομποτικής", το οποίο είναι ένα επαναλαμβανόμενο θέμα στα βιβλία του. Αυτά έχουν ήδη χρησιμοποιηθεί από πολλούς άλλους για να καθορίσουν νόμους που χρησιμοποιούνται στην πραγματικότητα και στη μυθιστοριογραφία. Παρουσιάστηκαν στο διήγημά του "Runaround" που εμπεριέχεται στη συλλογή "I, Robot" το 1942, οι νόμοι αναφέρουν τα εξής: (Jones)

1. Ένα ρομπότ δεν μπορεί να βλάψει ένα ανθρώπινο ον, ή μέσω της αδράνειας του να επιτρέψει να βλαφτεί ένα ανθρώπινο ον.
2. Ένα ρομπότ πρέπει να υπακούει τις διαταγές που του δίνει ένα ανθρώπινο ον, εκτός εάν αυτές οι διαταγές έρχονται σε αντίθεση με τον πρώτο νόμο.
3. Ένα ρομπότ οφείλει να προστατεύει την ύπαρξή του , εφόσον αυτό δεν έρχεται σε αντίθεση με το πρώτο ή δεύτερο νόμο.

The Three Laws of Robotics

1 - A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

2 - A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

3 - A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

*Handbook of Robotics,
56th Edition, 2058 A.D.*

1.4 Ορισμός

Η λέξη ρομπότ μπορεί να αναφέρεται τόσο σε υλικά ρομπότ όσο και σε εικονικά ρομπότ πράκτορες λογισμικού (virtual software agents), αλλά το τελευταίο καιρό συνήθως αναφέρεται ως bots. (ATIS) Δεν υπάρχει ομοφωνία σχετικά με ποιές από τις μηχανές πληρούν τις προϋποθέσεις ως ρομπότ, αλλά δεν υπάρχει γενική συμφωνία μεταξύ των εμπειρογνομώνων, και το επενδυτικό κοινό, ότι τα ρομπότ έχουν την τάση να κάνουν κάποια ή όλα από τα ακόλουθα: μετακινούνται, χειρίζονται ένα μηχανικό άκρο, να αισθάνονται και να χειραγωγούν το περιβάλλον τους και να επιδεικνύουν ευφυή συμπεριφορά - ειδικά συμπεριφορά που μιμείται ανθρώπους ή άλλα ζώα.

Δεν υπάρχει ένας ορισμός του ρομπότ που να ικανοποιεί τους πάντες και πολλοί άνθρωποι έχουν το δικό τους. (Answers) (2005) Για παράδειγμα ο Joseph Engelberger, πρωτοπόρος στη βιομηχανική ρομποτική, παρατήρησε κάποτε: «Δεν μπορώ να καθορίσω ένα ρομπότ, αλλά ξέρω ένα όταν βλέπω ένα». (Harris) Σύμφωνα με την εγκυκλοπαίδεια Britannica ένα ρομπότ είναι «κάθε μηχανήμα αυτόματης λειτουργίας που αντικαθιστά την ανθρώπινη προσπάθεια, αν και μπορεί να μην μοιάζει με ανθρώπινο όν στην εμφάνιση ή να εκτελεί λειτουργίες σε ένα ανθρωπόμορφο τρόπο». Η "Merriam-Webster" περιγράφει ένα ρομπότ ως «μια μηχανή που μοιάζει με ένα ανθρώπινο όν και εκτελεί διάφορες πολύπλοκες πράξεις (όπως το περπάτημα ή την ομιλία) ενός ανθρώπινου όντος», ή μια «συσκευή που εκτελεί αυτόματα περίπλοκες συχνά επαναλαμβανόμενες εργασίες» ή ενός «μηχανισμού καθοδηγούμενο με αυτόματες ρυθμίσεις». (Merriam-webster) Στην πράξη, "ρομπότ" συνήθως αναφέρεται σε ένα μηχανήμα το οποίο μπορεί να είναι ηλεκτρονικά προγραμματισμένο ώστε να πραγματοποιήσει μια ποικιλία από φυσικές εργασίες ή ενέργειες.

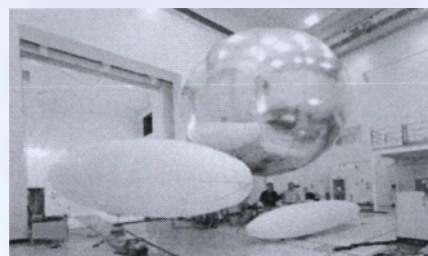
1.5 Κατάλογος των ρομπότ

Aerobot - ρομπότ ικανό για ανεξάρτητες πτήσεις σε άλλους πλανήτες.



Εικ.48 Geminoid F

Ανδρειδές (Android) - ανθρωποειδές ρομπότ.
Ρομπότ που μοιάζει με το σχήμα ή την μορφή ενός ανθρώπου.



Εικ.47 Έλεγχος φουσκώματος τριών πρωτότυπων αεροπλάνων της JPL

Αυτόματο - αρχικό αυτο-λειτουργικό ρομπότ που εκτελεί ακριβώς τις ίδιες ενέργειες ξανά και ξανά.



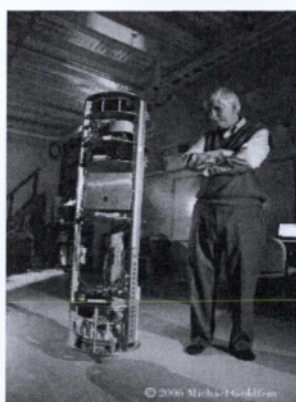
Εικ.50 Google Lexus FX450h, αυτόνομο όχημα

Αυτόνομο όχημα (Autonomous vehicle)
- όχημα εξοπλισμένο

με σύστημα αυτόματου πιλότου, το οποίο είναι ικανό να οδηγήσει από το ένα σημείο στο άλλο χωρίς τη συμβολή από τον χειριστή.



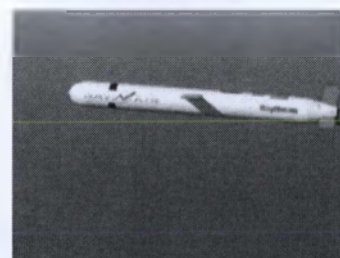
Εικ.49 The Invention of Hugo Cabret , Hugo (ταινία)



Εικ.51 Ballbot ισορροπεί στο CMU εργαστήριο, με τον Ralph Hollis να κοιτάζει

Ballbot - δυναμικά σταθερό κινητό ρομπότ σχεδιασμένο για να ισορροπήσει σε ένα ενιαίο σφαιρικό τροχό (δηλαδή, μια μπάλα).

Cruise πυραύλων - ρομπότ ελεγχόμενου κατευθυνόμενου βλήματος που φέρει ένα εκρηκτικό φορτίο.



Εικ.52 Tomahawk Block IV cruise πύραυλος, κατά της διάρκειας δοκιμών (US Navy)



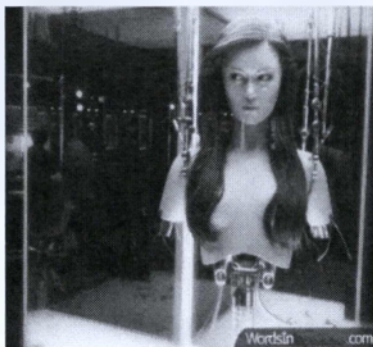
Εικ.53 Major Motoko Kusanagi από το Ghost in the Shell

Cyborg - επίσης γνωστό ως κυβερνητικός οργανισμός, ένα ον και με βιολογικά και με τεχνητά (π.χ. ηλεκτρονικό, μηχανικό ή ρομποτικό) τμήματα.

Explosive ordnance disposal robot (ρομπότ πυροτεχνουργί) - κινητό ρομπότ σχεδιασμένο για να αξιολογεί αν ένα αντικείμενο περιέχει εκρηκτικά. Μερικά φέρουν πυροκροτητές τους οποίους μπορούν να κατατίθενται στο αντικείμενο και να ενεργοποιούνται μετά την απόσυρση του ρομπότ.



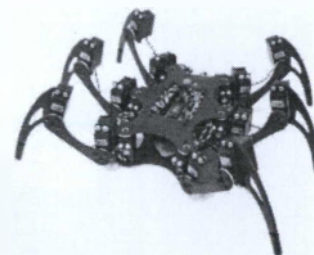
Εικ.54 QinetiQ, TALON®



Εικ.55 Sarah Connor chronicles (gynoid από τηλεοπτική σειρά)

Gynoid - ανθρωποειδές ρομπότ σχεδιασμένο να μοιάζει με ένα ανθρώπινο θηλυκό.

Hexapod (Walker) - Ένα εξάποδο ρομπότ που περιπατάει χρησιμοποιώντας ένα απλό είδος μετακίνησης που μοιάζει με έντομο.

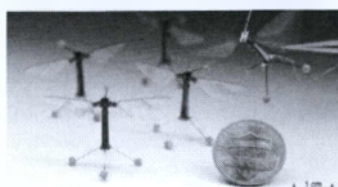


Εικ.56 Hexapod Phoenix



Εικ.57 KUKA KR 1000

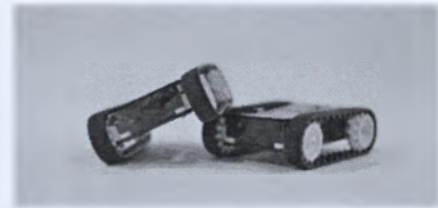
Βιομηχανικά ρομπότ - Επαναπρογραμματιζόμενος, πολυλειτουργικός βραχίονας σχεδιασμένος για τη μεταφορά υλικών, κατασκευαστικών τμημάτων, εργαλείων ή εξειδικευμένων συσκευών μέσω μεταβλητών προγραμματισμένων κινήσεων για την εκτέλεση μιας ποικιλίας καθηκόντων.



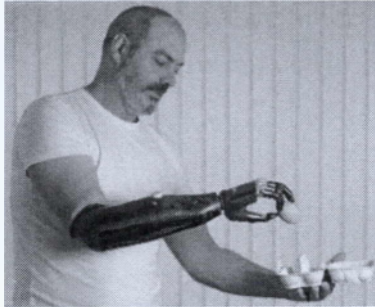
Εικ.58 Flight of the RoboBee

Ρομπότ έντομα (robot έντομα Insect robot) - μικρά ρομπότ σχεδιασμένα να μιμούνται συμπεριφορές εντόμων και όχι σύνθετες ανθρώπινες συμπεριφορές.

Mobile ρομπότ - Αυτοκινούμενα και αυτόνομα ρομπότ που μπορούν να κινούνται πάνω από μια μηχανικά αβίαστη διαδρομή.



Εικ.59 RK-1, A Wifi Arduino Mobile Robot



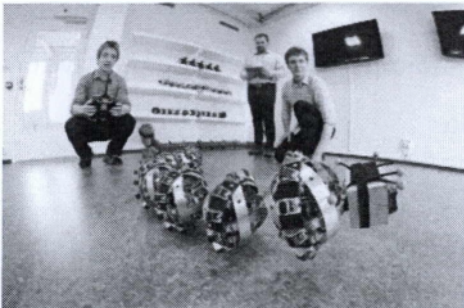
Εικ.61 Bebionic3 Myoelectric

Ρομπότ προσθετικού μέλους (Prosthetic robot) - προγραμματιζόμενος βραχίονας ή συσκευή που αντικαθιστά ένα χαμένο ανθρώπινο μέλος.

Service robot - μηχανήματα που επεκτείνουν τις ικανότητες του ανθρώπου.



Εικ.60 REEM-H2, PAL Robotics

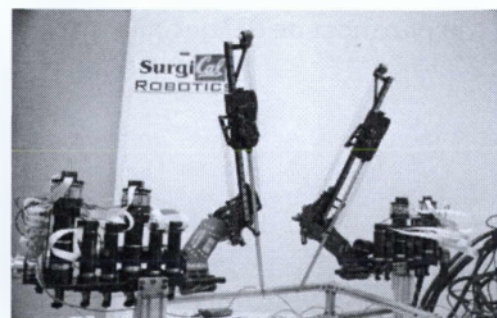


Εικ.62 SINTEF ερευνητές, Pal Liljebäck, Aksel Transeth και Knut Robert Fossum από το NTNU's CIRiS ασχολούνται με το Wheeka, το ρομποτικό φίδι.

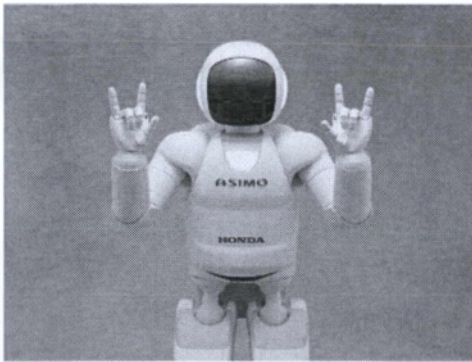
Snakebot - ρομπότ ή ρομποτικό εξάρτημα που μοιάζει με ένα πλοκάμι ή προβοσίδα του ελέφαντα, όπου πολλοί μικροί ενεργοποιητές χρησιμοποιούνται για να καταστεί δυνατή η συνεχής καμπυλόγραμμη κίνηση ενός μέρους του ρομπότ, με πολλούς βαθμούς ελευθερίας. Μια εφαρμογή σπανιότερη είναι το snakebot, όπου το σύνολο του ρομπότ είναι κινητό και μοιάζει σαν

φίδι, έτσι ώστε να αποκτήσουν πρόσβαση σε στενούς χώρους.

Surgical robot (Χειρουργικά ρομπότ) - εξ-αποστάσεως χειρισμού βραχίονες που χρησιμοποιούνται για τη χειρουργική επέμβαση κλειδαρότρυπας.



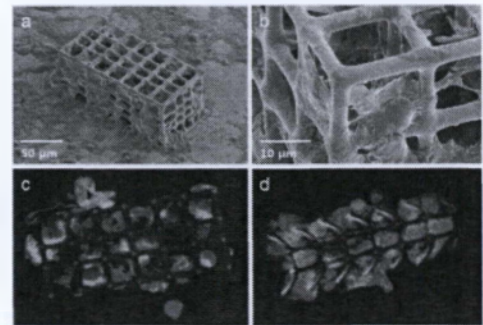
Εικ.63 RAVEN II, χειρουργικό ρομποτικό σύστημα



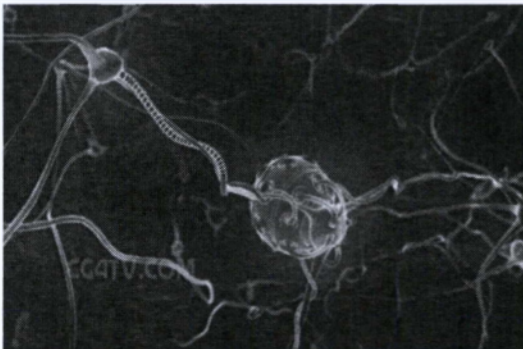
Εικ.64 Asimo, Honda's Robot

Walking robot - ρομπότ ικανό μετακίνησης με τα πόδια. Λόγω των δυσκολιών της ισορροπίας, δίποδα ρομπότ που να περιπατάνε μέχρι σήμερα είναι σπάνια και τα περισσότερα Walking robot έχουν χρησιμοποιήσει πολύκλωνους βηματισμούς που μοιάζουν με έντομα.

Microbot - μικροσκοπικά ρομπότ σχεδιασμένα για να εισχωρούν στο ανθρώπινο σώμα και να θεραπεύουν ασθένειες.



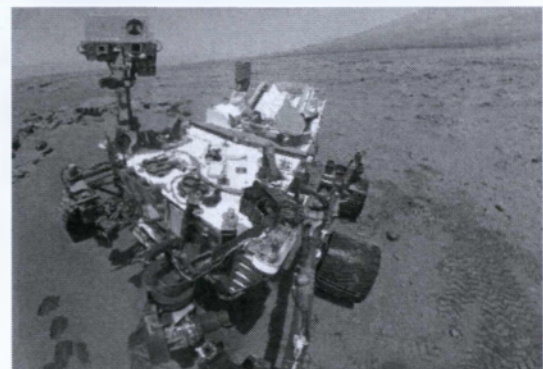
Εικ.65 Magnetic Microrobots



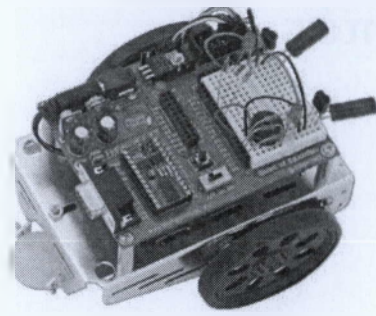
Εικ.66 Nanobots αντικαταστούν νευρώνες (νευρικά κύτταρα), απεικόνιση νανοτεχνολογίας

Nanobot - ίδια με τα microbot αλλά είναι λίγο μικρότερα.

Rover (εξερεύνηση του διαστήματος) - ένα ρομπότ με ρόδες σχεδιασμένο να περπατήσει σε άλλους πλανήτες.



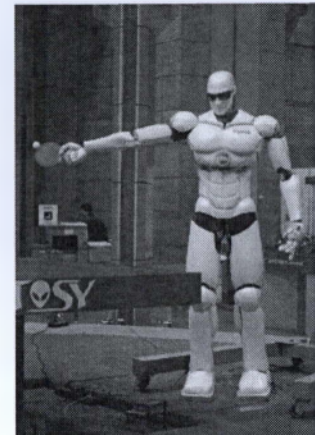
Εικ.67 Αυτοπροσωπογραφία του Curiosity, στο Gale Crater στην επιφάνεια του Άρη (31 Οκτωβρίου, 2012)



Εικ.68 Boe-Bot

Αυτόνομο ρομπότ (Autonomous robot) -ρομπότ που δεν ελέγχονται από τον άνθρωπο.

Ανθρωποειδές ρομπότ (Humanoid robots) - ρομπότ που το σχήμα του σώματος του έχει κατασκευαστεί για να μοιάζει και να λειτουργεί παρόμοια του ανθρώπινου σώματος. (technology) (Labor)



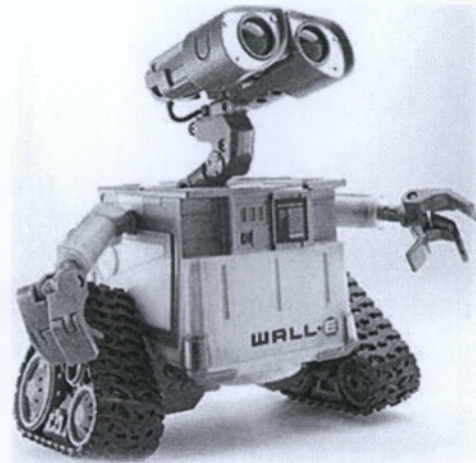
Εικ.69 ΤΟΡΠΙΟ – το ρομπότ της ΤΟΣΥ που παίζει Ping-Pong

Επί του παρόντος, υπάρχουν δύο κύριοι τύποι ρομπότ με βάση τη χρήση τους: η γενικής χρήσης αυτόνομα ρομπότ και τα ειδικά ρομπότ.

Τα ρομπότ μπορούν να ταξινομηθούν με την εξειδίκευση του σκοπού τους. Ένα ρομπότ θα μπορούσε να σχεδιαστεί για να εκτελεί μια συγκεκριμένη εργασία εξαιρετικά καλά, ή μια σειρά από καθήκοντα λιγότερο καλά. Φυσικά, όλα τα ρομπότ από τη φύση τους μπορούν να προγραμματιστούν εκ νέου να συμπεριφέρονται διαφορετικά, αλλά μερικά περιορίζονται από τη φυσική τους μορφή. Για παράδειγμα, ένας εργοστασιακός ρομποτικός βραχίονας μπορεί να εκτελέσει εργασίες, όπως η κοπή, συγκόλληση, κόλληση, ή σαν μια εκθεσιακή βόλτα στο χώρο, ενώ ένα pick-and-place ρομπότ μπορεί μόνο να συμπληρώνει πλακέτες τυπωμένων κυκλωμάτων.

1.6 Αυτόνομα ρομπότ

Τα Αυτόνομα ρομπότ είναι ρομπότ που μπορούν να εκτελέσουν εργασίες που επιθυμούμε σε αδόμητα περιβάλλοντα χωρίς συνεχή ανθρώπινη καθοδήγηση. Πολλά είδη ρομπότ έχουν κάποιο βαθμό αυτονομίας και μπορεί να είναι αυτόνομα με διαφορετικούς τρόπους. Ένας υψηλός βαθμός αυτονομίας είναι ιδιαίτερα επιθυμητός σε τομείς όπως η εξερεύνηση του διαστήματος, ο καθαρισμός του δαπέδου, το κούρεμα γρασιδιού, και την επεξεργασία των λυμάτων.



Εικ.70 WALL-E (2008,ταινία)

Μερικά σύγχρονα εργοστασιακά ρομπότ είναι «αυτόνομα» μέσα στα αυστηρά όρια του άμεσου (κατευθύνον) περιβάλλον τους. Μπορεί να μην υπάρχει κάθε (μεγάλος) βαθμός ελευθερίας στο περιβάλλον τους, αλλά ο χώρος εργασίας των εργοστασιακών ρομπότ είναι δύσκολος και μπορεί να περιέχουν συχνά χαοτικές και απρόβλεπτες μεταβλητές. Ο ακριβής προσανατολισμός και η θέση του επόμενου αντικείμενου της δουλειάς και (στα πιο προηγμένα εργοστάσια) ακόμη και ο τύπος του αντικείμενου και η απαιτούμενη εργασία πρέπει να προσδιορίζεται. Αυτό μπορεί να είναι απρόβλεπτο (τουλάχιστον από την πλευρά του ρομπότ).

Ένας σημαντικός τομέας της ρομποτικής έρευνας είναι να μπορέσει το ρομπότ να αντιμετωπίσει το περιβάλλον του, είτε αυτό είναι στη γη, υποβρύχια, στον αέρα, υπόγεια, ή στο διάστημα.

Ένα πλήρως αυτόνομο ρομπότ έχει την ικανότητα να :

1. Αποκτά πληροφορίες σχετικά με το περιβάλλον (Κανόνας # 1)
2. Να λειτουργεί για μεγάλο χρονικό διάστημα χωρίς την ανθρώπινη παρέμβαση (Κανόνας # 2)

3. Να μετακινεί είτε το σύνολο ή μέρος του εαυτού του καθ' όλην τη διάρκεια του περιβάλλον λειτουργίας του, χωρίς την ανθρώπινη παρέμβαση (Κανόνας # 3)
4. Να αποφεύγει καταστάσεις που είναι επιβλαβείς για τους ανθρώπους, την ιδιοκτησία, ή του εαυτού του, εκτός αν αυτά αποτελούν μέρος των προδιαγραφών σχεδιασμού του (Κανόνας # 4)

Ένα αυτόνομο ρομπότ μπορεί επίσης να μάθει ή να αποκτήσει νέες δυνατότητες όπως η προσαρμογή των στρατηγικών για την εκπλήρωση της αποστολής του ή την προσαρμογή του στα μεταβαλλόμενα περιβάλλοντα.

Τα αυτόνομα ρομπότ εξακολουθούν να απαιτούν τακτική συντήρηση, όπως και πολλά άλλα μηχανήματα.

1.6.1 Αυτο-συντήρηση

Η πρώτη προϋπόθεση για την πλήρη σωματική αυτονομία είναι η ικανότητα ενός ρομπότ να φροντίζει τον εαυτό του. Σήμερα στην αγορά, πολλά από τα ρομπότ που τροφοδοτούνται από μπαταρίες μπορούν να βρουν και να συνδεθούν σε έναν σταθμό φόρτισης, και μερικά παιχνίδια όπως το Aibo της Sony είναι ικανά για self-docking (Αυτο-σύνδεσης) για να φορτίσουν τις μπαταρίες τους.

Η αυτοσυντήρηση βασίζεται στην "ιδιοδεκτικότητα" ("proprioception"), ή στην αίσθηση της εσωτερικής κατάστασης του ατόμου. Στο παράδειγμα της φόρτισης μπαταρίας, το ρομπότ μπορεί να πει ιδιοδεκτικά (proprioceptively) ότι οι μπαταρίες του είναι χαμηλά και επιδιώκει στη συνέχεια το φορτιστή. Άλλος ένας κοινός ιδιοδεκτικού χαρακτήρα αισθητήρας είναι για την παρακολούθηση της θερμότητας. Θα απαιτείται αυξημένη ιδιοδεκτικότητα για τα ρομπότ που θα λειτουργούν αυτόνομα κοντά σε ανθρώπους και σε αντίξοες συνθήκες.

Κοινοί ιδιοδεκτικού χαρακτήρα αισθητήρες :

- Θερμικός (Thermal)
- Επίδρασης Hall (Hall Effect)

- Οπτικός (Optical)
- Επαφής (Contact)

1.6.2 Αίσθηση του περιβάλλοντος

Exteroception είναι το να αισθάνεσαι πράγματα για το περιβάλλον. Τα αυτόνομα ρομπότ πρέπει να έχουν μια σειρά από περιβαλλοντικούς αισθητήρες για να εκπληρώσουν την αποστολή τους και να αποτρέψουν οποιοδήποτε πρόβλημα.

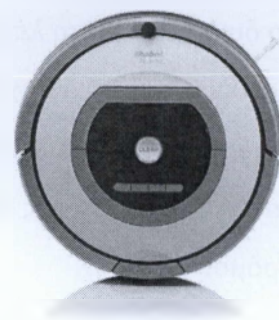
Οι πιο κοινοί exteroceptive αισθητήρες είναι :

- Ηλεκτρομαγνητικό φάσμα (Electromagnetic spectrum)
- Ήχος (Sound)
- Αφή (Touch)
- Χημικούς αισθητήρες (οσμή, ευοσμία) (Chemical sensors (smell, odor))
- Θερμοκρασία (Temperature)
- Η απόσταση μεταξύ πραγμάτων στο περιβάλλον (Range to things in the environment)
- Στάση (Κλίση) (Attitude (Inclination))

Μερικές ρομποτικές χορτοκοπτικές μηχανές προσαρμόζουν τον προγραμματισμό τους με βάση την ανίχνευση της ταχύτητας με την οποία το χορτάρι φυτρώνει, για να διατηρήσει ένα τέλειο γκαζόν . Ακόμα και οι ρομποτικές ηλεκτρικές σκούπες έχουν ανιχνευτές βρωμιάς ώστε να μετρούν πόση βρωμιά μαζεύεται σε κάθε περιοχή και να χρησιμοποιούν αυτές τις πληροφορίες για να παραμένουν σε μια περιοχή περισσότερο.

1.6.3 Επίδοση της εργασίας

Το επόμενο βήμα στην αυτόνομη συμπεριφορά είναι να εκτελούν πράγματι μια σωματική εργασία. Ένας νέος τομέας που δείχνει εμπορικά υποσχόμενος είναι τα οικιακά ρομπότ, με ένα κατακλυσμό από μικρά σκουπιδοφάγα ρομπότ ,ξεκινώντας με το iRobot και το Electrolux το 2002. Ενώ το επίπεδο της νοημοσύνης δεν είναι υψηλό σε αυτά τα συστήματα, περιηγούνται σε μεγάλες εκτάσεις και πλοηγούνται σε δύσκολες συνθήκες



Εικ.71 iRobot Roomba 760
(Ρομποτική ηλεκτρική σκούπα)

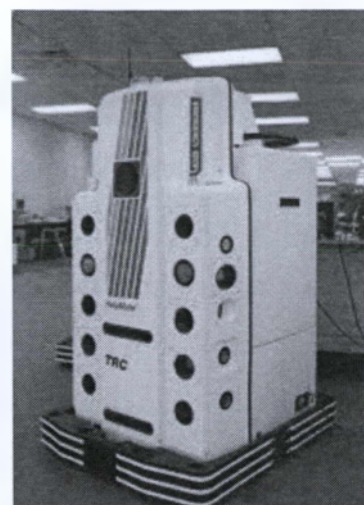
γύρω από τα σπίτια χρησιμοποιώντας επαφής (contact) και μη-επαφής (non-contact) αισθητήρες. Και τα δύο από αυτά τα ρομπότ χρησιμοποιούν ιδιόκτητους αλγορίθμους για την αύξηση της κάλυψης πάνω από μια απλή τυχαία αναπήδηση.

Το επόμενο επίπεδο της αυτόνομης εκτέλεσης των καθηκόντων απαιτεί ένα ρομπότ για την εκτέλεση των καθηκόντων υπό όρους. Για παράδειγμα, τα ρομπότ ασφάλειας μπορούν να προγραμματιστούν για να ανιχνεύουν τους εισβολείς και να ανταποκρίνονται με ένα συγκεκριμένο τρόπο ανάλογα με το σημείο που βρίσκεται ο εισβολέας .

1.6.4 Εσωτερική θέση αίσθησης και πλοήγησης

Ένα ρομπότ για να συσχετίζει λειτουργίες με ένα μέρος (localization), απαιτεί να γνωρίζει πού είναι και να είναι σε θέση να περιηγηθείτε point-to-point. Τέτοιου είδους πλοήγηση άρχισε με σύρμα-καθοδήγηση (wire-guidance) στη δεκαετία του 1970 και προόδευε στις αρχές του 2000 σε beacon-based τριγωνισμού(beacon-based triangulation). Τα τωρινά εμπορικά ρομπότ πλοηγούνται αυτόνομα με βάση τη διάγνωση φυσικών χαρακτηριστικών. Τα πρώτα εμπορικά ρομπότ που το κατάφερε αυτό ήταν το νοσοκομειακό ρομπότ HelpMate τις «Pyxis » και το CyberMotion ρομπότ φρουρός, τα οποία είχαν σχεδιαστεί από πρωτοπόρους στη ρομποτική τη δεκαετία του 1980. Αυτά τα ρομπότ χρησιμοποιούσαν αρχικά χειροποίητες κατόψεις CAD, ανίχνευση sonar και wall-following παραλλαγές για να περιηγηθούν σε κτίρια. Η επόμενη γενιά,

όπως το PatrolBot της εταιρίας MobileRobots και οι αυτόνομες αναπηρικές καρέκλας, (Mobilerobots), τα οποία εισήχθησαν το 2004, έχουν τη δυνατότητα να δημιουργούν το δικό τους χάρτη λέιζερ για ένα κτίριο και να περιηγηθούν σε ανοιχτούς χώρους, καθώς και τους διαδρόμους. Το σύστημα ελέγχου αλλάζει την πορεία του κατά τη διάρκεια της πλοήγησης αν κάτι μπλοκάρει το δρόμο.



Εικ.72 Pyxis's HelpMate

Στην αρχή, η αυτόνομη πλοήγηση βασίστηκε σε επίπεδες αισθητήρες, όπως τους εύρος-ανιχνευτές λέιζερ, που μπορούν να αισθανθούν μόνο σε ένα επίπεδο. Τα πιο προηγμένα συστήματα τώρα συναθροίζουν (συγχωνεύουν) πληροφορίες από διάφορους αισθητήρες τόσο για τον εντοπισμό (θέση) όσο και για την πλοήγηση. Συστήματα όπως το Motivity μπορούν να βασίζονται σε διαφορετικούς αισθητήρες σε διαφορετικές περιοχές, ανάλογα με ποιος παρέχει τα πιο αξιόπιστα δεδομένα εκείνη τη στιγμή, και μπορεί να χαρτογραφήσει ξανά ένα κτίριο αυτόνομα.

Αντί να ανέβουν σκάλες, το οποίο απαιτεί εξειδικευμένο hardware (υλικό), τα περισσότερα εσωτερικά ρομπότ πλοηγούνται σε περιοχές προσβάσιμες από άτομα με ειδικές ανάγκες, ελέγχοντας την λειτουργία ανελκυστήρων και ηλεκτρονικών εισόδων. (SpeciMinder) (Lokuge) Με τις εν λόγω ηλεκτρονικής ελέγχου πρόσβασης διασυνδέσεις (electronic access-control interfaces), τα ρομπότ μπορούν πλέον να πλοηγούνται(περιηγούνται) ελεύθερα σε εσωτερικούς χώρους. Να ανεβαίνουν σκάλες αυτόνομα και να ανοίγουν πόρτες με το χέρι είναι τα θέματα της έρευνας την τρέχουσα χρονική στιγμή. (Jeanne Dietsch)

Καθώς αυτές οι εσωτερικές τεχνικές συνεχίζουν να αναπτύσσονται, τα ρομπότ ηλεκτρικές σκούπες θα αποκτήσουν την ικανότητα να καθαρίζουν ένα συγκεκριμένο δωμάτιο καθορημένο από ένα χρήστη ή έναν ολόκληρο όροφο. Τα ρομπότ ασφαλείας θα είναι σε θέση να περικυκλώνουν(σε συνεργασία) συνεργατικά τους εισβολείς και να απομονώνουν τις εξόδους. Αυτές οι πρόοδοι φέρνουν και τις συνακόλουθες προσαπίσεις : (π.χ.) οι εσωτερικοί χάρτες των ρομπότ τυπικά επιτρέπουν να οριστούν "απαγορευμένες περιοχές " έτσι ώστε να προληφθούν ρομπότ από το να εισέρχονται αυτόνομα σε ορισμένες περιοχές-χώρες.

1.6.5 Εξωτερική αυτόνομη ανίχνευση θέσεως και πλοήγηση

Η εξωτερική αυτονομία είναι πιο εύκολο να επιτευχθεί στον αέρα, δεδομένου ότι τα εμπόδια είναι σπάνια. Τα βλήματα Cruise είναι αρκετά επικίνδυνα υψηλής αυτονομίας ρομπότ. Τα αεροσκάφη drone χωρίς πιλότο χρησιμοποιούνται ολοένα και περισσότερο για αναγνώριση. Μερικά από αυτά τα μη επανδρωμένα εναέρια οχήματα (UAV), είναι σε θέση να φέρουν σε πέρας όλη η αποστολή τους χωρίς καμία ανθρώπινη αλληλεπίδραση εκτός από την προσγείωση, όπου ένα άτομο παρεμβαίνει με τηλεχειρισμό. Αλλά και κάποια αεροσκάφη drone είναι σε θέση για μια ασφαλές αυτόματη προσγείωση.



Εικ.73 MDARS

Η εξωτερική αυτονομία είναι το πιο δύσκολο για τα οχήματα εδάφους, λόγω:

1. 3-διαστάσεων έδαφος
2. μεγάλες διαφορές στην πυκνότητα της επιφάνειας
3. απολύτου ανάγκης καιρικές συνθήκες και
4. αστάθεια του ανιχνευόμενου περιβάλλοντος.

Το έργο MDARS ορίστηκε στις ΗΠΑ και δημιούργησε ένα πρωτότυπο υπαίθριο ρομπότ επιτήρησης στη δεκαετία του 1990, το οποίο έχει είδη βγει στην παραγωγή από το 2004. Το General Dynamics MDARS ρομπότ μπορεί να πλοηγηθεί ημι-αυτόνομα και να εντοπίσει τους εισβολείς, χρησιμοποιώντας την αρχιτεκτονική λογισμικού MRHA που έχει σχεδιαστεί



Εικ.74 Rover Spirit

για όλα τα μη επανδρωμένα στρατιωτικά οχήματα. (Dynamics) Το ρομπότ Seekur ήταν το πρώτο εμπορικά διαθέσιμο ρομπότ που επιδείκνυε τις δυνατότητες του MDARS για γενική χρήση από αεροδρόμια, εγκαταστάσεις κοινής ωφελείας, εγκαταστάσεις διόρθωσης και εσωτερικής ασφάλειας. (Robots)

Τα ρομπότ rover στον Άρη MER-A και MER-B (γνωστά σήμερα ως rover Spirit και Opportunity rover) μπορούν να βρουν τη θέση του ήλιου και να πλοηγήσουν τις δικές τους διαδρομές σε προορισμούς πραγματοποιώντας τις εξής λειτουργίες:

- Χαρτογράφηση της επιφάνειας με 3-D όραση
- Υπολογισμός των ασφαλών και μη ασφαλών περιοχών στην επιφάνεια εντός του εν λόγω οπτικού πεδίου.
- Υπολογίζει τα βέλτιστα μονοπάτια σε όλη την ασφαλή περιοχή προς τον επιθυμητό προορισμό.
- Οδηγεί κατά μήκος της υπολογισμένης διαδρομής.
- Επαναλαμβάνει αυτό τον κύκλο μέχρις ότου είτε ο προορισμός να έχει επιτευχθεί, ή να μην υπάρχει καμία γνωστή διαδρομή για τον προορισμό.

Η προγραμματισμένη ESA Rover, Rover ExoMars, είναι ικανή να ορίζεται με βάση τον σχετικό εντοπισμό και τον απόλυτο εντοπισμό για να μεταφέρει αυτόνομα, ασφαλές και αποτελεσματικά τα trajectorys στους στόχους του , με την πραγματοποίηση των εξής λειτουργιών :

- Ανακατασκευάζει 3D μοντέλα του εδάφους που περιβάλλει το Rover χρησιμοποιώντας ένα ζεύγος από κάμερες στέρεο.
- Καθορίζει ασφαλές και μη ασφαλές περιοχές του εδάφους και την γενική «δυσκολία» του Rover να περιηγηθεί στο έδαφος.
- Υπολογίζει αποδοτικά μονοπάτια σε όλη την ασφαλή περιοχή προς τον επιθυμητό προορισμό.
- Οδηγεί το Rover κατά μήκος της προγραμματισμένης διαδρομής.

- Δημιουργεί ένα «χάρτη πλοήγησης» όλων των προηγούμενων δεδομένων πλοήγησης.

Το DARPA Grand Challenge και το DARPA Urban Challenge έχουν ενθαρρύνει την ανάπτυξη ακόμα πιο αυτόνομων δυνατοτήτων για τα οχήματα εδάφους, καθώς αυτός είναι ο αποδεικτικός στόχος για τα εναέρια ρομπότ από το 1990 ως μέρος του AUVSI International Aerial Robotics Competition (Διεθνή Εναέριο Ρομποτικό Διαγωνισμό).

1.6.6 Ανοιχτά προβλήματα στην αυτόνομη ρομποτική

Υπάρχουν πολλά ανοικτά προβλήματα στην αυτόνομη ρομποτική τα οποία είναι ειδικά στον τομέα αντί να είναι ένα μέρος της γενικής επιδίωξη της τεχνητής νοημοσύνης (AI). Σύμφωνα με βιβλίο «Αυτόνομα Ρομπότ» του George A. Bekey : Από τη βιολογική έμπνευση στην εκτέλεση και τον έλεγχο, τα προβλήματα περιλαμβάνουν πράγματα όπως διασφαλίζοντας το ρομπότ είναι σε θέση να λειτουργεί σωστά και να μην τρέχει αυτόνομα σε εμπόδια.

1.6.7 Ενεργειακή αυτονομία και αναζήτηση τροφής

Οι ερευνητές που ασχολούνται με τη δημιουργία της αληθής «τεχνητής ζωής» ανησυχούν όχι μόνο με τον ευφυή έλεγχο, αλλά πολύ περισσότερο με την ικανότητα του ρομπότ να βρίσκει τους δικούς του πόρους μέσω συλλογή τροφής (foraging ,ψάχνοντας για τροφή, το οποίο περιλαμβάνει τόσο την ενέργεια όσο και τα ανταλλακτικά).

Αυτό σχετίζεται με την αυτόνομη αναζήτηση τροφής, μια ανησυχία στο πλαίσιο των επιστημών της συμπεριφοράς της οικολογίας, της κοινωνικής ανθρωπολογίας και της οικολογίας ανθρώπινης συμπεριφοράς .Καθώς και της ρομποτικής, τεχνητής νοημοσύνης και τεχνητής ζωής.

1.7 Τομείς που Ασχολούνται

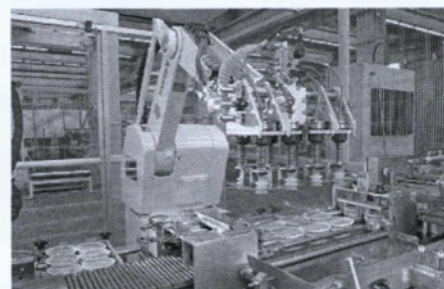
1.7.1 Εργοστασιακά Ρομπότ

Παραγωγή αυτοκινήτων

Κατά τη διάρκεια των τελευταίων τριών δεκαετιών, τα εργοστάσια αυτοκινήτων έχουν κυριευτεί από ρομπότ. Ένα τυπικό εργοστάσιο περιέχει εκατοντάδες βιομηχανικά ρομπότ που εργάζονται σε πλήρως αυτοματοποιημένες γραμμές παραγωγής, με ένα ρομπότ για κάθε δέκα εργάτες. Σε μια αυτοματοποιημένη γραμμή παραγωγής, ένα σασί αυτοκινήτου επί ενός μεταφορέα συγκολλείται, κολλάτε, βάφεται και τελικά συναρμολογείται σε μια ακολουθία σταθμών ρομπότ.

Συσκευασία

Επίσης τα βιομηχανικά ρομπότ χρησιμοποιούνται ευρέως για την παλετοποίηση και συσκευασία των μεταποιημένων αγαθών, για παράδειγμα, για την ταχεία λήψη εμπορευμάτων από το τέλος ενός ιμάντα μεταφοράς και την τοποθέτησή τους σε κουτιά, ή για τη φόρτωση και εκφόρτωση σε μηχανουργεία.



Εικ.75 Pick & Place ρομπότ πακετασίματος - Mariani, model RH360

Ηλεκτρονική

Μαζικής παραγωγής τυπωμένα κυκλώματα (PCB) παρασκευάζονται σχεδόν αποκλειστικά από pick-and-place ρομπότ, συνήθως με βραχίονες SCARA, τα οποία αφαιρούν μικροσκοπικά ηλεκτρονικά εξαρτήματα από ταινίες ή δίσκους, και τα τοποθετούν στα PCB με μεγάλη ακρίβεια. (Montag) Τέτοια ρομπότ μπορούν να τοποθετήσουν εκατοντάδες χιλιάδες εξαρτήματα ανά ώρα, ξεπερνώντας κατά πολύ έναν άνθρωπο στην ταχύτητα, την ακρίβεια και την αξιοπιστία. (pcb-experts)

Αυτοματοποιημένα καθοδηγούμενα οχήματα (AGVs)

Κινητά ρομπότ που ακολουθούν δείκτες ή καλώδια στο πάτωμα χρησιμοποιώντας την όραση (Itti) ή τα λέιζερ, τα οποία χρησιμοποιούνται για τη μεταφορά εμπορευμάτων γύρω από μεγάλες εγκαταστάσεις όπως αποθήκες, λιμάνια εμπορευματοκιβωτίων ή νοσοκομεία. (Automation)

Πρόωρα AGV Ρομπότ

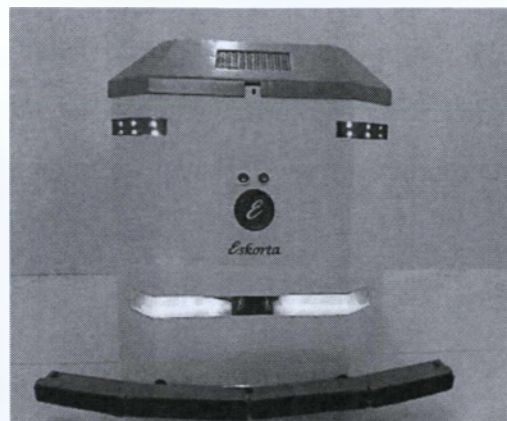
Περιορίζονται σε εργασίες που μπορούν να οριστούν με ακρίβεια και πρέπει να πραγματοποιηθούν με τον ίδιο τρόπο κάθε φορά. Απαιτείται πολύ λίγη ανατροφοδότηση και νοημοσύνη, και τα ρομπότ χρειάζονται μόνο τους πιο βασικούς αισθητήρες (exteroceptors). Οι περιορισμοί αυτών των AGVs είναι ότι τα μονοπάτια τους δεν μπορούν εύκολα να μεταβληθούν και δεν μπορούν να αλλάξουν τα μονοπάτια τους αν τα εμποδίζει κάποιο εμπόδιο. Εάν ένα AGV καταρρεύσει, μπορεί να σταματήσει το σύνολο της λειτουργίας.

Ενδιάμεσες AGV-Τεχνολογίες

Αναπτύχθηκαν για να επεκτείνουν τον τριγωνισμό (triangulation) από φάρους (ορόσημα διοπτρεύσεως) ή δίκτυα bar code (γραμμωτός κώδικας), για την σάρωση στο πάτωμα ή στο ταβάνι. Στα περισσότερα εργοστάσια, τα συστήματα τριγωνισμού τείνουν να απαιτούν μέτρια έως υψηλή συντήρηση, όπως η καθημερινή καθαριότητα όλων των φάρων ή των bar codes. Επίσης, εάν μια ψηλή παλέτα ή ένα μεγάλο όχημα μπλοκάρει τους φάρους ή το bar code αμαυρωθεί (παραμορφωθεί), τα AGVs μπορεί να χαθούν. Συχνά τέτοια AGVs σχεδιάζονται για να χρησιμοποιηθούν σε περιβάλλοντα χωρίς την ανθρώπινη παρουσία.

Ευφυή AGVs (i-AGVs)

Όπως τα SmartLoader, (webb) SpeciMinder, (SpeciMinder) ADAM, (Adam) Tug (Aethon) Eskorta, (Fennecfoxtech) και το MT 400 (Neobotix) με Motivity (htt) έχουν σχεδιαστεί για φιλικούς προς τους ανθρώπους χώρους εργασίας. Πλοηγούνται με την αναγνώριση φυσικών χαρακτηριστικών. Οι 3D σαρωτές ή άλλα μέσα της αίσθησης του περιβάλλοντος σε δύο ή τρεις διαστάσεις



Εικ.76 Fennec's Eskorta i-AGV ρομπότ

συμβάλουν στην εξάλειψη των συσσωρευτικών σφαλμάτων στους dead-reckoning υπολογισμούς της τωρινής θέσης του AGV. Μερικά AGVs μπορούν να δημιουργήσουν χάρτες του περιβάλλοντός τους με λέιζερ σάρωσης μαζί με ταυτόχρονο εντοπισμό και χαρτογράφηση (SLAM) και να χρησιμοποιούν τους χάρτες αυτούς για να περιηγηθούν σε πραγματικό χρόνο με άλλους αλγορίθμους σχεδιασμού διαδρομής και αποφυγής εμποδίων. Είναι σε θέση να λειτουργούν σε σύνθετα περιβάλλοντα και να εκτελούν μη επαναλαμβανόμενες και μη διαδοχικές εργασίες, όπως η μεταφορά photomasks σε ένα εργαστήριο ημιαγωγών, τα δείγματα στα νοσοκομεία και τα εμπορεύματα στις αποθήκες. Για τις δυναμικές περιοχές, όπως οι αποθήκες γεμάτες παλέτες, τα AGVs απαιτούν πρόσθετες στρατηγικές που χρησιμοποιούν τρισδιάστατους αισθητήρες, όπως ο χρόνος πτήσης (time-of-flight) ή οι στερεοσκοπικές κάμερες.

1.7.2 Βρώμικα, επικίνδυνα, αμβλύ ή απρόσιτα καθήκοντα

Υπάρχουν πολλές θέσεις εργασίας που οι άνθρωποι θα προτιμούσαν να αφήνουν σε ένα ρομπότ. Η εργασία μπορεί να είναι βαρετή, όπως ο οικιακός καθαρισμός, ή επικίνδυνη, όπως η εξερεύνηση μέσα σε ένα ηφαίστειο. (Robonlc System) Άλλες θέσεις εργασίας είναι φυσικά απροσπέλαστες, όπως η εξερεύνηση

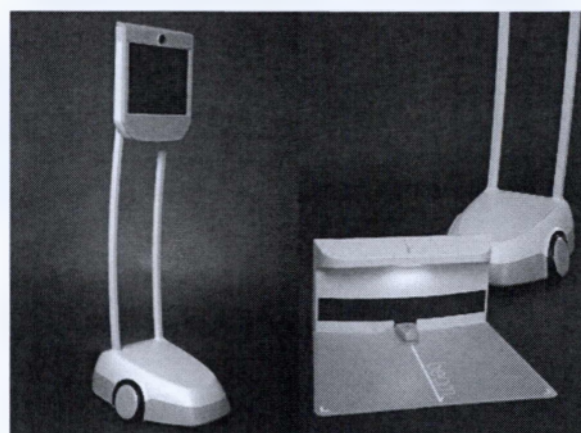
άλλου πλανήτη, (nasa.gov) ο καθαρισμός του εσωτερικού ενός μακρύ σωλήνα, ή την εκτέλεση λαπαροσκοπικής χειρουργικής επέμβασης. (Da vinci)

Διαστημικοί Εξερευνητές (Space probes)

Σχεδόν κάθε μη επανδρωμένο διαστημικό όχημα που διοργανώθηκε ποτέ ήταν ένα ρομπότ. Κάποια ξεκίνησαν στη δεκαετία του 1960 με πολύ περιορισμένες δυνατότητες, αλλά η ικανότητά τους να πετούν και να προσγειώνονται (στην περίπτωση του Luna 9) (Nasa.gov.luna9) αποτελεί ένδειξη της ιδιότητάς τους ως ρομπότ. Αυτό το κομμάτι περιλαμβάνει τους ανιχνευτές Voyager, τους ανιχνευτές Galileo, και άλλους πολλούς. (Nasa.gov)

Τηλερομπότ (Telerobots)

Τηλεχειριζόμενα ρομπότ (Teleoperated ρομπότ), ή τηλερομπότ είναι τηλεχειριζόμενες συσκευές εκ' αποστάσεως από έναν άνθρωπο χειριστή αντί να ακολουθούν μια προκαθορισμένη ακολουθία κινήσεων. Χρησιμοποιούνται όταν ένας άνθρωπος δεν μπορεί να είναι παρόν στην περιοχή για να εκτελέσει μια εργασία γιατί είναι επικίνδυνο, πολύ μακριά, ή απροσπέλαστο. Το ρομπότ μπορεί να είναι σε άλλο δωμάτιο ή σε άλλη χώρα, ή μπορεί να είναι σε μια πολύ διαφορετική κλίμακα για τον χειριστή. Για



Εικ.77 Το τηλερομπότ Beam Remote

παράδειγμα, ένα ρομπότ λαπαροσκοπικής χειρουργικής επιτρέπει στο χειρουργό να εργαστεί μέσα σε έναν ανθρώπινο ασθενή σε μια σχετικά μικρή κλίμακα σε σύγκριση με την ανοικτή χειρουργική επέμβαση, μειώνοντας σημαντικά τον χρόνο αποκατάστασης. (Da vinci) Μπορούν επίσης να χρησιμοποιηθούν για να αποφευχθεί η έκθεση των εργαζομένων σε επικίνδυνους και στενούς χώρους, όπως ο καθαρισμός αγωγών. Κατά την απενεργοποίηση μιας βόμβας, ο χειριστής στέλνει ένα μικρό τηλερομπότ για να την απενεργοποιήσει. Αρκετοί συγγραφείς έχουν χρησιμοποιήσει

μια συσκευή που ονομάζεται Longren να υπογράψουν τα βιβλία εξ αποστάσεως. (Fox-news) Τηλεχειριζόμενα αεροσκάφη ρομπότ, όπως το μη επανδρωμένο όχημα αέρος Predator, χρησιμοποιείται όλο και περισσότερο από το στρατό. Αυτά τα μη επανδρωμένα ρομπότ κηφήνες μπορούν να ανιχνεύουν την μορφολογία του εδάφους και να πυροβολούν στόχους. (military.factory), (Baglolle) Εκατοντάδες ρομπότ, όπως το PackBot της iRobot και το TALON της Foster-Miller χρησιμοποιούνταν στο Ιράκ και το Αφγανιστάν από το στρατό των ΗΠΑ για την εξουδετέρωση βομβών ή αυτοσχέδιων εκρηκτικών μηχανισμών (IED) σε μια δραστηριότητα που είναι γνωστή ως πυροτεχνουργία (EOD). (James Hannah, Associated Press)

Αυτόματα μηχανήματα συγκομιδής φρούτων

Χρησιμοποιούνται για να διαλέγουν τα φρούτα σε οπωρώνες με κόστος χαμηλότερο από εκείνο των ανθρώπινων εργατών.

Εγχώρια ρομπότ (Οικιακά ρομπότ , Domestic robots)

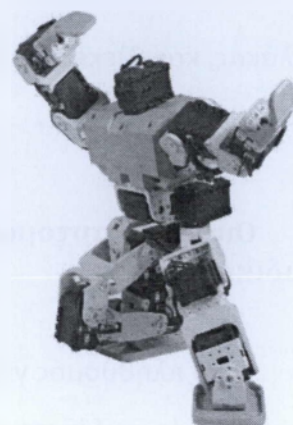
Τα εγχώρια ρομπότ είναι απλά ρομπότ αφιερωμένα σε ένα ενιαίο έργο , την εργασία μέσα στο σπίτι. Χρησιμοποιούνται σε απλές αλλά ανεπιθύμητες εργασίες, όπως το σκούπισμα ,το πλύσιμο των πατωμάτων και το κούρεμα του γκαζόν.

1.7.3 Σχολεία – Εκπαιδευτικά Ιδρύματα

Από τη δεκαετία του 1980, ρομπότ όπως οι «χελώνες» χρησιμοποιούνται στα σχολεία και προγραμματίζονται χρησιμοποιώντας τη γλώσσα Logo. (Roamer) Η Ρομποτική στα σχολεία του 21ου αιώνα έχει τρεις κύριες εφαρμογές, Ρομποτικά Kits, Εικονικούς καθηγητές και βοηθούς καθηγητών.

Ρομποτικά Κιτς (Robot kits)

Ρομποτικά κιτ όπως το Lego Mindstorms, το BIOLOID και το Olo της ROBOTIS ή τα BotBrain Εκπαιδευτικά ρομπότ μπορούν να βοηθήσουν τα παιδιά να μάθουν για τη φυσική, τα μαθηματικά, τον προγραμματισμό, και τα ηλεκτρονικά.



Εικ.78 Ανθρωποειδές ρομπότ, κατασκευασμένο χρησιμοποιώντας το Bioloid kit

Ρομποτικοί Διαγωνισμοί

Η Ρομποτική έχει επίσης εισαχθεί στη ζωή των μαθητών δημοτικού και γυμνασίου με την εταιρεία FIRST (For Inspiration and Recognition of Science and Technology ,για την έμπνευση και την αναγνώριση της Επιστήμης και Τεχνολογίας). Η οργάνωση είναι το θεμέλιο για τους διαγωνισμούς : FIRST Robotics Competition , FIRST LEGO League, Junior FIRST LEGO League και FIRST Tech Challenge.

Εικονικοί καθηγητές

Οι εικονικοί καθηγητές είναι κάποιο είδος «πράκτορα διεπαφής» (embodied agent) που βοηθά τα παιδιά να κάνουν τις εργασίες τους, για παράδειγμα, σε ομότιμη (peer to peer) βάση.

Βοηθοί Εκπαιδευτικών

Ρομπότ ως βοηθοί καθηγητών αφήστε τα παιδιά να είναι πιο διεκδικητική κατά τη διάρκεια της κατηγορίας και να πάρει περισσότερα κίνητρα. Η Νότια Κορέα είναι η πρώτη χώρα στην ανάπτυξη ενός προγράμματος για να έχουν ένα ρομπότ σε κάθε σχολείο.

1.7.4 Υγειονομική Περίθαλψη

Τα Ρομπότ στον τομέα της υγείας έχουν δύο βασικές λειτουργίες : 1)Εκείνα που βοηθούν ένα άτομο, όπως ένας πάσχων από μια ασθένεια όπως η σκλήρυνση κατά

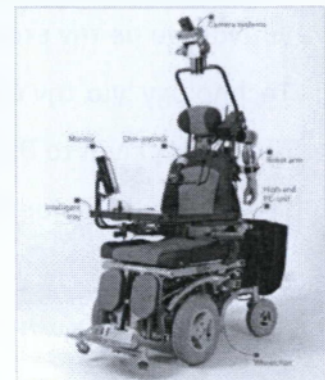
πλάκας, και 2) εκείνων που βοηθούν στα συνολικά συστήματα όπως τα φαρμακεία και τα νοσοκομεία.

Οικιακού αυτοματισμού ρομπότ για ηλικιωμένους και για άτομα με ειδικές ανάγκες

Ο πληθυσμός γηράσκει σε πολλές χώρες, ιδιαίτερα την Ιαπωνία, γεγονός που σημαίνει ότι αυξάνεται συνεχώς ο αριθμός των ηλικιωμένων που χρειάζονται φροντίδα, αλλά είναι σχετικά λιγότεροι οι νέοι για να τους φροντίζουν. (Pearce) Οι άνθρωποι είναι οι καλύτεροι φροντιστές, αλλά όταν δεν είναι διαθέσιμοι, τα ρομπότ αρχίζουν σταδιακά να καθιερώνονται.

Τα ρομπότ έχουν αναπτυχθεί κατά καιρούς από απλούς βασικούς ρομποτικούς βοηθούς, όπως το Handy 1, (Mike Topping) σε ημι-αυτόνομα ρομπότ, όπως το «FRIEND», που μπορούν να βοηθούν τους ηλικιωμένους και τα άτομα με ειδικές ανάγκες με τις κοινές εργασίες τους.

Το «FRIEND» είναι ένα ημι-αυτόνομο ρομπότ σχεδιασμένο για την υποστήριξη των ατόμων με αναπηρία και των ηλικιωμένων στις καθημερινές τους δραστηριότητες, όπως η προετοιμασία και το σερβίρισμα ενός γεύματος. Επίσης καταστεί δυνατό για τους ασθενείς που είναι παραπληγικοί, έχουν ασθένειες των μυών ή σοβαρή παράλυση (λόγω εγκεφαλικών επεισοδίων, κλπ.), να εκτελούν εργασίες χωρίς τη βοήθεια από άλλους ανθρώπους όπως τους θεραπευτές ή το νοσηλευτικό προσωπικό. (Qureshi)



Εικ.79 Ρομπότ παροχής φροντίδας, FRIEND

Φαρμακεία

Η εταιρεία Script Pro κατασκευάζει ένα ρομπότ σχεδιασμένο για να βοηθάει τα φαρμακεία να συμπληρώνουν τις συνταγές που αποτελούνται από του στόματος στερεές τροφές ή φάρμακα σε μορφή χαπιού. Ο φαρμακοποιός ή ο τεχνικός του φαρμακείου εισάγει τις πληροφορίες της συνταγής στο σύστημα πληροφοριών του. Το σύστημα, μετά τον προσδιορισμό εάν ή όχι το φάρμακο είναι στο ρομπότ, αποστέλλει την πληροφορία στο ρομπότ για την πλήρωση. Το ρομπότ διαθέτει 3 διαφορετικού

μεγέθους φιαλίδια για γέμισμα, το οποίο καθορίζεται από το μέγεθος του χαπιού. Ο τεχνικός του ρομπότ, ο χρήστης, ή ο φαρμακοποιός καθορίζει το μέγεθος που απαιτείται από το φιαλίδιο με βάση το δισκίο όταν το ρομπότ είναι εφοδιασμένο. Όταν γεμίσει το φιαλίδιο μεταφέρεται μέσω ενός ιμάντα σε μια βάση που το περιστρέφει και συνάπτει την ετικέτα του ασθενούς. Στη συνέχεια μέσω ενός άλλου μεταφορέα παραδίδεται το φιαλίδιο φαρμακευτικής αγωγής του ασθενούς σε μια υποδοχή με το όνομα του ασθενούς σε μια LED ένδειξη. Στη συνέχεια ο φαρμακοποιός ή ο τεχνικός ελέγχει το περιεχόμενο του φιαλιδίου για να εξασφαλιστεί ότι είναι το σωστό φάρμακο για το σωστό ασθενή και μετά σφραγίζει το φιαλίδιο και το στέλνει προς τα μπροστά για να ενταχθεί. Το ρομπότ είναι μια πολύ αποδοτική χρονικά συσκευή και το φαρμακείο εξαρτάται από αυτό για να συμπληρώνονται οι συνταγές. (Scriptpro)

Το ρομπότ RX της εταιρίας McKesson είναι ένα άλλο υγειονομικής περίθαλψης ρομποτικό προϊόν που βοηθά τα φαρμακεία να διανέμουν χιλιάδες φάρμακα καθημερινά με λίγα ή καθόλου λάθη. Το ρομπότ μπορεί να είναι δέκα μέτρα πλατύ και τριάντα μέτρα φαρδύ και μπορεί να διατηρεί εκατοντάδες διαφορετικά είδη φαρμάκων σε χιλιάδες δόσεις. Το φαρμακείο εξοικονομεί πολλούς πόρους, όπως τα μέλη του προσωπικού που ούτος η άλλος είναι δυσεύρετα σε μια βιομηχανία σπάνιων πόρων. Το ρομπότ χρησιμοποιεί έναν ηλεκτρομηχανικό κεφάλι σε



Εικ.80 Ρομπότ RX της McKesson

συνδυασμό με ένα πνευματικό σύστημα για να συλλαμβάνει κάθε δόση και να την παραδίδει στην τοποθεσία για εφοδιασμό ή για διανομή. Το κεφάλι κινείται κατά μήκος ενός άξονα, ενώ περιστρέφεται 180 μοίρες για να τραβήξει τα φάρμακα. Κατά τη διάρκεια αυτής της διαδικασίας, χρησιμοποιεί την τεχνολογία barcode για την επαλήθευση του σωστού φάρμακου. Μετά παρέχει το φάρμακο στον ασθενή μέσω ενός ειδικού κάδου με ένα μεταφορικό ιμάντα. Μόλις ο κάδος γεμίσει με όλα τα φάρμακα ενός συγκεκριμένου ασθενούς, απελευθερώνεται και επιστρέφεται με το μεταφορικό ιμάντα σε έναν τεχνικό ο οποίος τα φορτώνει σε ένα καλάθι για την παράδοση στον ασθενή. (Murray) (McKesson)

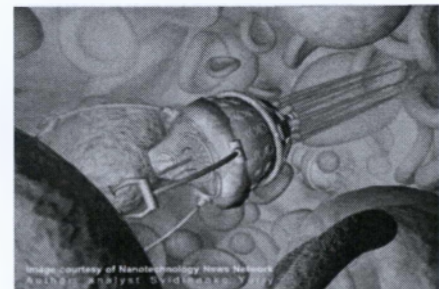
1.7.5 Ερευνητικά Ρομπότ

Ενώ τα περισσότερα ρομπότ σήμερα έχουν εγκατασταθεί σε εργοστάσια ή σε σπίτια, εκτελώντας κοπιαστικές και σωστικές εργασίες, πολλά νέα είδη ρομπότ αναπτύσσονται σε εργαστήρια σε όλο τον κόσμο. Μεγάλο μέρος της έρευνας στη ρομποτική δεν εστιάζει σε συγκεκριμένες βιομηχανικές εργασίες, αλλά σε έρευνες σχετικά με νέους τύπους ρομπότ, εναλλακτικούς τρόπους για να εξετάσουμε ή να σχεδιάσουμε ένα ρομπότ, και νέους τρόπους για την παρασκευή τους. Αναμένεται ότι αυτοί οι νέοι τύποι ρομπότ θα είναι σε θέση να λύσουν προβλήματα του πραγματικού κόσμου, όταν τελικά πραγματοποιηθούν.

Νανορομπότ

Η Νανορομποτική (Nanorobotics) είναι ένας αναδυόμενος τομέας της τεχνολογίας για τη δημιουργία μηχανών ή ρομπότ των οποίων τα μέρη είναι ακριβώς ή κοντά στην μικροσκοπική κλίμακα του μανομέτρου (10^{-9} μέτρα). Επίσης γνωστά και ως «nanobots» ή «nanites», που θα κατασκευάζονται από μοριακές μηχανές. Μέχρι στιγμής, οι ερευνητές έχουν παράγει ως επί το πλείστον μόνο τμήματα αυτών των πολύπλοκων συστημάτων, όπως έδρανα, αισθητήρες, και συνθετικά μοριακών κινητήρων, αλλά και ρομπότ που λειτουργούν έχουν κάνει την εμφάνισή τους στο διαγωνισμό Robocup (Nanobot Robocup contest).

(Nist.gov) Επίσης οι ερευνητές ελπίζουν να είναι σε θέση να δημιουργήσουν ολόκληρα ρομπότ τόσο μικρά όσο οι ιοί ή τα βακτήρια, τα οποία θα μπορούν να εκτελούν λειτουργίες σε μικροσκοπική κλίμακα. Πιθανές εφαρμογές περιλαμβάνουν τη μικρο χειρουργική επέμβαση (σε επίπεδο μεμονωμένων κυττάρων), το utility fog (Βοηθητικό πρόγραμμα προβολής ομίχλης), (Dvorsky) την βιομηχανοποίηση, τα όπλα και τον καθαρισμό. (wiseGeek) Μερικοί άνθρωποι έχουν υπαινιχθεί ότι αν υπήρχαν νανορομπότ που θα μπορούσαν να αναπαραχθούν, η γη θα μετατραπεί σε «grey goo» (σενάριο καταστροφής του κόσμου), ενώ άλλοι υποστηρίζουν ότι αυτό το υποθετικό αποτέλεσμα είναι ανοησίες. (CMStewart)



Εικ.81 Επισκευή κυττάρου με νανορομπότ

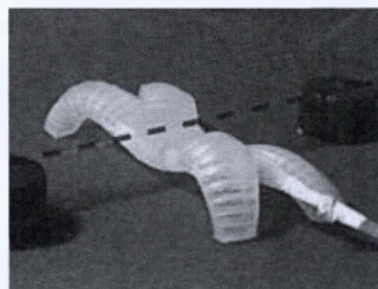
να εκτελούν λειτουργίες σε μικροσκοπική κλίμακα. Πιθανές εφαρμογές περιλαμβάνουν τη μικρο χειρουργική επέμβαση (σε επίπεδο μεμονωμένων κυττάρων), το utility fog (Βοηθητικό πρόγραμμα προβολής ομίχλης), (Dvorsky) την βιομηχανοποίηση, τα όπλα και τον καθαρισμό. (wiseGeek) Μερικοί άνθρωποι έχουν υπαινιχθεί ότι αν υπήρχαν νανορομπότ που θα μπορούσαν να αναπαραχθούν, η γη θα μετατραπεί σε «grey goo» (σενάριο καταστροφής του κόσμου), ενώ άλλοι υποστηρίζουν ότι αυτό το υποθετικό αποτέλεσμα είναι ανοησίες. (CMStewart)

Αναδιαρθρώσιμα Ρομπότ

Λίγοι ερευνητές έχουν διερευνήσει τη δυνατότητα της δημιουργίας ρομπότ που μπορούν να αλλοιώσουν τη φυσική τους μορφή για να ταιριάζουν σε μια συγκεκριμένη εργασία, (Hardesty) όπως το φανταστικό T-1000. Τα σημερινά ρομπότ είναι πολύ μακριά από την εκλεπτυσμένη αυτή ιδέα, που αποτελείται ως επί το πλείστον από ένα μικρό αριθμό μονάδων σε σχήμα κύβου τα οποία μπορούν να κινηθούν αναφορικά (συγγενικά) με τους γείτονές τους. Αρκετοί αλγόριθμοι έχουν σχεδιαστεί σε περίπτωση που οποιαδήποτε τέτοιο ρομπότ γίνει πραγματικότητα. (hindawi)

Απαλά Ρομπότ

Ρομπότ με φορείς σιλικόνης (σώμα από σιλικόνη) και ευέλικτους ενεργοποιητές (μύες αέρα (air muscles), ηλεκτρο-πολυμερή (electroactive polymers), και σιδηρομαγνητικά (ferrofluids)) που ελέγχονται με τη τεχνολογία της ασαφούς λογικής και των νευρωνικών δικτύων, φαίνονται και αισθάνονται διαφορετικά από τα ρομπότ με άκαμπτο σκελετό, και μπορούν να έχουν διαφορετικές συμπεριφορές. (Whitesides)



Εικ.82 Απαλό ρομπότ του Harvard

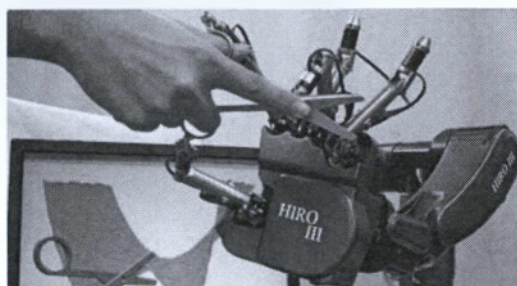
Ρομπότ Σμήνος

Εμπνευσμένο από αποικίες εντόμων, όπως τα μυρμήγκια και οι μέλισσες, οι ερευνητές μοντελοποιούν τη συμπεριφορά των σμηνών από χιλιάδες μικροσκοπικά ρομπότ που εκτελούν ένα χρήσιμο έργο, όπως η εύρεση κάτι κρυφού, τον καθαρισμό, ή την κατασκοπεία. Κάθε ρομπότ είναι αρκετά απλό, αλλά η αναδυόμενη συμπεριφορά του σμήνους είναι πιο πολύπλοκη. Το σύνολο των ρομπότ μπορεί να θεωρηθεί ως ένα ενιαίο κατανεμημένο σύστημα, κατά τον ίδιο τρόπο μια αποικία μυρμηγκιών μπορεί να θεωρηθεί ένας υπεροργανισμός, παρουσιάζοντας νοημοσύνη σμήνους. Τα μεγαλύτερα σμήνη μέχρι στιγμής που έχουν δημιουργηθεί περιλαμβάνουν το σμήνος iRobot,

(Saenz) το project SRI / MobileRobots CentiBots (ai.sri) και το Open-source μικρο-ρομπότ σμήνος (Open-source Micro-robotic Project swarm), (Deyle) τα οποία χρησιμοποιούνται για την έρευνα συλλογικής συμπεριφοράς. Επίσης τα σμήνη είναι πιο ανθεκτικά σε αποτυχίες, όταν ένα μεγάλο ρομπότ αποτύχει τότε καταστρέψετε όλη η αποστολή, ενώ ένα σμήνος μπορεί να συνεχίσει ακόμα και αν αποτύχουν αρκετά ρομπότ. Αυτό το πλεονέκτημα τα καθιστά ελκυστικά για τις αποστολές εξερεύνησης του διαστήματος, όπου η αποτυχία είναι εξαιρετικά δαπανηρή. (nasa.space swarm)

Ρομπότ απτικής διασύνδεσης

Η ρομποτική έχει επίσης εφαρμογή στο σχεδιασμό της εικονικής πραγματικότητας διασυνδέσεων. Εξειδικευμένες ρομποτικές εφαρμογές χρησιμοποιούνται ευρέως στην απτική ερευνητική κοινότητα. Αυτά τα ρομπότ, που ονομάζονται "απτικές διασυνδέσεις,"



Εικ.83 Hiro III, ρομπότ απτικής διασύνδεσης

επιτρέπουν την «ενεργοποίηση με την αφή» (touch-enabled) αλληλεπίδραση του χρήστη με το πραγματικό και το εικονικό περιβάλλον. Ρομποτικές δυνάμεις επιτρέπουν την προσομοίωση των μηχανικών ιδιοτήτων των «εικονικών» αντικείμενων, τα οποία οι χρήστες μπορούν να βιώσουν μέσα από την αίσθηση της αφής. (Member, IEEE)

1.8 Βιομηχανικά ρομπότ

Ένα βιομηχανικό ρομπότ ορίζεται από την ISO (iso 8373) ως ένα αυτόματα ελεγχόμενο, επαναπρογραμματιζόμενο, πολλαπλών χρήσεων βραχίονα προγραμματιζόμενο σε τρεις ή περισσότερους άξονες. Αυτό το πεδίο της ρομποτικής μπορεί να οριστεί ως η μελέτη, ο σχεδιασμός και η χρήση των ρομποτικών συστημάτων για την κατασκευή βιομηχανικών προϊόντων.

Τυπικές εφαρμογές των ρομπότ περιλαμβάνουν τη συγκόλληση, το χρωματισμό, τη συναρμολόγηση, και τον τόπο παραλαβής (όπως η συσκευασία, παλετοποίηση και SMT), την επιθεώρηση και τον έλεγχο των προϊόντων και όλα αυτά επιτυγχάνονται με υψηλή αντοχή, ταχύτητα και ακρίβεια.

Οι πιο συχνές χρησιμοποιούμενες προδιαγραφές ρομπότ είναι τα αρθρωτά ρομπότ (articulated robots), τα ρομπότ SCARA, τα Delta ρομπότ και τα Καρτεσιανά ρομπότ συντεταγμένων (Cartesian coordinate robots), (γνωστά και ως ασφάλινου σκελετού (gantry) ρομπότ ή x-y-z ρομπότ). (Robotworx) Στο πλαίσιο της γενικής ρομποτικής, τα περισσότερα είδη ρομπότ εμπίπτουν στην κατηγορία των ρομποτικών βραχιόνων (εγγενείς στη χρήση του βραχίονα λέξης στο προαναφερθέν πρότυπο ISO). Τα συγκεκριμένα ρομπότ παρουσιάζουν ποικίλους βαθμούς αυτονομίας:



Εικ.84 Scara THL400 ,Toshiba

- Μερικά ρομπότ είναι προγραμματισμένα για την πιστή εκτέλεση συγκεκριμένων ενεργειών ξανά και ξανά (επαναλαμβανόμενες πράξεις) χωρίς μεταβολές και με υψηλό βαθμό ακρίβειας. Αυτές οι ενέργειες προσδιορίζονται από προγραμματισμένες ρουτίνες που καθορίζουν την κατεύθυνση, την επιτάχυνση, την ταχύτητα, την επιβράδυνση, και την απόσταση από μια σειρά συντονισμένων κινήσεων.
- Άλλα ρομπότ είναι πολύ πιο ευέλικτα ως προς τον προσανατολισμό του αντικειμένου επί του οποίου λειτουργούν ή ακόμη ως προς το έργο που πρέπει να

πραγματοποιηθεί στο ίδιο το αντικείμενο, που το ρομπότ μπορεί να χρειαστεί να προσδιορίσει. Για παράδειγμα τα ρομπότ συχνά περιέχουν μηχανικής όρασης υποσυστήματα που ενεργούν ως τα "μάτια" τους, που συνδέονται με ισχυρούς υπολογιστές ή ελεγκτές. Η Τεχνητή Νοημοσύνη, ή ό, τι εμπλέκετε γι 'αυτή, γίνεται όλο ένα και πιο σημαντικός παράγοντας στα σύγχρονα βιομηχανικά ρομπότ.

1.8.1 Τεχνική περιγραφή

Καθορισμός των παραμέτρων

- Αριθμός αξόνων - δύο άξονες απαιτούνται για να φτάνουμε σε οποιοδήποτε σημείο ενός επιπέδου · τρεις άξονες απαιτούνται για την πρόσβαση σε κάθε σημείο του χώρου. Για να ελέγχουν πλήρως τον προσανατολισμό του άκρου του βραχίονα (δηλαδή καρπού) απαιτούνται τρεις περισσότεροι άξονες (yaw, pitch και roll). Μερικά σχέδια (π.χ. το ρομπότ SCARA) ανταλλάσσουν τους περιορισμούς στις δυνατότητες κίνησης για κόστος, ταχύτητα και ακρίβεια.
- «Βαθμούς ελευθερίας» (DOF) - το οποίο είναι συνήθως ίδιο με τον αριθμό των αξόνων. (Motion Controls Robotics)
- Working envelope - η περιοχή του χώρου που ένα ρομπότ μπορεί να φτάσει.
- Κινηματική (Kinematics)- η πραγματική ρύθμιση των άκαμπτων μελών και των αρθρώσεων του ρομπότ, το οποίο προσδιορίζει τις πιθανές κινήσεις του ρομπότ. Οι κατηγορίες των κινηματικών ρομπότ περιλαμβάνουν τα αρθρωτά ,τα καρτεσιανά, τα παράλληλα και τα SCARA.
- Φέρουσα ικανότητα ή payload - το βάρος που μπορεί να σηκώσει ένα ρομπότ.
- Ταχύτητα - πόσο γρήγορα το ρομπότ μπορεί να τοποθετήσει το άκρο του βραχίονα του. Αυτό μπορεί να ορίζεται από την άποψη της γωνιακής ή της γραμμικής ταχύτητας του κάθε άξονα ή ως σαν σύνθετη ταχύτητα, δηλαδή την ταχύτητα του άκρου του βραχίονα όταν όλοι οι άξονες κινούνται.
- Επιτάχυνση - πόσο γρήγορα ένας άξονας μπορεί να επιταχύνει. Δεδομένου ότι αυτό είναι ένας περιοριστικός παράγοντας ένα ρομπότ μπορεί να μην είναι σε θέση να

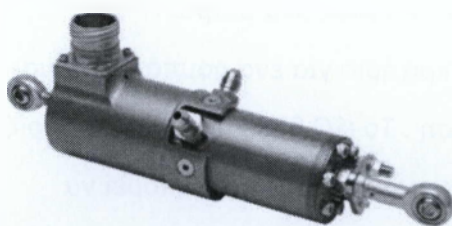
- φθάσει την καθορισμένη μέγιστη ταχύτητα για μετακινήσεις σε μικρές αποστάσεις ή μια σύνθετη διαδρομή που απαιτεί συχνές αλλαγές κατεύθυνσης.
- Ακρίβεια - πόσο πολύ ένα ρομπότ μπορεί να φτάσει μια εντεταλμένη θέση. Όταν η απόλυτη θέση του ρομπότ μετράται και συγκρίνεται με την εντολή θέσης το σφάλμα είναι ένα μέτρο της ακρίβειας. Η ακρίβεια μπορεί να βελτιωθεί με εξωτερικούς αισθητήρες για παράδειγμα ένα σύστημα όρασης ή υπέρυθρων. (Δείτε robot calibration). Επίσης μπορεί να ποικίλει ανάλογα με την ταχύτητα και τη θέση εντός του Working envelope (εργασιακού φάκελου) και με το payload (ωφέλιμο φορτίο).
 - Επαναληψιμότητα (Repeatability) - πόσο καλά το ρομπότ θα επιστρέψει σε μια προγραμματισμένη θέση. Αυτό δεν είναι το ίδιο με την ακρίβεια. Μπορεί να είναι ότι, όταν ζητείται να πάει σε μια συγκεκριμένη θέση X-Y-Z απέχει μόνο 1 χιλ. από αυτή τη θέση. Αυτό θα ήταν η ακρίβειά της, η οποία μπορεί να βελτιωθεί με βαθμονόμηση. Αλλά αν αυτή η θέση αποτυπώνεται στη μνήμη του ελεγκτή και κάθε φορά που αποστέλλεται εκεί επιστρέψει εντός του 0,1 χιλ. της αποτυπωμένης θέσης, τότε η επαναληψιμότητα θα είναι μέσα σε 0,1 χιλιοστο. (robotmatrix)

Η ακρίβεια και η επαναληψιμότητα είναι δυο διαφορετικά μέτρα. Η επαναληψιμότητα είναι συνήθως το πιο σημαντικό κριτήριο για ένα ρομπότ και είναι παρόμοια με την έννοια της «ακρίβειας» στη μέτρηση. Το ISO 9283 (robotiq) καθορίζει μια μέθοδο με την οποία τόσο η ακρίβεια όσο και η επαναληψιμότητα μπορεί να μετρηθεί. Τυπικά ένα ρομπότ αποστέλλεται σε μία συγκεκριμένη θέση αρκετές φορές και το σφάλμα μετράται σε κάθε επιστροφή στη θέση μετά την επίσκεψη 4 άλλων θέσεων. Στη συνέχεια προσδιορίζεται ποσοτικά χρησιμοποιώντας την τυπική απόκλιση των δειγμάτων σε όλες τις τρεις διαστάσεις. Ένα τυπικό ρομπότ μπορεί βέβαια να κάνει ένα σφάλμα θέσης που υπερβαίνει τον υπολογισμό και μπορεί να είναι ένα πρόβλημα για τη διαδικασία. Επιπλέον, η επαναληψιμότητα είναι διαφορετική σε διαφορετικά τμήματα του εκτεινόμενου χώρου (Working envelope) και αλλάζει επίσης με την ταχύτητα και το ωφέλιμο φορτίο (payload). Το ISO 9283 διευκρινίζει ότι η ακρίβεια και η επαναληψιμότητα πρέπει να μετρείται με τη μέγιστη ταχύτητα και το μέγιστο ωφέλιμο φορτίο. Όμως αυτό οδηγεί σε απαισιόδοξες τιμές, ενώ το ρομπότ θα μπορούσε να είναι πολύ πιο ακριβές και επαληθεύσιμο σε ελαφρύ φορτία και μικρές

ταχύτητες. Σε μια βιομηχανική διαδικασία η επαναληψιμότητα είναι επίσης θέμα αναφοράς στην ακρίβεια της ακραίας απόληξης, για παράδειγμα ένας gripper (άρπαγας), ακόμα και στο σχεδιασμό των «δακτύλων» που ταιριάζουν με το gripper στο αντικείμενο που πιάνεται. Για παράδειγμα, αν ένα ρομπότ κρατάει μια βίδα από το κεφάλι της, ο κοχλίας μπορεί να είναι σε μία τυχαία γωνία και μια προσπάθεια να εισαγάγει τη βίδα σε μια τρύπα θα μπορούσε εύκολα να αποτύχει. Αυτά και άλλα παρόμοια σενάρια μπορεί να βελτιωθούν π.χ. με «lead-ins» μετατρέποντας την είσοδο στην τρύπα σε κωνικό σχήμα.

Έλεγχος κίνησης (Motion control) - για ορισμένες εφαρμογές, όπως η απλή συναρμολόγηση pick-and-place, το ρομπότ χρειάζεται απλώς να επιστρέφει επανειλημμένα σε περιορισμένο αριθμό καθορισμένων θέσεων. Για πιο εξελιγμένες εφαρμογές, όπως η συγκόλληση και το φινιρίσμα (βαφή με ψεκασμό), η κίνηση πρέπει να ελέγχεται συνεχώς για να ακολουθεί μια διαδρομή στο χώρο, με ελεγχόμενο προσανατολισμό και ταχύτητα.

Πηγή ισχύος - κάποια ρομπότ χρησιμοποιούν ηλεκτρικούς κινητήρες και άλλα χρησιμοποιούν υδραυλικούς ενεργοποιητές. Οι ηλεκτρικοί κινητήρες είναι πιο γρήγοροι

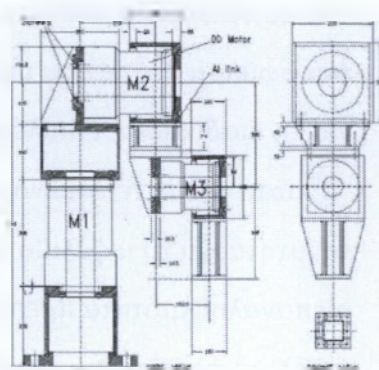


Εικ.85 Υδραυλικός ενεργοποιητής

ενώ οι υδραυλικοί ενεργοποιητές είναι ισχυρότεροι και πλεονεκτικοί σε εφαρμογές όπως η βαφή με ψεκασμό, όπου ένας σπινθήρας θα μπορούσε να πυροδοτήσει μια έκρηξη. Ωστόσο, χαμηλή στάθμη εσωτερικής συμπίεσης-αέρα του

βραχίονα μπορεί να αποτρέψει την είσοδο των εύφλεκτων ατμών, καθώς και άλλων προσμείξεων.

Μεταφορά- κάποια ρομπότ συνδέουν ηλεκτροκινητήρες στις αρθρώσεις μέσω των γραναζιών· άλλα συνδέουν τον κινητήρα στην άρθρωση απευθείας (direct drive). Χρησιμοποιώντας γρανάζια έχει ως αποτελέσματα την μετρήσιμη «backlash» (σπασμωδική κίνηση), το οποίο είναι η ελεύθερη κίνηση μέσα σε έναν άξονα. Μικρότεροι ρομποτικοί βραχίονες χρησιμοποιούν



Εικ.86 Αρχιτεκτονική του Direct Drive ρομπότ

συχνά υψηλές ταχύτητες, χαμηλές ροπές DC κινητήρων και γενικά απαιτούν υψηλή αναλογία δανειοδότησης' τα οποία είναι τα μειονεκτήματα της σπασμωδικής κίνησης. Σε τέτοιες περιπτώσεις η αρμονική κίνηση χρησιμοποιείται ποιο συχνά.

Ενδοτικότητα (Compliance)- είναι ένα ποσοστιαίο μέτρο στην γωνία ή την απόσταση που ένας ρομποτικός άξονας θα κινηθεί όταν μια δύναμη εφαρμοστεί σε αυτό. Λόγω της ενδοτικότητας, όταν ένα ρομπότ κινηθεί σε μία τοποθεσία φέροντας το μέγιστο ωφέλιμο φορτίο του, θα είναι σε μία θέση ελαφρώς χαμηλότερη από ό, τι όταν μεταφέρει μηδενικό ωφέλιμο φορτίο. Επίσης η ενδοτικότητα μπορεί να είναι υπεύθυνη για την υπέρβαση κατά την μεταφορά υψηλών ωφέλιμων φορτίων ,όπου στην συγκεκριμένη περίπτωση η επιτάχυνση θα πρέπει να μειωθεί.

1.8.2 Ρομποτικός Προγραμματισμός και Διεπαφές

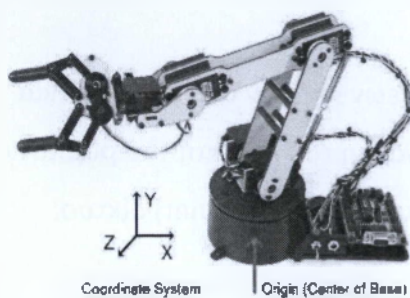
Η εγκατάσταση ή ο προγραμματισμός των κινήσεων και των αλληλουχιών για ένα βιομηχανικό ρομπότ συνήθως συμβαίνει με τη σύνδεση του ελεγκτή του ρομπότ σε ένα φορητό υπολογιστή, επιτραπέζιο υπολογιστή ή (εσωτερικό ή Internet) δίκτυο.

Ένα ρομπότ και μια συλλογή μηχανημάτων ή περιφερειακών αναφέρεται ως ένα workcell, ή cell. Ένα τυπικό cell μπορεί να περιέχει ένα μέρος τροφοδοσίας, μια μηχανή χύτευσης και ένα ρομπότ. Τα διάφορα μηχανήματα είναι «ενσωματωμένα» και ελέγχονται από ένα μόνο υπολογιστή ή PLC. Ο τρόπος που το ρομπότ αλληλεπιδρά με τα άλλα μηχανήματα στο cell πρέπει να προγραμματιστεί, τόσο σε σχέση με τις θέσεις τους στο cell όσο και στο συγχρονισμό με αυτά.

Λογισμικό: Ο υπολογιστής έχει εγκατασταθεί με το αντίστοιχο λογισμικό διεπαφής. Η χρήση ενός υπολογιστή απλοποιεί σημαντικά τη διαδικασία προγραμματισμού. Εξειδικευμένο λογισμικό του ρομπότ εκτελείται είτε στον ελεγκτή του ρομπότ είτε στον υπολογιστή ή και τα δύο, ανάλογα με το σχεδιασμό του συστήματος.

Υπάρχουν δύο βασικές ενότητες που πρέπει να μαθευτούν (ή προγραμματιστούν): τα δεδομένα θέσης (positional data) και η διαδικασία (procedure). Για παράδειγμα, σε μια εργασία για να κινηθεί μια βίδα από έναν τροφοδότη σε μία τρύπα οι θέσεις του τροφοδότη και της οπής πρέπει πρώτα να μαθευτούν ή να προγραμματιστούν. Δεύτερον, η διαδικασία για να πάρει τη βίδα από τον τροφοδότη προς την οπή πρέπει να προγραμματιστεί μαζί με οποιαδήποτε I / O που εμπλέκεται, για παράδειγμα ένα σήμα για να υποδεικνύει πότε ο κοχλίας μέσα στον τροφοδότη είναι έτοιμος να σηκωθεί. Ο σκοπός του λογισμικού του ρομπότ είναι να διευκολύνει τις δύο αυτές εργασίες προγραμματισμού.

Η διδασκαλία των θέσεων του ρομπότ μπορεί να επιτευχθεί με διάφορους τρόπους:



Εικ.87 X-Y-Z Σύστημα συντεταγμένων

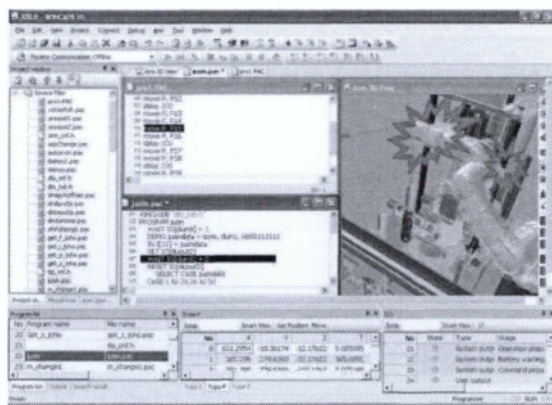
Εντολές Θέσεων : Το ρομπότ μπορεί να κατευθυνθεί προς την επιθυμητή θέση με ένα GUI ή text based commands (εντολές σε βάση κειμένου) όπου η απαιτούμενη θέση X-Y-Z μπορεί να προσδιοριστεί και επεξεργαστεί.

Συσκευή Διδασκαλίας (Teach Pendant): Οι θέσεις του ρομπότ μπορούν να διδαχτούν μέσω της συσκευής διδασκαλίας, η οποία είναι μια φορητή μονάδα ελέγχου και προγραμματισμού. Τα κοινά χαρακτηριστικά των μονάδων αυτών είναι η δυνατότητα να στέλνουν χειροκίνητα το ρομπότ στην επιθυμητή θέση, ή "inch" ή "jog" για να ρυθμίσουν τη θέση του. Έχουν επίσης μέσα για να αλλάζουν την ταχύτητα ,αφού μια χαμηλή ταχύτητα είναι συνήθως απαιτούμενη για την προσεκτική τοποθέτηση τους, ή κατά τη διάρκεια δοκιμών μέσα από μια νέα ή τροποποιημένη ρουτίνα. Επίσης ένα μεγάλο κουμπί διακοπής λειτουργίας έκτακτης ανάγκης είναι συνήθως συμπεριλαμβανόμενο. Συνήθως, μετά την πρώτη φορά που το ρομπότ προγραμματιστεί δεν χρειάζεται πλέον η χρήση της συσκευής διδασκαλίας. (WG.teach pendant)



Εικ.88 Fanuc A05B-2301-C370 RJ Series Spot Welding Teach Pendant

«Καθοδήγηση από τη μύτη» (Lead-by-the-nose): είναι μια τεχνική που προσφέρεται από πολλούς κατασκευαστές ρομπότ. Στη μέθοδο αυτή, ένας χρήστης κρατά το βραχίονα του ρομπότ, ενώ ένας άλλος εισάγει μια εντολή που απενεργοποιεί το ρομπότ με αποτέλεσμα να χαλαρώνει (μεταβεί στην λειτουργία «limb»). Ο χρήστης μετακινεί στη συνέχεια το ρομπότ με το χέρι στις απαιτούμενες θέσεις και / ή κατά μήκος μιας απαιτούμενης πορείας, ενώ το λογισμικό καταγράφει αυτές τις θέσεις στη μνήμη. Το πρόγραμμα μπορεί αργότερα να κινήσει το ρομπότ σε αυτές τις θέσεις ή κατά μήκος της διαδρομής που διδάχτηκε. Η τεχνική αυτή είναι δημοφιλής για εργασίες όπως η βαφή με ψεκασμό.



Εικ.90 Ρομποτικό πρόγραμμα που υποστηρίζει offline λειτουργικότητα.

Αποσυνδεδεμένος Προγραμματισμός (Offline programming): είναι όπου το σύνολο των κυττάρων, το ρομπότ και όλα τα μηχανήματα ή τα εργαλεία στο χώρο εργασίας έχουν χαρτογραφηθεί γραφικά. Το ρομπότ στη συνέχεια μπορεί να μετακινηθεί στην οθόνη με τη διαδικασία προσομοίωσης. Η τεχνική έχει περιορισμένη αξία επειδή βασίζεται στις ακριβείς

μέτρησης των θέσεων του αντιστοίχου εξοπλισμού και επίσης βασίζεται στην ακρίβεια της θέσης του ρομπότ το οποίο μπορεί ή δεν μπορεί να είναι σύμφωνες με ό, τι έχει προγραμματιστεί. (baga)

Άλλοι Τρόποι: Επιπλέον, οι χειριστές μηχανημάτων χρησιμοποιούν συχνά συσκευές διεπαφής του χρήστη, συνήθως συσκευές με οθόνη αφής, τα οποία χρησιμεύουν ως τον πίνακα ελέγχου του χειριστή. Ο χειριστής μπορεί να κινείται από πρόγραμμα σε πρόγραμμα, να κάνει προσαρμογές στο πλαίσιο ενός προγράμματος και επίσης να χειρίζεται μια σειρά από περιφερειακές συσκευές που



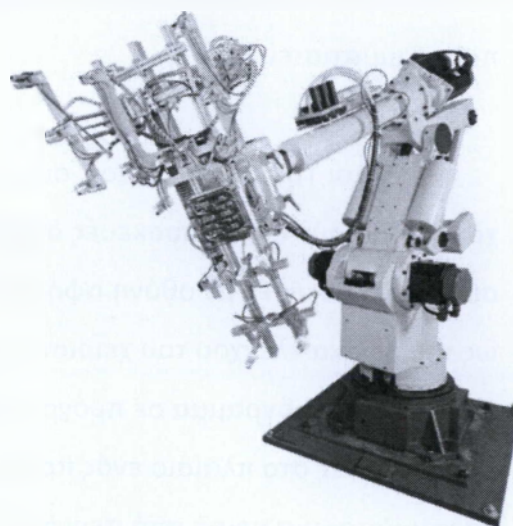
Εικ.89 Siemens Sinumerik CNC, πίνακας ελέγχου

μπορούν να ενσωματωθούν μέσα στο ίδιο το ρομποτικό σύστημα. Αυτές περιλαμβάνουν τελεστές τέλους (end effectors), τροφοδότες που παρέχουν στοιχεία για το ρομπότ, μεταφορικούς ιμάντες (conveyor belts), διακόπτες εκτάκτου ανάγκης, συστήματα μηχανικής όρασης, συστήματα αλληλασφάλισης (safety interlock), εκτυπωτές γραμμωτού κώδικα (bar code) και μια σχεδόν άπειρη ποικιλία από άλλες βιομηχανικές συσκευές οι οποίες είναι προσβάσιμες και ελεγχόμενες μέσω του πίνακα ελέγχου του χειριστή .

Η συσκευής διδασκαλίας (Teach pendant) ή το PC συνήθως αποσυνδέονται μετά τον προγραμματισμό τους και το ρομπότ στη συνέχεια λειτουργεί με πρόγραμμα που έχει εγκατασταθεί στον ελεγκτή του. Ωστόσο, ένας υπολογιστής χρησιμοποιείται συχνά για να «εποπτεύει» το ρομπότ και οποιαδήποτε περιφερειακά, ή να παρέχει επιπλέον χώρο αποθήκευσης για την πρόσβαση σε πολλές άλλες πολύπλοκες διαδρομές και τις ρουτίνες.

End-of-arm Εργαλεία

Το πιο ουσιαστικό περιφερικό του ρομπότ είναι ο τελεστής τέλους (end effector), ή end-of-arm εργαλεία (EOT). Κοινά παραδείγματα τελεστών τέλους (end effectors) περιλαμβάνουν συσκευές συγκόλλησης (όπως όπλα συγκόλλησης MIG, spot-συγκολλητές, κλπ.), πιστόλια ψεκασμού , συσκευές λείανσης και αφαίρεσης αιχμών (όπως δίσκος πεπιεσμένου αέρα ή ιμάντινοι τροχοί, γρέζια (burrs), κ.λπ.) και grippers (συσκευές που μπορούν να αρπάξουν ένα αντικείμενο, συνήθως ηλεκτρομηχανικά ή πεπιεσμένου αέρα) · ένα άλλο κοινό μέσο για να σηκώνουμε ένα αντικείμενο είναι με το κενό (Vacuum). Οι τελεστές τέλους είναι συχνά εξαιρετικά πολύπλοκοι, φτιαγμένοι για να ταιριάζουν με το χειρισμό του προϊόντος και συχνά ικανοί να μαζεύουν (σηκώνουν) μια



Εικ.91 Ένα end-of-arm-tool πάνω σε Fanuc ρομπότ

σειρά προϊόντων τι φορά. Μπορούν να χρησιμοποιούν διάφορους αισθητήρες για να βοηθούν το σύστημα του ρομπότ στον εντοπισμό, το χειρισμό και τη τοποθέτηση προϊόντων.

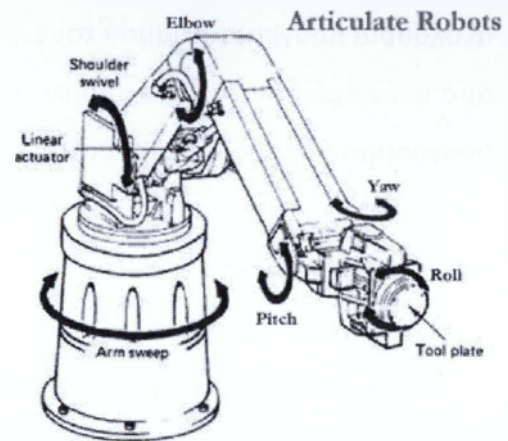
1.8.3 Έλεγχος της κίνησης

Για ένα δεδομένο ρομπότ οι μόνοι παράμετροι απαραίτητοι για να εντοπίσει εντελώς τον τελεστή τέλους (gripper, εργαλείο συγκόλλησης, welding torch, κλπ.) του ρομπότ είναι οι γωνίες για καθεμία άρθρωση ή οι μετατοπίσεις των γραμμικών αξόνων (ή ο συνδυασμός και των δύο για ρομποτικές μορφές όπως τα SCARA).

Ωστόσο, υπάρχουν πολλοί διαφορετικοί τρόποι για να καθοριστούν τα σημεία. Ο πιο κοινός και πιο βολικός τρόπος καθορισμού ενός σημείου είναι να καθοριστεί μια καρτεσιανή συντεταγμένη για αυτό, δηλαδή η θέση της «ακραίας απόληξης» σε mm στο επίπεδο X, Y και Z κατευθύνσεων σε σχέση με την προέλευση του ρομπότ. Επιπλέον, ανάλογα με τους τύπους των αρθρώσεων ένα συγκεκριμένο ρομπότ μπορεί να έχει, ο προσανατολισμός της ακραίας απόληξης στο yaw, pitch και roll, και η θέση του σημείου του εργαλείου σε σχέση με την μετωπική πλάκα του ρομπότ θα πρέπει επίσης να προσδιορίζονται. Για έναν αρθρωτό βραχίονα αυτές οι συντεταγμένες πρέπει να μετατραπούν σε γωνίες αρθρώσεων από τον ελεγκτή του ρομπότ, οι οποίες μετατροπές είναι γνωστές ως Καρτεσιανοί Μετασχηματισμοί και μπορεί να χρειαστούν να εκτελούνται επαναληπτικά ή αναδρομικά για ένα πολυαξονικό ρομπότ. Τα μαθηματικά των σχέσεων μεταξύ των γωνιακών αρθρώσεων και των πραγματικών χωρικών συντεταγμένων ονομάζεται Κινηματική (kinematics).

(αρχiv)

Η τοποθέτηση από καρτεσιανές συντεταγμένες μπορεί να γίνει με την είσοδο των συντεταγμένων στο σύστημα ή χρησιμοποιώντας μια συσκευή διδασκαλίας η



Εικ.92 Yaw Pitch και Roll sketch

οποία κινεί το ρομπότ σε X-Y-Z κατευθύνσεις. Είναι πολύ πιο εύκολο για τον χειριστή να απεικονίσει κινήσεις πάνω / κάτω, αριστερά / δεξιά, κλπ. από το να κινήσει κάθε άρθρωση τη φορά. Όταν η επιθυμητή θέση επιτευχθεί τότε ορίζεται ,κατά κάποιο τρόπο ειδικότερα, με τη χρήση του λογισμικού του ρομπότ, π.χ. P1 - P5 παρακάτω.

Τυπικός Προγραμματισμός

Τα περισσότερα αρθρωτά ρομπότ εκτελούν τις λειτουργίες τους αποθηκεύοντας μία σειρά θέσεων στη μνήμη, και κινούνται προς αυτές σε διάφορους χρόνους κατά την ακολουθία προγραμματισμού τους. Για παράδειγμα, ένα ρομπότ που κινεί πράγματα από το ένα μέρος στο άλλο μπορεί να έχει «pick and place» («επιλέγω και τοποθετώ») πρόγραμμα παρόμοιο με το ακόλουθο: (Stenmark)

Καθορισμός σημείων P1-P5:

1. Με ασφάλεια πάνω από το τεμάχιο επεξεργασίας (που ορίζεται ως P1)
2. 10 cm πάνω από δοχείο A (που ορίζεται ως P2)
3. Σε θέση να λάβει το υλικό από το δοχείο A (που ορίζεται ως P3)
4. 10 cm πάνω από το δοχείο B (που ορίζεται ως P4)
5. Σε θέση να λάβει το υλικό από το δοχείο B. (που ορίζεται ως P5)

Ορισμός του προγράμματος:

1. Μετακίνηση στο P1
2. Μετακίνηση στο P2
3. Μετακίνηση στο P3
4. Κλείσιμο gripper (άρπαγα , γάντζου)

5. Μετακίνηση στο P2
6. Μετακίνηση στο P4
7. Μετακίνηση στο P5
8. Άνοιγμα gripper (άρπαγα , γάντζου)
9. Μετακίνηση στο P4
10. Μετακίνηση στο P1 και φινίρισμα (τελος)

Παράδειγμα VAL προγράμματος :

PROGRAM PICKPLACE

```
1. MOVE P1
2. MOVE P2
3. MOVE P3
4. CLOSEI 0.00
5. MOVE P4
6. MOVE P5
7. OPENI 0.00
8. MOVE P1
.END
```

Epson RC+ (παράδειγμα για ένα vacuum μάζεμα) :

```
Function PickPlace
  Jump P1
  Jump P2
  Jump P3
  On vacuum
  Wait .1
  Jump P4
  Jump P5
  Off vacuum
  Wait .1
  Jump P1
```

Fend

ROBOFORTH (μια γλώσσα βασισμένη στη FORTH):

```

: PICKPLACE
P1
P3 GRIP WITHDRAW
P5 UNGRIP WITHDRAW
P1
;

```

(Με την Roboforth μπορούμε να καθορίσουμε τις θέσεις προσέγγισης για τις θέσεις έτσι ώστε να μην χρειάζονται τα P2 και P4). (wiki.robot software)

1.8.4 Ανωμαλίες

Το Αμερικανικό Εθνικό Πρότυπο για Βιομηχανικά Ρομπότ και Ρομποτικά Συστήματα- Απαιτήσεις ασφαλείας (American National Standard for Industrial Robots and Robot Systems - Safety Requirements) (ANSI / RIA R15.06-1999) ορίζει μια ιδιομορφία (ανωμαλία) ως «μια κατάσταση που προκαλείται από τη συγγραμμική ευθυγράμμιση δύο ή περισσότερων ρομποτικών αξόνων με αποτέλεσμα την απρόβλεπτη κίνηση και ταχύτητα του ρομπότ.» Είναι πιο συχνό στους ρομποτικούς βραχίονες που χρησιμοποιούν ένα «triple-roll καρπό» (triple-roll wrist), στο οποίο οι τρεις άξονες του καρπού, ο έλεγχος της εκτροπής, της κλίσης και του ρολού (yaw, pitch και roll), περνούν όλα μέσα από ένα κοινό σημείο. Ένα παράδειγμα ενός ιδιόμορφου καρπού είναι όταν η διαδρομή μέσω της οποίας το ρομπότ κινείται προκαλεί το πρώτο και το τρίτο άξονα του ρομποτικού καρπού (δηλαδή ρομποτικοί άξονες 4 και 6) να παραταχτούν. Στη συνέχεια ο δεύτερος άξονας του καρπού επιχειρεί να περιστραφεί κατά 360 ° σε μηδενικό χρόνο για να διατηρήσει τον προσανατολισμό της ακραίας απόληξης. Ένας άλλος κοινός όρος για αυτή την ιδιομορφία είναι το “wrist flip” (αναποδογύρισμα του καρπού) . Το αποτέλεσμα της ιδιομορφίας μπορεί να είναι αρκετά δραματική και μπορεί να έχει αρνητικές επιπτώσεις στο ρομποτικό βραχίονα, στο τελεστή τέλους (ακραία απόληξη) και στη διαδικασία. Ορισμένοι κατασκευαστές βιομηχανικών ρομπότ έχουν προσπαθήσει να παρακάμψουν την κατάσταση με το να μεταβάλουν ελαφρώς την πορεία του ρομπότ. Μία άλλη μέθοδος είναι να επιβραδύνουν την ταχύτητα πορείας του ρομπότ, έτσι ώστε να μειώσουν την ταχύτητα

που απαιτείται για να κάνει τη μετάβαση ο καρπός. Το ANSI / RIA έχει δώσει εντολή ότι οι κατασκευαστές των ρομπότ πρέπει να ενημερώνουν τον χρήστη για τις ιδιομορφίες (ανωμαλίες), εφόσον προκύψουν, ενώ το σύστημα χειρίζεται χειροκίνητα.

Ένας δεύτερος τύπος ιδιομορφίας στα , καρπό-κατανεμημένα ρομπότ κάθετων αρθρώσεων που έχουν έξι άξονες , συμβαίνει όταν το κέντρο του καρπού βρίσκεται σε έναν κύλινδρο που επικεντρώνεται γύρω από τον άξονα 1 και με ακτίνα ίση με την απόσταση μεταξύ των αξόνων 1 και 4. Αυτό ονομάζεται ιδιομορφία ώμου. Επίσης ορισμένοι κατασκευαστές ρομπότ αναφέρουν ιδιομορφίες ευθυγράμμισης, όπου οι άξονες 1 και 6 συμπίπτουν, το οποίο είναι απλά μια υπο-περίπτωση της «ιδιομορφίας ώμου». Όταν το ρομπότ περνά κοντά σε μια ιδιομορφία ώμου, η άρθρωση 1 αρχίζει να περιστρέφεται πολύ γρήγορα.

Το τρίτο και τελευταίο είδος της ιδιομορφίας στα καρπό-κατανεμημένα κάθετα αρθρωτά έξι αξόνων ρομπότ συμβαίνει όταν το κέντρο του καρπού έγκειται στο ίδιο επίπεδο με τους άξονες 2 και 3.

(Ένα βίντεο που απεικονίζει αυτά τα τρία είδη της Singular διαμορφώσεις είναι διαθέσιμες εδώ. (Cogo)) (osha)

1.8.5 Πρόσφατες και μελλοντικές εξελίξεις

Από το 2005, οι επιχειρήσεις των ρομποτικών βραχιόνων πλησιάζουν μια ώριμη κατάσταση, όπου μπορούν να παρέχουν αρκετή ταχύτητα, ακρίβεια και ευκολία στη χρήση για τις περισσότερες από τις εφαρμογές. Το Όραμα καθοδήγησης (γνωστό και ως μηχανική όραση) φέρνει μια μεγάλη ευελιξία στα ρομποτικά κύτταρα (cells). Ωστόσο, η ακραία απόληξη που συνδέεται με ένα ρομπότ είναι συχνά ένα απλό πνευματικό, 2-θέσιο τσοκ (chuck). Αυτό δεν επιτρέπει τα ρομποτικά κύτταρα να χειρίζονται εύκολα διαφορετικά μέρη, σε διαφορετικούς προσανατολισμούς.

Το Hand-in-hand μαζί με την αύξηση των off-line προγραμματισμένων εφαρμογών, έχει ως αποτέλεσμα η ρομποτική βαθμονόμηση να γίνεται όλο και πιο σημαντική προκειμένου να εγγυηθεί μια καλή ακρίβεια τοποθέτησης.

Άλλες εξελίξεις περιλαμβάνουν τη μείωση του μεγέθους των βιομηχανικών βραχιόνων για την ελαφρά βιομηχανική χρήση όπως η παραγωγή μικρών προϊόντων, τη σφράγιση και τη διανομή, τον έλεγχο ποιότητας και τη μεταφορά δειγμάτων στο εργαστήριο. Τέτοια ρομπότ συνήθως κατατάσσονται ως ρομπότ «πάγκου» («bench top»), τα όποια χρησιμοποιούνται σε εφαρμογές καταναλωτών (μικρο-ρομποτικοί βραχίονες). Τα Ρομπότ χρησιμοποιούνται στη φαρμακευτική έρευνα σε μια τεχνική που ονομάζεται υψηλής απόδοσης έλεγχος (High-throughput screening). Επίσης Οι βιομηχανικοί βραχίονες μπορούν να χρησιμοποιηθούν σε συνδυασμό ή ακόμα και τοποθετημένα σε αυτοματοποιημένα καθοδηγούμενα οχήματα (AGVs) , ώστε να καθιστούν την «αλυσίδα αυτοματισμού» (automation chain) πιο ευέλικτη μεταξύ των pick-up και των drop-off.

1.8.6 Η διάρθρωση της αγοράς

Σύμφωνα με τη Διεθνή Ομοσπονδία Ρομποτικής (IFR) στη μελέτη World Robotics 2013, υπήρχαν τουλάχιστον 1.373.000 επιχειρησιακά βιομηχανικά ρομπότ μέχρι το τέλος του 2013. Ο αριθμός αυτός αναμένεται να φτάσει στο 1.659.000 μέχρι το τέλος του 2016. (WR2013)

Το έτος 2011, η IFR εκτιμά ότι τα παγκόσμια προμήθεια βιομηχανικών ρομπότ αυξήθηκε κατά 38% (166.028 μονάδες) ,μακράν το υψηλότερο επίπεδο που έχει καταγραφεί ποτέ για ένα έτος, και ότι οι παγκόσμιες πωλήσεις κατά 46% (\$ 8.500.000.000). Συμπεριλαμβανομένου του κόστους των περιφερειακών μονάδων, του λογισμικού, και των συστημάτων σχεδιασμού η αξία της παγκόσμιας αγοράς για τα ρομποτικά συστήματα το 2011 εκτιμάται ότι είναι 25.5 εκατομμύρια δολάρια. (IFR press release)

Θα πρέπει να σημειωθεί ότι η τιμή αυτή συνήθως δεν περιλαμβάνει το κόστος των περιφερειακών μονάδων, του λογισμικού και τον σχεδιασμό συστημάτων. Συμπεριλαμβανομένων και των παραπάνω δαπανών θα μπορούσε να εκτιμηθεί ότι η πραγματική αξία των ρομποτικών συστημάτων της αγοράς είναι περίπου τρεις φορές

μεγαλύτερη, και η αξία της παγκόσμιας αγοράς για τα ρομποτικά συστήματα το 2011 εκτιμάται ότι είναι 25.5 εκατομμύρια δολάρια.

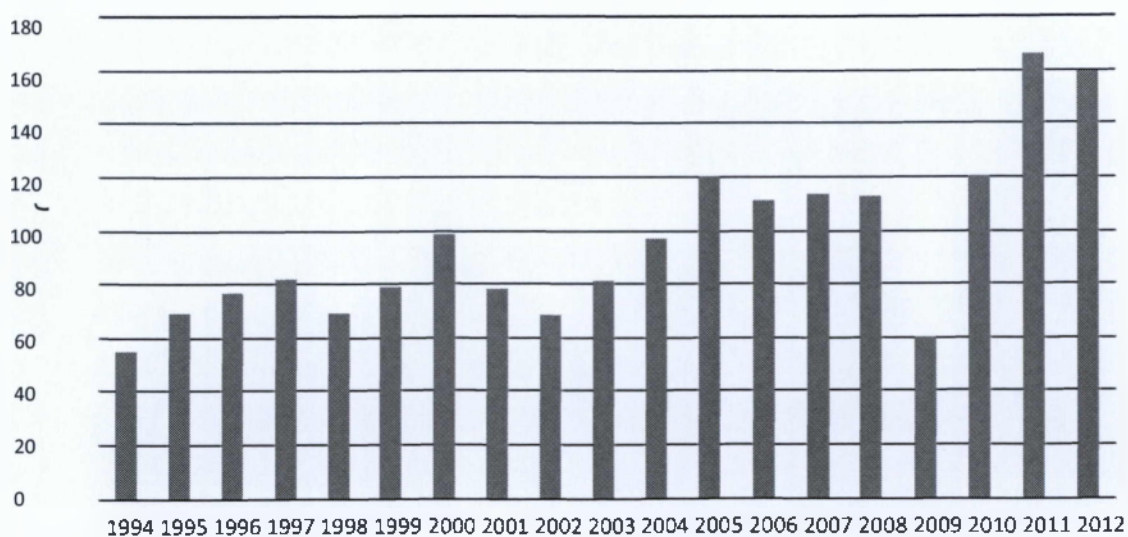
Τα τελευταία χρόνια η ιαπωνική κυβέρνηση έχει χρηματοδοτήσει μια πληθώρα ρομποτικών έργων, για παράδειγμα, 42 εκατομμύρια δολάρια για την πρώτη φάση ενός ανθρωποειδούς ρομποτικού έργου, και 10 εκατομμύρια δολάρια μεταξύ 2006 και 2010 για την ανάπτυξη βασικών ρομποτικών τεχνολογιών. Το 2005, υπήρχαν πάνω από 370.000 επιχειρησιακά βιομηχανικά ρομπότ στην Ιαπωνία, περίπου το 40% του παγκόσμιου συνόλου. Το 2007, ένας εθνικός τεχνολογικός χάρτης από το Υπουργείο Εμπορίου καλούσε για 1 εκατομμύριο βιομηχανικά ρομπότ εγκατεστημένα σε όλη τη χώρα μέχρι το 2025. (Tabuchi)

«Η ρομποτική βιομηχανία προχωρεί σε ένα λαμπρό μέλλον!» δήλωσε ο πρόεδρος της IFR, Δρ. Shinsuke Sakakibara, με την ευκαιρία της δημοσίευσης της μελέτης «World Robotics 2013 - Βιομηχανικά Ρομπότ», στις 18 Σεπτεμβρίου του 2013. «Το 2013, οι παγκόσμιες πωλήσεις ρομπότ θα αυξηθούν κατά περίπου 2%, δηλαδή 162.000 μονάδες. Η Στατιστική Υπηρεσία της IFR εκτιμά ότι μεταξύ 2014 και 2016 οι παγκόσμιες πωλήσεις ρομπότ θα αυξηθούν κατά περίπου 6% κατά μέσο όρο ετησίως. Το 2016, η ετήσια προμήθεια βιομηχανικών ρομπότ θα φτάσει πάνω από 190.000 μονάδες.» (2013-2016)

Η εκτιμώμενη ετήσια παγκόσμια προμήθεια βιομηχανικών ρομπότ (σε μονάδες):

Year	Supply	Year	Supply
1 9 9 8	69,000	2 0 0 6	1 1 2 , 0 0 0
1 9 9 9	79,000	2 0 0 7	1 1 4 , 0 0 0
2 0 0 0	99,000	2 0 0 8	1 1 3 , 0 0 0
2 0 0 1	78,000	2 0 0 9	6 0 , 0 0 0
2 0 0 2	69,000	2 0 1 0	1 2 0 , 5 8 5
2 0 0 3	81,000	2 0 1 1	1 6 6 , 0 2 8
2 0 0 4	97,000	2 0 1 2	1 8 0 , 9 5 0
2 0 0 5	120,000	2 0 1 3	2 0 7 , 5 0 0

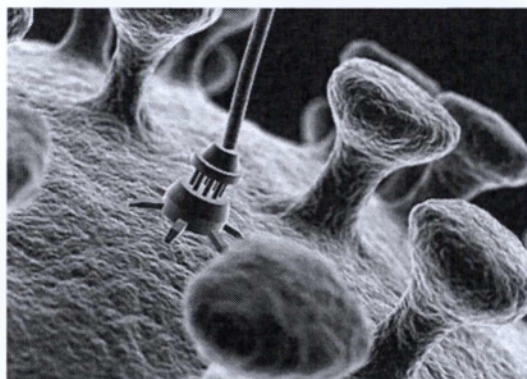
Η εκτιμώμενη ετήσια, παγκόσμια αποστολή των βιομηχανικών ρομπότ :



(vtu) (robots)

1.9 Νανομπότς - Νανορομπότς - Νανορομποτική

Η Νανορομποτική είναι ο αναδυόμενος τομέας της τεχνολογίας που δημιουργεί μηχανές ή ρομπότ των οποίων τα μέρη είναι ακριβώς ή κοντά στην κλίμακα του μανόμετρου (10^9 μέτρα). (EstOnline) (Whatis) Πιο συγκεκριμένα, η nanorobotics αναφέρεται στην πειθαρχία της μηχανικής νανοτεχνολογίας για το σχεδιασμό και τη δημιουργία νανορομπότ, με συσκευές που κυμαίνονται σε μέγεθος από 0.1 έως 10 μικρομέτρων και κατασκευασμένα από νανοκλίμακες ή από μοριακά εξαρτήματα. (Nano.gov) Τα ονόματα νανορομπότ, nanoids, nanites, νανομηχανές ή nanomites έχουν

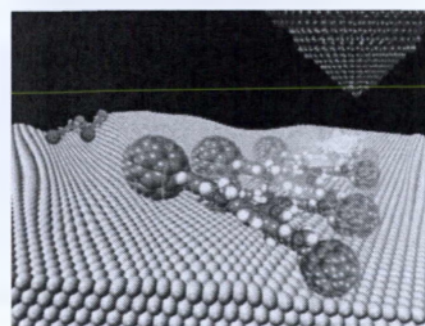


Εικ.93 Νανοκινητήρες τροφοδοτούμενοι από φωτόνια

επίσης χρησιμοποιηθεί για να περιγράψουν αυτές τις συσκευές, οι οποίες αυτή τη στιγμή είναι στο στάδιο της έρευνας και της ανάπτυξης. (innovetous)

Τα Νανορομπότ είναι σε μεγαλύτερο βαθμό στο στάδιο της έρευνας και ανάπτυξης, (SILBY), αλλά έχουν δοκιμαστεί και κάποιες

πρωτόγνωρες μοριακές μηχανές και νανορομπότ. Οι πρώτες χρήσιμες εφαρμογές των νανορομπότ είναι στην ιατρική τεχνολογία, όπου μια ομάδα επιστημών στη Νότια Κορέα από το Εθνικό και Καποδιστριακό Πανεπιστήμιο του Chonnam είναι οι πρώτοι που δημιούργησαν ένα nanorobot που μπορεί να θεραπεύσει τον καρκίνο. Η τεχνολογία, η οποία έχει ονομαστεί «Bacteriobot," είναι ένα γενετικά τροποποιημένο μη τοξικό βακτήριο σαλμονέλας, το οποίο έλκεται από χημικές ουσίες που απελευθερώνονται από τα καρκινικά κύτταρα. (Kubo) (Neal) Μία άλλη πιθανή εφαρμογή είναι η ανίχνευση των τοξικών χημικών ουσιών, και η μέτρηση των συγκεντρώσεών τους στο περιβάλλον. Το Rice University επιδείκνυε ένα μονομοριακό αυτοκίνητο που αναπτύχθηκε από μια χημική διεργασία και συμπεριλάμβανε buckyballs για τροχούς, το οποίο



Εικ.94 Μονομοριακό αυτοκίνητο, Rice University

ενεργοποιούταν με τον έλεγχο της περιβαλλοντικής θερμοκρασίας και με την τοποθέτηση της άκρης ενός σαρωτή σήραγγας (scanning tunneling) μικροσκοπίου.

Ένας άλλος ορισμός είναι ένα ρομπότ που επιτρέπει αλληλεπιδράσεις ακριβείας με αντικείμενα νανοκλίμακας, ή τον χειρισμό του με νανοκλιμακική ανάλυση. Τέτοιες συσκευές είναι πιο σχετικές με τη μικροσκοπία ή τη μικροσκοπία σάρωσης ανιχνευτή, αντί της περιγραφής των νανορομπότ ως μοριακά μηχανήματα. Ακλουθώντας τον ορισμό της μικροσκοπίας ακόμη και μια μεγάλη συσκευή όπως ένα μικροσκόπιο ατομικής δύναμης μπορεί να θεωρηθεί ένα μέσο νανορομποτικής (nanorobotic) όταν ρυθμιστεί ώστε να εκτελεί νανοχειρισμούς (nanomanipulation). Για την προοπτική αυτή, μακροκλίμακα ρομπότ ή μικρορομπότ (microrobots) που μπορούν να κινηθούν με νανοκλιμακική ακρίβεια μπορούν επίσης να θεωρηθούν νανορομπότ.

1.9.1 Νανορομποτική Θεωρία

Σύμφωνα με τον Richard Feynman , ο πρώην μεταπτυχιακός φοιτητής και συνεργάτης του Hibbs Albert ο οποίος αρχικά του πρότεινε (γύρω στο 1959) την ιδέα της ιατρικής χρήσης για τις θεωρητικές μικρομηχανές του Feynman (Feynman's theoretical micromachines). Ο Hibbs υποστήριξε ότι ορισμένες μηχανές επιδιόρθωσης μπορούν μια μέρα να μειωθούν σε μέγεθος στο σημείο που θα μπορούσε, θεωρητικά , να είναι ικανά να (όπως το έθεσε ο Feynman) «καταπιούν το γιατρό». Η ιδέα ενσωματώθηκε στο δοκίμιο του Feynman το 1959 «Υπάρχει αρκετός χώρος στο κάτω μέρος» . (IEEEghn) (Feynman)

Αφού τα νανορομπότ θα είναι μικροσκοπικά σε μέγεθος, θα ήταν ίσως αναγκαίο ένας πολύ μεγάλος αριθμός από αυτά να εργάζονται από κοινού για να εκτελούν μικροσκοπικά και μακροσκοπικά καθήκοντα. Τα σμήνη nanorobot, τόσο αυτά που ήταν ανάμεσα αναδιπλασιασμού (όπως στο «utility fog») όσο και εκείνων που ήταν ικανά για αβίαστη αναπαραγωγή στο φυσικό περιβάλλον (όπως στο «grey goo» και των λιγότερων κοινών παραλλαγών του), βρίσκονται σε πολλές ιστορίες επιστημονικής

φαντασίας, όπως τα Borg nanoprobes στο Star Trek και στο επεισόδιο «The New Breed» της σειράς «The Outer Limits».

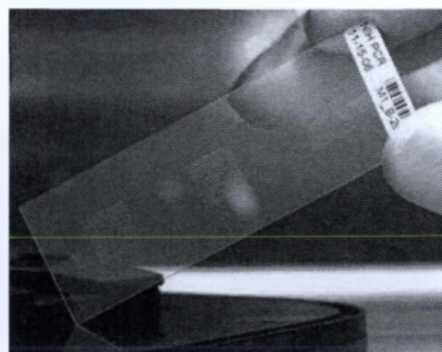
Μερικοί υποστηρικτές της νανορομποτικής (nanorobotics), ως αντίδραση στα τρομακτικά σενάρια του «grey goo», που νωρίτερα είχαν βοηθήσει στο να διαδοθεί, υποστήριζαν την ιδέα ότι τα νανορομπότ που είναι ικανά να πολλαπλασιαστούν έξω από ένα περιορισμένο εργοστασιακό περιβάλλον δεν αποτελούν απαραίτητο μέρος ενός δήθεν παραγωγικού τομέα της νανοτεχνολογίας, και ότι η διαδικασία της αυτοαντιγραφής, αν ήταν ποτέ να αναπτυχθεί, θα μπορούσαν να κατασκευαστούν εγγενώς ασφαλή. Επίσης ισχυρίζονται ότι τα τρέχοντα σχέδια τους για την ανάπτυξη και τη χρήση της μοριακής κατασκευής στην πραγματικότητα δεν περιλαμβάνουν «free-foraging replicators» (δωρεάν - τροφής αντιγραφείς). (M) (Nanoforum)

Η πιο λεπτομερής θεωρητική συζήτηση για τη νανορομποτική (nanorobotics), συμπεριλαμβανομένων των ειδικών θεμάτων σχεδιασμού, όπως η αναγνωριστική αίσθηση, η ενεργειακή επικοινωνία, η πλοήγηση, η χειραγώγηση, η μετακίνηση, και ο onboard υπολογισμός, έχει παρουσιαστεί σε ιατρικό συνέδριο της νανοϊατρικής από τον Robert Freitas. Μερικές από αυτές τις συζητήσεις παραμένουν στο επίπεδο των αδόμητων γενικοτήτων και δεν προσεγγίζουν το επίπεδο της λεπτομερής μηχανικής. (Jr.) (Zyvex)

1.9.2 Προσεγγίσεις της Νανορομποτικής

Βιοσίπ (Biochip)

Η κοινή χρήση της νανοηλεκτρονικής, της φωτολιθογραφίας και των νέων βιοϋλικών παρέχει μια πιθανή προσέγγιση για παρασκευή νανορομπότ για κοινές ιατρικές εφαρμογές, όπως για τη χειρουργική ενοργάνιση, τη διάγνωση και την παράδοση φαρμάκων. Αυτή η μέθοδος κατασκευής



Εικ.95 Τα Biochip περιέχουν πλέγματα, μικρά φρέατα ή "κουκίδες", καθένα από τα οποία περιέχουν πρωτεΐνες, αντισώματα ή νουκλεϊκά οξέα, που μπορούν να συνδεθούν σε ένα αντιγόνο ή μια αλληλουχία DNA

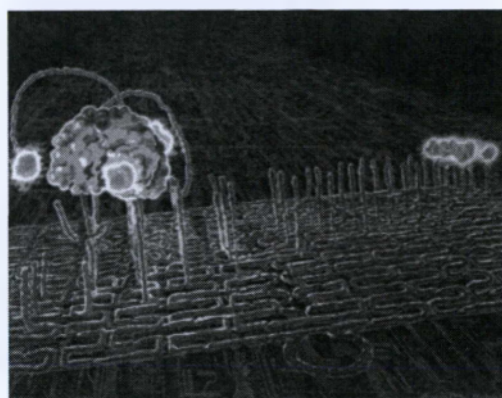
στην κλίμακα της νανοτεχνολογίας αυτή τη στιγμή χρησιμοποιείται στη βιομηχανία ηλεκτρονικών. Έτσι, πρακτικά τα νανορομπότ θα ενσωματωθούν με τις νανοηλεκτρονικές συσκευές, οι οποίες θα επιτρέψουν τηλε - λειτουργία και προηγμένες δυνατότητες για ιατρική ενοργάνιση. (Biochip)

Nubots

Το Nubot είναι η συντομογραφία για το "ρομπότ νουκλεϊκών οξέων". Τα Nubots είναι οργανικές μοριακές μηχανές στην νανοκλίμακα. Η δομή του DNA μπορεί να αποτελέσει μέσο για να συναρμολογηθούν 2D και 3D νανομηχανικές συσκευές. Οι μηχανές βασιζόμενες στο DNA μπορούν να ενεργοποιηθούν με τη χρήση μικρών μορίων, πρωτεϊνών και άλλων μορίων του DNA. (gnt) Βιολογικές πύλες κυκλωμάτων με βάση υλικών DNA έχουν κατασκευαστεί ως μοριακές μηχανές για να επιτρέψει την in-vitro απελευθέρωση φαρμάκου για στοχευμένα προβλήματα υγείας. Τέτοια συστήματα με βάση αυτά τα υλικά θα μπορούσαν να λειτουργήσουν πιο στενά με έξυπνα συστήματα διανομής βιοϋλικών φαρμάκων, ενώ δεν επιτρέπεται ακριβής in-vivo τηλεχειρισμός αυτών των σχεδιασμένων πρωτοτύπων. (Nubots)

Πλήρους Θέσης νανοσυνδεσμολογία (nanoassembly)

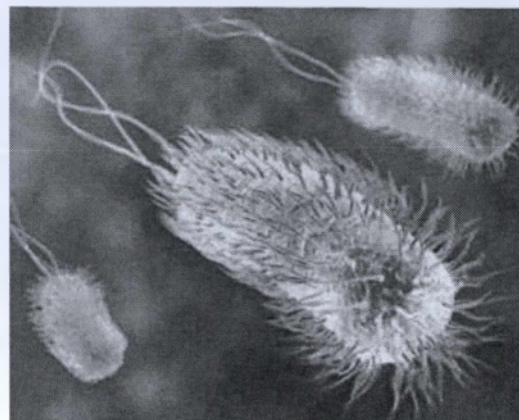
Η συνεργασία νανοεργοστασίων(Nanofactory Collaboration), ιδρύθηκε από τον Robert Freitas και τον Ralph Merkle το 2000 και με τη συμμετοχή 23 ερευνητών από 10 οργανισμούς και 4 χώρες, επικεντρώνεται στην ανάπτυξη μιας πρακτικής ερευνητικής ατζέντας, ειδικά για την ανάπτυξη ελεγχόμενης πλήρους θέσεως διαμαντοειδής μηχανοσύνδεσης (diamond mechanosynthesis) και για ένα αδαμαντοειδές νανοεργοστάσιο (diamondoid nanofactory) το οποίο θα έχει τη δυνατότητα της οικοδόμησης αδαμαντοειδών ιατρικών νανορομπότ (diamondoid medical nanorobots). (molecularassembler) (molecularassembler.c)



Εικ.96 Μοριακά ρομπότ σε γραμμές νανοσυναρμολόγησης, DNA περιπατητής κινείται κατά μήκος μιας σταθερής τροχιάς σε μια επιφάνεια.

Βασισμένα στα Βακτήρια

Η προσέγγιση αυτή προτείνει τη χρήση βιολογικών μικροοργανισμών , όπως τα βακτήριο *Escherichia coli* , τα οποία χρησιμοποιούν ένα μαστίγιο για προώθηση και τα ηλεκτρομαγνητικά πεδία ελέγχουν φυσιολογικά την κίνηση αυτού του είδους των βιολογικών ολοκληρωμένων συσκευών . (Whitesides)



Εικ.97 Whiplike Tails, νανορομπότ βασισμένο στο *Escherichia coli*

Ανοιχτή Τεχνολογία (Open Source)

Ένα έγγραφο με την πρόταση για νανοβοιτεχνολογική (nanobiotech) ανάπτυξη με Open Source τεχνολογίες έχει απευθυνθεί στην Γενική Συνέλευση του ΟΗΕ . Σύμφωνα με το έγγραφο που απέστειλε στον ΟΗΕ , με τον ίδιο τρόπο που το Open Source έχει τα τελευταία χρόνια επιταχύνει την ανάπτυξη των ηλεκτρονικών υπολογιστικών συστημάτων , μια παρόμοια προσέγγιση θα πρέπει να ωφελήσει την κοινωνία σε μεγάλο βαθμό και θα επιταχύνει την ανάπτυξη της νανορομποτικής (nanorobotics) . Η χρήση της νανοβοιτεχνολογίας θα πρέπει να καθιερωθεί ως μια ανθρώπινη κληρονομιά για τις επόμενες γενιές , και να αναπτυχθεί ως μια ανοιχτή (Open Source) τεχνολογία που βασίζεται σε ηθικές πρακτικές για ειρηνικούς σκοπούς . (nanotech-now) (UNO)

1.9.3 Αγώνας των νανορομπότ

Με τον ίδιο τρόπο που η ανάπτυξη της τεχνολογίας είχε την κούρσα του διαστήματος (space race) και την κούρσα των πυρηνικών εξοπλισμών (nuclear arms race) , έτσι επέρχεται και μια κούρσα για τα νανορομπότ. Υπάρχει αρκετό έδαφος που επιτρέπει τα νανορομπότ να συμπεριλαμβάνονται μεταξύ των αναδυόμενων τεχνολογιών. Μερικοί από τους λόγους είναι ότι, οι μεγάλες εταιρείες όπως η General Electric , η Hewlett - Packard και η Northrop Grumman έχουν αρχίσει να εργάζονται

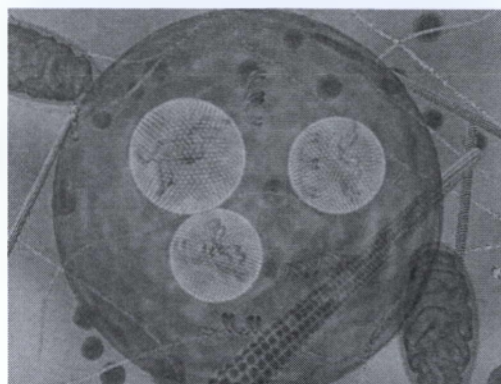
στην ανάπτυξη και την έρευνα των νανορομπότ, (grumman) οι χειρουργοί αρχίζουν να συμμετάσχουν και ξεκινούν να προτείνουν τρόπους για την εφαρμογή νανορομπότ για κοινές ιατρικές διαδικασίες, (Templeton) τα πανεπιστήμια και τα ερευνητικά ιδρύματα χορηγήθηκαν κονδύλια από τους κρατικούς φορείς που υπερβαίνουν τα 2 δισεκατομμύρια δολάρια για έρευνες ανάπτυξης νανοσυσκευών για την ιατρική. Επίσης οι τραπεζίτες έχουν επενδύσει στρατηγικά με την πρόθεση να αποκτήσουν προτέρων τα δικαιώματα σχετικά με τη μελλοντική εμπορευματοποίηση των νανορομπότ. Ορισμένες πτυχές των νανορομποτικών (nanorobot) δικαστικών αγώνων και τα συναφή θέματα που συνδέονται με το μονοπώλιο έχουν ήδη προκύψει. (lawbc) Ένας μεγάλος αριθμός των διπλωμάτων ευρεσιτεχνίας έχουν δοθεί πρόσφατα για τα νανορομπότ, το οποίο γίνεται ως επί το πλείστον για τους παράγοντες των διπλωμάτων ευρεσιτεχνίας, και οι εταιρείες ειδικεύονται αποκλειστικά στην οικοδόμηση του χαρτοφυλακίου διπλωμάτων ευρεσιτεχνίας, και των δικηγόρων. Μετά από μια μακριά σειρά των διπλωμάτων ευρεσιτεχνίας και δικαστικών αγώνων, για παράδειγμα, η εφεύρεση του ασυρμάτου ή ο πόλεμος των ρευμάτων , αναδυόμενοι τομείς της τεχνολογίας τείνουν να γίνονται μονοπώλιο , το οποίο συνήθως κυριαρχείται από μεγάλες εταιρείες. (EtcGroup) (Corporatwatch)

1.9.4 Πιθανές εφαρμογές στην νανοϊατρική

Πιθανές εφαρμογές για τη νανορομποτική (nanorobotics) στην ιατρική περιλαμβάνουν την έγκαιρη διάγνωση και τη στοχευμένη διανομή φαρμάκων για τον καρκίνο, (N.cancer) τα βιοϊατρικά όργανα, τη χειρουργική επέμβαση, τη φαρμακοκινητική, τη παρακολούθηση του διαβήτη, (N.dia.) και την υγειονομική περίθαλψη. (N.ra)

Στα εν λόγω σχέδια, η μελλοντική ιατρική νανοτεχνολογία αναμένεται να χρησιμοποιεί νανορομπότ που θα εγχέονται στον ασθενή και θα εκτελούν εργασίες σε κυτταρικό επίπεδο. Τέτοια νανορομπότ που προορίζονται για ιατρική χρήση θα πρέπει να είναι μη αντιγραφόμενα, καθώς η αντιγραφή θα αύξανε χωρίς λόγο την πολυπλοκότητα του μηχανισμού, θα μείωνε την αξιοπιστία και θα παρέμβαινε με το ιατρικό καθήκον.

Η νανοτεχνολογία προσφέρει ένα ευρύ φάσμα νέων τεχνολογιών για την ανάπτυξη εξατομικευμένων λύσεων που βελτιστοποιούν την παράδοση των φαρμακευτικών προϊόντων. Σήμερα, επιβλαβείς παρενέργειες των θεραπειών όπως η χημειοθεραπεία είναι συνήθως αποτέλεσμα των μεθόδων χορήγησης φαρμάκων που δεν εντοπίζουν τα κύτταρα για τα οποία προορίζονται με ακρίβεια. Ερευνητές στο Harvard και το MIT, ωστόσο, ήταν σε θέση να συνδέουν ειδικούς κλώνους RNA , σχεδόν 10 nm σε διάμετρο, σε νανο-σωματίδια και τα γέμιζαν με ένα φάρμακο χημειοθεραπείας.



Εικ.98 Νανοσωματίδια λιπιδίου (που φέρουν siRNA), εμφανίζονται καθώς μεταφέρονται μέσα σε κύτταρα, χρησιμοποιώντας ενδοκυτταρικά κυστίδια.

Αυτοί οι κλώνοι RNA έλκονται από τα καρκινικά κύτταρα και όταν τα νανοσωματίδια συναντούν ένα καρκινικό κύτταρο , προσκολλάται σε αυτό, και απελευθερώνουν το φάρμακο εντός του καρκινικού κυττάρου. Η κατευθυνόμενη μέθοδος παροχής φαρμάκου έχει μεγάλες δυνατότητες για τη θεραπεία ασθενών με καρκίνο , αποφεύγοντας παράλληλα αρνητικές επιπτώσεις (που συνήθως συνδέονται με ακατάλληλη παροχή του φαρμάκου). (N.C.mit-harvard)

Μία άλλη χρήσιμη εφαρμογή των νανορομποτ είναι ότι βοηθούν στην επισκευή των κυτταρικών ιστών, παράλληλα με τα λευκά αιμοσφαίρια. Η πρόσληψη των φλεγμονωδών κυττάρων ή των λευκών αιμοσφαιρίων (που περιλαμβάνουν ουδετερόφιλα, λεμφοκύτταρα, μονοκύτταρα και ιστοκύτταρα) προς την πληγείσα περιοχή είναι η πρώτη απάντηση των ιστών στον τραυματισμό. Λόγω του μικρού μεγέθους των νανορομποτ θα μπορούσαν να προσκολληθούν στην επιφάνεια των λευκών αιμοσφαιρίων που προσλαμβάνονται, για να αποσπάσουν την έξοδο τους μέσα από τα τοιχώματα των αιμοφόρων αγγείων και να φτάνουν στο σημείο του τραυματισμού, όπου μπορούν να βοηθήσουν στην επισκευαστική διαδικασία του ιστού. Ορισμένες ουσίες θα μπορούσαν ενδεχομένως να χρησιμοποιηθούν και για την επιτάχυνση της ανάκαμψης.

Η επιστήμη πίσω από αυτό το μηχανισμό είναι αρκετά περίπλοκη . Το πέρασμα των κυττάρων σε όλη την ενδοθήλιο του αίματος , μια διαδικασία γνωστή ως μετανάστευση , είναι ένας μηχανισμός που περιλαμβάνει την σύμπλεξη των

αισθητήριων νεύρων της κυτταρικής επιφανείας σε μόρια προσκόλλησης, την ενεργό άσκηση δύναμης και διαστολής των αγγειακών τοιχωμάτων, και τη φυσική παραμόρφωση των κυττάρων που μεταναστεύουν. Με την ένωση τους με τα μεταναστεύων φλεγμονώδη κύτταρα, τα ρομπότ στην πραγματικότητα αποκτούν «μια δωρεάν βόλτα» σε όλα τα αιμοφόρα αγγεία, παρακάμπτοντας την ανάγκη τους για ένα πολύπλοκο μηχανισμό μετανάστευσης. (N.RtissueC.)

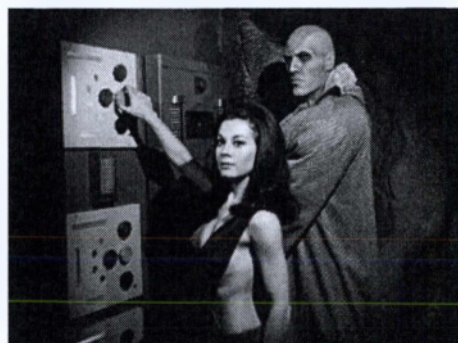
1.10 Ανδροειδές (Android)

Το Ανδροειδές (Android) είναι ένα ρομπότ ή ένας συνθετικός οργανισμός ,που έχει σχεδιαστεί για να μοιάζει και να λειτουργεί σαν έναν άνθρωπο, ειδικά όταν έχει σώμα το οποίο μοιάζει με ζωντανή σάρκα. Μέχρι πρόσφατα, τα ανδροειδή έχουν σε μεγάλο βαθμό παραμείνει στον χώρο της επιστημονικής φαντασίας, τα όποια εμφανίζονται συχνά στον κινηματογράφο και στην τηλεόραση. Ωστόσο, οι εξελίξεις στην ρομποτική τεχνολογία έχουν επιτρέψει το σχεδιασμό λειτουργικών και ρεαλιστικών ανθρωποειδών ρομπότ. (A.def)

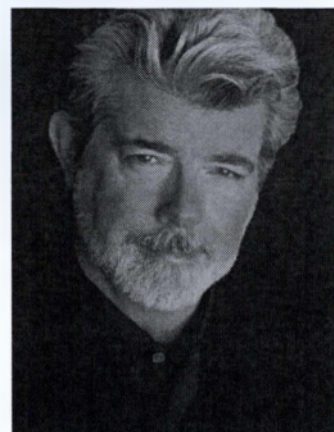
Ετυμολογία

Η λέξη επινοήθηκε από την ελληνική ρίζα άνδρ- «άνθρωπος» και την κατάληξη oid- «έχει τη μορφή ή την ομοιότητα του».

Ο όρος «droid» , επινοήθηκε από τον George Lucas για την πρωτότυπη ταινία Star Wars και τώρα χρησιμοποιείται ευρέως στην επιστημονική φαντασία, προερχόμενο ως μια επιτομή του "Android", αλλά έχει χρησιμοποιηθεί από τον Lucas και από πολλούς άλλους για να σημαίνει οποιοδήποτε ρομπότ , συμπεριλαμβανόμενου ευδιάκριτων μη - ανθρώπινων μορφών μηχανημάτων όπως ο R2 - D2 . Η λέξη "Android "



Εικ.99 Star Wars - What Are Little Girls Made Of? (Τηλ. Σειρά)



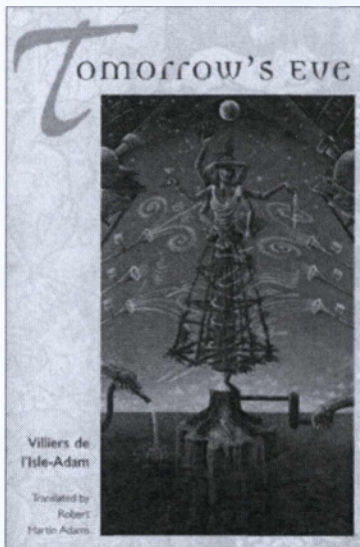
Εικ.100 George Lucas

χρησιμοποιήθηκε στο επεισόδιο «What Are Little Girls Made Of?»

της σειράς Star Trek : The Original Series . Η συντομογραφία " Andy " , επινοήθηκε ως μια υποτιμητική λέξη από το συγγραφέα Philip K. Dick στο μυθιστόρημά του «Do Androids Dream of Electric Sheep ?» , η οποία έχει κάποια περαιτέρω

χρήση , όπως στο πλαίσιο της τηλεοπτικής σειράς «Total Recall 2070». (And.pro)

Το αγγλικό λεξικό της Οξφόρδης περιγράφει την πρώτη χρήση (ως " androides ") στην εγκυκλοπαίδεια (Cyclopaedia ή A Universal Dictionary of Arts and Sciences) του Ephraim Chamber, στην αναφορά ενός αυτόματου που ο καθολικός Άγιος Albertus Magnus , σύμφωνα με τους ισχυρισμούς, δημιούργησε. (O android) Ο όρος "Android " εμφανίζεται στα διπλώματα ευρεσιτεχνίας των ΗΠΑ ήδη από το 1863 στην αναφορά για τις μινιατούρες των ανθρωπόμορφων αυτομάτων παιχνιδιών, και χρησιμοποιήθηκε σε μια πιο μοντέρνα αίσθηση από το γαλλικό συγγραφέα Auguste Villiers de l' Isle -Adam στο έργο του «Tomorrow's Eve» (1886). Η ιστορία χαρακτηρίζει ένα τεχνητό



Εικ.101 Tomorrow's Eve από τον Auguste de Villiers de l'Isle-Adam

ανθρωπόμορφο ρομπότ που ονομάζεται Hadaly , και όπως είπε ο αξιωματικός στην ιστορία , «Σε αυτή την εποχή της Realien (περίοδος της πραγματικότητας) προόδου , ποιος ξέρει τι συμβαίνει στο μυαλό των υπευθύνων για αυτές τις μηχανικές κούκλες» ("In this age of Realien advancement, who knows what goes on in the mind of those responsible for these mechanical dolls."). Ο όρος είχε αντίκτυπο στα αγγλικά περιοδικά επιστημονικής φαντασίας «Pulp» ξεκινώντας από το The Cometeers του Jack Williamson (1936) και η διάκριση μεταξύ των μηχανικών ρομπότ και των σαρκώδη ανδροειδών διαδόθηκε από το Future

Captain του Edmond Hamilton (1940-1944). (And.pro)

Παρά το γεγονός ότι τα ρομπότ του Karel Čapek στο R.U.R. (Rossum's Universal Robots) (1921) ήταν οργανικά τεχνητοί άνθρωποι , η λέξη "ρομπότ" κατά κύριο λόγο αναφερόταν σε μηχανικούς ανθρώπους, ζώα και άλλα όντα. Ο όρος "Android " μπορεί να σημαίνει είτε ένα από αυτά, ενώ ένα cyborg (κυβερνητικός οργανισμός ή βιονικός άνθρωπος) θα ήταν ένα πλάσμα που είναι ένας συνδυασμός βιολογικών και μηχανικών τμημάτων .

Οι συγγραφείς έχουν χρησιμοποιήσει τον όρο android σε περισσότερους διαφορετικούς τρόπους απ' ό,τι το ρομπότ ή το cyborg . Σε ορισμένα έργα φαντασίας, η διαφορά ανάμεσα σε ένα ρομπότ και ένα Android είναι μόνο η εμφάνισή τους, με τα ανδροειδή να φτιάχνονται για να μοιάζουν με ανθρώπους στο εξωτερικό αλλά να μοιάζουν με ρομπότ στο εσωτερικό τους. Σε άλλες ιστορίες, οι συγγραφείς έχουν

χρησιμοποιήσει τη λέξη "Android" να σημαίνει μια πλήρους οργανική, αλλά και τεχνητή, δημιουργία. Επίσης άλλες φανταστικές απεικονίσεις των androids εμπίπτουν κάπου στο ενδιάμεσο.

Ο Eric G. Wilson, ο οποίος ορίζει τα ανδροειδή ως ένα «συνθετικό ανθρώπινο ον», διακρίνει τρεις τύπους ανδροειδών, με βάση τη σύνθεση του σώματός τους :

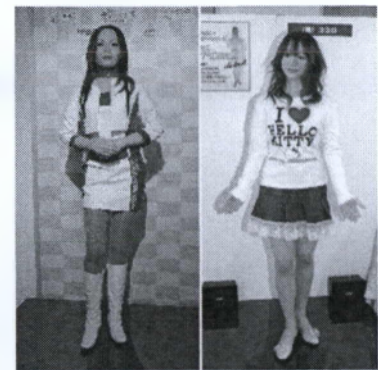
- Ο τύπος Μούμια - όπου τα ανδροειδή είναι κατασκευασμένα από «νεκρά πράγματα» ή «δύσκαμπτα, άψυχα, φυσικά υλικά», όπως είναι οι μούμιες, οι μαριονέτες, οι κούκλες και τα αγάλματα.
- Ο τύπος Golem - ανδροειδή κατασκευασμένη από εύκαμπτα, ενδεχομένως οργανικά υλικά, συμπεριλαμβανομένων των golems και των homunculi.
- Ο τύπος Αυτόματο - ανδροειδή που είναι ένα μείγμα νεκρών και ζωντανών τμημάτων, συμπεριλαμβανομένων των αυτομάτων και των ρομπότ.

Αν και η ανθρώπινη μορφολογία δεν είναι απαραίτητα η ιδανική μορφή για τα λειτουργικά ρομπότ, η γοητεία στην ανάπτυξη κάποιων ρομπότ που μπορούν να μιμούνται μπορεί να βρεθεί ιστορικά στην αφομοίωση των δύο εννοιών: simulacra (συσκευές που παρουσιάζουν ομοιότητα) και αυτομάτων-automata (συσκευές που έχουν ανεξαρτησία-αυτοτέλεια). (And.pro)

1.10.1 Τα ανδροειδή στην Ιαπωνία

Το Intelligent Robotics Lab, με διευθυντή τον Hiroshi Ishiguro στο Osaka University, και η Kokoro Co-Ltd, έχουν κάνει επίδειξη της Actroid στην Expo 2005 στο Aichi Prefecture της Ιαπωνίας. Το 2006 η Kokoro Co αναπτύσσει ένα νέο DER-2 ανδροειδές (android), όπου το ύψος του ανθρώπινου σώματος του είναι 165 εκατοστά και υπάρχουν σε όλο του το σώμα 47 κινητές μονάδες. Το DER2 δεν μπορεί να αλλάξει μόνο την έκφραση του, αλλά και να κινήσει τα χέρια και τα πόδια του και να στρίψει το

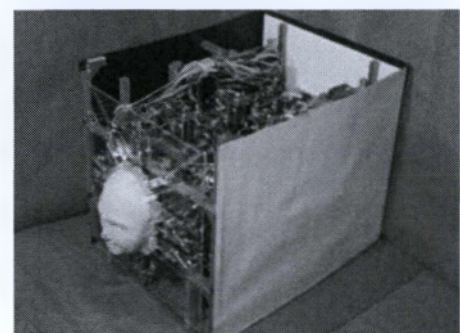
σώμα του. Το «σερβοσύστημα αέρα» ("air servosystem") που ανέπτυξε η Kokoro Co αρχικά χρησιμοποιούταν για τον ενεργοποιητή. Ως αποτέλεσμα της ύπαρξης ενός ενεργοποιητή που ελέγχεται επακριβώς με την πίεση του αέρα μέσω ενός σερβοσυστήματος (servosystem), η κίνηση είναι πολύ πιο ρευστή και υπάρχει πολύ λίγος θόρυβος. Το DER2 πραγματοποίησε ένα πιο λεπτό σώμα από εκείνο της προηγούμενης έκδοσης, χρησιμοποιώντας ένα μικρότερο κύλινδρο. Εξωτερικά το DER2 έχει πιο όμορφες αναλογίες, και σε σύγκριση με το προηγούμενο μοντέλο, έχει λεπτότερους βραχίονες και ένα ευρύτερο ρεπερτόριο εκφράσεων. Επίσης μόλις προγραμματιστεί, είναι σε θέση να χορογραφήσει κινήσεις και χειρονομίες με τη φωνή του. (DER-02)



Εικ.102 Ιαπωνικά ανδρoειδές DER 01 και DER 02

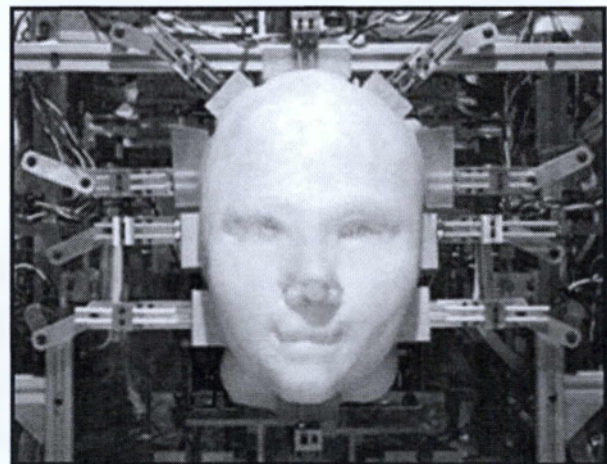
Το Intelligent Mechatronics Lab έχει αναπτύξει ένα ανδρoειδές κεφάλι (android head) που ονομάζεται Saya, το οποίο εκτέθηκε στο Robodex 2002 στη Γιοκοχάμα της Ιαπωνίας. Υπάρχουν διάφορες άλλες πρωτοβουλίες σε όλο τον κόσμο που αφορούν την έρευνα και την ανάπτυξη ανθρωποειδών αυτή τη στιγμή, οι οποίες ελπίζουμε ότι θα εισάγουν ένα ευρύτερο φάσμα της συνειδητοποιημένης τεχνολογίας στο κοντινό μέλλον. Τώρα η Saya εργάζεται στο Πανεπιστήμιο του Τόκιο ως ξεναγός. (R.H.saya)

Το Waseda University (Ιαπωνία) και οι κατασκευαστές της NTT DoCoMo έχουν καταφέρει να δημιουργήσουν ένα shape-shifting ρομπότ με όνομα WD-2 το οποίο είναι σε θέση να αλλάζει το πρόσωπό του. Στην αρχή, οι δημιουργοί αποφάσισαν τις θέσεις των σημείων που είναι απαραίτητα για να εκφράσουν το περίγραμμα, τα μάτια, τη μύτη και ούτω καθεξής από ένα συγκεκριμένο πρόσωπο. Το ρομπότ εκφράζει το πρόσωπο του με την κίνηση όλων των σημείων στις θέσεις απόφασης. Η πρώτη έκδοση του ρομπότ αναπτύχθηκε για πρώτη φορά το 2003 και μετά από ένα χρόνο έκαναν μια-δυο σημαντικές βελτιώσεις στο σχεδιασμό. Το ρομπότ διαθέτει μια ελαστική μάσκα φτιαγμένη από το μέσο ομοίωμα της



Εικ.103 WD-2, ανδρoειδές ρομπότ

κεφαλής και χρησιμοποιεί ένα σύστημα οδήγησης με μια 3DOF μονάδα. Το WD-2 μπορεί να αλλάξει τα χαρακτηριστικά του προσώπου του ενεργοποιώντας συγκεκριμένα σημεία του προσώπου πάνω στη μάσκα και κάθε στοιχείο κατέχει τρεις βαθμούς ελευθερίας, με ένα σύνολο 17 σημείων του προσώπου και 56 βαθμών ελευθερίας. Όσο για τα υλικά που χρησιμοποιούνται, η μάσκα του WD-2 είναι κατασκευασμένη από ένα εξαιρετικά ελαστικό υλικό που ονομάζεται Septom, αναμιγνύοντας μαζί με κομμάτια από μαλλί χάλυβα (steel wool) για μεγαλύτερη αντοχή. Άλλα τεχνικά χαρακτηριστικά αποκαλύπτουν έναν άξονα που κινείται πίσω από τη μάσκα στο επιθυμητό σημείο του προσώπου, κινούμενος από έναν κινητήρα συνεχούς ρεύματος μαζί με μία απλή τροχαλία και μια βίδα slide. Προφανώς, οι ερευνητές μπορούν επίσης να τροποποιήσουν το σχήμα της μάσκας με βάση πραγματικά ανθρώπινα πρόσωπα. Για την "αντιγραφή" ενός προσώπου, το μόνο που χρειάζεται είναι ένας 3D σαρωτής (3D scanner) για να καθορίσει τις θέσεις των 17 σημείων του προσώπου ενός ατόμου. Μετά από αυτό οδηγούνται σε καθορισμένες θέσεις χρησιμοποιώντας ένα φορητό υπολογιστή και 56 πίνακες ελέγχου του κινητήρα. Επιπλέον, οι ερευνητές αναφέρουν ότι το μεταλλασσόμενο ρομπότ μπορεί να εμφανίσει ακόμη και το στυλ των μαλλιών ενός ατόμου και το χρώμα του δέρματος, αν μια φωτογραφία του προσώπου τους προβληθεί στην 3D μάσκα. (WD-2)



Εικ.104 Ρομποτικό πρόσωπο WD-2 (κοντινό πλάνο)

1.10.2 Τα ανδροειδή στην Κορέα

Στο KITECH ερεύνησαν και ανέπτυξαν το EveR-1, ένα ανδροειδές (Android) διαπροσωπικής επικοινωνίας ικανό να μιμηθεί εκφράσεις των ανθρώπινων συναισθημάτων μέσω του «μυϊκού συστήματος» του προσώπου και ικανά για μια



Εικ. 105 EveR-1, Κορεάτικο ανδροειδές

στοιχειώδη συζήτηση, έχοντας ένα λεξιλόγιο περίπου 400 λέξεων. Έχει ύψος 160 εκατοστά και ζυγίζει 50 κιλά, ταιριάζοντας με το μέσο ποσοστό μιας 20χρονης κορεατικής γυναίκας. Το όνομα EveR-1 προέρχεται από τη Βιβλική Εύα, καθώς και το γράμμα R για το ρομπότ. Η προηγμένη επεξεργαστική ισχύ που κατέχει της επιτρέπει την αναγνώριση φωνής και τη φωνητική σύνθεση, την ίδια στιγμή που επεξεργάζεται το συγχρονισμό χειλιών (lip synchronization) και την οπτική αναγνώριση με τη χρήση των 90° μοιρών μικρο - CCD καμερών με την τεχνολογία αναγνώρισης προσώπου. Επίσης ένα ανεξάρτητο μικροτσίπ στο εσωτερικό του τεχνητού εγκέφαλου της χειρίζεται την έκφραση χειρονομίας, το συντονισμό του σώματος και την έκφραση συναισθημάτων. Ολόκληρο το σώμα της είναι κατασκευασμένο από εξαιρετικά προηγμένης συνθετικής ζελέ σιλικόνης (jelly silicon) και με 60 τεχνητές αρθρώσεις στο πρόσωπο, το λαιμό και στο κάτω μέρος του σώματος, είναι σε θέση να επιδείξει ρεαλιστικές εκφράσεις του προσώπου και να τραγουδάει ενώ ταυτόχρονα να χορεύει. (EveR-1)

Στη Νότια Κορέα, το Υπουργείο Πληροφοριών και Επικοινωνιών έχει ένα φιλόδοξο σχέδιο, να τεθεί ένα ρομπότ σε κάθε νοικοκυριό έως το 2020. Πολλές πόλεις ρομπότ έχουν προγραμματιστεί για τη χώρα : Το πρώτο θεματικό πάρκο κατασκευάζεται είδη και θα τεθεί στο κοινό το 2015- 2016 με κόστος από 728 δισ. , εκ των οποίων τα 59,5 δισ. ευρώ είναι άμεσες δημόσιες επενδύσεις. Η νέα πόλη ρομπότ θα διαθέτει κέντρα ρομποτικής έρευνας και ανάπτυξης , υψηλής ποιότητας εμπορικούς δρόμους που σχετίζονται με το θεματικό πάρκο, φιλικό προς το περιβάλλον ξενοδοχείο και διοικητικών γραφείων, καθώς και εκθεσιακούς χώρους. Ο νέος Χάρτης Ρομποτικής Δεοντολογίας (Robot Ethics Charter 2012) της χώρας έχει θεσπιστεί από το 2012 και

καταρτίστηκε με σκοπό να αποφευχθούν τα κοινωνικά δεινά που μπορεί να προκύψουν από την ανεπάρκεια των κοινωνικών και νομικών μέτρων για την αντιμετώπιση των ρομπότ στην κοινωνία. Οι κανόνες και οι νόμοι που έχουν θεσπιστεί αφορούν τα πρότυπα της ρομποτικής βιομηχανίας, τα δικαιώματα και τις ευθύνες των χρηστών /ιδιοκτητών και τα δικαιώματα και τις υποχρεώσεις των ρομπότ, έτσι ώστε να προληφθεί η κατάχρηση των ρομπότ από τους ανθρώπους και το αντίστροφο. (Robotland)

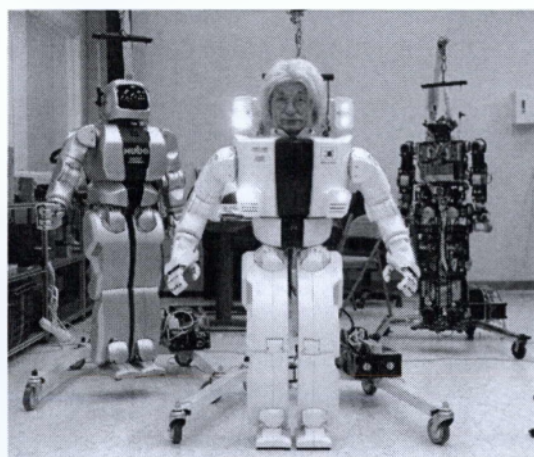


Εικ.106 Robotland, Θεματικό Πάρκο

1.10.3 Τα ανδρoειδή στις Ηνωμένες Πολιτείες

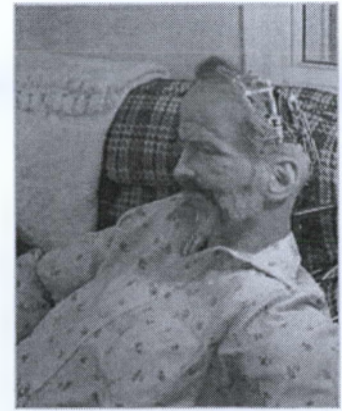
Ο Walt Disney και ένα προσωπικό από Imagineers δημιούργησαν τις «Μεγάλες Στιγμές με τον κ. Λίνκολν» (Great Moments with Mr. Lincoln), που έκανε το ντεμπούτο του το 1964 στη Διεθνή Έκθεση της Νέας Υόρκης (New York World's Fair). Στο οποίο έκανε την εμφάνιση στη σκηνή μια ακουστικό-ηλεκτρονική έκδοση του Προέδρου των ΗΠΑ Αβραάμ Λίνκολν. (R.dis.)

Η Hanson Robotics Inc. , του Τέξας και η KAIST παρουσίασαν ένα ανδρoειδές (Android) πορτρέτο του Άλμπερτ Αϊνστάιν, χρησιμοποιώντας την τεχνολογία προσώπου της Hanson και τοποθετημένο στο πραγματικό μέγεθος δίποδου ρομπότ της KAIST. Αυτό το ανδρoειδές του Einstein, που ονομάζεται επίσης και “Albert Hubo”, αποτελεί, συνεπώς, το πρώτο στην ιστορία ανδρoειδές ολόκληρου σώματος το οποίο περιπατάει. (Alb.Hubo)



Εικ.107 Albert Hubo, Albert Einstein ανδρoειδές πορτρέτο

Η Hanson Robotics, η FedEx Institute of Technology, και το Πανεπιστήμιο του Τέξας στο Arlington επίσης ανέπτυξε το ανδροειδές πορτρέτο του συγγραφέα επιστημονικής φαντασίας Philip K. Dick (δημιουργός του: Do Androids Dream of Electric Sheep, η βάση για την ταινία Blade Runner) , με πλήρες ομιλητικές δυνατότητες που ενσωματώνουν χιλιάδες σελίδες των έργων του συγγραφέα. Στις 5, Ιουνίου του 2012 ,ένα βιβλίο για ανδροειδές έργο του Philip K Dick είχε κυκλοφορήσει από τον Henry Holt. Έχει τον πιασάρικο τίτλο «Πώς να κατασκευάσουμε ένα ανδροειδές: Η αληθινή ιστορία της Ρομποτικής Ανάσταση του Philip K. Dick» (How to Build an Android: The True Story of Philip K. Dick's Robotic Resurrection). Πρόκειται για μια λεπτομερή και τεκμηριωμένη ανάλυση του Philip K Dick ανδροειδες έργου και συλλαμβάνει τα πυρετώδη γεγονότα γύρω από την κατασκευή του ανδροειδες, καθώς και της μυστηριώδης και ατυχής εξαφάνιση του στα τέλη του 2005. Επίσης το 2005 , το PKD (Philip K. Dick) ανδροειδες είχε κερδίσει το πρώτο βραβείο τεχνητής νοημοσύνης από τον AAAI. (P.k.Dick)



Εικ.108 Ανδροειδές του Philip K. Dick

1.10.4 Τα ανδροειδή στο Ηνωμένο Βασίλειο

Το 2001, ο Steve Grand, δημιουργός του παιχνιδιού «Creatures» στον υπολογιστή, δημιούργησε ένα ανδροειδές ή ανθρωποειδείς, και το ονόμασε Lucy . Η πρόθεση ήταν ότι θα πρέπει να μάθει τα πάντα, συμπεριλαμβανομένου και πώς να χρησιμοποιεί τις μηχανικές φωνητικές χορδές της για να μιλήσει. Τα συστήματα της δημιουργήθηκαν έτσι ώστε να είναι παρόμοια με έναν άνθρωπο. (R.Lucy)

1.11 Cyborg

Ένα cyborg (Σάμποργκ), συντομογραφία για το «κυβερνητικός οργανισμός» ("cybernetic organism"), είναι ένα ον με τόσο οργανικά όσο και τεχνητά μέρη. Ο όρος επινοήθηκε το 1960, όταν ο Manfred Clynes και ο Nathan S. Kline τον χρησιμοποίησαν σε ένα άρθρο σχετικά με τα πλεονεκτήματα των αυτορυθμιζόμενων ανθρωπομηχανικών συστημάτων στο διάστημα. Το σύγγραμμα του D.S. Halacy το 1965 «Cyborg: Evolution of the Superman» (Cyborg : Η εξέλιξη του Superman), χαρακτήρισε μια εισαγωγή η οποία μίλησε για ένα «νέο σύνορο» που είναι «όχι μόνο χώρος, αλλά και πιο βαθιά η σχέση μεταξύ του εσωτερικού χώρου και του «εξωτερικού χώρου, μια γέφυρα μεταξύ νου και της ύλης». (C.def.)

Η αρχή της δημιουργίας των Cyborg άρχισε όταν η HCI (human-computer interaction, αλληλεπίδραση ανθρώπου-υπολογιστή) ξεκίνησε την λειτουργία της. Υπάρχει μια σαφής διάκριση μεταξύ της ανθρώπινης και της μηχανικής τεχνολογίας στην HCI, η οποία διαφέρει από τα cyborgs σε αυτές τις cyborgs ενέργειες έξω από τις ανθρώπινες λειτουργίες. (HCI)

Ο όρος cyborg εφαρμόζεται συχνά σε έναν οργανισμό που έχουν ενισχυθεί οι ικανότητες του, χάρη στην τεχνολογία, αν και αυτό ίσως να υπεραπλουστεύει την αναγκαιότητα ανατροφοδότησης για τη ρύθμιση του υποσυστήματος. Ο πιο αυστηρός ορισμός του Cyborg σχεδόν πάντα θεωρείται ως η ενίσχυση ή η βελτίωση των φυσιολογικών δυνατοτήτων. Επίσης θα μπορούσαν θεωρητικά να είναι οποιοδήποτε είδος οργανισμού και ο όρος "Cybernetic organism" (« κυβερνητικός οργανισμός ») έχει εφαρμοστεί ακόμα και σε δίκτυα, όπως οδικά συστήματα, εταιρείες και κυβερνήσεις, οι οποίες έχουν χαρακτηριστεί ως τέτοια δίκτυα. Ο όρος μπορεί επίσης να εφαρμοστεί σε μικρό-οργανισμούς που έχουν τροποποιηθεί για να λειτουργούν σε υψηλότερα επίπεδα από τα μη τροποποιημένα πανομοιότυπα τους. Υποτίθεται ότι η τεχνολογία cyborg θα αποτελέσει μέρος της μελλοντικής εξέλιξης του ανθρώπου.

Τα μυθοπλαστικά cyborgs

παρουσιάζονται ως σύνθεση οργανικών και συνθετικών τμημάτων, και συχνά θέτουν το ζήτημα της διαφοράς μεταξύ ανθρώπου και μηχανής, καθώς κάποιοι ασχολούνται με την ηθική, την ελεύθερη βούληση, και την ενσυναίσθηση. Τα μυθοπλαστικά cyborgs



Εικ. 109 Terminator 2 (ταινία)

μπορεί να παρουσιαστούν ως ορατά μηχανήματα (π.χ. το Cybermen στο Doctor Who franchise ή η Borg από το Star Trek), ή σχεδόν δυσδιάκριτα από τους ανθρώπους (π.χ. οι Terminators από τις ταινίες Terminator, οι "ανθρώπινοι" Cylons από το re-imagining of Battlestar Galactica κλπ.). Η τηλεοπτική σειρά Six Million Dollar Man της δεκαετίας του 1970, η οποία βασίστηκε σε ένα από τα μυθιστορήματα του Martin Caidin με τίτλο Cyborg, παρουσίασε ένα από τα πιο διάσημα μυθοπλαστικά cyborgs , αναφερόμενο ως ένας βιονικός άνθρωπος. (S.m.d.mAN) Τα Cyborgs στα μυθιστορήματα συχνά παίζουν ένα περιφρονητικό από τους ανθρώπους ρόλο για την υπερβολική εξάρτηση τους από την τεχνολογία, ιδιαίτερα όταν χρησιμοποιούνται για πολέμους και με τρόπο που φαίνεται να απειλούν την ελεύθερη βούληση. Επίσης τα Cyborgs συχνά απεικονίζονται να διακατέχουν σωματικές ή διανοητικές ικανότητες που υπερβαίνουν κατά πολύ έναν ανθρώπινο οργανισμό (στις στρατιωτικές μορφές μπορεί να έχουν ενσωματωμένα όπλα, μεταξύ άλλων) .

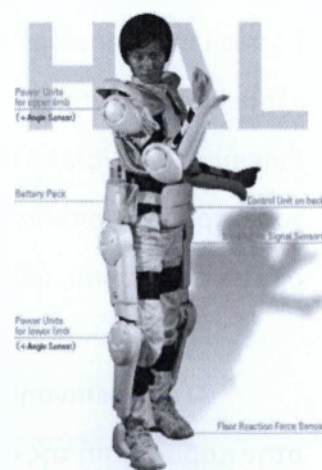
Ορισμός

Σύμφωνα με κάποιους ορισμούς του όρου, οι μεταφυσικές και οι σωματικές ικανότητες της ανθρωπότητας ακόμη και με τις πιο βασικές τεχνολογίες, μας έχουν κάνει ήδη cyborgs. Σε ένα τυπικό παράδειγμα , ένας άνθρωπος που του έχουν προσαρμόσει βηματοδότη της καρδιάς ή μια αντλία ινσουλίνης (αν το άτομο έχει διαβήτη) θα μπορούσε να θεωρηθεί cyborg , δεδομένου ότι αυτά τα μηχανικά μέρη ενισχύουν τους «φυσικούς» μηχανισμούς του οργανισμού μέσω των συνθετικών μηχανισμών ανάδρασης (ανατροφοδότησης). Μερικοί θεωρητικοί αναφέρουν τις τροποποιήσεις όπως οι φακοί επαφής ,τα ακουστικά βαρηκοΐας ή οι φακοί επαφής ως παραδείγματα προσαρμογής των ανθρώπων με την τεχνολογία για την ενίσχυση των

βιολογικών δυνατοτήτων τους. Ωστόσο, οι τροποποιήσεις αυτές είναι τόσο «cybernetic» (κυβερνητικές)όσο ένα στυλό ή ένα ξύλινο πόδι . Εμφυτεύματα , όπως τα κοχλιακά εμφυτεύματα , που συνδυάζουν μηχανικές τροποποιήσεις με οποιοδήποτε είδος αντιδραστικής απάντησης είναι οι πιο ακριβείς βελτιώσεις cyborg . (We.C.)

Επίσης ο όρος χρησιμοποιείται για την προσφώνηση των μιγμάτων ανθρώπων-τεχνολογίας στην θεωρία. Αυτό περιλαμβάνει όχι μόνο κοινά χρησιμοποιούμενα κομμάτια της τεχνολογίας, όπως τα κινητά τηλέφωνα, τους υπολογιστές, το διαδίκτυο, κλπ. , αλλά και αντικείμενα που δεν μπορούν ευρέως να θεωρηθούν τεχνολογία, για παράδειγμα, το στυλό, το χαρτί, την ομιλία και τη γλώσσα . Αυξάνοντας αυτές τις τεχνολογίες, και επικοινωνώντας με ανθρώπους σε άλλους χρόνους και τόπους, ένα άτομο γίνεται ικανό για πολύ περισσότερα από ό, τι ήταν πριν. Αυτό είναι παρόμοιο με τους υπολογιστές, οι οποίοι αποκτούν ισχύ με τη χρήση των πρωτοκόλλων Internet για να συνδεθούν με άλλους υπολογιστές. Οι Διακυβερνητικές (Cybernetic) τεχνολογίες περιλαμβάνουν αυτοκινητόδρομους, αγωγούς, ηλεκτρικά καλώδια, κτίρια, ηλεκτρικές εγκαταστάσεις, βιβλιοθήκες, και άλλες υποδομές που δύσκολα τις παρατηρούμε, οι οποίες είναι κρίσιμα τμήματα μέσα στη διακυβερνητική που δουλεύουμε. (Natural.B.C.)

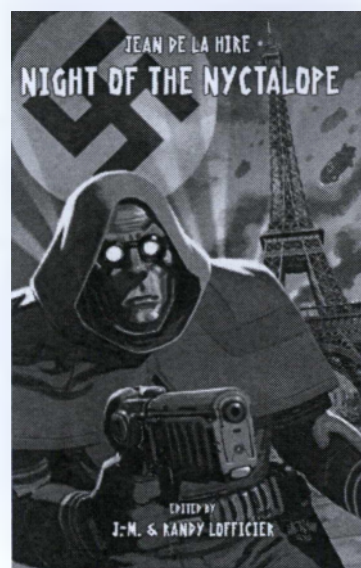
Ο συγγραφέας επιστημονικής φαντασίας Bruce Sterling στο σύμπαν του «Shaper / mechanist» πρότεινε μια ιδέα των εναλλακτικών cyborg που ονομάζεται Lobster, η οποία γίνεται όχι με τη χρήση εσωτερικών εμφυτευμάτων, αλλά με τη χρήση ενός εξωτερικού κελύφους (π.χ. Ενεργειακού Εξωσκελετού, Powered Exoskeleton). (Lob.C.) Σε αντίθεση με τα ανθρώπινα cyborgs που εμφανίζονται ανθρώπινα εξωτερικά ενώ είναι συνθετικά εσωτερικά, ο «Lobster» (Αστακός) φαίνεται απάνθρωπος εξωτερικά, αλλά περιέχει έναν άνθρωπο εσωτερικά. Το ηλεκτρονικό παιχνίδι Deus Ex : Invisible War παρουσιάζει κυρίως cyborgs που ονομάζονται Ομάρ , όπου "Ομάρ" είναι μια ρωσική μετάφραση της λέξης "Αστακός" (αφού τα Ομάρ είναι ρωσικής καταγωγής στο παιχνίδι). (Lob.C.)



Εικ.110 Cyberdyne Hal - 5, Ενεργειακός Εξωσκελετός

1.11.1 Προέλευση της έννοιας

Η έννοια ενός μίγματος ανθρώπου-μηχανής ήταν διαδεδομένη στην επιστημονική φαντασία πριν από τον Δεύτερο Παγκόσμιο Πόλεμο. Ήδη από το 1843 ,ο Edgar Allan Poe περιγράφει έναν άνθρωπο με εκτεταμένες προσθετικές στο διήγημα «The Man That Was Used Up». Το 1908 , ο Jean de la Hire παρουσίασε τον Nyctalope (ίσως το πρώτο αληθινό superhero, και το πρώτο λογοτεχνικό cyborg) στο μυθιστόρημα L' Homme Qui Peut Vivre Dans L' eau (Ο άνθρωπος που μπορεί να ζει στο νερό). (C.nyctalope) Ο Edmond Hamilton παρουσίασε εξερευνητές του διαστήματος με ένα μίγμα οργανικών και μηχανικών τμημάτων στο μυθιστόρημά του «The Comet Doom» (Ο Κομήτης Ντουμ)το 1928. Επίσης παρουσίασε αργότερα τον ομιλούμενο και ζωντανό εγκέφαλο ενός παλιού επιστήμονα, ονόματος Simon Wright, επιπλέοντας σε μια διαφανή θήκη, σε όλες τις περιπέτειες του διάσημου ήρωα του, τον Captain Future. Χρησιμοποιεί τον όρο cyborg ρητά ,το 1962 σε μια μικρή ιστορία με τίτλο «Μετά από την Ημέρα της Κρίσης » ("After a Judgment Day"), για να περιγράψει τα «μηχανικά ανάλογα» ("mechanical analogs") που ονομάζονται «Charlies », εξηγώντας ότι τα cyborgs είχαν κληθεί από την πρώτη δεκαετία του 1960 κυβερνητικοί οργανισμοί. (C.Story) Στο διήγημα "No Woman Born " το 1944 , η C. L. Moore έγραψε για την Deirdre, μια χορεύτρια, της οποίας το σώμα είχε καεί εντελώς και ο εγκέφαλος της είχε τοποθετηθεί σε ένα απρόσωπο, αλλά όμορφο και εύπλαστο μηχανικό σώμα. (C.dance)



Εικ.111 Nyctalope, από τον Jean de la Hire

Ο όρος επινοήθηκε από τον Manfred E. Clynes και τον Nathan S. Kline το 1960 στην παραπομπή της αντίληψής τους για ένα βελτιωμένο ανθρώπινο ον που θα μπορούσε να επιβιώσει σε εξωγήινα περιβάλλοντα:

«Για το εξωγενώς εκτεταμένο οργανωτικό συγκρότημα το οποίο λειτουργεί ασυνείδητα ως ένα ολοκληρωμένο ομοιοστατικό σύστημα, προτείνουμε τον όρο «Cyborg» - Manfred E. Clynes και Nathan S. Kline. (O.Cyborg)

Η ιδέα τους ήταν το αποτέλεσμα της σκέψης σχετικά με την ανάγκη για μια οικεία σχέση μεταξύ ανθρώπου και μηχανής, καθώς τα νέα σύνορα της εξερεύνησης του διαστήματος άρχιζαν να παίρνουν μέρος. Ένας σχεδιαστής των φυσιολογικών οργάνων και των ηλεκτρονικών συστημάτων επεξεργασίας δεδομένων, εν ονόματι Clynes ήταν ο κύριος ερευνητής στο Εργαστήριο Δυναμικής Προσομοίωσης στο Rockland State Hospital της Νέας Υόρκης.

Ο όρος εμφανίζεται για πρώτη φορά σε έντυπη μορφή πριν από πέντε μήνες, όταν η New York Times ανέφερε σχετικά με τις ψυχοφυσιολογικές πτυχές των διαστημικών πτήσεων φιλολογικής συγκέντρωσης όπου ο Clynes και ο Kline παρουσίασαν το πρώτο τους επιστημονικό έγγραφο.

« Ένα cyborg είναι ουσιαστικά ένα άνθρωπο-μηχανικό σύστημα στο οποίο οι μηχανισμοί ελέγχου του ανθρώπινου τμήματος έχουν τροποποιηθεί εξωτερικά από φάρμακα ή από κανονιστικές διατάξεις, έτσι ώστε το ον να μπορέσει να ζήσει σε ένα περιβάλλον διαφορετικό από το κανονικό.» (Clynes.Kline)

Ένα βιβλίο με τίτλο «Cyborg : Digital Destiny and Human Possibility» στην εποχή του Wearable computer (φορητού υπολογιστή) εκδόθηκε από την εκδοτική εταιρία Doubleday το 2001. Μερικές από τις ιδέες στο βιβλίο ενσωματώθηκαν στη 35 χιλιοστών κινηματογραφική ταινία Cyberman. (C.Cyberman)

1.11.2 Ιστούς Cyborg στη μηχανική

Οι cyborg ιστοί δομημένοι με νανοσωλήνες άνθρακα και κύτταρα φυτών ή μυκήτων έχουν χρησιμοποιηθεί στο σχεδιασμό τεχνητών ιστών για την παραγωγή νέων υλικών για τις μηχανικές και ηλεκτρικές χρήσεις. Το έργο παρουσιάστηκε από το Di Giacomo και το Magesca στο MRS 2013 Εαρινό Συνέδριο στις 3 Απριλίου, ομιλούμενος αριθμός SS4.04 . Το cyborg που απέκτησαν είναι φτηνό, ελαφρύ, έχει μοναδικές μηχανικές ιδιότητες και μπορεί να διαμορφωθεί σε επιθυμητές μορφές. Τα κύτταρα σε συνδυασμό με τα MWCNT συν- κατακρημνίζονται ως ένα ειδικό σύνολο κυττάρων και

νανοσωλήνων, σχηματίζοντας ένα παχύρρευστο υλικό. Παρομοίως, τα αποξηραμένα κύτταρα εξακολουθούν να ενεργούν ως μια σταθερή μήτρα (καλούπι) για το MWCNT δίκτυο. Όταν παρατηρείται από ένα οπτικό μικροσκόπιο το υλικό μοιάζει με ένα τεχνητό "ιστό" που αποτελείται από εξαιρετικά συμπυκνωμένα κύτταρα. Η επίδραση της ξήρανσης των κυττάρων εκδηλώνεται με την εμφάνιση του «κυττάρου - φαντάσματος» ("ghost cell") τους. Μια μάλλον ιδιαίτερη φυσική αλληλεπίδραση μεταξύ των MWCNT και των κυττάρων, που παρατηρήθηκε με ένα ηλεκτρονικό μικροσκόπιο, υποδηλώνει ότι το κυτταρικό τοίχωμα (το πιο εξωτερικό τμήμα των μυκητιακών και φυτικών κυττάρων) μπορεί να παίζει ένα σημαντικό ενεργό ρόλο στη δημιουργία και τη σταθεροποίηση ενός δικτύου CNTs. Αυτό το νέο υλικό μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα ηλεκτρονικών εφαρμογών από τη θέρμανση μέχρι και την αίσθηση, και έχει τη δυνατότητα να ανοίξει νέες σημαντικές κατευθύνσεις που πρέπει να αξιοποιηθούν για την ηλεκτρομαγνητική θωράκιση των ηλεκτρονικών ραδιοσυχνοτήτων και για την αεροδιαστημική τεχνολογία. Ειδικότερα, έχουν αναφερθεί υλικά cyborg ιστού, χρησιμοποιώντας κύτταρα *Candida albicans*, με την ιδιότητα της αίσθησης της θερμοκρασίας. (C.tissue)

1.11.3 Ατομικά – Προσωπικά Cyborgs

Γενικώς, ο όρος "Cyborg" χρησιμοποιείται για να αναφερθεί σε έναν άνθρωπο με βιονικά ή ρομποτικά εμφυτεύματα.

Στις τρέχουσες εφαρμογές προσθετικών, το C-Leg σύστημα που αναπτύχθηκε από τον Otto Bock HealthCare χρησιμοποιείται για να αντικαταστήσει ένα ανθρώπινο πόδι που έχει ακρωτηριαστεί λόγω τραυματισμού ή ασθένειας. Η χρήση των αισθητήρων στο τεχνητό C -Leg βοηθάει σημαντικά στο περπάτημα προσπαθώντας να αναπαράγει το φυσικό βηματισμό του χρήστη, όπως ήταν πριν από τον ακρωτηριασμό. (c-leg)

Τεχνητά μέλη, όπως το C -Leg και τα πιο προηγμένα iLimb θεωρούνται από κάποιους να είναι το πρώτο πραγματικό βήμα προς την επόμενη γενιά των cyborg εφαρμογών στον



Εικ.112 iLimb Ultra

πραγματικό κόσμο. Επιπλέον, τα κοχλιακά εμφυτεύματα και μαγνητικά εμφυτεύματα, τα οποία παρέχουν στους ανθρώπους τις αισθήσεις που διαφορετικά δεν θα είχαν, μπορεί να θεωρηθούν ως δημιουργία cyborgs .

Στην επιστήμη της όρασης, άμεσα εμφυτεύματα του εγκεφάλου έχουν χρησιμοποιηθεί για τη θεραπεία μη-συγγενής (επίκτητη) τύφλωσης. Ένας από τους πρώτους επιστήμονες που προέβαλλε μια εύχρηστη διεπαφή του εγκεφάλου για την αποκατάσταση της όρασης ήταν ο ιδιωτικός ερευνητής William Dobelle. Το πρώτο πρωτότυπο του Dobelle είχε εμφυτευθεί στον Jerry το 1978, έναν άνθρωπο που τυφλώθηκε στην ενήλικη ζωή του . Μία συστοιχία BCI που περιέχει 68 ηλεκτρόδια εμφυτεύτηκε στον οπτικό φλοιό του Jerry και πέτυχε την παραγωγή φωτοψιών (phosphenes), η αίσθηση του να βλέπεις φως, και το σύστημα περιλαμβάνει κάμερες που έχουν τοποθετηθεί σε γυαλιά για να στέλνουν σήματα στο εμφύτευμα. Αρχικά, το εμφύτευμα επέτρεψε στον Jerry να δει αποχρώσεις του γκρι σε ένα περιορισμένο οπτικό πεδίο σε χαμηλό frame-rate (ρυθμός καρέ). Επίσης, για αυτό απαιτείτο από αυτόν να είναι συνδεδεμένος με ένα mainframe (μεγάλο σύστημα υπολογιστή) δύο τόνων, αλλά η συρρίκνωση των ηλεκτρονικών και οι πιο γρήγοροι υπολογιστές έκαναν το τεχνητό μάτι του πιο φορητό και τώρα έχει τη δυνατότητα να εκτελεί απλές εργασίες χωρίς βοήθεια. (Jerry)

Το 1997, ο Philip Kennedy , ένας επιστήμονας και γιατρός σχεδίασε τον πρώτο ανθρώπινο cyborg του κόσμου που ονομάζεται Johnny Ray . Ο Ray ήταν ένας βετεράνος του Βιετνάμ στη Γεωργία ο οποίος υπέστη εγκεφαλικό επεισόδιο. Δυστυχώς, το σώμα του Ρέι, όπως ο γιατρός το ονόμασε, ήταν "εγκλωβισμένο". Ο Ray ήθελε τη παλιά του ζωή πίσω έτσι συμφώνησε στον πειραματισμό του Kennedy, ο οποίος ενσωμάτωσε ένα Neurotrophic ηλεκτρόδιο κοντά στο μέρος του εγκεφάλου του Ray, έτσι ώστε να αποκτούσε κάποια κίνηση πίσω στο σώμα του. Η εγχείρηση επέτυχε, αλλά το 2002 ο Johnny Ray απεβίωσε. (J.Ray)

Το 2002, ο Jens Naumann καναδικής καταγωγής, επίσης τυφλωμένος στην ενήλικη ζωή του , ήταν ο πρώτος σε μια σειρά από 16 ασθενείς που πλήρωσαν για να λάβουν το δεύτερο εμφύτευμα της γενιάς Dobelle, σημαίνοντας μία από τις πρώτες εμπορικές χρήσεις των BCI. Η δεύτερη γενιά της συσκευής χρησιμοποιούσε ένα πιο

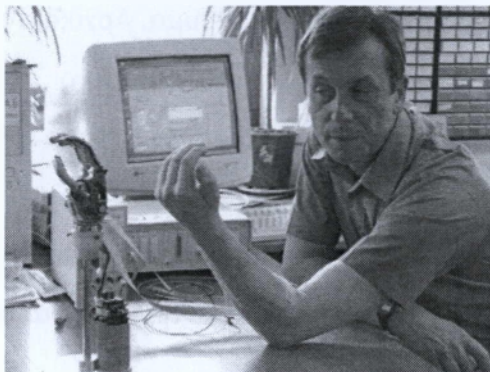


Εικ.113 Jens Naumann, ασθενής τεχνητής όρασης

εξελιγμένο εμφύτευμα που επέτρεπε την καλύτερη χαρτογράφηση των φωτοψιών σε συνεκτική όραση, τα οποία είναι απλωμένα σε όλο το οπτικό πεδίο, αυτό που οι ερευνητές αποκαλούν έναστρης-νύχτας (starry-night) αποτέλεσμα. Αμέσως μετά την εμφύτευση του, ο Jens ήταν σε θέση να χρησιμοποιεί την ατελή αποκατεστημένη όραση του για να οδηγήσει

σιγά-σιγά γύρω από το χώρο στάθμευσης του ερευνητικού ινστιτούτου. (jens)

Το 2002 , υπό τον τίτλο «Project Cyborg», ο Βρετανός επιστήμονας Kevin Warwick, είχε μια σειρά από 100 ηλεκτρόδια βαλμένα στο νευρικό του σύστημα



Εικ.114 Project Cyborg, από τον Kevin Warwick

προκειμένου να το συνδέσουν με το Internet. Με αυτό τον τρόπο πραγματοποίησε με επιτυχία μια σειρά πειραμάτων συμπεριλαμβανομένης της επέκτασης του νευρικού συστήματος του μέσω του Διαδικτύου για να ελέγξει ένα ρομποτικό χέρι, ένα ηχείο και τον ενισχυτή. Αυτό είναι μια μορφή εκτεταμένων αισθητηριακών πληροφοριών και η πρώτη άμεση ηλεκτρονική

επικοινωνία μεταξύ του νευρικού συστήματος δύο ανθρώπων. (K.Warwick)

Το 2004 , υπό τον τίτλο «Bridging the Island of the Colourblind Project» (Γεφυρώνοντας το νησί του προγράμματος αχρωματοψίας), ο Βρετανός και εντελώς αχρωμάτωσ καλλιτέχνης Neil Harbisson, άρχισε να φοράει ένα Eyeborg στο κεφάλι του, προκειμένου να ακούσει τα χρώματα. Η προσθετική του συσκευή είχε συμπεριληφθεί εντός του 2004 στην φωτογραφία διαβατηρίου του, που έχει ζητηθεί για να επιβεβαιώνει την cyborg κατάσταση του. Το 2012 στο TEDGlobal , ο Harbisson εξήγησε ότι ο ίδιος δεν αισθανόταν σαν ένα cyborg όταν άρχισε να



Εικ.115 Neil Harbisson, Cyborg καλλιτέχνης

χρησιμοποιεί το Eyeberg , το αισθάνθηκε όταν παρατήρησε ότι το λογισμικό και το μυαλό του είχαν ενωθεί και του είχαν δώσει μια επιπλέον αίσθηση. (N.Harbisson)

1.11.5 Πολλαπλασιασμός Cyborg στην κοινωνία

1.11.5.1 Στην Ιατρική

Στην ιατρική, υπάρχουν δύο σημαντικά και διαφορετικά είδη cyborgs : τα επανορθωτικά (restorative) και τα ενισχυτικά (enhanced). Οι Επανορθωτικές τεχνολογίες αποκαθιστούν την απώλεια λειτουργίας, οργάνων και άκρων. Η βασική πτυχή της επανορθωτικής δημιουργίας cyborgs (cyborgization) είναι η επισκευή των σπασμένων ή λείπων διεργασιών για να επανέλθει σε ένα υγιή ή μέσο επίπεδο λειτουργίας, και δεν υπάρχει καμία βελτίωση στις αρχικές ικανότητες και τις διεργασίες που χάθηκαν.

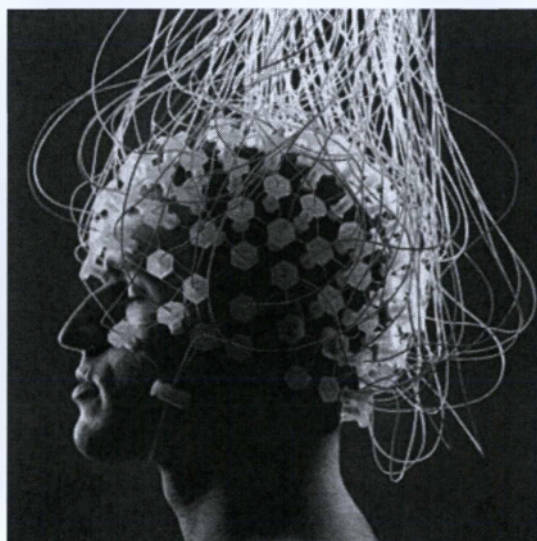
Αντιθέτως, το ενισχυτικό Cyborg ακολουθεί την αρχή της βέλτιστης απόδοσης: τη μεγιστοποίηση της απόδοσης (οι πληροφορίες ή οι τυποποιήσεις που λαμβάνονται) και την ελαχιστοποίηση των εισροών (η ενέργεια που δαπανάται κατά τη διαδικασία). Έτσι, η ενισχυμένη δημιουργία cyborg σκοπεύει στην υπέρβαση των συνηθισμένων διεργασιών ή ακόμα και στην απόκτηση νέων λειτουργιών που δεν ήταν αρχικά παρούσες. (C.medicine1)

Παρόλο που οι προσθετικές σε γενικές γραμμές συμπληρώνουν τα χαμένα ή καταστραμμένα μέρη του σώματος με την ενσωμάτωση ενός μηχανικού τεχνάσματος, τα βιονικά εμφυτεύματα στην ιατρική επιτρέπουν μοντέλα οργάνων ή μέρη του σώματος να μιμηθούν την αρχική λειτουργία πιο στενά. Ο Michael Chorost έγραψε τα απομνημονεύματα της εμπειρίας του με τα κοχλιακά εμφυτεύματα ή τα βιονικά αυτιά, με τίτλο « Ανακατασκευή: Πώς γίνοντας μέρος



Εικ.116 Jesse Sullivan, πλήρης λειτουργία ρομποτικού άκρου

ενός υπολογιστή με έκανε πιο ανθρώπινο». Ο Jesse Sullivan έγινε ένας από τους πρώτους ανθρώπους που λειτουργούσε ένα πλήρως ρομποτικό σκέλος μέσω ενός νεύρο-μυϊκού μοσχεύματος, επιτρέποντας του ένα πολύπλοκο φάσμα κινήσεων πέρα από εκείνων των προηγούμενων προσθετικών. (J.Sull.) Μέχρι το 2004, μια πλήρως λειτουργούσα τεχνητή καρδιά αναπτύχθηκε. Η συνεχής τεχνολογική εξέλιξη των βιονικών και των νανοτεχνολογιών αρχίζει να θέτει το ζήτημα της ενίσχυσης, και των μελλοντικών δυνατοτήτων των cyborgs που ξεπερνούν την αρχική λειτουργία του βιολογικού μοντέλου. Η ηθική και η επιθυμία της "ενισχυτικής προσθετικής" έχουν συζητηθεί, οι υποστηρικτές τους περιλαμβάνουν το μεταανθρωπιστικό κίνημα, με την πεποίθησή ότι οι νέες τεχνολογίες μπορούν να βοηθήσουν την ανθρώπινη φυλή στην ανάπτυξη πέραν του παρόντος, ξεπερνώντας τους κανονιστικούς περιορισμούς όπως είναι η γήρανση και οι ασθένειες, καθώς και άλλες γενικότερα ανικανότητες, όπως τους περιορισμούς στην ταχύτητα, τη δύναμη, την αντοχή και την ευφυΐα. Οι αντίπαλοι της έννοιας περιγράφουν ότι αυτό που πιστεύουν είναι προκαταλήψεις που προωθούν την ανάπτυξη και την αποδοχή των εν λόγω τεχνολογιών, δηλαδή μια προκατάληψη προς τη λειτουργικότητα και την αποτελεσματικότητα που υποχρεώνει τη σύμφωνη γνώμη σε μια προβολή του ανθρώπινου πληθυσμού, που δεν τονίζει σαν προσδιορισμό των χαρακτηριστικών τις πραγματικές εκδηλώσεις της ανθρωπότητας και της προσωπικότητας, υπέρ του καθορισμού όσον αφορά τις αναβαθμίσεις, τις εκδοχές, και τη χρησιμότητα. (C.medicine2)

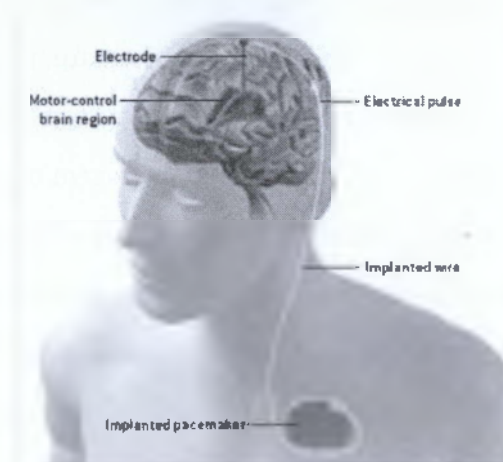


Εικ.117 Μια διασύνδεση εγκεφάλου με υπολογιστή

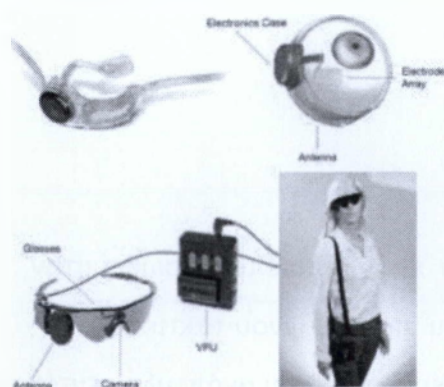
Μια διεπαφή (διασύνδεση) εγκεφάλου-υπολογιστή (brain-computer interface), ή BCI, παρέχει μια άμεση πορεία επικοινωνίας από τον εγκέφαλο σε μια εξωτερική συσκευή, δημιουργώντας ουσιαστικά ένα cyborg. Η ερευνά της επεμβατικής BCIS, που χρησιμοποιεί ηλεκτρόδια τα οποία εμφυτεύονται απευθείας στη φαιά ουσία του εγκεφάλου, έχει επικεντρωθεί στην αποκατάσταση

κατεστραμμένης όρασης σε τυφλούς και στο να παρέχει λειτουργικότητα σε παράλυτους ανθρώπους, ιδιαίτερα εκείνων με σοβαρές περιπτώσεις, όπως το σύνδρομο εγκλεισμού. Αυτή η τεχνολογία θα μπορέσει να επιτρέψει στους ανθρώπους που τους λείπουν ένα άκρο ή είναι σε αναπηρική καρέκλα, την εξουσία να ελέγχουν τις συσκευές που τους βοηθούν μέσω νευρικών σημάτων που αποστέλλονται από τα εμφυτεύματα του εγκεφάλου απευθείας στους υπολογιστές και τις συσκευές. Επίσης είναι πιθανό ότι αυτή η τεχνολογία θα χρησιμοποιηθεί τελικά με υγιή άτομα. (BCI)

Η Βαθιά Εγκεφαλική Διέγερση είναι μια νευρολογική χειρουργική διαδικασία που χρησιμοποιείται για τη θεραπεία ασθενών που διαγνώστηκαν με νόσο του Πάρκινσον, νόσο του Alzheimer, σύνδρομο Tourette, επιληψία, χρόνιες κεφαλαλγίες και ψυχικές διαταραχές. Μέσω της αναισθησίας του ασθενούς, εμφυτεύεται στην περιοχή του εγκεφάλου, όπου η αιτία της ασθένειας είναι παρούσα, βηματοδότες εγκεφάλου ή ηλεκτρόδια. Η περιοχή του εγκεφάλου στην συνέχεια διεγείρεται από ριπές ηλεκτρικού ρεύματος έτσι ώστε να διαταράξουν το επερχόμενο κύμα των επιληπτικών κρίσεων. Όπως όλες οι επεμβατικές διαδικασίες, η βαθιά εγκεφαλική διέγερση μπορεί να θέσει τον ασθενή σε μεγαλύτερο κίνδυνο, ωστόσο, έχουν υπάρξει περισσότερες βελτιώσεις τα τελευταία χρόνια με αυτή την διαδικασία από ό, τι οποιαδήποτε διαθέσιμη φαρμακευτική αγωγή. (DBS)



Εικ.118 Deep brain stimulation (DBS) (Βαθιά Εγκεφαλική Διέγερση)



Εικ.119 Argus II, Σύστημα πρόσθεσης του αμφιβληστροειδούς

Τα αμφιβληστροειδείς εμφυτεύματα είναι μια άλλη μορφή της cyborgization (δημιουργίας cyborg) στην ιατρική. Η θεωρία πίσω από την αμφιβληστροειδείς διέγερση για την αποκατάσταση της όρασης σε άτομα που πάσχουν από μελαγχρωστική αμφιβληστροειδοπάθεια (retinitis pigmentosa)

και η απώλεια της όρασης λόγω γήρανσης (συνθήκες υπό τις οποίες κάποιοι άνθρωποι έχουν ασυνήθιστα χαμηλό ποσό γαγγλιακών κυττάρων), είναι ότι το εμφύτευμα αμφιβληστροειδούς και η ηλεκτρική διέγερση θα μπορούσε να λειτουργήσει ως υποκατάστατο για τα ελλείποντα γαγγλιακά κύτταρα (κύτταρα τα οποία συνδέουν τον οφθαλμό με τον εγκέφαλο).

Ενώ οι εργασίες για την τελειοποίηση αυτής της τεχνολογίας εξακολουθούν να γίνονται, υπάρχουν ήδη σημαντικές πρόοδοι όσον αφορά τη χρήση της ηλεκτρονικής διέγερσης του αμφιβληστροειδούς, ώστε να επιτρέπει στο μάτι να αισθάνεται πρότυπα του φωτός. Μια εξειδικευμένη κάμερα φοριέται από τον παθόντα, όπως στα πλαίσια των γυαλιών τους, η οποία μετατρέπει την εικόνα σε ένα πρότυπο της ηλεκτρικής διέγερσης. Ένα τσιπ που βρίσκεται στο μάτι του χρήστη έπειτα διεγείρει ηλεκτρικά τον αμφιβληστροειδή με αυτό το μοτίβο, με το να διεγείρει ορισμένες νευρικές απολήξεις που μεταδίδουν την εικόνα των οπτικών κέντρων του εγκεφάλου, και η εικόνα εμφανίζεται στον χρήστη. Αν οι τεχνολογικές εξελίξεις προχωρήσουν όπως έχει προγραμματιστεί, αυτή η τεχνολογία θα μπορέσει να χρησιμοποιηθεί από χιλιάδες τυφλούς και να αποκατασταθεί η όραση στους περισσότερους από αυτούς. (R.implants)

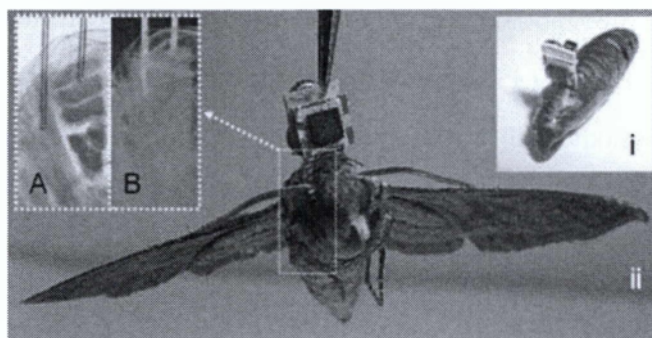
Στις 26 Αύγουστου του 2012 ένα άρθρο από την ιστοσελίδα του Πανεπιστημίου Χάρβαρντ, από τον Peter Reuell του Harvard Gazette, προβαίνει στη συζήτηση έρευνας τρισδιάστατου cyborg ιστού, που δημοσιεύεται στο περιοδικό Nature Materials, με πιθανές ιατρικές συνέπειες που γίνονται από τον Charles M. Lieber, τον Mark Hyman Jr. καθηγητή χημείας και τον Daniel Kohane, ένας Καθηγητής Αναισθησιολογίας της Ιατρικής Σχολής του Χάρβαρντ, στο Νοσοκομείο Παιδών της Βοστώνης. (H.Lieber)

1.11.5.2 Στο Στρατό

Έρευνες στρατιωτικών οργανώσεων έχουν πρόσφατα επικεντρωθεί στην αξιοποίηση των ζώων cyborg για τους σκοπούς του υποτιθέμενου τακτικού πλεονεκτήματος. Η DARPA ανακοίνωσε το ενδιαφέρον της στην ανάπτυξη «εντόμων cyborg» για τη μετάδοση δεδομένων από τους αισθητήρες που εμφυτεύονται στο

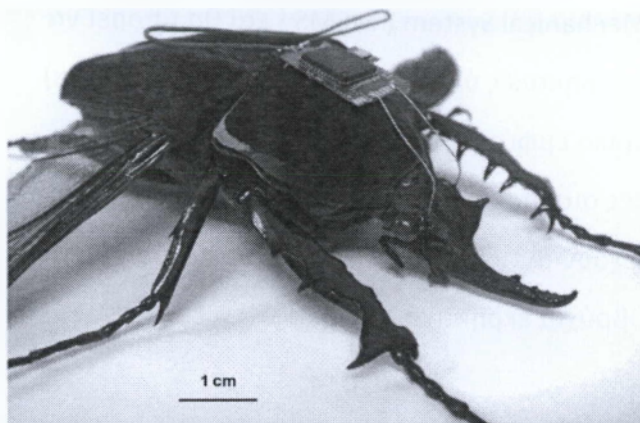
έντομο κατά τη διάρκεια που βρίσκεται στο στάδιο της νύμφης. Η κίνηση του εντόμου θα ελέγχεται από ένα Micro-Electro-Mechanical System (MEMS) και θα μπορεί να ερευνά ένα περιβάλλον ή να εντοπίζει εκρηκτικές ύλες και βλαβερά αέρια. (D.bugs) Ομοίως, η DARPA αναπτύσσει ένα νευρικό εμφύτευμα για να ελέγχει από απόσταση την κίνηση των καρχαριών . Οι μοναδικές αισθήσεις του καρχαρία θα μπορούσε στη συνέχεια να αξιοποιηθούν για να παρέχουν αντιδραστικά δεδομένα σε σχέση με τις εχθρικές κινήσεις του πλοίου ή τα υποβρύχια εκρηκτικά. (D.shark)

Το 2006, ερευνητές στο Πανεπιστήμιο του Cornell εφεύραν μια νέα χειρουργική επέμβαση για την εμφύτευση τεχνητών δόμων σε έντομα κατά τη διάρκεια της μεταμορφωτικής ανάπτυξης τους. Τα πρώτα έντομα cyborgs, σκώροι με ενσωματωμένο ηλεκτρονικό σύστημα στο θώρακα τους, καταδείχθηκαν από τους ίδιους ερευνητές. Η αρχική επιτυχία των τεχνικών ήταν αποτέλεσμα της αυξημένης έρευνας και της δημιουργίας ενός προγράμματος που ονομάζεται Hybrid-Insect - MEMS, HI - MEMS. Ο Στόχος της, σύμφωνα με το Microsystems Technology Office (γραφείο τεχνολογικών μικροσυστημάτων) της DARPA, είναι η ανάπτυξη «στενά συνδεδεμένων μηχανής-εντόμων διασυνδέσεων με την τοποθέτηση μικρο-μηχανικών συστημάτων μέσα στα έντομα κατά τα πρώτα στάδια της μεταμόρφωσης». (Hi-Mems)



Εικ.120 Moth Pupa Hybrid Insect – Micro Electronic Mechanical Systems (HI – MEMS)

Η χρήση των νευρικών εμφυτευμάτων έχει πρόσφατα δοκιμαστεί με επιτυχία στις κατσαρίδες. Χειρουργικά εφαρμοζόμενα ηλεκτρόδια τέθηκαν στο έντομο, τα οποία ελέγχονταν εξ αποστάσεως από έναν άνθρωπο. Τα αποτελέσματα, αν και μερικές φορές ήταν διαφορετικά, ουσιαστικά έδειξαν ότι η κατσαρίδα θα μπορούσε να ελέγχεται από τους παλμούς που λάμβανε μέσω των ηλεκτροδίων. Η DARPA χρηματοδοτεί τώρα αυτή την έρευνα, λόγω των προφανών ευεργετικών εφαρμογών της στις στρατιωτικές και σε άλλες περιοχές. (R.N.implants)



Εικ.121 Hybrid Insect - Micro Electronic Mechanical Systems (HI - MEMS)

Το 2009, στο Ινστιτούτο Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE), στο συνέδριο των μικρο-ηλεκτρονικών μηχανικών συστημάτων (MEMS) στην Ιταλία, οι ερευνητές επέδειξαν το πρώτο «ασύρματο» ιπτάμενο-σκαθάρι (flying-beetle) cyborg. Οι μηχανικοί του Πανεπιστημίου της Καλιφόρνιας στο Berkeley έχουν πρωτοπορήσει

τον σχεδιασμό ενός «τηλεκατευθυνόμενου σκαθαριού», το οποίο χρηματοδοτείται από το πρόγραμμα DARPA HI-MEMS. Επίσης το ίδιο έτος επέδειξαν τον ασύρματο έλεγχο ενός «lift-assisted» σκώρου-cyborg. (mems)

Τελικά, οι ερευνητές σχεδιάζουν να αναπτύξουν το HI - MEMS για τις λιβελλούλες, τις μέλισσες, τα ποντίκια και τα περιστέρια. Για να θεωρηθεί επιτυχία το HI - MEMS κυβερνητικό έντομο θα πρέπει να πετάξει 100 μέτρα από το σημείο εκκίνησης, καθοδηγούμενο μέσω του υπολογιστή, να πράξει μια ελεγχόμενη προσγείωση σε απόσταση 5 μέτρων από ένα συγκεκριμένο σημείο τερματισμού και μόλις προσγειωθεί θα πρέπει να παραμείνει στη θέση του. (D.mems)

1.11.5.3 Δημιουργία cyborgs για κρίσιμες μελέτες κωφών

Ο Joseph Michael Valente, περιγράφει τον "cyborgization" ως μια προσπάθεια κωδικοποίησης της «ομαλοποίησης» μέσω κοχλιακών εμφυτευμάτων σε μικρά κωφά παιδιά. Αντλώντας από την εργασία του Paddy Ladd πάνω στην επιστημολογία των Κωφών και την Cyborg οντολογία της Donna Haraway, ο Valente χρησιμοποιεί την έννοια του cyborg ως ένας τρόπος ανάδευσης της κατασκευής της τελειότητας του cyborg (για τα κωφά παιδιά αυτό θα σήμαινε ότι θα



Εικ.122 Κοχλιακό εμφύτευμα σε παιδί

γιατρεύονταν πλήρως). Ισχυρίζεται ότι οι κατασκευαστές κοχλιακών εμφυτευμάτων, διαφημίζουν και πωλούν τα κοχλιακά εμφυτεύματα ως μια μηχανική συσκευή, καθώς και ως ένα απλό ιατρικό «θαυματουργό φάρμακο» ("miracle cure"). Ο Valente επικρίνει τους ερευνητές κοχλιακών εμφυτευμάτων των οποίων οι σπουδές σε μεγάλο βαθμό μέχρι σήμερα δεν περιλαμβάνουν δέκτες κοχλιακών εμφυτευμάτων, παρ' ότι τα κοχλιακά εμφυτεύματα έχουν εγκριθεί από την Διεύθυνση Τροφίμων και Φαρμάκων των Ηνωμένων Πολιτειών (United States Food and Drug Administration, FDA) από το 1984. (Valente) Η Pamela J. Kincheloe συζητά την αναπαρουσίαση του κοχλιακού εμφυτεύματος στα μέσα μαζικής ενημέρωσης και το λαϊκό πολιτισμό ως μια περιπτωσιολογική μελέτη για τις σημερινές και τις μελλοντικές αντιδράσεις στην ανθρώπινη μεταλλαγή και ενίσχυση. (P.J.Kincheloe)

1.11.5.4 Ίδρυμα Cyborg

Το 2010, το Ίδρυμα Cyborg έγινε ο πρώτος διεθνής οργανισμός στον κόσμο αφιερωμένος στη βοήθεια των ανθρώπων για να γίνουν cyborgs. Το ίδρυμα δημιουργήθηκε από τον cyborg Neil Harbisson και τη Moon Ribas, ως απάντηση στον αυξανόμενο αριθμό των επιστολών και ηλεκτρονικών μηνυμάτων που έλαβαν από ανθρώπους σε όλο τον κόσμο που ενδιαφέρονταν να γίνουν cyborg. Βασικοί στόχοι του Ιδρύματος είναι να επεκτείνει τις ανθρώπινες αισθήσεις και ικανότητες, με τη δημιουργία και την εφαρμογή κυβερνητικών επεκτάσεων στο σώμα, με σκοπό την προώθηση της χρήσης της κυβερνητικής σε πολιτιστικές εκδηλώσεις και την υπεράσπιση των δικαιωμάτων cyborg. Το 2010, το ίδρυμα που εδρεύει στο Mataró (Βαρκελώνη), ήταν ο νικητής των βραβείων Cre@tic, που διοργανώθηκαν από το Tecnocampus Mataró. (C.foundation)

Το 2012, ο ισπανός σκηνοθέτη Ραφαέλ Duran Torrent, δημιούργησε μια ταινία μικρού μήκους με θέμα το Ίδρυμα Cyborg, και το 2013, η ταινία κέρδισε το Μεγάλο Βραβείο της Κριτικής Επιτροπής στο διαγωνισμό «Focus Forward Filmmakers» του φεστιβάλ «Sundance Film Festival» και βραβεύτηκε με 100.000 δολάρια. (D.torrent)

1.12 Διάσημες και μη Ρομποτικές Εφαρμογές

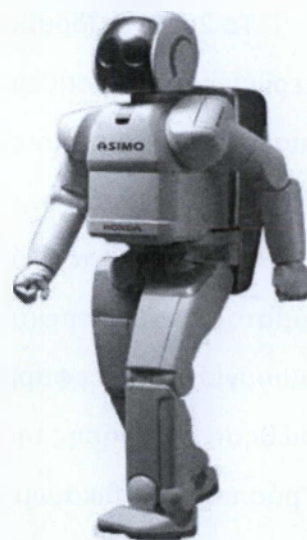
1.12.1 ASIMO

Ο Asimo είναι το τελειότερο Robot που έχει κατασκευάσει μέχρι σήμερα η παγκόσμια τεχνολογία. Μητέρα πνευματική, τεχνολογική και κατασκευαστική είναι η εταιρία HONDA η οποία έχει επενδύσει εκατομμύρια δολάρια στην ανάπτυξη Robot για 2 βασικούς λόγους και δημιουργήθηκε στο Honda's Research & Development Wako Fundamental Technical Research Center στην Ιαπωνία.

Ο πρώτος λόγος είναι ότι μέσα από το τμήμα ρομποτικής που έχει αναπτύξει, μπορεί και αναπτύσσει τεχνογνωσία ώστε να παράγει η ίδια τα μηχανήματα που χρειάζεται για να εξοπλίζει τα εργοστάσια της στην παραγωγή άλλων προϊόντων όπως Αυτοκίνητα, Μοτοσικλέτες κλπ.

Ο δεύτερος λόγος και σημαντικότερος είναι ότι σε λίγα χρόνια, τα Robot θα βγουν στην ευρεία κατανάλωση για διάφορες χρήσεις, χωρίς απαραίτητα να είναι με την ανθρώπινη μορφή. Συστήματα Robot θα εξοπλίζουν στο μέλλον και την πιο μικρή βιοτεχνία, ενώ δεν είναι απίθανο να τα δούμε και στο οικιακό μας περιβάλλον σε καθημερινές ανθρώπινες ανάγκες.

Το δημοφιλές ρομπότ, του ιαπωνικού κολοσσού της αυτοκινητοβιομηχανίας Honda, εμπλουτίστηκε με νέες δυνατότητες, οι οποίες φιλοδοξούν να του χαρίσουν περισσότερα ανθρώπινα στοιχεία. Έτσι, το ανανεωμένο ανθρωπόμορφο, κατά μια έννοια, ρομπότ μπορεί να κατεβαίνει σκάλες και να απαντάει σε ανθρώπινες εντολές. Το βελτιωμένο Asimo παρουσιάστηκε, εντυπωσιάζοντας το κοινό, τη Δευτέρα 12 Νοεμβρίου. Με έναν καθαρά μηχανικό τόνο ρώτησε "Τι θα θέλατε;" στρέφοντας την προσοχή του προς τους δημοσιογράφους και "παίζοντας" με τα δάχτυλα του ενός χεριού του...



Εικ.123 Ρομπότ Asimo

Η Honda Motor ανακοίνωσε την δημιουργία νέων τεχνολογιών για τη νέα γενιά του προηγμένου ρομπότ, ASIMO, στοχεύοντας σε ένα νέο επίπεδο κίνησης το οποίο θα βοηθήσει την λειτουργία του ASIMO και θα αλληλεπιδρά με τους ανθρώπους αφού θα επεξεργάζεται γρήγορα την πληροφορία και το γύρω περιβάλλον. Με την ανίχνευση των κινήσεων των ανθρώπων μέσω των οπτικών αισθητήρων στο κεφάλι του και των αισθητήρων δύναμης οι οποίοι έχουν εκ νέου προστεθεί στους καρπούς του, ο ASIMO μπορεί τώρα να κινείται σε συγχρονισμό με τους ανθρώπους διευκολύνοντας την λήψη ή την απόδοση ενός αντικειμένου, την χειραψία εν αρμονία με τις κινήσεις του ατόμου που βρίσκεται απέναντί του και μπορεί να κάνει βήμα εμπρός ή πίσω ανάλογα με την κατεύθυνση του χεριού που του προτείνεται ή τραβιέται. Με την εξέλιξη αυτών των νέων τεχνολογιών, η Honda θα επιδιώξει την ανάπτυξη ενός ASIMO που θα είναι χρήσιμος στους ανθρώπους. Ακόμα δεν έχει διαπιστωθεί πού και σε ποιο βαθμό θα είναι έτοιμος ο απλός καταναλωτής να χρειαστεί έναν τέτοιο «βοηθό», όμως η HONDA αναπτύσσοντας τεχνολογία και τεχνογνωσία, δεν κερδίζει μόνο τις εντυπώσεις, αλλά είναι έτοιμη να προσφέρει προϊόν όταν η αγορά το ζητήσει.

Η προσπάθεια αυτή ξεκίνησε από την HONDA το 1986 όταν παρουσίασε για πρώτη φορά το πρώτο ανθρωποειδές robot το EO το οποίο ήταν πειραματικό και αυτό που ενδιέφερε την εταιρία κατά κύριο λόγο ήταν η ισορροπία και μόνο. Η εξέλιξη αυτή έφτασε μέχρι τον Σεπτέμβριο του 1997 όπου παρουσιάστηκε για πρώτη φορά ο P3 ο οποίος είχε 100% ισορροπία και αυτονομία μεγαλύτερη από τα προηγούμενα μοντέλα. Στο μοντέλο αυτό δόθηκε πλησιέστερη μορφή με τον άνθρωπο και είχε ύψος 1,60 και βάρος 130 κιλά. Στην πορεία πέρα από την ανάγκη για βελτίωση σε όλα τα επίπεδα, παρατηρήθηκε από την κατασκευαστική ομάδα, ότι όσο το ROBOT ήταν πιο κοντό, τόσο πιο αγαπητό ήταν από τον απλό κόσμο και έτσι στην Κατασκευάστηκε ο ASIMO με ύψος 1,20 και βάρος μόλις 52 κιλά. Η κατασκευή του τελείωσε τον Οκτώβριο του 2000 ενώ είχε ξεκινήσει τον Ιούλιο του 1999. Το όνομα του είναι παρμένο από τα αρχικά της πρότασης Advance Step Innovative Mobility

Μια σημαντική βελτίωση στον έλεγχο του ASIMO είναι ότι μπορούσαν να ελεγχθούν οι βασικές του λειτουργίες και ο προγραμματισμός του, μόνο με ένα Palmtop εν αντιθέσει με τον P3 ο οποίος χρειαζόταν ένα ολόκληρο workstation. Μετά από παρατηρήσεις των απλών ανθρώπων και στατιστική έρευνα που έκανε η εταιρία για την αίσθηση που προκαλούσε ένα ROBOT στον άνθρωπο, έγιναν κάποιες ακόμη

βελτιώσεις πέραν από το ύψος. Π.χ. Παρόλο το γεγονός ότι ο ASIMO φέρει 2 κάμερες για μάτια, οι κατασκευαστές προτίμησαν μία μάσκα αντίστοιχη με το κράνος μοτοσυκλέτας για να είναι πιο φιλικός. Η λογική είναι ότι ο άνθρωπος πρέπει να τον βλέπει ως μια φιλική κατασκευή και όχι ως μια ισχυρότερη οντότητα και κατά επέκταση απειλή.

Βεβαίως όλα αυτά έχουν υπολογιστεί και στις επαφές του ASIMO με τους ανθρώπους όπου συμπεριφέρεται με αγνές κινήσεις (χαιρετισμός, υπόκλιση, χειραψία κλπ) προκαλώντας την συμπάθεια όλων όσων έχουν έρθει σε επαφή μαζί του. Σύμφωνα με τους κατασκευαστές του η φιλοσοφία της κατασκευής του είναι να μπορεί να είναι χρήσιμος σε ανθρώπους με αναπηρία και κινητικά προβλήματα. Από μηχανικής και



Εικ.124 Ρομπότ Asimo

κατασκευαστικής πλευράς, στο εσωτερικό του ASIMO λειτουργούν αδιάκοπα 26 μικροί κινητήρες Servomotors για 26 αντίστοιχες αρθρώσεις προς όλες τις κατευθύνσεις. Το πλαίσιο του είναι από μαγνήσιο και πολλά μέρη από άνθρακονύματα ενώ τα καλώδια του μπορούν να αντέξουν σε αρκετά μεγάλες και μικρές θερμοκρασίες. Σε πλήρη λειτουργία ο ASIMO καταναλώνει τις μπαταρίες του (40V nickel metal hydride) σε 30 λεπτά και χρειάζεται περίπου 4 ώρες για να επαναφορτιστεί. Στην πλάτη του είναι προσαρμοσμένος ο εγκέφαλος για όλες

ανεξαρτήτως τις λειτουργίες και αποτελείται από 1 Motherboard με 4 επεξεργαστές.

Το πιο σημαντικό σημείο βέβαια σε μια τέτοια κατασκευή και τεχνολογία είναι το Software το οποίο είναι γραμμένο σε γλώσσα C και φέρει εκτός των άλλων, λειτουργικό σύστημα, σύστημα κατανομής ενέργειας, σύστημα τηλεμετρίας και εντοπισμού βλαβών και ένα υπερεξελιγμένο πρόγραμμα νευρωνικών δικτύων το οποίο κατά κόρον έχει να κάνει με την ισορροπία και τον βηματισμό του, το οποίο καταναλώνει και το 99% την υπολογιστικής του ισχύος. Ο βηματισμός είναι και το μεγαλύτερο τεχνολογικό επίτευγμα αυτής της κατασκευής, όσο κι αν δείχνει απλό. Η επεξεργαστική ισχύς του μηχανήματος φτάνει τα 450 εκατομμύρια μαθηματικούς υπολογισμούς το δευτερόλεπτο, και τόσοι χρειάζονται περίπου για να μπορέσει αυτό

το Robot να κρατήσει ισορροπία και να υπολογίσει το «επόμενο βήμα» του. Λαμβάνει υπόψη του κλίση του εδάφους, ολισθηρότητα, αντίσταση του αέρα, κινητή επιφάνεια και εδώ να αναφέρω ότι σε demo παρουσίαση του ASIMO σε ειδική εκδήλωση στην Ιαπωνία, το Robot κατάφερε να περπατήσει επάνω σε βαρέλι! Κάτι που είναι δύσκολο ακόμη και για έναν άνθρωπο. Η μέση ωριαία ταχύτητα που πιάνει στην αργή του ταχύτητα, είναι 1,6 χιλιόμετρα την ώρα, ενώ φτάνει τα 4,8 χιλιόμετρα την ώρα στο γρήγορο βηματισμό. Ο ASIMO περπατάει και στις πιο δύσκολες συνθήκες

Από πλευράς νοημοσύνης ο ASIMO είναι φτιαγμένος να βοηθάει κυρίως άτομα με κινητικές δυσκολίες και έχει δυνατότητα με εντολή από τον «δικό του» άνθρωπο, να βάζει και βγάζει πρίζες, ανοίγει και κλείνει συσκευές, να λειτουργεί υπολογιστή, να στέλνει email, ενώ είναι αδύνατον να βλάψει άνθρωπο και δίνει πάντα προτεραιότητα σε αυτόν. Η γλώσσα που μιλάει είναι μόνο Ιαπωνικά και γνωρίζει να δώσει σωστές απαντήσεις σε δεκάδες καθημερινές ερωτήσεις. Το κεφάλι όπου διακρίνονται οι 2 κάμερες αντί για μάτια. Ένας νέος κόσμος δημιουργείται μέσα από την εξέλιξη των τεχνολογιών. Χωρίς φανφάρες, χωρίς μεγάλα λόγια και χωρίς την εκρηκτική εξέλιξη που ίσως θα περιμέναμε και περίμεναν οι πρόγονοι μας. Οι υπολογιστές και οι εξέλιξη των επεξεργαστών, δίνουν σήμερα στους επιστήμονες την δυνατότητα για ταχύτερη και ευκολότερη ανάπτυξη των πιο δύσκολων τεχνολογιών και ερευνών.

Τα Robot όσο μας διεγείρουν την περιέργεια και την φαντασία, άλλο τόσο θα καταρρίπτουν τους μύθους που η επιστημονική φαντασία έχει αναπτύξει για αυτά. Κάθε τεχνολογικό επίτευγμα, είναι απόρροια της ανάγκης του ανθρώπου για ανάπτυξη και εξέλιξη, ως βραχίονας της προσωπικής τους εξέλιξης. Ο άνθρωπος είναι έτοιμος να συζητήσει και να συνεργαστεί με οποιοδήποτε μηχάνημα όταν αυτό είναι φτιαγμένο να τον εξυπηρετεί και αυτό έχει αποδεικνύεται στην σημερινή κοινωνία όπου ακόμη και τα πιο κλασικά ανθρωποκεντρικά επαγγέλματα όπως ο γιατρός, συνεργάζονται αρμονικά με τα πιο εξελιγμένα μηχανήματα. Η κοινωνία του αύριο θα υποδεχθεί και θα δεχθεί τα ROBOT με την ίδια άνεση που δεχθήκαμε το Internet, το κινητό τηλέφωνο, το Home Cinema και όλα αυτά τα οποία όχι μόνο δεν είναι απειλή προς εμάς, αλλά τα θεωρούμε και μέσω την κοινωνικής μας καταξίωσης. Ο ASIMO είναι ότι καλύτερο έχει να δείξει σήμερα η εξέλιξη της ανθρωπότητας και αυτό είναι κάτι που ικανοποιεί τους ανθρώπους που αντιμετωπίζουν την ζωή με ανοικτό μυαλό. (AsimoR.)

1.12.2 AIBO

Ο Aibo, απόγονος των συμπαθητικών σκύλων-ρομπότ της Sony, αποκτά μια πιο άγρια φυσιολογία. Σε αντίθεση με την προηγούμενη γενιά, που εμφάνιζε πιο φιλικά χαρακτηριστικά, το τελευταίο μέλος της οικογένειας, ο ERS-220, διαθέτει πιο "μηχανικό" και επιθετικό σχεδιασμό. Το νέο σκυλί-ρομπότ έχει γίνει πιο "μυώδες", καταλαβαίνει περισσότερες φωνητικές εντολές από τους προκατόχους του και αντιδρά καλύτερα στους διάφορους θορύβους. Μπορεί να διαθέτει 16 συνολικά αρθρώσεις, ωστόσο δεν είναι τόσο κινητικός όσο τα αδέρφια του - για παράδειγμα, δεν μπορεί να κουνήσει τα αυτιά του. Το πρόγραμμα καθοδήγησής του μπορεί να τον κάνει περισσότερο επιθετικό, ωστόσο εξακολουθεί να είναι φιλικός και, κατά τον κατασκευαστή του, αξιαγάπητος.

Σκύλος, και τι σκύλος! Με 16MB RAM, 64μπιτο επεξεργαστή RISC της MIPS, μονάδα εξωτερικής μνήμης για τα προγράμματά του, ψηφιακή κάμερα CCD με 180.000 pixels, αισθητήρα αφής και θερμότητας, άλλους αισθητήρες θέσης και κίνησης, μετρητή αποστάσεων με υπέρυθρες ακτίνες και δεκαοχτώ ολόκληρους βαθμούς ελευθερίας στις κινήσεις του. Άμα είναι να πάρει κανείς σκύλο, είναι φρόνιμο να πάρει ό,τι καλύτερο έχει να προσφέρει η τεχνολογία.



Εικ.125 Ρομποτικός Σκύλος Aibo

Ο AIBO της Sony είναι ένας σκύλος για τον 21ο αιώνα. Η τιμή του είναι ελαφρώς τσουχτερή βεβαίως αλλά η πρωτοπορία κοστίζει. Άλλωστε με τα 2.000 δολάρια περίπου που πληρώνεις, ο σκύλος που παίρνεις έχει αυτονομία για ενενήντα ολόκληρα λεπτά. Πάντως, τον AIBO μετά τα ενενήντα λεπτά θα πρέπει να τον ξαναφορτίσετε. Πρόκειται για μείζονος σημασίας πρόβλημα, αλλά αν πιστεύετε ότι αποτελεί επαρκή αποτρεπτικό παράγοντα, μπορείτε να περιμένετε μέχρι να βγει ο AIBO version 2.0, που θα καταλαβαίνει ότι οι μπαταρίες του πάνε να αδειάσουν και θα συνδέεται μόνος του στην πρίζα.

Ξέρω ότι ήδη έχετε αρχίσει να ανησυχείτε. Σαν οράματα περνούν μπροστά από τα μάτια σας ζοφερές εικόνες από ταινίες επιστημονικής φαντασίας του '50 και του '60, με τα ρομπότ να εξεγείρονται κατά της ανθρώπινης βλακειάς και να αποφασίζουν να πάρουν την κατάσταση στα χέρια τους. Πρόκειται για κίνδυνο ορατό πλην όμως υπερβολικό. Από τον AIBO δεν κινδυνεύετε. Αν τον πιάσουν τα τρελά του και αρχίσει να γαβγίζει ανεξέλεγκτα ή να χορεύει ασταμάτητα χούλα-χουπ, δεν έχετε παρά να πατήσετε ένα κουμπάκι που βρίσκεται κρυμμένο στο στέρνο του (αν δεν αποφασίσει να σας δαγκώσει, βέβαια) και αμέσως θα πέσει για ύπνο.

Τα χαρίσματα του AIBO, ωστόσο, δεν περιορίζονται σε αυτά που έχουμε ήδη πει. Ο σκύλος μπορεί να λειτουργήσει είτε αυτόνομα, είτε με τηλεχειρισμό. Εκεί που, για να πείσετε τον κανονικό σας σκύλο να έρθει προς το μέρος σας, θα πρέπει να του τάξετε την Άρτα με τα Γιάννενα, ο AIBO θα σας υπακούσει αμέσως μόλις πατήσετε το κατάλληλο κουμπί του τηλεκοντρόλ. Θα σας κοιτάξει ερωτηματικά δίνοντας στο κεφάλι του την κατάλληλη κλίση (χρησιμοποιώντας τους τρεις επιτρεπόμενους βαθμούς ελευθερίας), ενώ η ουρά του, με τους δύο βαθμούς ελευθερίας που της επιτρέπονται, θα σας δώσει ένα καλό feedback για το πώς σας αντιμετωπίζει ο 64μπιτος επεξεργαστής RISC που έχει στο μυαλό του.



Εικ.126 Ρομποτικός Σκύλος Aibo

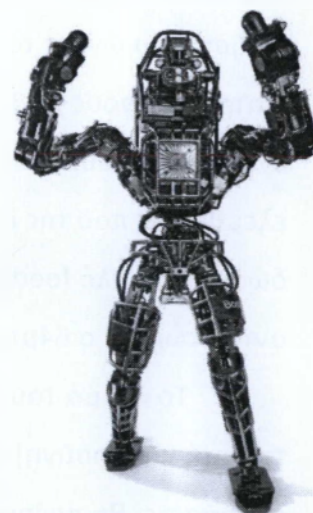
Το όνομα του AIBO βγαίνει από τις λέξεις Artificial Intelligence roBOT (ρομπότ με τεχνητή νοημοσύνη) αλλά στα ιαπωνικά σημαίνει επίσης "σύντροφος". Για να τον αγοράσετε, θα πρέπει να επισκεφθείτε το σχετικό τόπο της Sony στο Internet (www.world.sony.com/robot) και να έχετε εύκαιρα τα στοιχεία της πιστωτικής σας κάρτας. Δεν γνωρίζουμε αν γίνονται αποστολές στις χώρες της Ευρώπης, γιατί διαβάζουμε ότι η πρώτη έκδοση του ηλεκτρονικού σκύλου προσφέρεται μόνο σε περιορισμένα αντίτυπα.

Διαβάζουμε επίσης ότι μέσα στα πρώτα είκοσι λεπτά της διάθεσής τους ξεπουλήθηκαν και τα τρεις χιλιάδες κομμάτια που προορίζονταν για την ιαπωνική αγορά. Παράξενος λαός οι Ιάπωνες. Παράξενος πολιτισμός ο ιαπωνικός. Για σκεφτείτε •

τόσα και τόσα πράγματα που γνωρίζουν εξαιρετική επιτυχία στην Ιαπωνία αφήνουν όλους εμάς τους υπόλοιπους παγερά αδιάφορους. Πάρτε το καραόκι. Δεν υπάρχει Ιάπωνας που να σέβεται τον εαυτό του που να μην έχει τραγουδήσει μια φορά το "Strangers in the Night" σε καραόκι. Στην Ευρώπη και τις ΗΠΑ το καραόκι αποτελεί ατραξιόν "τουριστάδικων" κλαμπ που βασίζονται, μεταξύ άλλων, και στη συνδρομή των Ιαπώνων τουριστών για την επιβίωσή τους. Πάρτε το Tamagotchi. Πόσα παιδάκια ξέρετε που να έχουν πάθει κόλλημα με το ηλεκτρονικό ζωάκι τσέπης; Μόδα ήταν και πέρασε. Στην Ιαπωνία το Tamagotchi έγινε αντικείμενο μελετών πλήθους παιδοψυχολόγων. Δεν ξέρουμε αν η μοίρα του AIBO θα είναι παρόμοια. Η ερευνητική εταιρεία Jupiter Communications, πάντως, που ασχολήθηκε με το θέμα, είχε πολλές επιφυλάξεις. Όπως δήλωσε χαρακτηριστικά ένας από τους αναλυτές της, "Οι Αμερικανοί θα φανούν ιδιαίτερα διστακτικοί να πληρώσουν 2.000 δολάρια για έναν προγραμματιζόμενο σκύλο που, αν του πετάξεις κάτι, δεν θα μπορεί να στο φέρει πίσω". Αυτό είναι ένα πρόβλημα. Όχι η διστακτικότητα των Αμερικανών αλλά το ότι δεν θα σου φέρνει πίσω ο AIBO αυτό που θα του πετάξεις. (AiboR.)

1.12.3 ATLAS

Η DARPA και Boston Dynamics σχεδίασε ένα ρομπότ ύψους 1,89 και βάρους περίπου 150 κιλών, το οποίο θα μπορεί να προσφέρει βοήθεια σε περιπτώσεις φυσικών ή πυρηνικών καταστροφών (πχ Φουκουσίμα), αλλά και να κινείται σε πεδία μαχών. Πρόκειται για το πιο εξελιγμένο ρομπότ, που έχει δημιουργηθεί μέχρι σήμερα, όσον αφορά τις τεχνικές του προδιαγραφές. Οι 28 υδραυλικές αρθρώσεις του, μπορούν να μιμηθούν σχεδόν όλες τις ανθρώπινες κινήσεις, ενώ οι ικανότητες που έχει στα χέρια, ξεπερνούν αυτές ενός φυσιολογικού ανθρώπου.

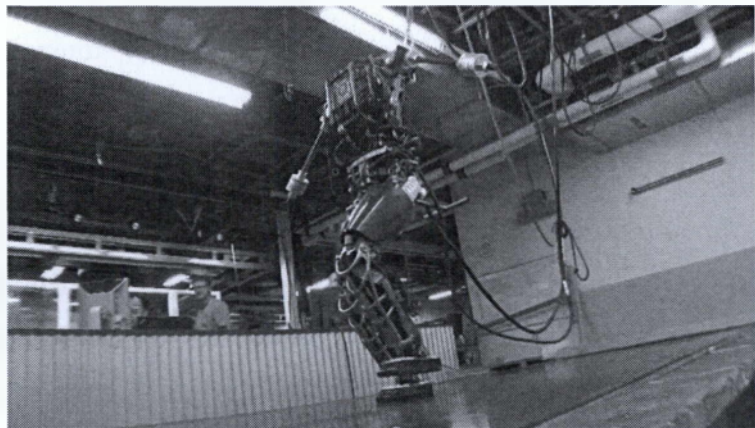


Εικ.127 Atlas Ρομπότ

Το ρομπότ, που ονομάζεται Atlas, έχει ραντάρ, παρόμοιο με αυτά που χρησιμοποιούνται στα αυτόματα οχήματα της Google και έχει τη δυνατότητα να κινείται αυτόνομα όσον αφορά την ενέργεια. Αυτό είναι και ένα στοιχείο, που το διαφοροποιεί και ο λόγος είναι το εξής. Ο Atlas, έχει σχεδιαστεί για να μπορεί να

εισέρχεται σε ζώνες καταστροφής, όπου θα ήταν επικίνδυνο να προσεγγίσουν άνθρωποι. Κινείται με ευκολία σε δύσκολες συνθήκες και βραχώδεις περιοχές και θα μπορούσε να μεταφέρει τραυματίες, χάρη στην ελαστικότητα των άκρων του.

Ωστόσο, δεν θα πρέπει να παραβλέψουμε ότι με τις κατάλληλες προσθήκες, το συγκεκριμένο ρομπότ θα μπορούσε να μετατραπεί σε μια πολεμική μηχανή, δυνατότερη και πολύ πιο ακριβής από κάθε στρατιώτη. Όπως αναφέρει η DailyMail, το μόνο που του λείπει είναι η νόηση, ωστόσο πάνω από 12 ομάδες τεχνικών, εργάζονται, ώστε να το βελτιώσουν και σε αυτό τον τομέα. Κάθε ομάδα θα "φορτώσει" το λογισμικό της στο ρομπότ και θα το δοκιμάσει σε μια σειρά καταστροφών για να δει τι επιλέγει και πώς αντιδρά. Η ομάδα που θα δημιουργήσει το καλύτερο λογισμικό, θα χρηματοδοτηθεί από την DARPA με 2 εκατ. δολάρια σε διαγωνισμό που θα γίνει την επόμενη χρονιά. Παρά τις καλές προθέσεις πάντως, το δημιούργημα θυμίζει αρκετά τον "Εξολοθρευτή" και κανείς δεν μπορεί να προβλέψει με σιγουριά ποια θα είναι η πραγματική χρήση του στο μέλλον και αν τελικά χρησιμοποιηθεί κυρίως στα πεδία των μαχών. (AtlasR.)



Εικ.128 Atlas Ρομπότ

1.13 Τα ρομπότ στην κοινωνία

Περίπου το ήμισυ του συνόλου των ρομπότ στον κόσμο βρίσκονται στην Ασία, 32 % στην Ευρώπη, 16 % στη Βόρεια Αμερική, 1% στην Αυστραλασία και άλλο 1 % στην Αφρική. Το 40 % του συνόλου των ρομπότ στον κόσμο βρίσκονται στην Ιαπωνία, το οποίο την καθιστά, τη χώρα με τον υψηλότερο αριθμό ρομπότ στον κόσμο. (statistics)

1.13.1 Περιφερειακές (χωρικές) προοπτικές

Στην Ιαπωνία και σε άλλες ασιατικές χώρες, οι ιδέες των μελλοντικών ρομπότ έχουν κυρίως θετική ανταπόκριση, και υπάρχει η υποψία ότι η έναρξη της προ-ρομποτικής κοινωνίας πιθανόν να οφείλεται στο διάσημο «Astro Boy». Στις Ασιατικές κοινωνίες, όπως η Ιαπωνία και πιο πρόσφατα η Νότια Κορέα και η Κίνα, πιστεύουν ότι τα ρομπότ είναι πιο ισάξια με τους ανθρώπους, διότι τα χρησιμοποιούν για την φροντίδα των ηλικιωμένων τους, να παίζουν με ή να διδάσκουν τα παιδιά, ή να αντικαταστούν τα κατοικίδια ζώα, κλπ. Η γενική άποψη των Ασιατικών πολιτισμών είναι ότι όσο μεγαλύτερη είναι η πρόσδος των ρομπότ, τόσο το καλύτερο. (A.count-U.c)



Εικ.129 Astro Boy, σελίδα τίτλου για το "Mystery Man of the Blast Furnace," 1961

«Αυτό είναι το άνοιγμα (ξεκίνημα) μιας εποχής στην οποία τα ανθρώπινα όντα και τα ρομπότ μπορούν να συνυπάρξουν», λέει η ιαπωνική εταιρεία Mitsubishi για ένα από τα πολλά ανθρωπιστικά ρομπότ στην Ιαπωνία. (M.humanitarianR.) Επίσης, όπως είδαμε παραπάνω, η Νότια Κορέα έχει στόχο να βάλει ένα ρομπότ σε κάθε σπίτι από το 2015 -2020.

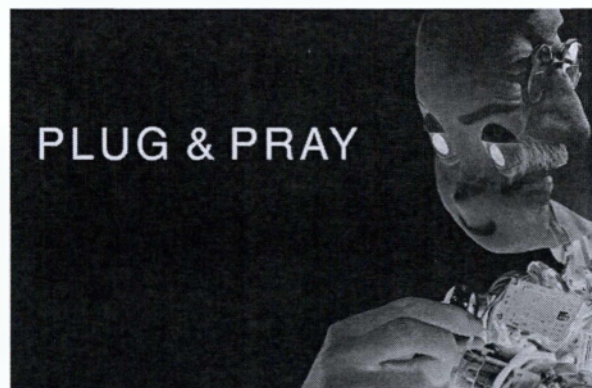
Οι δυτικές κοινωνίες είναι πιο πιθανό να είναι κατά, ή ακόμα και να φοβούνται την ανάπτυξη της ρομποτικής, λόγω των μέσων ενημέρωσης σε ταινίες και στη λογοτεχνία που παρουσιάζουν την αντικατάσταση των ανθρώπων από τα ρομπότ. Ορισμένοι πιστεύουν ότι η Δύση θεωρεί τα ρομπότ ως «απειλή» για το μέλλον των

ανθρώπων, εν μέρει λόγω θρησκευτικών πεποιθήσεων σχετικά με το ρόλο του ανθρώπου και της κοινωνίας. Προφανώς, αυτά τα όρια δεν είναι σαφή, αλλά υπάρχει μια σημαντική διαφορά μεταξύ των δύο πολιτιστικών απόψεων .

1.13.2 Αυτονομία και ηθικά ζητήματα

Καθώς τα ρομπότ έχουν γίνει πιο προηγμένα και εξελιγμένα, εμπειρογνώμονες και ακαδημαϊκοί διερευνούν όλο και περισσότερο τα ερωτήματα: ποια δεοντολογία (ηθική) μπορεί να κυβερνήσει τη συμπεριφορά των ρομπότ, και κατά πόσο τα ρομπότ θα μπορέσουν να είναι σε θέση να διεκδικήσουν κάθε είδους κοινωνικών, πολιτιστικών, ηθικών ή νομικών δικαιωμάτων. Μία επιστημονική ομάδα ανέφερε ότι είναι πιθανό να υπάρξει ένας εγκέφαλος ρομπότ από το 2019, ενώ άλλοι προβλέπουν ανακαλύψεις ρομποτικής νοημοσύνης έως το 2050 .

Πρόσφατες πρόοδοι έχουν εξελίξει την ρομποτική συμπεριφορά, και ο κοινωνικός αντίκτυπος των ευφυών ρομπότ είναι αντικείμενο μιας ταινίας ντοκιμαντέρ του 2010 που ονομάζεται Plug & Pray . (Plug&Pray)



Εικ.130 Plug & Pray ντοκιμαντέρ ,2010

Ο Vernor Vinge έφυσε να εννοηθεί ότι μπορεί να έρθει μια στιγμή όπου οι υπολογιστές και τα ρομπότ θα είναι πιο έξυπνα από τους ανθρώπους, και την αποκαλεί «η Μοναδικότητα» ("the Singularity"). Ο ίδιος προτείνει ότι μια τέτοια κατάσταση μπορεί να είναι κάπως ή ενδεχομένως πολύ επικίνδυνο για τον άνθρωπο, το οποίο συζητείται από μια φιλοσοφία που ονομάζεται Singularitarianism . (V.Vinge)

Το 2009, εμπειρογνώμονες συμμετείχαν σε ένα συνέδριο που διοργάνωσε η Ένωση για την Πρόοδο της Τεχνητής Νοημοσύνης (AAAI) για να συζητήσουν αν οι υπολογιστές και τα ρομπότ είναι σε θέση να αποκτήσουν αυτονομία, και πόσο αυτές οι ικανότητες θα μπορέσουν να αποτελέσουν απειλή ή κίνδυνο. Παρατήρησαν ότι ορισμένα ρομπότ έχουν αποκτήσει διάφορες μορφές ημι-αυτονομίας, μεταξύ άλλων

και τη δυνατότητα να βρίσκουν τις δικές τους πηγές ενέργειας και να είναι σε θέση να επιλέγουν ανεξάρτητα τους στόχους για να τους επιτεθούν με υπάρχοντα όπλα. Επίσης επισήμαναν ότι ορισμένοι ιοί υπολογιστών μπορούν να αποφύγουν την εξάλειψη και να επιτύχουν την κατάσταση που ονομάζεται



Εικ.131 AAAI-Logo

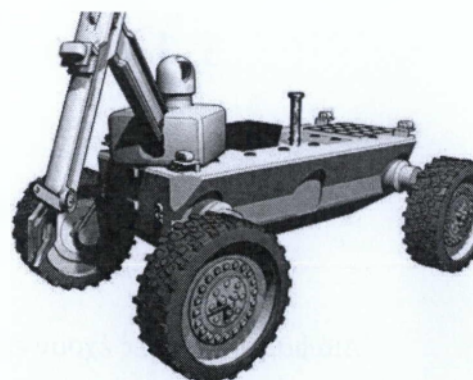
«cockroach intelligence» («νοημοσύνη της κατσαρίδας»). Ακόμα σημείωσαν ότι η αυτο-επίγνωση , όπως απεικονίζεται στην επιστημονική φαντασία, είναι μάλλον απίθανη, αλλά και ότι υπήρχαν άλλοι πιθανοί κίνδυνοι και παγίδες. (aaai.treat) Τέλος διάφορες πηγές των μέσων ενημέρωσης και των επιστημονικών ομάδων έχουν σημειώσει ξεχωριστές τάσεις σε διαφορετικές περιοχές που θα μπορούσαν όλες μαζί να αποτελέσουν μεγαλύτερες ρομποτικές λειτουργίες και αυτονομία, οι οποίες παρουσιάζουν κάποιες εγγενείς ανησυχίες. (S.worry)

1.13.3 Στρατιωτικά ρομπότ

Ορισμένοι εμπειρογνώμονες και ακαδημαϊκοί έχουν αμφισβητήσει τη χρήση των ρομπότ για στρατιωτικές μάχες, ιδίως όταν σε τέτοια ρομπότ δίνεται κάποιος βαθμός αυτόνομων λειτουργιών. Υπάρχουν επίσης ανησυχίες σχετικά με την τεχνολογία που θα μπορούσε να επιτρέψει σε ορισμένα ένοπλα ρομπότ να ελέγχονται κυρίως από άλλα ρομπότ. Το Ναυτικό των ΗΠΑ έχει χρηματοδοτήσει μια έκθεση που δείχνει ότι, καθώς τα στρατιωτικά ρομπότ γίνονται πιο σύνθετα, θα πρέπει να υπάρχει μεγαλύτερη προσοχή στις επιπτώσεις των ικανοτήτων τους να πραγματοποιούν αυτόνομες αποφάσεις. Ένας ερευνητής αναφέρει ότι τα αυτόνομα ρομπότ μπορεί να είναι πιο ανθρώπινα, καθώς θα μπορούσαν να λαμβάνουν αποφάσεις πιο αποτελεσματικά, ωστόσο άλλοι ειδικοί το αμφισβητούν. (Killer.R.)

Το ρομπότ EATR έχει προκαλέσει τις ανησυχίες του κοινού όσον αφορά τη πηγή καυσίμων του, καθώς μπορεί να ανεφοδιάζεται συνεχώς χρησιμοποιώντας οργανικές ουσίες. Παρά το γεγονός ότι ο κινητήρας για την EATR έχει σχεδιαστεί για να λειτουργεί

με βιομάζα και με βλάστηση, ειδικά επιλεγμένα από τους αισθητήρες του, τα οποία μπορεί να βρει στο πεδίο της μάχης ή σε άλλα τοπικά περιβάλλοντα, το έργο έχει δηλώσει ότι επίσης μπορεί να χρησιμοποιηθεί και το λίπος του κοτόπουλου. (EATR)



Εικ.132 EATR (Energetically Autonomous Tactical Robot)

Ο Manuel De Landa έχει σημειώσει ότι

τα έξυπνα βλήματα και οι αυτόνομες βόμβες που είναι εξοπλισμένα με τεχνητή αντίληψη μπορούν να θεωρηθούν ρομπότ, διότι κάνουν ορισμένες από τις αποφάσεις τους αυτόνομα. Πιστεύει ότι αυτό αποτελεί μια σημαντική και επικίνδυνη τάση στην οποία οι άνθρωποι παραδίνουν σημαντικές αποφάσεις στις μηχανές. (M.De.Landa)

1.13.4 Σχέση με την ανεργία

Ένα πρόσφατο παράδειγμα της ανθρώπινης αντικατάστασης περιλαμβάνει την Ταϊβανή εταιρεία τεχνολογίας Foxconn που, τον Ιούλιο του 2011, ανακοίνωσε ένα τριετές σχέδιο για την αντικατάσταση των εργαζομένων με περισσότερα ρομπότ. Επί του παρόντος, η εταιρεία χρησιμοποιεί δέκα χιλιάδες ρομπότ, αλλά θα τα αυξήσει σε ένα εκατομμύριο ρομπότ κατά τη διάρκεια μιας τριετούς περιόδου. (Foxconn)

Τα ρομπότ υπηρεσιών διαφόρων ποικιλιών, συμπεριλαμβανομένων των ιατρικών ρομπότ, υποβρυχίων ρομπότ, ρομπότ επιτήρησης, ρομπότ κατεδάφισης και άλλα είδη των ρομπότ που εκτελούν ένα πλήθος θέσεων εργασίας κερδίζουν σε αριθμούς. Τα υπηρεσιακά ρομπότ είναι καθημερινά εργαλεία για την ανθρωπότητα, μπορούν να καθαρίσουν τα πατώματα, να κόψουν το γκαζόν, να φρουρούν τα σπίτια και επίσης βοηθάνε ηλικιωμένους και άτομα με ειδικές ανάγκες, κάνουν κάποιες χειρουργικές επεμβάσεις, επιθεωρούν σωλήνες και τοποθεσίες που είναι επικίνδυνα για τους ανθρώπους, βοηθούν στην καταπολέμηση των πυρκαγιών και στην εξουδετέρωση των εκρηκτικών μηχανισμών. (Robotworks)

1.14 Το μέλλον των ρομπότ

1.14.1 Τεχνολογικές τάσεις

Διάφορες τεχνικές έχουν εμφανιστεί για να αναπτύξουν την επιστήμη της ρομποτικής και του ρομπότ. Μια μέθοδος είναι η εξελικτική ρομποτική, στην οποία ένας αριθμός διαφορετικών ρομπότ υποβάλλονται σε δοκιμασίες, και εκείνα που έχουν τις καλύτερες επιδόσεις χρησιμοποιούνται ως πρότυπο για τη δημιουργία μιας μεταγενέστερης «γενιάς» ρομπότ. Μια άλλη μέθοδος είναι η αναπτυξιακή ρομποτικής, η οποία παρακολουθεί τις αλλαγές και την ανάπτυξη σε ένα ενιαίο ρομπότ στους τομείς της επίλυσης προβλημάτων και των άλλων λειτουργιών .

Πολλές μελλοντικές εφαρμογές της ρομποτικής φαίνονται προφανές για τους ανθρώπους , ακόμα κι αν είναι πολύ πέρα από τις δυνατότητες των ρομπότ που είναι διαθέσιμα κατά το χρόνο της πρόβλεψης. Ήδη από το 1982 οι άνθρωποι ήταν σίγουροι ότι κάποια μέρα τα ρομπότ θα ήταν σε θέση να:

1. καθαρίζουν μέρη αφαιρώντας molding flash
2. βάφουν με σπρέι τα αυτοκίνητα με απολύτως καμία ανθρώπινη παρουσία
3. συσκευάζουν τα πράγματα σε κουτιά , για παράδειγμα , να προσανατολίζουν και να τοποθετούν σοκολάτας σε κουτιά
4. συναρμολογούν μια ηλεκτρική καλωδίωση
5. φορτώνουν κουτιά σε φορτηγά – πρόβλημα συσκευασίας
6. χειρίζονται μαλακά εμπορεύματα , όπως ενδύματα και υποδήματα
7. κουρεύουν πρόβατα
8. δημιουργούν προσθετικά μέλη για τους μη αρτιμελής ανθρώπους
9. μαγειρεύουν fast food και να εργάζονται σε άλλες βιομηχανίες υπηρεσιών.

Γενικά τέτοιες προβλέψεις είναι υπερβολικά αισιόδοξες βάση του χρονοδιαγράμματος .

1.14.2 Τεχνολογική ανάπτυξη

Γενικές τάσεις

Η Ιαπωνία ελπίζει να έχει πλήρη κλίμακα εμπορευματοποίησης των υπηρεσιακών ρομπότ από το 2025. Μεγάλο μέρος της τεχνολογικής έρευνας στην Ιαπωνία οδηγείται από τους Ιαπωνικούς κυβερνητικούς οργανισμούς, ιδίως του υπουργείου Εμπορίου. (Jetro)

Καθώς τα ρομπότ γίνονται πιο προηγμένα, θα μπορέσει να υπάρξει ένα πρότυπο υπολογιστικό λειτουργικό σύστημα σχεδιασμένο κυρίως για τα ρομπότ; Το Robot Operating System (Ρομποτικό Λειτουργικό Σύστημα) είναι ένα open-source σύνολο προγραμμάτων που αναπτύσσονται στο Πανεπιστήμιο του Στάνφορντ, το Ινστιτούτο Τεχνολογίας της Μασαχουσέτης και το Τεχνικό Πανεπιστήμιο του Μονάχου στη Γερμανία, μεταξύ άλλων. Το ROS παρέχει τρόπους για να προγραμματίσουμε την πλοήγηση και τα άκρα ενός ρομπότ, ανεξάρτητα από το εκάστοτε hardware, επίσης παρέχει υψηλού επιπέδου εντολές για στοιχεία όπως η αναγνώριση εικόνας και το άνοιγμα των θυρών. Όταν το ROS ξεκινάει επάνω στον υπολογιστή ενός ρομπότ, αποκτά δεδομένα σχετικά με τα χαρακτηριστικά όπως το μήκος και την κίνηση των άκρων του ρομπότ και μεταφέρει τα δεδομένα σε υψηλότερα επίπεδα αλγορίθμων. (ROS) Η Microsoft αναπτύσσει επίσης ένα «Windows για τα ρομπότ» σύστημα με το Robotics Developer Studio της, το οποίο είναι διαθέσιμο από το 2007. (RDS)

Εξέλιξη της αγοράς

Η σημερινή αγορά δεν είναι πλήρως ώριμη. Δεν έχουν ακόμα εμφανιστεί ένα ή περισσότερα στρώματα συμβατότητας του λογισμικού, έτσι ώστε να καταστεί δυνατή η ανάπτυξη ενός πλούσιου οικοσυστήματος ρομποτικής (ένα παρόμοιο με τους προσωπικούς υπολογιστές σήμερα). Το πιο συχνά χρησιμοποιούμενο λογισμικό στην έρευνα της ρομποτικής είναι η δωρεάν λύσεις λογισμικού (Free Software), όπως το Player / Stage ή cross-platform τεχνολογίες όπως το Urbi. Η Microsoft εργάζεται αυτή τη στιγμή σε αυτή την κατεύθυνση με το νέο ιδιόκτητο λογισμικό της «Microsoft Robotics

Studio». Η χρήση εργαλείων ανοικτού κώδικα βοηθά στην συνεχή βελτίωση των εργαλείων και αλγορίθμων για την ρομποτική έρευνα από το σημείο που μία ομάδα το αφήνει. (RDS)

Προβλεπόμενο Ρομποτικό Χρονοδιάγραμμα

- 2015-2020 : κάθε σπιτικό της Νότιας Κορέας και πολλά ευρωπαϊκά νοικοκυριά θα έχουν ένα ρομπότ , το Υπουργείο Πληροφοριών και Επικοινωνιών (Νότια Κορέα).
- 2018 : τα ρομπότ θα πραγματοποιήσουν συστηματικά χειρουργικές επεμβάσεις, κυβέρνηση Νότιας Κορέας.
- 2022 : Τα ευφυή ρομπότ που θα ανιχνεύουν το περιβάλλον τους, λαμβάνουν αποφάσεις, μαθαίνουν, θα χρησιμοποιούνται στο 30 % των νοικοκυριών και των οργανώσεων, TechCast.
- 2030 : ρομπότ ικανά να εκτελούν σε ανθρώπινο επίπεδο στις περισσότερες χειρωνακτικές εργασίες, Marshall Brain. (M.B.)
- 2034 : ρομπότ (σπιτικά συστήματα αυτοματισμού) που θα εκτελούν τις περισσότερες οικιακές εργασίες , Helen Greiner , πρόεδρος της iRobot. (H.G.)
- 2050 : ρομπότ «εγκεφάλους» που βασίζονται σε υπολογιστές που εκτελούν 100 τρισεκατομμύρια εντολές ανά δευτερόλεπτο θα αρχίσουν να ανταγωνίζονται την ανθρώπινη νοημοσύνη.

Στρατιωτικά ρομπότ :

- 2015 : το ένα τρίτο της δύναμης των ΗΠΑ θα αποτελείται από ρομπότ, Υπουργείο Άμυνας των ΗΠΑ.
- 2035 : πρώτοι πλήρως αυτόνομοι στρατιώτες ρομπότ σε λειτουργία, Υπουργείο Άμυνας των ΗΠΑ. (usA)

Εξελίξεις που σχετίζονται με τη ρομποτική από την έκθεση της NISTEP (Future Technology in Japan toward the Year 2030)στην Ιαπωνία: (Nistep)

- 2013-2014 : γεωργικά ρομπότ
- 2013-2017 : ρομπότ που μεριμνούν για τους ηλικιωμένους
- 2017 : ιατρικά ρομπότ που εκτελούν χαμηλή επεμβατική χειρουργική
- 2017-2019 : ρομπότ νοικοκυριού με πλήρης χρήση .
- 2019-2021 : νανορομπότ

Ρομποτικά δικαιώματα

Σύμφωνα με έρευνα που ανατέθηκε από το Γραφείο της Επιστήμης, του Ηνωμένου Βασιλείου (UK Office of Science) και το Horizon Scanning Centre της Innovation, τα ρομπότ θα μπορούσαν μια μέρα να ζητήσουν τα ίδια δικαιώματα ως άνθρωποι πολίτες. Η μελέτη προειδοποιεί επίσης ότι η αύξηση των ρομπότ θα μπορούσε να πιέσει τους πόρους και το περιβάλλον. (R.Rights)

1.14.3 Εφαρμογές

Η Caterpillar σχεδιάζει να αναπτύξει τηλεχειριζόμενα μηχανήματα και αναμένει να αναπτύξει πλήρως αυτόνομα βαριά ρομπότ μέχρι το 2021. (C.Robot)

Τα Ρομπότ χρησιμοποιούνται ολοένα και περισσότερο στην κατασκευή (από το 1960) και στην αυτοκινητοβιομηχανία που μπορεί να ανέλθει πάνω από το ήμισυ της εργασίας. Υπάρχουν ακόμη και «lights off» εργοστάσια , όπως ένα εργοστάσιο κατασκευής ηλεκτρολογίων IBM στο Τέξας που είναι 100 % αυτοματοποιημένο. (IBM-f)

Ρομπότ όπως το HOSPI χρησιμοποιούνται ως συνοδοί σε νοσοκομεία, με αποτέλεσμα την καλύτερη πρόσληψη φαρμάκων και βοήθεια των νοσοκόμων. (Hopsi) Άλλα καθήκοντα νοσοκομείου που εκτελούνται από τα ρομπότ είναι οι ρεσεψιονίστ , οι οδηγοί, οι αχθοφόροι βοηθοί και τα πιο εξειδικευμένα, όπως οι χειρουργικοί βοηθοί ρομπότ, όπως το Da Vinci. (D.Vinci)

Ρομποτικό Φίδι που θα βοηθήσει στην μελλοντική έρευνα στον Άρη

Ένα ρομπότ ο οποίο μοιάζει σαν φίδι θα μπορούσε να βοηθήσει τα συμβατικά τροχοφόρα ρόβερ στην έρευνα του Άρη, προσφέροντας καλύτερη κινητικότητα και ευελιξία .Οι ερευνητές της SITNEF, σε συνεργασία με τον Ευρωπαϊκό Οργανισμό Διαστήματος (ESA), αξιολογούν την σκοπιμότητα της έννοιας του crawling ρομπότ (έρποντος ρομπότ) που θα μπορούσε να χρησιμοποιηθεί σε συνδυασμό με ένα συμβατικό rover , και πιστεύουν ότι μια τέτοια αποστολή θα ανοίξει νέες δυνατότητες έρευνας .

«Η Ευελιξία είναι μια πρόκληση. Το Spirit rover χάθηκε αφότου κόλλησε στην άμμο στον Άρη. Τα οχήματα δεν μπορούν να πάνε σε πολλά από τα σημεία από τα οποία τα δείγματα πρέπει να ληφθούν», δήλωσε ο Pål Liljebäck , ένας ερευνητής στο SINTEF , εξηγώντας το μεγάλο πλεονέκτημα της ιδέας ενός φιδίσιου ρομπότ.

Σε αντίθεση με τα ρόβερ , τα ρομπότ φίδια μπορούσαν να έχουν πρόσβαση σε εξαιρετικά σφιχτές και ασταθείς θέσεις , συλλέγοντας δείγματα από περιοχές που τα ρόβερ δεν μπορούσαν ούτε να πλησιάσουν .

« Εξετάζουμε διάφορες εναλλακτικές λύσεις για να καταστεί δυνατόν το Rover και το ρομπότ να εργαστούν από κοινού " , δήλωσε ο Άξελ Transeth . "Αφού το rover έχει μια ισχυρή πηγή ενέργειας , θα μπορούσε να προσφέρει στο ρομπότ φίδι με ενέργεια μέσα από ένα καλώδιο που εκτείνεται μεταξύ του rover και του ρομπότ . Εάν το ρομπότ θα έπρεπε να χρησιμοποιεί τις μπαταρίες του, τότε θα ξέμενε από τροφοδοσία και θα το χάναμε » , είπε .

Οι ερευνητές προβλέπουν το ρομπότ φίδι θα μπορούσε να αναπτυχθεί από έναν από τους βραχίονες του ρόβερ που θα μπορούσε να αποσπαστεί , εάν χρειαζόταν και λειτουργήσει ανεξάρτητα . Μετά την εκτέλεση των καθηκόντων του , το φίδι , εξακολουθώντας να είναι συνδεδεμένο με το rover μέσω ενός καλωδίου , θα μπορούσε να συνδεθεί ξανά στο κύριο rover . Η επικοινωνία μεταξύ του ζεύγους θα διευκολυνθεί μέσω σημάτων που θα μεταδίδονται μέσω του καλωδίου .

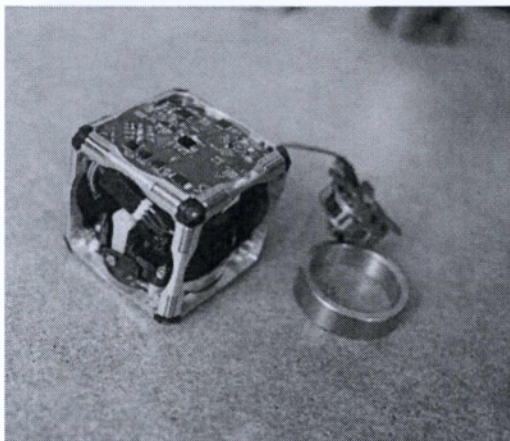
Ο συνδυασμός ενός rover και του φιδίσιου βραχίονα θα επιτρέψει την αξιοποίηση των πλεονεκτημάτων και των δύο συστημάτων - την ικανότητα του rover να διασχίζει μεγάλες αποστάσεις και την ικανότητα του φιδιού να φτάνει σε δυσπρόσιτα σημεία .

" Η σύνδεση μεταξύ του ρομπότ και το rover σημαίνει επίσης ότι το ρομπότ φίδι θα είναι σε θέση να βοηθήσει το όχημα αν κολλήσει. Σε μια τέτοια κατάσταση , το ρομπότ θα μπορούσε να χαμηλώνει στο έδαφος και να συσπειρώνεται σε έναν βράχο έτσι ώστε να επιτρέπει στο rover να ξεκολλήσει μέσω του βαρούλκου του καλωδίου, το οποίο συνήθως το rover θα χρησιμοποιούσε για να τραβήξει το ρομπότ φίδι προς το μέρος του.» , είπε ο Liljebäck.

Τα αποτελέσματα της μελέτης σκοπιμότητας θα πρέπει να είναι διαθέσιμες στο τέλος του 2013 .

Μέχρι στιγμής , η NASA έχει προσγειωθεί τέσσερις ηλιακής ενέργειας ρόβερ στον Άρη , όλα εξοπλισμένα με έξι τροχούς και ρομποτικούς βραχίονες για να λαμβάνουν δείγματα εδάφους και να λειτουργούν κάμερες. (S.robot)

Ερευνητές του MIT προτείνουν αυτο-συναρμολογούμενα ρομπότ



Εικ.133 Πρωτότυπο για νέο συναρμολογούμενο ρομπότ

Ο επιστημονικός ερευνητής John Romanishin, στην Επιστήμη των Υπολογιστών του MIT και Εργαστηρίου Τεχνητής Νοημοσύνης (CSAIL) ,έχει από καιρό οραματιστεί για το άμεσο μέλλον ρομποτικές ενότητες γνωστές ως M-Blocks. Ο Romanishin έχει συνεργαστεί με τον καθηγητή του, Daniela Rus, και τον συνάδελφό του, μεταδιδακτορικό Kyle Gilpin, προς τους πρωτότυπους ρομποτικούς κύβους που δεν έχουν εξωτερικά κινούμενα μέρη και είναι σε θέση να αναρριχηθούν πάνω, γύρω ,ακόμα και να κάνουν άλματα πάνω στους άλλους κύβους.

Μέχρι τώρα, τα ρομπότ εξαρτιόνταν από βραχίονες ή προσαρτήματα να για κινούνται . «Θέλαμε μια απλούστερη προσέγγιση», λέει ο Ουκρανός, που χρησιμοποιεί λιγότερα κινούμενα μέρη. Μέσα σε κάθε M-Block είναι ένα σφόνδυλο που περιστρέφεται στις 20.000 στροφές ανά λεπτό, δημιουργώντας αρκετή στροφορμή όταν φρενάρει, με αποτέλεσμα οι κύβοι συναρμολογούνται σε νέους σχηματισμούς. Σε κάθε

πλευρά και γωνία των κύβων υπάρχουν μαγνήτες, που συνδέουν φυσικά τους κύβους όταν ωθούνται από το σφόνδυλο.

«Αυτό είναι συναρπαστικό, επειδή ένα ρομπότ σχεδιασμένο για ένα ενιαίο έργο έχει μια σταθερή αρχιτεκτονική», λέει ο καθηγητής Rus. Μια «σταθερή αρχιτεκτονική» μπορεί να εκτελέσει ένα ενιαίο έργο, αλλά κάτω από αδιευκρίνιστα ή απρόβλεπτα περιβάλλοντα θα έχει κακές επιδόσεις αν δεν είναι σε θέση να προσαρμοστεί. Έτσι, «είναι καλύτερο να μελετούμε modular (μορφωματικά) ρομπότ που μπορούν να επιτύχουν ό, τι σχήμα είναι απαραίτητο για το χειρισμό, τη πλοήγηση ή την αίσθηση για τις ανάγκες μιας εργασίας.»

«Είναι ένα από αυτά τα πράγματα που η modular-ρομποτική κοινότητα έχει προσπαθήσει να επιτύχει για μεγάλο χρονικό διάστημα.» Πώς τα M-Blocks ξεχωρίζουν, μόνο με την παρουσίαση πολλών τρόπων μετακίνησης; «Δεν είναι απλά κύβοι που κτυπάνε γύρω γύρω, αλλά πολλαπλές εργασίες μαζί, σύροντας ο ένας τον άλλο. Αυτό το είδος της συνεταιριστικής συμπεριφοράς της ομάδας είναι άνευ προηγουμένου.»

Δεδομένου ότι το ρομποτικό σύστημα γίνεται ολοένα και πιο καθορισμένο και απλουστευμένο, ο στόχος θα είναι σμίκρυνση, σμήνη από αυτο-συναρμολογούμενα microbots σε πιο πολύπλοκα αντικείμενα και δομές.

«Θέλουμε εκατοντάδες κύβους, διασκορπισμένοι τυχαία στο πάτωμα, να είναι σε θέση να εντοπίσει ο ένας τον άλλο, να συγχωνευτούν και να μετατραπούν αυτόνομα σε μια καρέκλα ή μια σκάλα ή ένα γραφείο, ανάλογος με το τι θα τους ζητηθεί», λέει ο Romanishin.

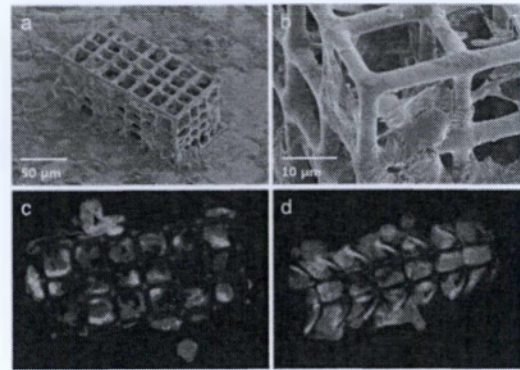
Ίσως να είναι απλά θέμα χρόνου μέχρι ολόκληρα κτίρια είναι αυτο-συναρμολογούμενα και ανακατασκευασμένα με την ίδια λογική που απαιτείται για να δημιουργηθούν αυτοί οι πρωτότυποι ρομποτικοί κύβοι. (Mit.N.O)

Μαγνητικά νανορομπότ που θα μπορούν να μεταφέρουν φάρμακα στο μυαλό μας

Αυτά τα μικροσκοπικά κλουβιά , με διαστάσεις 100 μικρομέτρων μήκους και 40 μικρομέτρων εύρους , που μπορεί να μοιάζουν πολύ, αλλά είναι τα νέα ημι - φορτηγά στοχευόμενης παροχής φαρμάκων . Αναπτύχθηκαν από μια ομάδα Κινέζων ερευνητών , σε συνδυασμό με Ελβετικά και Νότια Κορεάτικα ινστιτούτα . Τα nickel-coated (νικέλιο –

επίστρωση) microbots κατευθύνονται ασύρματα με ηλεκτρομαγνητικά πεδία, και χάρη σε αυτόν τον εξωτερικό έλεγχο, τα microbots μπορούν να μεταφέρουν πολύτιμα φορτία ακριβώς στην περιοχή που ο οργανισμός χρειάζεται, συμπεριλαμβανομένων των ευαίσθητων περιοχών, όπως το μυαλό ή τα μάτια.

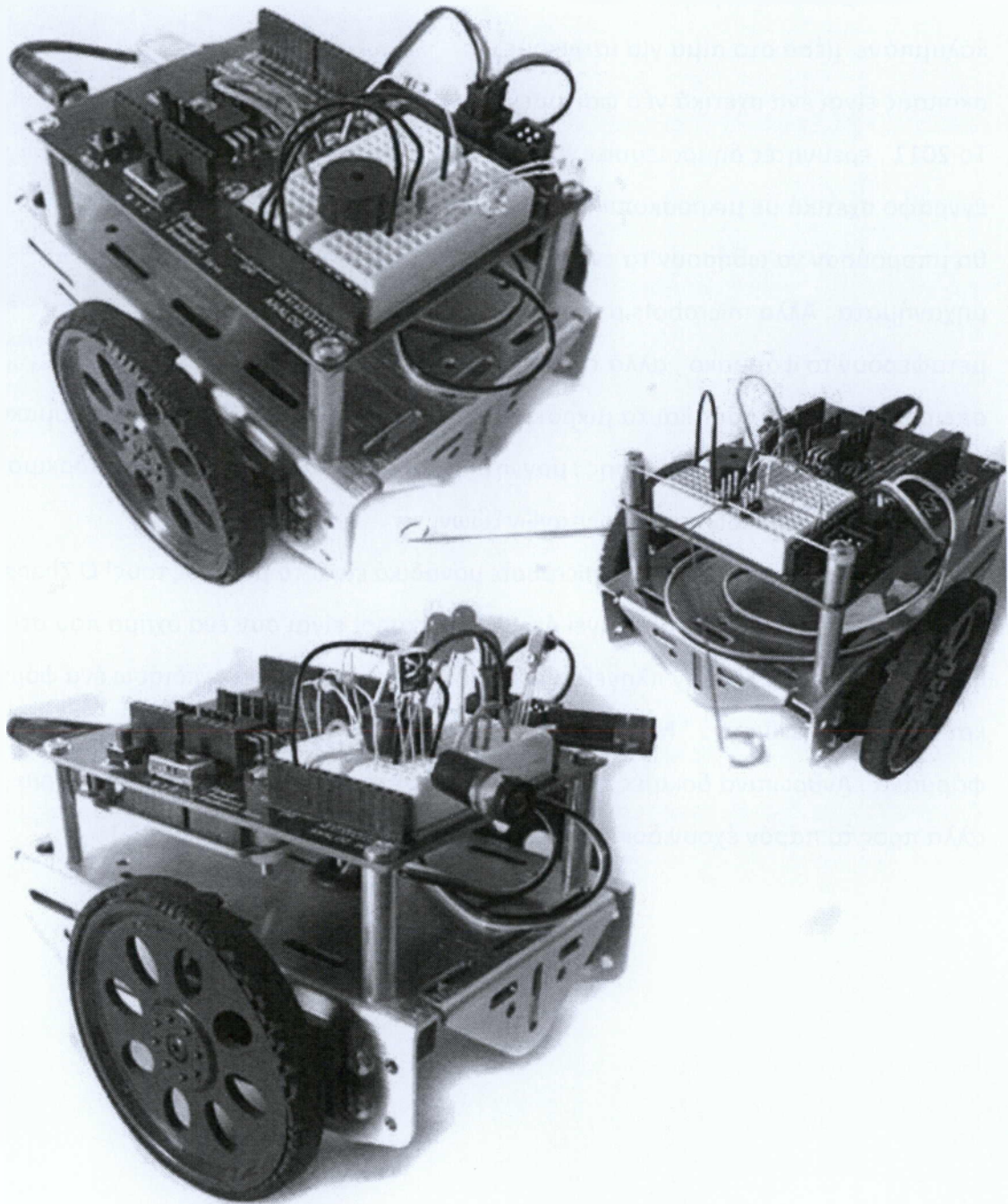
Μικροσκοπικά ρομπότ που κολυμπάνε μέσα στο αίμα για ιατρικούς σκοπούς είναι ένα σχετικά νέο φαινόμενο. Το 2011, ερευνητές δημοσίευσαν ένα έγγραφο σχετικά με μικροσκοπικά μοτέρ που θα μπορούσαν να ωθήσουν τα εν λόγω μηχανήματα. Άλλα microbots μπορούν να μεταφέρουν το φάρμακο, αλλά το σπειροειδές σχήμα τους και τα μικρότερα σώματά τους περιορίζουν πόσο φάρμακο μπορούν να μεταφέρουν. Επίσης, μαγνητικά κατευθυντήρια ρομπότ έχουν δοκιμαστεί και ποιο παλιά στο εσωτερικό ζωντανών ζώων.



Εικ.134 Μαγνητικά νανορομπότ, που μπορούν να εισχωρήσουν φάρμακα στον εγκέφαλο.

Αυτό που κάνει αυτά τα microbots μοναδικό είναι το μέγεθος τους! Ο Zhang Li, ένας ερευνητής για το έργο, εξηγεί ότι «ένα microbot είναι σαν ένα όχημα που στέλνει φάρμακα απευθείας στην πληγείσα περιοχή. Και εγώ θέλω να σχεδιάσω ένα φορηγό και όχι ένα αυτοκίνητο.» Μεγαλύτερα ρομπότ σημαίνουν και περισσότερα παραδοτέα φάρμακα. Ανθρώπινα δοκιμές αυτών των ρομπότ είναι πιθανόν δεκαετίες μακριά, αλλά προς το παρόν έχουν δοκιμαστεί σε κουνέλια και ποντίκια. (Nano.R.)

2. ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ



2.1 BOE Shield-Bot

2.1.1 Λίγα λόγια για το BOE Shield-Bot

Το αρχικό Boe-Bot ρομπότ με τον BASIC Stamp 2 εγκέφαλο εισήχθη από την Parallax Inc το 1999. Το Boe-Bot απέκτησε άμεση δημοτικότητα, το οποίο συνέχισε να βελτιώνετε και όλο και περισσότερα σχολεία το συμπεριλάμβαναν στα προγράμματα ρομποτικής, ηλεκτρονικών, προγραμματισμού και τη φυσικής.

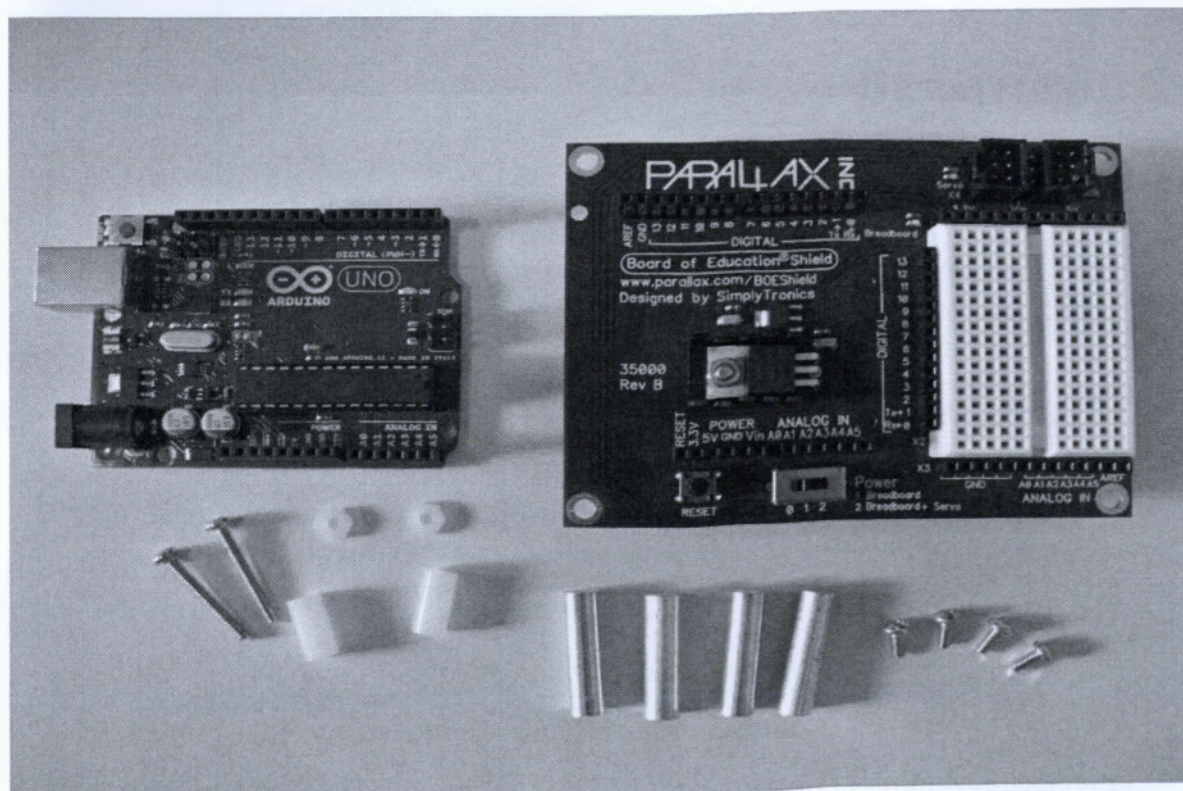
Ο Arduino μικροελεγκτής έφθασε στη σκηνή το 2005, και έχει ήδη το δικό του επίπεδο δημοτικότητάς , ειδικά στην DIY (do-it-yourself) κοινότητα χόμπι. Το υλικό και το λογισμικό Arduino έχει πολλές λειτουργικές ομοιότητες με τον BASIC Stamp μικροελεγκτή και λογισμικό. Η Parallax συνεργάστηκε με τη SimplyTronics για να σχεδιάσει το Board of Education® Shield, το οποίο καθιστά το υλικό Arduino συμβατό με το Boe-Bot. Μαζί, η λειτουργική ομοιότητα και η συμβατότητα του υλικού δημιούργησε πολύ απλά σχέδια, porting προγραμματιστικών και εκπαιδευτικών παραδειγμάτων Ρομποτικής με το Boe-Bot.

2.2 Εγκατάσταση του Board of Education Shield

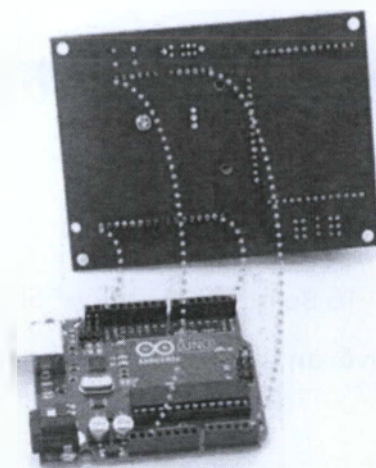
Το Board of Education® Shield καθιστά εύκολο την κατασκευή κυκλωμάτων και την σύνδεση των servos (βοηθητικά μοτέρ) στη μονάδα Arduino.

Κατάλογος εξαρτημάτων:

- (1) Arduino
- (1) Board of Education Shield
- (4) 2,5cm στρογγυλά διαχωριστικά αλουμινίου
- (4) Επίπεδης κεφαλής βίδες, 0,6cm 4-40
- (2) 1,3cm στρογγυλά, νάιλον διαχωριστικά
- (2) Νάιλον παξιμάδια, 4-40
- (2) Επίπεδες βίδες, 2,2cm, 4-4

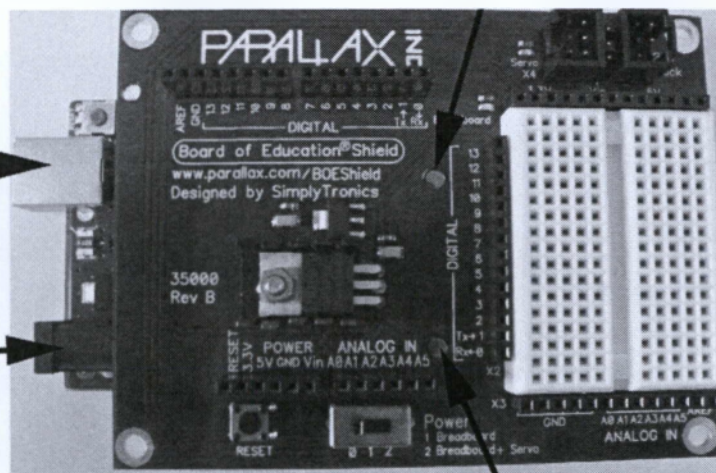


Οι τέσσερις ομάδες των ακίδων κάτω από το Board of Education® Shield συνδέονται μέσα στις τέσσερις υποδοχές του Arduino. Υπάρχουν επίσης τρεις τρύπες στο Board of Education® Shield που ευθυγραμμίζονται με τις οπές στη μονάδα Arduino, που έχουν σχεδιαστεί για να συνδέονται οι δύο μονάδες με τις βίδες και τα νάιλον διαχωριστικά.



Η τοποθέτηση των υλικών διαφέρει λίγο για τα διάφορα μοντέλα Arduino. Μερικά μπορεί να χωρέσουν μόνο ένα ή δύο διαχωριστικά νάιλον για τη συγκράτηση των μονάδων μεταξύ τους. Αυτό είναι αποδεκτό, αλλά θα πρέπει να προσέξουμε ποιες τρύπες μπορούμε να χρησιμοποιήσουμε πριν την τοποθέτηση του Board of Education® Shield. Στη συγκεκριμένη περίπτωση, χρησιμοποιούμε το μοντέλο Uno R3.

Οι θύρες του Arduino για τη τροφοδοσία και το προγραμματισμό δεν εφάπτονται στο Board of Education Shield



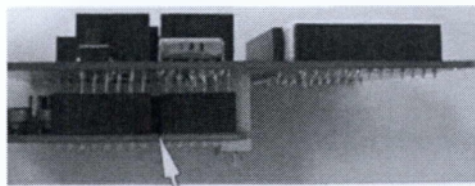
Σιγουρευόμαστε ότι οι τρύπες στο Board of Education Shield ευθυγραμμίζονται με αυτές του Arduino

Σε κάθε τρύπα που χρησιμοποιούμε στη μονάδα Arduino μας, βάζουμε μια 2,2cm βίδα και μετά σε κάθε βίδα τοποθετούμε και ένα νάιλον διαχωριστικό. Αφού έχουμε ευθυγραμμίσει τις ακίδες και τις βίδες να ταιριάζουν και από τις 2 μονάδες, πιέζουμε απαλά τις δυο μονάδες μεταξύ τους μέχρις ότου οι ακίδες (πύροι) να είναι

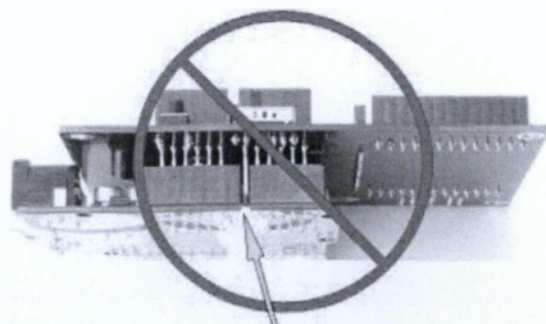
καλά στερεωμένοι στις υποδοχές τους. Οι υποδοχές δεν θα καλύπτουν εντελώς τις ακίδες, περίπου 5 mm από τις ακίδες θα εξακολουθεί να είναι εκτεθειμένο ανάμεσα στο κάτω μέρος της ασπίδας και την κορυφή των υποδοχών. Έπειτα περνάμε ένα νάιλον παξιμάδι σε κάθε βίδα και σφίγγουμε απαλά.



Ελέγχουμε, για να είστε απολύτως βέβαιοι, ότι οι ακίδες εφάπτονται σωστά στις υποδοχές. Είναι δυνατόν να μην ευθυγραμμιστούν σωστά οι ακίδες, με αποτέλεσμα να προκαλέσει βλάβη στην μονάδα όταν είναι ενεργοποιημένη.

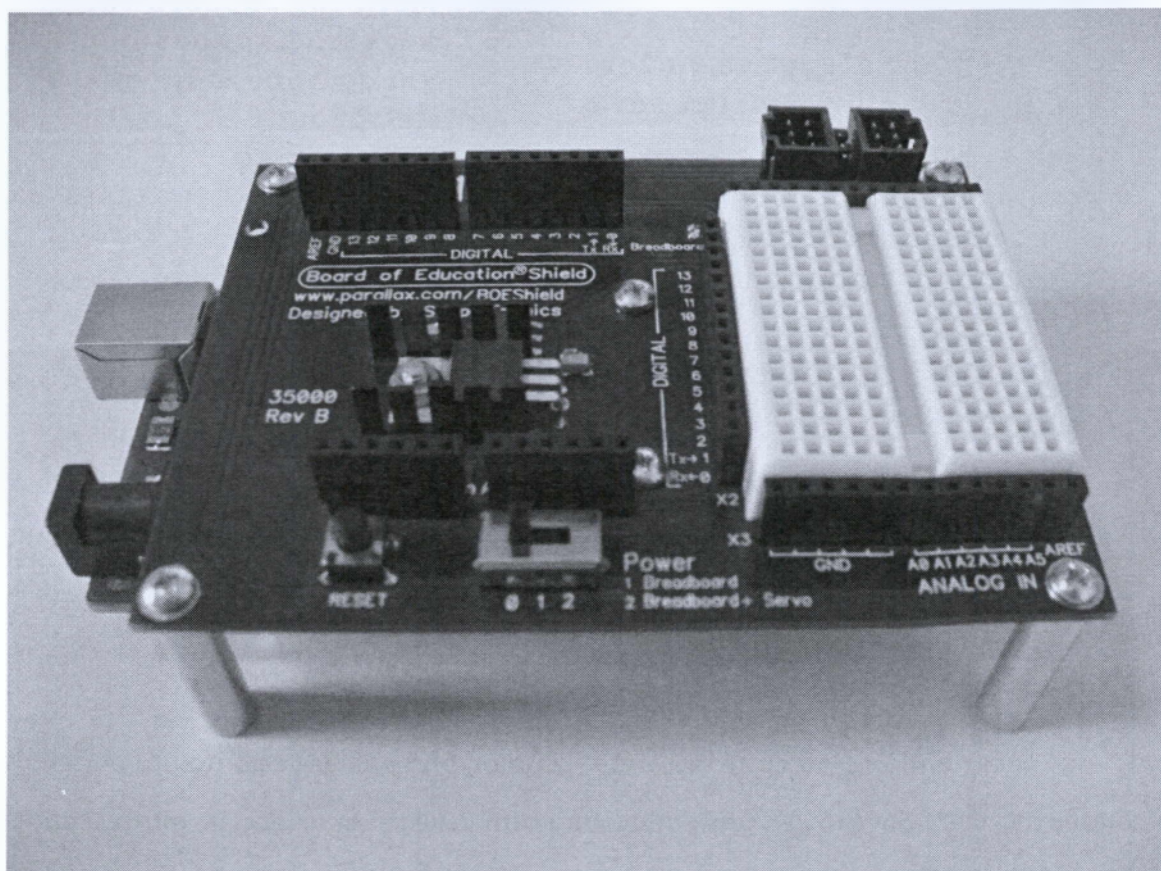


Σωστό: Το κενό στα pins ευθυγραμμίζεται με το κενό των υποδοχών.



Λάθος: Υπάρχει ένα pin στο κενό των υποδοχών.

Τέλος για να διατηρήσουμε τις δυο συνδεδεμένες μονάδες μακριά από οποιαδήποτε επιφάνεια και να είναι έτυμες για την σύνδεση τους στο σκελετό του ρομπότ μας, τοποθετούμε διαχωριστικά σε κάθε γωνία της πλακέτας. Οπότε περνάμε μια 0,6cm βίδα μέσα από κάθε γωνιακή τρύπα τις πλακέτας και βιδώνουμε σε κάθε βίδα ένα 2,5cm αλουμινένιο διαχωριστικό.



2.3 Κατασκευή και δοκιμή των ενδεικτικών LED

Οι ενδεικτικές λυχνίες είναι ένα μέσο για να βλέπουμε μια αναπαράσταση του τι συμβαίνει στο εσωτερικό μιας συσκευής, ή τα πρότυπα επικοινωνίας μεταξύ των δύο συσκευών.

2.3.1 Παρουσίαση της Αντίστασης

Ένας αντιστάτης είναι ένα στοιχείο που αντιστέκεται στην ροή του ηλεκτρικού ρεύματος, δηλαδή το ρεύμα. Κάθε αντίσταση έχει μια τιμή που χαρακτηρίζει πόσο έντονα αντιστέκεται τη ροή του ρεύματος. Αυτή η τιμή της αντίστασης ονομάζεται ohm, και το σήμα για την ohm είναι το ελληνικό γράμμα ωμέγα (Ω).

Η αντίσταση έχει δύο σύρματα (που ονομάζονται «leads», αγωγοί και προφέρονται «leeds»), τα οποία προεξέχουν από το κάθε άκρο της. Η κεραμική θήκη μεταξύ των δύο αγωγών είναι το μέρος που αντιστέκεται τη ροή του ρεύματος. Τα περισσότερα σχέδια κυκλωμάτων χρησιμοποιούν την ακανόνιστη γραμμή σύμβολων με μια ετικέτα αριθμών για να υποδείξουν μια αντίσταση συγκεκριμένης τιμής, όπως τη 470 Ω αντίσταση σε αυτή την περίπτωση. Αυτό ονομάζεται σχηματικό σύμβολο.

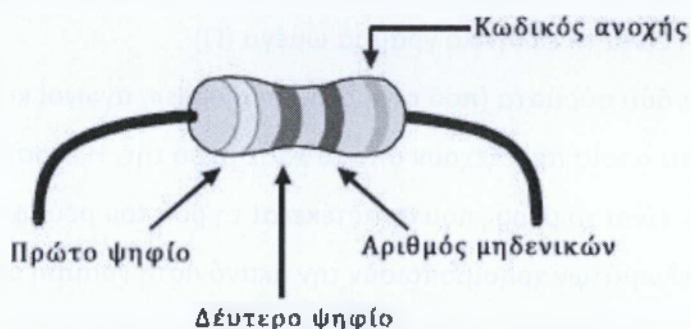


Οι αντιστάσεις που χρησιμοποιούμε έχουν χρωματιστές ρίγες που δείχνουν ποιες είναι οι αξίες αντοχή τους. Υπάρχουν διαφορετικοί συνδυασμοί χρωμάτων για κάθε τιμή της αντίστασης. Για παράδειγμα, ο κώδικας χρώματος για την αντίσταση 470 Ω είναι κίτρινο-ιώδες-καφέ.

Μπορεί να υπάρχει και μια τέταρτη λωρίδα που υποδεικνύει την ανοχή της αντίστασης. Η ανοχή μετράται σε επί τοις εκατό, και δείχνει πόσο μακριά είναι η αληθινή αντίσταση του είδους σε σχέση με την σημειωμένη αντίσταση. Η τέταρτη λωρίδα θα μπορούσε να είναι χρυσή (5%), ασημένια (10%) ή να μην υπάρχει λωρίδα (20%).

Κάθε γραμμή χρωμάτων για την υπόθεση της αντίστασης αντιστοιχεί σε έναν αριθμό, όπως παρατίθενται στον παρακάτω πίνακα.

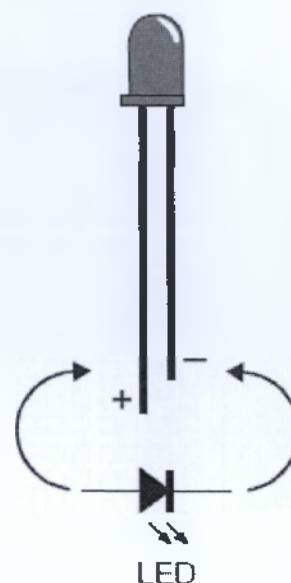
1. Κώδικας αξιών χρωμάτων αντίστασης										
Μονάδα	0	1	2	3	4	5	6	7	8	9
Χρώμα	μαύρο	καφέ	κόκκινο	πορτοκαλί	κίτρινο	πράσινο	μπλε	ιώδης	γκρι	άσπρο



2.3.2 Παρουσίαση του LED

Μια δίοδος είναι μια μονόδρομη βαλβίδα ηλεκτρικού ρεύματος, και μια δίοδο εκπομπής φωτός (LED), εκπέμπει φως όταν το ρεύμα περνά μέσα από αυτό. Δεδομένου ότι ένα LED είναι μια μονόδρομη βαλβίδα ρεύματος, θα πρέπει να σιγουρευτούμε για το σωστό τρόπο σύνδεσης, έτσι ώστε να λειτουργήσει όπως θα έπρεπε.

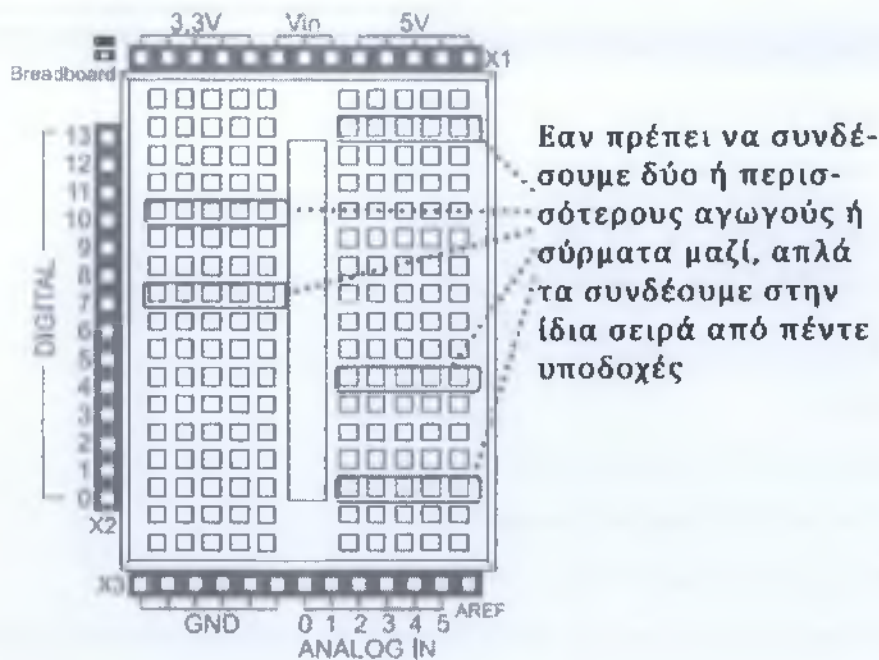
Ένα LED έχει δύο τερματικά: την άνοδο και την κάθοδο. Ο μόνυβδος (καλωδιο) ανόδου είναι επισημασμένο με το θετικό πρόσημο (+) στο τμήμα σχεδίασης, και είναι το πλατύ μέρος του τριγώνου στο σχηματικό σύμβολο. Ο μόνυβδος καθόδου είναι η καρφίτσα επισημασμένη με το αρνητικό πρόσημο (-), και είναι η γραμμή κατά μήκος του σημείου του τριγώνου στο σχηματικό σύμβολο.



Όταν χτίζουμε ένα LED κύκλωμα, θα πρέπει να βεβαιωθούμε ότι τα καλώδια ανόδου και καθόδου συνδέονται με το κύκλωμα σωστά. Μπορείτε να τα ξεχωρίσετε από το σχήμα της πλαστικής θήκης του LED. Αν κοιτάξετε προσεκτικά την θήκη, είναι ως επί το πλείστον στρογγυλή, αλλά υπάρχει ένα μικρό επίπεδο σημείο κοντά στο δεξιό καλώδιο, το οποίο μας υποδεικνύει ότι είναι η κάθοδος. Επίσης τα καλώδια του LED είναι σε διαφορετικά μήκη, συνήθως, το μικρότερο καλώδιο συνδέεται με την κάθοδο(-) και αντίστοιχα το μεγαλύτερο με την άνοδο(+).

2.3.3 Παρουσίαση της Prototyping Area

Η λευκή περιοχή με τις πολλές τετράγωνες υποδοχές σε αυτό ονομάζεται solderless breadboard. Αυτό το breadboard έχει 17 σειρές υποδοχών. Σε κάθε σειρά, υπάρχουν δύο ομάδες πενταπλών υποδοχών που χωρίζονται από μια τάφρο στη μέση. Όλες οι υποδοχές σε μια πενταπλή ομάδα υποδοχών συνδέονται μεταξύ τους υπόγεια με ένα αγώγιμο μεταλλικό κλιπ. Έτσι, δύο καλώδια που έχουν συνδεθεί στην ίδια ομάδα πενταπλών υποδοχών παρουσιάζουν ηλεκτρική επαφή. Με τον ίδιο τρόπο θα συνδέουμε και εμείς τα στοιχεία μας, όπως ένα LED και μια αντίσταση, για την κατασκευή κυκλωμάτων.



Επίσης η πρωτότυπη περιοχή έχει μαύρες υποδοχές κατά μήκος της πάνω, κάτω και της αριστερής πλευράς.

- Πάνω: οι υποδοχές έχουν τρεις τάσεις τροφοδοσίας για το breadboard: 3,3 V, Vin (τάση εισόδου είτε από μπαταρία ή το καλώδιο προγραμματισμού), και 5 V.
- Κάτω-αριστερά: Οι πρώτες έξι υποδοχές κατά μήκος της κάτω-αριστερής πλευράς είναι ακροδέκτες γείωσης, επισημαίνοντας GND, σκεφτείτε τους ως τάση τροφοδοσίας που είναι 0 V. Συλλογικά, τα 3.3V, Vin, 5V και GND ονομάζονται ακροδέκτες ηλεκτρικής ενέργειας και θα πρέπει να χρησιμοποιηθούν για να τροφοδοτηθεί το κύκλωμα μας με ηλεκτρική ενέργεια.
- Κάτω-δεξιά: Οι υποδοχές «ANALOG IN» κατά μήκος της κάτω-δεξιάς πλευράς είναι για τη μέτρηση μεταβλητών τάσεις . Αυτές συνδέονται με τις «ANALOG IN» υποδοχές του Arduino.
- Αριστερά: Οι υποδοχές DIGITAL στην αριστερή πλευρά έχουν ετικέτες από 0 έως 13. Θα χρησιμοποιήσουμε αυτές τις υποδοχές για να συνδέσουμε το κύκλωμα μας στις ψηφιακές ακίδες εισόδου / εξόδου του Arduino.

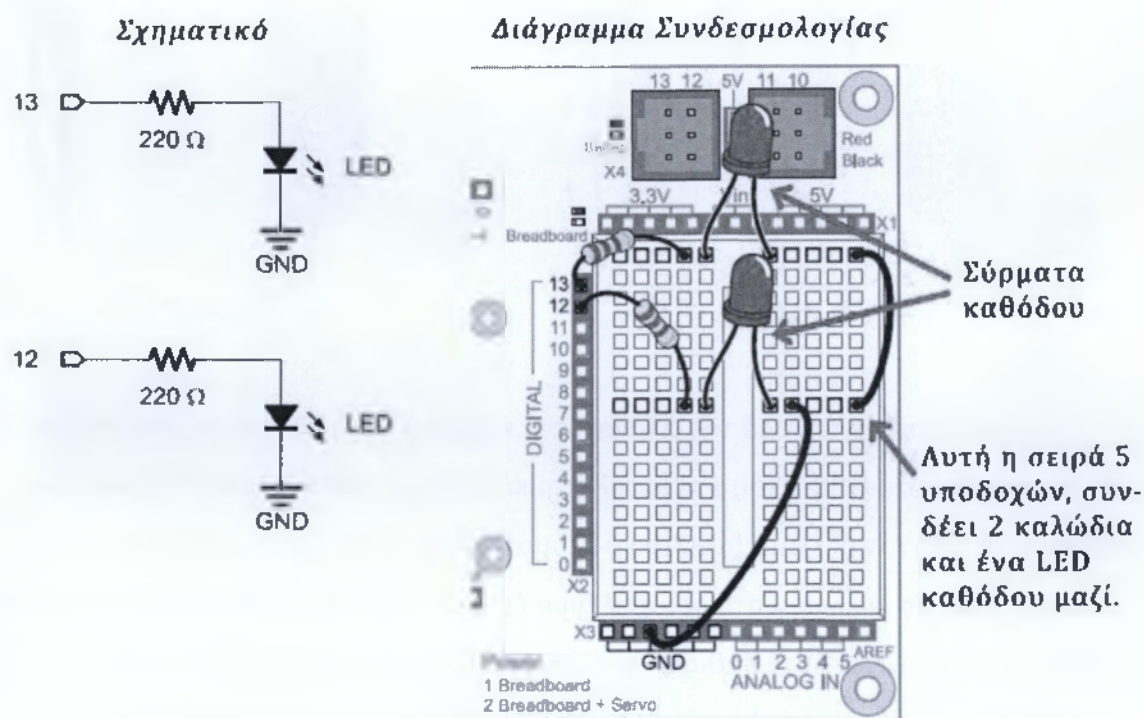
2.3.4 Δοκιμή LED κυκλώματος

Εξαρτήματα κυκλώματος:

(2) Κόκκινα LED

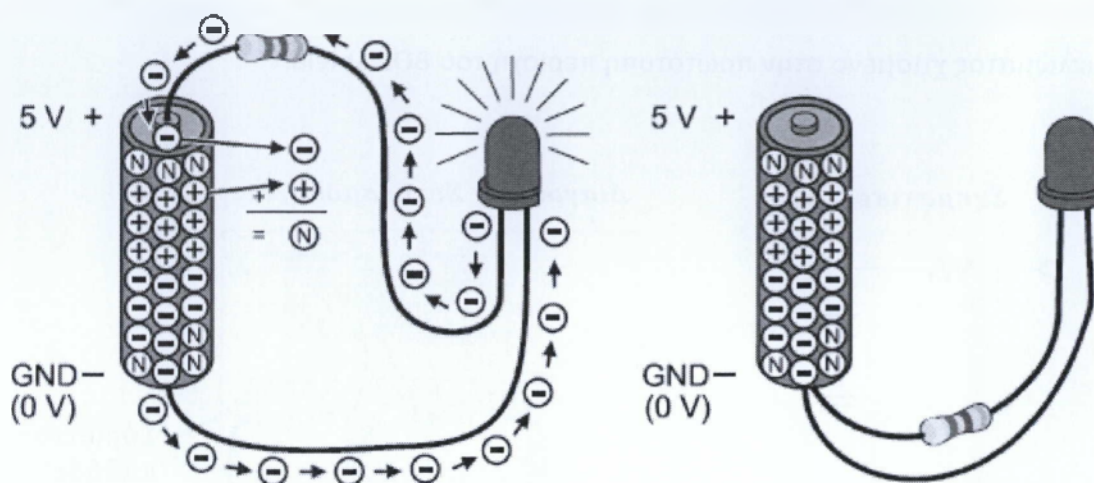
(2) Αντιστάσεις, 220 Ω (κόκκινο-κόκκινο-καφέ)

Η παρακάτω εικόνα δείχνει σχηματικά το κύκλωμα μιας ενδεικτικής λυχνίας LED στα αριστερά, και στα δεξιά ένα παράδειγμα διαγράμματος καλωδίωσης του κυκλώματος χτισμένο στην πρωτότυπη περιοχή του BOE Shield.



Βεβαιωνόμαστε ότι τα σύρματα καθόδου των LED μας είναι συνδεδεμένα στο GND. Κάθε καλώδιο καθόδου πρέπει να είναι συνδεδεμένο στην ίδια γραμμή όπως και τα καλώδια που πηγαίνουν στις υποδοχές GND. Τέλος βεβαιωνόμαστε ότι τα μακρύτερα καλώδια ανόδου είναι συνδεδεμένα στην ίδια γραμμή με τις αντιστάσεις.

Η επόμενη εικόνα μας δώνει μια ιδέα για το τι συμβαίνει όταν προγραμματίζουμε τον Arduino για τον έλεγχο του κυκλώματος LED. Φανταστείτε ότι έχετε μια μπαταρία 5 volt (5 V). Το Board of Education Shield έχει μια συσκευή που ονομάζεται ρυθμιστής τάσης, το οποίο τροφοδοτεί 5 volts στις υποδοχές που έχουν ονομαστεί 5V. Όταν συνδέουμε το τέλος της ανόδου του LED κυκλώματος σε 5 V, είναι σαν να το συνδέουμε στο θετικό πόλο μιας μπαταρίας 5 V. Όταν συνδέουμε το κύκλωμα σε GND, είναι σαν να το συνδέουμε στον αρνητικό πόλο μιας μπαταρίας 5 V.



Στην αριστερή πλευρά της εικόνας, ένα καλώδιο του LED είναι συνδεδεμένο σε 5 V και το άλλο στο GND. Έτσι, 5 V της ηλεκτρικής πίεσης προκαλεί τα ηλεκτρόνια να ρέουν μέσα από το κύκλωμα (ηλεκτρικό ρεύμα), και αυτό το ρεύμα αναγκάζει το LED να εκπέμπει φως. Το κύκλωμα στη δεξιά πλευρά έχει και τα δύο άκρα του LED κυκλώματος συνδεδεμένα με GND. Αυτό καθιστά την ίδια τάση (0 V) και στα δύο άκρα του κυκλώματος. Απουσία της ηλεκτρικής πίεσης = καθόλου ρεύμα = κανένα φως.

2.3.5 Πως ένα πρόγραμμα ανάβει και σβήνει τα LED

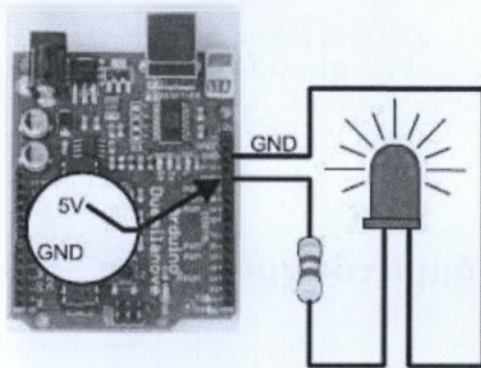
Ας ξεκινήσουμε με ένα πρόγραμμα που κάνει το κύκλωμα LED, που συνδέεται με το ψηφιακό pin 13, να αναβοσβήνει. Πρώτον, το πρόγραμμα θα πρέπει να δώσει εντολή στον Arduino για να ρυθμίσει την κατεύθυνση του pin 13 σε output (εξόδου), χρησιμοποιώντας τη εντολή `pinMode (pin, mode)`. Η παράμετρος `pin` είναι ο αριθμός

ενός ψηφιακού pin I / O, και το mode (λειτουργία) πρέπει να είναι είτε INPUT (εισόδου) ή OUTPUT (εξόδου).

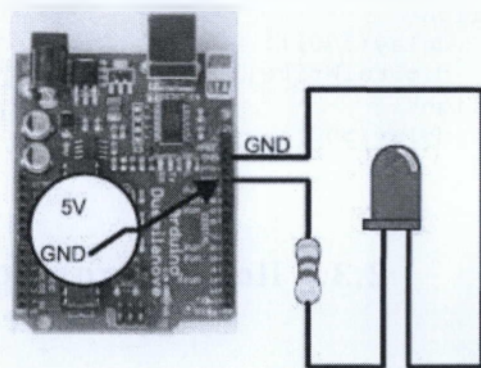
```
void setup()                                     // Built-in initialization block
{
  pinMode(13, OUTPUT);                          // Set digital pin 13 -> output
}
```

Τώρα που η ψηφιακή pin 13 έχει οριστεί ως output, μπορούμε να χρησιμοποιήσουμε την εντολή digitalWrite για να ανάβουμε και να σβήνουμε τη λυχνία LED. Ρίξτε μια ματιά στην παρακάτω εικόνα. Στα αριστερά, το digitalWrite (13, HIGH) κάνει τον μικροελεγκτή του Arduino να συνδέει την ψηφιακή pin 13 σε 5 V, το οποίο ανάβει το LED. Στα δεξιά, το digitalWrite (13, LOW) συνδέει το pin 13 σε GND (0 V) για να σβήσει η λυχνία LED.

```
digitalWrite(13, HIGH);
```



```
digitalWrite(13, LOW);
```



Παρακάτω, έχουμε τη λειτουργία του βρόχου (loop). Πρώτον, η εντολή digitalWrite (13, HIGH) ανάβει το LED και η εντολή delay(500) το κρατά αναμμένο για μισό δευτερόλεπτο. Στη συνέχεια η digitalWrite (13, LOW) σβήνει το LED, η οποία επίσης ακολουθείται από μια delay(500). Δεδομένου ότι είναι μέσα στο μπλοκ της λειτουργία του βρόχου, οι δηλώσεις θα επαναλαμβάνονται αυτόματα, με αποτέλεσμα το LED να αναβοσβήνει (on / off) μια φορά κάθε δευτερόλεπτο.

```
void loop()                                     // Main loop auto-repeats
{
  digitalWrite(13, HIGH);                      // Pin 13 = 5 V, LED emits light
  delay(500);                                  // ..for 0.5 seconds
  digitalWrite(13, LOW);                      // Pin 13 = 0 V, LED no light
  delay(500);                                  // ..for 0.5 seconds
}
```

2.3.6 Παράδειγμα Προγράμματος: HighLowLed

Επανασυνδέουμε τον Arduino με το PC μας και συνδυάζοντας τα 2 παραπάνω διαγράμματα γραφούμε τον παρακάτω κώδικα.

```

/*
Robotics with the BOE Shield - HighLowLed
Turn LED connected to digital pin 13 on/off once every second.
*/

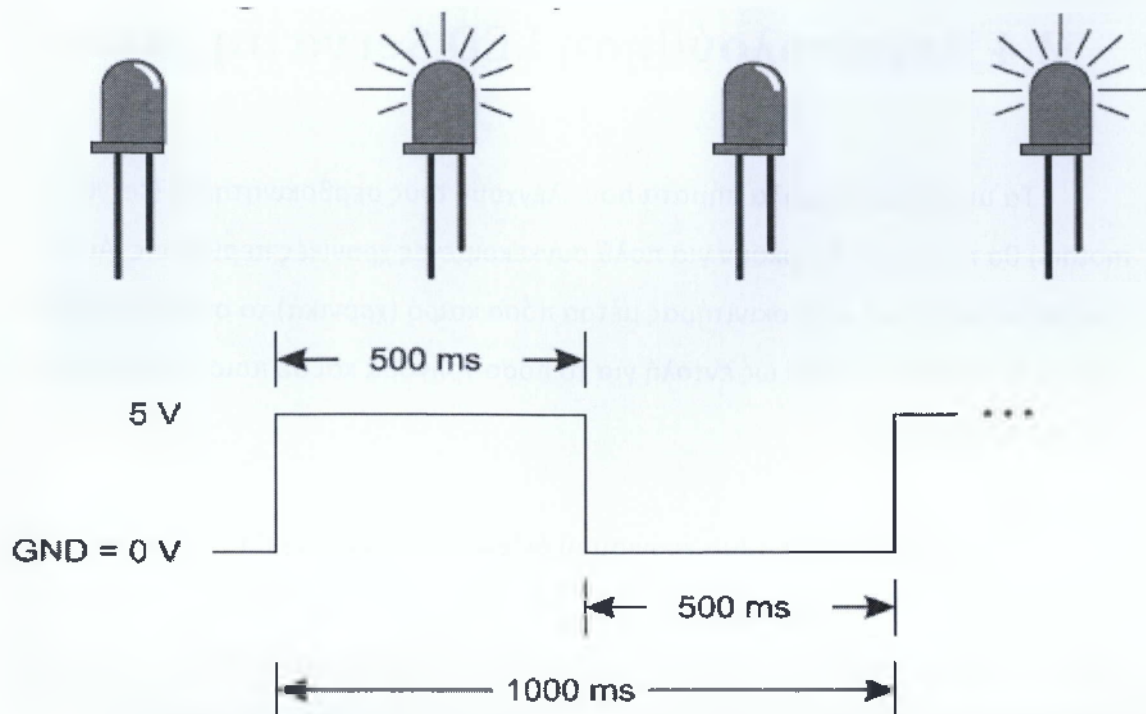
void setup()                                // Built-in initialization
block
{
  pinMode(13, OUTPUT);                       // Set digital pin 13 ->
output
}

void loop()                                  // Main loop auto-repeats
{
  digitalWrite(13, HIGH);                    // Pin 13 = 5 V, LED emits
light
  delay(500);                                // ..for 0.5 seconds
  digitalWrite(13, LOW);                     // Pin 13 = 0 V, LED no
light
  delay(500);                                // ..for 0.5 seconds
}

```

2.3.7 Παρουσίαση του διαγράμματος χρονισμού

Ένα διάγραμμα χρονισμού (Timing Diagram) είναι μία γραφική παράσταση που συσχετίζει τα ψηλά και χαμηλά στάδια ενός σήματος σε σχέση με το χρόνο. Το παρακάτω διάγραμμα χρονισμού μας δείχνει ένα 1000 ms κομμάτι των HIGH (5 V) και LOW (0 V) σημάτων από το σχεδιάγραμμα HighLowLed. Έτσι μπορούμε να δούμε καλύτερα πώς η delay(500) ελέγχει το ρυθμό εναλλαγής φωτεινότητας.



Μπορούμε πολύ εύκολα να μειώσουμε τον χρόνο καθυστέρησης από 500 σε 250 (`delay(250)`), έτσι ώστε να αναβοσβήνουν πιο γρήγορα. Επίσης μπορούμε να κάνουμε και τα 2 LED να αναβοσβήνουν την ίδια στιγμή ή ακόμα και να αναβοσβήνουν εναλλάξ.

```
pinMode(13, OUTPUT);           // Set digital pin 13 -> output
pinMode(12, OUTPUT);           // Set digital pin 12 -> output

digitalWrite(13, HIGH);         // Pin 13 = 5 V, LED emits light
digitalWrite(12, HIGH);         // Pin 12 = 5 V, LED emits light
delay(500);                     // ..for 0.5 seconds
digitalWrite(13, LOW);          // Pin 13 = 0 V, LED no light
digitalWrite(12, LOW);          // Pin 12 = 0 V, LED no light
delay(500);                     // ..for 0.5 seconds
```

2.4 Παρακολούθηση LED Servo σημάτων

Τα υψηλά και χαμηλά σήματα που ελέγχουν τους σερβοκινητήρες (servo motors) θα πρέπει να διαρκούν για πολύ συγκεκριμένες χρονικές περιόδους. Αυτό συμβαίνει γιατί ένας σερβοκινητήρας μέτρα πόσο καιρό (χρονικά) το σήμα παραμένει ψηλά, και το χρησιμοποιεί ως εντολή για το πόσο γρήγορα και σε ποια κατεύθυνση θα στραφεί ο κινητήρας.

Το παρακάτω διάγραμμα χρονισμού δείχνει ένα σήμα σερβο (servo) το οποίο θα κάνει τον τροχό να στρίψει με πλήρης ταχύτητα αριστερόστροφα. Υπάρχει μια μεγάλη διαφορά όμως: όλα τα σήματα σε αυτό το διάγραμμα χρονισμού διαρκούν 100 φορές περισσότερο από ό, τι θα ήταν αν είχαν τον έλεγχο ενός σερβο. Αυτό επιβραδύνει αρκετά ώστε να μπορούμε να δούμε τι συμβαίνει.



2.4.1 Παράδειγμα Προγράμματος: ServoSlowMoCcw

/*

Robotics with the BOE Shield - ServoSlowMoCcw

Send 1/100th speed servo signals for viewing with an LED. */

```
void setup() // Built in initialization block
{
  pinMode(13, OUTPUT); // Set digital pin 13 -> output
}

void loop() // Main loop auto-repeats
{
  digitalWrite(13, HIGH); // Pin 13 = 5 V, LED emits light
  delay(170); // ..for 0.17 seconds
  digitalWrite(13, LOW); // Pin 13 = 0 V, LED no light
  delay(1830); // ..for 1.83 seconds
}
```

2.5 Σύνδεση Σερβοκινητήρων και μπαταριών

Από τη σκοπιά της ρομποτικής πλοήγησης, τα σερβος συνεχόμενης περιστροφής προσφέρουν ένα μεγάλο συνδυασμό απλότητας, χρησιμότητας και χαμηλής τιμής. Τα servos συνεχόμενης περιστροφής της Parallax είναι οι κινητήρες που θα κάνουν τους τροχούς του BOE Shield-Bot να περιστρέφονται, πάντα σύμφωνα με τον έλεγχο από τον Arduino.

2.5.1 Σύνδεση των Servos με το BOE Shield

Κατάλογος εξαρτημάτων:

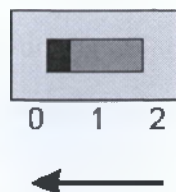
(2) Parallax κινητήρες συνεχόμενης περιστροφής

BOE Shield με κατασκευασμένα και ελεγμένα LED κυκλώματα



Πάντα πριν κάνουμε οποιαδήποτε αλλαγή στην πλακέτα μας ή σε κάποιο άλλο εξάρτημα του ρομπότ μας πρέπει να :

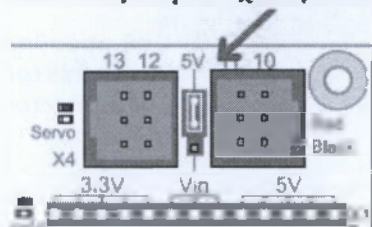
1. Ρυθμίσουμε τον διακόπτη τροφοδοσίας πάνω στην πλακέτα μας στην θέση 0.



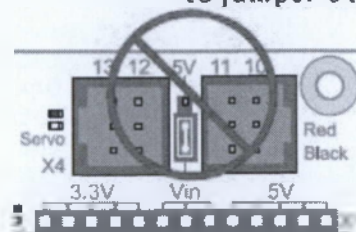
2. Αποσυνδέσουμε όλες τις πηγές τροφοδοσίας από τον Arduino, συμπεριλαμβανομένου και του καλωδίου USB.

Μεταξύ των σερβοκεφαλών στην πλακέτα (BOE Shield) είναι ένα jumper που συνδέει το τροφοδοτικό των σέρβο είτε σε Vin είτε σε 5V. Για να το μετακινήσουμε, τραβάμε προς τα πάνω και έξω από το ζευγάρι των ακίδων που καλύπτει, στη συνέχεια το ωθούμε στο ζευγάρι των ακίδων που θέλουμε. Η μπαταρία του BOE Shield-Bot θα μας παράσχει 7,5 V και αφού τα servos κυμαίνονται από 4-6 V, θέλουμε να βεβαιωθούμε ότι το jumper έχει οριστεί σε 5V. Επίσης μια σταθερή τάση τροφοδοσίας 5 V θα υποστηρίξει μια σταθερή ταχύτητα και μια πιο ακριβή πλοήγηση, από μια τάση που θα ποικίλλει καθώς θα ξεφορτίζονται οι μπαταρίες.

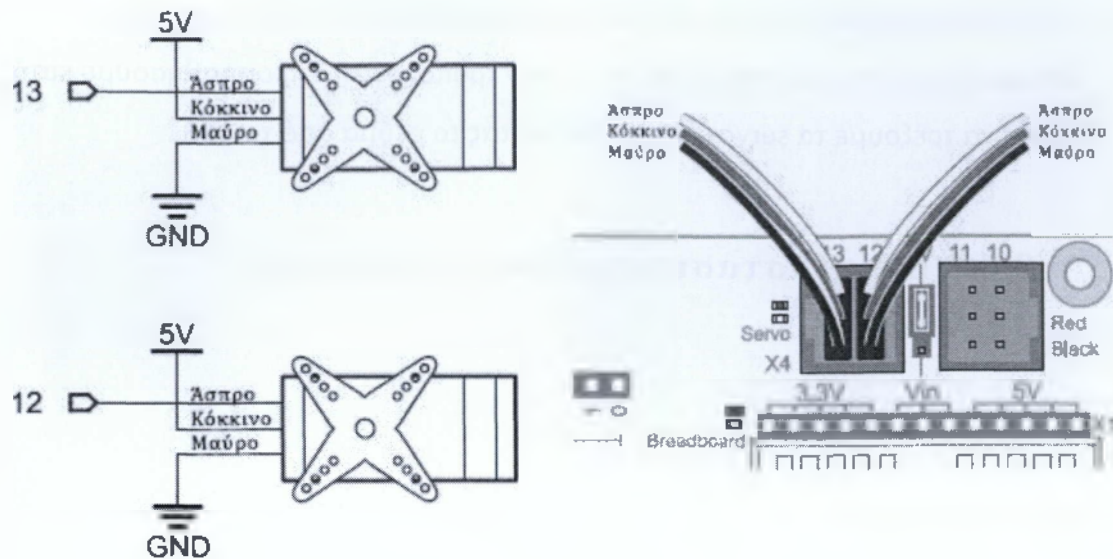
Αυτό το jumper έχει οριστεί στα 5 V



Δεν πρέπει να βάλουμε το jumper στο vin



Αφού είμαστε σίγουροι ότι το jumper είναι στα 5V, συνδέουμε τα servos στη BOE Shield όπως φαίνεται στο παρακάτω διάγραμμα. Το αριστερό σέρβο συνδέεται στη θύρα 13 και το δεξιό σέρβο συνδέεται στη θύρα 12. Επίσης, βεβαιωνόμαστε ότι έχουμε βάλει τα χρώματα των καλωδίων όπως φαίνονται στο σχήμα, με το μαύρο καλώδιο πιο κοντά στο breadboard και το λευκό καλώδιο πιο κοντά στην άκρη της πλακέτας.



2.5.2 Σύνδεση της θήκης των μπαταριών με το BOE Shield

Για να τροφοδοτούνται σωστά τα servos, θα πρέπει να μεταβούμε σε μια εξωτερική μπαταρία τώρα. Όταν τα servos κάνουν απότομες αλλαγές κατεύθυνσης ή ωθούνται ενάντια στην αντίσταση της περιστροφής, μπορούν να αντλήσουν περισσότερο ρεύμα από ότι μια θύρα USB έχει σχεδιαστεί για να παρέχει. Επίσης, δεν θα ήταν διασκεδαστικό για το BOE Shield-Bot να είναι δεμένο με τον υπολογιστή για πάντα! Έτσι, από εδώ και πέρα θα πρέπει να χρησιμοποιούμε μια εξωτερική θήκη μπαταριών με πέντε μπαταρίες AA 1,5 V. Αυτό θα προμηθεύει το σύστημά σας με 7,5 V και άφθονο ρεύμα για τους ρυθμιστές τάσης και τα servos.

Από εδώ και πέρα θα πρέπει πάντα να προσέχουμε 2 πράγματα :

1. Να αποσυνδέουμε την μπαταρία όταν τελειώνουμε να πειραματιζόμαστε. Ακόμα και όταν ο διακόπτης λειτουργίας της πλακέτας είναι απενεργοποιημένος (θέση-0), γιατί ο Arduino θα εξακολουθεί να αντλεί δύναμη από τις μπαταρίες.

- Επίσης να αποσυνδέουμε το καλώδιο προγραμματισμού, κάθε φορά που αποσυνδέουμε την μπαταρία. Με αυτόν τον τρόπο, δεν θα προσπαθήσουμε κατά λάθος να τρέξουμε τα servos χρησιμοποιώντας το ρεύμα από το USB.

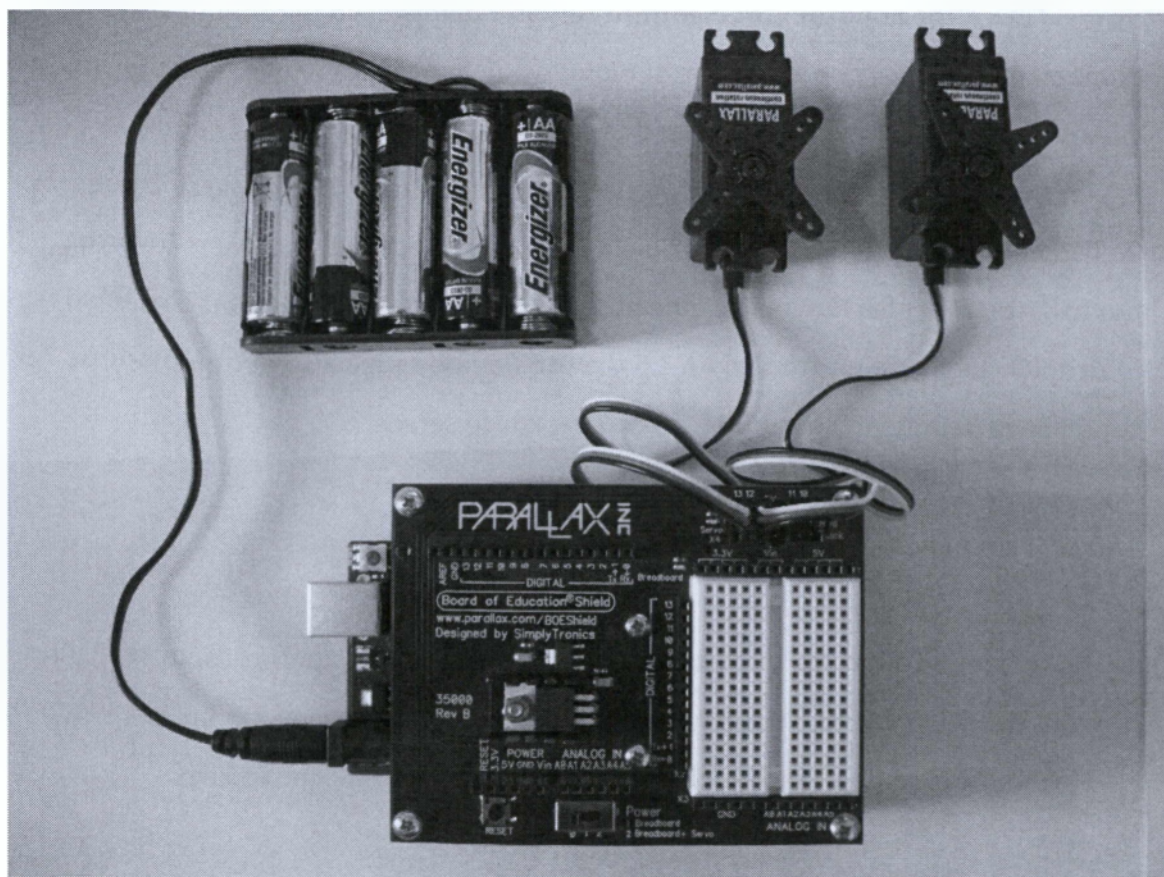
2.5.3 Εγκατάσταση της θήκης μπαταριών

Κατάλογος εξαρτημάτων :

(5) AA αλκαλικές μπαταρίες

(1) θήκη 5 μπαταριών

Τοποθετούμε τις μπαταρίες στην θήκη μπαταριών και συνδέουμε τη μπαταρία στην υποδοχή ενέργειας του Arduino. Όταν τελειώσουμε, θα πρέπει να μοιάζει με την παρακάτω εικόνα.



2.5.4 Ζυγοστάθμιση των σερβοκινητήρων

Απαραίτητο εργαλείο για να ρυθμίσουμε τα servos, έτσι ώστε πραγματικά να μένουν ακίνητοι είναι ένα κατσαβίδι με 1/8 "(3,18 mm) ή μικρότερο άξονα.



Εάν ένα σέρβο δεν έχει κεντραριστεί, μπορεί να στρίβει, να δονείται ή να παράγει ένα παράξενο βουητό όταν λαμβάνει την εντολή «stay-still».

Οπότε για να κεντράρουμε τον αριστερό σέρβο συνδέουμε το καλώδιο του προγραμματισμού, και εκτελούμε το πρόγραμμα LeftServoStayStill.

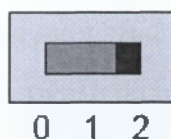
```

/*
Robotics with the BOE Shield - LeftServoStayStill
Generate signal to make the servo stay still for centering.
*/

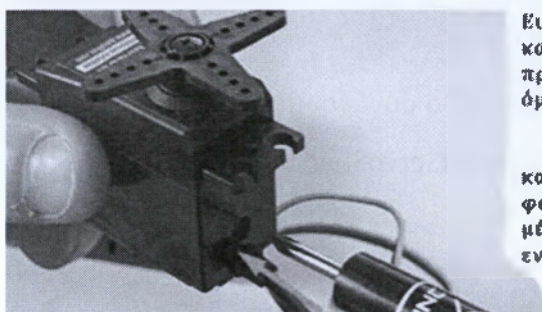
#include <Servo.h>           // Include servo library
Servo servoLeft;           // Declare left servo
void setup()                // Built in initialization block
{
  servoLeft.attach(13);     // Attach left signal to pin 13
  servoLeft.writeMicroseconds(1500); // 1.5 ms stay still signal
}
void loop()                 // Main loop auto-repeats
{
  // Empty, nothing needs repeating
}

```

Επίσης ρυθμίζουμε το διακόπτη του ρεύματος στην θέση 2, για την παροχή ενέργειας στα servos.

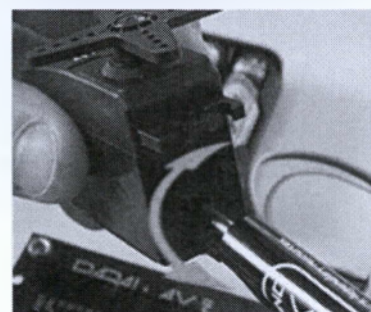


Στη συνέχεια χρησιμοποιούμε ένα κατσαβίδι για να ρυθμίσουμε απαλά το ποτενσιόμετρο στο σέρβο όπως φαίνεται στην παρακάτω εικόνα. Δεν πρέπει να πιέζουμε πάρα πολύ σκληρά! Ρυθμίζουμε το ποτενσιόμετρο λίγο μέχρι να βρούμε τη σωστή θέση που κάνει το σέρβο να σταματήσει οποιαδήποτε λειτουργία (στροφή, βουητό ή δόνηση).



Εισάγουμε την άκρη του κατσαβιδιού στην οπή πρόσβασης του ποτενσιόμετρου,

και στη συνέχεια στρέφουμε το ποτενσιόμετρο μέχρι το σέρβο να είναι εντελώς ακίνητο.



Έπειτα πραγματοποιούμε την ίδια διαδικασία με το διάγραμμα `RightServoStayStill` για να κεντράρουμε και τον δεξιό σέρβο.

```

/*
 Robotics with the BOE Shield - RightServoStayStill
 Transmit the center or stay still signal on pin 12 for center
 adjustment.
 */

#include <Servo.h> // Include servo library

Servo servoRight; // Declare right servo

void setup() // Built-in initialization block
{
  servoRight.attach(12); // Attach right signal to pin 12
  servoRight.writeMicroseconds(1500); // 1.5 ms stay still signal
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

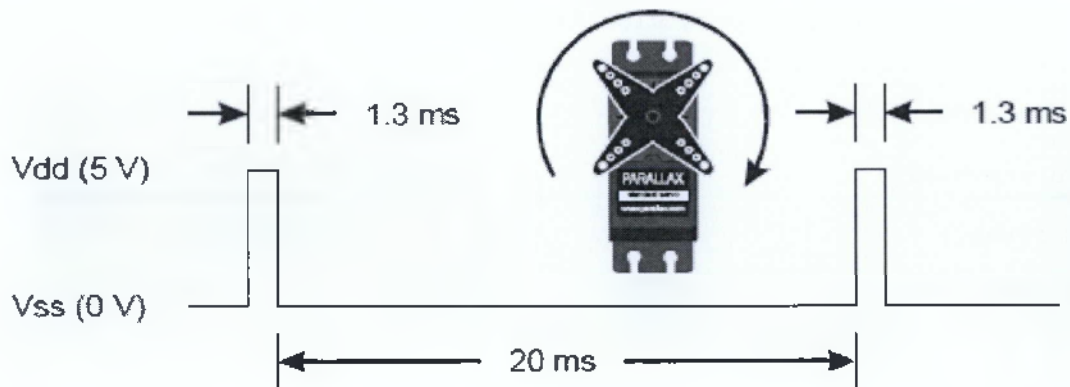
```

2.5.5 Δοκιμή των σερβοκινητήρων

Υπάρχει ένα τελευταίο πράγμα που πρέπει να κάνουμε πριν από τη συναρμολόγηση του BOE Shield-Bot, και αυτό είναι η δοκιμή των servos. Αυτό θα το επιτύχουμε κάνοντας τα servos να στρίβουν σε διαφορετικές ταχύτητες και κατευθύνσεις. Αυτό είναι ένα παράδειγμα της δοκιμής υποσυστήματος (subsystem testing), μια καλή συνήθεια που αξίζει να αναπτυχθεί.

Το εύρος των παλμών ελέγχουν τη ταχύτητα και τη διεύθυνση

Αυτό το διάγραμμα χρονισμού δείχνει πώς ένα Parallax servo συνεχούς περιστροφής γυρίζει δεξιόστροφα με πλήρης ταχύτητας όταν του στέλνουμε 1,3 ms παλμούς. Η πλήρης ταχύτητα κυμαίνεται συνήθως σε φάσμα στροφών (RPM) 50-60.



Με το παρακάτω πρόγραμμα, παρατηρούμε τον αριστερό σέρβο να περιστρέφεται δεξιόστροφα με 50 – 60 RPM.

```

/*
Robotics with the BOE Shield - LeftServoClockwise
Generate a servo full speed clockwise signal on digital pin 13.
*/

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left servo

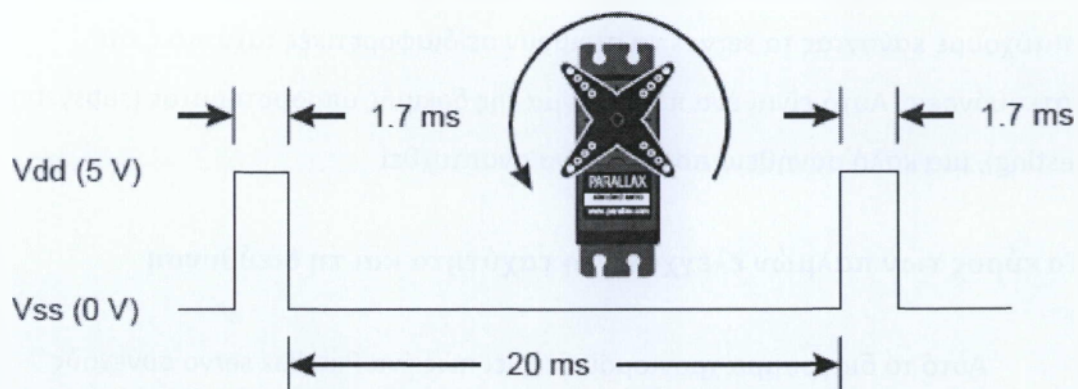
void setup() // Built in initialization block
{
  servoLeft.attach(13); // Attach left signal to pin 13
  servoLeft.writeMicroseconds(1300); // 1.3 ms full speed clockwise
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating

```

}

Αν τώρα αλλάζουμε την τιμή της εντολής `servoLeft.writeMicroseconds` από 1300 σε 1700 θα έχουμε πλήρης περιστροφή αριστερόστροφα με 50-60 RPM.



```

/*
Robotics with the BOE Shield - LeftServoCounterclockwise
Generate a servo full speed clockwise signal on digital pin 13.
*/

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left servo

void setup() // Built in initialization block
{
  servoLeft.attach(13); // Attach left signal to pin 13
  servoLeft.writeMicroseconds(1700); // 1.7 ms full speed counterclockwise
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

```

Αντίστοιχα τροποποιούμε και τα σχεδιαγράμματα για να τεστάrouμε τον δεξιό σέρβο να περιστρέφεται δεξιόστροφα και αριστερόστροφα με πλήρης ταχύτητα στα 50-60 RPM. Αντικαταστούμε όλες τις εμφανίσεις του `servoLeft` με `servoRight` και όλα τα υποδείγματα των 13 με 12.

```

/*
Robotics with the BOE Shield - RightServoClockwise
Generate a servo full speed clockwise signal on digital pin 12.
*/

#include <Servo.h> // Include servo library

Servo servoRight; // Declare right servo

```

```

void setup() // Built in initialization block
{
  servoRight.attach(12); // Attach right signal to pin 12
  servoRight.writeMicroseconds(1300); // 1.3 ms full speed clockwise
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

/*
Robotics with the BOE Shield - RightServoCounterclockwise
Generate a servo full speed clockwise signal on digital pin 13.
*/

#include <Servo.h> // Include servo library

Servo servoRight; // Declare right servo

void setup() // Built in initialization block
{
  servoRight.attach(12); // Attach right signal to pin 12
  servoRight.writeMicroseconds(1700); // 1.7 ms full speed counterclockwise
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

```

2.5.5.1 Έλεγχος της ταχύτητας και της κατεύθυνσης των Servo

Για να μπορέσουμε να έχουμε μια πετυχημένη πλοήγηση του ρομπότ μας θα πρέπει να ελέγχουμε και τα δύο servos ταυτόχρονα. Στο παρακάτω πρόγραμμα έχουμε το servo στην θύρα 13 (αριστερό σέρβο) να περιστρέφεται αριστερόστροφα και ταυτόχρονα το servo στην θύρα 12 (δεξιό σέρβο) να περιστρέφεται δεξιόστροφα.

```

/*
Robotics with the BOE Shield - ServosOppositeDirections
Generate a servo full speed counterclockwise signal with pin 13 and
full speed clockwise signal with pin 12.
*/

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left servo signal
Servo servoRight; // Declare right servo signal

void setup() // Built in initialization block
{
  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12
}

```

```

servoLeft.writeMicroseconds(1700); // 1.7 ms -> counterclockwise
servoRight.writeMicroseconds(1300); // 1.3 ms -> clockwise
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

```

Αυτή η ρύθμιση αντίθετης κατεύθυνσης είναι πολύ σημαντική. Για παράδειγμα όταν τα servos τοποθετηθούν σε κάθε πλευρά του πλαισίου, το ένα θα πρέπει να περιστρέφεται δεξιόστροφα, ενώ το άλλο αριστερόστροφα για να μπορέσει το BOE Shield-Bot να κινηθεί σε μια ευθεία γραμμή.

Διαφορετικοί συνδυασμοί των παραμέτρων writeMicroseconds θα μας χρησιμεύσουν κατ'επανάληψη για τον προγραμματισμό κίνησης του BOE Shield-Bot μας. Με τη δοκιμή διάφορων πιθανών συνδυασμών συμπληρώσαμε έναν Περιγραφικό Πίνακα 2-1, ο οποίος μας βοηθάει να κατανοήσουμε τη συμπεριφορά του ρομπότ μας δίνοντας στους σέρβο διαφορετικές τιμές, με αποτέλεσμα να αλλάζει και η κατεύθυνση κίνησης.

Πίνακας 2 : writeMicroseconds Συνδυασμοί			
Pin 13 servoLeft	Pin 12 servoRight	Περιγραφή	Συμπεριφορά
1700	1300	Πλήρης ταχύτητα, pin 13 servo αριστερόστροφα, pin 12 servo δεξιόστροφα	Πλήρης ταχύτητα, κίνηση προς τα εμπρός
1300	1700	Πλήρης ταχύτητα, pin 13 servo δεξιόστροφα, pin 12 servo αριστερόστροφα	Πλήρης ταχύτητα, κίνηση προς τα πίσω
1700	1700	Πλήρης ταχύτητα, pin 13 servo αριστερόστροφα, pin 12 servo αριστερόστροφα	Επιτόπου στροφή προς τα δεξιά
1300	1300	Πλήρης ταχύτητα, pin 13 servo δεξιόστροφα, pin 12 servo δεξιόστροφα	Επιτόπου στροφή προς τα αριστερά
1500	1700	pin 13 servo ακίνητο, pin 12 servo πλήρης ταχύτητα αριστερόστροφα	Κίνηση προς τα πίσω + στροφή προς τα αριστερά (αριστερή οπισθογωνία)

1300	1500	pin 13 servo πλήρης ταχύτητα δεξιόστροφα, pin 12 servo ακίνητο	Κίνηση προς τα πίσω + στροφή προς τα δεξιά (δεξιά οπισθογωνία)
1500	1500	pin 13 servo ακίνητο, pin 12 servo ακίνητο	Σταθερά ακίνητο
1520	1480	Χαμηλή ταχύτητα, pin 13 servo αριστερόστροφα, pin 12 servo δεξιόστροφα	Χαμηλή ταχύτητα, κίνηση προς τα εμπρός
1540	1460	Μέτρια ταχύτητα, pin 13 servo αριστερόστροφα, pin 12 servo δεξιόστροφα	Μέτρια ταχύτητα, κίνηση προς τα εμπρός
1700	1450	pin 13 servo πλήρης ταχύτητα αριστερόστροφα, pin 12 servo μέτρια ταχύτητα δεξιόστροφα	Ανοιχτή στροφή προς τα δεξιά
1550	1300	pin 13 servo μέτρια ταχύτητα αριστερόστροφα, pin 12 servo πλήρης ταχύτητα δεξιόστροφα	Ανοιχτή στροφή προς τα αριστερά

2.5.5.2 Πώς να ελέγχεται ο χρόνος κίνησης των Servo

Είναι εύκολο να ελέγχουμε τον χρόνο που τα servos τρέχουν όταν χρησιμοποιούν τη βιβλιοθήκη Servo. Μόλις έχουν ρυθμιστεί, τα servos διατηρούν την κίνησή τους μέχρι να λάβουν μια νέα ρύθμιση. Οπότε, για να κάνουμε ένα servo να κινείται για ένα ορισμένο χρονικό διάστημα, το μόνο που έχουμε να κάνουμε είναι να τοποθετήσουμε μια καθυστέρηση (delay) μετά από κάθε ρύθμιση.

Στο παράδειγμα του παρακάτω σχεδιαγράμματος έχουμε και τα 2 servos να περιστρέφονται δεξιόστροφα για 3 δευτερόλεπτα, μετά αριστερόστροφα για άλλα 3 δευτερόλεπτα και στο τέλος να σταματάμε.

```

/*
Robotics with the BOE Shield - ServoRunTimes
Generate a servo full speed counterclockwise signal with pin 13 and
full speed clockwise signal with pin 12.
*/

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left servo signal
Servo servoRight; // Declare right servo signal

void setup() // Built in initialization block

```

```
{
  servoLeft.attach(13);           // Attach left signal to pin 13
  servoRight.attach(12);          // Attach right signal to pin 12

  servoLeft.writeMicroseconds(1300); // Pin 13 clockwise
  servoRight.writeMicroseconds(1300); // Pin 12 clockwise
  delay(3000);                    // ..for 3 seconds
  servoLeft.writeMicroseconds(1700); // Pin 13 counterclockwise
  servoRight.writeMicroseconds(1700); // Pin 12 counterclockwise
  delay(3000);                    // ..for 3 seconds
  servoLeft.writeMicroseconds(1500); // Pin 13 stay still
  servoRight.writeMicroseconds(1500); // Pin 12 stay still
}

void loop()                       // Main loop auto-repeats
{                                  // Empty, nothing needs repeating
}
```

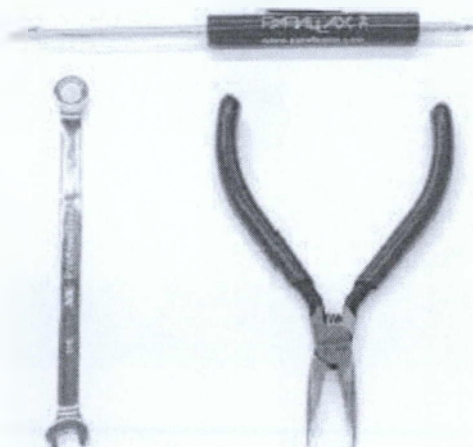

2.6 Συναρμολόγηση του BOE-Shield-Bot

Εργαλεία :

(1) κατσαβίδι

(1) 0,6cm κλειδί (προαιρετικό αλλά πρακτικό)

(1) μυτοσίμπιδο (προαιρετικό)



Πρώτα από όλα πρέπει να αποσυνδέσουμε το καλώδιο προγραμματισμού και να ρυθμίσουμε το διακόπτη τροφοδοσίας το BOE Shield στη θέση 0. Επίσης πρέπει να αποσυνδέσουμε τα servos από το BOE Shield και την μπαταρία από το Arduino.

Ξεκινάμε τη συναρμολόγηση του BOE-Shield-Bot με την τοποθέτηση της βάσης του ρομπότ .

Κατάλογος εξαρτημάτων :

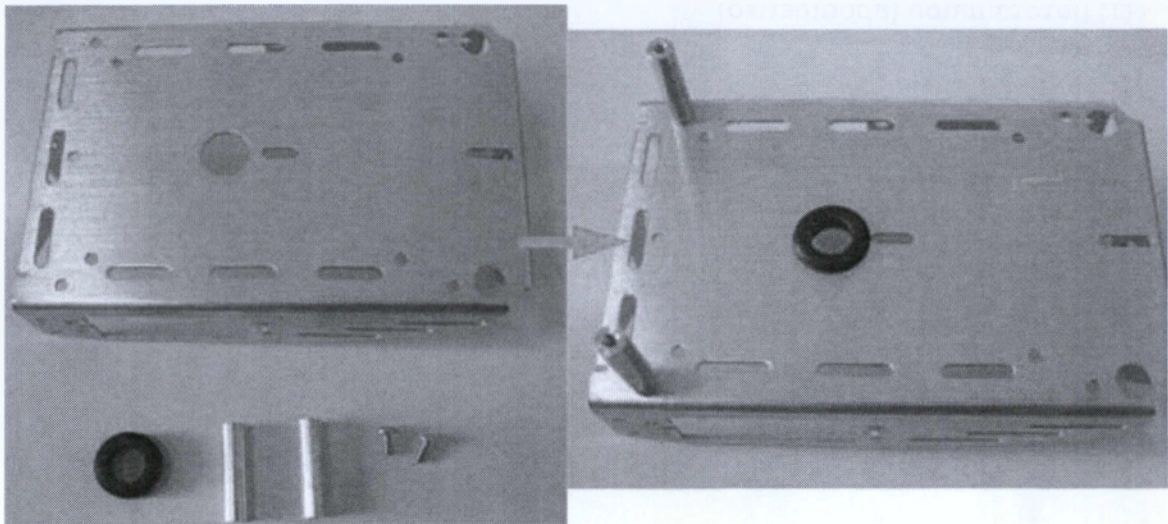
(1) ρομπότ σασί (βάση)

(2) 2,5cm διαχωριστικά (αφαιρούνται από το BOE Shield)

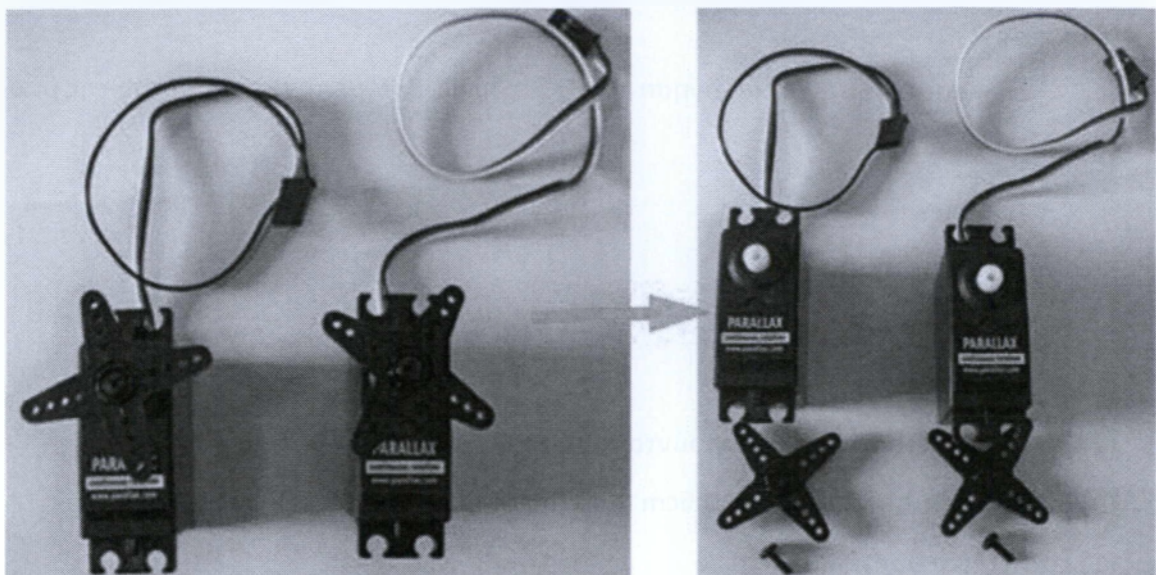
(2) βίδες με επίπεδη κεφαλή, 0,6cm 4-40 (αφαιρούνται από το BOE Shield)

(1) λαστιχένιο δακτύλιο, 1cm

Τοποθετούμε το 1cm λαστιχένιο δακτύλιο μέσα στην οπή στο κέντρο του πλαισίου. Και βεβαιωνόμαστε ότι το αυλάκι στην εξωτερική μεριά του ελαστικού κρίκου εδράζεται επί της μεταλλικής άκρης της τρύπας. Μετά αφαιρούμε τα 1" διαχωριστικά αλουμινίου από την BOE Shield, και τα αποθηκεύουμε μαζί με τις βίδες τους. Χρησιμοποιούμε τις δύο 0,6cm βίδες για να συνδέσουμε δύο από τα διαχωριστικά στην μπροστινή κορυφή του πλαισίου, όπως φαίνεται στην παρακάτω εικόνα.



Στη συνέχεια ,χρησιμοποιώντας το κατσαβίδι αφαιρούμε τις βίδες που κρατούν τις κεφαλές των servo (που προηγουμένως είχαμε κεντράρει) στους άξονες εξόδου. Τραβάμε προς τα πάνω κάθε κεφαλή από τον άξονα εξόδου και αποθηκεύουμε τις βίδες γιατί θα τις χρειαστούμε για να βιδώσουμε τους άξονες.



Το επόμενο στάδιο είναι η τοποθέτηση των servos κάτω από την βάση του ρομπότ.

Κατάλογος εξαρτημάτων :

(2) BOE Shield-Bot Σασί, μερικώς συναρμολογημένο.

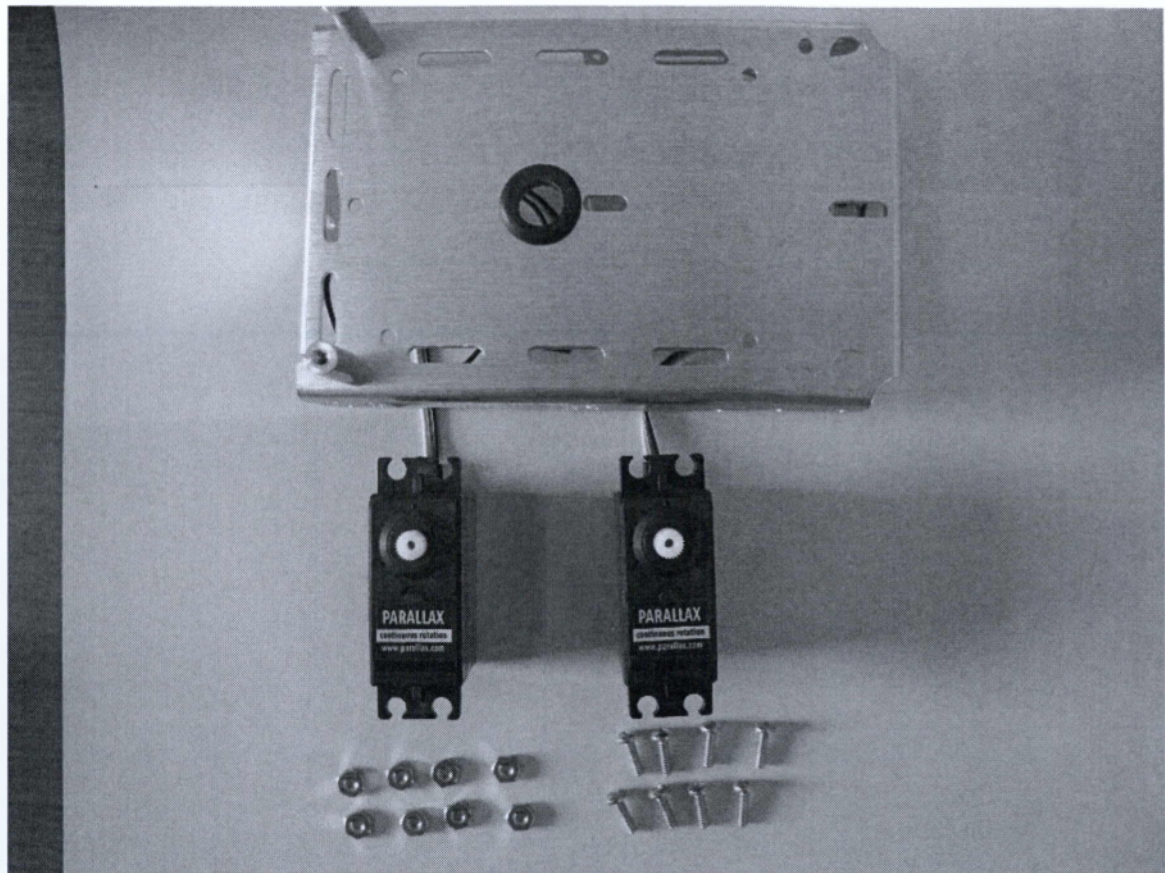
(2) Parallax συνεχής εναλλαγής servos

(8) Κυλινδρικής κεφαλής Βίδες 1cm, 4-40

(8) παξιμάδια, 4-40

κολλητική ταινία

στυλό

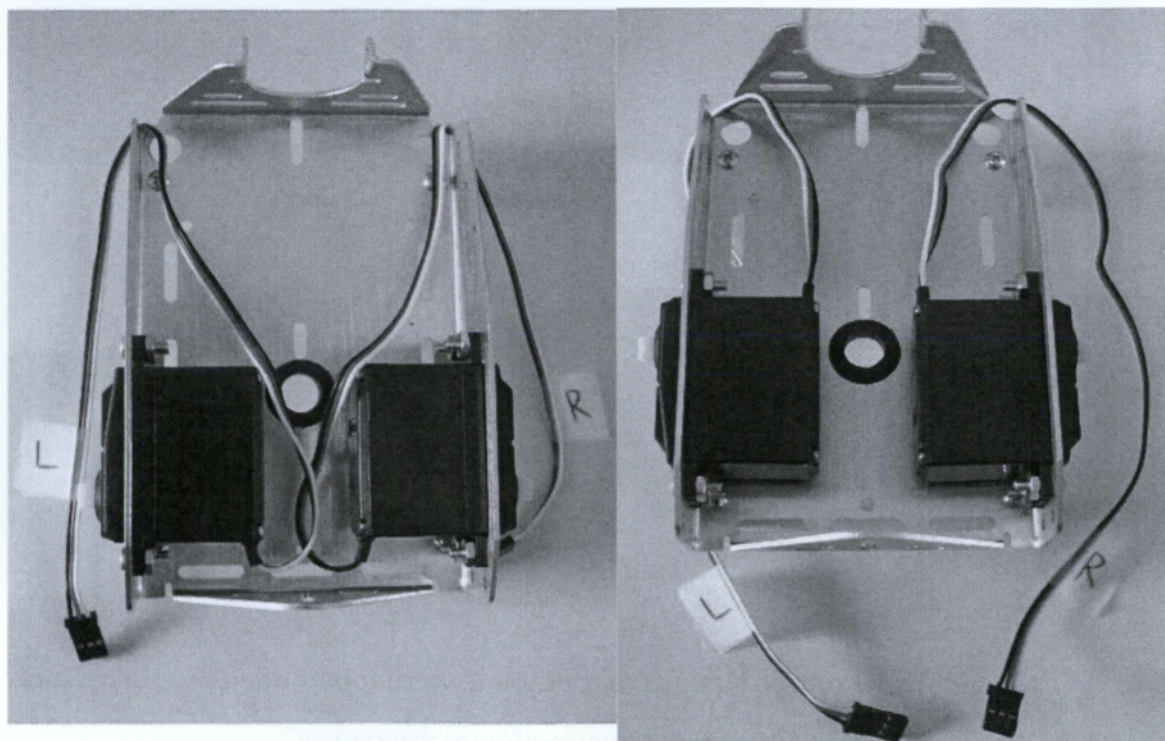


Μπορούμε να αποφασίσουμε πώς θέλουμε να τοποθετήσουμε τα servos από τις δύο επιλογές που περιγράφονται και απεικονίζονται παρακάτω.

Έξω από τα εμπρός (αριστερά) - τα servos με τις θύρες πρόσβασης του ποτενσιόμετρου τους να βλέπουν προς το μπροστινό μέρος της βάσης. Αυτό επιτρέπει την εύκολη πρόσβαση για την ρύθμιση των ποτενσιόμετρων αφού έχουμε ολοκληρώσει τη συναρμολόγηση του ρομπότ, και επίσης καθιστά πιο γρήγορη την αντικατάσταση των servo. Ωστόσο, αυτό δίνει στο BOE Shield-Bot ένα μεγαλύτερο και ευρύτερο μεταξόνιο, οπότε θα είναι λιγότερο ευκίνητο στους ελιγμούς και μπορεί να χρειαστεί περισσότερους παλμούς για να μπορέσει να στρίβει.

Μέσα προς τα πίσω (δεξιά) - τα servos με τις θύρες πρόσβασης των ποτενσιόμετρων στραμμένες προς την μπαταρία. Αυτός ο τρόπος τοποθετεί τους άξονες κοντά στο κέντρο του BOE-Shield Bot, έτσι ώστε να έχει μέγιστη ευελιξία. Εάν ήμασταν επιμελείς με το κεντράρισμα των servos πριν από ολοκλήρωση της συναρμολόγησης του BOE Shield-Bot, τότε δεν θα έχουμε κανένα πρόβλημα.

Προφανώς και στις 2 περιπτώσεις βιδώνουμε τα servo με τις βίδες και τα παξιμάδια που έχουμε και πολύ χρήσιμο είναι να χρησιμοποιήσουμε κολλητική ταινία (χαρτοταινία) για να μαρκάρουμε τα servo αριστερά (L) και δεξιά (R).



Επίσης έχουμε να τοποθετήσουμε και την θήκη των μπαταριών κάτω από την βάση του ρομπότ.

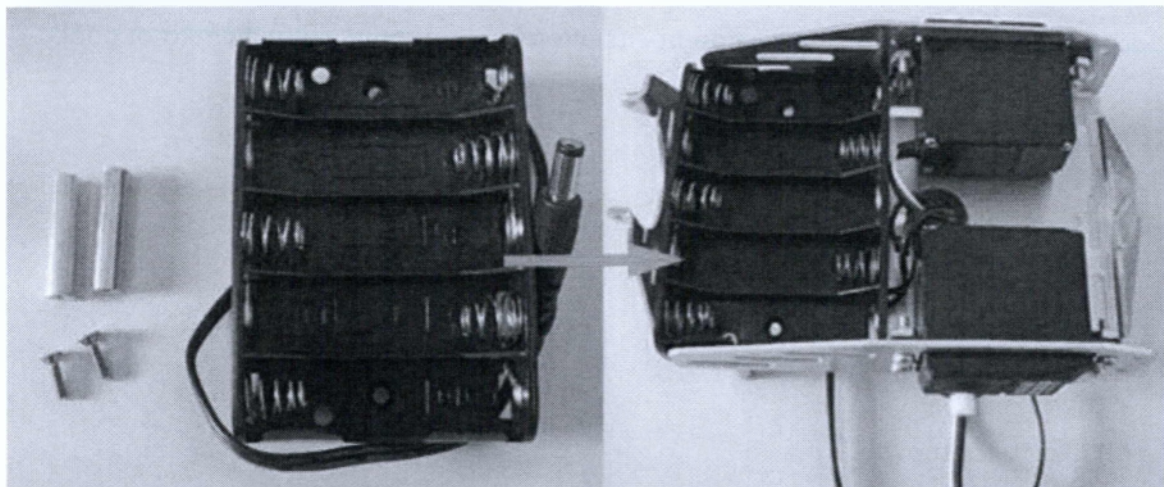
Κατάλογος εξαρτημάτων :

(2) επίπεδης κεφαλής βίδες, 1cm 4-40 (αφαιρούνται από το BOE Shield)

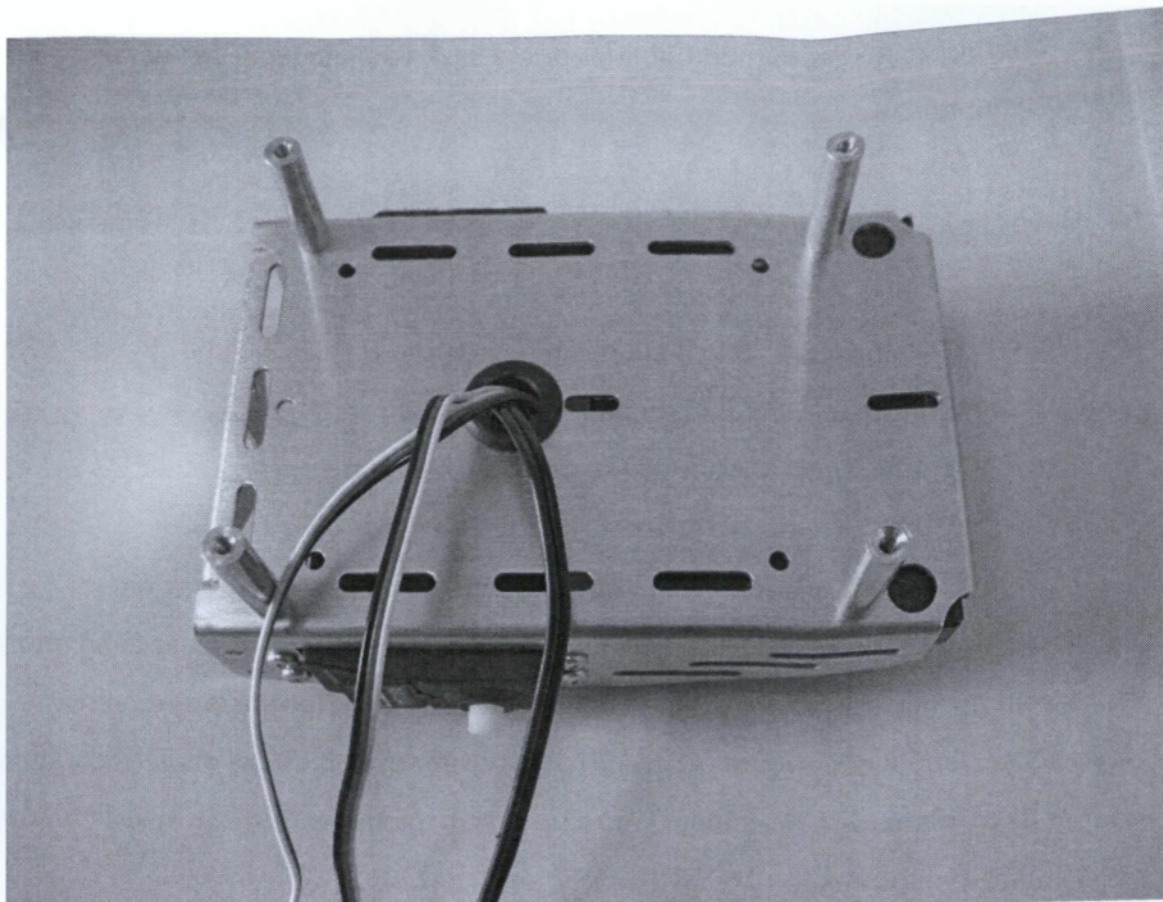
(2) 2,5cm διαχωριστικά (αφαιρούνται από το BOE Shield)

(1) θήκη 5 μπαταριών με 2,1 χιλιοστών κέντρο-θετικό βύσμα

Τοποθετούμε την άδεια θήκη μπαταριών στο εσωτερικό της βάσης όπως φαίνεται στην παρακάτω εικόνα. Τοποθετούμε τις δύο βίδες με επίπεδη κεφαλή από το εσωτερικό της θήκης μπαταριών, χρησιμοποιώντας την μικρότερη σειρά των οπών (τρύπες) που ευθυγραμμίζονται με τις οπές στερέωσης στην βάση, οι οποίες δείχνονται από τα βέλη. Μετά, από την κορυφή του πλαισίου, περνάμε τα διαχωριστικά 1 "σε κάθε βίδα και σφίγγουμε καλά.



Αφού τελειώσουμε με το παραπάνω , καλό θα είναι να τραβήξουμε τα καλώδια τροφοδοσίας των servo και τα καλώδια της μπαταρίας μέσω του λαστιχένιου δακτυλίου στο κέντρο του πλαισίου, όπως φαίνεται παρακάτω.



Τα επόμενα κομμάτια που πρέπει να τοποθετήσουμε είναι φυσικά οι ρόδες.

Κατάλογος εξαρτημάτων :

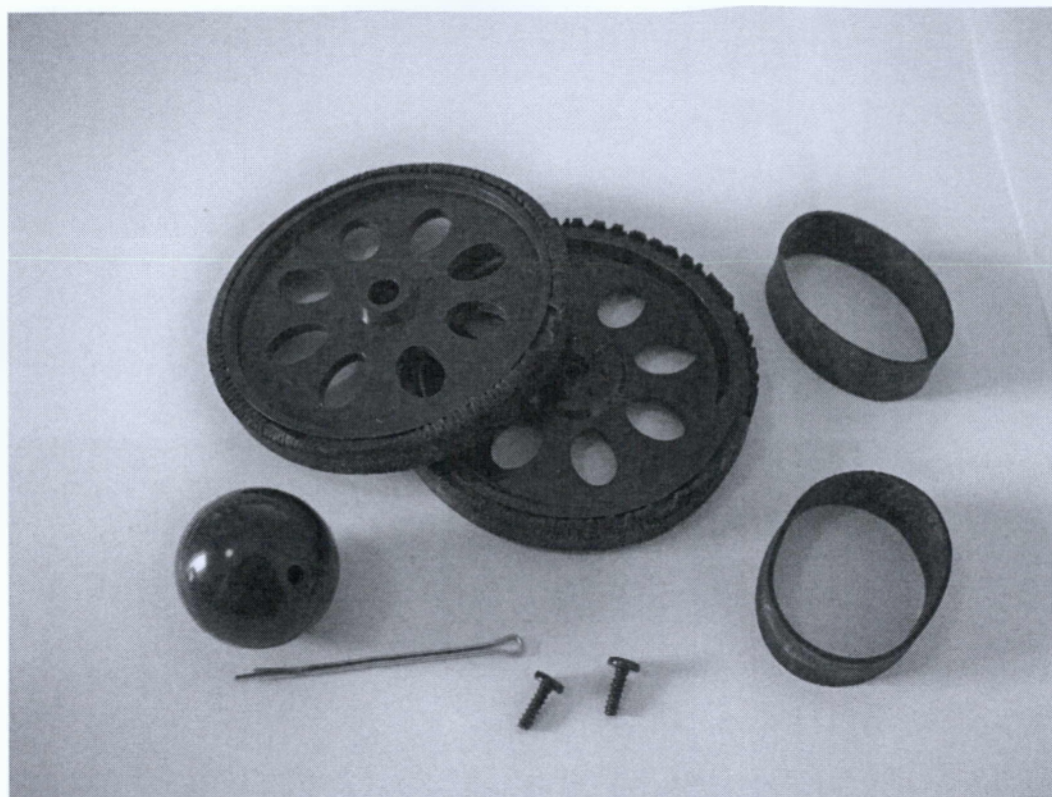
(1) περόνη 0,2cm

(1) τροχός μπάλα (για το πίσω μέρος)

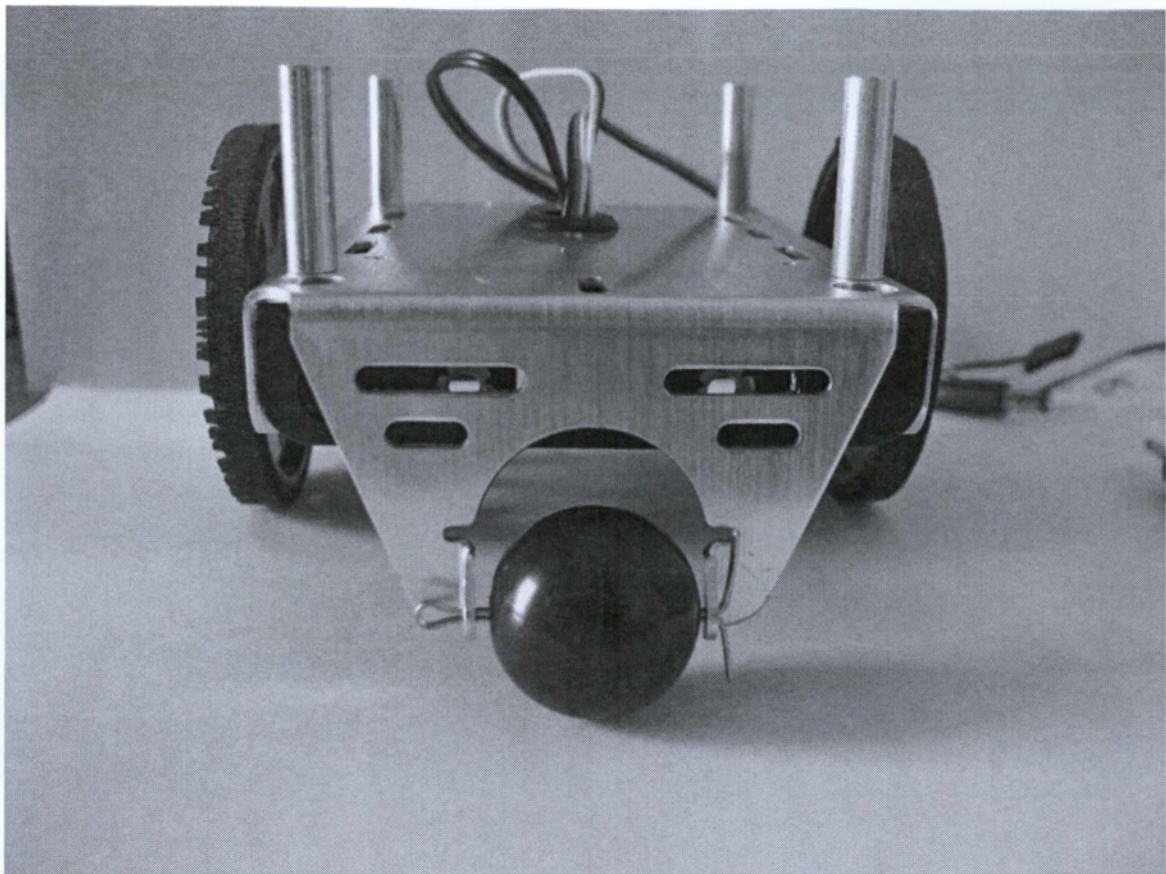
(2) ελαστικά λάστιχο

(2) πλαστικοί τροχοί

(2) βίδες που έχουμε αποθηκεύσει όταν αφαιρέσαμε τις κεφαλές από τα servo.



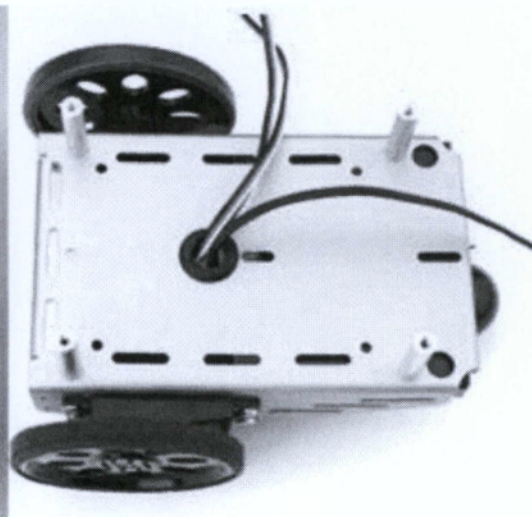
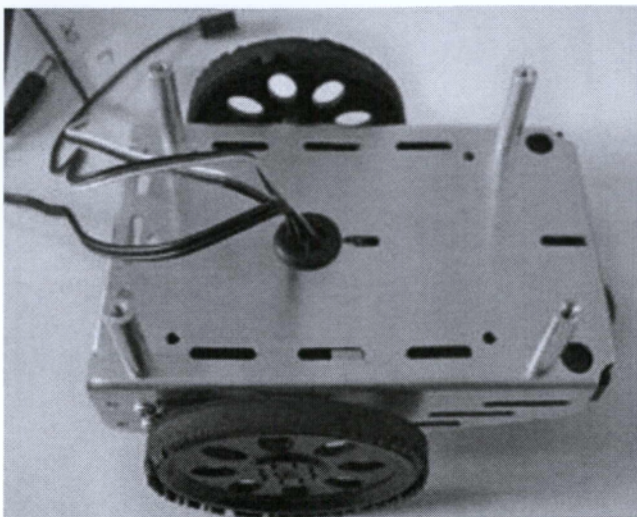
Ο τροχός της ουράς του ρομπότ είναι απλώς μια πλαστική σφαίρα με μια τρύπα στο κέντρο της. Η περόνη συγκρατεί τον τροχό στην βάση και λειτουργεί ως άξονας για τον τροχό. Για την τοποθέτηση της πρέπει να στοιχίσουμε την τρύπα στον τροχό της ουράς με τις οπές στο τμήμα ουράς της βάσης και να περάσουμε την περόνη μέσα από τις τρεις αυτές οπές (αριστερό σασί, ουραίο τροχό, σασί δεξιά). Τέλος λυγίζουμε τα άκρα της περόνης μεταξύ τους έτσι ώστε να μην μπορεί να γλιστρήσει πίσω από την τρύπα.



Μετά πατάμε κάθε πλαστικό τροχό πάνω σε ένα άξονα εξόδου servo, φροντίζοντας οι γραμμές του άξονα να βυθίζονται στις εσοχές του τροχού και στη συνέχεια ασφαλίζουμε με τις βίδες των servo. Τελειώνουμε τοποθετώντας τα ελαστικά λάστιχα στην εξωτερική πλευρά του κάθε τροχού.

Αριστερά: «Μέσα-προς τα πίσω» servos

Δεξιά: «Έξω-προς τα εμπρός» servos



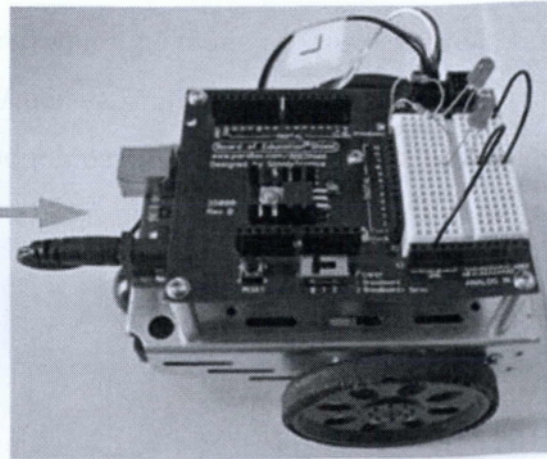
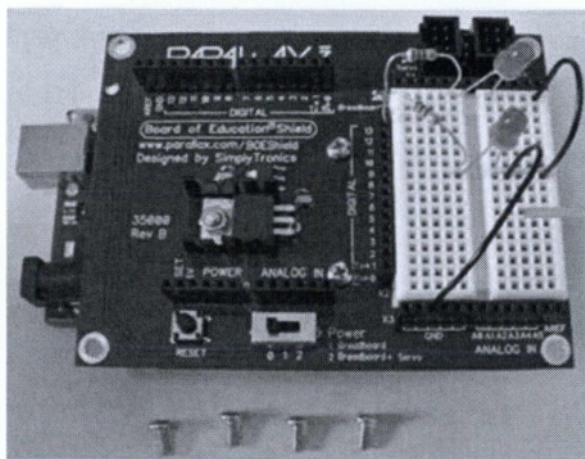
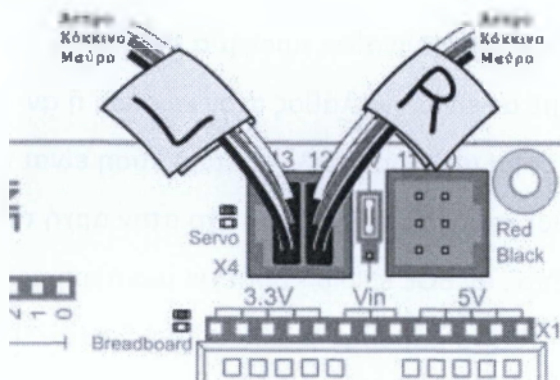
Τέλος πρέπει να συνδέσουμε το BOE Shield με την βάση.

Κατάλογος εξαρτημάτων :

(4) βίδες με επίπεδη κεφαλή, 0,6cm 4-40

(1) Board of Education Shield τοποθετημένο στον Arduino και ασφαλισμένος με διαχωριστικά.

Τοποθετούμε το BOE Shield στα τέσσερα διαχωριστικά έτσι ώστε να είναι ευθυγραμμισμένες με τις τέσσερις οπές στερέωσης στις εξωτερικές γωνίες της πλακέτας. Βεβαιωνόμαστε ότι το λευκό breadboard είναι πιο κοντά στους κινητήριους τροχούς (και όχι στο τροχό της ουράς) και βιδώνουμε την πλακέτα με τα διαχωριστικά της βάσης. Έπειτα επανασυνδέουμε τα servo στις κεφαλές των servo πάνω στην πλακέτα.



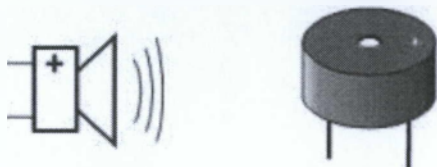
2.7 Λειτουργιά και επαναφορά Δείκτη

Όταν η παροχή τάσης πέφτει κάτω από το επίπεδο που μια συσκευή χρειάζεται για να λειτουργήσει σωστά , ονομάζεται brownout. Η συσκευή (Arduino) συνήθως κλείνει μέχρις ότου η τάση τροφοδοσίας να επιστρέψει στα κανονικά επίπεδα και στη συνέχεια επανεκκινεί ότι σχεδιάγραμμα έτρεχε .

Τα Brownout συνήθως συμβαίνουν όταν η ισχύς των μπαταριών είναι σε χαμηλό επίπεδο, και τα servos ξαφνικά απαιτούν περισσότερη δύναμη . Για παράδειγμα , αν το BOE Shield - Bot αλλάξει από πλήρης ταχύτητα προς τα εμπρός σε πλήρη ταχύτητα προς τα πίσω , τα servos έχουν να κάνουν επιπλέον εργασία σταματώντας και στη συνέχεια πηγαίνοντας προς την άλλη κατεύθυνση . Για αυτό το σκοπό χρειάζονται περισσότερο ρεύμα , και προσπαθώντας να τραβήξουν αυτό το ρεύμα από κουρασμένες μπαταριές , η τάση εξόδου βυθίζεται αρκετά ώστε να προκαλέσει brownout.

Τώρα , φανταστείτε ότι το BOE Shield - Bot πλοηγείται μέσω μιας ρουτίνας , και ξαφνικά σταματάει για μια στιγμή και στη συνέχεια πηγαίνει προς μια εντελώς απροσδόκητη κατεύθυνση . Πώς θα ξέρουμε αν είναι ένα λάθος στον κώδικα, ή αν πρόκειται για μια περίοδο χαμηλής τάσης ; Μια απλή , αποτελεσματική λύση είναι να προσθέσουμε ένα ηχείο στο BOE Shield - Bot και να το παίξει έναν ήχο στην αρχή του κάθε σχεδιαγράμματος . Με αυτόν τον τρόπο , αν BOE Shield - Bot έχει μια περίοδο χαμηλής τάσης (brownout) ενώ πλοηγείται , θα το μάθουμε αμέσως γιατί θα παίξει τον ήχο έναρξης .

Θα χρησιμοποιήσουμε μια συσκευή που ονομάζεται πιεζοηλεκτρικό ηχείο (piezospeaker) , το οποίο δημιουργεί διαφορετικούς ήχους , ανάλογα με τη συχνότητα των υψηλών / χαμηλών σημάτων που λαμβάνει από το Arduino . Το σχηματικό σύμβολο και ο πραγματικός σχεδιασμός του ηχείου δείχνονται παρακάτω .

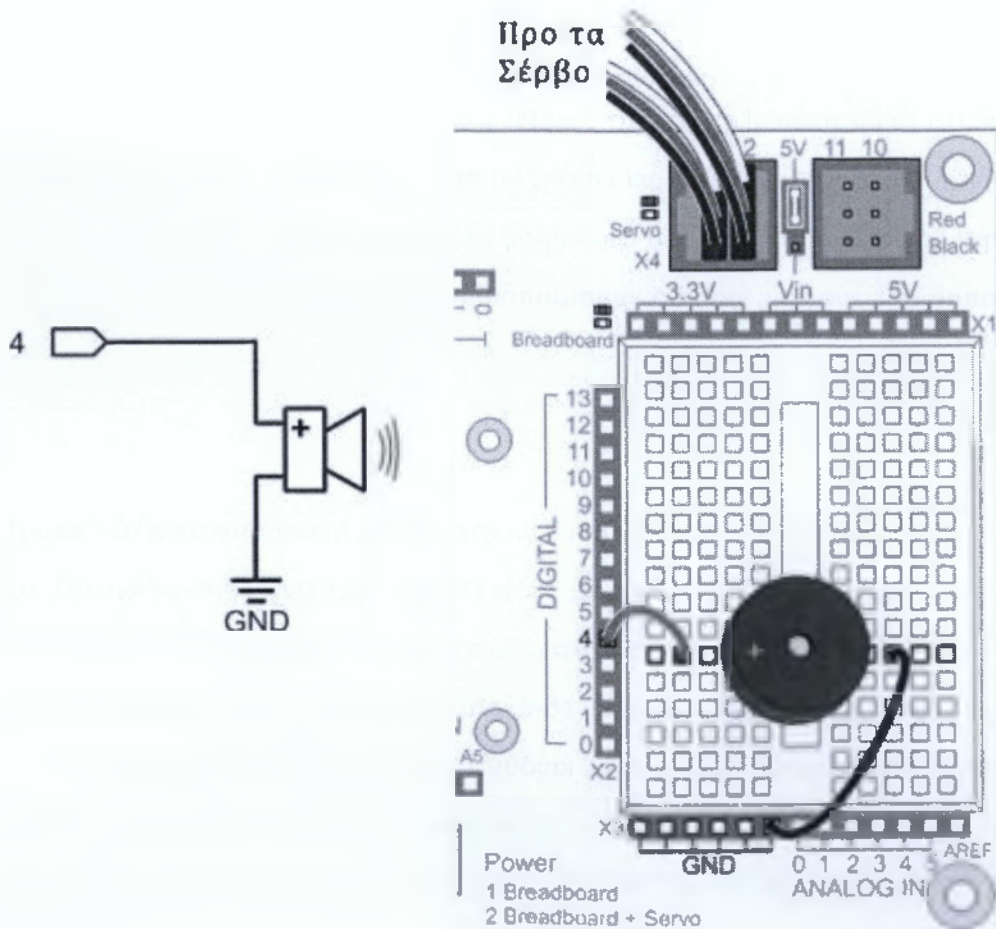


2.7.1 Κατασκευή του Piezospeaker κυκλώματος

Απαραίτητα Εξαρτήματα :

- (1) piezospeaker
- (2) καλώδια jumper

Η παρακάτω εικόνα δείχνει ένα διάγραμμα καλωδίωσης για την προσθήκη ενός piezospeaker στο breadboard. Τοποθετούμε το piezospeaker ακριβώς όπως φαίνεται στο breadboard και δεν θα χρειαστεί να το μετακινήσουμε, τα υπόλοιπα κυκλώματα θα προστίθενται ή θα αφαιρούνται γύρω από αυτό .



2.7.2 Προγραμματισμός του Piezospeaker

Η εντολή `tone` του Arduino, όπως μαρτυρά και το όνομά της, στέλνει σήματα στα ηχεία έτσι ώστε να τα κάνει να παίζουν ήχους.

Υπάρχουν δύο επιλογές για την κλήση της συνάρτησης `tone`. Η πρώτη μας επιτρέπει να ορίσουμε το `pin` και τη συχνότητα (`pitch`) του ήχου. Η άλλη μας επιτρέπει να ορίσουμε το `pin`, τη συχνότητα και τη διάρκεια (σε χιλιοστά του δευτερολέπτου). Θα χρησιμοποιήσουμε τη δεύτερη επιλογή, δεδομένου ότι δεν χρειαζόμαστε τον ήχο να συνεχίζεται επ'άπειρον.

```
tone(pin, frequency)
tone(pin, frequency, duration)
```

Αυτό το `piezospeaker` έχει σχεδιαστεί για να παίζει 4,5 τόνους kHz για συναγερμούς καπνού, αλλά μπορεί επίσης να παίζει μια ποικιλία από ηχητικούς τόνους και συνήθως ακούγεται καλύτερα στο εύρος 1 kHz έως 3,5 kHz. Ο τόνος προειδοποίησης έναρξης που θα χρησιμοποιήσουμε είναι:

```
tone(4, 3000, 1000);
delay(1000);
```

Αυτός ο κώδικας θα κάνει το `pin 4` να στείλει μια σειρά από υψηλά / χαμηλά σήματα που θα επαναλαμβάνονται στα 3 kHz (3000 φορές ανά δευτερόλεπτο), και ο τόνος θα διαρκέσει για 1000 ms (1 δευτερόλεπτο). Η λειτουργία-εντολή `tone` (τόνου) συνεχίζεται στο παρασκήνιο, ενώ το σχεδιάγραμμα κινείται προς την επόμενη εντολή. Δεν θέλουμε τα `servos` να αρχίσουν να κινούνται έως ότου τελειώσει ο τόνος να παίζεται, έτσι η εντολή ακολουθείται από την καθυστέρηση `delay(1000)` για να αφήσει το τόνο να τελειώσει πριν το σχεδιάγραμμα (κώδικας) προχωρήσει στον έλεγχο των `servo`.

2.7.3 Παράδειγμα προγράμματος StartResetIndicator

```

/*
 * Robotics with the BOE Shield - StartResetIndicator
 * Test the piezospeaker circuit.
 */

void setup() // Built in initialization block
{
  Serial.begin(9600);
  Serial.println("Beep!");

  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone
}

void loop() // Main loop auto-repeats
{
  Serial.println("Waiting for reset...");
  delay(1000);
}

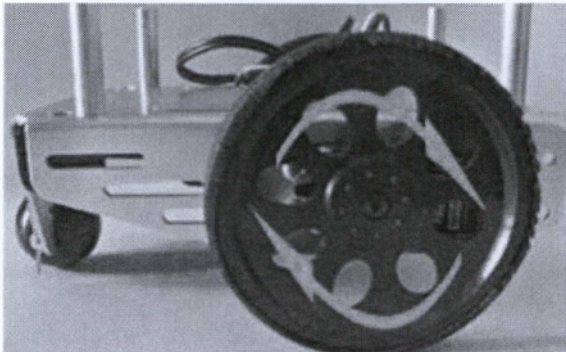
```

Το StartResetIndicator ξεκινά εμφανίζοντας το μήνυμα "Beep" στο Serial Monitor. Στη συνέχεια, αμέσως μετά την εκτύπωση του μηνύματος, η εντολή τόνου (tone();) εκπέμπει έναν ήχο 3 kHz για το πιεζοηλεκτρικό ηχείο για 1 δευτερόλεπτο. Επειδή οι οδηγίες εκτελούνται τόσο γρήγορα από το Arduino, θα πρέπει να φαίνεται σαν το μήνυμα να εμφανίζεται την ίδια στιγμή που το piezospeaker αρχίζει να παίζει τον ήχο.

Όταν ο τόνος ολοκληρώνεται, το σχεδιάγραμμα μπαίνει σε λειτουργία βρόχου, η οποία εμφανίζει το ίδιο μήνυμα, «Αναμονή για την επαναφορά ...», ξανά και ξανά. Κάθε φορά που το κουμπί επαναφοράς στο BOE Shield πατηθεί ή η ισχύς αποσυνδεθεί και να επανασυνδεθεί, το σχεδιάγραμμα ξεκινά ξανά με το μήνυμα "Beep" και τον τόνο 3 kHz.

2.8 Επανάλεγχος των Servo

Το επόμενο παράδειγμα δοκιμάζει το σέρβο που συνδέεται με το δεξιό τροχό, όπως φαίνεται παρακάτω. Το πρόγραμμα κάνει αυτό το τροχό να στρέφεται δεξιόστροφα για τρία δευτερόλεπτα, στη συνέχεια να σταματήσει για ένα δευτερόλεπτο, και μετά να στρέφεται αριστερόστροφα για τρία δευτερόλεπτα.



Δεξιόστροφα 3 δευτερόλεπτα

Σταμάτημα 1 δευτερόλεπτο

Αριστερόστροφα 3 δευτερόλεπτα

```

/*
 * Robotics with the BOE Shield - RightServoTest
 * Right servo turns clockwise three seconds, stops 1 second, then
 * counterclockwise three seconds.
 */

#include <Servo.h>                                // Include servo library

Servo servoRight;                                 // Declare right servo

void setup()                                     // Built in initialization block
{
  servoRight.attach(12);                          // Attach right signal to pin 12

  servoRight.writeMicroseconds(1300);             // Right wheel clockwise
  delay(3000);                                     // ...for 3 seconds

  servoRight.writeMicroseconds(1500);             // Stay still
  delay(1000);                                     // ...for 3 seconds

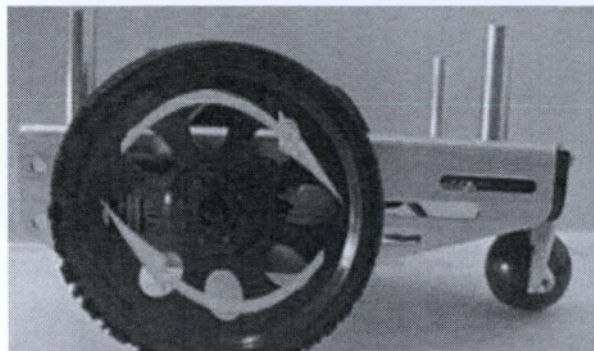
  servoRight.writeMicroseconds(1700);             // Right wheel counterclockwise
  delay(3000);                                     // ...for 3 seconds

  servoRight.writeMicroseconds(1500);             // Right wheel counterclockwise
}

void loop()                                       // Main loop auto-repeats
{                                                 // Empty, nothing needs repeating
}

```

Αντίστοιχα ελέγχουμε το σέρβο που συνδέεται με τον αριστερό τροχό, αλλάζοντας τις παραμέτρους μας στο πρόγραμμα.



Δεξιόστροφα 3 δευτερόλεπτα

Σταμάτημα 1 δευτερόλεπτο

Αριστερόστροφα 3 δευτερόλεπτα

```

/*
 * Robotics with the BOE Shield - LeftServoTest
 * Left servo turns clockwise three seconds, stops 1 second, then
 * counterclockwise three seconds.
 */

#include <Servo.h>                                // Include servo library

Servo servoLeft;                                  // Declare left servo

void setup()                                       // Built in initialization block
{
  servoLeft.attach(13);                           // Attach left signal to pin 13

  servoLeft.writeMicroseconds(1300);              // Left wheel clockwise
  delay(3000);                                     // ...for 3 seconds

  servoLeft.writeMicroseconds(1500);              // Stay still
  delay(1000);                                     // ...for 3 seconds

  servoLeft.writeMicroseconds(1700);              // Left wheel
  counterclockwise
  delay(3000);                                     // ...for 3 seconds

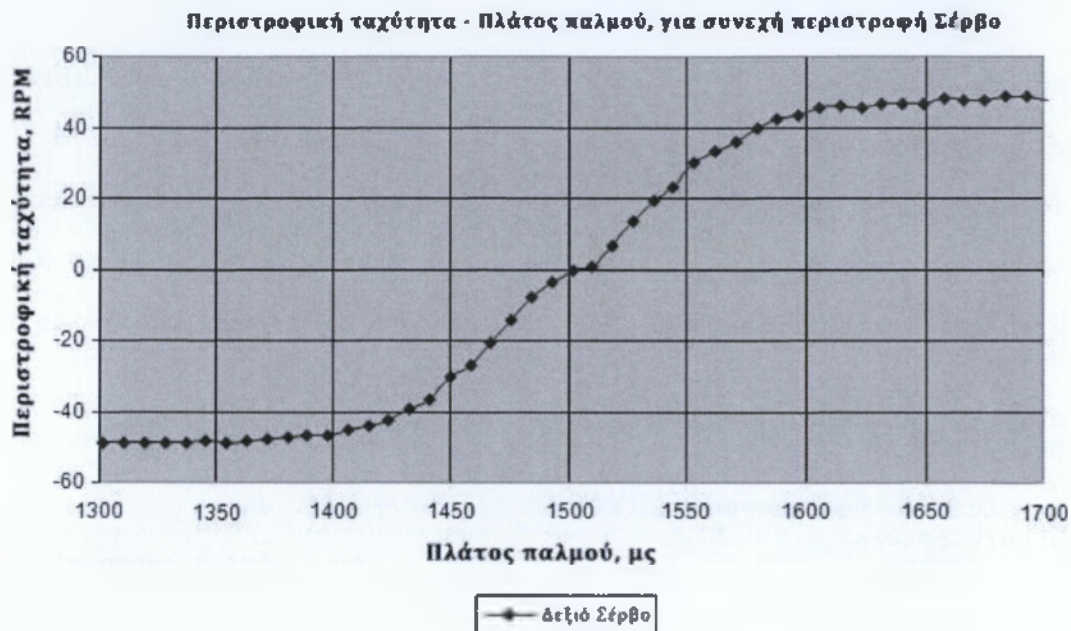
  servoLeft.writeMicroseconds(1500);              // Left wheel counterclockwise
}

void loop()                                       // Main loop auto-repeats
{                                                 // Empty, nothing needs repeating
}

```

2.9 Ρύθμιση ελέγχου ταχύτητας

Το παρακάτω γράφημα δείχνει το χρόνο παλμού (pulse time) σε σχέση με τη ταχύτητα των servo. Ο οριζόντιος άξονας του γραφήματος δείχνει το εύρος παλμού σε μικροδευτερόλεπτα (μs) και ο κάθετος άξονας δείχνει την απόκριση του σερβο σε RPM. Η δεξιόστροφη περιστροφή εμφανίζεται ως αρνητική και η αριστερόστροφη ως θετική. Αυτό το συγκεκριμένο γράφημα των σερβο, το οποίο ονομάζεται και καμπύλη μεταφοράς (transfer curve), κυμαίνεται περίπου από -48 με 48 RPM σε όλο το εύρος των παλμών δοκιμής από 1300 έως 1700 μικροδευτερόλεπτα.



Τρεις λόγοι για τους οποίους το γράφημα Μεταφοράς Καμπύλης είναι χρήσιμο :

1. Μπορούμε να πάρουμε μια καλή ιδέα για το τι να περιμένουμε από το σέρβο μας για ένα ορισμένο πλάτος παλμού . Ακολουθήστε την κάθετη γραμμή επάνω από το 1500 όπου το γράφημα τη διασχίζει , στη συνέχεια, ακολουθήστε την οριζόντια γραμμή και θα δείτε ότι υπάρχει μηδενική περιστροφή για τους 1500 μs παλμών . Γνωρίζαμε ήδη από τη συναρμολόγηση των servo ότι στα 1500 σταματά ένα σερβο , αλλά τώρα μπορούμε να παρατηρήσουμε ποιο συλλογικά αυτές τις τιμές.

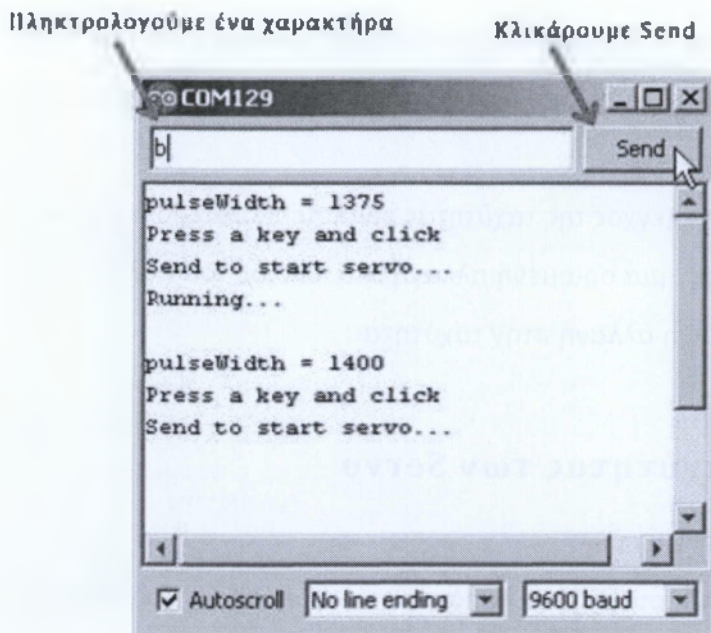
2. Η ταχύτητα δεν αλλάζει σημαντικά μεταξύ 1300 και 1400 μs παλμών . Έτσι , τα 1300 μs παλμών για τη δεξιόστροφη πλήρης ταχύτητα είναι υπερβολή ; Το ίδιο ισχύει και για τα 1600 έναντι των 1700 μs παλμών για την αριστερόστροφη περιστροφή . Αυτές οι υπερβολικές ρυθμίσεις ταχύτητας είναι χρήσιμες, επειδή είναι πιο πιθανό να οδηγηθεί σε συγγενικές ταχύτητες από ό, τι να πάρει δύο τιμές στο εύρος από 1400 έως 1600 μs.
3. Μεταξύ 1400 και 1600 μs , ο έλεγχος της ταχύτητας είναι περισσότερο ή λιγότερο γραμμικός . Σε αυτό το εύρος , μια ορισμένη αλλαγή στο πλάτος του παλμού, έχει ως αποτέλεσμα, μία αντίστοιχη αλλαγή στην ταχύτητα .

2.9.1 Έλεγχος ταχύτητας των Servo

Με το πρόγραμμα που παρουσιάζεται παρακάτω, μπορούμε να ελέγξουμε την ταχύτητα των servo RPM (και της κατεύθυνσης) για τιμές των παλμών από 1375 μs έως 1625 μs σε στάδια των 25 μs . Αυτές οι μετρήσεις της ταχύτητας θα μας βοηθήσουν στο να γίνει σαφές πως ο έλεγχος της διάρκειας παλμού των servo στο εύρος 1400-1600 μs ελέγχει και την ταχύτητα των servo . Το σχεδιάγραμμα ξεκινά εμφανίζοντας τη διάρκεια του παλμού, που είναι έτοιμη να σταλθεί ως σήμα ελέγχου των servo . Στη συνέχεια , περιμένει για να στείλουμε στον Arduino ένα χαρακτήρα με το Serial Monitor πριν να ξεκινήσει το servo . Τρέχει το servo για έξι δευτερόλεπτα , και κατά τη διάρκεια αυτού του χρόνου , μπορούμε να μετρήσουμε τον αριθμό των πλήρων στροφών που κάνει ο τροχός. Μετά από αυτό , ο βρόχος for επαναλαμβάνεται , και αυξάνει τη διάρκεια παλμού κατά 25 για το επόμενο τεστ .

Τοποθετούμε ένα σημάδι (σαν ένα κομμάτι κολλητική ταινία) στον τροχό έτσι ώστε να μπορούμε να δούμε πώς στρέφεται κατά τη διάρκεια των δοκιμών ταχύτητας του τροχού. Επίσης ρυθμίζουμε το BOE Shield - Bot έτσι ώστε οι τροχοί να μπορούν να περιστρέφονται ελεύθερα . Έπειτα ανοίγουμε το Serial Monitor και ορίζουμε το drop-down μενού με το «No line ending» και « 9600 baud» . Στη συνέχεια κάνουμε κλικ στο παράθυρο transmit στην κορυφή , πληκτρολογούμε οποιονδήποτε χαρακτήρα και κάνουμε κλικ στο κουμπί Αποστολή (send). Τέλος μετράμε τον αριθμό των στροφών

που κάνει ο τροχός και πολλαπλασιάζουμε επί 10 για τα RPMs . (Πρέπει να σημειώνουμε και την κατεύθυνση γιατί αλλάζει μετά από το 5το τεστ .)



```

/*
  Robotics with the BOE Shield - TestServoSpeed
  Send a character from the Serial Monitor to the Arduino to make it run
  the
  left servo for 6 seconds. Starts with 1375 us pulses and increases by
  25 us with each repetition, up to 1625 us. This sketch is useful for
  graphing speed vs. pulse width.
*/

#include <Servo.h>                                     // Include servo library

Servo servoLeft;                                       // Declare left servo signal
Servo servoRight;                                      // Declare right servo signal

void setup()                                           // Built in initialization block
{
  tone(4, 3000, 1000);                                 // Play tone for 1 second
  delay(1000);                                         // Delay to finish tone

  Serial.begin(9600);                                  // Set data rate to 9600 bps
  servoLeft.attach(13);                                // Attach left signal to P13
}

void loop()                                           // Main loop auto-repeats
{
  // Loop counts with pulseWidth from 1375 to 1625 in increments of 25.
  for(int pulseWidth = 1375; pulseWidth <= 1625; pulseWidth += 25)
  {
    Serial.print("pulseWidth = ");                    // Display pulseWidth value
    Serial.println(pulseWidth);
    Serial.println("Press a key and click");          // User prompt
  }
}

```

```

Serial.println("Send to start servo...");

while(Serial.available() == 0);           // Wait for character
Serial.read();                             // Clear character

Serial.println("Running...");
servoLeft.writeMicroseconds(pulseWidth); // Pin 13 servo speed =
pulse
delay(6000);                               // ..for 6 seconds
servoLeft.writeMicroseconds(1500);        // Pin 13 servo speed = stop
}

```

2.9.2 Πως λειτουργεί το πρόγραμμα TestServoSpeed

Το πρόγραμμα TestServoSpeed αυξάνει την τιμή μιας μεταβλητής που ονομάζεται εύρος παλμού (pulseWidth) για 25 κάθε φορά μέσα από ένα βρόχο.

```

// Loop counts with pulseWidth from 1375 to 1625 in increments of 25.
for(int pulseWidth = 1375; pulseWidth <= 1625; pulseWidth += 25)

```

Με κάθε επανάληψη του βρόχου «for», εμφανίζει την τιμή του επόμενου παλμού που θα σταλθεί στον servo του pin 13, μαζί με μια ενέργεια του χρήστη.

```

Serial.print("pulseWidth = ");           // Display pulseWidth value
Serial.println(pulseWidth);
Serial.println("Press a key and click"); // User prompt
Serial.println("Send to start servo...");

```

Μετά από το Serial.begin στο βρόχο ρύθμισης (setup loop), ο Arduino αναρριεί κάποια μνήμη για τους χαρακτήρες που έρχονται από το Serial Monitor . Αυτή η μνήμη συνήθως ονομάζεται σειριακό ρυθμιστικό (serial buffer) , και εκεί είναι που αποθηκεύονται οι τιμές ASCII από το Serial Monitor . Κάθε φορά που χρησιμοποιούμε το Serial.read για να πάρει ένα χαρακτήρα από το buffer (ρυθμιστικό) , ο Arduino αφαιρεί 1 από τον αριθμό των χαρακτήρων σε αναμονή στο buffer .

Μια κλήση στην Serial.available θα μας πει πόσοι χαρακτήρες είναι στο buffer. Αυτό το σχεδιάγραμμα χρησιμοποιεί την while(Serial.available() == 0) για να περιμένει μέχρι που το Serial Monitor στείλει ένα χαρακτήρα . Πριν προχωρήσει στο τρέξιμο των servo , χρησιμοποιεί τη Serial.read () για να αφαιρέσει το χαρακτήρα από το buffer . Θα

μπορούσαμε να χρησιμοποιήσουμε την `int myVar = Serial.read ()` για να αντιγράψουμε το χαρακτήρα σε μια μεταβλητή , αλλά δεδομένου ότι ο κώδικας δεν χρησιμοποιεί την αξία του χαρακτήρα για τη λήψη αποφάσεων , απλά καλεί την `Serial.read` και δεν αντιγράφει το αποτέλεσμα πουθενά. Το σημαντικό είναι ότι πρέπει να καθαρίσει το `buffer`, έτσι ώστε η `Serial.available ()` να επιστρέφει μηδέν την επόμενη φορά μέσω του βρόχου .

```
while(Serial.available() == 0);      // Wait for character
Serial.read();                       // Clear character
```

Αφού ο Arduino λάβει ένα χαρακτήρα από το πληκτρολόγιο, εμφανίζει το μήνυμα "Running ...» και στη συνέχεια κάνει τα servo να στρέφονται για 6 δευτερόλεπτα. Ο βρόχος `for` σε αυτό το κώδικας ξεκινά τη μεταβλητή `pulseWidth` στο 1375 και προσθέτει 25 σε αυτό με κάθε επανάληψη. Έτσι, για πρώτη φορά μέσω του βρόχου η `servoLeft` είναι 1375, τη δεύτερη φορά μέσα από αυτό είναι στα 1400, και ούτω καθεξής για όλη τη διαδρομή μέχρι το 1625.

Κάθε φορά μέσω του βρόχου, η `servoLeft.writeMicroseconds (pulseWidth)` χρησιμοποιεί την τιμή που η `pulseWidth` αποθηκεύει για την ρύθμιση της ταχύτητας του servo. Με αυτό τον τρόπο ενημερώνει την ταχύτητα του servo κάθε φορά που στέλνουμε ένα χαρακτήρα στον Arduino με το Serial Monitor.

```
Serial.println("Running...");
servoLeft.writeMicroseconds(pulseWidth); // Pin 13 speed=pulse
delay(6000);                             // ..for 6 seconds
servoLeft.writeMicroseconds(1500);       // Pin 13 speed=stop
```

2.9.3 Καταγραφή των δικών μας δεδομένων

Ο βρόχος στο πρόγραμμα TestServoSpeed μπορεί να τροποποιηθεί για να δοκιμαστεί σε κάθε πιθανή τιμή, έτσι ώστε να εξοικονομήσουμε χρόνο.

Αλλάζουμε τη δήλωση στην TestServoSpeed από:

```
for(int pulseWidth=1375; pulseWidth <= 1625; pulseWidth += 25)
```

Σε ...:

```
for(int pulseWidth=1300; pulseWidth <= 1700; pulseWidth += 20)
```

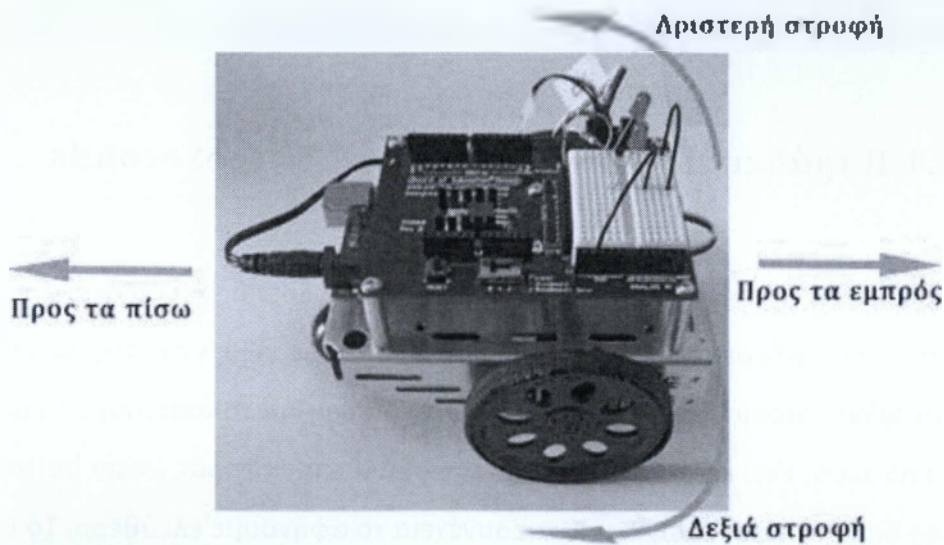
Τοποθετούμε το τροποποιημένο σχέδιο στον Arduino και το χρησιμοποιούμε για να συμπληρώσουμε κάθε RPM σε ένα εύρος από 1300 – 1700 μς, το οποίο αυξάνεται κάθε φορά κατά 20. (Αν θέλουμε να συμπληρώσουμε όλο τον πίνακα, χρησιμοποιούμε το pulseWidth += 10 στην παράμετρο increment.) Για να επαναλάβουμε τις μετρήσεις αυτές για τον δεξιό τροχό, αντικαταστήσουμε όλες τις τιμές από 13 σε 12 στο σχεδιάγραμμα.

3. Pulse Width and RPM for Parallax Continuous Rotation Servo							
Pulse Width (μς)	Rotational Velocity (RPM)	Pulse Width (μς)	Rotational Velocity (RPM)	Pulse Width (μς)	Rotational Velocity (RPM)	Pulse Width (μς)	Rotational Velocity (RPM)
1300	39	1410	35	1520	9	1630	38
1310	39	1420	34	1530	15	1640	38
1320	38	1430	32	1540	22	1650	38
1330	38	1440	29	1550	26	1670	39
1340	38	1450	26	1560	29	1680	39
1350	38	1460	22	1570	32	1690	39
1360	38	1470	15	1580	34	1700	39
1370	38	1480	9	1590	35		

1380	38	1490	3	1600	36		
1390	37	1500	0	1610	37		
1400	36	1510	3	1620	38		

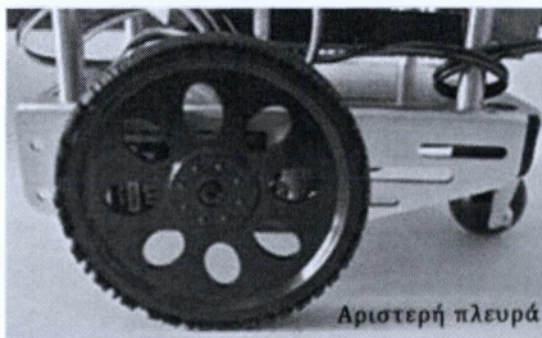
2.10 Πλοήγηση του BOE Shield-Bot

Το πρώτο βήμα είναι να προσανατολιστούμε! Η παρακάτω εικόνα μας δείχνει το μπροστινό μέρος, το πίσω, την αριστερή και τη δεξιά στροφή από την οπτική γωνία του BOE Shield-Bot.

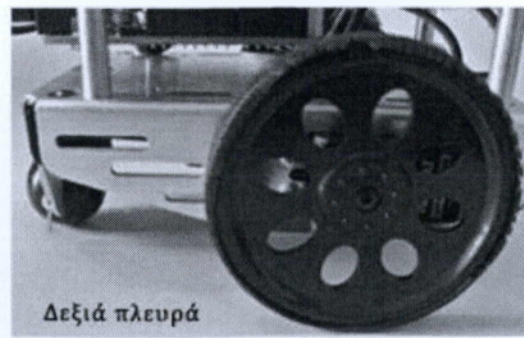


Πορεία προς τα εμπρός

Έχετε ποτέ σκεφτεί, σε ποια κατεύθυνση πρέπει να στραφούν οι ρόδες ενός αυτοκινήτου για προχωρήσει προς τα εμπρός; Οι τροχοί περιστρέφονται σε αντίθετες κατευθύνσεις στις αντίθετες πλευρές του αυτοκινήτου. Ομοίως, για να προχωρήσει το BOE Shield-Bot προς τα μπροστά, ο αριστερός του τροχός πρέπει να γυρίζει αριστερόστροφα και ο δεξιός του τροχός πρέπει να γυρίζει δεξιόστροφα.



← Αριστερόστροφα προς τα μπροστά



Δεξιόστροφα προς τα μπροστά →

Να θυμάστε ότι ένα πρόγραμμα πρέπει να χρησιμοποιεί την εντολή `writeMicroseconds` της βιβλιοθήκης `Servo` για να ελέγχει την ταχύτητα και την κατεύθυνση του κάθε servo. Στη συνέχεια, μπορεί να χρησιμοποιήσει την εντολή καθυστέρησης για να κρατήσει την πορεία των servos για ορισμένες ποσότητες του χρόνου (για κάποια χρονική στιγμή) πριν από την επιλογή νέων ταχυτήτων και κατευθύνσεων. Στο παρακάτω παράδειγμα θα δώσουμε την εντολή στο BOE Shield-Bot να κινηθεί μπροστά για περίπου τρία δευτερόλεπτα και στη συνέχεια να σταματήσει.

2.10.1 Παράδειγμα προγράμματος `F.ThreeSeconds`

Βεβαιωνόμαστε ότι ο διακόπτης λειτουργίας του BOE Shield έχει οριστεί στο 1 και η μπαταρία είναι συνδεδεμένη στον Arduino. Αφού έχουμε σώσει και φορτώσει το κώδικα στον Arduino, αποσυνδέουμε το καλώδιο programming και τοποθετούμε το BOE Shield-Bot στο πάτωμα. Ενώ κρατάμε πατημένο το κουμπί επαναφοράς (reset button), μετακινούμε το διακόπτη στη θέση 3, και στη συνέχεια το αφήνουμε ελεύθερο. Το BOE Shield-Bot πρέπει να προωθηθεί μπροστά για τρία δευτερόλεπτα και μετά να σταματήσει.

```
// Robotics with the BOE Shield - ForwardThreeSeconds
// Make the BOE Shield-Bot roll forward for three seconds, then stop.

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12

  // Full speed forward
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(3000); // ...for 3 seconds

  servoLeft.detach(); // Stop sending servo signals
  servoRight.detach();
}
```



```
void loop()                                // Main loop auto-repeats
{                                           // Empty, nothing needs
  repeating
}
```

2.10.2 Πως λειτουργεί το πρόγραμμα F.ThreeSeconds

Πρώτον, η βιβλιοθήκη Servo πρέπει να συμπεριληφθεί, έτσι ώστε το πρόγραμμά μας να μπορέσει να προσπελάσει τις λειτουργίες του:

```
#include <Servo.h>                          // Include servo library
```

Στη συνέχεια, οι υποδείξεις των Servo πρέπει να δηλωθούν και να δοθούν μοναδικά ονόματα για κάθε τροχό:

```
Servo servoLeft;                            // Declare left & right servos
Servo servoRight;
```

Ένα πρόγραμμα ξεκινά αυτόματα με τη λειτουργία setup. Τρέχει τον κώδικα που περιέχει μια φορά πριν από τη μετάβαση στην λειτουργία loop, η οποία αρχίζει να επαναλαμβάνεται αυτόματα. Από τη στιγμή που θέλουμε το BOE Shield-Bot να πάει μπροστά και να σταματήσει, όλος ο κώδικας μπορεί να τοποθετηθεί στη λειτουργία setup και να αφήσουμε τη λειτουργία loop άδεια, το οποίο δεν είναι λάθος.

Όπως συμβαίνει με όλα τα προγράμματα κίνησης, η πρώτη δράση που η setup λαμβάνει είναι να κάνει το piezospeaker να ακουστεί. Η κλήση της συνάρτησης tone μεταδίδει ένα σήμα στο ψηφιακό pin 4, το οποίο κάνει το piezospeaker να παίξει έναν ήχο 3 kHz που διαρκεί για 1 δευτερόλεπτο. Αφότου ότι η λειτουργία tone λειτουργεί στο παρασκήνιο, ο κώδικας κινείται, η delay(1000) αποτρέπει την κίνηση του BOE Shield-Bot έως ότου ο τόνος να σταματήσει.

```
void setup()                                // Built-in initialization
{
  tone(4, 3000, 1000);                       // Play tone for 1 second
  delay(1000);                               // Delay to finish tone
```

Στη συνέχεια, το αντικείμενο servoLeft συνδέεται με το ψηφιακό pin 13 και το αντικείμενο servoRight συνδέεται με το pin 12. Αυτό κάνει τις κλήσεις προς το

`servoLeft.writeMicroseconds` να επηρεάζουν τα σήματα ελέγχου των servo που στέλνονται στο pin 13. Ομοίως, οι κλήσεις προς το `servoRight.writeMicroseconds` θα επηρεάσουν τα σήματα που στέλνονται στο pin 12.

```
servoLeft.attach(13);           // Attach left signal to pin 13
servoRight.attach(12);        // Attach right signal to pin 12
```

Να θυμάστε ότι χρειαζόμαστε τους δεξιούς και αριστερούς τροχούς του BOE Shield-Bot να περιστρέφονται σε αντίθετες κατευθύνσεις για την προώθηση του. Η κλήση της συνάρτησης `servoLeft.writeMicroseconds(1700)` καθιστά το αριστερό servo να γυρίζει με πλήρη ταχύτητα αριστερόστροφα, και η κλήση της συνάρτησης `servoRight.writeMicroseconds(1300)` καθιστά το δεξιό servo να γυρίζει με πλήρη ταχύτητα δεξιόστροφα. Το αποτέλεσμα είναι η κίνηση προς τα εμπρός. Η κλήση της συνάρτησης `delay(3000)` κρατάει τα servo σε κίνηση με αυτή την ταχύτητα για τρία πλήρη δευτερόλεπτα. Μετά την καθυστέρηση (`delay(3000)`), η `servoLeft.detach` και η `servoRight.detach` διακόπτει τα σήματα των servo, το οποίο φέρει το ρομπότ σε ακινησία.

```
// Full speed forward
servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
servoRight.writeMicroseconds(1300); // Right wheel clockwise
delay(3000);                       // ...for 3 seconds

servoLeft.detach();                // Stop sending servo signals
servoRight.detach();
```

Μετά την λειτουργία του `setup`, το πρόγραμμα προχωρά αυτόματα στη λειτουργία του βρόχου (`loop`), η οποία επαναλαμβάνεται επ'αόριστον. Σε αυτή την περίπτωση την αφήνουμε κενή, επειδή το πρόγραμμα είναι ολοκληρωμένο, έτσι επαναλαμβάνει τίποτα, ξανά και ξανά, επ'αόριστον.

```
void loop()                        // Main loop auto-repeats
{
}
```

2.10.3 Διάφορες κινήσεις του BOE Shield-Bot

Το μόνο που χρειάζεται για να πάρουμε διαφορετικές κινήσεις από το BOE Shield-Bot είναι διαφορετικοί συνδυασμοί των `us` παραμέτρων στις κλήσεις των `servoLeft` και `servoRight writeMicroseconds`. Για παράδειγμα, αυτές οι δύο κλήσεις των συναρτήσεων θα κάνουν το BOE Shield-Bot να πάει προς τα πίσω:

```
// Full speed backwards
servoLeft.writeMicroseconds(1300); // Left wheel clockwise
servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
```

Αυτές οι δύο κλήσεις θα κάνουν το BOE Shield-Bot να περιστρέφεται σε μια θέση στρίβοντας προς τα αριστερά:

```
// Turn left in place
servoLeft.writeMicroseconds(1300); // Left wheel clockwise
servoRight.writeMicroseconds(1300); // Right wheel clockwise
```

Αντιστηχα, το BOE Shield-Bot περιστρέφεται σε μια θέση στρίβοντας προς τα δεξιά:

```
// Turn right in place
servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
```

2.10.3.1 Παράδειγμα προγράμματος ForwardLeftRightBackward

Με το παρακάτω παράδειγμα, συνδυάζουμε όλες αυτές τις εντολές σε ένα μόνο πρόγραμμα, έτσι ώστε το BOE Shield-Bot να προχωρήσει ευθεία, να στρίψει αριστερά, να στρίψει δεξιά και στη συνέχεια να κινηθεί προς τα πίσω.

```
// Robotics with the BOE Shield - ForwardLeftRightBackward
// Move forward, left, right, then backward for testing and tuning.

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;
```

```

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12

  // Full speed forward
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(2000); // ...for 2 seconds

  // Turn left in place
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(600); // ...for 0.6 seconds

  // Turn right in place
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(600); // ...for 0.6 seconds

  // Full speed backward
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(2000); // ...for 2 seconds

  servoLeft.detach(); // Stop sending servo signals
  servoRight.detach();
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

```

2.10.3.2 Ρύθμιση της ταχύτητας για βέλτιστη ευθεία

Αν το BOE Shield-Bot στρίβει ελαφρώς όταν το θέλουμε να πάει εξολοκλήρου ευθεία, η λύση είναι αρκετά απλή, απλά επιβραδύνουμε τον ταχύτερο τροχό. Από το γράφημα καμπύλη μεταφοράς των servo, θυμόμαστε ότι έχουμε καλύτερο έλεγχο της ταχύτητας των servo στο εύρος των 1400-1600 μς.

4. Us Parameters in writeMicroseconds(us)

Πλήρης ταχύτητα δεξιόστροφα	Αρχή ζώνης Γραμμικής ταχύτητας	Τέλεια ακινητοποίηση	Τελείωμα ζώνης Γραμμικής ταχύτητας	Πλήρης ταχύτητα αριστερόστροφα
1300	1400	1500	1600	1700

Ας πούμε ότι το BOE Shield-Bot μετακινείται σταδιακά προς τα αριστερά. Αυτό σημαίνει ότι ο δεξιός τροχός γυρίζει πιο γρήγορα από τον αριστερό. Δεδομένου ότι ο αριστερός τροχός έχει ήδη πιάσει τη μέγιστη ταχύτητα, για να ισιώσουμε την πορεία του ρομπότ θα πρέπει ο δεξιός τροχός να επιβραδυνθεί. Για να επιτευχτεί αυτό, θα πρέπει να αλλάξουμε την παράμετρο «us» της `servoRight.writeMicroseconds(us)` σε μια τιμή πιο κοντά στο 1500. Στην αρχή δοκιμάζουμε το 1400, αν εξακολουθεί να πηγαίνει πάρα πολύ γρήγορα, τότε αυξάνουμε την τιμή της παραμέτρου κατά 10 κάθε φορά, μέχρις ότου το BOE Shield-Bot να σταματήσει να στρέφεται προς τα αριστερά. Εάν οποιαδήποτε προσαρμογή υπερβεί την τιμή που θέλουμε να φτάσουμε και το BOE Shield-Bot ξεκινά να στρέφεται προς τα δεξιά, τότε αρχίζουμε να μειώνουμε την παράμετρο μας σε μικρότερες ποσότητες και συνεχίζουμε μέχρις ότου το BOE Shield-Bot να μην αποκλίνει καθόλου κινούμενο ευθεία. Αυτό ονομάζεται *iterative process* (επαναληπτική διαδικασία), πράγμα που σημαίνει ότι χρειάζονται επαναλαμβανόμενες προσπάθειες και βελτιώσεις για να φτάσουμε στην σωστή τιμή.

2.10.3.3 Συντονισμός των στροφών

Το χρονικό διάστημα που το BOE Shield-Bot ξοδεύει για να περιστραφεί σε μια καθορισμένη θέση, καθορίζει και το πόσο πολύ θα στρίψει. Έτσι, για να συντονίσουμε μια στροφή, το μόνο που χρειάζεται να κάνουμε είναι να ρυθμίσουμε την παράμετρο `ms` της συνάρτησης `delay`, έτσι ώστε να το κάνουμε να γυρίσει σε διαφορετικό χρονικό διάστημα. Ας πούμε ότι το BOE Shield-Bot γυρίζει μόλις λίγο παραπάνω από 90° (1/4 του πλήρους κύκλου). Δοκιμάζουμε `delay(580)` ή ίσως και `delay(560)`, αν δεν γυρίσει αρκετά, το κάνουμε να τρέξει περισσότερο με την αύξηση της παραμέτρου `ms` της συνάρτησης `delay`, δοκιμάζοντας να την αυξάνουμε (ή μειώνουμε) κατά 20 `ms` κάθε φορά.

Αν βρεθούμε στην κατάσταση που μια τιμή ελαφρώς υπερβαίνει τις 90° και η άλλη ελαφρώς υστερεί, επιλέγουμε την τιμή που το κάνει να στραφεί λίγο πιο μακριά, στη συνέχεια επιβραδύνουμε λίγο τα servos. Στην περίπτωση της αριστερής στροφής, και οι 2 `us` παράμετροι της `writeMicroseconds` θα πρέπει να αλλάξουν από το 1300 σε κάτι πιο κοντά στο 1500. Ξεκινάμε με το 1400 και στη συνέχεια αυξάνουμε σταδιακά τις

τιμές για να επιβραδύνουμε και τα δύο servos. Για την δεξιά στροφή, ξεκινάμε αλλάζοντας τις παραμέτρους από 1700 σε 1600, και στη συνέχεια να πειραματιζόμαστε μειώνοντας κατά 10 μέχρις ότου να έχουμε τα αποτελέσματα που θέλουμε.

2.10.3.4 Επιτόπια στροφή 90°

Τροποποιούμε το πρόγραμμα ForwardLeftRightBackward έτσι ώστε να κάνει ακριβή 90° επιτόπια στροφή. Ενημερώνουμε την ετικέτα του κάθε servo με μια σημείωση σχετικά με την κατάλληλη «ms» παράμετρο της συνάρτησης «delay» για μια στροφή 90°. Επίσης ενημερώνουμε την παράμετρο «ms» της συνάρτησης «delay» με τις τιμές που καθορίσαμε για την ευθεία και προς τα πίσω κίνηση του ρομπότ.

```
// Robotics with the BOE Shield - ForwardPivotLeftBackward
// Move forward, pivot forward left, pivot backward left, then backward
for testing and tuning.

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12

  // Full speed forward
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(2000); // ...for 2 seconds

  // Pivot forward-left
  servoLeft.writeMicroseconds(1500); // Left wheel stop
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(960); // ...for 0.96 seconds

  // Pivot backward-left
  servoLeft.writeMicroseconds(1500); // Left wheel stop
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(960); // ...for 0.96 seconds

  // Full speed backward
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(2000); // ...for 2 seconds
```

```

servoLeft.detach(); // Stop sending servo signals
servoRight.detach();

}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

// Robotics with the BOE Shield - ForwardPivotRightBackward
// Move forward, pivot forward right, pivot backward right, then
backward for testing and tuning.

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12

  // Full speed forward
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(2000); // ...for 2 seconds

  // Pivot forward-right
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1500); // Right wheel stop
  delay(960); // ...for 0.96 seconds

  // Pivot backward-right
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1500); // Right wheel stop
  delay(960); // ...for 0.96 seconds

  // Full speed backward
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(2000); // ...for 2 seconds

  servoLeft.detach(); // Stop sending servo signals
  servoRight.detach();

}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

```

2.10.4 Υπολογισμός Αποστάσεων

Σε πολλούς διαγωνισμούς ρομποτικής, μια πιο ακριβή πλοήγηση του ρομπότ σημαίνει και καλύτερα αποτελέσματα. Ένας δημοφιλής διαγωνισμός ρομποτικής (entry-level) ονομάζεται «dead reckoning». Ο μοναδικός στόχος του διαγωνισμού είναι να κάνεις το ρομπότ σου να πάει σε μία ή περισσότερες θέσεις και στη συνέχεια να επιστρέψει ακριβώς από όπου ξεκίνησε. Σίγουρα θα θυμάστε να ρωτάτε τους γονείς σας αυτή την ερώτηση, ξανά και ξανά, ενώ ταξιδεύατε για διακοπές ή συγγενικά σπίτια:

«Φτάνουμε σύντομα;»

Ίσως όταν μεγαλώσατε λίγο, και μάθατε διαίρεση στο σχολείο, αρχίσατε να παρακολουθείτε (και να μετράτε) τις πινακίδες για να δείτε πόσο μακριά ήσασταν μέχρι τον προορισμό σας. Στη συνέχεια, θα ελέγχατε το ταχύμετρο του αυτοκινήτου. Με τη διαίρεση της ταχύτητας στην απόσταση, είχατε μια αρκετά καλή εκτίμηση του χρόνου που θα χρειαζόταν για να φτάσετε εκεί. Μπορεί να μην είχατε σκεφτεί με αυτούς τους ακριβείς όρους, αλλά η επομένη εξίσωση ήταν αυτή που χρησιμοποιούσατε:

$$time = \frac{distance}{speed}$$

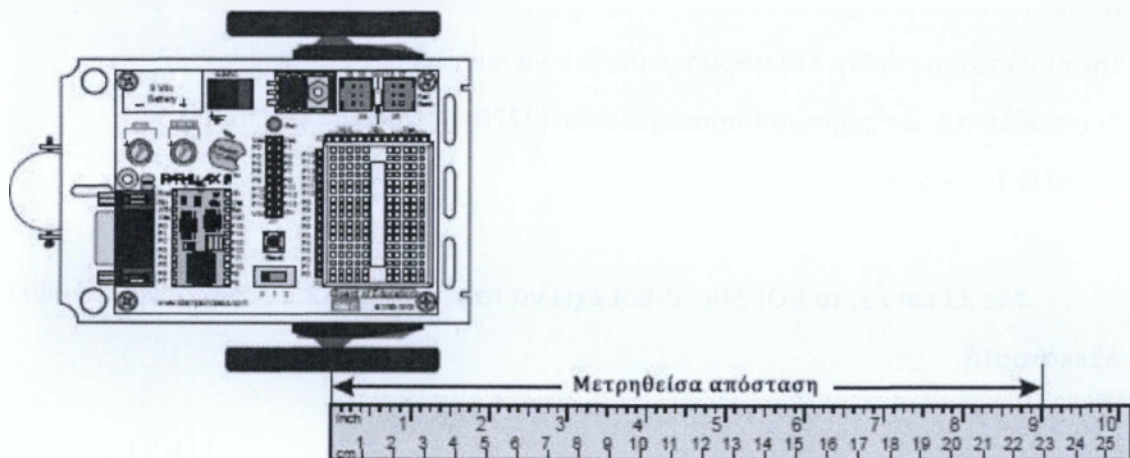
Αν είμαστε 200 χιλιόμετρα μακριά από τον προορισμό μας, και αν ταξιδεύαμε με 100 χιλιόμετρα την ώρα, θα φτάναμε στον προορισμό μας σε 2 ώρες.

$$\begin{aligned} time &= \frac{200 \text{ kilometers}}{100 \text{ kilometers/hour}} \\ &= 200 \text{ km} \times \frac{1 \text{ hour}}{100 \text{ km}} \\ &= 2 \text{ hours} \end{aligned}$$

Μπορούμε να κάνουμε την ίδια άσκηση με το BOE Shield-Bot, με την εξαίρεση ότι μπορούμε να αποφασίζουμε πόσο μακριά είναι ο προορισμός. Εδώ είναι η εξίσωση που χρησιμοποιούμε:

$$\text{servo run time} = \frac{\text{BOE Shield - Bot distance}}{\text{BOE Shield - Bot speed}}$$

Τρέχουμε το πρόγραμμα ForwardOneSecond και τοποθετούμε το BOE Shield-Bot δίπλα σε έναν χάρακα. Σιγουρευόμαστε ότι το σημείο όπου ο τροχός αγγίζει το έδαφος είναι ευθυγραμμισμένο με τη θέση των 0 cm στο χάρακα. Τέλος μετράμε πόσο μακριά ταξίδεψε το BOE Shield-Bot με την καταγραφή της μέτρησης, όπου ο τροχός αγγίζει πλέον το έδαφος.



```
// Robotics with the BOE Shield - ForwardOneSecond
// Make the BOE Shield-Bot roll forward for one seconds, then stop.

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right
servos
Servo servoRight;

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12
  // Full speed forward
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(1000); // ...for 1 second
```

```

servoLeft.detach(); // Stop sending servo signals
servoRight.detach();
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

```

Η απόσταση που μόλις καταγράψαμε (23cm) είναι η ταχύτητα του BOE Shield-Bot σε μονάδες ανά δευτερόλεπτο. Τώρα, μπορούμε να υπολογίσουμε πόσα δευτερόλεπτα το BOE Shield-Bot πρέπει να κινηθεί για να καλύψει τη συγκεκριμένη απόσταση.

Λαμβάνοντας υπόψη ότι οι υπολογισμοί μας είναι σε δευτερόλεπτα και η συνάρτηση delay χρειάζεται παραμέτρους που είναι σε χιλιοστά του δευτερολέπτου, πρέπει να μετατρέψουμε τα δευτερόλεπτα σε χιλιοστά του δευτερολέπτου πολλαπλασιάζοντας με 1000. Στη συνέχεια, χρησιμοποιούμε αυτήν την τιμή στην κλήση της συνάρτησης delay. Για παράδειγμα, για να κινηθεί το ρομπότ για 2,22 δευτερόλεπτα, θα χρησιμοποιήσουμε delay(2220) μετά από την κλήση της writeMicroseconds.

Με 23 cm / s, το BOE Shield-Bot έχει να ταξιδέψει για 2,22 s ώστε να καλύψει 51 εκατοστά.

$$\begin{aligned}
 time &= \frac{51 \text{ cm}}{23 \text{ cm/s}} & time(ms) &= time(s) \times \frac{1000 \text{ ms}}{s} \\
 &= 51 \text{ cm} \times \frac{1 \text{ s}}{23 \text{ cm}} & &= 2.22 \text{ s} \times \frac{1000 \text{ ms}}{s} \\
 &= 2.22 \text{ s} & &= 2220 \text{ ms}
 \end{aligned}$$

Όποτε χρησιμοποιούμε την delay(2220) μετά από την κλήση της συνάρτησης writeMicroseconds:

```

servoLeft.writeMicroseconds(1700);
servoRight.writeMicroseconds(1300);
delay(2220);

```

$$ms\ servo\ run\ time = \frac{BOE\ Shield - Bot\ distance}{BOE\ Shield - bot\ speed} \times \frac{1000\ ms}{s}$$

2.10.5 Σταδιακή λειτουργία ελιγμών

Το «Ramping» είναι ένας τρόπος για να αυξήσουμε ή να μειώσουμε σταδιακά την ταχύτητα των servos αντί για την απότομη έναρξη ή διακοπή τους. Αυτή η τεχνική μπορεί να αυξήσει το προσδόκιμο ζωής και των μπαταριών και των servo.

Το παρακάτω πρόγραμμα είναι ένα παράδειγμα για το πώς σταδιακά επιταχύνει μέχρι να φτάσει τη μέγιστη ταχύτητα. Ο βρόχος «for» δηλώνει μια int μεταβλητή που ονομάζεται speed (ταχύτητα) και τη χρησιμοποιεί για να επαναληφθεί ο βρόχος 100 φορές. Με κάθε επανάληψη του βρόχου, η τιμή της speed αυξάνεται κατά 2, λόγω της έκφρασης «speed+=2» στην παράμετρο «increment» του βρόχου. Επειδή η μεταβλητή speed είναι σε κάθε κλήση της παραμέτρου us της writeMicroseconds, επηρεάζει την τιμή κάθε φορά που ο βρόχος επαναλαμβάνεται. Με τα 20 ms καθυστέρηση μεταξύ κάθε επανάληψης, ο βρόχος επαναλαμβάνεται περίπου 50 φορές ανά δευτερόλεπτο. Αυτό σημαίνει ότι περνάει 1 δευτερόλεπτο για να φτάσει η μεταβλητή speed στο 100 σε βήματα των 2, και σε εκείνο το σημείο, και τα δύο servos θα κινούνται για τη μέγιστη ταχύτητα.

```
for(int speed = 0; speed <= 100; speed+=2)
{
  servoLeft.writeMicroseconds(1500+speed);
  servoRight.writeMicroseconds(1500-speed);
  delay(20);
}
```

Ας ρίξουμε μια πιο προσεκτική ματιά στα στάδια του βρόχου for:

- Πρώτο στάδιο: η speed (ταχύτητα) είναι 0, έτσι και οι δύο κλήσης της writeMicroseconds καταλήγουν με τις «us» παραμέτρους στα 1500.
- Δεύτερο στάδιο: η ταχύτητα είναι 2, οπότε έχουμε servoLeft.writeMicroseconds (1502) και servoRight.writeMicroseconds (1498).

- Τρίτο στάδιο: η ταχύτητα είναι 4, οπότε έχουμε `servoLeft.writeMicroseconds (1504)` και `servoRight.writeMicroseconds (1496)`.
- Συνεχίζουμε με αυτό τον τρόπο μέχρι το ...
- 50^οο στάδιο: η ταχύτητα είναι 100, με `servoLeft.writeMicroseconds (1600)` και `servoRight.writeMicroseconds (1400)`. Αυτές οι τιμές είναι πολύ κοντά στην μέγιστη ταχύτητα, οι τιμές 1700 και 1300 είναι υπερβολή.

2.10.5.1 Παράδειγμα προγράμματος - StartAndStopWithRamping

```
// Robotics with the BOE Shield - StartAndStopWithRamping
// Ramp up, go forward, ramp down.

#include <Servo.h>                                // Include servo library

Servo servoLeft;                                // Declare left and right servos
Servo servoRight;

void setup()                                     // Built-in initialization block
{
  tone(4, 3000, 1000);                          // Play tone for 1 second
  delay(1000);                                   // Delay to finish tone

  servoLeft.attach(13);                          // Attach left signal to pin 13
  servoRight.attach(12);                        // Attach right signal to pin 12

  for(int speed = 0; speed <= 100; speed += 2) // Ramp up to full speed.
  {
    servoLeft.writeMicroseconds(1500+speed); // us =
1500,1502,...1598,1600
    servoRight.writeMicroseconds(1500-speed); // us =
1500,1498,...1402,1400
    delay(20);                                   // 20 ms at each speed
  }

  delay(1500);                                   // Full speed for 1.5 seconds

  for(int speed = 100; speed >= 0; speed -= 2) // Ramp from full speed
to stop
  {
    servoLeft.writeMicroseconds(1500+speed); // us =
1600,1598,...1502,1500
    servoRight.writeMicroseconds(1500-speed); // us =
1400,1402,...1498,1500
    delay(20);                                   // 20 ms at each speed
  }

  servoLeft.detach();                            // Stop sending servo signals
  servoRight.detach();

}
void loop()                                     // Main loop auto-repeats
{                                               // Empty, nothing to repeat
}
```

2.10.5.2 Προσθήκη Ramping σε άλλους ελιγμούς

Μπορούμε να δημιουργήσουμε προγράμματα που να συνδυάζουν ramping με άλλους ελιγμούς. Εδώ είναι ένα παράδειγμα για το πώς επιταχύνει μέχρι την μέγιστη ταχύτητα πηγαίνοντας προς τα πίσω αντί για εμπρός. Η μόνη διαφορά είναι ότι η τιμή της ταχύτητας ξεκινάει στο μηδέν και φτάνει έως το -100.

```
for(int speed = 0; speed >= -100; speed -= 2)// Ramp stop to full
reverse
{
  servoLeft.writeMicroseconds(1500+speed); // us = 1500,1498,
1496...1400
  servoRight.writeMicroseconds(1500-speed); // us = 1500,1502,
1508...1600
  delay(20); // 20 ms at each speed
}
```

Επίσης μπορεί να επιταχύνει κατά την διάρκεια μιας στροφής. Το παρακάτω παράδειγμα παρουσιάζει επιταχυνόμενη στροφή προς τα δεξιά. Παρατηρούμε ότι αντί για 1500+speed για έναν τροχό και 1500-speed για τον άλλο, τώρα είναι και τα δυο 1500+speed. Για επιταχυνόμενη αριστερή στροφή, θα πρέπει και τα δυο να είναι 1500-speed.

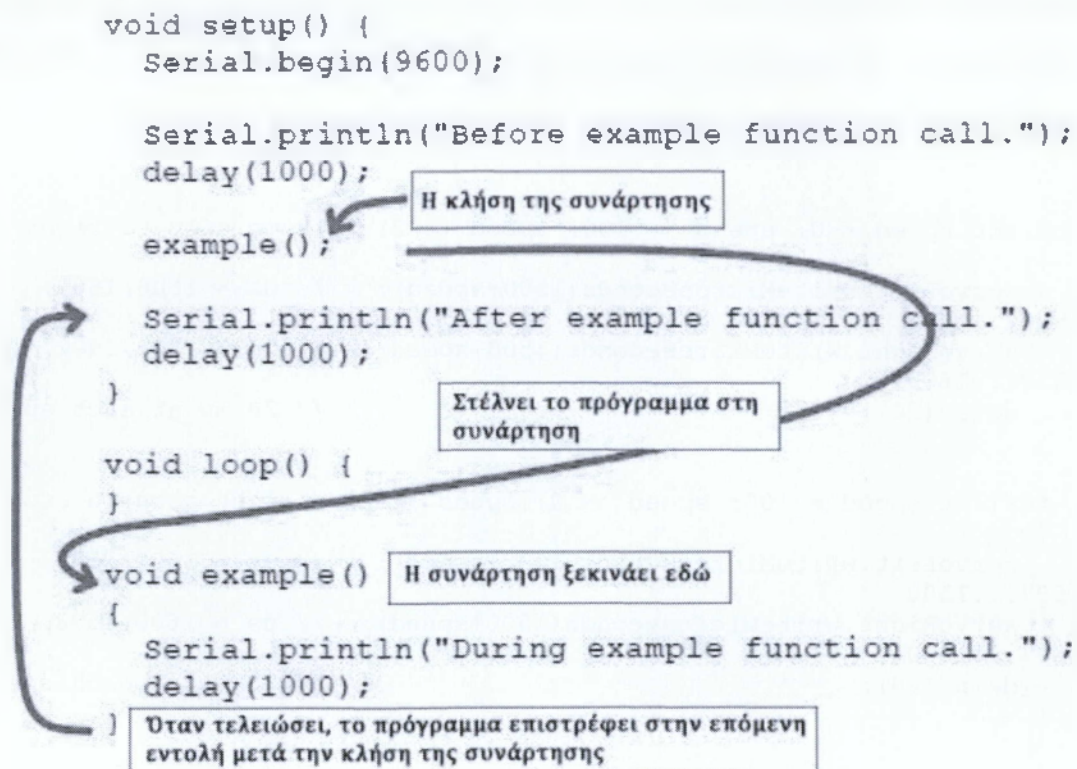
```
for(int speed = 0; speed <= 100; speed += 2) // Ramp stop to right turn
{
  servoLeft.writeMicroseconds(1500+speed); // us = 1500,1502,
1508...1600
  servoRight.writeMicroseconds(1500+speed); // us = 1500,1502,
1508...1600
  delay(20); // 20 ms at each speed
}

for(int speed = 100; speed >= 0; speed -= 2)// right turn to stop
{
  servoLeft.writeMicroseconds(1500+speed); // us = 1600,1598,
1597...1500
  servoRight.writeMicroseconds(1500+speed); // us = 1600,1598,
1597...1500
  delay(20); // 20 ms at each speed
}
```

2.10.6 Απλούστευση πλοήγησης με Συναρτήσεις

Ένας βολικός τρόπος για να εκτελέσουμε προ-προγραμματιστικούς ελιγμούς είναι με συναρτήσεις. Οι συναρτήσεις `setup` (ρύθμισης) και `loop` (βρόχου) είναι φτιαγμένες στη γλώσσα `Arduino`, αλλά μπορούμε να προσθέσουμε περισσότερες συναρτήσεις που κάνουν συγκεκριμένα ενέργειες για το πρόγραμμα μας.

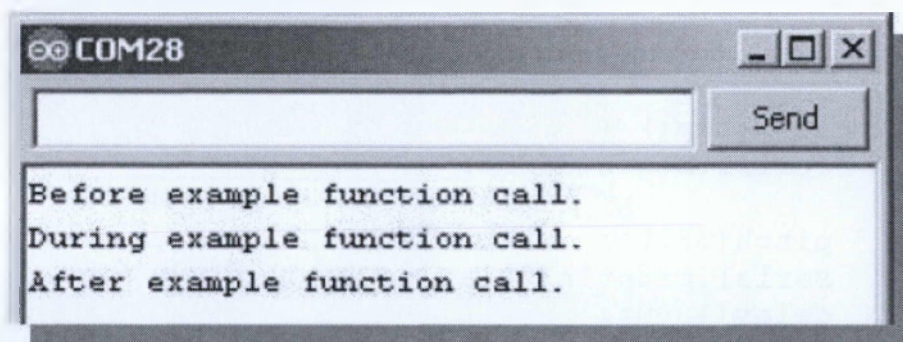
Το παρακάτω κομμάτι ενός προγράμματος περιέχει μια συνάρτηση που ονομάζεται `example` (παράδειγμα), η οποία προστίθεται στο τέλος, κάτω από τη συνάρτηση `loop`. Ξεκινάει και ονομάζεται με τον ορισμό της συνάρτησης «`void example()`». Η κενή παρένθεση σημαίνει ότι δεν χρειάζεται οποιαδήποτε παραμέτρους για να κάνει τη δουλειά της, και το «`void`» δείχνει ότι δεν επιστρέφει τιμή. Οι αγκύλες `{}` που ακολουθούν αυτόν τον ορισμό περιέχουν το μπλοκ του κώδικα της συνάρτησης `example`.



Υπάρχει μια κλήση της συνάρτησης `example` στη συνάρτηση `setup`, που επισημαίνονται στο παραπάνω κομμάτι του κώδικα. Η γραμμή `example();` λέει στο πρόγραμμα να πάει να βρει τη συνάρτηση με αυτό το όνομα, να εκτελέσει το κώδικα

της και να επανέλθει όταν γίνει. Έτσι, το πρόγραμμα πάει προς τα κάτω στο void example() και εκτελεί τις δύο εντολές στα άγκιστρα του. Στη συνέχεια, επιστρέφει στη κλήση της συνάρτησης και συνεχίζει από εκεί. Εδώ είναι η σειρά των γεγονότων που βλέπουμε όταν εκτελέσουμε το πρόγραμμα:

1. Το Serial Monitor εμφανίζει τη πρόταση «Before example function call.»
2. Μετά από ένα δευτερόλεπτο καθυστέρηση, η οθόνη εμφανίζει τη πρόταση «During example function call.». Επειδή η κλήση της example() στέλνει τον κώδικα στην void example(), η οποία έχει τη γραμμή του κώδικα που τυπώνει το μήνυμα και ακολουθείται από μια καθυστέρηση 1 δευτερολέπτου. Στη συνέχεια, η συνάρτηση επιστρέφει μετά την κλήση της example, μέσα στην setup.
3. Το Serial Monitor εμφανίζει τη πρόταση «After example function call.»



2.10.6.1 Παράδειγμα προγράμματος - SimpleFunctionCall

```
// Robotics with the BOE Shield - SimpleFunctionCall
// This sketch demonstrates a simple function call.

void setup() {
  Serial.begin(9600);

  Serial.println("Before example function call.");
  delay(1000);

  example(); // This is the function call

  Serial.println("After example function call.");
  delay(1000);
}

void loop() {
}

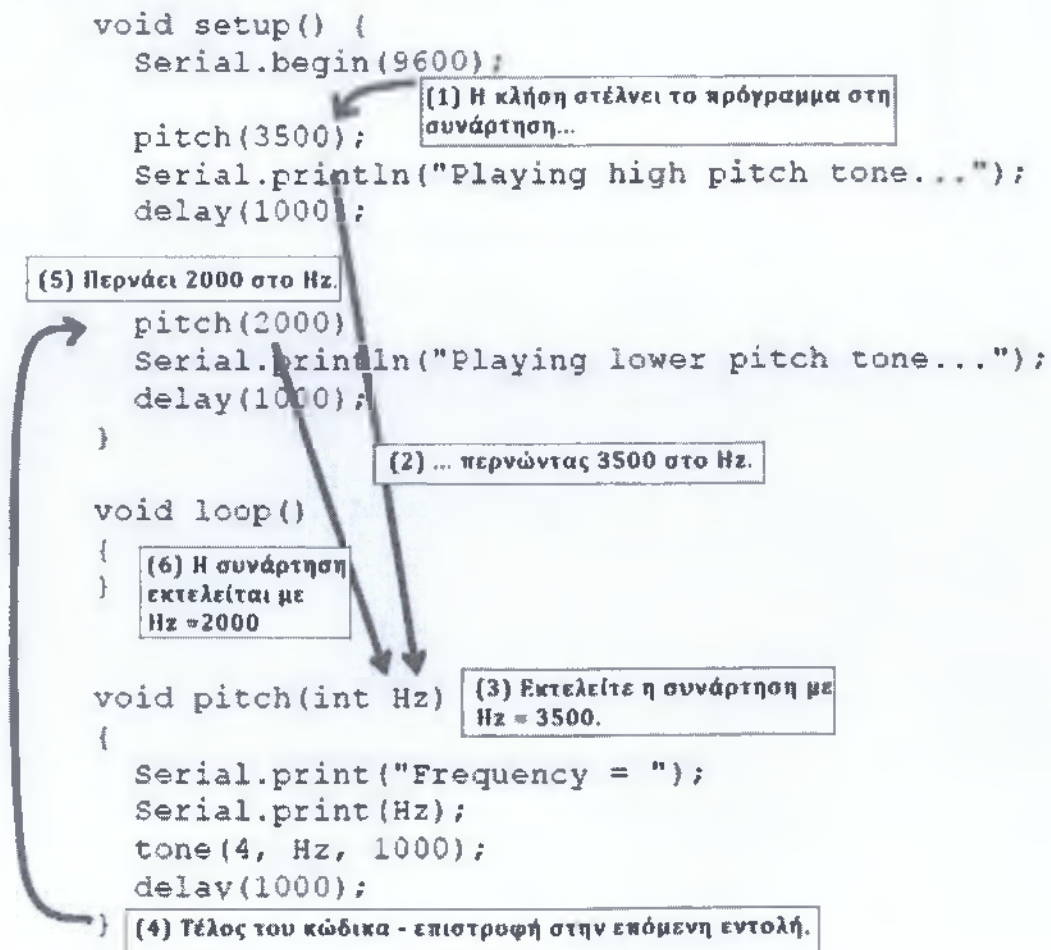
void example() // This is the function
```

```
{
  Serial.println("During example function call.");
  delay(1000);
}
```

2.10.6.2 Κλήση συναρτήσεων με παραμέτρους

Μια συνάρτηση μπορεί να έχει μία ή περισσότερες παραμέτρους (δεδομένα που η συνάρτηση λαμβάνει και χρησιμοποιεί όταν καλείται). Το παρακάτω πρόγραμμα δείχνει τη συνάρτηση `pitch`, η οποία δηλώνεται με `void pitch(int Hz)`.

Έτσι, όταν η πρώτη κλήση της συνάρτησης `pitch` γίνεται με `pitch(3500)`, η ακέραια τιμή 3500 περνιέται στη `Hz`. Αυτό φορτίζει τη `Hz` με την τιμή 3500, για να χρησιμοποιηθεί κατά τη διάρκεια του κώδικα της συνάρτησης `pitch`. Η δεύτερη κλήση, `pitch(2000)`, αρχικοποιεί τη `Hz` με την τιμή 2000 κατά τη διάρκεια του δεύτερου ταξιδιού μέσω του κώδικα της συνάρτησης `pitch`.



2.10.6.3 Παράδειγμα προγράμματος - FunctionCallWithParameter

```
// Robotics with the BOE Shield - FunctionCallWithParameter
// This program demonstrates a function call with a parameter.

void setup() {
  Serial.begin(9600);

  Serial.println("Playing higher pitch tone...");
  pitch(3500);      // pitch function call passes 3500 to Hz parameter
  delay(1000);

  Serial.println("Playing lower pitch tone...");
  pitch(2000);     // pitch function call passes 2000 to Hz parameter
  delay(1000);
}

void loop()
{
}

void pitch(int Hz)    // pitch function with Hz declared as a parameter
{
  Serial.print("Frequency = ");
  Serial.println(Hz);
  tone(4, Hz, 1000);
  delay(1000);
}
```

2.10.6.4 Επέκταση συναρτήσεων με δυο παραμέτρους

Εδώ είναι μια τροποποιημένη συνάρτηση pitch που δέχεται δύο παραμέτρους: Hz και ms. Αυτή η νέα συνάρτηση Pitch ελέγχει πόσο χρόνο διαρκεί ο τόνος.

```
void pitch(int Hz, int ms)
{
  Serial.print("Frequency = ");
  Serial.println(Hz);
  tone(4, Hz, ms);
  delay(ms);
}
```

Εδώ είναι οι δύο κλήσεις στην τροποποιημένης συνάρτησης pitch, μία για τόνο 0.5 δευτερολέπτων και 3500 Hz, και η άλλη για 1,5 δευτερόλεπτα και 2000 Hz:

```
pitch(3500, 500);
pitch(2000, 1500);
```

Κάθε μία από τις κλήσεις της `pitch` περιλαμβάνουν δύο τιμές, μία για να περάσει στην παράμετρο `Hz`, και άλλη μια για να περάσει στην παράμετρο `ms`. Ο αριθμός των τιμών σε μια κλήση συνάρτησης πρέπει να ταιριάζει με τον αριθμό των παραμέτρων στον ορισμό της εν λόγω συνάρτησης, αλλιώς το πρόγραμμα δεν θα τρέχει.

```
// Robotics with the BOE Shield - FunctionCallWith2Parameter
// This program demonstrates a function call with 2 parameters.

void setup() {
  Serial.begin(9600);

  Serial.println("Playing higher pitch tone...");

  pitch(3500, 500); // pitch function call passes 3500 to Hz parameter
  and 500 to ms parameter

  delay(1000);

  Serial.println("Playing lower pitch tone...");

  pitch(2000, 1500); // pitch function call passes 2000 to Hz
  parameter and 1500 to ms parameter

  delay(1000);
}

void loop()
{
}

void pitch(int Hz, int ms) // pitch function with Hz and ms declared
as parameters
{
  Serial.print("Frequency = ");
  Serial.println(Hz);
  tone(4, Hz, ms);
  delay(ms);
}
```

2.10.6.5 Εισαγωγή ελιγμών μέσα σε συναρτήσεις

Στο παρακάτω παράδειγμα βάλαμε τις πορείες forward, turnLeft, turnRight, και backward μέσα σε συναρτήσεις.

```
// Robotics with the BOE Shield - MovementsWithSimpleFunctions
// Move forward, left, right, then backward for testing and tuning.

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12

  forward(2000); // Go forward for 2 seconds
  turnLeft(600); // Turn left for 0.6 seconds
  turnRight(600); // Turn right for 0.6 seconds
  backward(2000); // go backward for 2 seconds

  disableServos(); // Stay still indefinitely
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

void forward(int time) // Forward function
{
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(time); // Maneuver for time ms
}

void turnLeft(int time) // Left turn function
{
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(time); // Maneuver for time ms
}

void turnRight(int time) // Right turn function
{
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(time); // Maneuver for time ms
}

void backward(int time) // Backward function
{
```

```

servoLeft.writeMicroseconds(1300); // Left wheel clockwise
servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
delay(time); // Maneuver for time ms
}

void disableServos() // Halt servo signals
{
servoLeft.detach(); // Stop sending servo signals
servoRight.detach();
}

```

2.10.6.6 Κλήση συναρτήσεων σε βρόχο

Απενεργοποιούμε τη κλήση της συνάρτησης `disableServos()` με το να την μετατρέψουμε σε σχόλιο. (τοποθετούμε δύο καθέτους στα αριστερά της, όπως αυτό: `// disableServos`) Τοποθετούμε τις κλήσεις των συναρτήσεων `forward(2000)`, `turnLeft(600)`, `turnRight(600)` και `backward(2000)` στη συνάρτηση `loop`. Θα πρέπει να μοιάζει με αυτό όταν τελειώσει:

```

// Robotics with the BOE Shield - MovementsWithFunctionsInLoop
// Move forward, left, right, then backward for ever.

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
tone(4, 3000, 1000); // Play tone for 1 second
delay(1000); // Delay to finish tone

servoLeft.attach(13); // Attach left signal to pin 13
servoRight.attach(12); // Attach right signal to pin 12

//disableServos(); // Stay still indefinitely
}

void loop() // Main loop auto-repeats
{
forward(2000); // Go forward for 2 seconds
turnLeft(600); // Turn left for 0.6 seconds
turnRight(600); // Turn right for 0.6 seconds
backward(2000); // go backward for 2 seconds
}

void forward(int time) // Forward function
{
servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
servoRight.writeMicroseconds(1300); // Right wheel clockwise
delay(time); // Maneuver for time ms
}

```

```

void turnLeft(int time)                // Left turn function
{
  servoLeft.writeMicroseconds(1300);  // Left wheel clockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(time);                        // Maneuver for time ms
}

void turnRight(int time)               // Right turn function
{
  servoLeft.writeMicroseconds(1700);  // Left wheel counterclockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(time);                        // Maneuver for time ms
}

void backward(int time)               // Backward function
{
  servoLeft.writeMicroseconds(1300);  // Left wheel clockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(time);                        // Maneuver for time ms
}

```

2.10.7 Εξατομικευμένη συνάρτηση ελιγμών

Το τελευταίο πρόγραμμα, `MovementsWithSimpleFunctions`, ήταν κάπως μεγάλο και άχαρο. Και οι τέσσερις συναρτήσεις που χρησιμοποιούσε για να οδηγήσει το ρομπότ είναι σχεδόν ίδιες. Το πρόγραμμα `TestManeuverFunction` εκμεταλλεύεται αυτές τις ομοιότητες και παρέχει μια ενιαία συνάρτηση που ονομάζεται «`maneuver`», η οποία δέχεται τρεις παραμέτρους: `speedLeft`, `speedRight` και `msTime`.

```
void maneuver(int speedLeft, int speedRight, int msTime)
```

Με τη συνάρτηση «`maneuver`» δεν χρειάζεται να σκαφτόμαστε την δεξιόστροφη και αριστερόστροφη περιστροφή πια. Οι κανόνες για τη `speedLeft` και τη `speedRight` είναι:

- Θετικές τιμές για τη μετακίνηση του ρομπότ προς τα εμπρός
- Αρνητικές τιμές για τη μετακίνηση του ρομπότ προς τα πίσω
- 200 για πλήρης ταχύτητα προς τα εμπρός
- -200 για πλήρης ταχύτητα προς τα πίσω
- 0 για ακινητοποίηση
- Από 100 έως -100 για γραμμικό έλεγχο ταχύτητας

Οι κανόνες για την msTime είναι:

- Θετικές τιμές για τον αριθμό των ms που θα εκτελείται η maneuver
- -1 για την απενεργοποίηση του σήματος των servo

Παράδειγμα για το πώς η κλήση της συνάρτησης θα μοιάζει για την ακολουθία forward-backward-left-right-stop:

```
maneuver(200, 200, 2000);           // Forward 2 seconds
maneuver(-200, 200, 600);          // Left 0.6 seconds
maneuver(200, -200, 600);          // Right 0.6 seconds
maneuver(-200, -200, 2000);        // Backward 2 seconds
maneuver(0, 0, -1);                 // Disable servos
```

2.10.7.1 Παράδειγμα προγράμματος - TestManeuverFunction

```
// Robotics with the BOE Shield - TestManeuverFunction
// Move forward, left, right, then backward with maneuver function.

#include <Servo.h>                    // Include servo library

Servo servoLeft;                     // Declare left and right servos
Servo servoRight;

void setup()                          // Built-in initialization block
{
  tone(4, 3000, 1000);                // Play tone for 1 second
  delay(1000);                          // Delay to finish tone

  servoLeft.attach(13);                // Attach left signal to pin 13
  servoRight.attach(12);               // Attach right signal to pin 12

  maneuver(200, 200, 2000);            // Forward 2 seconds
  maneuver(-200, 200, 600);            // Left 0.6 seconds
  maneuver(200, -200, 600);            // Right 0.6 seconds
  maneuver(-200, -200, 2000);          // Backward 2 seconds
  maneuver(0, 0, -1);                  // Disable servos
}

void loop()                            // Main loop auto-repeats
{                                       // Empty, nothing needs repeating
}

void maneuver(int speedLeft, int speedRight, int msTime)
{
  //speedLeft, speedRight ranges: Backward  Linear  Stop  Linear  Forward
  //                               -200    -100.....0.....100    200
```

```

servoLeft.writeMicroseconds(1500 + speedLeft); // Set Left servo
speed
servoRight.writeMicroseconds(1500 - speedRight); // Set right servo
speed
if(msTime== -1) // if msTime = -1
{
servoLeft.detach(); // Stop servo signals
servoRight.detach();
}
delay(msTime); // Delay for msTime
}

```

2.10.8 Ακολουθίες ελιγμών με Πίνακες

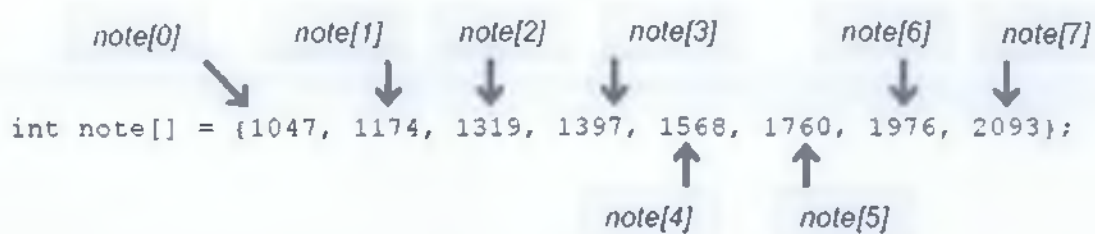
Ορισμένες ρομποτικές εφαρμογές απαιτούν μια ακολουθία από ελιγμούς (maneuvers). Όταν το BOE Shield - Bot εντοπίσει ότι έχει πέσει πάνω σ'ένα εμπόδιο, θα πρέπει πρώτα να πάει προς τα πίσω, στη συνέχεια να στρίψει και να προσπαθήσει να πάει πάλι μπροστά. Αυτό είναι μια απλή ακολουθία με δύο ελιγμούς .

Άλλες ακολουθίες είναι πιο περίτεχνες. Ένα παράδειγμα για μακριούς και σύνθετους ελιγμούς είναι ο προγραμματισμός για ένα ρομποτικό διαγωνισμό χορού. Για να μπορέσει να χορέψει για ένα ολόκληρο τραγούδι , το ρομπότ θα χρειαστεί μια αρκετά μεγάλη λίστα ελιγμών και χρόνων. Εάν στο πρόγραμμα μας θα πρέπει να αποθηκεύουμε λίστες ελιγμών, η μεταβλητή array (πίνακας)είναι το καλύτερο εργαλείο για την αποθήκευση αυτών των καταλόγων .

Ένας πίνακας είναι μια συλλογή από μεταβλητές με ένα κοινό όνομα. Κάθε μεταβλητή στον πίνακα αναφέρεται ως ένα στοιχείο. Εδώ είναι μια δήλωση πίνακα με οκτώ στοιχεία:

```
int note[] = {1047, 1174, 1319, 1397, 1568, 1760, 1976, 2093};
```

Το όνομα του πίνακα είναι το «note» και κάθε στοιχείο του πίνακα μπορεί να προσεγγιστεί από τον δείκτη. Η αρίθμηση των δεικτών του πίνακα ξεκινά πάντα από το 0 έως ότου όπου εμείς θέλουμε , στη συγκεκριμένη περίπτωση μέχρι το 7. Το παρακάτω διάγραμμα δείχνει πώς το note[0] αποθηκεύει το 1047, το note[1] αποθηκεύει το 1174, και ούτω καθεξής, μέχρι το note[7] το οποίο αποθηκεύει το 2093.



Ας πούμε ότι ο κώδικας πρέπει να αντιγράψει τη `note[3]`, η οποία αποθηκεύει την τιμή 1397, σε μια μεταβλητή που ονομάζεται `myVar`. Ο κώδικας θα μπορούσε να το κάνει έτσι:

```
myVar = note[3];
```

Στο πρόγραμμα μπορούμε να αλλάξουμε τις τιμές των στοιχείων του πίνακα. Για παράδειγμα, μπορούμε να αλλάξουμε την τιμή 1976 με την 1975 με την έκφραση:

```
note[3] = 1975;
```

Ένας πίνακας δεν χρειάζεται να προ-αρχικοποιηθεί με τιμές όπως παρουσιάζεται στην παραπάνω εικόνα. Για παράδειγμα, θα μπορούσαμε απλά να δηλώσουμε έναν πίνακα με 8 στοιχεία, όπως αυτό:

```
int myArray[8];
```

Στη συνέχεια, το πρόγραμμα θα μπορούσε να καλύψει τις τιμές του κάθε στοιχείου αργότερα, ίσως με μετρήσεις αισθητήρων, τιμές που εισάγονται από το `Serial Monitor`, ή με ότι αριθμούς θα έπρεπε να αποθηκεύσουμε.

Η παρακάτω εικόνα μας δείχνει τις μουσικές νότες στη δεξιά πλευρά του πληκτρολογίου-πιάνου. Κάθε πάτημα ενός πλήκτρου σε ένα βασικό πιάνο κάνει μια χορδή (ή ένα ηχείο αν είναι ηλεκτρικό) να δονείται σε μια συγκεκριμένη συχνότητα.

1046.5	1108.7	1174.7	1244.5	1318.5	1396.9	1480.0	1568.0	1661.2	1760.0	1864.7	1975.5	2093.0	2217.5	2349.3	2489.0	2637.0	2793.8	2960.0	3136.0	3322.4	3520.0	3729.3	3951.1	4186.0	
C6	D6	E6	F6	G6	A6	B6	C7	D7	E7	F7	G7	A7	B7	C8											
C 6 # D 6 # E 6 # F 6 # G 6 # A 6 # B 6 # C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	D 6 # E 6 # F 6 # G 6 # A 6 # B 6 # C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	E 6 # F 6 # G 6 # A 6 # B 6 # C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	F 6 # G 6 # A 6 # B 6 # C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	G 6 # A 6 # B 6 # C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	A 6 # B 6 # C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	B 6 # C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	C 7 # D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	D 7 # E 7 # F 7 # G 7 # A 7 # B 7 # C 8	E 7 # F 7 # G 7 # A 7 # B 7 # C 8	F 7 # G 7 # A 7 # B 7 # C 8	G 7 # A 7 # B 7 # C 8	A 7 # B 7 # C 8	B 7 # C 8	C 8											

2.10.8.1 Χρησιμοποιώντας στοιχεία του πίνακα

Ένα στοιχείο του πίνακα δεν πρέπει απαραίτητως να αντιγραφεί σε μια άλλη μεταβλητή για να χρησιμοποιηθεί η τιμή του. Για παράδειγμα, το πρόγραμμα θα μπορούσε απλά να εκτυπώνει την τιμή της `note[3]` στο Serial Monitor με τον παρακάτω τρόπο:

```
Serial.print(note[3]);
```

Δεδομένου ότι οι τιμές στον πίνακα είναι μουσικές νότες, θα μπορούσαμε επίσης να παίξουμε τη `note[3]` στο piezospeaker του BOE Shield-Bot. Κάπως έτσι:

```
tone(4, note[3], 500);
```

2.10.8.2 Παράδειγμα προγράμματος - PlayOneNote

Το παρακάτω παράδειγμα εκτυπώνει την τιμή ενός στοιχείου του πίνακα στον Serial Monitor, καθώς επίσης και χρησιμοποιεί αυτήν την τιμή για να κάνει το piezospeaker να παίξει μια μουσική νότα.

```

/ Robotics with the BOE Shield - PlayOneNote
// Displays and plays one element from note array.

int note[] = {1047, 1147, 1319, 1397, 1568, 1760, 1976, 2093};

void setup()
{
  Serial.begin(9600);

  Serial.print("note = ");
  Serial.println(note[3]);

  tone(4, note[3], 500);
  delay(750);
}

void loop()
{
}

```

2.10.8.3 Παράδειγμα προγράμματος - PlayNotesWithLoop

Πολλές εφαρμογές χρησιμοποιούν μεταβλητές για να έχουν πρόσβαση στα στοιχεία ενός πίνακα. Το επόμενο πρόγραμμα (PlayAnotherNote) δηλώνει μια μεταβλητή που ονομάζεται «index» και τη χρησιμοποιεί για να επιλέξει ένα στοιχείο του πίνακα.

Ο γνωστός for βρόχος μπορεί να προσαυξήσει αυτόματα την τιμή του index. Ο κώδικας για να παίξει και να εμφανίσει τις νότες (notes) είναι μέσα στο βρόχο for και ο index χρησιμοποιείται για να επιλέξει το στοιχείο του πίνακα. Για το πρώτο πέρασμα μέσα από το βρόχο, ο δείκτης θα είναι 0, οπότε η τιμή που είναι αποθηκευμένη στο note[0] θα χρησιμοποιείται κάθε φορά που το note[index] θα εμφανίζεται στην συνάρτηση print ή tone. Με κάθε πέρασμα μέσα από το βρόχο, ο index (δείκτης) αυξάνεται μέχρι το πρόγραμμα να εμφανίσει και να παίξει όλες τις τιμές (νοτες) του πίνακα.

```

// Robotics with the BOE Shield - PlayNotesWithLoop
// Displays and plays another element from note array.

int note[] = {1047, 1147, 1319, 1397, 1568, 1760, 1976, 2093};

void setup()
{
  Serial.begin(9600);

```

```

for(int index = 0; index < 8; index++)
{
  Serial.print("index = ");
  Serial.println(index);

  Serial.print("note[index] = ");
  Serial.println(note[index]);

  tone(4, note[index], 500);
  delay(750);
}

void loop()
{
}

```

Επίσης μπορούμε να το κάνουμε να παίζει με ανάποδη φορά τις νότες κάπως έτσι :

```

for(int index = 7; index >= 0; index--)

```

2.10.8.4 Χρησιμοποιώντας τη συνάρτηση sizeof

Ας πούμε ότι θέλουμε να συνθέσουμε ένα μουσικό κομμάτι που έχει περισσότερες ή λιγότερες νότες. Είναι εύκολο να ξεχάσουμε να ενημερώσουμε το βρόχο για να παίζει το σωστό αριθμό από νότες. Η βιβλιοθήκη Arduino έχει τη συνάρτηση `sizeof` που μπορεί να βοηθήσει με αυτό. Αυτή η συνάρτηση μπορεί να διαβάσει το μέγεθος του πίνακα σε bytes και το μέγεθος του είδους της μεταβλητής του πίνακα (όπως `int`). Ο κώδικας μπορεί στη συνέχεια να διαιρέσει τον αριθμό των bytes για το είδος της μεταβλητής στον αριθμό των bytes του πίνακα. Το αποτέλεσμα είναι ο αριθμός των στοιχείων του πίνακα.

Το παρακάτω κομμάτι κώδικα φορτώνει μια μεταβλητή που ονομάζεται `elementCount` με τον αριθμό των στοιχείων του πίνακα «note» :

```

int note[] = {1047, 1147, 1319, 1397, 1568, 1760, 1976, 2093};
int elementCount = sizeof(note) / sizeof(int);

```

Αργότερα, ο βρόχος «for» μπορεί να χρησιμοποιήσει τη μεταβλητή `elementCount` να παίζει όλες τις νότες του πίνακα, ακόμα και αν προσθέσουμε ή διαγράψουμε στοιχεία:

```

for(int index = 0; index < elementCount; index++)

// Robotics with the BOE Shield - PlayAllNotesInArray
// Uses sizeof to determine number of elements in the array
// and then displays and prints each note value in the sequence.

int note[] = {1047, 1147, 1319, 1397, 1568, 1760, 1976, 2093};

void setup()
{
  Serial.begin(9600);

  int elementCount = sizeof(note) / sizeof(int);

  Serial.print("Number of elements in array = ");
  Serial.println(elementCount);

  for(int index = 0; index < elementCount; index++)
  {
    Serial.print("index = ");
    Serial.println(index);

    Serial.print("note[index] = ");
    Serial.println(note[index]);

    tone(4, note[index], 500);
    delay(750);
  }
}

void loop()
{
}

```

2.10.8.5 Πλοήγηση με Πίνακες

Στο παρακάτω κομμάτι κώδικα αναγράφονται οι τρεις πίνακες για κάθε παράμετρο στη συνάρτηση «maneuver» (ελιγμών). Μαζί, κάνουν την ίδια ακολουθία προς τα εμπρός-αριστερά-δεξιά-προς τα πίσω-stop που έχουμε δει πιο πάνω.

```

//          Forward   left   right   backward   stop
//          index     0       1       2         3         4
int speedsLeft[] = {200,  -200,  200,   -200,    0};
int speedsRight[] = {200,  200,  -200,   -200,    0};
int times[] = {2000,  600,  600,   2000,   -1};

```

Ένα πρόγραμμα μπορεί στη συνέχεια να χρησιμοποιήσει αυτόν τον κώδικα σε μία από τις συναρτήσεις του για να εκτελέσει όλες τις κινήσεις:

```
// Determine number of elements in sequence list.
int elementCount = sizeof(times) / sizeof(int);

// Fetch successive elements from each sequence list and feed them
// to maneuver function.
for(int index = 0; index < elementCount; index++)
{
    maneuver(speedsLeft[index], speedsRight[index], times[index]);
}

```

Κάθε φορά μέσω του βρόχου, ο «index» (δείκτης) αυξάνεται κατά 1. Έτσι, με κάθε κλήση της «maneuver», το επόμενο στοιχείο σε κάθε πίνακα μεταβιβάζεται ως παράμετρος στη συνάρτηση «maneuver». Την πρώτη φορά μέσα στο βρόχο, ο «index» είναι 0, έτσι ώστε οι παράμετροι της συνάντησης γίνουν το μηδενικό στοιχείο από κάθε πίνακα, όπως: maneuver(speedsLeft[0], speedsRight[0], times[0]). Οι πραγματικές τιμές που παίρνανε στη συνάρτηση «maneuver» μοιάζει με: maneuver(200, 200, 2000). Τη δεύτερη φορά μέσα στο βρόχο, ο «index» είναι 1, οπότε η συνάρτηση μοιάζει με : maneuver(speedsLeft[1], speedsRight[1], times[1]), η οποία μεταφράζεται σε maneuver(-200, 200, 2000).

2.10.8.6 Παράδειγμα προγράμματος - ManeuverSequence

```
// Robotics with the BOE Shield - ManeuverSequence
// Move forward, left, right, then backward with an array and the
// maneuver function.

#include <Servo.h> // Include servo library

//          Forward   left   right   backward   stop
//          index     0     1     2         3         4
int speedsLeft[] = {200,  -200,  200,   -200,    0};
int speedsRight[] = {200,  200,  -200,   -200,    0};
int times[] = {2000,  600,  600,   2000,   -1};

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
    tone(4, 3000, 1000); // Play tone for 1 second
    delay(1000); // Delay to finish tone

    servoLeft.attach(13); // Attach left signal to pin 13
    servoRight.attach(12); // Attach right signal to pin 12

    // Determine number of elements in sequence list.

```

```

int elementCount = sizeof(times) / sizeof(int);

// Fetch successive elements from each sequence list and feed them
// to maneuver function.
for(int index = 0; index < elementCount; index++)
{
    maneuver(speedsLeft[index], speedsRight[index], times[index]);
}

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

void maneuver(int speedLeft, int speedRight, int msTime)
{
// speedLeft, speedRight ranges: Backward Linear Stop Linear Forward
//                               -200   -100.....0.....100   200
    servoLeft.writeMicroseconds(1500 + speedLeft); // Set Left servo
speed
    servoRight.writeMicroseconds(1500 - speedRight); // Set right servo
speed
    if(msTime===-1) // if msTime = -1
    {
        servoLeft.detach(); // Stop servo signals
        servoRight.detach();
    }
    delay(msTime); // Delay for msTime
}

```

2.10.8.7 Πινάκες χαρακτήρων και switch-case

Το παρακάτω παράδειγμα είναι ένα πρόγραμμα για την εκτέλεση ελιγμών (κινήσεων) χρησιμοποιώντας μια λίστα από χαρακτήρες σε έναν πίνακα. Κάθε χαρακτήρας αντιπροσωπεύει μια συγκεκριμένη κίνηση, με 200 ms χρόνο λειτουργίας. Δεδομένου ότι ο χρόνος λειτουργίας είναι σταθερός, δεν είναι τόσο ευέλικτος όσο η τελευταία προσέγγιση, αλλά σίγουρα είναι πολύ απλό να χτίσεις μια γρήγορη ακολουθία κινήσεων (ελιγμών).

f = forward b = backward l = left r = right s = stop

Οι πίνακες χαρακτήρων δεν χρησιμοποιούν λίστες στοιχείων διαχωρισμένες με κόμμα, αλλά μια συνεχή σειρά χαρακτήρων. Εδώ είναι ένα παράδειγμα της ίδιας αλληλουχία προς τα εμπρός-αριστερά-δεξιά-προς τα πίσω-stop σε μια σειρά χαρακτήρων:

```
char maneuvers[] = "ffffffffflllrrrrbbbbbbbbbs";
```

Η σειρά των χαρακτήρων του πίνακα έχει 10 f, αφού κάθε χαρακτήρας αντιπροσωπεύει 200 ms χρόνου λειτουργίας, το οποίο κάνει το BOE Shield-Bot να κινείται ευθεία για 2 δευτερόλεπτα. Στη συνέχεια, τρεις l χαρακτήρες κάνουν 600 ms αριστερή στροφή, τρεις χαρακτήρες r κάνουν παρόμοια δεξιά στροφή, που ακολουθείται από 10 β χαρακτήρες για να πάει προς τα πίσω 2 δευτερόλεπτα και ένα χαρακτήρα s για να σταματήσει και να ολοκληρωθεί η ακολουθία.

2.10.8.8 Παράδειγμα προγράμματος - ControlWithCharacters

```
// Robotics with the BOE Shield - ControlWithCharacters
// Move forward, left, right, then backward for testing and tuning.

#include <Servo.h> // Include servo library

char maneuvers[] = "ffffffffflllrrrrbbbbbbbbbs";

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to P13
  servoRight.attach(12); // Attach right signal to P12

  // Parse maneuvers and feed each successive character to the go
  function.
  int index = 0;
  do
  {
    go(maneuvers[index]);
  } while(maneuvers[index++] != 's');

void loop() // Main loop auto-repeats
{ // Empty, nothing needs repeating
}

void go(char c) // go function
{
  switch(c) // Switch to code based on c
  {
    case 'f': // c contains 'f'
      servoLeft.writeMicroseconds(1700); // Full speed forward
      servoRight.writeMicroseconds(1300);
      break;
```

```

case 'b': // c contains 'b'
  servoLeft.writeMicroseconds(1300); // Full speed backward
  servoRight.writeMicroseconds(1700);
  break;
case 'l': // c contains 'l'
  servoLeft.writeMicroseconds(1300); // Rotate left in place
  servoRight.writeMicroseconds(1300);
  break;
case 'r': // c contains 'r'
  servoLeft.writeMicroseconds(1700); // Rotate right in place
  servoRight.writeMicroseconds(1700);
  break;
case 's': // c contains 's'
  servoLeft.writeMicroseconds(1500); // Stop
  servoRight.writeMicroseconds(1500);
  break;
}
delay(200); // Execute for 0.2 seconds
}

```

Μετά από το πίνακα `char maneuvers` και τη συνηθισμένη αρχικοποίηση, αυτές οι γραμμές κώδικα φέρνουν τους χαρακτήρες από το πίνακα και τους περνά στη συνάρτηση `go` (εξηγείται στη συνέχεια).

```

int index = 0;
do
{
  go(maneuvers[index]);
} while(maneuvers[index++] != 's');

```

Πρώτον, ο `index` έχει δηλωθεί και αρχικοποιηθεί στο μηδέν, για να χρησιμοποιηθεί σε ένα `do - while` βρόχο. Παρόμοιος με ένα κανονικό `while` βρόχο, ο `do - while` εκτελεί επανειλημμένα εντολές μέσα στο μπλοκ του κώδικα του, ενώ μια συνθήκη είναι αληθής, αλλά το `while` μέρος έρχεται μετά από το μπλοκ έτσι ώστε να εκτελεστεί τουλάχιστον μία φορά. Κάθε φορά μέσα στο βρόχο, το `go(maneuvers[index])` περνάει το χαρακτήρα του `maneuvers[index]` στη «`go`» συνάρτηση. Το `++` στο `index++` προσθέτει 1 στη μεταβλητή του `index` για την επόμενη φορά μέσα στο βρόχο (αυτό είναι το «`post increment operator`»). Αυτή η λούπα συνεχίζεται μέχρις ότου `while(maneuvers[index] != 's')`, που σημαίνει «μέχρις ότου η τιμή που εισπράττει από τον πίνακα `maneuvers` δεν είναι ίση με το 's'».

Η συνάρτηση «`go`» λαμβάνει κάθε χαρακτήρα που περνά από την `c` παράμετρο της, και τον αξιολογεί με μια `case-by-case` (περίπτωση - κατά - περίπτωση) βάση,

χρησιμοποιώντας μια `switch / case` δήλωση. Για καθένα από τα πέντε γράμματα στο πίνακα χαρακτήρων «`maneuvers`», υπάρχει μια αντίστοιχη `case` δήλωση στο μπλοκ του `switch(c)` που θα εκτελεστεί αν ο χαρακτήρας ληφθεί από τη `go`.

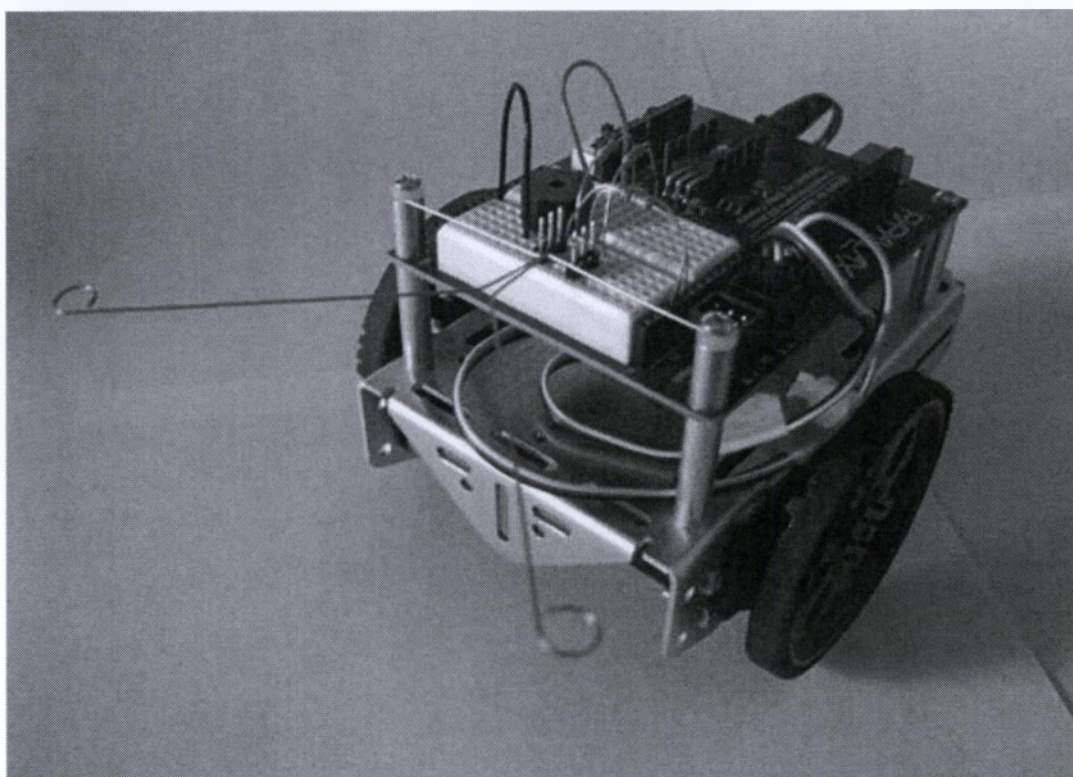
Αν η κλήση της συνάρτησης «`go`» περάσει το χαρακτήρα `f` στην παράμετρο `c`, εκτελείται ο κώδικας στην `case f` (περίπτωση `f`, πλήρης ταχύτητα εμπρός). Αν περάσει `b`, εκτελείται ο κώδικας «πλήρης ταχύτητα προ τα πίσω». Η «`break`» σε κάθε περίπτωση (`case`) αποδεσμεύει το «`switch`» μπλοκ και το πρόγραμμα κινείται στην επόμενη εντολή, η οποία είναι η `delay(200)`. Έτσι, κάθε κλήση της «`go`» έχει ως αποτελέσματα μια κίνηση (έναν ελιγμο) `200 ms`. Χωρίς τη «`break`» στο τέλος κάθε περίπτωσης, το πρόγραμμα θα συνέχιζε στο κώδικα για την επόμενη περίπτωση, με αποτέλεσμα να κινηθεί με τρόπο που δεν είχε ζητηθεί.

```
void go(char c) // go function
{
  switch(c) // Switch to based on c
  {
    case 'f': // c contains 'f'
      servoLeft.writeMicroseconds(1700); // Full speed forward
      servoRight.writeMicroseconds(1300);
      break;
    case 'b': // c contains 'b'
      servoLeft.writeMicroseconds(1300); // Full speed backward
      servoRight.writeMicroseconds(1700);
      break;
    case 'l': // c contains 'l'
      servoLeft.writeMicroseconds(1300); // Rotate left in place
      servoRight.writeMicroseconds(1300);
      break;
    case 'r': // c contains 'r'
      servoLeft.writeMicroseconds(1700); // Rotate right in place
      servoRight.writeMicroseconds(1700);
      break;
    case 's': // c contains 's'
      servoLeft.writeMicroseconds(1500); // Stop
      servoRight.writeMicroseconds(1500);
      break;
  }
  delay(200); // Execute for 0.2 s
}
```

2.11 Απτική Πλοήγηση με Whiskers

Οι «tactile switches» (συνδετήρας ηλεκτρικών συρμάτων αφής) που επίσης ονομάζονται και «bumper switches» ή «touch switches», έχουν πολλές χρήσεις στη ρομποτική. Ένα ρομπότ, προγραμματισμένο να παίρνει ένα αντικείμενο και να το μετακινεί σε ένα μεταφορικό ιμάντα, μπορεί να βασίζεται σε ένα «tactile switch» (απτικό συνδετήρα) για να εντοπίσει το αντικείμενο. Αυτοματοποιημένες γραμμές εργοστάσιων μπορεί να χρησιμοποιούν τα «tactile switches» για να μετράνε τα αντικείμενα και να ευθυγραμμίζουν τα μέρη (αντικείμενα) για ένα συγκεκριμένο βήμα στη διαδικασία κατασκευής. Σε κάθε περίπτωση, οι απτικοί συνδετήρες παρέχουν εισροές που προκαλούν κάποια μορφή προγραμματισμένης εξόδου. Οι εισροές παρακολουθούνται ηλεκτρονικά από τον επεξεργαστή του εξοπλισμού, ο οποίος λαμβάνει διαφορετικές ενέργειες ανάλογα με το αν ο συνδετήρας είναι πατημένος ή όχι.

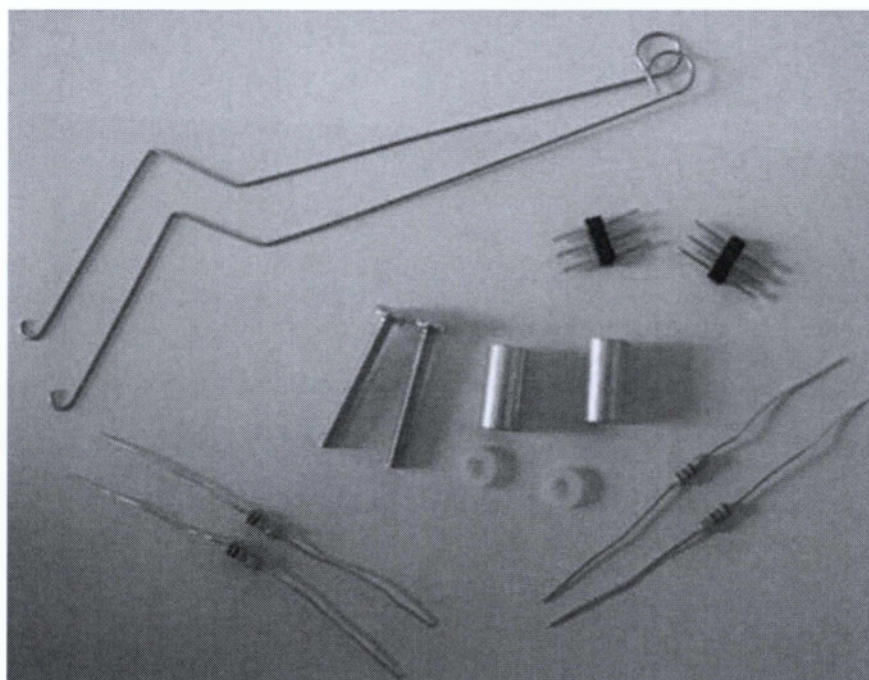
Στη συγκεκριμένη περίπτωση χρησιμοποιούμε απτικούς συνδετήρες που ονομάζονται «Whiskers», οι οποίοι δίνουν στο BOE Shield-Bot τη δυνατότητα να ανιχνεύει το περιβάλλον του μέσω της αφής, καθώς περιπλανιέται.



2.11.1 Κατασκευή και έλεγχος των Whiskers

Κατάλογος εξαρτημάτων :

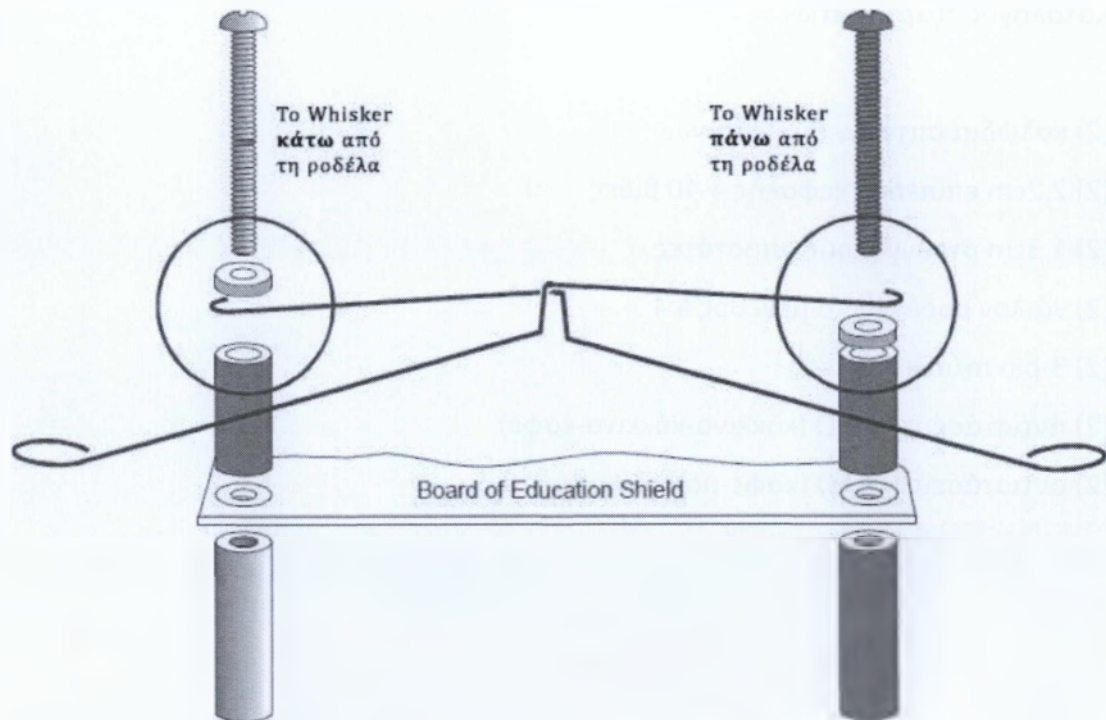
- (2) καλώδια απτικών συνδετήρων
- (2) 2,2cm επίπεδης κεφαλής 4-40 βίδες
- (2) 1,3cm στρογγυλούς αποστάτες
- (2) νάιλον ροδέλες, το μέγεθος # 4
- (2) 3-pin m/m κεφάλες
- (2) αντιστάσεις, 220 Ω (κόκκινο-κόκκινο-καφέ)
- (2) αντιστάσεις, 10 kΩ (καφέ-μαύρο-πορτοκαλί)



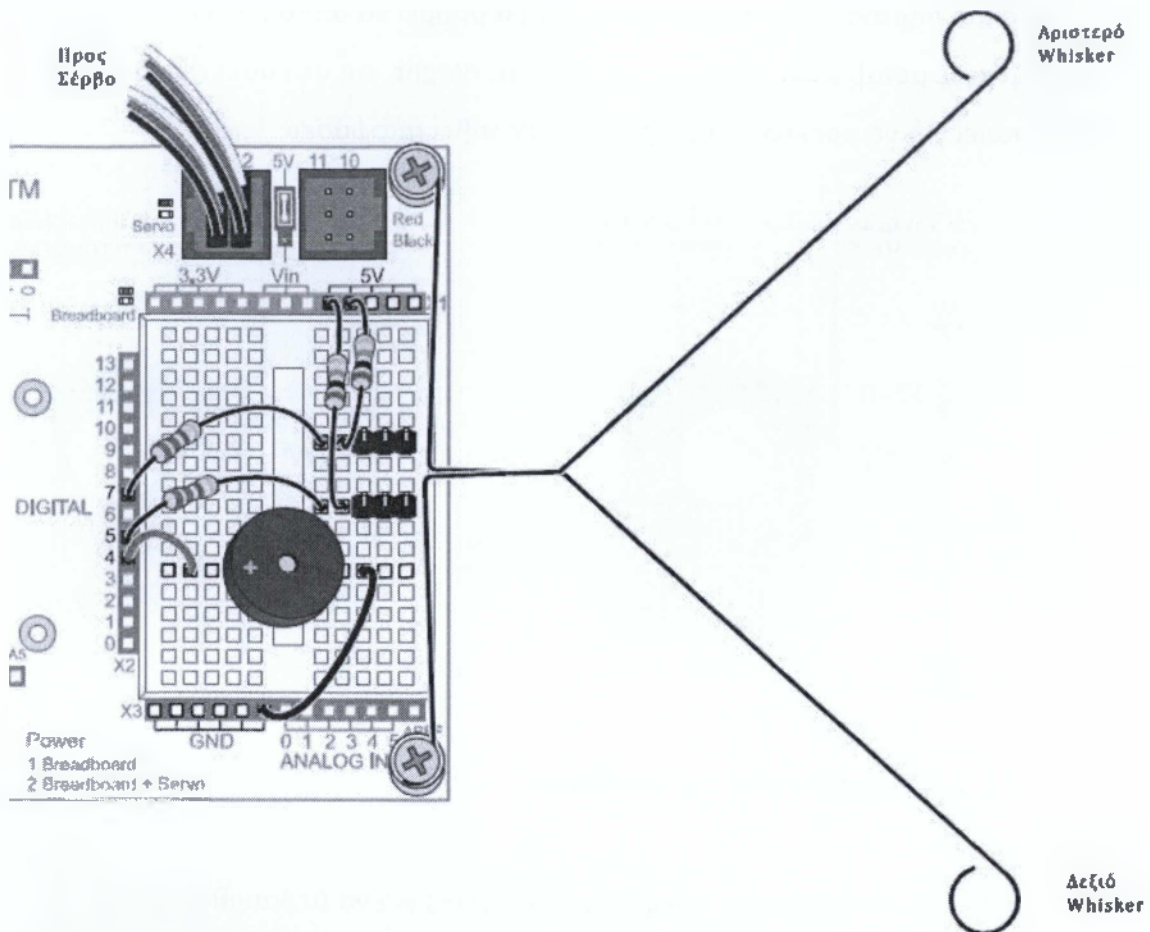
Αφαιρούμε τα LED κυκλώματα που χρησιμοποιούσαμε ως οθόνες σήματος για τον έλεγχο της πλοήγησης των servo, και τις δύο μπροστινές βίδες που συγκρατούν την πλακέτα μας με τα μπροστά διαχωριστικά. Έπειτα περνάμε μια νάιλον ροδέλα και ένα στρογγυλό αποστάτη σε κάθε μια από τις 2,2cm βίδες.

Στερεώνουμε τις βίδες μέσα από τις τρύπες τις πλακέτας (και των μεταλλικών διαχωριστικών) χωρίς να τις σφίξουμε τελείως και τοποθετούμε τα άκρα των καλωδίων (απτικών συνδετήρων) γύρω από τις βίδες, το ένα πάνω από τη μια ροδέλα και το άλλο

κάτω από την άλλη ροδέλα, τοποθετώντας τα έτσι ώστε να διασταυρώνονται χωρίς να ακουμπάνε μεταξύ τους, και σφίγγουμε τις βίδες στα διαχωριστικά.



Στη συνέχεια χρησιμοποιούμε τις 220 Ω αντιστάσεις (κόκκινο-κόκκινο-καφέ) για να συνδέσουμε τα ψηφιακά pin 5 και 7 με τις αντίστοιχες 3-pin κεφαλές. Χρησιμοποιούμε τις 10 kΩ αντιστάσεις (καφέ-μαύρο-πορτοκαλί) για να συνδέσουμε 5 V σε κάθε κεφαλή 3-pin και τέλος φροντίζουμε να ρυθμίσουμε κάθε whisker, έτσι ώστε να είναι κοντά αλλά να μην αγγίζει τις 3-pin κεφαλές πάνω στο breadboard (ράστερ). Μια απόσταση περίπου 3 mm είναι ότι πρέπει.

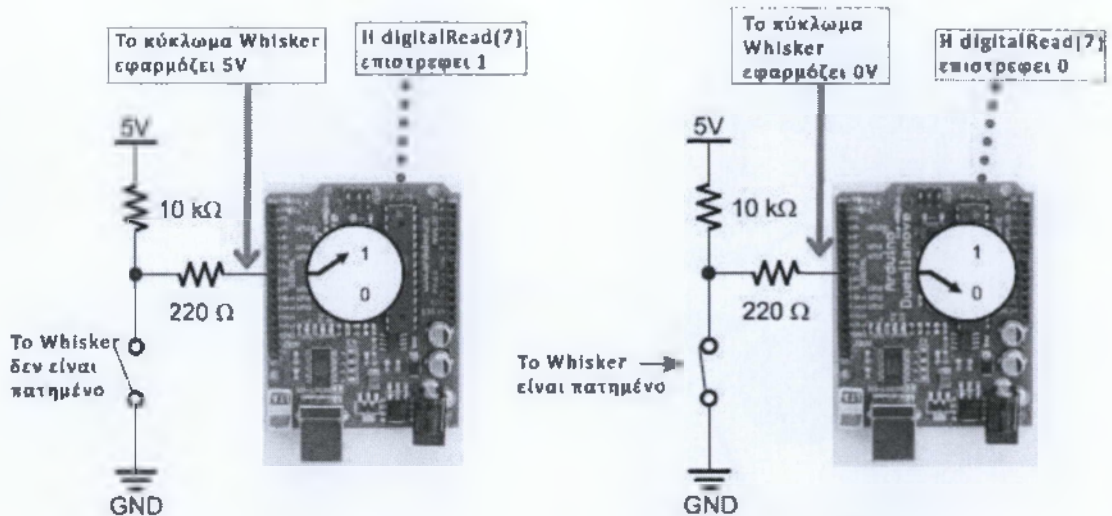


2.11.1.1 Πως λειτουργούν οι διακόπτες των Whiskers

Τα whiskers είναι συνδεδεμένα στη γείωση (V_{ss}), επειδή οι τρύπες στις εξωτερικές άκρες τις πλακέτας είναι συνδεδεμένες με V_{ss} . Τα μεταλλικά διαχωριστικά και οι βίδες παρέχουν την ηλεκτρική σύνδεση σε κάθε whisker. Δεδομένου ότι κάθε whisker συνδέεται με ψηφιακό I/O, το Arduino μπορεί να προγραμματιστεί για να διακρίνει ποια τάση εφαρμόζεται σε κάθε κύκλωμα, 5 V ή 0 V. Για να ξεκινήσουμε, ορίζουμε κάθε pin (ακίδα) σε λειτουργία εισόδου (input) με την εντολή `pinMode (pin, mode)` και στη συνέχεια ανιχνεύουμε την κατάσταση του pin (HIGH ή LOW) με τη συνάρτηση `digitalRead (pin)`.

Στα αριστερά της παρακάτω εικόνας, το κύκλωμα εφαρμόζει 5 V όταν το whisker δεν έχει πιεστεί, έτσι η `digitalRead (7)` επιστρέφει 1 (HIGH). Στη δεξιά πλευρά, το κύκλωμα εφαρμόζει 0 V όταν το whisker πιεστεί, έτσι η `digitalRead (7)` επιστρέφει 0 (LOW).

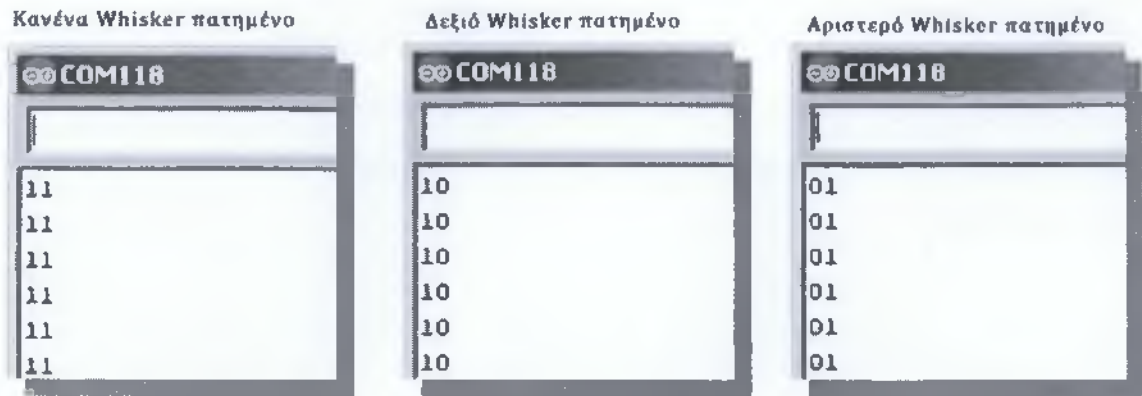
Το πιο σημαντικό είναι ότι το πρόγραμμα μπορεί να αποθηκεύει τις τιμές επιστροφής σε μεταβλητές, όπως τη `wLeft` και τη `wRight`, και στη συνέχεια να τις χρησιμοποιεί για να προκαλέσει ενέργειες ή να λάβει αποφάσεις.



2.11.1.2 Ελέγχοντας τα Whiskers

Το επόμενο πρόγραμμα δοκιμάζει τα whiskers για να βεβαιωθούμε ότι λειτουργούν σωστά, προβάλλοντας τις δυαδικές τιμές που επιστρέφει η `digitalRead(7)` και η `digitalRead(5)`. Με αυτό τον τρόπο, μπορούμε να πατήσουμε κάθε whisker μέχρι να ακουμπήσει με τη κεφαλή 3-pin (που έχουμε τοποθετήσει πάνω στο breadboard), και να δούμε αν η ψηφιακή pin του Arduino στέλνει την ηλεκτρική επαφή.

Όταν κανένα από τα whiskers δεν πιέζονται ενάντια στο 3-pin header, πρέπει στο Serial Monitor να εμφανίζονται δύο στήλες από 1, ένα για κάθε whisker. Αν πατήσουμε το δεξιό whisker, η δεξιά στήλη θα πρέπει να αναφέρει 0, και η οθόνη θα πρέπει να εμφανίζει 10. Εάν πατήσουμε το αριστερό whisker, η αριστερή στήλη θα πρέπει να αναφέρει 1 και η οθόνη θα πρέπει να εμφανίζει 01. Φυσικά, εάν πατήσουμε και τα δύο whiskers, θα πρέπει να εμφανίζει 00.



2.11.1.3 Παραδειγμα προγραμματος - DisplayWhiskerStates

```

/*
 * Robotics with the BOE Shield - DisplayWhiskerStates
 * Display left and right whisker states in Serial Monitor.
 * 1 indicates no contact; 0 indicates contact.
 */

void setup()                // Built-in initialization block
{
  tone(4, 3000, 1000);      // Play tone for 1 second
  delay(1000);              // Delay to finish tone
  pinMode(7, INPUT);        // Set right whisker pin to input
  pinMode(5, INPUT);        // Set left whisker pin to input

  Serial.begin(9600);       // Set data rate to 9600 bps
}

void loop()                 // Main loop auto-repeats
{
  byte wLeft = digitalRead(5); // Copy left result to wLeft
  byte wRight = digitalRead(7); // Copy right result to wRight

  Serial.print(wLeft);       // Display left whisker state
  Serial.println(wRight);    // Display right whisker state

  delay(50);                // Pause for 50 ms
}

```

2.11.1.4 Πως λειτουργεί το DisplayWhiskerStates

Στη συνάρτηση `setup`, η `pinMode (7, INPUT)` και η `pinMode (5, INPUT)` ορίζουν τα ψηφιακά pin 7 και 5 ως εισόδους, ώστε να μπορούν να παρακολουθούν τις τάσεις που εφαρμόζονται από τα κυκλώματα των whiskers.

```
pinMode(7, INPUT);          // Set right whisker pin to input
```

```
pinMode(5, INPUT);          // Set left whisker pin to input
```

Στη συνάρτηση loop, κάθε κλήση της digitalRead επιστρέφει 0 αν το whisker πατηθεί ή 1 αν δεν πατηθεί. Οι τιμές αυτές αντιγράφονται σε μεταβλητές που ονομάζονται wLeft και wRight, τα οποία είναι συντομογραφία για whisker-left και whisker-right.

```
byte wLeft = digitalRead(5);    // Copy left result to wLeft
byte wRight = digitalRead(7);   // Copy right result to wRight
```

Στη συνέχεια, η Serial.print εμφανίζει την τιμή της wLeft στο Serial Monitor και η Serial.println εμφανίζει την τιμή του wRight και μια μεταφορά μετ'επιστροφής.

```
Serial.print(wLeft);           // Display left whisker state
Serial.println(wRight);       // Display right whisker state
```

Πριν από την επόμενη επανάληψη του βρόχου, υπάρχει μια delay(50). Αυτό επιβραδύνει τον αριθμό των μηνυμάτων που το Serial Monitor λαμβάνει κάθε δευτερόλεπτο. Αν και είναι πιθανόν να μην χρειάζεται, το αφήνουμε για να αποφευχθούν πιθανές υπερβάσεις του buffer του υπολογιστή (πάρα πολλά στοιχεία για να αποθηκευτούν) (computer buffer overruns) για παλιότερο hardware και ορισμένα λειτουργικά συστήματα.

2.11.1.5 Εμφωλευμένες κλήσεις συναρτήσεων

Το πρόγραμμα στην πραγματικότητα δεν χρειάζεται να χρησιμοποιεί μεταβλητές για να αποθηκεύει τις τιμές από τη digitalRead. Αντί αυτού, οι τιμές (1 ή 0) που επιστρέφει η digitalRead μπορούν να χρησιμοποιηθούν άμεσα με την εκχώρηση της κλήσης της συνάρτησης μέσα στη Serial.print και στέλνοντας τις τιμές επιστροφής της κατ' ευθείαν στο Serial Monitor. Σε αυτή την περίπτωση, η συνάρτηση loop θα μοιάζει κάπως έτσι:

```
void loop()                  // Main loop auto-repeats
{
  Serial.print(digitalRead(5)); // Display wLeft
  Serial.println(digitalRead(7)); // Display wRight
```



```

delay(50); // Pause for 50 ms
}

```

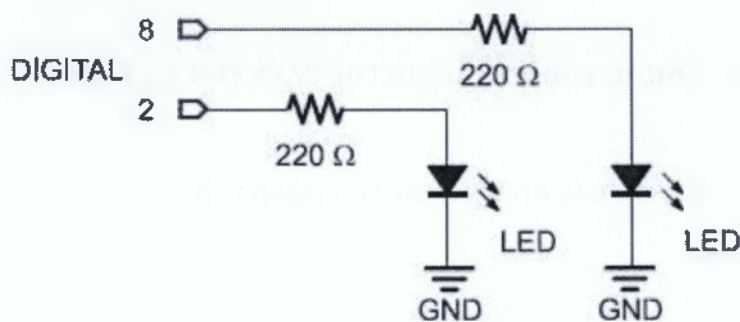
2.11.2 Επιτόπιος έλεγχος των Whiskers

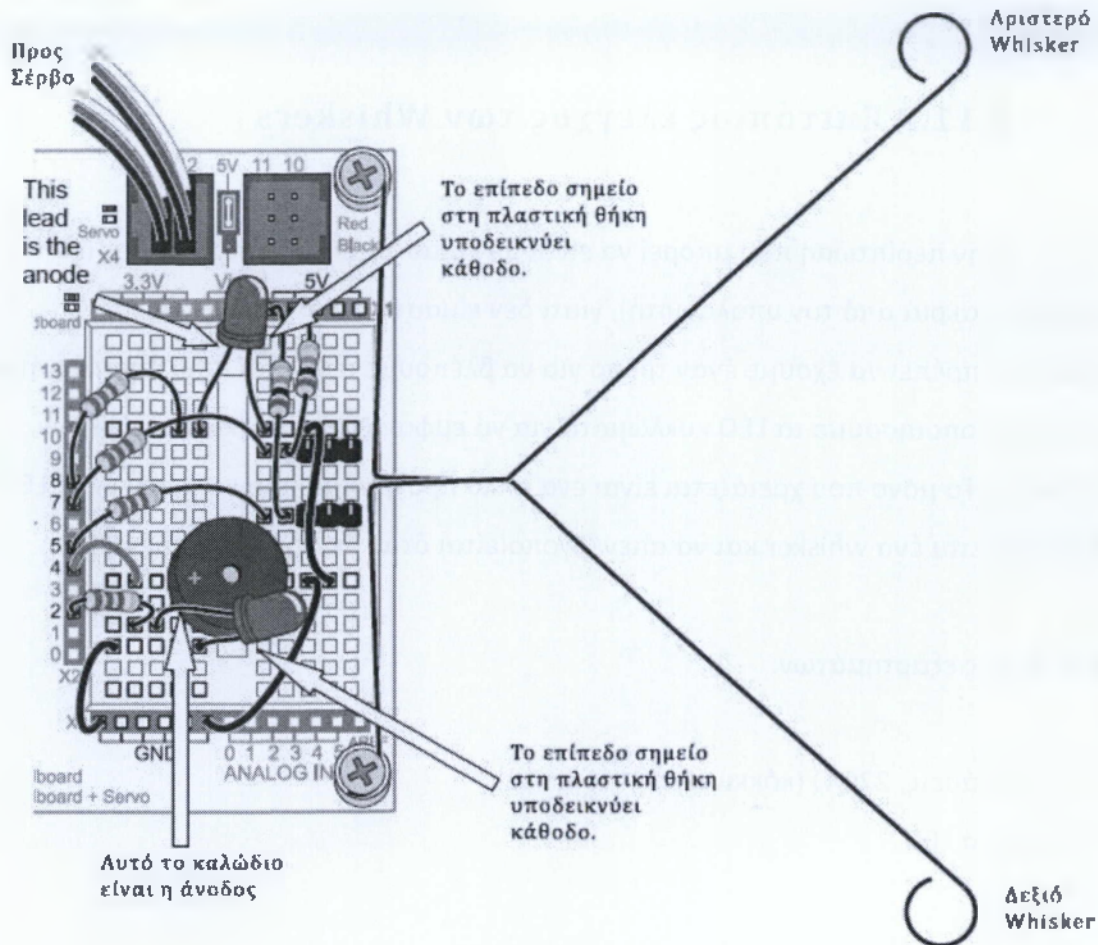
Στην περίπτωση που μπορεί να είναι μην διαθέσιμο το Serial Monitor (να είμαστε μακριά από τον υπολογιστή), γιατί δεν είμαστε συνδεδεμένοι με το usb καλώδιο, πρέπει να έχουμε έναν τρόπο για να βλέπουμε τι γίνεται. Μια λύση θα ήταν να χρησιμοποιήσουμε τα LED κυκλώματα για να εμφανίζουμε την κατάσταση των whiskers. Το μόνο που χρειάζεται είναι ένα απλό πρόγραμμα που να ανάβει ένα LED όταν πιέζεται ένα whisker και να απενεργοποιείται όταν δεν είναι πατημένο.

Κατάλογος εξαρτημάτων:

(2) αντιστάσεις, 220 Ω (κόκκινο-κόκκινο-καφέ)

(2) κόκκινα LED





2.11.2.1 Προγραμματισμός του κυκλώματος ελέγχου με LED

Προσθέτουμε κλήσεις της pinMode στη συνάρτηση «setup», θέτοντας τα ψηφιακά pin 8 και 2 σε «output».

```
pinMode(8, OUTPUT); // Left LED indicator -> output
pinMode(2, OUTPUT); // Right LED indicator -> output
```

Για να κάνουμε τις δηλώσεις εισόδου (input states) των whisker να ελέγχουν τα LED, τοποθετούμε δύο if ... else δηλώσεις μεταξύ των εντολών Serial.println (Wright) και delay(50):

```
if(wLeft == 0) // If left whisker contact
{
    digitalWrite(8, HIGH); // Left LED on
}
else // If no left whisker contact
{
    digitalWrite(8, LOW); // Left LED off
```

```

}

if(wRight == 0)           // If right whisker contact
{
  digitalWrite(2, HIGH); // Right LED on
}
else                     // If no right whisker contact
{
  digitalWrite(2, LOW);  // Right LED off
}

```

Οι if ... else δηλώνσεις εκτελούν τμήματα του κώδικα με βάση κάποιες συνθήκες.

Στη συγκεκριμένη περίπτωση, αν η wLeft αποθηκεύει 0, εκτελείτε η κλήση της digitalWrite (8, HIGH). Αν η wLeft αποθηκεύει 1, εκτελείτε η κλήση της digitalWrite (8, LOW). Με αποτέλεσμα το αριστερό LED να ανάβει όταν το αριστερό whisker πιέζεται ή να απενεργοποιείται όταν δεν είναι πατημένο. Το δεύτερο if ... else κάνει την ίδια δουλειά με τη wRight και το δεξιό κύκλωμα LED.

```

/*
 * Robotics with the BOE Shield - TestWhiskersWithLeds
 * Display left and right whisker states in Serial Monitor.
 * 1 indicates no contact; 0 indicates contact.
 * Display whisker states with LEDs. LED on indicates contact;
 * off indicates none.
 */

void setup()                // Built-in initialization block
{
  pinMode(7, INPUT);        // Set right whisker pin to input
  pinMode(5, INPUT);        // Set left whisker pin to input
  pinMode(8, OUTPUT);       // Left LED indicator -> output
  pinMode(2, OUTPUT);       // Right LED indicator -> output

  tone(4, 3000, 1000);      // Play tone for 1 second
  delay(1000);              // Delay to finish tone

  Serial.begin(9600);       // Set serial data rate to 9600
}

void loop()                 // Main loop auto-repeats
{
  byte wLeft = digitalRead(5); // Copy left result to wLeft
  byte wRight = digitalRead(7); // Copy right result to wRight

  if(wLeft == 0)           // If left whisker contact
  {
    digitalWrite(8, HIGH); // Left LED on
  }
  else                     // If no left whisker contact
  {
    digitalWrite(8, LOW);  // Left LED off
  }

  if(wRight == 0)         // If right whisker contact

```

```

    (
      digitalWrite(2, HIGH);          // Right LED on
    )
  else                                // If no right whisker contact
  (
    digitalWrite(2, LOW);           // Right LED off
  )

  Serial.print(wLeft);               // Display wLeft
  Serial.println(wRight);            // Display wRight
  delay(50);                          // Pause for 50 ms
)

```

2.11.3 Πλοήγηση με τα Whiskers

Μέχρι στιγμής τα προγράμματα μας κάνουν το BOE Shield-Bot να εκτελεί μια λίστα κινήσεων προκαθορισμένες από εμάς (τον προγραμματιστή). Τώρα που μπορούμε να γράψουμε ένα πρόγραμμα που ο Arduino ελέγχει τους διακόπτες των whisker και προκαλεί μια ενέργεια ως απάντηση, μπορούμε επίσης να γράψουμε ένα πρόγραμμα που επιτρέπει στο BOE Shield-Bot να κινείται και να επιλέγει τους δικούς του ελιγμούς, αν πέσει πάνω σε κάτι. (αυτόνομη ρομποτική πλοήγηση)

Το `RoamingWithWhiskers` κάνει το BOE Shield-Bot να προχωράει μπροστά, ενώ παρακολουθεί τις εισροές (input) των whisker, μέχρι να συναντήσει κάποιο. Από τη στιγμή που ο Arduino αισθανθεί την ηλεκτρική επαφή των whisker, χρησιμοποιεί μια `if ... else if ... else` δήλωση για να αποφασίσει τι θα κάνει. Ο κώδικας αποφάσεων ελέγχει για διάφορους συνδυασμούς των whisker που πιέζονται / δεν πιέζονται, και καλεί συναρτήσεις πλοήγησης για να εκτελέσει πίσω-εμπρός-τέλος-στροφή ελιγμούς. Έπειτα, το BOE Shield-Bot συνεχίζει την κίνηση προς τα εμπρός μέχρι να πέσει πάνω σε κάποιο άλλο εμπόδιο.

2.11.3.1 Παράδειγμα προγράμματος - `RoamingWithWhiskers`

```

// Robotics with the BOE Shield - RoamingWithWhiskers
// Go forward. Back up and turn if whiskers indicate BOE Shield bot
// bumped
// into something.

#include <Servo.h>                                // Include servo library

Servo servoLeft;                                // Declare left and right servos

```

```

Servo servoRight;

void setup() // Built-in initialization block
{
  pinMode(7, INPUT); // Set right whisker pin to input
  pinMode(5, INPUT); // Set left whisker pin to input

  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12
}

void loop() // Main loop auto-repeats
{
  byte wLeft = digitalRead(5); // Copy left result to wLeft
  byte wRight = digitalRead(7); // Copy right result to wRight

  if((wLeft == 0) && (wRight == 0)) // If both whiskers contact
  {
    backward(1000); // Back up 1 second
    turnLeft(800); // Turn left about 120 degrees
  }
  else if(wLeft == 0) // If only left whisker contact
  {
    backward(1000); // Back up 1 second
    turnRight(400); // Turn right about 60 degrees
  }
  else if(wRight == 0) // If only right whisker contact
  {
    backward(1000); // Back up 1 second
    turnLeft(400); // Turn left about 60 degrees
  }
  else // Otherwise, no whisker contact
  {
    forward(20); // Forward 1/50 of a second
  }
}

void forward(int time) // Forward function
{
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(time); // Maneuver for time ms
}

void turnLeft(int time) // Left turn function
{
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(time); // Maneuver for time ms
}

void turnRight(int time) // Right turn function
{
  servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(time); // Maneuver for time ms
}

```

```

void backward(int time) // Backward function
{
  servoLeft.writeMicroseconds(1300); // Left wheel clockwise
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(time); // Maneuver for time ms
}

```

2.11.3.2 Πως λειτουργεί το RoamingWithWhiskers

Η if ... else if ... else δήλωση στη συνάρτηση loop (βρόχου) ελέγχει τα whiskers για τυχόν καταστάσεις που απαιτούν προσοχή. Η δήλωση ξεκινά με if ((wLeft == 0) && (wRight == 0)), το οποίο μεταφράζεται σε «αν η μεταβλητή wLeft και η μεταβλητή wRight είναι ίσες με μηδέν». Αν και οι δύο μεταβλητές είναι μηδέν, εκτελούνται οι δύο κλήσεις στο if μπλοκ του κώδικα: backward(1000) και turnLeft(800).

```

if((wLeft == 0) && (wRight == 0)) // If both whiskers contact
{
  backward(1000); // Back up 1 second
  turnLeft(800); // Turn left about 120 degrees
}

```

Στην if ... else if ... else δήλωση, το πρόγραμμα παραλείπει μπλοκ του κώδικα με προϋποθέσεις που δεν είναι αληθείς, και συνεχίζει να ελέγχει μέχρι να βρει είτε μια κατάσταση που είναι αληθείς ή να τελειώσουν οι προϋποθέσεις. Όταν το πρόγραμμα βρίσκει μια αληθείς δήλωση, εκτελεί ό, τι είναι στο μπλοκ του κώδικα του, και έπειτα μεταβαίνει στο τέλος της if ... else if ... else χωρίς να ελέγξει περαιτέρω προϋποθέσεις.

Έτσι, αν και οι δύο whiskers δεν πιέζονται, η πρώτη if δήλωση δεν είναι αληθείς και μπλοκ του κώδικα της παραλείπεται. Το σκίτσο θα ελέγξει την πρώτη else if δήλωση, και αν είναι το αριστερό whisker που πιέζεται, θα εκτελεστούν οι κλήσεις στο μπλοκ του κώδικα αυτής της δήλωσης. Μετά από τη οπισθοχώρηση για ένα δευτερόλεπτο (backward(1000)) και στρίβοντας δεξιά για 0,4 δευτερόλεπτα (turnRight(400)), το πρόγραμμα προσπερνάει το υπόλοιπο των συνθηκών και προχωρά σε ό, τι έρχεται μετά από την τελευταία else δήλωση.

```

else if(wLeft == 0) // If only left whisker contact
{
  backward(1000); // Back up 1 second
  turnRight(400); // Turn right about 60 degrees
}

```

Αν είναι το δεξιό whisker που ανιχνεύει το εμπόδιο, τα δύο πρώτα μπλοκ του κώδικα θα πρέπει να πασαλειφθούν και να εκτελεστεί η else if (Wright == 0) δήλωση.

```
else if(wRight == 0)           // If only right whisker contact
{
    backward(1000);           // Back up 1 second
    turnLeft(400);           // Turn left about 60 degrees
}
```

Μια else συνάρτηση προϋπόθεσης λειτουργεί ως ένα catch-all για την περίπτωση που καμία από τις δηλώσεις που προηγήθηκαν ήταν αληθείς. Δεν είναι απαραίτητο, αλλά στην προκειμένη περίπτωση, είναι χρήσιμο όταν δεν πατηθεί κανένα whisker. Αυτή η δήλωση επιτρέπει στο BOE Shield-Bot να κυλήσει προς τα εμπρός για 20 ms πριν από την επανάληψη του βρόχου. Το μικρό αυτό χρονικό διάστημα προς τα εμπρός, πριν από τον επανέλεγχο, επιτρέπει στο BOE Shield-Bot να ανταποκρίνεται γρήγορα στις αλλαγές των αισθητήρων των whiskers καθώς κυλά προς τα εμπρός.

```
else                           // Otherwise, no whisker contact
{
    forward(20);               // Forward 1/50 of a second
}
```

Επίσης μπορούμε να τροποποιήσουμε τις if ... else if ... else δηλώσεις, έτσι ώστε οι LED δείκτες να μεταδίδουν τους οποίους ελιγμούς κάνει το BOE Shield-Bot. Απλά προσθέτουμε digitalWrite κλήσεις που στέλνουν HIGH και LOW σήματα στα κυκλώματα ένδειξης LED, όπως στο παρακάτω παράδειγμα:

```
if((wLeft == 0) && (wRight == 0)) // If both whiskers contact
{
    digitalWrite(8, HIGH);       // Left LED on
    digitalWrite(2, HIGH);      // Right LED on
    backward(1000);             // Back up 1 second
    turnLeft(800);              // Turn left about 120 degrees
}
else if(wLeft == 0)            // If only left whisker contact
{
    digitalWrite(8, HIGH);       // Left LED on
    digitalWrite(2, LOW);        // Right LED off
    backward(1000);             // Back up 1 second
    turnRight(400);             // Turn right about 60 degrees
}
else if(wRight == 0)          // If only right whisker contact
```

```

    digitalWrite(8, LOW);           // Left LED off
    digitalWrite(2, HIGH);         // Right LED on
    backward(1000);                 // Back up 1 second
    turnLeft(400);                 // Turn left about 60 degrees
}
else                                // Otherwise, no whisker contact
{
    digitalWrite(8, LOW);         // Left LED off
    digitalWrite(2, LOW);         // Right LED off
    forward(20);                  // Forward 1/50 of a second
}

```

Θα πρέπει να ρυθμίσουμε και τα ψηφιακά pin σαν εξόδους(OUTPUT) στη συνάρτηση setup, έτσι ώστε να μπορέσει να παρέχει ρεύμα στα LEDS:

```

pinMode(8, OUTPUT);               // Left LED indicator -> output
pinMode(2, OUTPUT);               // Right LED indicator -> output

```

2.11.4 Τεχνητή Νοημοσύνη για αποφυγή γωνιών

Παρατηρήσουμε ότι με το τελευταίο πρόγραμμα, το BOE Shield-Bot τείνει να κολλάει σε γωνίες. Καθώς εισέρχεται σε μια γωνία, το αριστερό whisker εφάπτεται στον αριστερό τοίχο, επομένως κινείται προς τα πίσω και στρίβει δεξιά. Όταν το BOE Shield-Bot κινηθεί πάλι προς τα εμπρός, το δεξιό whisker εφάπτεται στον τοίχο, οπότε κινείται προ τα πίσω και στρίβει αριστερά. Τότε έρχεται πάλι σε επαφή με το αριστερό τοίχωμα, και στη συνέχεια με το δεξί τοίχωμα, και ούτω καθεξής, μέχρις ότου κάποιος να το βγάλει από την δύσκολη αυτή κατάσταση.

Το πρόγραμμα RoamingWithWhiskers μπορεί να επεκταθεί για να εντοπίζει αυτό το πρόβλημα και να πράττει ανάλογα. Το κόλπο είναι να μετράει τον αριθμό των φορών που τα εναλλασσόμενα whiskers έρχονται σε επαφή με αντικείμενα. Για να το κάνουμε αυτό, το πρόγραμμα πρέπει να θυμάται σε ποια κατάσταση ήταν κάθε whisker κατά τη διάρκεια της προηγούμενης επαφής. Στη συνέχεια, θα πρέπει να συγκρίνει τις εν λόγω καταστάσεις με την τωρινή κατάσταση επαφής του whisker, και αν είναι αντίθετα, να προσθέτει ένα μετρητή. Εάν ο μετρητής ξεπεράσει ένα όριο που εμείς (ο προγραμματιστής) έχουμε καθορίσει, τότε ήρθε η ώρα να κάνει μια στροφή 180° και να ξεφύγει από τη γωνία, μηδενίζοντας ταυτόχρονα και το μετρητή.

2.11.4.1 Παράδειγμα προγράμματος - EscapingCorners

```

/*
 * Robotics with the BOE Shield - EscapingCorners
 * Count number of alternate whisker contacts, and if it exceeds 4, get
 *out of the corner.
 */

#include <Servo.h>                                // Include servo library

Servo servoLeft;                                  // Declare left and right servos
Servo servoRight;

byte wLeftOld;                                    // Previous loop whisker values
byte wRightOld;
byte counter;                                     // For counting alternate corners

void setup()                                     // Built-in initialization block
{
  pinMode(7, INPUT);                             // Set right whisker pin to input
  pinMode(5, INPUT);                             // Set left whisker pin to input
  pinMode(8, OUTPUT);                             // Left LED indicator -> output
  pinMode(2, OUTPUT);                             // Right LED indicator -> output

  tone(4, 3000, 1000);                            // Play tone for 1 second
  delay(1000);                                    // Delay to finish tone

  servoLeft.attach(13);                           // Attach left signal to pin 13
  servoRight.attach(12);                          // Attach right signal to pin 12

  wLeftOld = 0;                                   // Init. previous whisker states
  wRightOld = 1;
  counter = 0;                                    // Initialize counter to 0
}

void loop()                                       // Main loop auto-repeats
{
  // Corner Escape

  byte wLeft = digitalRead(5);                    // Copy right result to wLeft
  byte wRight = digitalRead(7);                   // Copy left result to wRight

  if(wLeft != wRight)                             // One whisker pressed?
  {                                                 // Alternate from last time?
    if ((wLeft != wLeftOld) && (wRight != wRightOld))
    {
      counter++;                                  // Increase count by one
      wLeftOld = wLeft;                           // Record current for next rep
      wRightOld = wRight;
      if(counter == 4)                             // Stuck in a corner?
      {
        wLeft = 0;                                 // Set up for U-turn
        wRight = 0;
        counter = 0;                               // Clear alternate corner count
      }
    }
  }
}

```

```

    }
    else // Not alternate from last time
    {
        counter = 0; // Clear alternate corner count
    }
}

// Whisker Navigation
if((wLeft == 0) && (wRight == 0)) // If both whiskers contact
{
    backward(1000); // Back up 1 second
    turnLeft(800); // Turn left about 120 degrees
}
else if(wLeft == 0) // If only left whisker contact
{
    backward(1000); // Back up 1 second
    turnRight(400); // Turn right about 60 degrees
}
else if(wRight == 0) // If only right whisker contact
{
    backward(1000); // Back up 1 second
    turnLeft(400); // Turn left about 60 degrees
}
else // Otherwise, no whisker contact
{
    forward(20); // Forward 1/50 of a second
}
}

void forward(int time) // Forward function
{
    servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
    servoRight.writeMicroseconds(1300); // Right wheel clockwise
    delay(time); // Maneuver for time ms
}

void turnLeft(int time) // Left turn function
{
    servoLeft.writeMicroseconds(1300); // Left wheel clockwise
    servoRight.writeMicroseconds(1300); // Right wheel clockwise
    delay(time); // Maneuver for time ms
}

void turnRight(int time) // Right turn function
{
    servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
    servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
    delay(time); // Maneuver for time ms
}

void backward(int time) // Backward function
{
    servoLeft.writeMicroseconds(1300); // Left wheel clockwise
    servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
    delay(time); // Maneuver for time ms
}

```

2.11.4.2 Πως λειτουργεί το EscapingCorners

Πρώτον, προστίθενται τρεις καθολικές μεταβλητές byte: wLeftOld, wRightOld και counter. Οι μεταβλητές wLeftOld και wRightOld αποθηκεύουν τις καταστάσεις των whisker από προηγούμενες επαφές τους, έτσι ώστε να μπορούν να συγκριθούν με τις καταστάσεις της τρέχουσας επαφής. Στη συνέχεια, ο μετρητής χρησιμοποιείται για να καταγράψει τον αριθμό των διαδοχικών και εναλλασσόμενων επαφών.

```
byte wLeftOld;           // Previous loop whisker values
byte wRightOld;
byte counter;          // For counting alternate corners
```

Αυτές οι μεταβλητές αρχικοποιούνται στη συνάρτηση setup. Η μεταβλητή counter μπορεί να ξεκινάει με μηδέν, αλλά μια από τις "old" μεταβλητές πρέπει να οριστεί με 1. Δεδομένου ότι η ρουτίνα για την ανίχνευση των γωνιών ψάχνει πάντα για μια εναλλασσόμενη διάταξη και τη συγκρίνει με την προηγούμενη εναλλασσόμενη διάταξη, πρέπει να υπάρχει πάντα μια αρχική εναλλασσόμενη διάταξη για να ξεκινήσει. Έτσι, στη wLeftOld και στη wRightOld ανατίθενται αρχικές τιμές στη συνάρτηση setup πριν ξεκινήσει η συνάρτηση loop τον έλεγχο και την τροποποίηση των τιμών τους.

```
wLeftOld = 0;           // Initialize previous whisker
wRightOld = 1;         // states
counter = 0;           // Initialize counter to 0
```

Το παρακάτω κομμάτι κώδικα «// Corner Escape» ελέγχει αν κάποιο από τα whiskers πιέζεται, χρησιμοποιώντας τον τελεστή ανισότητας (!=) σε μια if δήλωση. Το if(wLeft != wRight) μεταφράζεται σε : «αν η μεταβλητή wLeft δεν είναι ίση με την μεταβλητή wRight...»

```
// Corner Escape
if(wLeft != wRight)    // One whisker pressed?
```

Αν δεν είναι ίσες σημαίνει ότι ένα από τα whisker πιέζεται, και το πρόγραμμα πρέπει να ελέγξει αν είναι το αντίθετο μοτίβο με την προηγούμενη επαφή των whisker. Για να γίνει αυτό, μια if δήλωση ελέγχει εάν οι τρέχουσες τιμές wLeft και wRight είναι

διαφορετικές από τις προηγούμενες. Αν και οι δύο συνθήκες είναι αληθείς, προσθέτει 1 στην μεταβλητή counter (που παρακολουθεί τις εναλλακτικές επαφές των whisker) και αποθηκεύει την τρέχουσα διάταξη των whisker, θέτοντας τη wLeftOld και τη wRightOld ίσες με τις τρέχουσες wLeft και wRight αντίστοιχα.

```
if((wLeft != wLeftOld) && (wRight != wRightOld))
{
    counter++;           // Increase count by one
    wLeftOld = wLeft;   // Record current for next rep
    wRightOld = wRight;
```

Όταν η δήλωση if (counter == 4) είναι αληθείς, το μπλοκ του κώδικα της ξεγελά τη ρουτίνα πλοήγησης των whisker στη σκέψη ότι και τα δύο whisker πιέζονται, θέτοντας τη wLeft και τη wRight στο μηδέν. Αυτό κάνει τη ρουτίνα πλοήγησης να πιστέψει ότι και τα δύο whisker έχουν πιεστεί, έτσι ώστε να κάνει μια U-στροφή (στροφή 180°).

```
if(counter == 4)      // Stuck in a corner?
{
    wLeft = 0;        // Set up whisker states for U-turn
    wRight = 0;
    counter = 0;      // Clear alternate corner count
}
```

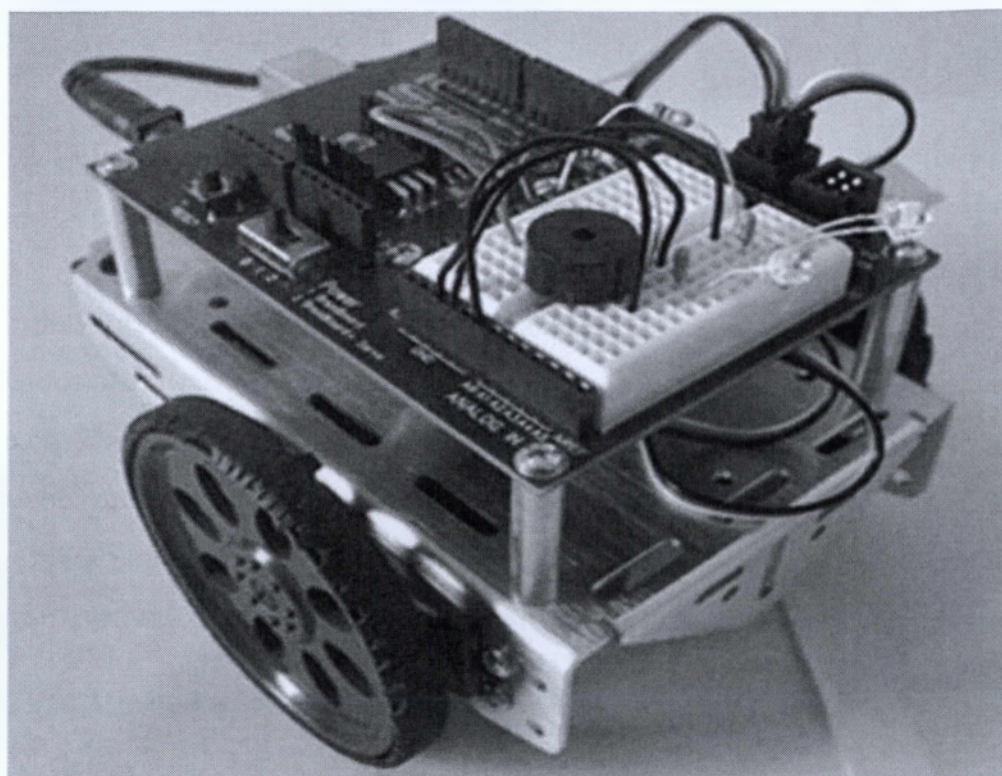
Αλλά, αν η συνθήκη στην if ((wLeft != wLeftOld) && (wRight != wRightOld)) δεν είναι αληθείς, σημαίνει ότι δεν υπάρχει καμία αλληλουχία από εναλλασσόμενες επαφές whisker. Στην περίπτωση αυτή η μεταβλητή counter ορίζεται στο μηδέν, έτσι ώστε να μπορεί να αρχίσει να υπολογίζει και πάλι όταν βρει μια γωνία.

```
else                  // Not alternate from last time
{
    counter = 0;      // Clear alternate corner count
}
```

2.12 Φωτοευαίσθητη Πλοήγηση με Φωτοτρανζίστορ

Οι αισθητήρες φωτός έχουν πολλές εφαρμογές στη ρομποτική και τον έλεγχο της βιομηχανικής (βρίσκουν την άκρη του ρολού του υφάσματος σε ένα υφαντουργείο, προσδιορίζουν πότε να ενεργοποιήσουν τα φώτα του δρόμου σε διαφορετικές χρονικές στιγμές του έτους, για να τραβήξουμε φωτογραφίες, για την παροχή νερού στις καλλιέργειες των φυτών.

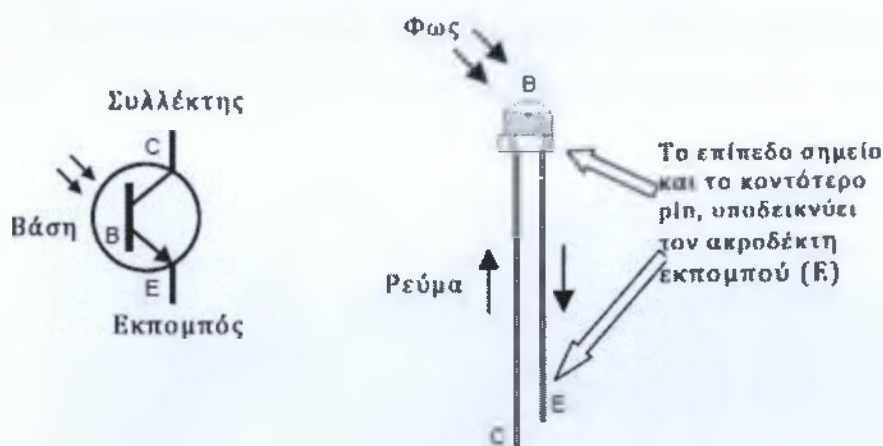
Οι αισθητήρες φωτός στο ρομπότ μας ανταποκρίνονται στο ορατό φως, καθώς επίσης και σε ένα αόρατο είδος του φωτός που ονομάζεται υπέρυθρες. Αυτοί οι αισθητήρες μπορούν να χρησιμοποιηθούν σε διάφορα κυκλώματα, που το Arduino μπορεί να παρακολουθεί για την ανίχνευση αλλαγών στο επίπεδο του φωτός. Με αυτές τις πληροφορίες, το πρόγραμμα μας μπορεί να επεκταθεί για την πλοήγηση του BOE Shield-Bot με το φως, όπως η οδήγηση προς μία δέσμη φωτός ενός φακού ή μια ανοιχτή πόρτα αφήνοντας το φως να μπαίνει σε ένα σκοτεινό δωμάτιο.



2.12.1 Παρουσίαση του the Φωτοτρανζίστορ

Ένα τρανζίστορ είναι σαν μια βαλβίδα που ρυθμίζει την ποσότητα του ηλεκτρικού ρεύματος που περνά μέσα των δύο από τα τρία τερματικά του. Το τρίτο τερματικό ρυθμίζει πόσο ρεύμα περνά μέσα από τα άλλα δύο. Ανάλογα με τον τύπο του τρανζίστορ, η ροή του ρεύματος μπορεί να ρυθμίζεται από τη τάση, το ρεύμα, ή στην περίπτωση του φωτοτρανζίστορ, από το φως.

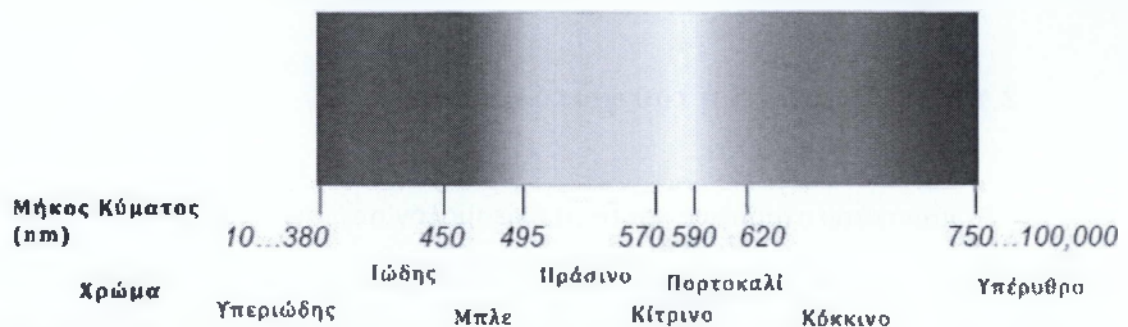
Το παρακάτω σχήμα δείχνει το διάγραμμα και το σχέδιο του φωτοτρανζίστορ που θα χρησιμοποιήσουμε. Η ένταση του φωτός που λάμπει στη βάση του φωτοτρανζίστορ (τερματικό Β) καθορίζει πόσο ρεύμα θα επιτρέψει να περάσει μέσα από το συλλέκτη (τερματικό C), και έξω από το τρανζίστορ μέσω του εκπομπού (τερματικό E). Λαμπρότερο φως οδηγεί σε περισσότερο ρεύμα και αντίστοιχα, λιγότερο φως οδηγεί σε λιγότερο ρεύμα.



Το φωτοτρανζίστορ μοιάζει λίγο με ένα LED, έχοντας δύο ομοιότητες. Κατ' αρχάς, εάν συνδέσουμε το φωτοτρανζίστορ ανάποδα στο κύκλωμα μας, δεν θα λειτουργήσει σωστά. Δεύτερον, έχει δύο διαφορετικού μήκους ρήν (περόνες) και ένα επίπεδο σημείο στην πλαστική θήκη της για τον εντοπισμό των τερματικών της. Το μεγαλύτερο από τα δύο ρήν υποδεικνύει το τερματικό του συλλέκτη, και το πιο κοντό υποδεικνύει τον εκπομπό, ο οποίος συνδέεται πιο κοντά στο επίπεδο σημείο της διάφανης πλαστικής θήκης του φωτοτρανζίστορ.

2.12.1.1 Κύματα φωτός

Στον ωκεανό, μπορούμε να μετρήσουμε την απόσταση μεταξύ των κορυφών των δύο γειτονικών κυμάτων σε πόδια ή μέτρα. Με το φως, το οποίο ταξιδεύει επίσης σε κύματα, η απόσταση μεταξύ γειτονικών κορυφών μετρείται σε νανόμετρα (nm), τα οποία είναι δισεκατομμυριοστά του μέτρου. Το παρακάτω σχήμα δείχνει τα μήκη κύματος για τα χρώματα του φωτός που είμαστε εξοικειωμένοι, μαζί με κάποια που το ανθρώπινο μάτι δεν μπορεί να ανιχνεύσει, όπως υπεριώδεις και υπέρυθρες.



Το φωτοτρανζίστορ που θα χρησιμοποιήσουμε στο ρομπότ είναι πιο ευαίσθητο σε 850 nm μήκη κύματος, τα οποία είναι στην υπέρυθρη περιοχή. Το υπέρυθρο φως δεν είναι ορατό στο ανθρώπινο μάτι, αλλά πολλές διαφορετικές πηγές φωτός εκπέμπουν σημαντικές ποσότητες, όπως το αλογόνο, οι λαμπτήρες πυρακτώσεως και φυσικά ο ήλιος. Αυτό το φωτοτρανζίστορ ανταποκρίνεται επίσης στο ορατό φως, αν και είναι λιγότερο ευαίσθητο, ιδιαίτερα σε μήκη κύματος μικρότερα των 450 nm.

2.12.2 Απλοί αισθητήρες φωτός

Κατάλογος εξαρτημάτων :

- (1) φωτοτρανζίστορ
- (2) καλώδια
- (1) αντίσταση, 2 kΩ (κόκκινο-μαύρο-κόκκινο)
- (1) λάμπα πυρακτώσεως ή φθορισμού

Μετά από κάποιες δοκιμές, και ανάλογα με τις συνθήκες φωτισμού στο χώρο της

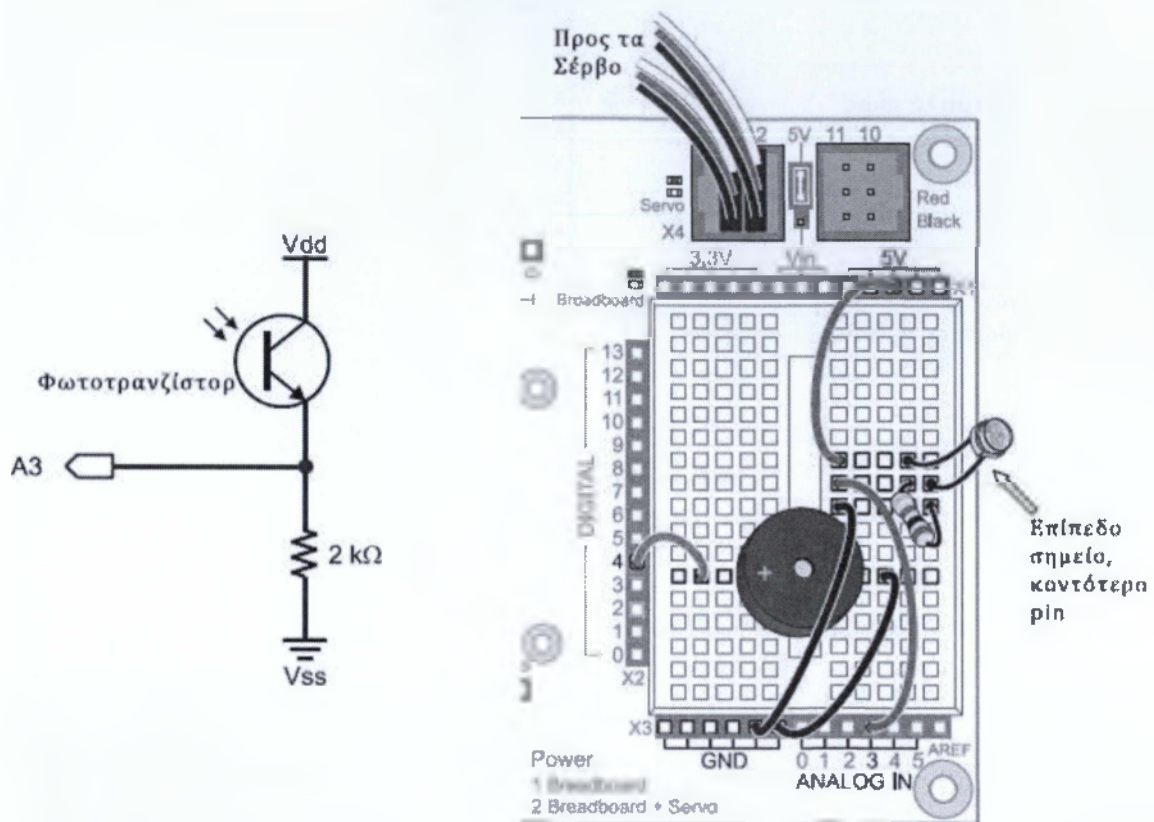
εργασίας μας, μπορεί να αντικαταστήσουμε την αντίσταση 2 kΩ με μία από τις παρακάτω αντιστάσεις:

- (1) αντίσταση, 220 Ω (κόκκινο-κόκκινο-καφέ)
- (1) αντίσταση, 470 Ω (κίτρινο-βιολετί-καφέ)
- (1) αντίσταση, 1 kΩ (καφέ-μαύρο-κόκκινο)
- (1) αντίσταση, 4,7 kΩ (κίτρινο-βιολετί-κόκκινο)
- (1) αντίσταση, 10 kΩ (καφέ-μαύρο-πορτοκαλί)

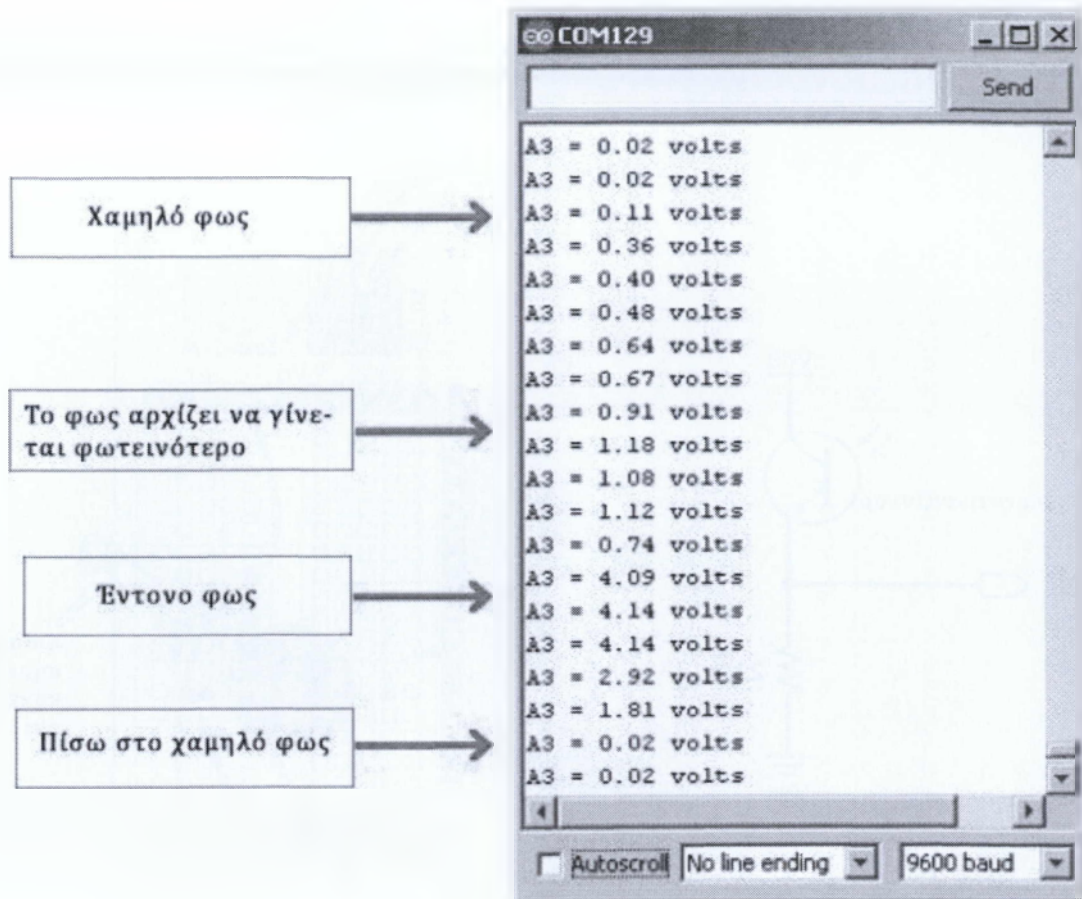
2.12.2.1 Κατασκευή του φωτο-ανιχνευτή

Το παρακάτω σχηματικό και της συνδεσμολογίας διάγραμμα μας δείχνουν ένα κύκλωμα που είναι πολύ παρόμοιο με αυτό που υπάρχει στα φώτα των δρόμων που ανάβουν αυτόματα τη νύχτα. Το κύκλωμα εξάγει μία τάση που ποικίλλει ανάλογα με το πόσο φως λάμπει στο φωτοτρανζίστορ, και ο Arduino παρακολουθεί το επίπεδο τάσης με ένα αναλογικού εισόδου pin.

2.12.2.2 Παράδειγμα προγράμματος - PhototransistorVoltage



Το πρόγραμμα PhototransistorVoltage εμφανίζει στο Serial Monitor την τάση που μετράται στο A3 (ένα από τα πέντε αναλογικά κανάλια εισόδου του Arduino, τα οποία είναι προσβάσιμα μέσω της BOE Shield). Στο κύκλωμα που μόλις είδαμε, ένα καλώδιο συνδέει το A3 στη σειρά όπου συναντιούνται ο εκπομπός (emitter) του φωτοτρανζίστορ και η αντίσταση. Η τάση σε αυτό το τμήμα του κυκλώματος θα αλλάζει όταν τροποποιείται το επίπεδο του φωτός που ανιχνεύεται από το φωτοτρανζίστορ. Το παρακάτω Serial screenshot Monitor δείχνει κάποιες μετρήσεις τάσεως.



Αν το φως του περιβάλλοντος είναι πιο φωτεινό από τα φώτα φθορισμού , ή έχουμε ένα φωτεινό φακό , μπορεί να χρειαστεί να αντικαταστήσουμε την αντίσταση 2 kΩ με μια μικρότερη (1 kΩ , 470 Ω , ή ακόμα και 220 Ω για πραγματικά λαμπερά φώτα). Αν το φως του περιβάλλοντος είναι χαμηλό και χρησιμοποιούμε λάμπα φθορισμού (φωτιστικό γραφείου) ή ένα LED φακό, μπορεί να χρειαστεί να αλλάξουμε την αντίσταση 2 kΩ με μια μεγαλύτερη, όπως 4,7 kΩ ή ακόμα και 10 kΩ.

```

/*
 * Robotics with the BOE Shield - PhototransistorVoltage
 * Display voltage of phototransistor circuit output connected to A3 in
 * the serial monitor.
 */

void setup()                               // Built-in initialization
block
{
  Serial.begin(9600);                       // Set data rate to 9600
  bps
}

void loop()                                 // Main loop auto-repeats
{
  Serial.print("A3 = ");                    // Display "A3 = "

```

```

    Serial.print(volts(A3));           // Display measured A3
    volts
    Serial.println(" volts");        // Display " volts" &
    newline
    delay(1000);                      // Delay for 1 second
}

float volts(int adPin)               // Measures volts at adPin
{                                     // Returns floating point
    voltage
    return float(analogRead(adPin)) * 5.0 / 1024.0;
}

```

2.12.2.3 Ακίνητοποίηση κάτω από φετινό φως

Το πρόγραμμα HaltUnderBrightLight κάνει το BOE Shield – Bot να κινείται προς τα εμπρός μέχρι το φωτοτρανζίστορ ανιχνεύσει φως που είναι αρκετά πιο φωτεινό, για να κάνει τη τάση που εφαρμόζεται στο A3 να υπερβεί τα 3,5V. Μπορούμε να αλλάξουμε την τιμή (3,5 V) ανάλογα με το φως που χρησιμοποιούμε και τις τιμές που καταγράψαμε στο Serial Monitor, έτσι ώστε να παρκάρει ακριβώς κάτω από αυτό το έντονο φως .

```

/*
 * Robotics with the BOE Shield - HaltUnderBrightLight
 * Display voltage of phototransistor circuit output connected to A3 in
 * the serial monitor.
 */

#include <Servo.h>                   // Include servo library

Servo servoLeft;                    // Declare left and right servos
Servo servoRight;

void setup()                         // Built-in initialization block
{
    tone(4, 3000, 1000);            // Play tone for 1 second
    delay(1000);                    // Delay to finish tone

    servoLeft.attach(13);           // Attach left signal to pin 13
    servoRight.attach(12);          // Attach right signal to pin 12

    servoLeft.writeMicroseconds(1700); // Full speed forward
    servoRight.writeMicroseconds(1300);

}

void loop()                          // Main loop auto-repeats
{
    if(volts(A3) > 3.5)             // If A3 voltage greater than 3.5
    {
        servoLeft.detach();         // Stop servo signals
        servoRight.detach();
    }
}

```

```

    }
}

float volts(int adPin) // Measures volts at adPin
{ // Returns floating point voltage
  return float(analogRead(adPin)) * 5.0 / 1024.0;
}

```

2.12.2.4 Πως λειτουργεί η συνάρτηση Volts

Οι υποδοχές A0, A1... A5 του Arduino συνδέονται με pins μικροελεγκτή Atmel που έχουν ρυθμιστεί για μετατροπή αναλογικού σε ψηφιακού. Οι μικροελεγκτές μετράνε την τάση χωρίζοντας ένα εύρος της τάσης σε πολλούς αριθμούς, με κάθε αριθμό να αντιπροσωπεύει μια τάση. Κάθε μία από τις αναλογικές εισόδους του Arduino έχει ανάλυση 10-bit, που σημαίνει ότι χρησιμοποιεί 10 δυαδικά ψηφία για να περιγράψει τη μέτρηση της τάσης. Με 10 δυαδικά ψηφία μπορούμε να υπολογίσουμε συνολικά 1024 επίπεδα τάσης, συμπεριλαμβανόμενου και του μηδέν.

Από προεπιλογή, η λειτουργία analogRead του Arduino έχει ρυθμιστεί να χρησιμοποιεί τις 0 ... 1023 τιμές για να περιγράψει που εμπίπτει μια μέτρηση τάσης σε μια κλίμακα 5 V. Αν χωρίσουμε 5 V σε 1024 διαφορετικά επίπεδα, κάθε επίπεδο είναι $5/1024$ του βολτ ($5/1024$ του βολτ είναι περίπου 0.004882813 V, ή περίπου 4,89 χιλιοστά του βολτ). Οπότε, για να μετατρέψουμε μια τιμή που επιστρέφεται από το analogRead σε V (Volt), πρέπει να την πολλαπλασιάσουμε με 5 και να τη διαιρέσουμε με το 1024.

Παράδειγμα:

Η συνάρτηση analogRead επιστρέφει 645. Πόσα βολτ είναι αυτό;

Απάντηση:

$$\begin{aligned}
 V &= 645 \times \frac{5V}{1024} \\
 &= 3.1494140625 V \\
 &\approx 3.15 V
 \end{aligned}$$

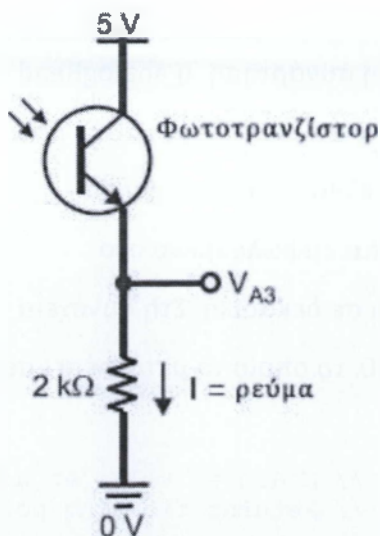
Στο αμέσως παραπάνω πρόγραμμα καλούμε τη volts συνάρτηση με volt(A3), η οποία περνάει το A3 στην παράμετρο adPin. Μέσα στη συνάρτηση, η analogRead (adPin) γίνεται analogRead (A3) και επιστρέφει μια τιμή από το 0 έως το 1023 που αντιπροσωπεύει την τάση που εφαρμόζεται στο A3. Η κλήση της analogRead επιστρέφει έναν ακέραιο, αλλά δεδομένου ότι βρίσκεται εμφωλευμένο στο float(analogRead (adPin)), η ακέραια τιμή μετατρέπεται σε δεκαδική. Στη συνέχεια πολλαπλασιάζεται με το 5.0 και διαιρείται με το 1024,0, το οποίο το μετατρέπει σε Volt.

```
float volts(int adPin)           // Measures volts at adPin
{                                 // Returns floating point
  voltage
  return float(analogRead(adPin)) * 5.0 / 1024.0;
}
```

2.12.2.5 Πως λειτουργεί το κύκλωμα του φωτοτρανζίστορ

Ένας αντιστάτης " αντιστέκεται " τη ροή του ρεύματος. Η τάση σε ένα κύκλωμα με αντίσταση μπορεί να παρομοιαστεί με την πίεση του νερού . Για μια δεδομένη ποσότητα ηλεκτρικού ρεύματος , περισσότερη τάση (πίεση) χάνεται σε μεγαλύτερη αντιστάτη από μια μικρότερη που έχει την ίδια ποσότητα ρεύματος και διέρχεται από αυτήν. Αν κρατήσουμε την αντίσταση σταθερή και μεταβάλλουμε το ρεύμα, μπορούμε να μετρήσουμε περισσότερη τάση (πτώση πίεσης) σε όλη την αντίσταση με περισσότερο ρεύμα , ή λιγότερη τάση με λιγότερο ρεύμα.

Οι αναλογικές είσοδοι του Arduino είναι αόρατες στο κύκλωμα του φωτοτρανζίστορ, οπότε η A3 παρακολουθεί το κύκλωμα , αλλά δεν έχει καμία επίδραση σε αυτό. Στο παρακάτω κύκλωμα τα 5V της ηλεκτρικής πίεσης (τάση) δραστηριοποιεί τα ηλεκτρόνια των μπαταριών να ρεύσουν μέσα από αυτό. Ο λόγος που η τάση στο A3 (V_{A3}) αλλάζει με το φως είναι επειδή το φωτοτρανζίστορ αφήνει περισσότερο ρεύμα όταν περάσει περισσότερο φως σε αυτό, ή αντίστοιχα λιγότερο ρεύμα με λιγότερο φως. Το ρεύμα, το οποίο είναι σημασμένο με «I», πρέπει επίσης να περάσει μέσα από τον αντιστάτη. Όταν περισσότερο ρεύμα περνά μέσα από έναν αντιστάτη, η τάση δια μέσου του θα είναι υψηλότερη και χαμηλότερη όταν περνά λιγότερο ρεύμα. Αφού από το ένα άκρο του αντιστάτη είναι συνδεδεμένη μια γείωση με $V_{SS} = 0 V$, η τάση στο V_{A3} ανεβαίνει με περισσότερο ρεύμα και κατεβαίνει με λιγότερο ρεύμα.



Αν αντικαταστήσουμε την αντίσταση $2\text{ k}\Omega$ με μια αντίσταση $1\text{ k}\Omega$, η V_{A3} θα βλέπει μικρότερες τιμές για τα ίδια ρεύματα. Στην πραγματικότητα, θα χρειαστεί διπλάσιο ρεύμα για να πάει την V_{A3} στο ίδιο επίπεδο τάσης, πράγμα που σημαίνει ότι το φως θα πρέπει να είναι δύο φορές πιο φωτεινό για να φθάσει στο επίπεδο $3,5\text{ V}$ (η προεπιλεγμένη τάση του `HaltUnderBrightLight`, έτσι ώστε να κάνει στάση το `BOE Shield-Bot`). Έτσι, μια μικρότερη αντίσταση σε σειρά με το φωτοτρανζίστορ καθιστά το κύκλωμα λιγότερο ευαίσθητο στο φως.

Εάν τώρα αντικαταστήσουμε την αντίσταση $2\text{ k}\Omega$ με μια αντίσταση $10\text{ k}\Omega$, η V_{A3} θα είναι 5 φορές μεγαλύτερη με το ίδιο ρεύμα, και θα πάρει μόνο το $1/5^{\text{o}}$ του φωτός για τη δημιουργία $1/5^{\text{o}}$ ρεύματος για πάει την V_{A3} στο επίπεδο $3,5\text{ V}$. Έτσι, μία μεγαλύτερη αντίσταση κάνει το κύκλωμα πιο ευαίσθητο στο φως.

2.12.2.6 Νόμος του Ohm

Το ρεύμα και η αντίσταση είναι οι δύο ιδιότητες που επηρεάζουν την τάση στο V_{A3} , και ο νόμος του Ohm εξηγεί πώς αυτό λειτουργεί. Ο Νόμος του Ohm αναφέρει ότι η τάση (V) μέσω μιας αντίστασης είναι ίση με το ρεύμα (I) που διέρχεται από αυτή πολλαπλασιασμένο με την αντίστασή της (R). Οπότε, αν γνωρίζουμε δύο από αυτές τις τιμές, μπορούμε να χρησιμοποιήσουμε το νόμο του Ohm για να υπολογίσουμε το τρίτο:

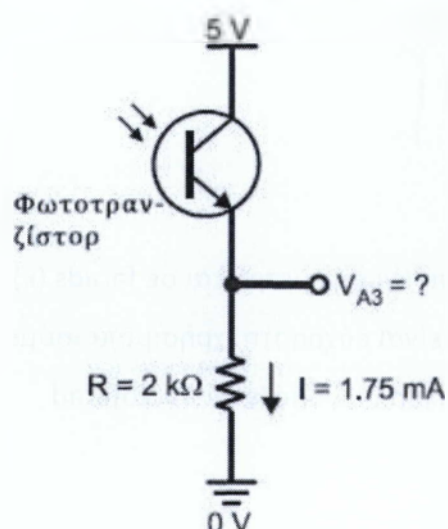
$$V = I \times R$$

Η αντίσταση (R) μετριέται σε ohms (συντομογραφία, Ω) και το ρεύμα (I) σε αμπέρ ή amps (συντομογραφία, A).

Στα παρακάτω παραδείγματα υπολογίζουμε τη τάση στη V_{A3} , χρησιμοποιώντας το νόμο του Ohm, αφήνοντας δύο διαφορετικές ποσότητες ρεύματος μέσω του κυκλώματος:

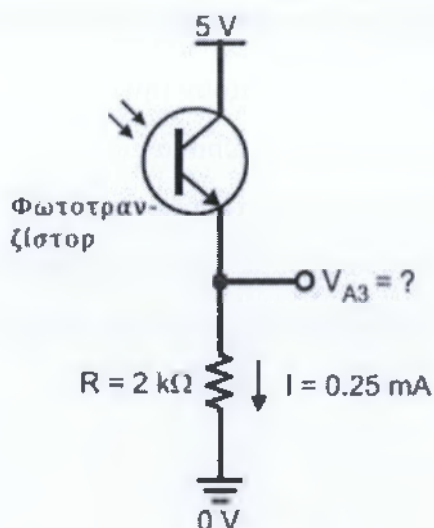
1. 1.75 mA, η οποία θα μπορούσε να συμβεί ως αποτέλεσμα αρκετά έντονου φωτός
2. 0.25 mA, το οποίο θα μπορούσε να συμβεί με λιγότερα έντονο φως

Παράδειγμα 1: $I = 1.75 \text{ mA}$ και $R = 2 \text{ k}\Omega$



$$\begin{aligned}
 V_{A3} &= I \times R \\
 &= 1.75 \text{ mA} \times 2 \text{ k}\Omega \\
 &= \frac{1.75}{1000} \text{ A} \times 2000 \Omega \\
 &= 1.75 \text{ A} \times 2 \Omega \\
 &= 3.5 \text{ A}\Omega \\
 &= 3.5 \text{ V}
 \end{aligned}$$

Παράδειγμα 2: $I = 0.25 \text{ mA}$ και $R = 2 \text{ k}\Omega$

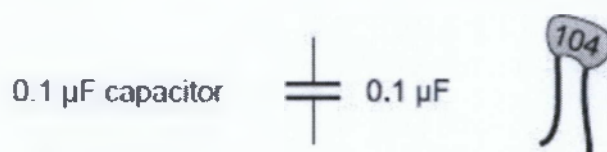


$$\begin{aligned}
 V_{A3} &= I \times R \\
 &= 0.25 \text{ mA} \times 2 \text{ k}\Omega \\
 &= \frac{0.25}{1000} \text{ A} \times 2000 \Omega \\
 &= 0.25 \text{ A} \times 2 \Omega \\
 &= 0.5 \text{ A}\Omega \\
 &= 0.5 \text{ V}
 \end{aligned}$$

2.12.3 Μέτρηση επιπέδων φωτός σε μεγαλύτερο εύρος

2.12.3.1 Παρουσίαση του Πυκνωτή

Ένας πυκνωτής είναι μια συσκευή που αποθηκεύει φορτίο και είναι ένα θεμελιώδες δομικό στοιχείο πολλών κυκλωμάτων. Οι μπαταρίες είναι επίσης συσκευές που αποθηκεύουν φορτίο, και είναι βολικό να σκεφτόμαστε τους πυκνωτές ως μικροσκοπικές μπαταρίες που μπορούν να φορτιστούν, να αποφορτιστούν και να επαναφορτιστούν.



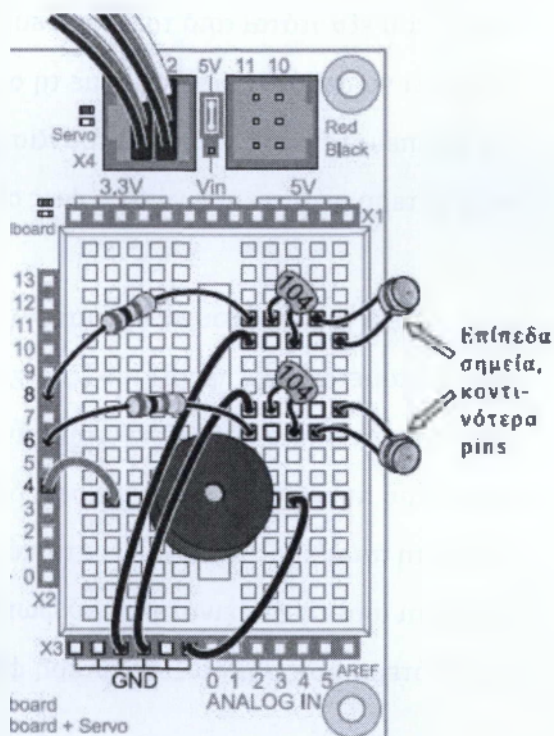
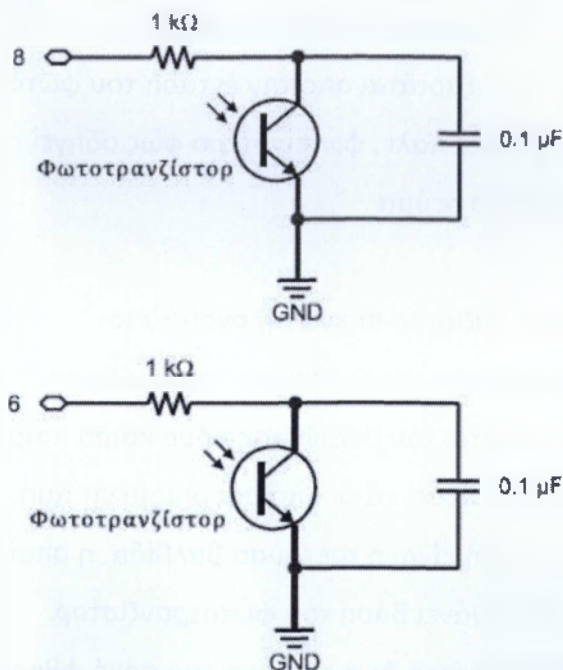
Το φορτίο που μπορεί να αποθηκεύσει ένας πυκνωτής μετριέται σε farads (F). Το farad είναι μια πολύ μεγάλη τιμή, και επειδή δεν είναι εύχρηστη, χρησιμοποιούμε το εκατομμυριοστό του farad. Ένα εκατομμυριοστό του farad ονομάζεται microfarad (συντομογραφία, μF).

2.12.3.2 Κατασκευή των φωτοευαίσθητων ματιών

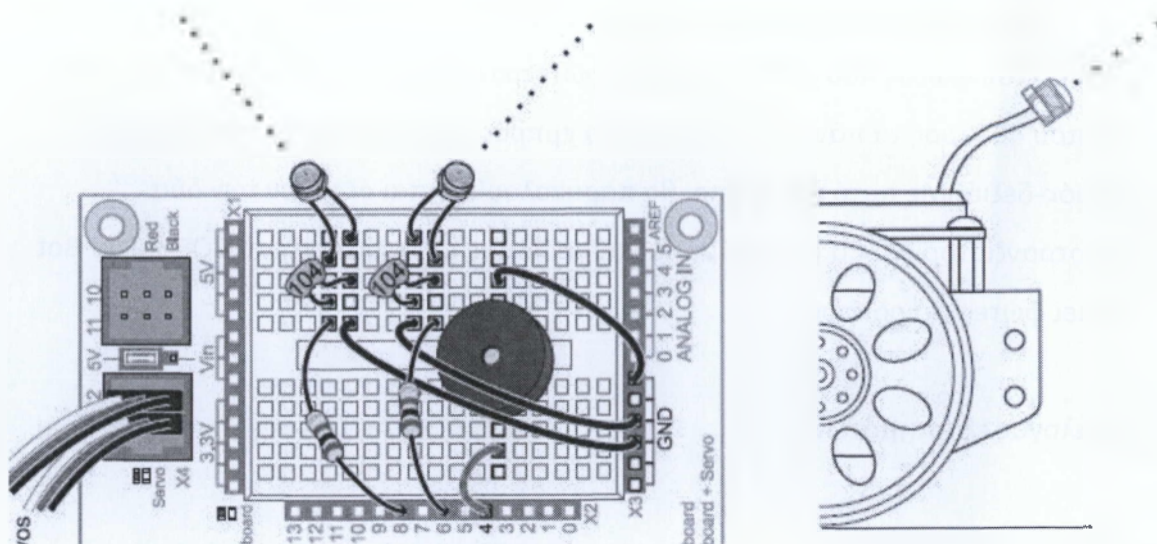
Τα επόμενα κυκλώματα θα μπορούν να ανταποκριθούν ανεξάρτητα με το επίπεδο του φωτός που φθάνει σε κάθε φωτοτρανζίστορ. Θα πρέπει έχουν μια κλίση περίπου 45° προς τα πάνω, το ένα προς τα εμπρός-αριστερά και το άλλο προς τα εμπρός-δεξιά. Με αυτό τον τρόπο, θα παρακολουθούνται οι τιμών των δύο φωτοτρανζίστορ και θα μπορούμε να καθορίσουμε ποια πλευρά του BOE Shield-Bot βλέπει φωτεινότερο φως.

Κατάλογος εξαρτημάτων:

- (2) φωτοτρανζίστορς
- (2) πυκνωτές, 0,1 μF (104)
- (2) αντιστάσεις, 1 k Ω (καφέ-μαύρο-κόκκινο)
- (2) καλώδια

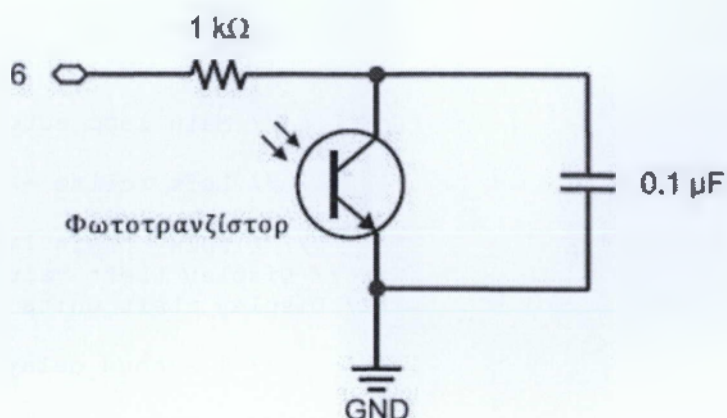


Προσαρμόζουμε τα φωτοτρανζίστορ σε 45° πάνω από το breadboard, και 90° προς τα έξω, όπως φαίνεται παρακάτω.



Κάθε πυκνωτής μπορεί να φορτιστεί με 5V και στη συνέχεια αφήνεται να στραγγιστεί μέσα από φωτοτρανζίστορ του. Το ποσοστό που ο πυκνωτής χάνει το φορτίο του εξαρτάται από το πόσο ρεύμα το φωτοτρανζίστορ (βαλβίδα ρεύματος) επιτρέπει να περάσει, το οποίο με τη σειρά του εξαρτάται από την ένταση του φωτός που λάμπει στη βάση του φωτοτρανζίστορ του. Και πάλι, φωτεινότερο φως οδηγεί σε περισσότερο ρεύμα και λιγότερο φως σε λιγότερο ρεύμα.

Αυτό το είδος του κυκλώματος (φωτοτρανζίστορ-πυκνωτή) ονομάζεται «κύκλωμα μεταφοράς φορτίου» (charge transfer circuit). Ο Arduino υπολογίζει το ρυθμό με τον οποίο κάθε πυκνωτής χάνει το φορτίο του μετρώντας πόσο καιρό παίρνει η τάση του πυκνωτή να παρακάσει, δηλαδή να πέσει κάτω από μια ορισμένη τιμή. Ο χρόνος τη παρακμής αντιστοιχεί στο πόσο ανοιχτή είναι η τρέχουσα βαλβίδα, η οποία ελέγχεται από τη φωτεινότητα του φωτός που φθάνει βάση του φωτοτρανζίστορ. Περισσότερο φως σημαίνει ταχύτερη φθορά, λιγότερο φως σημαίνει πιο αργή φθορά.



2.12.3.3 Έλεγχος του κυκλώματος του φωτοτρανζίστορ

Το πρόγραμμα LeftLightSensor φορτίζει τον πυκνωτή στη pin 8 QT κυκλώματος, μετρά την φθορά του χρόνου τάσης και τέλος την εμφανίζει στο Serial Monitor. Στο παρακάτω πρόγραμμα, μικρότεροι αριθμοί σημαίνουν φωτεινότερο φως.

```

/*
 * Robotics with the BOE Shield - LeftLightSensor
 * Measures and displays microsecond decay time for left light sensor.
 */

void setup()                                     // Built-in initialization
block
{
  tone(4, 3000, 1000);                          // Play tone for 1 second
  delay(1000);                                   // Delay to finish tone

```

```

    Serial.begin(9600);           // Set data rate to 9600 bps
}

void loop()                      // Main loop auto-repeats
{
    long tLeft = rcTime(8);      // Left rcTime -> tLeft

    Serial.print("tLeft = ");   // Display tLeft label
    Serial.print(tLeft);        // Display tLeft value
    Serial.println(" us");      // Display tLeft units + newline

    delay(1000);                // 1 second delay
}

long rcTime(int pin)            // rcTime function at pin
{                               // ..returns decay time
    pinMode(pin, OUTPUT);       // Charge capacitor
    digitalWrite(pin, HIGH);   // ..by setting pin output-high
    delay(1);                  // ..for 5 ms
    pinMode(pin, INPUT);       // Set pin to input
    digitalWrite(pin, LOW);    // ..with no pullup
    long time = micros();      // Mark the time
    while(digitalRead(pin));   // Wait for voltage < threshold
    time = micros() - time;    // Calculate decay time
    return time;               // Return decay time
}

```

Πριν προχωρήσουμε στην πλοήγηση, θα πρέπει να εκτελέσουμε την ίδια δοκιμή στα δεξιά (pin 6) κύκλωμα του φωτοτρανζίστορ.

```

/*
 * Robotics with the BOE Shield - RightLightSensor
 * Measures and displays microsecond decay time for Right light sensor.
 */

void setup()                    // Built-in initialization
block
{
    tone(4, 3000, 1000);       // Play tone for 1 second
    delay(1000);               // Delay to finish tone

    Serial.begin(9600);        // Set data rate to 9600 bps
}

void loop()                    // Main loop auto-repeats
{
    long tRight = rcTime(6);   // Right rcTime -> tRight

    Serial.print("tRight = "); // Display tLeft label
    Serial.print(tRight);      // Display tLeft value
    Serial.println(" us");     // Display tLeft units + newline

    delay(1000);               // 1 second delay
}

long rcTime(int pin)          // rcTime function at pin
                               // ..returns decay time

```

```

{
  pinMode(pin, OUTPUT);           // Charge capacitor
  digitalWrite(pin, HIGH);       // ..by setting pin ouput=
high
  delay(1);                       // ..for 5 ms
  pinMode(pin, INPUT);           // Set pin to input
  digitalWrite(pin, LOW);        // ..with no pullup
  long time = micros();          // Mark the time
  while(digitalRead(pin));       // Wait for voltage <
threshold
  time = micros() - time;        // Calculate decay time
  return time;                   // Return decay time
}

```

2.12.3.4 rcTime και φθορά της τάσης

Όταν τα επίπεδα φωτισμού είναι χαμηλά, η συνάρτηση rcTime μπορεί να πάρει μετρήσεις του χρόνου που να είναι πολύ μεγάλες για int ή για word μεταβλητές. Το επόμενο βήμα στη χωρητικότητα αποθήκευσης είναι μια long μεταβλητή, η οποία μπορεί να αποθηκεύσει τιμές από -2.147.483.648 έως 2.147.483.647. Οπότε, ο ορισμός της συνάρτησης long rcTime(int pin) έχει συσταθεί για να κάνει τη συνάρτηση να επιστρέφει μια long τιμή, όταν καλείται. Θα πρέπει επίσης να γνωρίζουμε τα pin για τη μέτρηση μας.

```
long rcTime(int pin)
```

Μια μέτρηση μεταφοράς φορτίου χρειάζεται επτά βήματα: (1) Ρυθμίζει το I/O pin με high για τη φόρτιση του πυκνωτή. (2) Περιμένει αρκετά μεγάλο χρονικό διάστημα για το πυκνωτή να φορτιστεί. (3) Αλλάζει το I/O pin σε input. (4) Ελέγχει το χρόνο. (5) Περιμένει την τάση να πάει κάτω από το όριο των 2,1 V του Arduino. (6) Ελέγχει ξανά την ώρα. (7) Αφαιρεί το χρόνο του 3^{του} βήματος από το χρόνο στο βήμα-6 (αυτό είναι το χρονικό διάστημα που ο πυκνωτής χάνει το φορτίο του).

```

pinMode(pin, OUTPUT);           // Step 1, part 1
digitalWrite(pin, HIGH);        // Step 1, part 2
delay(1);                       // Step 2
pinMode(pin, INPUT);           // Step 3 part 1
digitalWrite(pin, LOW);        // Step 3, part 2
long time = micros();          // Step 4
while(digitalRead(pin));       // Step 5
time = micros() - time;        // Step 6 & 7

```

```
    return time;
}
```

Σε αυτό το πρόγραμμα, το βήμα-1 έχει δύο υπο-στάδια. Πρώτον, το `pinMode(pin, OUTPUT)` καθορίζει το I/O `pin` σε μια έξοδο, και μετά το `digitalWrite (pin, HIGH)` προμηθεύει το κύκλωμα με 5 V. Το βήμα-3 έχει επίσης δύο υπο-στάδια, επειδή το I/O `pin` στέλνει ένα high (υψηλό) σήμα. Όταν το πρόγραμμα αλλάζει την κατεύθυνση του I/O `pin` από `output-high` σε `input`, προσθέτει 10 kΩ αντίσταση στο κύκλωμα, η οποία πρέπει να αφαιρεθεί. Προσθέτοντας τη `digitalWrite (pin, LOW)` μετά τη `pinMode (pin, INPUT)` αφαιρεί την αντίσταση και επιτρέπει στο πυκνωτή να εξαντλήσει το φορτίο του κανονικά μέσω του φωτοτρανζίστορ.

2.12.4 Μετρήσεις φωτός για τη πλοήγηση

Η δήλωση `if(tLeft > 2500)` ίσως λειτουργήσει καλά για την αποφυγή των σκιών σε ένα δωμάτιο, αλλά σε ένα άλλο με φωτεινότερα φώτα, δεν θα μπορεί να εντοπίσει ούτε μια σκιά. Για μια σωστή πλοήγηση, δεν έχει σημασία ο πραγματικός αριθμός που αναφέρει τα επίπεδα φωτισμού σε κάθε αισθητήρα, αλλά η διαφορά στο πόσο φως οι δύο αισθητήρες ανιχνεύουν, έτσι ώστε το ρομπότ να στραφεί προς τον αισθητήρα που βλέπει λαμπρότερο φως (ή μακριά από αυτόν, ανάλογα με το τι θέλουμε). Οπότε, απλά διαιρούμε τη δεξιά μέτρηση του αισθητήρα με το άθροισμα και των δύο (το αποτέλεσμα θα είναι πάντα στο εύρος από 0 έως 1). Αυτή η τεχνική είναι ένα παράδειγμα μιας κανονικοποιημένης διαφορικής μέτρησης.

$$\textit{normalized differential shade} = \frac{tRight}{tRight + tLeft}$$

Για παράδειγμα, μια κανονικοποιημένη διαφορική μέτρηση των 0,25 σημαίνει «πως το φως είναι 1/2 ποιο φωτεινό πάνω από το δεξιό αισθητήρα, από ότι είναι πάνω από τον αριστερό». Οι πραγματικές τιμές των `tRight` και `tLeft` μπορεί να είναι μικρές σε ένα φωτεινό δωμάτιο ή μεγάλες σε ένα σκοτεινό δωμάτιο, αλλά η απάντηση θα εξακολουθεί να είναι 0,25 αν το φως είναι 1/2 ποιο φωτεινά στο δεξιό αισθητήρα. Μια

μέτρηση του 0,5 θα σήμαινε ότι οι τιμές t_{Right} και t_{Left} είναι ίσες, δηλαδή οι αισθητήρες ανιχνεύουν το ίδιο επίπεδο φωτεινότητας.

Αν αφαιρέσουμε 0,5 από την κανονικοποιημένη μέτρηση διαφορικής σκιάς (normalized differential shade measurement) τα αποτελέσματα θα κυμαίνονται από -0,5 έως 0,5 αντί από 0 έως 1 (μια μέτρηση του 0 σημαίνει ίση φωτεινότητα). Το αποτέλεσμα είναι μια μηδενικής αιτιολόγησης κανονικοποιημένη μέτρηση διαφορικής σκιάς (zero-justified normalized differential shade measurement, ndShade στα προγράμματα).

$$\text{zero justified normalized differential shade} = \frac{t_{Right}}{t_{Right} + t_{Left}} - 0.5$$

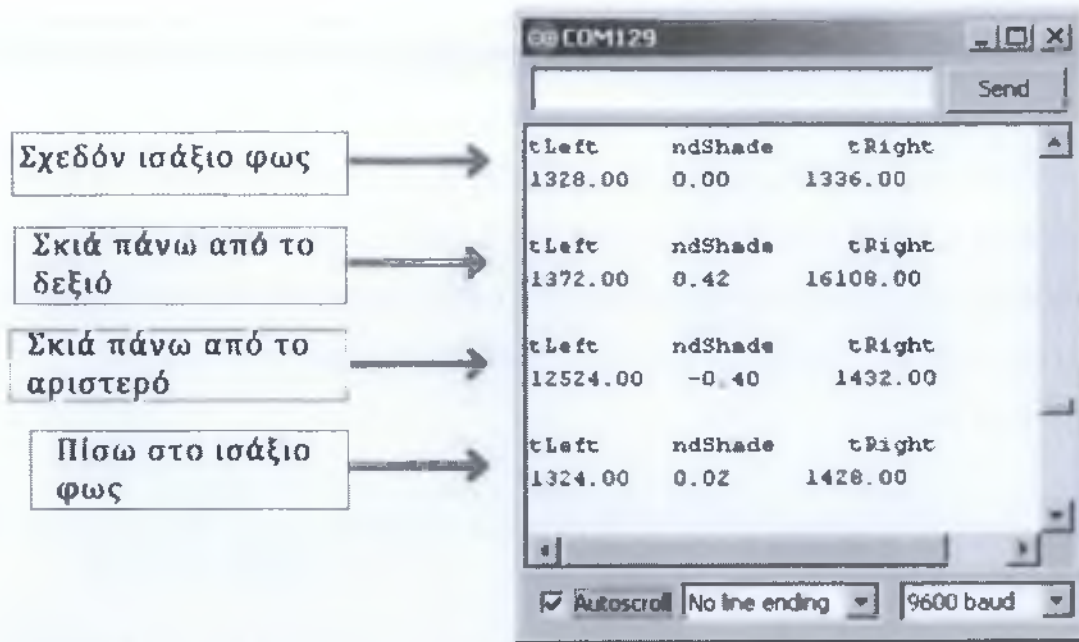
Αλλά γιατί να το κάνουμε; Το εύρος τιμών από -0,5 έως 0,5 είναι το κατάλληλο για τα προγράμματα πλοήγησης, διότι οι θετικές και αρνητικές τιμές μπορούν να χρησιμοποιηθούν για να κλιμακώσουν τις ταχύτητες των τροχών.

```
float ndShade; // Normalized differential shade
ndShade = tRight / (tLeft + tRight) - 0.5; // Calculate it and subtract
0.5
```

Πρώτα δηλώνεται η δεκαδική μεταβλητή, ονομαζόμενη ndShade, που θα αποθηκευτεί η τελική μέτρηση. Στη συνέχεια, γίνονται τα μαθηματικά της μηδενικής αιτιολόγησης κανονικοποιημένης μέτρησης διαφορικής σκιάς. Το αποτέλεσμα θα είναι μια τιμή στο εύρος από -0,5 έως 0,5 που αντιπροσωπεύει το κλάσμα της συνολικής σκιάς που ανιχνεύει η t_{Right} , σε σύγκριση με τη t_{Left} . Όταν η ndShade είναι 0 σημαίνει ότι η t_{Right} και η t_{Left} έχουν ίδιες τιμές, οπότε οι αισθητήρες ανιχνεύουν εξίσου έντονο φως. Όσο η ndShade συγκλίνει στη -0,5, τόσο πιο σκοτεινή είναι η σκιά πάνω από το δεξιό αισθητήρα, και αντίστοιχα, όσο η ndShade συγκλίνει στη 0,5, τόσο πιο σκοτεινή είναι η σκιά πάνω από τον αριστερό αισθητήρα.

2.12.4.1 Παράδειγμα προγράμματος - LightSensorValues

Το παρακάτω screencapture μας δείχνει ένα παράδειγμα της Serial Monitor με την λειτουργία του προγράμματος LightSensorValues.



```

/*
 * Robotics with the BOE Shield - LightSensorValues
 * Displays tLeft, ndShade and tRight in the Serial Monitor.
 */

void setup()                                     // Built-in initialization
block
{
  tone(4, 3000, 1000);                           // Play tone for 1 second
  delay(1000);                                    // Delay to finish tone

  Serial.begin(9600);                             // Set data rate to 9600 bps
}

void loop()                                       // Main loop auto-repeats
{
  float tLeft = float(rcTime(8));                // Get left light & make float
  float tRight = float(rcTime(6));               // Get right light & make float

  float ndShade;                                  // Normalized differential shade
  ndShade = tRight / (tLeft + tRight) - 0.5;    // Calculate it and
  subtract 0.5

  // Display heading
  Serial.println("tLeft      ndShade      tRight");

  Serial.print(tLeft);                            // Display tLeft value
  Serial.print("  ");                             // Display spaces
  Serial.print(ndShade);                          // Display ndShade value
  Serial.print("    ");                           // Display more spaces
  Serial.println(tRight);                         // Display tRight value
  Serial.println(' ');                            // Add an extra newline

  delay(1000);                                    // 1 second delay
}

long rcTime(int pin)                              // rcTime measures decay at pin
{

```



```

pinMode(pin, OUTPUT);           // Charge capacitor
digitalWrite(pin, HIGH);        // ..by setting pin output-high
delay(5);                       // ..for 5 ms
pinMode(pin, INPUT);           // Set pin to input
digitalWrite(pin, LOW);         // ..with no pullup
long time = micros();           // Mark the time
while(digitalRead(pin));        // Wait for voltage < threshold
time = micros() - time;         // Calculate decay time
return time;                    // Returns decay time
)

```

2.12.5 Δοκιμή μιας φωτοκίνητης ρουτίνας

Μπορούμε να χρησιμοποιήσετε τη μεταβλητή `ndShade` έτσι ώστε να στρίβει το BOE Shield-Bot λίγο ή πολύ, ανάλογα με την αντίθεση μεταξύ του φωτός που ανιχνεύεται σε κάθε πλευρά. Η παρακάτω `if` δήλωση αποφεύγει τυχόν σκιές που θα παρουσιάζονται στη δεξιά πλευρά του BOE Shield-Bot. Αρχίζει με τη δήλωση δύο `int` μεταβλητών, τη `speedLeft` και τη `speedRight`, οι οποίες δεν δηλώνονται μέσα στο `if ... else` κομμάτι, διότι θα χρησιμοποιηθούν και στη συνάρτηση `loop`. Στη συνέχεια, η `if(ndShade > 0.0)` έχει ένα μπλοκ κώδικα που θα εκτελεστεί αν εντοπιστεί σκιά στη δεξιά πλευρά του ρομπότ, επιβραδύνοντας τον αριστερό τροχό για την στροφή του ρομπότ μακριά από το σκοτάδι. Αυτό επιτυγχάνεται πολλαπλασιάζοντας τη `ndShade` με `1000.0` και μετά αφαιρώντας από το `200`. Πριν από την ανάθεση του αποτελέσματος στη `speedLeft`, η `int(200.0 - (ndShade * 1000.0))` μετατρέπει την απάντηση από δεκαδική σε ακέραια, έτσι ώστε η τιμή να είναι συμβατή με τη συνάρτηση ελιγμών, η οποία χρειάζεται ακέραια τιμή.

```

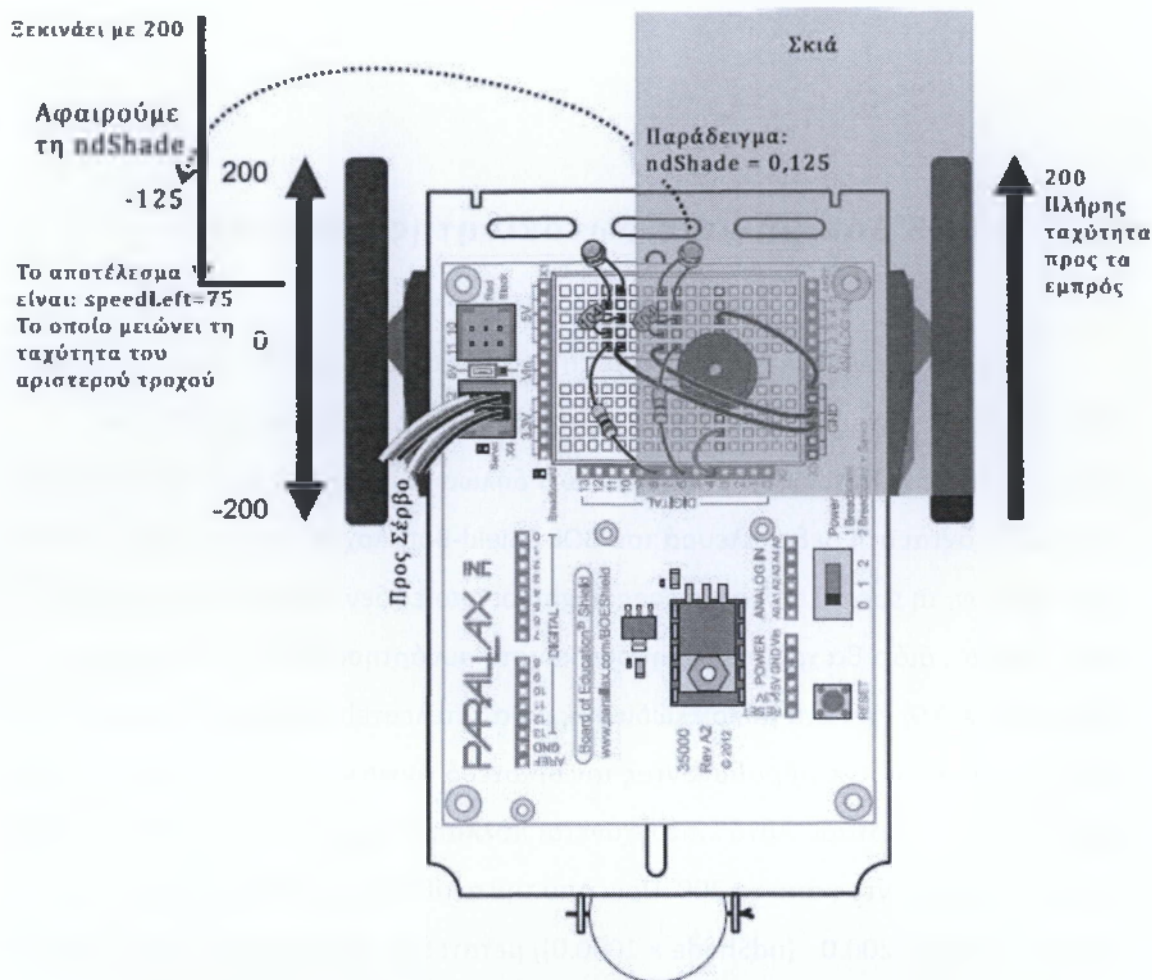
int speedLeft, speedRight;      // Declare speed variables

if (ndShade > 0.0)              // Shade on right?
{                                // Slow down left wheel
  speedLeft = int(200.0 - (ndShade * 1000.0));
  speedLeft = constrain(speedLeft, -200, 200);
  speedRight = 200;             // Full speed right wheel
}

```

Το παρακάτω παράδειγμα να δείχνει πώς λειτουργεί ο παραπάνω κώδικας όταν η `ndShade` είναι `0.125`. Ο αριστερός τροχός επιβραδύνει επειδή $200 - (0.125 \times 1000) = 75$, δηλαδή βάζει τον τροχό σε περίπου $\frac{3}{4}$ της πλήρους ταχύτητας (`100`). Εν τω μεταξύ,

από την άλλη πλευρά η speedRight έχει οριστεί με 200 για πλήρης ταχύτητα προς τα εμπρός.



Όσο μεγαλύτερη είναι η $ndShade$, τόσο περισσότερο αφαιρεί από το 200. Σε αυτό το παράδειγμα δεν έχουμε πρόβλημα, αλλά αν η $ndShade$ ήταν 0.45, θα προσπαθούμε να αποθηκεύσει -250 στη μεταβλητή $speedLeft$. Δεδομένου ότι οι ταχύτητες που θέλουμε να περάσουν στη συνάρτηση ελιγμών πρέπει να είναι στην περιοχή από -200 έως 200, θα χρησιμοποιήσουμε τη συνάρτηση περιορισμού (`constrain`) του Arduino ώστε να αποτρέψουμε τη $speedLeft$ να ξεπερνά τα όρια (`speedLeft = constrain(speedLeft, -200, 200)`).

Αν η `if` δεν ήταν αληθείς, τότε η `else` δήλωση στρέφει το ρομπότ μακριά από την αριστερή σκιά, επιβραδύνοντας το δεξιό τροχό και κρατώντας τον αριστερό τροχό σε πλήρη ταχύτητα προς τα εμπρός. Σε αυτή την περίπτωση η τιμή του ($ndShade * 1000$) προστίθεται με το 200, γιατί αν η $ndShade$ είναι -0,125 (πάντα αρνητική για την

αριστερή σκιά), οι πράξεις στην $speedRight = int(200.0 + (ndShade * 1000.0))$ θα αξιολογηθούν σε $200 + (-1.25 \times 1000) = 200 - 125 = 75$.

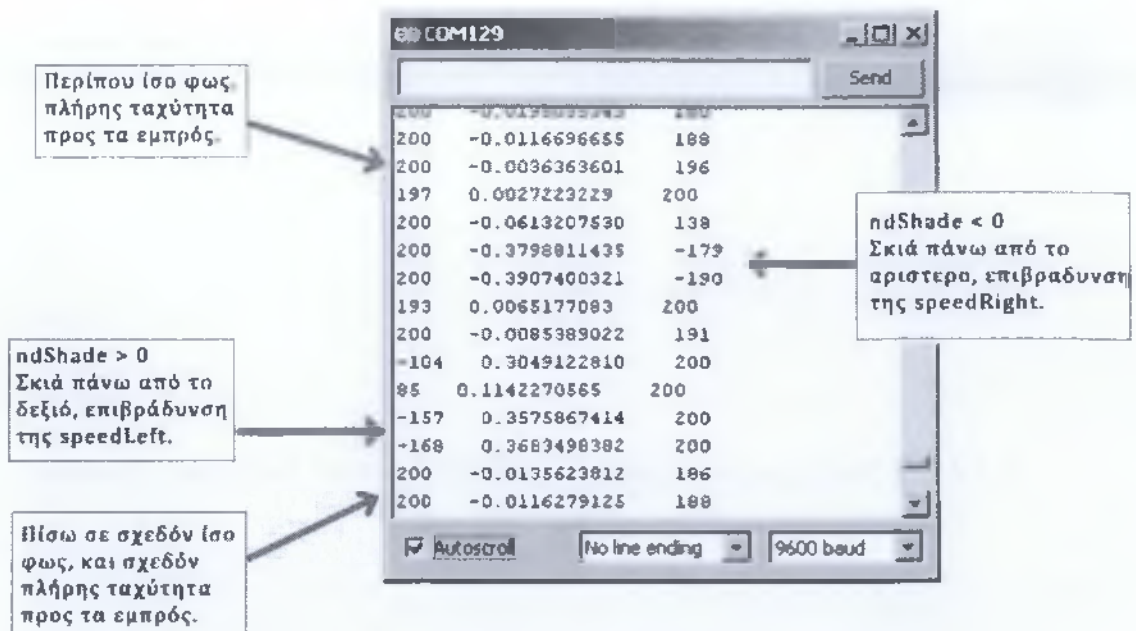
```
else // Shade on Left?
{ // Slow down right wheel
  speedRight = int(200.0 + (ndShade * 1000.0));
  speedRight = constrain(speedRight, -200, 200);
  speedLeft = 200; // Full speed left wheel
}
```

2.12.5.1 Έλεγχος αποφάσεων πλοήγησης με το Serial Monitor

```
Serial.print(speedLeft, DEC); // Display speedLeft
Serial.print(" "); // Spaces
Serial.print(ndShade, DEC); // Display ndShade
Serial.print(" "); // More spaces
Serial.println(speedRight, DEC); // Display speedRight

delay(2000); // 1 second delay
}
```

Οι κλήσεις των `print` και `println` παρουσιάζονται στο Serial Monitor, όπου οι τιμές τις `speedLeft` αναφέρονται στην αριστερή στήλη, τις `speedRight` στη δεξιά στήλη και τις `ndShade` ανάμεσα τους. Η πλευρά με το λαμπρότερο φως εμφανίζεται πάντα με 200 (πλήρη ταχύτητα) και η άλλη επιβραδύνει με τιμές κάτω από 200 (όσο πιο σκούρα η σκιά, τόσο μικρότερος και ο αριθμός).



2.12.5.2 Παράδειγμα προγράμματος - Light Seeking Display

```

/*
 * Robotics with the BOE Shield - LightSeekingDisplay
 * Displays speedLeft, ndShade, and speedRight in Serial Monitor.
 * Verifies
 * that wheel speeds respond correctly to left/right light/shade
 * conditions.
 */

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  Serial.begin(9600); // Set data rate to 9600 bps
}

void loop() // Main loop auto-repeats
{
  float tLeft = float(rcTime(8)); // Get left light & make float
  float tRight = float(rcTime(6)); // Get right light & make float

  float ndShade; // Normalized differential shade
  ndShade = tRight / (tLeft+tRight) - 0.5; // Calculate it and
  subtract 0.5

  int speedLeft, speedRight; // Declare speed variables

  if (ndShade > 0.0) // Shade on right?
  { // Slow down left wheel
    speedLeft = int(200.0 - (ndShade * 1000.0));
    speedLeft = constrain(speedLeft, -200, 200);
    speedRight = 200; // Full speed right wheel
  }
}

```

```

else // Shade on Left?
{ // Slow down right wheel
  speedRight = int(200.0 + (ndShade * 1000.0));
  speedRight = constrain(speedRight, -200, 200);
  speedLeft = 200; // Full speed left wheel
}

Serial.print(speedLeft, DEC); // Display speedLeft
Serial.print(" "); // Spaces
Serial.print(ndShade, DEC); // Display ndShade
Serial.print(" "); // More spaces
Serial.println(speedRight, DEC); // Display speedRight

delay(1000); // 1 second delay
}

long rcTime(int pin) // rcTime measures decay at pin
{
  pinMode(pin, OUTPUT); // Charge capacitor
  digitalWrite(pin, HIGH); // ..by setting pin output-high
  delay(5); // ..for 5 ms
  pinMode(pin, INPUT); // Set pin to input
  digitalWrite(pin, LOW); // ..with no pullup
  long time = micros(); // Mark the time
  while(digitalRead(pin)); // Wait for voltage < threshold
  time = micros() - time; // Calculate decay time
  return time; // Returns decay time
}

```

2.12.6 Πλοήγηση του BOE Shield-Bot με φως

Το πρόγραμμα LightSeekingDisplay για να μπορέσει να πραγματοποιεί τις ενέργειες αντί να της εμφανίζει, πρέπει να αφαιρέσουμε τις κλήσεις της Serial.print, να προσθέσουμε τον κώδικα των servo, να προσθέσουμε τις συναρτήσεις maneuver (ελιγμών) και να προσθέσουμε μια κλήση στη συνάρτηση loop για να περνά η speedLeft και η speedRight στη συνάρτηση maneuver.

```

/*
 * Robotics with the BOE Shield - LightSeekingShieldBot
 * Roams toward light and away from shade.
 */

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone
}

```

```

servoLeft.attach(13);           // Attach left signal to pin 13
servoRight.attach(12);         // Attach right signal to pin 12
)

void loop()                     // Main loop auto-repeats
{
  float tLeft = float(rcTime(8)); // Get left light & make float
  float tRight = float(rcTime(6)); // Get right light & make float

  float ndShade;                // Normalized differential shade
  ndShade = tRight / (tLeft+tRight) - 0.5; // Calculate it and
  subtract 0.5

  int speedLeft, speedRight;     // Declare speed variables

  if (ndShade > 0.0)             // Shade on right?
  {                               // Slow down left wheel
    speedLeft = int(200.0 - (ndShade * 1000.0));
    speedLeft = constrain(speedLeft, -200, 200);
    speedRight = 200;           // Full speed right wheel
  }
  else                           // Shade on Left?
  {                               // Slow down right wheel
    speedRight = int(200.0 + (ndShade * 1000.0));
    speedRight = constrain(speedRight, -200, 200);
    speedLeft = 200;           // Full speed left wheel
  }

  maneuver(speedLeft, speedRight, 20); // Set wheel speeds
)

long rcTime(int pin)            // rcTime measures decay at pin
{
  pinMode(pin, OUTPUT);         // Charge capacitor
  digitalWrite(pin, HIGH);      // ..by setting pin output-high
  delay(5);                     // ..for 5 ms
  pinMode(pin, INPUT);          // Set pin to input
  digitalWrite(pin, LOW);       // ..with no pullup
  long time = micros();          // Mark the time
  while(digitalRead(pin));      // Wait for voltage < threshold
  time = micros() - time;       // Calculate decay time
  return time;                  // Returns decay time
}

// maneuver function
void maneuver(int speedLeft, int speedRight, int msTime)
{
  servoLeft.writeMicroseconds(1500 + speedLeft); // Set left servo speed
  servoRight.writeMicroseconds(1500 - speedRight); // Set right servo
  speed
  if(msTime==-1)                // if msTime = -1
  {
    servoLeft.detach();         // Stop servo signals
    servoRight.detach();
  }
  delay(msTime);                // Delay for msTime
)

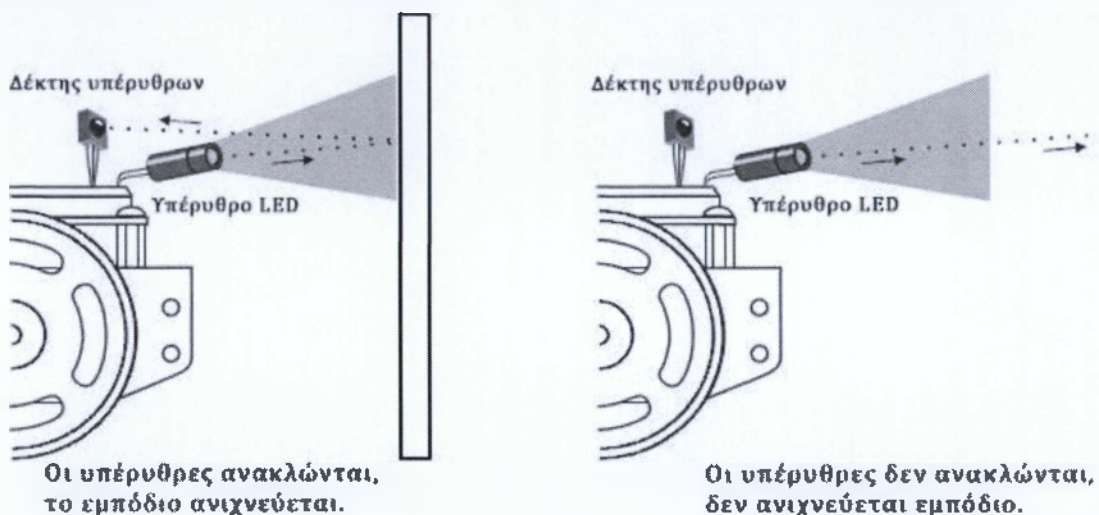
```

Αν θέλουμε περισσότερη ευαισθησία στο φως, μπορούμε στις παρακάτω εντολές να αλλάξουμε την τιμή 1000 σε μια μεγαλύτερη:

```
speedLeft = int(200.0 - (ndShade * 1000.0));  
speedRight = int(200.0 + (ndShade * 1000.0));
```

2.13 Πλοήγηση με Υπέρυθρους Προβολείς

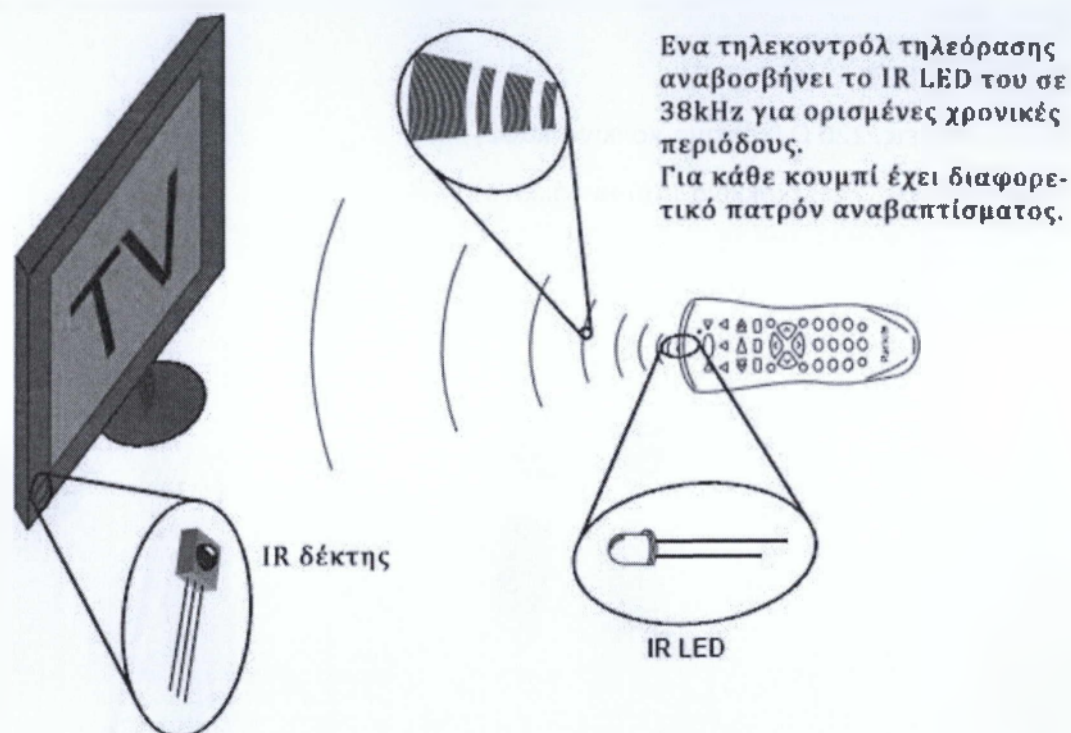
Το BOE Shield-Bot μπορεί να χρησιμοποιεί τα whiskers για να ανιχνεύει, αλλά μόνο τα εμπόδια στα οποία πέφτει πάνω τους. Όμως θα ήταν καλύτερα το ρομπότ μας να μπορούσε απλά να "δει" τα αντικείμενα και στη συνέχεια να αποφασίσει τι θα κάνει με αυτά. Λοιπόν, αυτό το επιτυγχάνει με τους υπέρυθρους προβολείς και μάτια που παρουσιάζονται παρακάτω. Κάθε προβολέας είναι ένα υπέρυθρο LED μέσα σε ένα σωλήνα που κατευθύνει το φως προς τα εμπρός, όπως ακριβώς σε ένα φακό. Κάθε μάτι είναι ένας δέκτης υπέρυθρων που στέλνει στον Arduino ψηλά και χαμηλά σήματα για να δηλώσει αν ανιχνεύει το φως του υπέρυθρου LED που αντανακλάται σε ένα αντικείμενο.



2.13.1 Υπέρυθρα Σήματα

Οι δέκτες υπέρυθρων που χρησιμοποιούμε ανιχνεύουν το υπέρυθρο φως (συντομογραφία IR) παρόμοια με τα φωτοτρανζίστορ που είδαμε προηγουμένως. Όμως, υπάρχει μια διαφορά, αυτοί οι δέκτες έχουν σχεδιαστεί να ανιχνεύουν το υπέρυθρο φως που αναβοσβήνει πολύ γρήγορα.

Το υπέρυθρο LED, που χρησιμοποιούμε ως προβολέα, είναι στην πραγματικότητα το ίδιο με αυτό που μπορούμε να βρούμε σε οποιοδήποτε τηλεχειριστήριο τηλεόρασης. Το τηλεχειριστήριο αναβοσβήνει το IR LED για να στείλει μηνύματα στην τηλεόρασή και ο μικροελεγκτής στην τηλεόρασή συλλέγει αυτά τα μηνύματα με ένα δέκτη υπέρυθρων, όπως αυτό που χρησιμοποιούμε.



Το τηλεχειριστήριο της τηλεόρασης στέλνει μηνύματα αναβοσβήνοντας το IR LED πολύ γρήγορα, σε ποσοστό περίπου 38 kHz (περίπου 38.000 φορές το δευτερόλεπτο). Ο δέκτης IR ανταποκρίνεται μόνο σε υπέρυθρες που αναβοσβήνουν με αυτό το ρυθμό. Αυτό εμποδίζει άλλες πηγές, όπως ο ήλιος και τα φώτα πυρακτώσεως, να παρερμηνεύονται ως μηνύματα από το τηλεχειριστήριο. Οπότε, ο Arduino θα πρέπει να αναβοσβήνει το IR LED με 38 kHz, για να στείλει μηνύματα που ο δέκτης IR μπορεί να ανιχνεύσει.

2.13.2 Κατασκευή και έλεγχος των ανιχνευτών

Κατάλογος εξαρτημάτων:

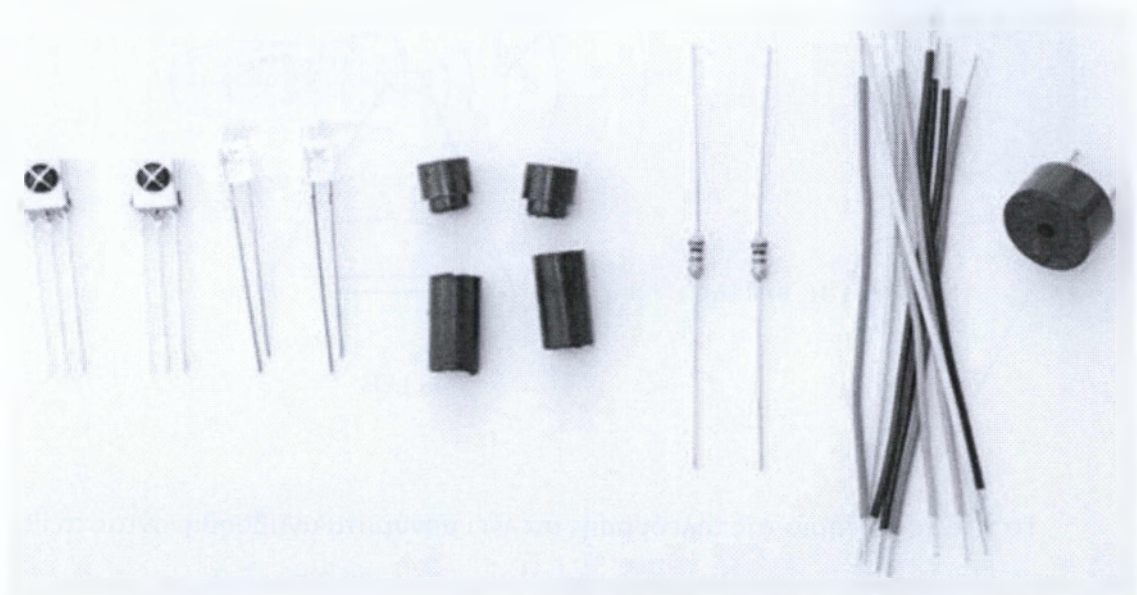
(2) Δέκτες IR

(2) IR LEDs

(2) Θήκες IR LED

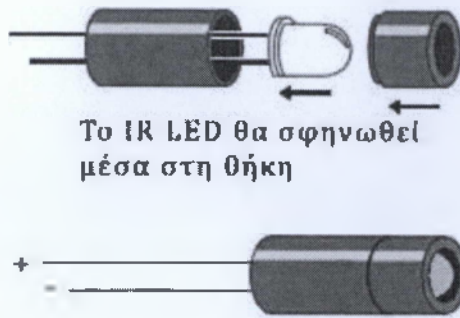
(2) Αντιστάσεις, 220 Ω (κόκκινο-κόκκινο-καφέ)

(2) Αντιστάσεις, 2 kΩ (κόκκινο-μαύρο-κόκκινο)

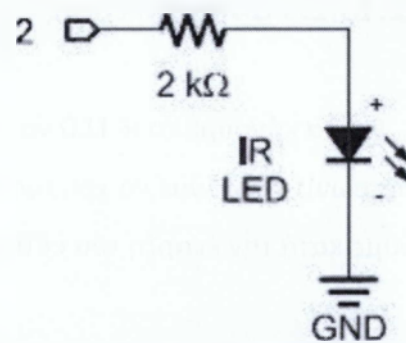
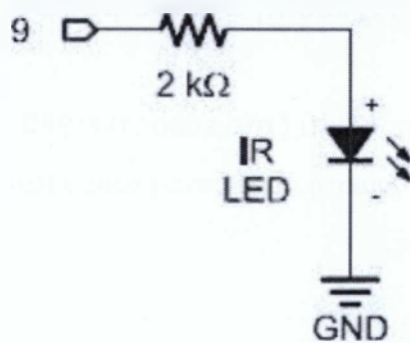
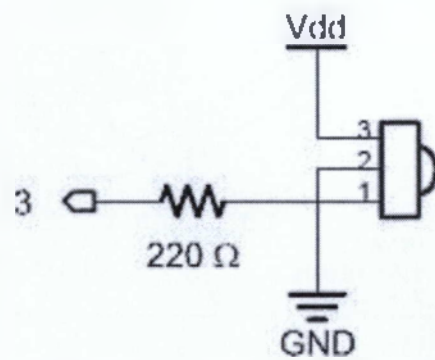
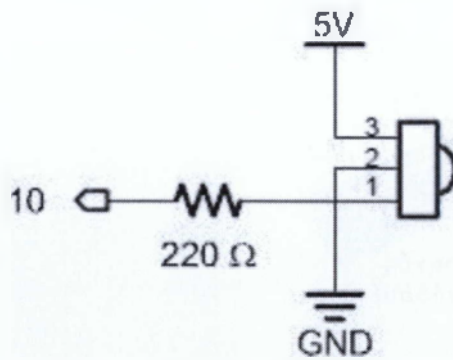


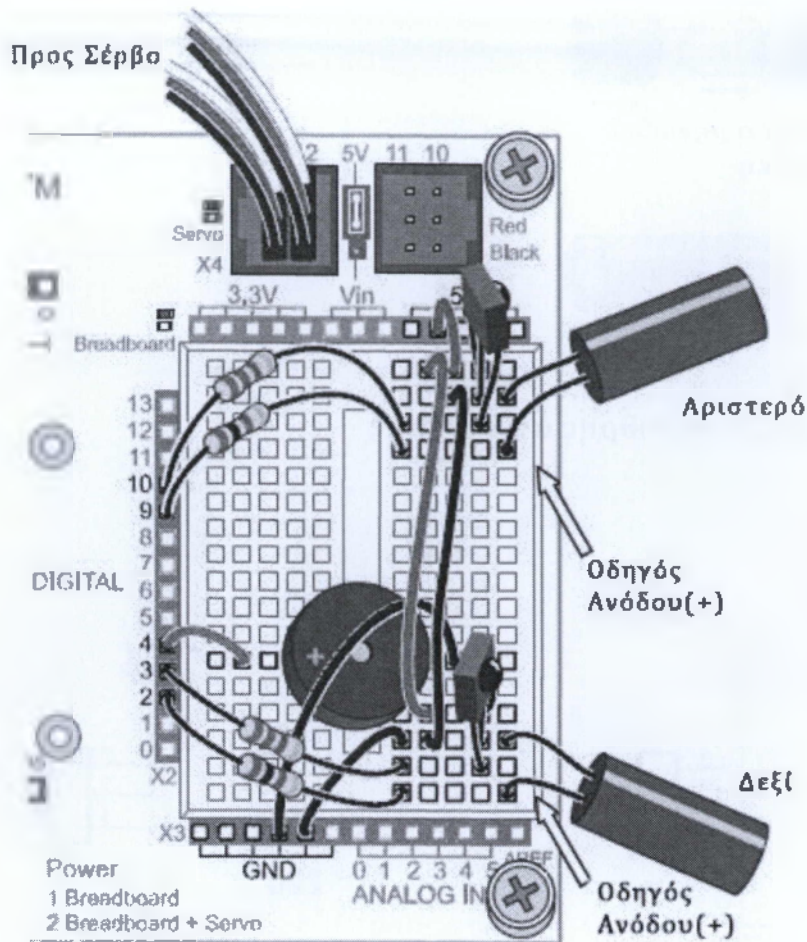
2.13.2.1 Συναρμολόγηση των IR προβολέων

Τοποθετούμε το υπέρυθρο LED στην LED θήκη (το μεγαλύτερο από τους δύο σωλήνες) και βεβαιωνόμαστε ότι το IR LED κουμπώνει μέσα στη θήκη όπως φαίνεται παρακάτω. Έπειτα κουμπώνουμε το μικρό σωλήνα, που ταιριάζει ακριβώς, πάνω από τη πλαστική θήκη του IR LED.



2.13.2.2 Κύκλωμα υπέρυθρης ανίχνευσης





2.13.2.3 Δοκιμαστικός κώδικας για ανίχνευση αντικειμένων

Για να κάνουμε το IR LED να αναβοσβήνει με 38 kHz (στο εύρος των 980 μανομέτρων), μπορούμε να χρησιμοποιήσουμε τη γνωστή συνάρτηση τόνου (tone) που βλέπουμε κατά την έναρξη του κάθε προγράμματος.

Η Υπέρυθρη ανίχνευση υλοποιείται σε τρία βήματα:

1. Αναβοσβήνει το IR LED σε κλίμακα 38 kHz.
2. Καθυστερεί για ένα χιλιοστό του δευτερολέπτου ή περισσότερο για να δώσει χρόνο στο IR δέκτη να στείλει ένα χαμηλό σήμα ως απάντηση στο υπέρυθρο φως που ανακλάται σε ένα αντικείμενο.
3. Ελέγχει την κατάσταση του IR δέκτη είτε για ένα υψηλό σήμα (δεν ανιχνεύονται IR), ή για ένα χαμηλό σήμα (ανιχνεύονται IR).

Στο παρακάτω παράδειγμα παρουσιάζονται τα τρία βήματα που εφαρμόζονται στο αριστερό IR LED (pin 9) και του IR δέκτη (pin 10).

```
tone(9, 38000, 8);           // IRLED 38 kHz for at least 1 ms
delay(1);                   // Wait 1 ms
int ir = digitalRead(10);    // IR receiver -> ir variable
```

Η συνάρτηση `tone` παράγει ένα τετραγωνικό κύμα που επαναλαμβάνει τον υψηλό και χαμηλό κύκλο του 38000 φορές ανά δευτερόλεπτο, κάνοντας την επεξεργασία του στο παρασκήνιο. Δεδομένου ότι ο IR δέκτης χρειάζεται μερικά δέκατα του χιλιοστού του δευτερολέπτου για να ανταποκριθεί στο σήμα των 38 kHz, η `delay(1)` αποτρέπει την κλήση της `ir = digitalRead(10)` από την αντιγραφή της εξόδου του IR δέκτη μέχρι να είναι έτοιμο.

Το επόμενο πρόγραμμα ορίζει μια συνάρτηση που ονομάζεται `irDetect` με τρεις παραμέτρους: μια για τη δήλωση του IR LED pin, μια για να οριστεί το pin του IR δέκτη, και μια για να ρυθμίζει τη συχνότητα που αναβοσβήνει το IR LED.

```
int irDetect(int irLedPin, int irReceiverPin, long frequency)
```

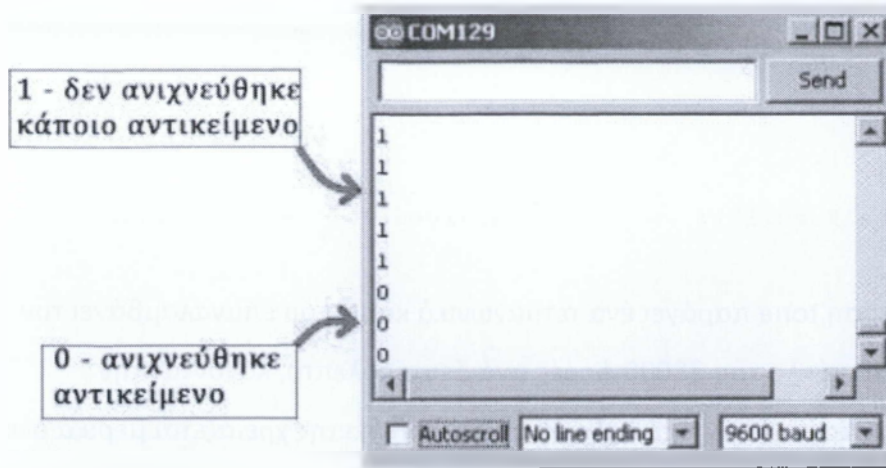
Στη συνάρτηση `loop` θα δούμε μια κλήση για την `irDetect`:

```
int irLeft = irDetect(9, 10, 38000);           // Check for object
```

Αυτή η κλήση περνά 9 στην παράμετρο `irLedPin`, 10 έως `irReceiverPin`, και 38000 στην παράμετρο συχνότητας (`frequency`). Η συνάρτηση εκτελεί αυτά τα τρία βήματα για την υπέρυθρη ανίχνευση και επιστρέφει 1 αν δεν έχει ανιχνευτεί κάποιο αντικείμενο, ή 0 εάν έχει εντοπιστεί. Έπειτα, η τιμή επιστροφής αποθηκεύονται στο `irLeft`.

2.13.2.4 Παράδειγμα προγράμματος - TestLeftIr

Το πρόγραμμα δοκιμάζει μόνο τον αριστερό IR δέκτη του ρομπότ, με τον ακριβώς ίδιο τρόπο μπορούμε να δοκιμάσουμε και τον δεξιό δέκτη.



```

/*
 * Robotics with the BOE Shield - TestLeftIR
 * Display 1 if the left IR detector does not detect an object,
 * or 0 if it does.
 */

void setup()                                     // Built-in initialization
block
{
  tone(4, 3000, 1000);                          // Play tone for 1 second
  delay(1000);                                  // Delay to finish tone

  pinMode(10, INPUT);  pinMode(9, OUTPUT);      // Left IR LED & Receiver
  Serial.begin(9600);                            // Set data rate to 9600 bps
}

void loop()                                     // Main loop auto-repeats
{
  int irLeft = irDetect(9, 10, 38000);          // Check for object

  Serial.println(irLeft);                       // Display 1/0 no detect/detect

  delay(100);                                   // 0.1 second delay
}

// IR Object Detection Function

int irDetect(int irLedPin, int irReceiverPin, long frequency)
{
  tone(irLedPin, frequency, 8);                // IRLED 38 kHz for at least 1 ms
  delay(1);                                    // Wait 1 ms
  int ir = digitalRead(irReceiverPin);         // IR receiver -> ir variable
  delay(1);                                    // Down time before recheck
  return ir;                                   // Return 1 no detect, 0 detect
}

```

2.13.3 Δοκιμή στο χώρο

Επειδή δεν είναι χρήσιμο να είμαστε κοντά σε ένα PC για να αντιμετωπίσουμε τυχόν προβλήματα στα κυκλώματα του ανιχνευτή υπέρυθρων, θα προσθέσουμε LED που θα μας αναδεικνύουν την ανίχνευση ενός αντικείμενου χωρίς τη βοήθεια του Serial Monitor.

Επίσης, πρέπει να αποτρέψουμε τυχόν παρεμβολές των υπέρυθρων από τα φώτα φθορισμού. Ορισμένα φθορίζοντα φώτα στέλνουν σήματα που μοιάζουν με το σήμα που στέλνεται από τα υπέρυθρα LED. Η συσκευή μέσα σε ένα φωτιστικό φθορισμού που ελέγχει την τάση για την λάμπα ονομάζεται ballast (έρμα). Μερικά από αυτά τα ballast λειτουργούν στο ίδιο εύρος συχνοτήτων του ανιχνευτή IR, προκαλώντας την λάμπα να εκπέμπει ένα υπέρυθρο σήμα 38,5 kHz. Όταν χρησιμοποιούμε την IR ανίχνευση αντικειμένων για την πλοήγηση, η παρεμβολή ballast (έρματος) μπορεί να προκαλέσει περίεργες συμπεριφορές στο ρομπότ, οπότε, παρακάτω θα δούμε ένα πρόγραμμα που αντιμετωπίζει αυτό το πρόβλημα.

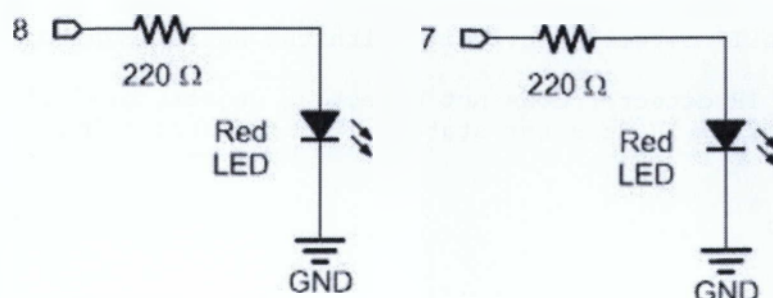
2.13.3.1 Εισαγωγή ενδεικτικών LED κυκλωμάτων

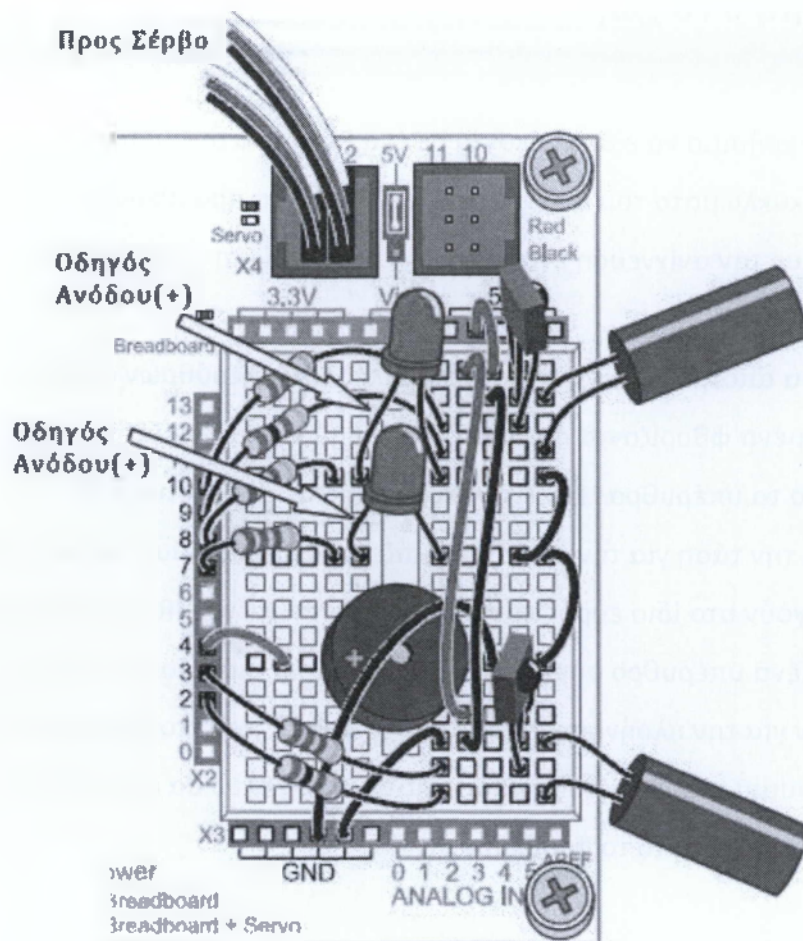
Τα κυκλώματα των ενδεικτικών λυχνιών LED είναι παρόμοια με αυτά που χρησιμοποιήσαμε στα whiskers.

Κατάλογος εξαρτημάτων:

(2) κόκκινα LEDs

(2) Αντιστάσεις, 220 Ω (κόκκινο-κόκκινο-καφέ)





2.13.3.2 Παράδειγμα προγράμματος - TestBothIrAndIndicators

Υπάρχουν αρκετά στοιχεία που εμπλέκονται σε αυτό το σύστημα, τα οποία αυξάνουν την πιθανότητα σφάλματος καλωδίωσης. Γι' αυτό είναι σημαντικό να έχουμε ένα τεστ πρόγραμμα που να μας δείχνει τι ανιχνεύουν οι υπέρυθροι ανιχνευτές, χωρίς να αποσυνδέσουμε το καλώδιο προγραμματισμού από το PC.

```

/*
 * Robotics with the BOE Shield - TestBothIrAndInciators
 * Test both IR detection circuits with the Serial Monitor. Displays 1
if
 * the left IR detector does not detect an object, or 0 if it does.
 * Also displays IR detector states with indicator LEDs.
 */

void setup() // Built-in initialization
block
{
  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone
}

```



```

pinMode(10, INPUT); pinMode(9, OUTPUT); // Left IR LED & Receiver
pinMode(3, INPUT); pinMode(2, OUTPUT); // Right IR LED & Receiver
pinMode(8, OUTPUT); pinMode(7, OUTPUT); // Indicator LEDs

Serial.begin(9600); // Set data rate to 9600 bps
}

void loop() // Main loop auto-repeats
{
int irLeft = irDetect(9, 10, 38000); // Check for object on left
int irRight = irDetect(2, 3, 38000); // Check for object on right

digitalWrite(8, !irLeft); // LED states opposite of IR
digitalWrite(7, !irRight);

Serial.print(irLeft); // Display 1/0 no detect/detect
Serial.print(" "); // Display 1/0 no detect/detect
Serial.println(irRight); // Display 1/0 no detect/detect

delay(100); // 0.1 second delay
}

// IR Object Detection Function

int irDetect(int irLedPin, int irReceiverPin, long frequency)
{
tone(irLedPin, frequency, 8); // IRLED 38 kHz for at least 1 ms
delay(1); // Wait 1 ms
int ir = digitalRead(irReceiverPin); // IR receiver -> ir variable
delay(1); // Down time before recheck
return ir; // Return 1 no detect, 0 detect
}

```

Μετά από τις δύο κλήσεις της συνάρτησης `irDetect`, η `irLeft` και η `irRight` αποθηκεύουν 1 αν δεν έχει εντοπιστεί κάποιο αντικείμενο, ή 0 αν έχει εντοπιστεί. Στο πρόγραμμα θα μπορούσε να χρησιμοποιηθεί ένα `if ... else` για να αναβοσβήνουν τα LED ανάλογα με τις τιμές των `irLeft` και `irRight`. Αλλά εδώ χρησιμοποιούμε μια άλλη προσέγγιση, που χρησιμοποιεί άμεσα τις τιμές των `irLeft` και `irRight`. Όταν η `irLeft` αποθηκεύει 0, θέλουμε το LED της να ανάβει και όταν αποθηκεύει 1, θέλουμε το LED να απενεργοποιείται. Εφόσον η τιμή 1 κάνει τα LED να ανάβουν και η 0 να σβήνουν, χρησιμοποιώντας τη `digitalWrite(8, irLeft)` θα μας έδινε το αντίθετο αποτέλεσμα από αυτό που θέλαμε. Έτσι, χρησιμοποιώντας τον τελεστή «!» αναστρέφουμε τη τιμή των `irLeft` και `irRight` ώστε όταν αποθηκεύουν μηδέν, η λυχνία LED να ανάβει, και όταν αποθηκεύουν 1, να σβήνει.

2.13.3 Ανίχνευση για υπέρυθρες παρεμβολές

Η ιδέα πίσω από αυτό το πρόγραμμα είναι πολύ απλή. Δοκιμάζουμε το περιβάλλον στο οποίο θα κινείται το ρομπότ με τους IR δέκτες και περιμένουμε να δούμε αν θα υπάρξει κάποια παρεμβολή (ανίχνευση IR). Κάθε φορά που το BOE Shield-Bot ανιχνεύει υπέρυθρες ακτίνες, θα παίζει έναν ήχο, θα ενεργοποιεί τα LED και θα εμφανίζει μια προειδοποίηση στο Serial Monitor.

```

/*
 * Robotics with the BOE Shield - IrInterferenceSniffer
 * Test for external sources of infrared interference.  If IR
interference is
 * detected: Serial Monitor displays warning, piezospeaker plays alarm,
and
 * indicator lights flash.
 */

void setup()                                // Built-in initialization
block
{
  tone(4, 3000, 1000);                       // Play tone for 1 second
  delay(1000);                               // Delay to finish tone

  pinMode(10, INPUT);                        // Left IR Receiver
  pinMode(3, INPUT);                         // Right IR Receiver
  pinMode(8, OUTPUT);                        // Left indicator LED
  pinMode(7, OUTPUT);                        // Right indicator LED

  Serial.begin(9600);                         // Set data rate to 9600 bps
}

void loop()                                  // Main loop auto-repeats
{
  int irLeft = digitalRead(10);              // Check for IR on left
  int irRight = digitalRead(3);              // Check for IR on right

  if((irLeft == 0) || (irRight == 0))        // If left OR right detects
  {
    Serial.println("IR interference!!!");     // Display warning
    for(int i = 0; i < 5; i++)               // Repeat 5 times
    {
      digitalWrite(7, HIGH);                 // Turn indicator LEDs on
      digitalWrite(8, HIGH);
      tone(4, 4000, 10);                     // Sound alarm tone
      delay(20);                             // 10 ms tone, 10 between tones
      digitalWrite(7, LOW);                  // Turn indicator LEDs off
      digitalWrite(8, LOW);
    }
  }
}
}

```

2.13.4 Ρυθμίσεις στο εύρος ανίχνευσης

Μπορούμε να αυξήσουμε την εμβέλεια ανίχνευσης του ρομπότ, αυξάνοντας τη φωτεινότητα των υπέρυθρων LED. Μια μικρότερη αντίσταση επιτρέπει περισσότερο ρεύμα να ρέει μέσα από ένα LED, με αποτέλεσμα να λάμπει πιο έντονα. Για να ξεχωρίσουμε την αντίσταση που χρειαζόμαστε, εξετάζουμε την επίδραση διαφορετικών τιμών αντίστασης τόσο στα κόκκινα όσο και στα υπέρυθρα LED.

Κατάλογος εξαρτημάτων:

- (2) Αντιστάσεις, 470 Ω (κίτρινο-ιώδες-καφέ)
- (2) Αντιστάσεις, 220 Ω (κόκκινο-κόκκινο-καφέ)
- (2) αντιστάσεις, 1 kΩ (καφέ-μαύρο-κόκκινο)
- (2) Αντιστάσεις, 4,7 kΩ (κίτρινο-βιολετί-κόκκινο)

Αρχικά, χρησιμοποιούμε ένα από τα κόκκινα LEDs για να δούμε τη διαφορά φωτεινότητας των LED ανάλογα με την αντίσταση. Το μόνο πράγμα που χρειάζεται για να δοκιμάσουμε το LED, είναι το παρακάτω πρόγραμμα που στέλνει ένα υψηλό σήμα στο LED.

```
// Robotics with the BOE Shield - LeftLedOn
// Turn on left LED for brightness testing

void setup()                               // Built-in initialization
block
{
  tone(4, 3000, 1000);                       // Play tone for 1 second
  delay(1000);                               // Delay to finish tone

  pinMode(8, OUTPUT);                       // Left indicator LED
  digitalWrite(8, HIGH);
}

void loop()                                 // Main loop auto-repeats
{
}
```

Αφού δοκιμάζουμε όλες τις αντιστάσεις που έχουμε με τα κόκκινα LEDs, αντιλαμβανόμαστε ότι μικρότερες αντιστάσεις κάνουν τα LED να λάμπουν περισσότερο. Μια λογική υπόθεση θα ήταν ότι φωτεινότερα IR LEDs μπορεί να καταστούν δυνατή την

ανίχνευση αντικειμένων που βρίσκονται μακρύτερα. Οπότε ανοίγουμε πάλι το πρόγραμμα `TestBothIrAndIndicators` και δοκιμάζουμε όλες τις αντιστάσεις που έχουμε. Οι συσχέτιση των αντιστάσεων με την απόσταση της ανίχνευσης παρουσιάζονται στον παρακάτω πίνακα.

5. Απόσταση Ανίχνευσης των Αντιστάσεων	
Αντιστάσεις IR LED (Ω)	Μεγίστη Απόσταση Ανίχνευσης (m)
4700	0,5
2000	1,2
1000	2,4
470	5
220	10

2.13.5 Ανίχνευση αντικειμένων και Αποφυγή

Το ενδιαφέρον, με τους ανιχνευτές IR, είναι ότι τα αποτελέσματα (output) τους είναι ακριβώς όπως των whiskers. Όταν δεν ανιχνεύεται αντικείμενο, η έξοδος (output) είναι υψηλή (high) και όταν ανιχνεύουν, η έξοδος είναι χαμηλή (low). Οπότε, τροποποιούμε το πρόγραμμα `RoamingWithWhiskers` έτσι ώστε να λειτουργεί με τους ανιχνευτές IR. Τα βήματα είναι τα εξής:

1. Προσθέτουμε τη συνάρτηση `irDetect`.

```
int irDetect(int irLedPin, int irReceiverPin, long frequency)
{
  tone(irLedPin, frequency, 8);
  delay(1);
  int ir = digitalRead(irReceiverPin);
  delay(1);
  return ir;
}
```

2. Αντικαταστήσουμε τις `digitalRead` κλήσεις:

```
byte wLeft = digitalRead(5);
byte wRight = digitalRead(7);
```

... Με τις κλήσεις προς τη `irDetect`:

```
int irLeft = irDetect(9, 10, 38000);
int irRight = irDetect(2, 3, 38000);
```

3. Αντικαταστήσουμε όλες τις περιπτώσεις των `wLeft` και `wRight` με `irLeft` και `irRight` αντίστοιχα.

2.13.5.1 Παράδειγμα προγράμματος - `RoamingWithIr`

```
/*
 * Robotics with the BOE Shield - RoamingWithIr
 * Adaptation of RoamingWithWhiskers with IR object detection instead of
 * contact switches.
 */

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  pinMode(10, INPUT); pinMode(9, OUTPUT); // Left IR LED & Receiver
  pinMode(3, INPUT); pinMode(2, OUTPUT); // Right IR LED & Receiver

  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12
}

void loop() // Main loop auto-repeats
{
  int irLeft = irDetect(9, 10, 38000); // Check for object on left
  int irRight = irDetect(2, 3, 38000); // Check for object on right

  if((irLeft == 0) && (irRight == 0)) // If both sides detect
  {
    backward(1000); // Back up 1 second
    turnLeft(800); // Turn left about 120 degrees
  }
  else if(irLeft == 0) // If only left side detects
  {
    backward(1000); // Back up 1 second
    turnRight(400); // Turn right about 60 degrees
  }
}
```

```

    }
    else if(irRight == 0) // If only right side detects
    {
        backward(1000); // Back up 1 second
        turnLeft(400); // Turn left about 60 degrees
    }
    else // Otherwise, no IR detected
    {
        forward(20); // Forward 1/50 of a second
    }
}

int irDetect(int irLedPin, int irReceiverPin, long frequency)
{
    tone(irLedPin, frequency, 8); // IRLED 38 kHz for at least 1 ms
    delay(1); // Wait 1 ms
    int ir = digitalRead(irReceiverPin); // IR receiver -> ir variable
    delay(1); // Down time before recheck
    return ir; // Return 1 no detect, 0
}

void forward(int time) // Forward function
{
    servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
    servoRight.writeMicroseconds(1300); // Right wheel clockwise
    delay(time); // Maneuver for time ms
}

void turnLeft(int time) // Left turn function
{
    servoLeft.writeMicroseconds(1300); // Left wheel clockwise
    servoRight.writeMicroseconds(1300); // Right wheel clockwise
    delay(time); // Maneuver for time ms
}

void turnRight(int time) // Right turn function
{
    servoLeft.writeMicroseconds(1700); // Left wheel counterclockwise
    servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
    delay(time); // Maneuver for time ms
}

void backward(int time) // Backward function
{
    servoLeft.writeMicroseconds(1300); // Left wheel clockwise
    servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
    delay(time); // Maneuver for time ms
}

```

2.13.6 Υπέρυθρη πλοήγηση Υψηλής Απόδοσης

Η τεχνοτροπία των προ-προγραμματισμένων ελιγμών είναι ότι καλύτερο για τα whiskers, αλλά ανώφελα αργή κατά τη χρήση των ανιχνευτών IR. Με τα whiskers το BOE Shield-Bot έπρεπε να έρθει σε επαφή και στη συνέχεια να πάει προς τα πίσω για να περιηγηθεί γύρω από τα εμπόδια. Με τις υπέρυθρες, το ρομπότ πρέπει να ανιχνεύει τα περισσότερα εμπόδια πριν έρθει σε επαφή με αυτά, και να βρίσκει μια σαφή πορεία γύρω από το εμπόδιο.

Για να επιτύχουμε κάτι τέτοιο, μπορούμε να μειώσουμε όλες τις διάρκειες των ελιγμών σε 20 ms, πράγμα που σημαίνει ότι το πρόγραμμα θα ελέγχει ξανά και ξανά για αντικείμενα σχεδόν 50 φορές ανά δευτερόλεπτο (περίπου 40 φορές ανά δευτερόλεπτο, επειδή η εκτέλεση του κώδικα ανίχνευσης IR χρειάζεται αρκετό χρόνο). Οπότε, καθώς το BOE Shield-Bot περιηγείται, θα εκτελεί μια σειρά από μικρές στροφές για να αποφεύγει εμπόδια, πριν καν τα συναντήσει, χωρίς να στρέφεται παραπάνω από ότι χρειάζεται.

2.13.6.1 Παράδειγμα προγράμματος - FastIrRoaming

```

/*
 * Robotics with the BOE Shield - FastIrRoaming
 * Adaptation of RoamingWithWhiskers with IR object detection instead of
 * contact switches
 */

#include <Servo.h> // Include servo library

Servo servoLeft; // Declare left and right servos
Servo servoRight;

void setup() // Built-in initialization block
{
  pinMode(10, INPUT); pinMode(9, OUTPUT); // Left IR LED & Receiver
  pinMode(3, INPUT); pinMode(2, OUTPUT); // Right IR LED & Receiver

  tone(4, 3000, 1000); // Play tone for 1 second
  delay(1000); // Delay to finish tone

  servoLeft.attach(13); // Attach left signal to pin 13
  servoRight.attach(12); // Attach right signal to pin 12
}

void loop() // Main loop auto-repeats
{

```

```

int irLeft = irDetect(9, 10, 38000); // Check for object on left
int irRight = irDetect(2, 3, 38000); // Check for object on right

if((irLeft == 0) && (irRight == 0)) // If both sides detect
{
  maneuver(-200, -200, 20); // Backward 20 milliseconds
}
else if(irLeft == 0) // If only left side detects
{
  maneuver(200, -200, 20); // Right for 20 ms
}
else if(irRight == 0) // If only right side detects
{
  maneuver(-200, 200, 20); // Left for 20 ms
}
else // Otherwise, no IR detects
{
  maneuver(200, 200, 20); // Forward 20 ms
}
}

int irDetect(int irLedPin, int irReceiverPin, long frequency)
{
  tone(irLedPin, frequency, 8); // IRLED 38 kHz for at least 1 ms
  delay(1); // Wait 1 ms
  int ir = digitalRead(irReceiverPin); // IR receiver -> ir variable
  delay(1); // Down time before recheck
  return ir; // Return 1 no detect, 0 detect
}

void maneuver(int speedLeft, int speedRight, int msTime)
{
  //speedLeft, speedRight ranges :Backward Linear Stop Linear Forward
  // -200 -100.....0.....100 200
  servoLeft.writeMicroseconds(1500 + speedLeft); // Set left servo speed
  servoRight.writeMicroseconds(1500 - speedRight); // Set right servo
  speed
  if(msTime===-1) // if msTime = -1
  {
    servoLeft.detach(); // Stop servo signals
    servoRight.detach();
  }
  delay(msTime); // Delay for msTime
}

```

Το παραπάνω πρόγραμμα χρησιμοποιεί μια ποιο άμεση συνάρτηση ελιγμών (maneuver), που αναμένει τρεις παραμέτρους: speedLeft, speedRight και msTime. Οι δύο παράμετροι της ταχύτητας χρησιμοποιούν το 200 για να κινηθούν με μέγιστη ταχύτητα προς τα εμπρός, το -200 για μέγιστη επιτάχυνση προς τα πίσω, και τις τιμές μεταξύ -100 και +100 για τον έλεγχο της γραμμικής ταχύτητας. Επίσης, η τιμή της msTime είναι 20, οπότε, κάθε ελιγμός εκτελείται για 20 ms πριν την επιστροφή στη συνάρτηση loop (βρόχου).


```

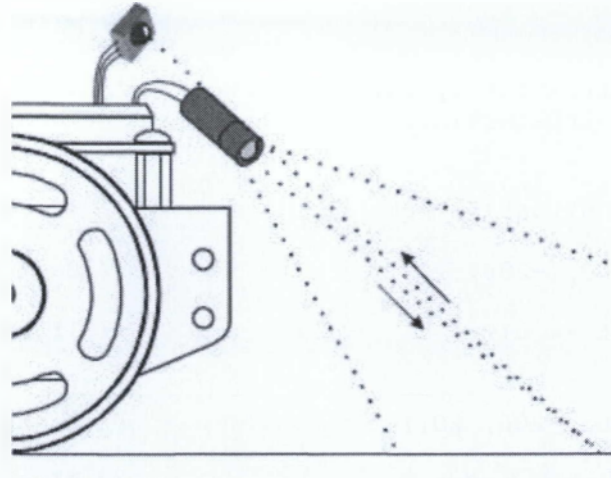
void loop() // Main loop auto-repeats
{
  int irLeft = irDetect(9, 10, 38000); // Check for object on left
  int irRight = irDetect(2, 3, 38000); // Check for object on
right
  if((irLeft == 0) && (irRight == 0)) // If both sides detect
  {
    maneuver(-200, -200, 20); // Backward 20 milliseconds
  }
  else if(irLeft == 0) // If only left side
detects
  {
    maneuver(200, -200, 20); // Right for 20 ms
  }
  else if(irRight == 0) // If only right side
detects
  {
    maneuver(-200, 200, 20); // Left for 20 ms
  }
  else // Otherwise, no IR detects
  {
    maneuver(200, 200, 20); // Backward 20 ms
  }
}

```

2.13.7 Ανίχνευση πτώσεων

Υπάρχουν επίσης εφαρμογές όπου το BOE Shield-Bot πρέπει να λάβει τα κατάλληλα μέτρα, όταν ένα αντικείμενο δεν έχει εντοπιστεί. Για παράδειγμα, εάν το BOE Shield-Bot είναι σε μια περιορισμένη περιοχή, όπως ένα τραπέζι, οι ανιχνευτές υπερύθρων πρέπει να ψάχνουν προς τα κάτω για την επιφάνεια του τραπεζιού.

Για να περιηγηθεί το ρομπότ μας γύρω από ένα τραπέζι, χωρίς να πέφτει από την άκρη, πρέπει να προσαρμόσουμε τις δηλώσεις των if ... else if ... else στο πρόγραμμα FastIrRoaming. Πρώτα απ' όλα, αντί να επιταχύνει προς τα πίσω, θα πρέπει να κινείται προς τα μπροστά, για 20 ms, όταν θα ανιχνεύει αντικείμενα. Επίσης, θα πρέπει να στρέφεται προς τα αντικείμενα και να στρίβει για περίπου 375 ms (ανάλογα την περιοχή) όταν αντιλαμβάνεται την άκρη του τραπεζιού (δηλαδή, την μη ύπαρξη των αντικειμένων).



```

/*
 * Robotics with the BOE Shield - AvoidTableEdge
 * Adaptation of FastIrRoaming for table edge avoidance
 */

#include <Servo.h>                                // Include servo library

Servo servoLeft;                                  // Declare left and right servos
Servo servoRight;

void setup()                                       // Built-in initialization block
{
  pinMode(10, INPUT); pinMode(9, OUTPUT);         // Left IR LED & Receiver
  pinMode(3, INPUT);  pinMode(2, OUTPUT);         // Right IR LED & Receiver

  tone(4, 3000, 1000);                            // Play tone for 1 second
  delay(1000);                                     // Delay to finish tone

  servoLeft.attach(13);                            // Attach left signal to pin 13
  servoRight.attach(12);                           // Attach right signal to pin 12
}

void loop()                                       // Main loop auto-repeats
{
  int irLeft = irDetect(9, 10, 38000);           // Check for object on left
  int irRight = irDetect(2, 3, 38000);           // Check for object on right

  if((irLeft == 0) && (irRight == 0))           // Both sides see table surface
  {
    maneuver(200, 200, 20);                       // Forward 20 milliseconds
  }
  else if(irLeft == 0)                            // Left OK, drop-off on right
  {
    maneuver(-200, 200, 375);                     // Left for 375 ms
  }
  else if(irRight == 0)                          // Right OK, drop-off on left
  {
    maneuver(200, -200, 375);                    // Right for 375 ms
  }
  else                                           // Drop-off straight ahead

```

```

    {
        maneuver(-200, -200, 250);           // Backward 250 ms before retry
    }
}

int irDetect(int irLedPin, int irReceiverPin, long frequency)
{
    tone(irLedPin, frequency, 8);           // IRLED 38 kHz for at least 1 ms
    delay(1);                               // Wait 1 ms
    int ir = digitalRead(irReceiverPin);     // IR receiver -> ir variable
    delay(1);                               // Down time before recheck
    return ir;                              // Return 1 no detect, 0 detect
}

void maneuver(int speedLeft, int speedRight, int msTime)
{
    //speedLeft, speedRight ranges: Backward Linear Stop Linear Forward
    //                               -200    -100.....0.....100    200
    servoLeft.writeMicroseconds(1500 + speedLeft); // Set left servo speed
    servoRight.writeMicroseconds(1500 - speedRight); // Set right servo
    speed
    if(msTime===-1)                          // if msTime = -1
    {
        servoLeft.detach();                  // Stop servo signals
        servoRight.detach();
    }
    delay(msTime);                          // Delay for msTime
}

```

Οι διαφορές στις δυο δηλώσεις if...else if...else :

<pre> // From FastIrRoaming if((irLeft == 0) && (irRight == 0)) { maneuver(-200, -200, 20); } else if(irLeft == 0) { maneuver(200, -200, 20); } else if(irRight == 0) { maneuver(-200, 200, 20); } else { maneuver(200, 200, 20); } </pre>	<pre> //From AvoidTableEdge if((irLeft == 0) &&(irRight==0)) { maneuver(200, 200, 20); } else if(irLeft == 0) { maneuver(-200, 200, 375); } else if(irRight == 0) { maneuver(200, -200, 375); } else { maneuver(-200, -200, 250); } </pre>
---	---

(robot)

Βιβλιογραφία-Αναφορές-Links

- (Ηλεκτρονικό). - <http://www.robots.ox.ac.uk/~lav/Videos/index.html>.
- 2005 Robot Nexus Ex.** [Ηλεκτρονικό]. -
<http://www.virtuar.com/click/2005/robonexus/index.htm>.
- 2013-2016** [Ηλεκτρονικό]. - <http://www.ifr.org/news/ifr-press-release/the-robotics-industry-is-looking-into-a-bright-future-551/>.
- A.count-U.c** [Ηλεκτρονικό]. -
<http://www.marketsandmarkets.com/ResearchInsight/medical-robotic-systems.asp>.
- A.def** [Ηλεκτρονικό]. - <http://www.techrepublic.com/blog/geekend/the-difference-between-robots-and-robots/>.
- A.lovelace** [Ηλεκτρονικό]. - <http://www.sdsc.edu/ScienceWomen/lovelace.html>.
- A.Magnus** [Ηλεκτρονικό]. - http://books.google.gr/books?id=25IDvKU-vGwC&pg=PA261&lpg=PA261&dq=Albertus+Magnus+man+of+brass&source=bl&ots=oAQH2rDElu&sig=7y8SjYH_08aayNpeC44tQSjpAS8&hl=el&sa=X&ei=I3R7U5_JO8v40gW9w4FY&ved=0CEkQ6AEwAg#v=onepage&q=Albertus%20Magnus%20man%20of%20brass&f=
- A.Math** [Ηλεκτρονικό]. - <http://www.mathepic.com/mathematics/automated-mathematician/>.
- A.T.1936** [Ηλεκτρονικό]. -
<http://classes.soe.ucsc.edu/cmeps210/Winter11/Papers/turing-1936.pdf>.
- aaai.treat** [Ηλεκτρονικό]. -
<http://www.nytimes.com/2009/07/26/science/26robot.html>.
- abc** [Ηλεκτρονικό]. - <http://history-computer.com/ModernComputer/Electronic/Atanasoff.html>.
- Adam** [Ηλεκτρονικό]. - <http://www.adamrobot.com/en-ca/page/home>.
- Aethon** [Ηλεκτρονικό]. - <http://www.aethon.com/tug/how-it-works/>.
- ai.sri** [Ηλεκτρονικό]. - <http://www.ai.sri.com/centibots/>.
- aibo** [Ηλεκτρονικό]. -
http://www.sony.net/SonyInfo/News/Press_Archive/199905/99-046/index.html.
- AiboR.** [Ηλεκτρονικό]. - <http://www.sony-aibo.co.uk/>.

Alb.Hubo [Ηλεκτρονικό]. -

http://www.ri.cmu.edu/pub_files/pub4/oh_jun_ho_2006_1/oh_jun_ho_2006_1.pdf.

al-jazari [Ηλεκτρονικό]. - <http://history-computer.com/Dreamers/Arabic.html>.

And.pro [Ηλεκτρονικό]. - <http://emereo.net/success/android-robot-etymology/>.

Answers [Ηλεκτρονικό]. - <http://www.answers.com/topic/robot>.

AppleII [Ηλεκτρονικό]. - <http://oldcomputers.net/appleii.html>.

Aquino Judith [Ηλεκτρονικό]. - <http://www.businessinsider.com/9-jobs-that-are-already-being-replaced-by-robots-2011-3?op=1>.

arxitas [Ηλεκτρονικό]. - <http://plato.stanford.edu/entries/archytas/#Opt>.

arxiv [Ηλεκτρονικό]. - <http://arxiv.org/ftp/arxiv/papers/1309/1309.2093.pdf>.

Asimo [Ηλεκτρονικό]. - <http://asimo.honda.com/>.

AsimoR. [Ηλεκτρονικό]. - <http://asimo.honda.com/>.

ATIS [Ηλεκτρονικό]. -

<http://web.archive.org/web/20070202054754/http://www.atis.org/tg2k/>.

AtlasR. [Ηλεκτρονικό]. - http://www.bostondynamics.com/robot_Atlas.html.

Automation Savant [Ηλεκτρονικό]. - <http://www.agvsystems.com/vehicles>.

B.P. [Ηλεκτρονικό]. - <http://www.biography.com/people/blaise-pascal-9434176#inventions-and-discoveries&awesm=~οEOSUXFpF5oulc>.

Baglole Joel [Ηλεκτρονικό]. - <http://usmilitary.about.com/od/uavs/a/mq1.htm>.

bara [Ηλεκτρονικό]. - <http://www.bara.org.uk/robots/robot-programming-methods.html>.

Bartleby [Ηλεκτρονικό]. -

<http://web.archive.org/web/20080512121943/www.bartleby.com/61/roots/IE363.html>.

BCI [Ηλεκτρονικό]. - <http://research.microsoft.com/en-us/um/people/desney/publications/bcihci-chapter1.pdf>.

binac [Ηλεκτρονικό]. - <http://edn.com/electronics-blogs/edn-moments/4398199/BINAC-gets-under-way--October-9--1947>.

Biochip [Ηλεκτρονικό]. -

http://www.nature.com/nbt/journal/v18/n10s/full/nbt1000_IT43.html.

Bletchley [Ηλεκτρονικό]. -

<http://oliverrobinson.net/photos/bletchley/index.html>.

C.Cyberman [Ηλεκτρονικό]. - http://infosources.org/what_is/Cyborg.html.

C.dance [Ηλεκτρονικό]. - <http://www.sffchronicles.co.uk/forum/4772-no-woman-born-c-l-moore.html>.

C.def. [Ηλεκτρονικό]. - http://lookup.computerlanguage.com/host_app/search?cid=C999999&term=cyborg&lookup.x=0&lookup.y=0.

C.foundation [Ηλεκτρονικό]. - <http://eyeborg.wix.com/cyborg>.

C.foundation [Ηλεκτρονικό]. - <http://eyeborg.wix.com/cyborg>.

C.medicine1 [Ηλεκτρονικό]. - <http://www.cyborgdb.org/dokko.htm>.

C.medicine2 [Ηλεκτρονικό]. - <http://www.studymode.com/essays/Cyborg-In-Medicine-1070414.html>.

C.nyctalope [Ηλεκτρονικό]. - <http://www.coolfrenchcomics.com/nyctalope.htm>.

C.Robot [Ηλεκτρονικό]. - <http://www.robotster.org/entry/predicting-future-robotics-lets-talk/>.

C.Story [Ηλεκτρονικό]. - <http://strangepegs.blogspot.gr/2012/04/a-to-z-of-fiction-to-reality-cyborgs.html>.

C.tissue [Ηλεκτρονικό]. - <http://emereo.net/success/cyborg-cyborg-tissues-in-engineering/>.

c4004 [Ηλεκτρονικό]. - <http://www.extremetech.com/computing/105029-intel-4004-the-first-cpu-is-40-years-old-today>.

Canadarm2 [Ηλεκτρονικό]. - http://www.nasa.gov/mission_pages/station/structure/elements/mss.html.

chess [Ηλεκτρονικό]. - <http://chessprogramming.wikispaces.com/Deep+Thought>.

chip [Ηλεκτρονικό]. - http://www.nobelprize.org/educational/physics/integrated_circuit/history/.

cknife [Ηλεκτρονικό]. - <http://stanfordhospital.org/clinicsmedServices/COE/cyberknife/>.

c-leg [Ηλεκτρονικό]. - http://www.ottobock.com/cps/rde/xbcrr/ob_cn_zh/im_646a231_gb_product_line_c-leg_prothetics.pdf.

Clynes.Kline [Ηλεκτρονικό]. - <http://www.stopthecrime.net/info/0099.html>.

cm [Ηλεκτρονικό]. - <http://longnow.org/essays/richard-feynman-connection-machine/>.

CMStewart [Ηλεκτρονικό]. - <http://www.singularityweblog.com/our-grey-goo-future-possibility-and-probability/>.

Colossus [Ηλεκτρονικό]. - <http://history-computer.com/ModernComputer/Electronic/Colossus.html>.

cornell [Ηλεκτρονικό]. - http://creativemachines.cornell.edu/self_replication.

Coro [Ηλεκτρονικό]. - <http://coro.etsmtl.ca/blog/?p=107>.

Corporatewatch [Ηλεκτρονικό]. - <http://www.corporatewatch.org/?lid=579>.

cyc [Ηλεκτρονικό]. - <http://www.cyc.com/company>.

D.bugs [Ηλεκτρονικό]. - <http://www.washingtontimes.com/news/2006/mar/13/20060313-120147-9229r/>.

D.mems [Ηλεκτρονικό]. - <http://web.archive.org/web/20110210141306/http://www.darpa.mil/mto/programs/himems/>.

D.shark [Ηλεκτρονικό]. - <http://www.livescience.com/603-military-plans-cyborg-sharks.html>.

D.torrent [Ηλεκτρονικό]. - <http://www.hollywoodreporter.com/news/sundance-2013-rafel-duran-torrent-414593>.

D.Vinci [Ηλεκτρονικό]. - <http://www.davincisurgery.com/>.

Da vinci [Ηλεκτρονικό] // http://biomed.brown.edu/Courses/BI108/BI108_2005_Groups/04/davinci.html. - <http://www.davincisurgery.com/da-vinci-general-surgery/da-vinci-surgery/>.

dande-dante2 [Ηλεκτρονικό]. - <http://www.nytimes.com/1994/07/24/us/robot-set-to-explore-a-volcano-that-s-too-dangerous-for-humans-to-enter.html>

<http://cyberneticzoo.com/walking-machines/1992-4-dante-dante-ii-john-e-bares-william-red-whittaker-american/>

<http://www.thetech.org/exhibits/online/robotics/universal/page06.html>. -

<http://www.nytimes.com/1992/12/08/science/robot-named-dante-to-explore-inferno-of-antarctic-volcano.html>.

Darpa2004 [Ηλεκτρονικό]. - <http://edition.cnn.com/2004/TECH/ptech/03/14/darpa.race/>.

Darpa2005 [Ηλεκτρονικό]. - <http://robots.stanford.edu/papers/thrun.stanley05.pdf>.

Dblue [Ηλεκτρονικό]. - <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/>.

DBS [Ηλεκτρονικό]. - http://www.ninds.nih.gov/disorders/deep_brain_stimulation/deep_brain_stimulation.htm.

dda [Ηλεκτρονικό]. - <http://diva.library.cmu.edu/Kanade/kanadearm.html>.

Dendral [Ηλεκτρονικό]. - <http://deepblue.lib.umich.edu/bitstream/handle/2027.42/30758/0000409.pdf?sequence=1>.

DER-02 [Ηλεκτρονικό]. - <http://pinktentacle.com/2006/10/actroid-der2-fembot-loves-hello-kitty/>.

Deyle Travis [Ηλεκτρονικό]. - <http://www.hizook.com/blog/2009/11/08/open-hardware-micro-robot-swarm-project>.

Dvorsky George [Ηλεκτρονικό]. - <http://io9.com/5932880/how-utility-fogs-could-become-the-technology-that-changes-the-world>.

Dynamics General [Ηλεκτρονικό]. - <http://www.gdrs.com/robotics/programs/program.asp?UniqueID=27>.

EATR [Ηλεκτρονικό]. - <http://unnamedharald.hubpages.com/hub/Military-Robots-That-Hunt-For-Food>.

edvac [Ηλεκτρονικό]. - <http://lecture.eingang.org/edvac.html>.

EdwardF. [Ηλεκτρονικό]. - <http://www.computerhistory.org/fellowawards/hall/bios/Edward,Feigenbaum/>.

engine Analytical [Ηλεκτρονικό]. - Charles Babbage – Life of Charles Babbage and the Analytical Engine.

Eniac [Ηλεκτρονικό]. - <http://3gym-n-ionias.att.sch.gr/inform/history/eniac1.htm>.

EstOnlan [Ηλεκτρονικό]. - <http://encc.ee/nanotechnology/nanorobotics/>.

EtcGroup [Ηλεκτρονικό]. - <http://www.etcgroup.org/issues/corporate-monopolies>.

EveR-1 [Ηλεκτρονικό]. - http://www.swisstalents.org/enews/documents/20066/ST_NewsK_May_2006.pdf.

Famulus [Ηλεκτρονικό]. - <http://www.kuka-robotics.com/usa/en/company/group/milestones/1973.htm>.

Fennecfoxtech [Ηλεκτρονικό]. -

<http://fennecfoxtech.com/docs/EskortaV1%20Robot%20user%20case.pdf>.

Feynman Richard P. [Ηλεκτρονικό]. -

<http://www.its.caltech.edu/~feynman/plenty.html>.

Floppy.disk [Ηλεκτρονικό]. - [http://history-](http://history-computer.com/ModernComputer/Basis/floppy_disk.html)

[computer.com/ModernComputer/Basis/floppy_disk.html](http://history-computer.com/ModernComputer/Basis/floppy_disk.html).

Foxconn [Ηλεκτρονικό]. - [http://www.scmp.com/business/china-](http://www.scmp.com/business/china-business/article/1473311/it-will-be-some-time-robots-will-replace-foxconn-workers)

[business/article/1473311/it-will-be-some-time-robots-will-replace-foxconn-workers](http://www.scmp.com/business/china-business/article/1473311/it-will-be-some-time-robots-will-replace-foxconn-workers).

Fox-news [Ηλεκτρονικό]. - [http://www.foxnews.com/story/2006/03/07/writer-](http://www.foxnews.com/story/2006/03/07/writer-margaret-atwood-unveils-long-distance-pen/)

[margaret-atwood-unveils-long-distance-pen/](http://www.foxnews.com/story/2006/03/07/writer-margaret-atwood-unveils-long-distance-pen/).

frankenstein [Ηλεκτρονικό]. - [http://members.aon.at/frankenstein/frankenstein-](http://members.aon.at/frankenstein/frankenstein-novel.htm)

[novel.htm](http://members.aon.at/frankenstein/frankenstein-novel.htm).

Freddy [Ηλεκτρονικό]. - <http://www.aiai.ed.ac.uk/project/freddy/>.

G.Boole [Ηλεκτρονικό]. - <http://plato.stanford.edu/entries/boole/>.

G.H. [Ηλεκτρονικό]. - <http://www.spacedaily.com/news/uav-01d.html>.

G.P. [Ηλεκτρονικό]. - [http://history-](http://history-computer.com/MechanicalCalculators/18thCentury/Poleni.html)

[computer.com/MechanicalCalculators/18thCentury/Poleni.html](http://history-computer.com/MechanicalCalculators/18thCentury/Poleni.html).

Gakutensoku [Ηλεκτρονικό]. - <http://www.nippon.com/en/views/b00906/>.

Garco [Ηλεκτρονικό]. - [http://cyberneticzoo.com/robots/1953-garco-harvey-g-](http://cyberneticzoo.com/robots/1953-garco-harvey-g-chapman-jr-american/)

[chapman-jr-american/](http://cyberneticzoo.com/robots/1953-garco-harvey-g-chapman-jr-american/).

genghis [Ηλεκτρονικό]. - <http://groups.csail.mit.edu/lbr/genghis/>.

gnt [Ηλεκτρονικό]. - <http://www.gktoday.in/nubots/>.

golem prague [Ηλεκτρονικό]. - <http://www.prague.net/golem>.

googlecars [Ηλεκτρονικό]. - [http://techcrunch.com/2012/08/07/google-cars-](http://techcrunch.com/2012/08/07/google-cars-300000-miles-without-accident/)

[300000-miles-without-accident/](http://techcrunch.com/2012/08/07/google-cars-300000-miles-without-accident/).

Greece TheAncientWeb [Ηλεκτρονικό]. -

<https://theancientweb.wordpress.com/?s=%CE%A0%CF%85%CE%B3%CE%BC%CE%B1%CE%BB%CE%AF%CF%89%CE%BD%CE%B1+>.

greekhistory [Ηλεκτρονικό]. -

<http://www.agiotatos.gr/greekhistory/mythology/2009-07-21-17-20-20.html>.

grumman northrop [Ηλεκτρονικό]. - [http://www.nanotech-](http://www.nanotech-now.com/news.cgi?story_id=27839)

[now.com/news.cgi?story_id=27839](http://www.nanotech-now.com/news.cgi?story_id=27839).

H.Dreyfuss [Ηλεκτρονικό]. - <http://mitpress.mit.edu/books/what-computers-still-cant-do>.

H.G. [Ηλεκτρονικό]. - <http://knowledge.wharton.upenn.edu/article/is-there-a-robot-in-your-future-helen-greiner-thinks-so/>.

H.Lieber [Ηλεκτρονικό]. - <http://news.harvard.edu/gazette/story/2012/08/merging-the-biological-electronic/>.

hal9000 [Ηλεκτρονικό]. - <http://www.imdb.com/title/tt0062622/>.

Hardesty Larry [Ηλεκτρονικό]. - <http://web.mit.edu/newsoffice/2013/simple-scheme-for-self-assembling-robots-1004.html>.

Harris Tom howstuffworks [Ηλεκτρονικό]. - <http://science.howstuffworks.com/robot.htm>.

HCI [Ηλεκτρονικό]. - <http://dl.acm.org/citation.cfm?id=1568348>.

hellasmythology [Ηλεκτρονικό]. - <https://sites.google.com/site/hellasmythology/heroes/kadmos>.

Hi-Mems [Ηλεκτρονικό]. - <http://backcountryvoices.wordpress.com/2013/11/23/darpa-hybrid-insect-mems-hi-mems/>.

hindawi [Ηλεκτρονικό]. - <http://www.hindawi.com/journals/jr/2011/794251/>.

Hopsi [Ηλεκτρονικό]. - <http://www.gizmag.com/panasonic-hospi-r-delivery-robot/29565/>.

I.Asimov [Ηλεκτρονικό]. - <http://www.singularitysymposium.com/laws-of-robotics.html>.

ibm [Ηλεκτρονικό]. - http://www-03.ibm.com/ibm/history/exhibits/701/701_intro.html.

IBM-f [Ηλεκτρονικό]. - <http://www-03.ibm.com/press/us/en/pressrelease/21294.wss>.

ibmpc [Ηλεκτρονικό]. - http://www-03.ibm.com/ibm/history/exhibits/pc/pc_1.html.

IFR press release [Ηλεκτρονικό]. - <http://www.ifr.org/news/ifr-press-release/the-continuing-success-story-of-industrial-robots-414/>.

innoveteus [Ηλεκτρονικό]. - <http://www.innovateus.net/science/what-are-nanorobots>.

- iso 8373** [Ηλεκτρονικό]. - <http://www.ifr.org/industrial-robots/>.
- isobot** [Ηλεκτρονικό]. - <http://www.isobotrobot.com/eng/>.
- Itti Christian Siagian* Chin-Kai Chang* Laurent** [Ηλεκτρονικό]. - http://ilab.usc.edu/publications/doc/Siagian_etal13icra.pdf.
- J.D.** [Ηλεκτρονικό]. - <http://www.darkroastedblend.com/2010/12/amazing-automatons-victorian-robots.html>.
- J.deV.** [Ηλεκτρονικό]. - <http://history-computer.com/Dreamers/Vaucanson.html>.
- J.M.** [Ηλεκτρονικό]. - http://www.robotiksystem.com/robotics_history.html.
- J.Ray** [Ηλεκτρονικό]. - <http://discovermagazine.com/2008/oct/26-rise-of-the-cyborgs>.
- J.Sull.** [Ηλεκτρονικό]. - <http://www.ric.org/research/accomplishments/Bionic/>.
- J.Weizenbaum** [Ηλεκτρονικό]. - <http://tech.mit.edu/V128/N12/weizenbaum.html>.
- James Hannah, Associated Press** [Ηλεκτρονικό]. - http://usatoday30.usatoday.com/news/nation/2007-03-30-military-robots_N.htm.
- Jeanne Dietsch Stuart Rich,** [Ηλεκτρονικό]. - <http://www.automatedbuildings.com/news/feb09/articles/mobile/090125024654mobile.htm>.
- jens** [Ηλεκτρονικό]. - <http://www.mnn.com/leaderboard/stories/7-real-life-human-cyborgs>.
- Jerry** [Ηλεκτρονικό]. - <https://nfb.org/images/nfb/publications/bm/bm00/bm0011/bm001107.htm>.
- Jetro** [Ηλεκτρονικό]. - http://www.jetro.go.jp/en/reports/market/pdf/2006_10_c.pdf.
- Jones Josh** [Ηλεκτρονικό]. - http://www.openculture.com/2012/10/isaac_asimov_explains_his_three_laws_of_robotics.html.
- Jr. Robert A. Freitas** [Ηλεκτρονικό]. - <http://www.rfreitas.com/>.
- K.Warwick** [Ηλεκτρονικό]. - <http://www.kevinwarwick.com/cyborg2.htm>.
- Killer.R.** [Ηλεκτρονικό]. - <http://www.wired.com/2009/08/robot-three-way-portends-autonomous-future/>
<http://www.dailytech.com/New%20Navyfunded%20Report%20Warns%20of%20War%2>

ORobots%20Going%20Terminator/article14298.htm

<http://www.engadget.com/2009/02/18/navy-report-warns-of-robot-uprising-suggests-a-strong-moral-com/>. - <http://news.bbc.co.uk/2/hi/technology/8182003.stm>.

KRM [Ηλεκτρονικό]. - <http://www.kurzweiltech.com/kcp.html>.

ktisivios [Ηλεκτρονικό]. -

<http://www.mlahanas.de/Greeks/HeronAlexandria2.htm>.

Kubo Angela Erika [Ηλεκτρονικό]. - <http://thediomat.com/2014/01/south-korean-scientists-develop-cancer-treating-nanorobots/>.

L.daV. [Ηλεκτρονικό]. - <http://www.allonrobots.com/leonardo-da-vinci.html>.

L.X.P [Ηλεκτρονικό]. - <http://www.googlelunarprize.org/>.

Labor U.S. Department of [Ηλεκτρονικό]. -

https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html#app_iv:4_1.

lawbc [Ηλεκτρονικό]. - <http://nanotech.lawbc.com/articles/legalregulatory-issues/>.

lawbc [Ηλεκτρονικό]. - <http://nanotech.lawbc.com/articles/legalregulatory-issues/>.

LEO [Ηλεκτρονικό]. - <http://www.kzwp.com/lyons/leo.htm>.

lisp [Ηλεκτρονικό]. - <http://www-formal.stanford.edu/jmc/history/lisp/lisp.html>.

Lob.C. [Ηλεκτρονικό]. - <http://www.adstuck.com/blog/understanding-cyborgcybernetic-organism-the-future-of-ar-technology/>.

logo [Ηλεκτρονικό]. - <http://www.papert.org/>.

Lokuge Yasith Kanchana [Ηλεκτρονικό]. -

<http://www.challenge.toradex.com/projects/10125-indoor-navigation-framework-for-mapping-and-localization-of-multiple-robotic-wheelchairs>.

lunokhod1 [Ηλεκτρονικό]. - <http://www.astronautix.com/craft/lunokhod.htm>.

M Abhilash [Ηλεκτρονικό]. - <http://www.ijpbs.net/51.pdf>.

M.B. [Ηλεκτρονικό]. - <http://www.marshallbrain.com/robotic-nation.htm>.

M.De.Landa [Ηλεκτρονικό]. - <http://www.egs.edu/faculty/manuel-de-landa/biography/>.

M.humanitarianR. [Ηλεκτρονικό]. -

http://www.mitsubishielectric.com/company/csr/back_issues/pdf/2012/philanthropic_activities.pdf.

MacHack [Ηλεκτρονικό]. - <https://chessprogramming.wikispaces.com/Mac+Hack>.

maniac1 [Ηλεκτρονικό]. - <http://chessprogramming.wikispaces.com/MANIAC+I>.

MarkI [Ηλεκτρονικό]. - http://www.ieeeighn.org/wiki/index.php/Harvard_Mark_I.

MCKesson [Ηλεκτρονικό]. -

http://www.mckesson.com/uploadedfiles/mckessoncom/content/pharmacies/_body_components/_right_rails/robot-rxoverviewbrochure.pdf.

Member, IEEE [Ηλεκτρονικό]. - <http://robo.mech.gifu-u.ac.jp/~endo/pdf/11th.pdf>.

mems D.Cyborg-hi [Ηλεκτρονικό]. - <http://www.technovelgy.com/ct/Science-Fiction-News.asp?NewsNum=1420>.

Merriam-webster [Ηλεκτρονικό]. - <http://www.merriam-webster.com/dictionary/robot>.

Metropolis [Ηλεκτρονικό]. - <http://www.imdb.com/title/tt0017136/>.

Mike Topping Jane Smith [Ηλεκτρονικό]. -

http://www.techknowlogia.org/TKL_Articles/PDF/430.pdf.

military.factory [Ηλεκτρονικό]. -

http://www.militaryfactory.com/aircraft/detail.asp?aircraft_id=46.

Mit.N.O [Ηλεκτρονικό]. - <http://newsoffice.mit.edu/2013/simple-scheme-for-self-assembling-robots-1004>.

Mobilerobots [Ηλεκτρονικό]. -

<http://www.mobilerobots.com/ResearchRobots/ResearchPatrolBot.aspx>.

molecularassembler [Ηλεκτρονικό]. -

<http://www.molecularassembler.com/Nanofactory/>.

molecularassembler.c [Ηλεκτρονικό]. -

<http://www.molecularassembler.com/Nanofactory/Challenges.htm>.

Montag [Ηλεκτρονικό]. -

http://www.globalsmt.de/index.php?option=com_content&view=article&id=4572&Itemid=122.

Moore [Ηλεκτρονικό]. - <http://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>.

mosi [Ηλεκτρονικό]. -

<http://www.mosi.org.uk/media/34368825/ferranti%20mark%20i%20computer.pdf>.

Motion Controls Robotics [Ηλεκτρονικό]. -

http://www.motioncontrolsrobotics.com/tech_talk_article.cfm?a=29&title=Unraveling-Degrees-of-Freedom-and-Robot-Axis:-What-does-it-mean-to-have-a-multiple-axis-pick-and-place-or-multiple-axis-robot.

Mule Spinning [Ηλεκτρονικό]. -

<http://www.saburchill.com/history/chapters/IR/010.html>.

Murray Peter [Ηλεκτρονικό]. - <http://singularityhub.com/2012/06/03/meet-robot-rx-the-robot-pharmacist-doling-out-350-million-doses-per-year/>.

mycin [Ηλεκτρονικό]. -

<http://www.britannica.com/EBchecked/topic/37146/artificial-intelligence-AI/219101/MYCIN>.

N.asimo [Ηλεκτρονικό]. - <http://world.honda.com/news/2005/c051213.html>.

N.C.mit-harvard [Ηλεκτρονικό]. -

<http://wyss.harvard.edu/viewpressrelease/75/researchers-at-harvards-wyss-institute-develop-dna-nanorobot-to-trigger-targeted-therapeutic-responses>.

N.cancer [Ηλεκτρονικό]. - <http://www.ibtimes.com/cancer-fighting-robot-korean-scientists-develop-nanorobots-are-more-efficient-chemotherapy-video>.

N.dia. [Ηλεκτρονικό]. - <http://www.nanorobotdesign.com/article/diabetes/>.

N.Harbisson [Ηλεκτρονικό]. - <http://www.dezeen.com/2013/11/20/interview-with-human-cyborg-neil-harbisson/>.

N.ra [Ηλεκτρονικό]. -

<http://www.transhumanism.org/index.php/WTA/more/227/>.

N.RtissueC. [Ηλεκτρονικό]. - <http://gen-nano.blogspot.gr/2012/09/nanorobotics.html>.

N.Wiener [Ηλεκτρονικό]. -

<http://www.britannica.com/EBchecked/topic/643306/Norbert-Wiener>.

Nano.gov [Ηλεκτρονικό]. - <http://www.nano.gov/nanotech-101/what/definition>.

Nano.R. [Ηλεκτρονικό]. - <http://www.popsci.com/technology/article/2013-09/minute-nanobots-carry-medicine-and-cells?dom=PSC&loc=recent&lnk=1&con=these-magnetic-nanobots-could-carry-drugs-into-your-brain>.

Nanoforum [Ηλεκτρονικό]. -

<http://nanotechforu1.blogspot.gr/2009/12/nanorobotics-theory.html>.

nanotech-now [Ηλεκτρονικό]. - [http://www.nanotech-](http://www.nanotech-now.com/news.cgi?story_id=35023)

[now.com/news.cgi?story_id=35023](http://www.nanotech-now.com/news.cgi?story_id=35023).

nasa.gov [Ηλεκτρονικό]. - <http://mars.jpl.nasa.gov/MPF/rover/sojourner.html>.

Nasa.gov [Ηλεκτρονικό]. -

<http://www.nasa.gov/centers/jpl/education/spaceprobe-20100225.html>.

Nasa.gov.luna9 [Ηλεκτρονικό]. -

<http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1966-006A>.

nasa.space swarm [Ηλεκτρονικό]. -

http://www.nasa.gov/offices/oct/home/feature_cosmos.html#.T7KGYvxmxwo.facebook.

Natural.B.C. [Ηλεκτρονικό]. -

[http://it.mesce.ac.in/downloads/CriticalPerspectives/booksforreview%20CPT%20S7/Cla rk%20E.%20Natural-Born%20Cyborgs-%20Minds,%20Technologies,%20and%20the%20Future%20of%20Human%20Intelligenc e.pdf](http://it.mesce.ac.in/downloads/CriticalPerspectives/booksforreview%20CPT%20S7/Cla rk%20E.%20Natural-Born%20Cyborgs-%20Minds,%20Technologies,%20and%20the%20Future%20of%20Human%20Intelligence.pdf).

Neal Ryan W. [Ηλεκτρονικό]. - <http://www.ibtimes.com/cancer-fighting-robot-korean-scientists-develop-nanorobots-are-more-efficient-chemotherapy-video>.

Needham Joseph 53 sellida , 80 selida pdf [Ηλεκτρονικό]. -

http://monoskop.org/images/e/e4/Needham_Joseph_Science_and_Civilisation_in_China_Vol_2_History_of_Scientific_Thought.pdf.

Neobotix [Ηλεκτρονικό]. - <http://www.neobotix-robots.com/transport-system-mt-400.html>.

Nist.gov [Ηλεκτρονικό]. -

http://www.nist.gov/public_affairs/factsheet/2009robocup.cfm.

Nistep [Ηλεκτρονικό]. -

<http://www.nistep.go.jp/achiev/ftx/eng/rep071e/idx071e.html>.

Nubots [Ηλεκτρονικό]. - <http://www.spacedaily.com/news/spacemedicine-05f.html>.

O.android [Ηλεκτρονικό]. -

<http://www.oxforddictionaries.com/definition/english/android>.

O.Cyborg [Ηλεκτρονικό]. - <http://cyberneticzoo.com/wp-content/uploads/2012/01/cyborgs-Astronautics-sep1960.pdf>.

Online Etymology Dictionary [Ηλεκτρονικό]. - http://www.etymonline.com/index.php?allowed_in_frame=0&search=robot&searchmode=or.

osha [Ηλεκτρονικό]. - https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html.

Oxford Dictionaries oxforddictionaries [Ηλεκτρονικό]. - http://www.oxforddictionaries.com/definition/english/robotics#m_en_gb0714530.

P.J.Kincheloe [Ηλεκτρονικό]. - <http://journal.media-culture.org.au/index.php/mcjournal/article/viewArticle/254>.

P.k.Dick [Ηλεκτρονικό]. - <http://www.pkdandroid.org/>.

p3 [Ηλεκτρονικό]. - <http://world.honda.com/ASIMO/P3/>.

P3 [Ηλεκτρονικό]. - <http://www.plasticpals.com/?p=25341>.

pcb-experts [Ηλεκτρονικό]. - <http://www.pcb-experts.com/PCB-mass-production.html>.

Pearce Fred [Ηλεκτρονικό]. - http://www.newscientist.com/article/dn24822-japans-ageing-population-could-actually-be-good-news.html#.UwEe_oWPqHs.

PEARCE JEREMY [Ηλεκτρονικό]. - http://www.nytimes.com/2011/08/16/business/george-devol-developer-of-robot-arm-dies-at-99.html?_r=0.

Plug&Play [Ηλεκτρονικό]. - <http://www.plugandpray-film.de/en/content.html>.

politikh [Ηλεκτρονικό]. - <http://myweb.brooklyn.liu.edu/mcuonzo/AristotleSlavery.htm>.

progr. [Ηλεκτρονικό]. - <http://museum.ipsj.or.jp/en/computer/other/0002.html>.

prolog [Ηλεκτρονικό]. - http://www.ehow.com/facts_7176214_history-prolog-programming-language.html.

Prospector [Ηλεκτρονικό]. - <http://www.sri.com/work/publications/prospector-computer-based-consultation-system-mineral-exploration>.

Qureshi Aqeel [Ηλεκτρονικό]. - <http://globalaccessibilitynews.com/2012/03/16/german-researchers-develop-assistant-robot-for-people-with-disabilities/>.

R.2 [Ηλεκτρονικό]. -

<http://ir.uiowa.edu/cgi/viewcontent.cgi?article=1850&context=mff>.

R.A. [Ηλεκτρονικό]. - <http://www.history.co.uk/biographies/richard-arkwright>.

R.bacon [Ηλεκτρονικό]. -

http://www.themodernword.com/joyce/joyce_intro.html.

R.dis. [Ηλεκτρονικό]. -

<https://disneyland.disney.go.com/attractions/disneyland/disneyland-story/>.

R.H.saya [Ηλεκτρονικό]. - <http://cdn.intechopen.com/pdfs-wm/19461.pdf>.

R.implants [Ηλεκτρονικό]. -

http://ophthalmology.stanford.edu/research/basic_retinal_prosthesis.html.

R.Lucy [Ηλεκτρονικό]. - http://www.scienceagogo.com/news/cyber_life.shtml.

R.N.implants [Ηλεκτρονικό]. -

<http://www.theguardian.com/science/2013/feb/17/race-to-create-insect-cyborgs>.

R.Rights [Ηλεκτρονικό]. - <http://www.bioethics.ac.uk/events/robots---rights--will-artificial-intelligence-change-the-meaning-of-human-rights-.php>.

R.vanW. [Ηλεκτρονικό]. - <http://goprojects.blogspot.gr/2013/10/advancements-in-robots.html>.

racter [Ηλεκτρονικό]. - <http://www.ubu.com/historical/racter/index.html>.

Raj.R. [Ηλεκτρονικό]. - <http://www.rr.cs.cmu.edu/>.

ranchoa. [Ηλεκτρονικό]. -

<http://www.computerhistory.org/timeline/?category=rai>.

RB.AF [Ηλεκτρονικό]. - <http://people.csail.mit.edu/brooks/papers/fast-cheap.pdf>.

RDS [Ηλεκτρονικό]. - <http://msdn.microsoft.com/en-us/library/bb648760.aspx>.

Roamer [Ηλεκτρονικό]. -

<http://roamerrobot.tumblr.com/post/23079345849/the-history-of-turtle-robots>.

Robinson [Ηλεκτρονικό]. -

http://en.wikipedia.org/wiki/Heath_Robinson_%28codebreaking_machine%29.

robomatrix [Ηλεκτρονικό]. -

<http://www.robotmatrix.org/RoboticSpecification.htm>.

Robonaut2 [Ηλεκτρονικό]. -

http://www.nasa.gov/mission_pages/station/main/robonaut.html#.U3o8tCijktU.

robot Boe-bot [Ηλεκτρονικό]. - <http://learn.parallax.com/node/125>.

robotiq [Ηλεκτρονικό]. - <http://blog.robotiq.com/bid/36551/Robotic-Gripper-Repeatability-Definition-and-Measurement>.

Robotland [Ηλεκτρονικό]. - http://www.robotland.or.kr/n_eng/c02/c02_04.php.

robots fundamental of [Ηλεκτρονικό]. - <http://203.115.126.36:8080/xmlui/bitstream/handle/123456789/551/fundamentals%20of%20robotics.pdf?sequence=1>.

Robots Mobile [Ηλεκτρονικό]. - <http://www.mobilerobots.com/ResearchRobots/Seekur.aspx>.

robotuna [Ηλεκτρονικό]. - <http://robotuna.wordpress.com/>.

Robotworks [Ηλεκτρονικό]. - <http://www.wired.com/2012/12/ff-robots-will-take-our-jobs/all/>.

Robotworx [Ηλεκτρονικό]. - <http://www.robots.com/faq/show/what-are-some-industrial-robot-basics>.

Robovolc System [Ηλεκτρονικό] // <http://www.nytimes.com/1992/12/08/science/robot-named-dante-to-explore-inferno-of-antarctic-volcano.html>. - <http://www.robovolc.dees.unict.it/system/system.htm>.

rocail [Ηλεκτρονικό]. - <https://archive.org/details/livesnecromance04godwgoog>.

Roomba [Ηλεκτρονικό]. - <http://www.irobot.com/us/learn/home/roomba.aspx>.

ROS [Ηλεκτρονικό]. - <http://www.ros.org/>.

rur [Ηλεκτρονικό]. - <http://www.mindfully.org/Reform/RUR-Capek-1920.htm>.

S.Cart [Ηλεκτρονικό]. - <http://www.stanford.edu/~learnest/cart.htm>.

S.Gripper [Ηλεκτρονικό]. - http://www-robot.mes.titech.ac.jp/hirose/robot/snake/sg/sg_e.html.

S.m.d.mAN [Ηλεκτρονικό]. - <http://www.imdb.com/title/tt0071054/>.

S.O.Rover [Ηλεκτρονικό]. - http://marsrovers.jpl.nasa.gov/mission/traverse_maps.html.

S.robot [Ηλεκτρονικό]. - <http://eandt.theiet.org/news/2013/sep/snake-robot-mars.cfm>.

S.Running [Ηλεκτρονικό]. - <http://www.imdb.com/title/tt0067756/>.

S.Wars [Ηλεκτρονικό]. - <http://www.imdb.com/title/tt0076759/>.

S.worry [Ηλεκτρονικό]. - http://www.nytimes.com/2009/07/26/science/26robot.html?_r=1&ref=todayspaper.

Saenz Aaron [Ηλεκτρονικό]. - <http://singularityhub.com/2010/09/09/irobots-military-swarm-of-wifi-bots-flips-into-action/>.

Scara [Ηλεκτρονικό]. - <http://www.robothalloffame.org/inductees/06inductees/scara.html>.

Scriptpro [Ηλεκτρονικό]. - http://www.scriptpro.com/uploadedFiles/Images/Products/RoboticDispensing/Robots/T-abletCapsuleDispensing/Robot_brochure.pdf.

Shakey [Ηλεκτρονικό]. - <http://www.robothalloffame.org/inductees/04inductees/shakey.html>.

shock [Ηλεκτρονικό]. - <http://www.computerhistory.org/semiconductor/timeline/1951-First.html>.

Shuttle Flying [Ηλεκτρονικό]. - <http://www.saburchill.com/history/chapters/IR/009.html>.

SILBY BRENT [Ηλεκτρονικό]. - <http://www.deflogic.com/articles/nanomachines.html>.

song su [Ηλεκτρονικό]. - http://fourriverscharter.org/projects/Inventions/pages/china_waterclock.htm.

SpeciMinder [Ηλεκτρονικό]. - <http://www.ccsrobotics.com/products/speciminder.html>.

Srover [Ηλεκτρονικό]. - http://www.nasa.gov/mission_pages/mars-pathfinder/.

Staford.arm [Ηλεκτρονικό]. - <http://infolab.stanford.edu/pub/voy/museum/pictures/display/1-Robot.htm>.

Starfish [Ηλεκτρονικό]. - http://creativemachines.cornell.edu/emergent_self_models.

statistics IFR [Ηλεκτρονικό]. - <http://www.ifr.org/service-robots/statistics/>.

Stenmark Maj [Ηλεκτρονικό]. - http://fileadmin.cs.lth.se/cs/Personal/Maj_Stenmark/isr2013_mspn.pdf.

sylvester [Ηλεκτρονικό]. - <https://archive.org/details/livesnecromance04godwgoog>.

System/360 IBM [Ηλεκτρονικό]. - <http://www.computermuseum.li/Testpage/IBM-360-1964.htm>.

T. Turing [Ηλεκτρονικό]. - <http://www.loebner.net/Prizef/TuringArticle.html>.

T.S.Arm [Ηλεκτρονικό]. - <http://www.used-robots.com/education/the-history-and-benefits-of-industrial-robots>.

Tabuchi Hiroko [Ηλεκτρονικό]. - http://www.nbcnews.com/id/23438322/ns/technology_and_science-innovation/t/japan-looks-robot-future/#.UihrET_4WVY.

Tanaka Hisashige [Ηλεκτρονικό]. - <http://www.tofugu.com/2013/07/29/tanaka-hisashige-father-of-toshiba-edison-of-japan/>.

technology Joseph A. Angelo (2007). Robotics: a reference guide to the new [Ηλεκτρονικό]. - pp. 258–327. - http://books.google.gr/books?id=73kNFV4sDx8C&pg=PA258&dq=robotics+glossary+terms&hl=en&ei=IN1CTcm1DYScgQfQoPyDAG&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q&f=false.

Templeton Graham [Ηλεκτρονικό]. - <http://www.geek.com/science/magnet-guided-nanorobots-could-change-eye-surgery-forever-1560518/>.

tentacle [Ηλεκτρονικό]. - <http://video.mit.edu/watch/ai-history-minsky-tentacle-arm-12074/>.

Tesla [Ηλεκτρονικό]. - <http://www.pbs.org/tesla/index.html>.

unimate [Ηλεκτρονικό]. - <http://www.botmag.com/index.php/the-rise-and-fall-of-unimation-inc-story-of-robotics-innovation-triumph-that-changed-the-world>.

Univac [Ηλεκτρονικό]. - <http://www.thocp.net/hardware/univac.htm>.

UNO [Ηλεκτρονικό]. - <http://www.cannxs.org/open.pdf>.

usa [Ηλεκτρονικό]. - <http://rt.com/usa/gen-cone-drone-army-050/>.

V.B. [Ηλεκτρονικό]. - <http://museum.mit.edu/150/22>.

V.de.H [Ηλεκτρονικό]. - <http://history-computer.com/Dreamers/VillarddeHonnecourt.html>.

V.Vinge [Ηλεκτρονικό]. - <http://www.singularitysymposium.com/vernor-vinge.html>.

Valente [Ηλεκτρονικό]. - <http://qix.sagepub.com/content/17/7/639>.

Voyagers.1.2 [Ηλεκτρονικό]. - <http://voyager.jpl.nasa.gov/index.html>.

vtu [Ηλεκτρονικό]. - <http://elearning.vtu.ac.in/11/enotes/ComplntManf/unit8-nan.pdf>.

wabot2 [Ηλεκτρονικό]. -

http://www.humanoid.waseda.ac.jp/booklet/kato_2.html.

Walter [Ηλεκτρονικό]. -

<http://cyberneticzoo.com/category/cyberneticanimals/grey-walter-cyberneticanimals/>.

waterclock [Ηλεκτρονικό]. - <http://www.factmonster.com/ipka/A0855491.html>

http://www-groups.dcs.st-and.ac.uk/history/HistTopics/Water_clocks.html

http://fourriverscharter.org/projects/Inventions/pages/china_waterclock.htm. -

<http://inventors.about.com/library/weekly/aa071401a.htm>.

WD-2 [Ηλεκτρονικό]. -

<http://www.takanishi.mech.waseda.ac.jp/top/research/docomo/>.

We.C. [Ηλεκτρονικό]. - [http://www.livescience.com/10317-future-cyborgs-](http://www.livescience.com/10317-future-cyborgs-walk.html)

[walk.html](http://www.livescience.com/10317-future-cyborgs-walk.html).

webb Daifuku [Ηλεκτρονικό]. - [http://www.daifukuwebb.com/products/trailer-](http://www.daifukuwebb.com/products/trailer-loading-vehicle-smartloader/)

[loading-vehicle-smartloader/](http://www.daifukuwebb.com/products/trailer-loading-vehicle-smartloader/).

Westinghouse R. [Ηλεκτρονικό]. - [http://history-](http://history-computer.com/Dreamers/Elektro.html)

[computer.com/Dreamers/Elektro.html](http://history-computer.com/Dreamers/Elektro.html).

WG.teach pendant [Ηλεκτρονικό]. - <http://www.wisegeek.com/what-is-a-teach->

[pendant.htm](http://www.wisegeek.com/what-is-a-teach-).

Whatis [Ηλεκτρονικό]. - <http://whatis.techtarget.com/definition/nanorobot>.

Whitesides [Ηλεκτρονικό]. -

<http://gmwgroup.harvard.edu/research/index.php?page=23>.

Whitesides George M. [Ηλεκτρονικό]. -

http://www.tmi.vu.lt/legacy/pfk/funkc_dariniai/nanostructures/nano_robots.htm.

wiki.robot software [Ηλεκτρονικό]. -

http://en.wikipedia.org/wiki/Robot_software#Examples_of_programming_languages_for_Industrial_Robots.

wiseGeek [Ηλεκτρονικό]. - [http://www.wisegeek.com/what-are-the-](http://www.wisegeek.com/what-are-the-applications-of-nanorobots.htm)

[applications-of-nanorobots.htm](http://www.wisegeek.com/what-are-the-applications-of-nanorobots.htm).

WR2013 [Ηλεκτρονικό]. -

http://www.worldrobotics.org/uploads/media/Executive_Summary_WR_2013.pdf.

Xcon [Ηλεκτρονικό]. -

<http://www.prenhall.com/divisions/bp/app/alter/student/useful/ch12dec.html>.

Zunt Dominik [Ηλεκτρονικό]. - <http://capek.misto.cz/english/robot.html>.

Zuse [Ηλεκτρονικό]. - <http://history-computer.com/ModernComputer/Relays/Zuse.html>.

Zuse [Ηλεκτρονικό]. - http://www.horst-zuse.homepage.t-online.de/Konrad_Zuse_index_english_html/zkg_artikel_2.html.

Zyvex [Ηλεκτρονικό]. - <http://www.zyvex.com/nanotech/selfRep.html>.

A.I.mit [Ηλεκτρονικό]. - <http://web.mit.edu/6.933/www/Fall2001/AI/aiLab.pdf>.

Z3 [Ηλεκτρονικό]. - <http://www.computermuseum.li/Testpage/Z3-Computer-1939.htm>.

IEEEghn [Ηλεκτρονικό]. - http://www.ieeeghn.org/wiki/index.php/Richard_Feynman_and_Micromachines.

M.M. [Ηλεκτρονικό]. - <http://web.media.mit.edu/~minsky/papers/frames/frames.html>.

P2 [Ηλεκτρονικό]. - <http://www.plasticpals.com/?p=25340>.

Ταλως [Ηλεκτρονικό]. - http://el.wikipedia.org/wiki/%CE%A4%CE%AC%CE%BB%CF%89%CF%82_%28%CE%BC%CF%85%CE%B8%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1%29.