

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΙΑΣ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μελέτη και Εφαρμογές Αυτοματισμού σε Ρομποτικό Βραχίονα



**ΣΠΟΥΔΑΣΤΕΣ: ΜΑΡΓΑΡΗ ΔΕΣΠΟΙΝΑ
ΡΕΜΠΗΣ ΔΗΜΗΤΡΙΟΣ**

**ΕΠΙΒΛΕΠΟΝ ΚΑΘΗΓΗΤΗΣ: ΘΕΟΧΑΡΑΤΟΣ ΧΡΗΣΤΟΣ
ΕΡΓΑΣΤΗΡΙΑΚΟΣ ΣΥΝΕΡΓΑΤΗΣ**

ΠΑΤΡΑ 2012

	<u>ΣΕΛΙΔΑ</u>
<u>ΠΡΟΛΟΓΟΣ</u>	6
<u>ΠΕΡΙΛΗΨΗ</u>	8
<u>ΕΙΣΑΓΩΓΗ</u>	10

1^ο ΚΑΦΑΛΑΙΟ

1.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	15
1.2 ΒΙΟΜΗΧΑΝΙΚΑ ΡΟΜΠΟΤ	17
1.3 ΤΟΜΕΙΣ ΕΦΑΡΜΟΓΗΣ	18

2^ο ΚΑΦΑΛΑΙΟ

2.1 ΕΙΣΑΓΩΓΗ.....	26
2.2 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΚΑΙ ΕΙΔΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ	26
2.2.1 ΒΑΘΜΟΙ ΚΙΝΗΤΙΚΟΤΗΤΑΣ ΚΑΙ ΒΑΘΜΟΙ ΕΛΕΥΘΕΡΙΑΣ	27
2.2.2 ΤΑΞΙΝΟΜΗΣΗ ΒΡΑΧΙΟΝΩΝ ΒΑΣΗ ΤΗΣ ΓΕΩΜΕΤΡΙΚΗΣ ΔΙΑΜΟΡΦΩΣΗΣ ΤΟΥΣ	28
2.3 ΡΟΜΠΟΤΙΚΟΣ ΒΡΑΧΙΟΝΑΣ LAB-VOLT	31
2.3.1 Ο ΕΛΕΓΚΤΗΣ ΡΟΜΠΟΤ	33
2.3.2 Ο ΟΠΤΙΚΟΣ ΚΩΔΙΚΟΠΟΙΗΤΗΣ	33
2.3.3 ΕΥΡΩΣ ΕΡΓΑΣΙΑΣ	34
2.3.4 ΣΥΣΚΕΥΗ ΕΠΙΔΡΑΣΗΣ	34
2.3.5 ΣΗΜΕΙΑ ΘΕΣΕΩΝ	34
2.3.6 ΣΗΜΕΙΑ ΕΛΕΓΧΟΥ	35
2.3.7 ΑΡΧΙΚΗ ΘΕΣΗ (HOME POSITION)	36
2.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	36
2.4.1 ΜΕΘΟΔΟΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	36
2.4.2 ΑΠΟ ΣΗΜΕΙΟ ΣΕ ΣΗΜΕΙΟ ΠΡΟΓΡΑΜΜΑ	37
2.4.3 ΣΤΟΙΧΕΙΩΔΗ ΠΡΟΓΡΑΜΜΑΤΑ	37
2.4.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΗΣΙΜΟΠΟΙΟΝΤΑΣ ΤΟ ΦΟΡΗΤΟ ΤΕΡΜΑΤΙΚΟ Η ΤΗ ΡΟΜΠΟΤΙΚΗ	39
2.4.5 ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ	39
2.4.6 ΤΡΟΠΟΠΟΙΗΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ	40
2.4.7 ΕΠΙΜΕΛΗΤΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	40
2.5 ΕΙΣΟΔΟΙ (INPUTS) ΚΑΙ ΕΞΟΔΟΙ (OUTPUTS)	41
2.5.1 ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ	41

3^ο ΚΕΦΑΛΑΙΟ

3.1 Περιγραφή εντολών	45
3.1.1 CAROUSEL (Μύλος)	45
3.1.2 CNCINPUT (Είσοδος cnc)	45
3.1.3 CNCOUTPUT (Έξοδος cnc)	46
3.1.4 DELAY (Καθυστέρηση)	47

3.1.5 DO (Εκτέλεσε)	47
3.1.6 DRIVETO (Πήγαινε προς)	48
3.1.7 ELSE (Αλλιώς)	49
3.1.8 END (Τέλος)	50
3.1.9 ENDIF (Τέλος εάν)	50
3.1.10 ERROR (Σφάλμα)	51
3.1.11 GOSUB (Πήγαινε υπογείως)	52
3.1.12 HOME (Σπίτι)	53
3.1.13 IF (Εάν)	53
3.1.14 INPUT (Είσοδος)	54
3.1.15 LATHE (Διατομή)	55
3.1.16 MESSAGE (Μήνυμα)	55
3.1.17 MILL	56
3.1.18 MOVETO (Πήγαινε προς)	56
3.1.19 OUTPUT (Έξοδος)	57
3.1.20 RELAY (Επιβράδυνση)	57
3.1.21 REPEAT (Επανάληψη)	58
3.1.22 RESTART (Επανεκκίνηση)	58
3.1.23 RETURN (Επιστροφή)	59
3.1.24 RUNTO (Εκτέλεσε έως)	59
3.1.25 SPEED (Ταχύτητα)	60
3.1.26 SUB (Υπό)	61
3.1.27 UNTIL (Έως)	61
3.1.28 WHILE (Μέχρι)	62
3.2 ΕΝΤΟΛΕΣ	64

4° ΚΕΦΑΛΑΙΟ

4.1 ΕΙΣΑΓΩΓΗ	66
4.2 ΤΡΟΠΟΙ ΓΡΑΦΗΣ ΤΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ	66
4.2.1 ΕΝΤΟΛΕΣ ΚΕΙΜΕΝΟΥ	66
4.2.2 ΓΡΑΦΗΜΑΤΑ ΡΟΗΣ	67
4.2.3 ΕΝΤΟΛΕΣ ΜΕ ΤΗ ΧΡΗΣΗ ΕΙΚΟΝΙΔΙΩΝ	69

5° ΚΕΦΑΛΑΙΟ

5.1 ΕΙΣΑΓΩΓΗ.....	76
5.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ..	76
5.3 ΠΡΟΓΡΑΜΜΑ ΓΡΑΜΜΩΝ ΕΝΤΟΛΩΝ ΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ..	77
5.4 ΠΡΟΓΡΑΜΜΑ ΕΙΚΟΝΙΔΙΩΝ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ	91
5.5 ΠΡΟΓΡΑΜΜΑ ΓΡΑΦΗΜΑΤΩΝ ΡΟΗΣ ΡΟΜΠΟΤΙΟΥ ΒΡΑΧΙΟΝΑ	98

<u>ΣΥΜΠΕΡΑΣΜΑΤΑ</u>	101
----------------------------------	-----

<u>ΒΙΒΛΙΟΓΡΑΦΙΑ</u>	105
----------------------------------	-----

ΠΡΟΛΟΓΟΣ

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε στο Τμήμα Μηχανολογίας του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πάτρας και αναφέρεται στην εικονική κατασκευή λειτουργίας ενός ρομποτικού παρκινγκ. Στην εργασία μας παραθέτουμε γενικές πληροφορίες για τα ρομπότ και την ιστορία τους, καθώς και τη συμβολή τους στην παγκόσμια βιομηχανία και οικονομική ανάπτυξη. Εστιάζουμε στο προγραμματισμό και τις εφαρμογές των ρομποτικών βραχιόνων μέσα από μικρές απλές εφαρμογές, αλλά και την κύρια εφαρμογή του ρομποτικού παρκινγκ. Επίσης αναφέρονται πλεονεκτήματα και μειονεκτήματα των ρομποτικών μηχανών, καθώς και του ρομποτικού παρκινγκ. Τέλος γίνεται αναφορά σε μερικές ενδεικτικές μελλοντικές εφαρμογές.

Ευχαριστούμε θερμά τον Επιβλέποντα καθηγητή μας κ. Χρήστο Θεοχαράτο για την πολύτιμη βοήθειά του και τη καθοδήγηση που μας πρόσφερε για την πραγματοποίηση αυτής της πτυχιακής εργασίας. Επίσης θα θέλαμε να ευχαριστήσουμε ιδιαίτερα τον κ. Παναγιώτη Ράπτη, Ε.Τ.Π. του Τμήματος Ηλεκτρολογίας, για την συνεισφορά του στην εκπόνηση της εργασίας, την κατάλληλη καθοδήγησή του στην αυτοματοποίηση των διαδικασιών ελέγχου του ρομποτικού βραχίονα, καθώς επίσης και στον προγραμματισμού αυτού.

Τέλος, θα θέλαμε να ευχαριστήσουμε ιδιαίτερα τον Καθηγητή κ. Απόστολο Υφαντή, Προϊστάμενο και Υπεύθυνο του Εργαστηρίου Επεξεργασίας Σημάτων και Αυτομάτου Ελέγχου, για την φιλική συνεργασία του, και την παραχώρηση του εργαστηριακού χώρου και του εξοπλισμού του εργαστηρίου με σκοπό για την εκτέλεση του πειραματικού μέρους αυτής της εργασίας

Μάργαρη Δέσποινα
Ρεμπής Δημήτριος
Νοέμβριος 2012

ПЕРІЛНΨН

ΠΕΡΙΛΗΨΗ

Η πτυχιακή αυτή εργασία αναφέρεται στην εικονική κατασκευή λειτουργίας ενός ρομποτικού πάρκινγκ. Το πρόβλημα χώρου στάθμευσης στα κέντρα των μεγάλων πόλεων είναι ένα φλέγον ζήτημα που απασχολεί παγκοσμίως τους πολιτικούς των διαφόρων χωρών. Το ρομποτικό πάρκινγκ δίνει μια σημαντική λύση σε αυτό το πρόβλημα καθώς στον ίδιο χώρο με ένα απλό συμβατικό πάρκινγκ μπορούμε να σταθμεύσουμε μεγαλύτερο αριθμό αυτοκινήτων με μεγαλύτερη ασφάλεια από κλοπές, καταστροφές αλλά και από τις καιρικές συνθήκες καθώς το ρομποτικό παρκινγκ είναι στεγασμένο και πρόσβαση στο χώρο φύλαξης των οχημάτων έχει μόνο ο ρομποτικός βραχίονας και το προσωπικό.

Στο πρώτο κεφάλαιο αναφέρεται η ιστορική εξέλιξη των ρομποτικών συστημάτων, ορισμένοι βιομηχανικοί ορισμοί για τους αυτοματισμούς και τέλος οι εφαρμογές τους.

Στο δεύτερο κεφάλαιο γίνεται αναλυτική αναφορά στους ρομποτικούς βραχίονες και αναλύονται τα μέρη που αποτελούνται καθώς και οι τρόποι προγραμματισμού τους. Επίσης αναλύεται ο ρομποτικός βραχίονας Lab-Volt με τον οποίο ασχοληθήκαμε εμείς, καθώς και διάφορες εντολές προγραμματισμού που χρησιμοποιήθηκαν.

Στο τρίτο κεφάλαιο παραθέτονται όλες οι εντολές που υπάρχουν στον προγραμματισμό του βραχίονα Lab-volt.

Στο τέταρτο κεφάλαιο επισημαίνονται οι τρόποι που μπορεί να γραφεί ένα πρόγραμμα καθώς και διάφορες μικρές απλές εφαρμογές με την βοήθεια του ρομποτικού μας βραχίονα.

Στο πέμπτο κεφάλαιο αναλύουμε το πρόγραμμα του πειράματος του ρομποτικού πάρκινγκ που εκτελέσαμε, τα βήματα που ακολουθήσαμε και χρησιμοποιώντας κατάλληλες φωτογραφίες δίνουμε και οπτική επαφή με το θέμα.

Τέλος παραθέτουμε τα συμπεράσματα που βγάλαμε από αυτή την εργασία και αναφέρουμε κάποιες εφαρμογές που μπορούν να χρησιμοποιηθούν σε ένα πραγματικό ρομποτικό πάρκινγκ.

ΕΙΣΑΓΩΓΗ

ΕΙΣΑΓΩΓΗ

Εδώ και πολλές δεκαετίες η ρομποτική ως ερευνητική πλατφόρμα και τεχνολογία εφαρμογής γνωρίζει μεγάλη άνθηση και αποδοχή. Η εποχή που διανύουμε μπορεί να θεωρηθεί ως η ωρίμανση και ενηλικίωση της και αυτό φαίνεται τόσο από τις εφαρμογές και την εξάπλωσή της, όσο και από τα ερευνητικά αποτελέσματα. Προηγμένες ερευνητικά και τεχνολογικά χώρες (Ιαπωνία, ΗΠΑ, χώρες της Δυτικής και Ανατολικής Ευρώπης) έχουν αναπτύξει και συνεχίζουν να στηρίζουν την έρευνα των αυτόματων συστημάτων. Οι λόγοι έχουν να κάνουν τόσο με τα οικονομικά οφέλη για τις εθνικές τους οικονομίες που προκύπτουν από την αυτοματοποίηση και με τον εν γένει εκσυγχρονισμό της παραγωγής, όσο και με τις προσδοκίες και προοπτικές που τροφοδοτούνται από τα αποτελέσματα σε επιστημονικό επίπεδο.

Ένας άλλος λόγος για την εξάπλωση της ρομποτικής είναι η στενά εξαρτημένη και αλληλεπιδραστική σχέση της με τεχνολογίες αιχμής όπως η Πληροφορική, η Ηλεκτρονική, τα Συστήματα Αυτόματου Ελέγχου και Τεχνητής Νοημοσύνης, η Τεχνολογία Αισθητήρων, κ.ά. η οποία λειτουργεί πολλαπλασιαστικά αλλά και πυροδοτεί εξελίξεις.

Οι ρομποτικές εφαρμογές είναι ιδιαίτερα δημοφιλείς στις ημέρες μας. Αρκετά συχνά, τα μέσα μαζικής επικοινωνίας αναφέρονται σε επιτεύγματα ερευνητικών εργαστηρίων ενώ πολυάριθμες ερευνητικές ομάδες ασχολούνται σε έργα με ιδιαίτερα υψηλές απαιτήσεις και προσδοκίες. Σχεδόν καθημερινά, γίνεται λόγος για ρομπότ που εκτελούν εργασίες στο διάστημα και ελέγχονται από τη Γη, που πραγματοποιούν υποβρύχιες εξερευνήσεις και εργασίες, που συμμετέχουν σε ιατρικές εγχειρήσεις, ή ακόμα και ρομπότ που συναναστρέφονται με ανθρώπους. Έχουν ακόμη παρουσιασθεί ρομπότ που έχουν διάφορες μορφές, είναι ανθρωποειδή, διαθέτουν όραση και ακοή, έχουν αντίληψη των μεταβολών του περιβάλλοντος τους, έχουν δάκτυλα και διαχειρίζονται άγνωστα αντικείμενα με επιδεξιότητα, κινούνται σε μη δομημένους ή σε γνωστούς χώρους, θέτουν στόχους και παίρνουν αποφάσεις, κλπ. Εξάλλου, υπάρχουν σήμερα σε παγκόσμιο επίπεδο περίπου 800.000 ρομποτικές εγκαταστάσεις, κυρίως στη βιομηχανία και ο αριθμός αυτός αυξάνεται με έναν μέσο ετήσιο ρυθμό της τάξης του 10%. Τα ρομπότ συνεπώς αποτελούν ήδη μέρος της ζωής μας και η συμμετοχή τους στα επόμενα χρόνια θα είναι όλο και μεγαλύτερη.

Μία σύντομη ιστορική αναδρομή καταδεικνύει ότι η πρόθεση δημιουργίας ανθρωπομορφικών μηχανών-υπηρετών, δηλαδή κατά κάποιον τρόπο πρωτόγονων ρομπότ, υπήρχε από την αρχαιότητα, είτε μέσα στη μυθολογία πολλών λαών, είτε μέσα από τη δημιουργία αυτόματων μηχανισμών, με ή χωρίς ανθρώπινη μορφή. Στα «ρομπότ» αυτά παρατηρούνται, τηρουμένων των αναλογιών, ομοιότητες στη σύλληψη, λειτουργία και χρήση πράγμα που δηλώνει ότι πρέπει να είχαν επιχειρηθεί κάποιες προσπάθειες.

Στις αρχές του 20ου αιώνα, η αντίληψη των ανθρώπων για τα ρομπότ, εκφράσθηκε κυρίως μέσα από την τέχνη, σε κινηματογραφικές ταινίες ή μυθιστορήματα επιστημονικής φαντασίας, τα οποία παρουσίαζαν τα ρομπότ ως δημιουργήματα ανώτερων, ακόμα και εξωγήινων πολιτισμών, τα οποία είτε βρίσκονται στην υπηρεσία του ανθρώπου και κάποια στιγμή επαναστατούν υποδουλώνοντας τον, είτε συμμετέχουν στη διακυβέρνηση την οποία ασκούν με ψυχρό και σκληρό τρόπο. Το γεγονός αυτό συνδέεται με την έκρυθμη κατάσταση του πρώτου μισού του αιώνα μας καθώς και με τις εκρηκτικές διαστάσεις των επιστημονικών και τεχνολογικών επιτευγμάτων της εποχής αυτής. Η Ηλεκτρονική, η Κυβερνητική και η Πληροφορική τροφοδότησαν τη ρομποτική καθιστώντας εφικτή την ανάπτυξη και υλοποίηση των ρομποτικών μηχανισμών με τη μορφή που έχουν σήμερα.

Η ανάπτυξη αυτή έχει σήμερα προσλάβει εντυπωσιακούς ρύπους με αποτέλεσμα η επιστήμη της ρομποτικής να έχει ξεφύγει από το στενό επιστημονικό πλαίσιο των δεκαετιών του '50 και του '60. Τα σημερινά ρομποτικά συστήματα εξυπηρετούν όχι μόνο εφαρμογές έρευνας, αλλά και βιομηχανικές, εξερευνητικές, εφαρμογές της καθημερινής ζωής. Η τρέχουσα τάση της ρομποτικής είναι η ολοκλήρωση προηγμένων αισθητηρίων συστημάτων, η προσαρμοστικότητα και η αντίληψη σε διάφορες συνθήκες περιβάλλοντος, η λήψη αποφάσεων στα περιβάλλοντα αυτά, η αυτονομία, δηλαδή η λειτουργία «κατ' εικόνα και ομοίωση» του ανθρώπου.

Η εκτεταμένη εξάπλωση της ρομποτικής έχει συνεπώς τη βάση της στην πρόθεση δημιουργίας ανθρωπόμορφων μηχανών. Η πρόθεση αυτή οδήγησε στην ολοκλήρωση και το συνδυασμό πολλών γνωστικών πεδίων, εφόσον η δημιουργία εξελιγμένων μηχανών με ικανότητες και επιδόσεις ανθρώπων απαιτεί συνδυασμένη γνώση και προσπάθεια.

Οι κύριες συνιστώσες επιστήμες που συνδράμουν και συνθέτουν τη ρομποτική έρευνα και τεχνολογία είναι οι ακόλουθες:

- Μηχανική, όσον αφορά θέματα ανάλυσης και σχεδίασης των μηχανικών τμημάτων (σύνδεσμοι, αρθρώσεις, κ.ά.), μηχανισμών (σύστημα μετάδοσης της κίνησης-γρανάζια, εργαλεία δράσης, κ.ά.) και της λειτουργίας των (έδραση, αποφυγή συντονισμών, κ.ά.). Μαθηματικά, αλγόριθμοι και αριθμητική ανάλυση για την αποδοτική υλοποίηση των σύνθετων υπολογισμών και στρατηγικών ελέγχου, τις γεωμετρικές αναπαραστάσεις, τη μαθηματική μοντελοποίηση και προσομοίωση, κ.ά.

- Συστήματα αυτομάτου ελέγχου για τη μελέτη και το σχεδιασμό του συστήματος ελέγχου του βραχίονα, βασισμένο στο μοντέλο του χειριστή με ανάδραση πληροφορίας από τους εσωτερικούς αισθητήρες.
- Τεχνητή νοημοσύνη (ή ευφυής έλεγχος) για την ενσωμάτωση εξωτερικής αισθητήριας ανατροφοδότησης, δηλαδή πληροφορίας υψηλού επιπέδου στο σύστημα ελέγχου του ρομπότ, για την επίτευξη ευφυούς και «αυτόνομης» συμπεριφοράς.
- Τεχνολογία λογισμικού και προγραμματισμός, για την κωδικοποίηση και εκτέλεση των αλγορίθμων και της διαχείρισης του ρομποτικού συστήματος, σε επίπεδο προγραμματισμού εφαρμογής (σε περιβάλλοντα ή σε γλώσσες) και λειτουργικού συστήματος.
- Ηλεκτρονική, στην υλοποίηση των προηγούμενων λειτουργιών με αναλογικά και ψηφιακά κυκλώματα, μικροεπεξεργαστές και αισθητήρια.
- Συστήματα παραγωγής για ανάλυση, σχεδιασμό και υλοποίηση ολοκληρωμένων βιομηχανικών εφαρμογών με ρομπότ.

Στη λίστα που ακολουθεί αναφέρονται τα πιο συχνά πλεονεκτήματα και μειονεκτήματα των ρομπότ, όπως έχουν καταγραφεί σε άρθρα περιοδικών και σε βιβλία.

Πλεονεκτήματα:

1. Ασφάλεια των εργαζομένων.
 - ✓ Επικίνδυνα περιβάλλοντα.
Τοξικά αέρια.
Υψηλές θερμοκρασίες.
Ραδιενέργεια
 - ✓ Επικίνδυνες εργασίες.
Φόρτωση και εκφόρτωση επικίνδυνων εργαλείων.
2. Υψηλότερη παραγωγικότητα,
24ωρη λειτουργία
Υψηλότερη ταχύτητα λειτουργίας στις περισσότερες εφαρμογές
Λιγότερα σφάλματα που έχουν σαν αποτέλεσμα λιγότερα απορριπτέα κομμάτια ή κατεστραμμένες μηχανές
Ομοιόμορφη ποιότητα των παραγομένων προϊόντων
3. Ευελιξία παραγωγικής μονάδας
Μικρότερος χρόνος προσαρμογής σε αλλαγές της μονάδας
Ευκολότερη αντιμετώπιση των αλλαγών
Αντίσταση στην παλαιώση με ενθάρρυνση ανανεωτικών αλλαγών.
Ικανότητα λειτουργίας υπό ασυνήθιστους προσανατολισμούς, όπως με προσάρτηση στην οροφή ή στον τοίχο.
4. Δυνατότητα λειτουργίας υπό αντίξοες συνθήκες.
Υποθαλάσσιες και διαστημικές έρευνες και επισκευές
Ηφαίστεια, σήραγγες, κλπ.

Μειονεκτήματα:

1. Δυσκολία προγραμματισμού ανάλογα με την εφαρμογή
2. Σχετικά μικρή ταχύτητα λειτουργίας, ειδικά για σερβοελεγχόμενα συστήματα συνεχούς τροχιάς.
3. Η ακρίβεια τοποθέτησης των ρομποτικών συνιστωσών περιορίζεται από τον «τζόγο» στις ρομποτικές αρθρώσεις και την ευκαμψία ή «κύρτωση» των ρομποτικών συνδέσμων. Όσο μεγαλύτερο είναι το φορτίο σε σχέση με το μέγεθος του ρομπότ, τόσο μεγαλύτερο είναι το πρόβλημα αυτό.
4. Κόστος εγκατάστασης, συντήρησης, εκμάθησης, κλπ.

Όλα τα παραπάνω αναδεικνύουν την γενική χρησιμότητα των ρομπότ και βασιζόμενοι σε αυτή την εξέλιξη, κοιτώντας στο μέλλον σκεφτήκαμε να σχεδιάσουμε μια εφαρμογή η οποία θα κάνει ευκολότερη την ζωή και την καθημερινότητά μας. Το ρομποτικό πάρκινγκ είναι μια εφαρμογή που δεν είναι διαδεδομένη στη χώρα μας και θα βοηθήσει στο πρόβλημα των χώρων στάθμευσης που αντιμετωπίζουμε.

ΠΡΩΤΟ ΚΕΦΑΛΑΙΟ

ΠΡΩΤΟ ΚΕΦΑΛΑΙΟ

1.1 Ιστορική αναδρομή

Αυτοματοποιημένη λέγεται μια μηχανική συσκευή η οποία μπορεί να υποκαθιστά τον άνθρωπο σε διάφορες εργασίες (ROBOT) . Ένα ρομπότ μπορεί να δράσει κάτω από τον απευθείας έλεγχο ενός ανθρώπου ή αυτόνομα κάτω από τον έλεγχο ενός προγραμματισμένου υπολογιστή.

Η λέξη ρομπότ προέρχεται από το σλαβικό *robota* που σημαίνει εργασία. Καθιερώθηκε ως όρος με την σημερινή του έννοια το 1920 από τον Τσέχο θεατρικό συγγραφέα Karel Čapek στο έργο του "R.U.R." (Rossum's Universal Robots), όπου σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ) της τεχνολογικής εξέλιξης και που τελικά εξοντώνουν τους δημιουργούς τους. Η ρομπωτική αποτελεί την εξέλιξη της τεχνολογίας των αυτόματων και αυτοματοποιημένων συστημάτων και μηχανών. Γεννήθηκε καταρχήν απτήν ανάγκη εξερεύνησης του διαστήματος και εξασφάλισης της υπεροχής του πολεμικού ανταγωνισμού μεταξύ των υπερδυνάμεων.

Από τα αρχαία χρόνια υπάρχουν αναφορές για διάφορες κατασκευές αυτοματισμού. Ο Ήφαιστος ακόμα είχε φτιάξει χρυσές γυναίκες (θεραπαινίδες) να τον βοηθάνε στο εργαστήριο, να τον στηρίζουν για να περπατάει καλύτερα, αλλά και 20 τρίποδα τραπέζια που υπηρετούσαν τους Θεούς μεταφέροντας μόνα τους ότι χρειάζονται. Επίσης ένα από τα πιο γνωστά αρχαία ρομπότ στην Ελλάδα ήταν ο διάσημος Τάλως.



Τάλως

Ο Τάλως ήταν τεράστιος ανθρωπόμορφος και χάλκινος. Προστάτευε την Κρήτη από τους εχθρούς της και επέβλεπε την εφαρμογή των νόμων. Μπορούσε να κινείται πολύ γρήγορα και ήταν σε θέση να κάνει σε μία μέρα τρεις φορές τον γύρο της Κρήτης (περίπου 250 km/h). Είχε την δύναμη να εκσφενδονίζει τεράστιους βράχους εναντίον των αντιπάλων του ή να τους καίει με την αναπνοή του που πετούσε φωτιά.

Με αυτόν τον τρόπο έδιωχνε τα εχθρικά πλοία, προστατεύοντας την Κρήτη. Υπάρχουν αναφορές για αυτοματισμούς και στην αρχαία Αίγυπτο.

Μετά τον Β΄ Παγκόσμιο πόλεμο και την έκρηξη της τεχνολογίας κατά την διάρκειά του, γίνεται η συνάντηση του πετυχημένου εφευρέτη George C. Devol και του μηχανικού Joseph F. Engelberger έτσι το πρώτο βιομηχανικό ρομπότ είναι γεγονός το Unimate εγκαθίσταται σε ένα εργοστάσιο και συνεργάζεται με μηχανήματα θερμαινόμενης χύτευσης, και εκτελεί συγκολλήσεις και οι δυο εργασίες είναι ιδιαίτερα απεχθείς θέσεις εργασίας για τον άνθρωπο. Επίσης μετά από την πετρελαϊκή κρίση το 1973, η ανάπτυξη για την βιομηχανία και τις άλλες οικονομικές δραστηριότητες ήταν ραγδαία και η συμβολή των αυτοματοποιημένων συστημάτων και μηχανών υπήρξε θετικότερη και βοήθησε στο ξεπέρασμα αυτής της οικονομικής κρίσης.

Αρχές του 1990 η NASA απέτυχε στην χρήση του Dante, ενός ρομπότ με οχτώ πόδια που είχε σαν αποστολή την συλλογή αερίων και μάγματος στην Ανταρκτική.



Dante 2

Μία βλάβη στα καλώδια δεν επέτρεψε στο ρομπότ να εισέλθει στο ενεργό ηφαίστειο. Ένα χρόνο μετά ο Dante 2 εισήλθε με επιτυχία στο ηφαίστειο φέροντας εις πέρας την αποστολή του. Το 1996, το βιομετρικό ρομπότ Robotuna δημιουργείται από τον φοιτητή David Barrett στο Massachusetts Institute of Technology, με σκοπό να μελετήσει πως κολυμπούν τα ψάρια στο νερό. Το πρώτο ανθρωπόμορφο ρομπότ, το P2 αποτέλεσε τον θεμέλιο λίθο των ερευνών της εταιρείας Honda πάνω στην δημιουργία ανδρειδών (ανθρωπόμορφα ρομπότ που θα μπορούν να μιμούνται τον άνθρωπο σε εμφάνιση και συμπεριφορά). Είχε ύψος 1.80 μέτρα και οι κινήσεις του ήταν πολύ κοντά στις

ανθρώπινες. Το 1997 συμβαίνει κάτι μοναδικό για τα τότε δεδομένα. Το σκακιστικό πρόγραμμα Deep Blue της εταιρείας λογισμικού IBM, κερδίζει τον παγκόσμιο πρωταθλητή σκακιού Gary Kasparov, στο προχωρημένο επίπεδο.

Το 2000 αποτελεί σταθμό στην επιστήμη της ρομποτικής. Η διαρκώς δραστήρια Honda, παρουσιάζει την πρώτη έκδοση του ASIMO (Advanced Step in Innovative Mobility), ενός ρομπότ που είναι σε θέση να τρέχει, να περπατά, να επικοινωνεί με τους ανθρώπους, να



Robonaut (Ρομπότ αστροναυτές)

αναγνωρίζει εκφράσεις και περιβάλλοντα και να αλληλεπιδρά με αυτά. Το ύψος του είναι 1.30 μέτρα ενώ ζυγίζει 54. Η όψη του θυμίζει έναν μίνι αστροναύτη και αποτελεί εξέλιξη παλαιότερων ρομποτικών μοντέλων της ίδιας εταιρείας. Την ίδια χρονιά η Sony παρουσιάζει και αυτή με την σειρά της ανθρωπόμορφα ρομπότ με το όνομα **Sony Dream Robots**, Σύμφωνα με τα Ηνωμένα Έθνη, τον Οκτώβριο του 2000 υπήρχαν 742.500 βιομηχανικά ρομπότ στον κόσμο, με περισσότερο από τα μισά αυτών να χρησιμοποιούνται μόνο στην Ιαπωνία. Έκτοτε, οι εξελίξεις στον χώρο των ανθρωπόμορφων ρομπότ παραμένουν στάσιμες. Με εξαίρεση την διαρκή κινητικότητα της Honda, με τα νέα μοντέλα του Asimo και το καινούριο δημιούργημα της NASA τον Robonaut, ένα τηλεκατευθυνόμενο ρομπότ-αστροναύτη που θα βοηθήσει αρκετά στην συλλογή στοιχείων από πλανήτες.

1.2 Βιομηχανικά ρομπότ-ορισμοί

Οι ρομποτικές διατάξεις ορίσθηκαν από το υπουργείο Μεγάλης Βρετανίας ως “ επαναπρογραμματιζόμενες μηχανικές χειριστικές διατάξεις”. Ο ορισμός αυτός είναι παρόμοιος με τον ορισμό του Ινστιτούτου Ρομπότ της Αμερικής, δηλαδή: “Robot είναι μια επαναπρογραμματιζόμενη πολυλειτουργική χειριστική διάταξη σχεδιασμένη για την μετακίνηση υλικών, εξαρτημάτων, εργαλείων και εξειδικευμένων διατάξεων, μέσω μεταβλητών προγραμματισμένων κινήσεων για την εκτέλεση μιας σειράς εργασιών”.

Ο ορισμός αυτός έχει γίνει ευρέως αποδεκτός. Ο Διεθνής Οργανισμός Τυποποίησης (ISO) έχει υιοθετήσει έναν παρόμοιο ορισμό: “Το βιομηχανικό robot είναι μια αυτοματοποιημένη ως προς τις κινήσεις στο χώρο, επαναπρογραμματιζόμενη, πολυλειτουργική χειριστική διάταξη, που έχει περισσότερους από έναν άξονες ικανούς να χειρίζονται υλικά, εξαρτήματα, εργαλεία ή εξειδικευμένες διατάξεις μέσω μεταβλητών προγραμματιζόμενων κινήσεων για την εκτέλεση μιας σειράς εργασιών”. Η Οικονομική Επιτροπή για την Ευρώπη (ESE), Ομάδα εργασίας σε θέματα κατασκευαστικών βιομηχανιών και αυτοματισμού, έχει επίσης αποδεχθεί τον παραπάνω ορισμό.

Ένα ρομπότ αποτελείται από δύο βασικά μέρη: τη μηχανική δομή (περιλαμβανομένου και του κινητήριου συστήματος) και την δομή ηλεκτρονικών εντολών. Το λογισμικό σύστημα παρέχει στο ρομπότ την προγραμματιζόμενη μνήμη του και επίσης του επιτρέπει να συγχρονίζεται με τον υπόλοιπο μηχανικό εξοπλισμό. Η δυνατότητα επαναπρογραμματισμού προσθέτει προσαρμοστικότητα και επιτρέπει την εκτέλεση πολλαπλών καθηκόντων καθώς επίσης και την ικανότητα χειρισμού μιας ποικιλίας προϊόντων.

Είναι επίσης πιθανόν τα ρομπότ να ταξινομηθούν σύμφωνα με τις διάφορες λειτουργικές κατηγορίες. Έτσι, κατά την ταξινόμηση της Ιαπωνικής

Ένωσης Βιομηχανικών Ρομπότ (JIRA) τα βιομηχανικά ρομπότ κατατάσσονται ως ακολούθως:

1. Χειριστικό ρομπότ. Αυτό ελέγχεται άμεσα από τον χειριστή.
2. Ρομπότ ακολουθίας. Ο χειριστής λειτουργεί εκτελώντας μια προκαθορισμένη ακολουθία ενεργειών και διακρίνονται σε:
 - i) σταθερής ακολουθίας, όπου δύσκολα επιδέχεται αλλαγή της προκαθορισμένης ακολουθίας και
 - ii) μεταβλητής ακολουθίας, με εύκολη την τροποποίηση της ακολουθίας.
3. Ρομπότ Ανάδρασης. Ο αυτόματος χειριστής συγκρατεί στην μνήμη του μια ακολουθία που του δείχνει αυτό που τον ελέγχει.
4. Ρομπότ με Αριθμητικό Έλεγχο. Το χειριστικό αυτό ρομπότ παίρνει εντολές από κάποιο σύστημα αριθμητικού ελέγχου.
5. Έξυπνο ρομπότ. Το ρομπότ αυτό είναι ικανό να καταλάβει τις απαιτούμενες λειτουργίες μέσω αισθητήριας ικανότητας και δυνατότητας αναγνώρισης.

1.3 Τομείς εφαρμογής

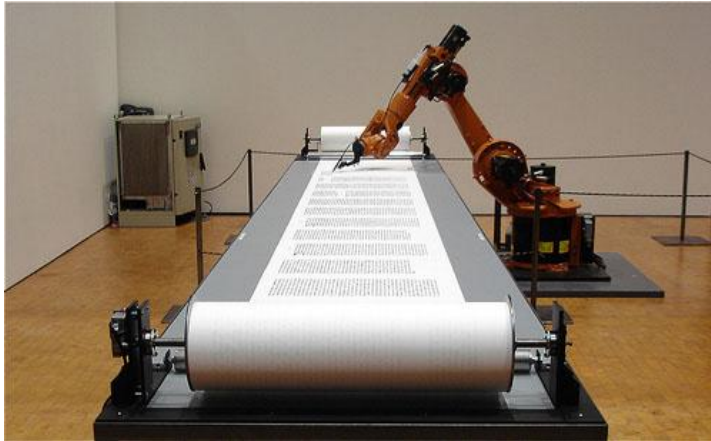
Γενικά, η Ρομποτική (Robotics) είναι κλάδος της τεχνολογίας που ασχολείται με τη σχεδίαση, την ανάπτυξη και τη μελέτη ρομπότ. Η επιστήμη της Ρομποτικής αποτελεί συνδυασμό πολλών άλλων επιστημών, κυρίως δε της πληροφορικής, της ηλεκτρονικής και της μηχανολογίας.

Τα ρομπότ είναι αυτόματες μηχανές με προγραμματισμένη συμπεριφορά, η χρήση των οποίων αποσκοπεί στην αντικατάσταση του ανθρώπου στην εκτέλεση έργου, τόσο σε φυσικό επίπεδο όσο και σε επίπεδο λήψης αποφάσεων.

Η ρομποτική επιστήμη έχει κάνει άλματα προόδου και έχει προσφέρει αρκετά τεχνολογικά θαύματα. Τα ρομποτικά συστήματα συνεχώς εξελίσσονται και είναι ήδη μέρος της ζωής μας σε πολλούς τομείς όπως στη βιομηχανία, την ιατρική, τη διασκέδαση και την προσωπική βοήθεια.

Κατά κύριο λόγο τα ρομπότ χρησιμοποιούνται στην βιομηχανία, όπου και αντικαθιστούν τα ανθρώπινα χέρια, δουλεύοντας ακατάπαυστα, πιο γρήγορα και πιο οικονομικά. Έχουν όμως πολλές εφαρμογές όπως:

© στην βιομηχανία



Ρομποτικός βραχίονας που γράφει κείμενο

© στην διαστημική τεχνολογία



Ρομποτικό όχημα της NASA

- την ιατρική (ακριβείς χειρουργικές επεμβάσεις)



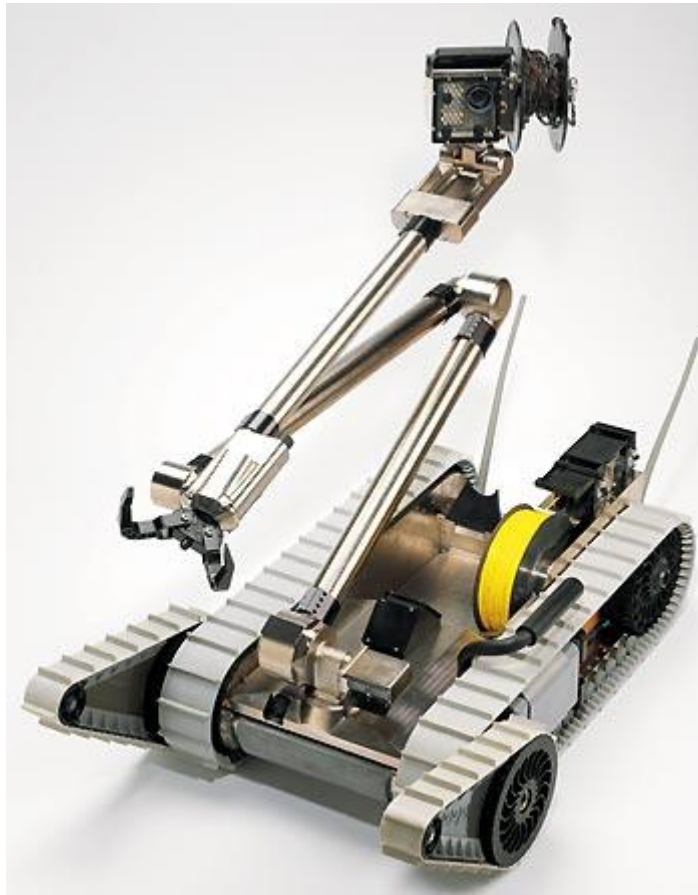
Χειρουργική Ρομποτική

- σε στρατιωτικά προγράμματα



Ρομπότ που ανιχνεύει εκρηκτικές ύλες

📍 σε εγκαταστάσεις πυρηνικής ενέργειας



Ρομπωτικό όχημα που ελέγχει τα επίπεδα ραδιενέργειας

📍 σε ακριβείς συγκολλήσεις και χρωματισμούς



Βιομηχανικά ρομπότ που κατασκευάζουν αυτοκίνητα

- στην ανύψωση και μετακίνηση ογκώδη αντικειμένων



Ρομποτικός διασώστης

- στο χτίσιμο κτηρίων αλλά και την συναρμολόγηση αντικειμένων πολύ πιο γρήγορα και με περισσότερη ακρίβεια



Ρομποτ επιδιόρθωσης δορυφόρων στο διάστημα

📍 στην οικιακή βοήθεια



Ρομπότ που καθαρίζουν το σπίτι

📍 αλλά και για την ανθρώπινη ψυχαγωγία



Ρομποτικός σκύλος

Σε κάποιους τομείς από αυτούς η χρήση του ανθρώπινου δυναμικού είναι είτε επικίνδυνη, είτε δαπανηρή, είτε δύσκολη σε άλλους ικανοποιείται η ανάγκη του ανθρώπου να διευκολύνει την καθημερινότητά του και να ψυχαγωγηθεί στον ελεύθερο χρόνο του. Με την χρήση των ρομπότ αποφεύγουμε τυχόν τραυματισμούς αλλά και σωματική καταπόνηση επίσης κερδίζουμε πολύτιμο χρόνο, διευρύνουμε τους τεχνολογικούς μας ορίζοντες και τέλος διασκεδάζουμε.



Ρομπότ μαέστρος



Ρομπότ που παίζει φλάουτο

ΔΕΥΤΕΡΟ ΚΕΦΑΛΑΙΟ

ΔΕΥΤΕΡΟ ΚΕΦΑΛΑΙΟ

2.1 ΕΙΣΑΓΩΓΗ

Σήμερα χρησιμοποιούνται πολλά είδη ρομπότ μεγάλου και μικρού μεγέθους και πολλά από αυτά είναι φτιαγμένα έτσι ώστε να αναλογούν τα μέρη τους σε ανθρώπινα μέλη. Εμείς θα ασχοληθούμε με έναν ρομποτικό βραχίονα που αποτελείται από τον κύριο βραχίονα, με τον ώμο και τον αγκώνα και από ένα αντιβράχιο, με τον καρπό και μια πένσα που αντιστοιχούν στα δάχτυλα του ανθρώπινου χεριού.

2.2 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΚΑΙ ΕΙΔΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ

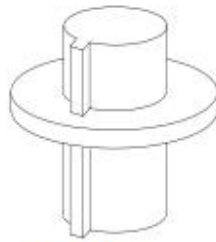
Όπως έχουμε ήδη σημειώσει ένας ρομποτικός βραχίονας αποτελείται από μια σειρά διαδοχικών στερεών σωμάτων που ονομάζονται σύνδεσμοι. Οι σύνδεσμοι συνδέονται ανά δυο μεταξύ τους μέσω αρθρώσεων σχηματίζοντας μία κινηματική αλυσίδα. Οι αρθρώσεις μπορεί να είναι:

- Πρισματικές : σχετική μεταφορική κίνηση μεταξύ δύο διαδοχικών συνδέσμων
- Περιτροφικές : υλοποιούν σχετική περιστροφική κίνηση μεταξύ δύο διαδοχικών συνδέσμων
- Σφαιρικές : υλοποιούν σφαιρική περιστροφική κίνηση μεταξύ δύο διαδοχικών συνδέσμων

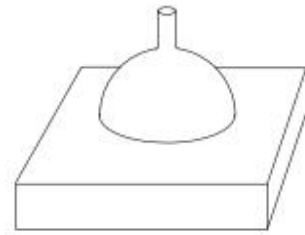
και παρέχουν στην κατασκευή από έναν βαθμό κινητικότητας (Σχήματα 9-11). Με τη σειρά της, μια κινηματική αλυσίδα χαρακτηρίζεται ως ανοικτή όταν υπάρχει μια μόνο διαδοχή συνδέσμων που να συνδέει τα δύο άκρα του βραχίονα και κλειστή όταν οι σύνδεσμοι που τη συνιστούν σχηματίζουν βρόχο.



Σχήμα 9 Πρισματική άρθρωση



Σχήμα 10 Περιστροφική άρθρωση



Σχήμα 11 Σφαιρική άρθρωση

2.2.1 Βαθμοί κινητικότητας και βαθμοί ελευθερίας

Κρίνεται σκόπιμο να επισημάνουμε τη διαφορά που υπάρχει ανάμεσα στους βαθμούς κινητικότητας και τους βαθμούς ελευθερίας που απαιτούνται για την εκτέλεση ενός έργου. Για ένα βραχίονα το πλήθος των βαθμών κινητικότητας είναι σταθερό και ίσο με το πλήθος των αρθρώσεων του (πρισματικών ή/και περιστροφικών). Από την άλλη πλευρά οι βαθμοί ελευθερίας είναι άμεσα συνδεδεμένοι με το συγκεκριμένο έργο που καλείτε να φέρει εις πέρας ο βραχίονας. Για την γενικευμένη περίπτωση που θέλουμε να τοποθετήσουμε και να προσανατολίσουμε ένα αντικείμενο στον τρισδιάστατο χώρο απαιτούνται 6 βαθμοί ελευθερίας (3 για να τοποθετήσουμε ένα σημείο του αντικειμένου στο χώρο και 3 για να προσανατολίσουμε το αντικείμενο ως προς ένα σύστημα συντεταγμένων αναφοράς). Είναι προφανές ότι ένας ρομποτικός βραχίονας με 6 βαθμούς κινητικότητας μπορεί να ανταπεξέλθει σε αυτό το έργο, όπως επίσης και σε οποιοδήποτε άλλο έργο που απαιτεί μέχρι 6 βαθμούς ελευθερίας.

Ως χώρος εργασίας ορίζεται ο τρισδιάστατος χώρος τον οποίο μπορεί να σαρώσει η άκρη του ρομποτικού μηχανισμού. Το μέγεθος και η γεωμετρική μορφή του χώρου αυτού εξαρτώνται από την κατασκευαστική δομή του ρομπότ.

Από τα πιο σημαντικά μεγέθη ενός βιομηχανικού βραχίονα είναι το ωφέλιμο φορτίο, η επαναληψιμότητα και η ακρίβεια. Πιο συγκεκριμένα τα παραπάνω μεγέθη αναφέρονται στα εξής :

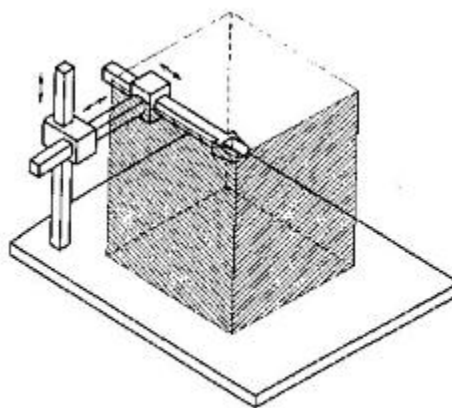
- **Ωφέλιμο φορτίο:** είναι το βάρος που μπορεί να μεταφέρει το άκρο του βραχίονα. Ως σημείο εφαρμογής του βάρους θεωρείται η φλάντζα του καρπού. Το προδιαγραφόμενο αυτό φορτίο δεν είναι σταθερό και εξαρτάται από την ταχύτητα με την οποία πρόκειται να κινηθεί ο καρπός.
- **Επαναληψιμότητα:** εκφράζει την δυνατότητα του βραχίονα να γυρίζει στο ίδιο σημείο μετά από αρκετές επαναλήψεις και δίνεται ως εύρος μέσα το οποίο ο βραχίονας θα τερματίσει την κίνηση. Η απόκλιση οφείλεται στο ότι κατά την λειτουργία του το ρομπότ είναι δυνατό να χάσει λίγο από τη μέτρηση της θέσης με αποτέλεσμα να μη μπορεί να επιστρέψει στη συγκεκριμένη θέση μετά από ορισμένους κύκλους λειτουργίας.

- Δεδομένου ότι στις συνήθεις βιομηχανικές εφαρμογές οι επιθυμητές κινήσεις διδάσκονται στο ρομπότ αντιλαμβάνεται κανείς τη σπουδαιότητα της επαναληψιμότητας.
- Ακρίβεια: είναι η ικανότητα του ρομπότ να πηγαίνει ακριβώς στη θέση που του έχει δοθεί εντολή να πάει. Η ακρίβεια εξαρτάται κυρίως από τη διακριτότητα των εξαρτημάτων ελέγχου, τη μηχανολογική σύνδεση των μελών του και το ελάχιστο επιτρεπόμενο σφάλμα που επιβάλλει η ευστάθεια της λειτουργίας των ρομπότ. Η ακρίβεια επηρεάζεται από το είδος και το μέγεθος του εκάστοτε φορτίου, σε αντίθεση με την επαναληψιμότητα.

2.2.2 Ταξινόμηση βραχιόνων βάσει της Γεωμετρικής Διαμόρφωσής τους:

Ο τύπος και η διαδοχή των αρθρώσεων ενός βραχίονα επιτρέπει την ταξινόμηση των ρομπότ σε διάφορες κατηγορίες, οι οποίες αναφέρονται παρακάτω. Οι αρθρώσεις που μας απασχολούν στο σημείο αυτό είναι οι τρεις πρώτες του βραχίονα και κατά συνέπεια εξαιρούνται οι αρθρώσεις του καρπού. Θα έχουμε λοιπόν τα εξής:

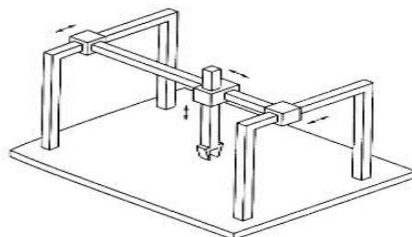
- Καρτεσιανοί βραχίονες: η καρτεσιανή γεωμετρία υλοποιείται με τρεις διαδοχικές πρισματικές αρθρώσεις. Οι άξονες των αρθρώσεων αυτών είναι ανά δύο κάθετοι μεταξύ τους (Σχήμα 12). Η καρτεσιανή δομή παρέχει μεγάλη δυσκαμψία και σταθερή ακρίβεια σε ολόκληρο το χώρο εργασίας που είναι ένα παραλληλεπίπεδο. Βασικό μειονέκτημα της κατασκευής είναι η μειωμένη επιδεξιότητα κίνησης λόγω της πρισματικής φύσης των αρθρώσεων.



Σχήμα 12 Καρτεσιανός Βραχίονας

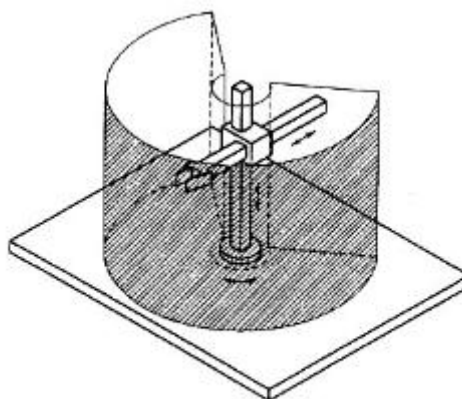
- Βραχίονες Gantry: οι βραχίονες Gantry είναι στην ουσία καρτεσιανοί, διαφέρουν όμως από τους τελευταίους στον τρόπο προσέγγισης του ενδιαφέροντος αντικείμενου (Σχήμα 13). Ειδικότερα ο βραχίονας Gantry προσεγγίζει το αντικείμενο από πάνω, τη στιγμή που ένας κλασικός

καρτεσιανός βραχίονας προσεγγίζει το αντικείμενο από το πλάι. Άμεσες συνέπειες της διαφοροποίησης αυτής είναι η αύξηση του χώρου εργασίας και της δυσκαμψίας, καθώς επίσης και η δυνατότητα χειρισμού μεγάλων και βαριών αντικειμένων.



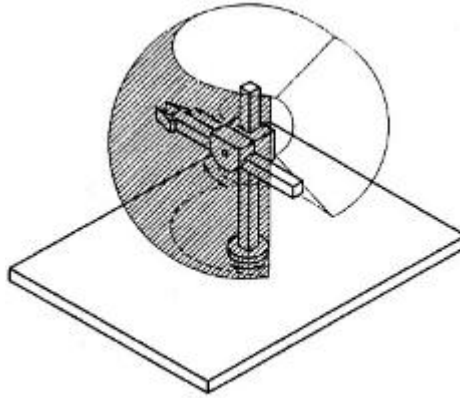
Σχήμα 13 Βραχίονας Gantry

- Κυλινδρικοί βραχίονες: στους κυλινδρικούς βραχίονες η πρώτη πρισματική άρθρωση της καρτεσιανής δομής έχει αντικατασταθεί από μια περιστροφική άρθρωση (Σχήμα 14). Οι συγκεκριμένοι βραχίονες χαρακτηρίζονται από καλή δυσκαμψία, όμως η ακρίβεια της θέσης του καρπού μειώνεται καθώς η οριζόντια μετατόπιση αυξάνεται. Ο χώρος εργασίας στην περίπτωση αυτή είναι τμήμα κυλίνδρου. Σημαντικό μειονέκτημα της συγκεκριμένης γεωμετρίας είναι το ότι ο βραχίονας εισέρχεται στο χώρο εργασίας και τον περιορίζει.



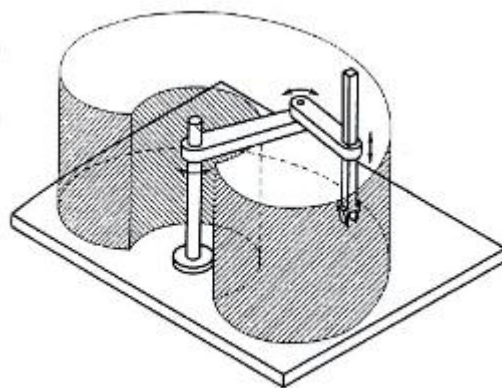
Σχήμα 14 Κυλινδρικός Βραχίονας

- Σφαιρικοί βραχίονες: στους βραχίονες αυτούς αντικαθίσταται πλέον και η δεύτερη πρισματική άρθρωση της καρτεσιανής δομής με περιστροφική (Σχήμα 15). Η μηχανολογική πολυπλοκότητα αυξάνει ενώ η δυσκαμψία μειώνεται με την αύξηση της ακτινικής απόστασης. Ο χώρος εργασίας είναι τμήμα σφαίρας και περιέχει ένα μέρος της βάσης με άμεση συνέπεια τη δυνατότητα χειρισμού αντικειμένων που βρίσκονται στο έδαφος.

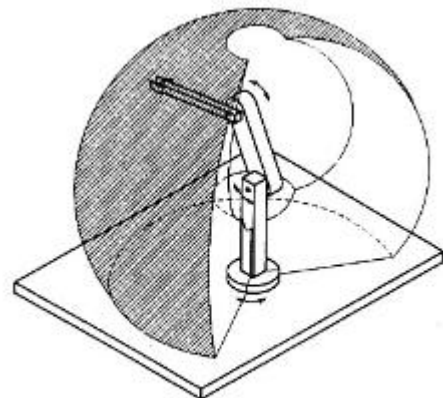


Σχήμα 15 Σφαιρικός Βραχίονας

- Βραχίονες SCARA: η γεωμετρία SCARA είναι ειδική και περιλαμβάνει δύο περιστροφικές και μία πρισματική άρθρωση τοποθετημένες κατά τέτοιο τρόπο ώστε οι άξονες κίνησης να είναι παράλληλοι μεταξύ τους (Σχήμα 16). Το όνομα SCARA προέρχεται από τα αρχικά των λέξεων Selective Compliance Assembly Robot Arm. Η συγκεκριμένη γεωμετρία παρέχει μεγάλη δυσκαμψία σε κατακόρυφη φόρτιση και ελαστικότητα σε οριζόντια. Η ακρίβεια τοποθέτησης του καρπού μειώνεται με την αύξηση της απόστασης του από τον άξονα της πρώτης άρθρωσης.
- Ανθρωπομορφικοί βραχίονες: η ανθρωπομορφική γεωμετρία υλοποιείται με τρεις διαδοχικές περιστροφικές αρθρώσεις. Ειδικότερα, ο άξονας περιστροφής της πρώτης άρθρωσης είναι κατακόρυφος και κάθετος στους άξονες περιστροφής των επόμενων δυο αρθρώσεων, οι οποίοι είναι παράλληλοι μεταξύ τους (Σχήμα 17). Η συγκεκριμένη δομή παρέχει τη μεγαλύτερη επιδεξιότητα από όλες τις προηγούμενες, καθώς όλες οι αρθρώσεις είναι περιστροφικές. Ωστόσο η ακρίβεια του καρπού δεν είναι σταθερή εντός του χώρου εργασίας που έχει τη μορφή σφαίρας.



Σχήμα 16 Βραχίονας SCARA

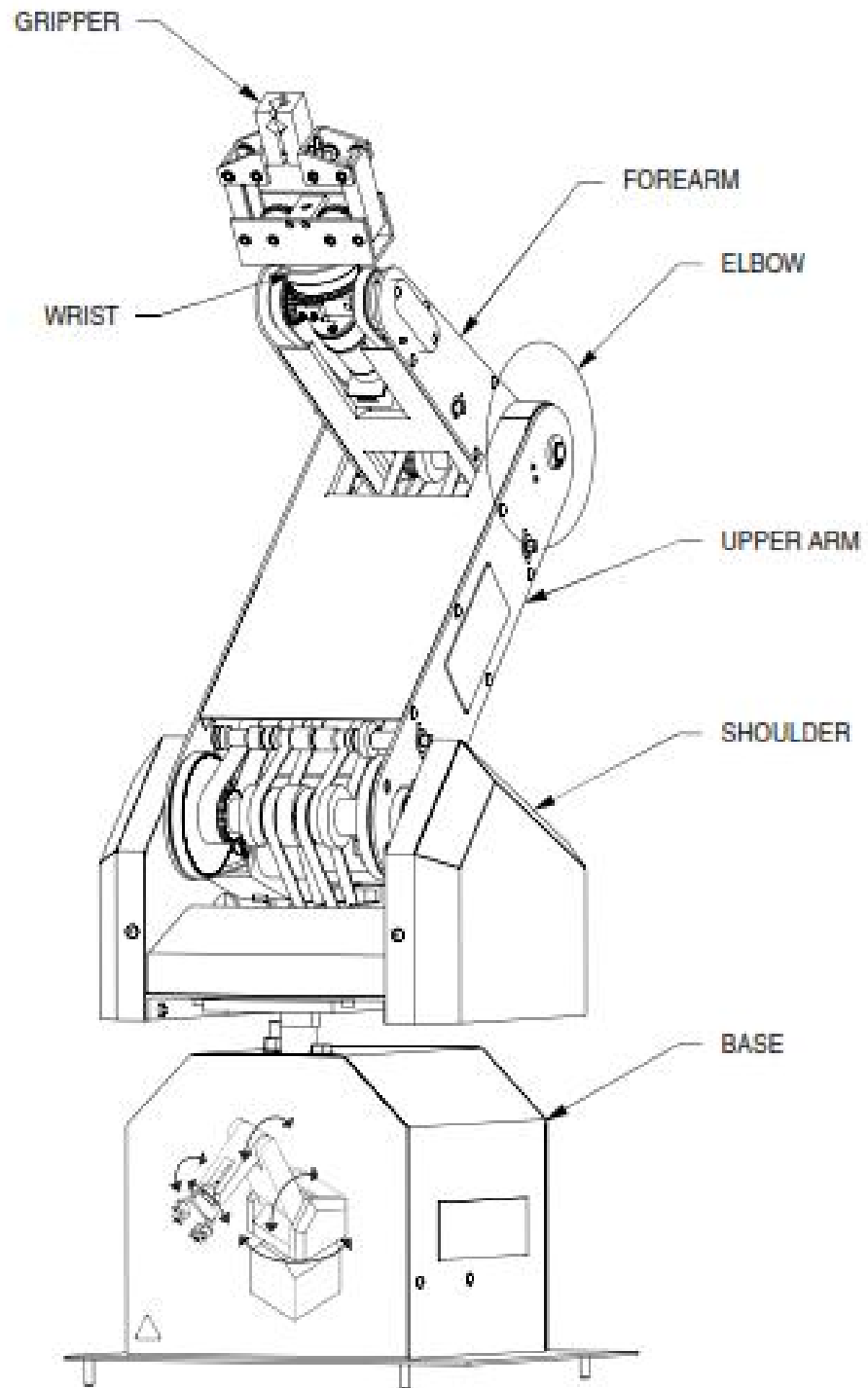


Σχήμα 17 Ανθρωπομορφικός Βραχίονας

2.3 Ρομποτικός βραχίονας Lab-Volt

Το Lab-Volt Servo robot ανήκει στους ανθρωπομορφικούς βραχίονες, αποτελείται από τον ρομποτικό βραχίονα, τον ελεγκτή χειρός του Servo, ένα φορητό τερματικό, πλέγματα, ώστε ο βραχίονας να προσδένεται σταθερά στα κατάλληλα σημεία που θέλουμε και τέλος το cd με το κατάλληλο λογισμικό για την διαχείριση του βραχίονα.

Στην εικόνα 2-1 παρακάτω φαίνεται το γενικό μηχανικό σχεδιάγραμμα του Servo ρομπότ. Ο ρομποτικός μας βραχίονας έχει πέντε περιστροφικούς άξονες δηλαδή πέντε βαθμούς ελευθερίας: βάση, ώμος, αγκώνας, καρπός, και πένσα. Η βάση υποστηρίζει τον μηχανισμό του βραχίονα και στεγάζει την μηχανή που δίνει την περιστροφή στον ώμο. Ο ώμος περιστρέφεται στη βάση και στεγάζει τους μηχανισμούς που συνδέονται και κινούν μαζί με τις ζώνες τα άλλα τμήματα του βραχίονα.



Εικόνα 2-1: Μηχανικά τμήματα του servo-ρομπότ

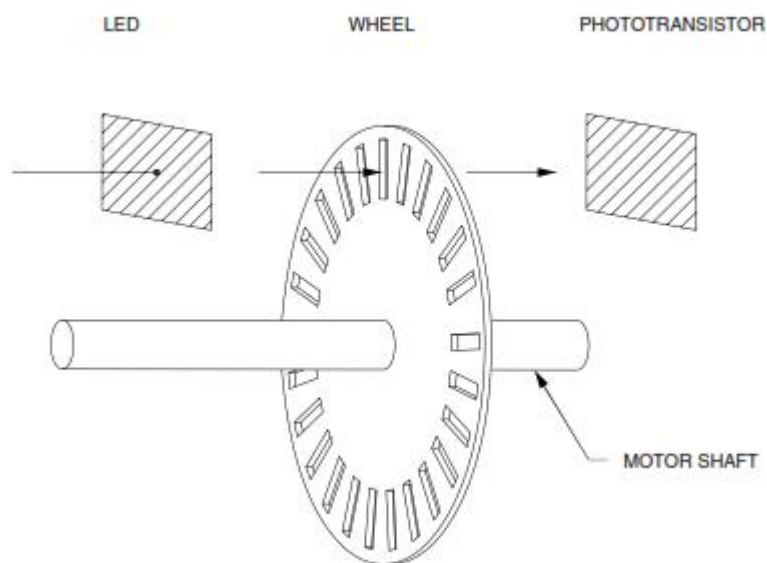
Το χαμηλότερο όριο του ανώτερου βραχίονα φέρει τα εργαλεία και τις τροχαλίες που οδηγούν το αντιβράχιο, καρπός και πένσα. Ο ανώτερος βραχίονας και το αντιβράχιο κινούνται πάνω-κάτω. Ο καρπός και η πένσα λειτουργούν μαζί για να πάρουν τα αντικείμενα και να τα κινήσουν σε επιθυμητές θέσεις. Ο καρπός κινείται πάνω κάτω για να ελέγξει την πένσα δυο-δάχτυλων. Μπορεί επίσης να περιστρέφεται δεξιόστροφα 360 μοίρες ή και αντίστροφα. Τα δάχτυλα κίνησης των πενσών μπορούν να ανοίγουν και να κλείνουν. Η λειτουργία του servo-ρομπότ ελέγχεται χρησιμοποιώντας το φορητό τεμαχικό ή από έναν οικοδεσπότη υπολογιστή με λογισμικό ρομποτικής.

2.3.1 Ο ελεγκτής ρομπότ

Ο ελεγκτής ρομπότ είναι ο εγκέφαλος του servo-ρομπότ. Είναι ένας εξειδικευμένος υπολογιστής με τους διακόπτες και τις ικανότητες προγραμματισμού που του επιτρέπουν να ελέγχει το servo-ρομπότ αλλά και εξωτερικές συσκευές.

2.3.2 Ο οπτικός κωδικοποιητής:

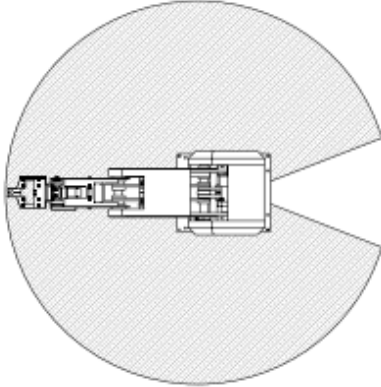
Ο οπτικός κωδικοποιητής αποτελείται από μια LED (Light Emitting Diode) δίοδο που στέλνει φως μέσω μιας λεπτά καλυμμένης ρόδας κώδικα που επικολλείται άκαμπτα επάνω στον άξονα μηχανών όπως φαίνεται στην εικόνα 2-2. Ένας δέκτης (φωτοτρανζίστορ) ανιχνεύει τις ελαφριές αλλαγές (φως/σκοτάδι). Οι ελαφριές αλλαγές μετατρέπονται σε ηλεκτρικούς σφυγμούς και στέλνονται πίσω στον ελεγκτή ρομπότ. Ο ελεγκτής ρομπότ συγκρίνει το σήμα ελέγχου με το σήμα της ανατροφοδότησης για να εξασφαλίσει ότι η θέση είναι πάντα σωστή.



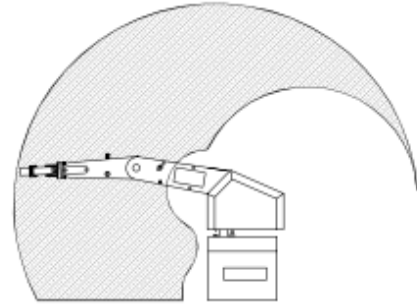
Εικόνα 2-2: Οπτικός κωδικοποιητής του Servo-ρομπότ

2.3.3 Εύρος εργασίας:

Το εύρος εργασίας του Servo-ρομπότ είναι το μονοπάτι των σημείων που μπορεί να επιτευχθεί από ένα ρομπότ. Στο παρακάτω εικόνα 2-3 φαίνεται το εύρος εργασίας ενός servo-ρομπότ.



ΚΑΤΟΨΗ



ΠΛΑΓΙΑ ΟΨΗ

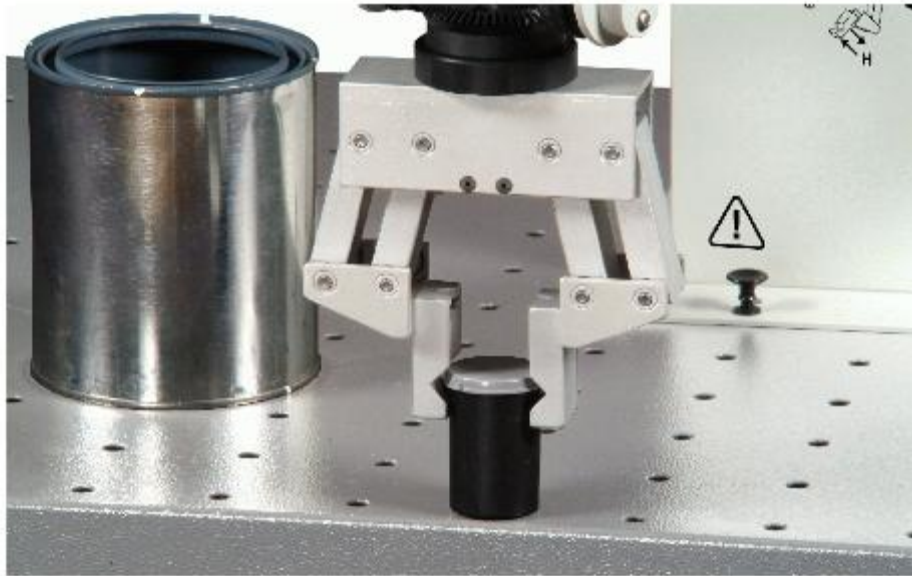
Εικόνα 2-3: Εύρος εργασίας ρομπωτικού βραχίονα

2.3.4 Συσκευή επίδρασης:

Το τέλος του servo-ρομπότ, η συσκευή επίδρασης, είναι το άκρο που συνδέεται με το βραχίονα ενός ρομπότ. Θα μπορούσε να είναι οποιοσδήποτε μηχανισμός συμπεριλαμβανομένης μιας πένσας, ενός φανού συγκόλλησης, ενός ακροφύσιου χρωμάτων, μιας κενής αντλίας, ενός χειρουργικού νυστεριού, μιας λυχνίας συγκόλλησης, ή οτιδήποτε που χρησιμοποιείται για να κάνει μια εργασία. Μερικά ρομπότ μπορούν να αλλάξουν το άκρο τους και να επαναπρογραμματιστούν για ένα διαφορετικό σύνολο στοιχειωδών εργασιών. Το άκρο, η συσκευή επίδρασης, του servo-ρομπότ του συστήματός μας είναι μια μηχανική πένσα με δυο δάχτυλα.

2.3.5 Σημεία θέσεων :

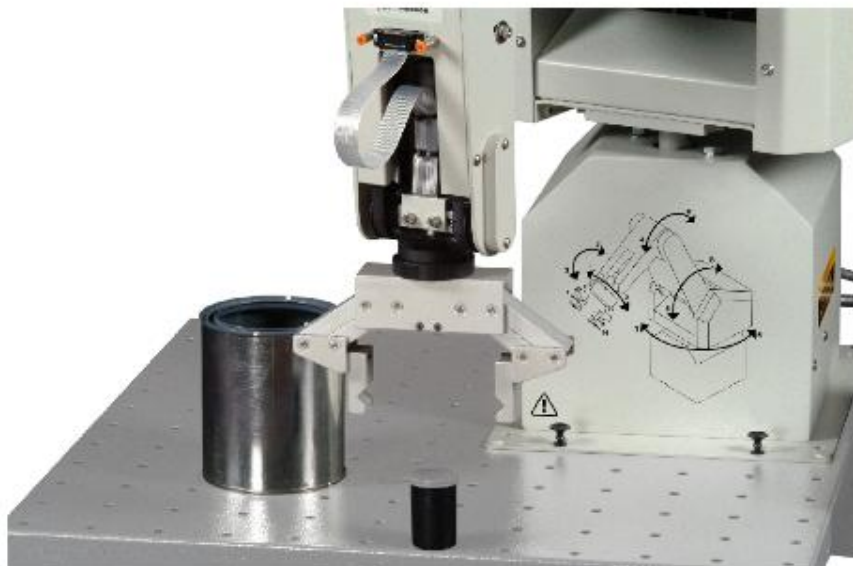
Τα σημεία θέσεων είναι αρχεία που περιέχουν τις απαιτούμενες πληροφορίες για τη θέση κάθε σημείου ώστε να υπολογιστεί ένα μονοπάτι που θα επιτρέψει στο ρομπότ να κινηθεί, χωρίς να το εμποδίσουν άλλα αντικείμενα, στις απαραίτητες θέσεις. Οι πληροφορίες που περιλαμβάνονται σε ένα σημείο θέσης μπορεί να είναι το όνομα σημείου, οι συντεταγμένες κάθε άρθρωσης του ρομπότ, καθώς και η ταχύτητά του βραχίονα. Στην εικόνα 2-4 βλέπουμε ένα σημείο μιας συγκεκριμένης αποθηκευμένης θέσης.



Εικόνα 2-4: Σημείο θέσης

2.3.6 Σημεία ελέγχου :

Τα σημεία ελέγχου είναι αρχεία που περιέχουν τις πληροφορίες για τη θέση των εισόδων και εξόδων του ελεγκτή ρομπότ. Οι πληροφορίες μπορούν επίσης να περιλάβουν το όνομα σημείου και το χρόνο καθυστέρησης. Τα σημεία ελέγχου χρησιμοποιούνται για να ελέγξουν την εκτέλεση μιας ακολουθίας σημείων καθώς και την λειτουργία των εξωτερικών συσκευών. Σώζονται και παρεμβάλλονται σε ένα πρόγραμμα ως σημεία θέσης.



Εικόνα 2-5: Σημείο ελέγχου

2.3.7 Αρχική θέση (Homeposition) :

Τα προγράμματα αναπτύσσονται συνήθως για να χρησιμοποιηθούν περισσότερο από μία φορά. Προκειμένου να επαναληφθεί το κάθε πρόγραμμα, ο εξοπλισμός θα πρέπει να τοποθετηθεί στις ίδιες θέσεις τον ίδιο προσανατολισμό όπως όταν το δημιουργήσαμε. Έτσι τα σημεία θέσης του προγράμματος και τα σημεία του Servo Robot πρέπει να ξεκινούν από την ίδια θέση.

Η αρχική θέση του Servo Robot καλείται homeposition. Η θέση μπορεί να είναι μια σκληρή αρχική θέση ή μια μαλακή αρχική θέση. Το σκληρό homeposition είναι ένα σταθερό σημείο αναφοράς όπου ένα ρομπότ πηγαίνει να επαναριθμήσει τις παραμέτρους των μηχανών του στον αρχικό τρόπο. Ένας σκληρός βασικός προσδιορισμός θέσης απαιτείται στην αρχή κάθε περιόδου επικοινωνίας.

Το μαλακό homeposition είναι μια θέση που εμείς αναθέτουμε στο ρομπότ. Αντιστοιχεί στη θέση από όπου το ρομπότ αρχίζει τις μετακινήσεις του. Μετά από την ολοκλήρωση όλων των ανατεθειμένων στοιχειωδών εργασιών, αυτή είναι η θέση όπου το ρομπότ θα έρθει να στηριχτεί όταν το προγραμματίζουμε να πάει σπίτι με την εντολή home. Μπορούμε να ποικίλουμε το μαλακό homeposition σύμφωνα με τις στοιχειώδεις εργασίες που θέλουμε να αποδώσει το ρομπότ.

2.4 Προγραμματισμός

2.4.1 Μέθοδοι προγραμματισμού:

Υπάρχουν τέσσερις βασικές μέθοδοι για να προγραμματίσεις ένα servo-ρομπότ. Χειροκίνητα (Manual) από το εγχειρίδιο, άμεσα (Lead-through), από πέρασμα (Walkthrough) και τέλος προγραμματισμός από λογισμικό (Software programming).

I. (Manual)

Η χειροκίνητη μέθοδος προγραμματισμού ολοκληρώνεται από έναν χειριστή που θέτει τις απαραίτητες τάσεις, τους διακόπτες, τα έγκεντρα και τα ηλεκτρικά καλώδια. Αυτός ο τύπος προγραμματισμού χρησιμοποιείται σε λιγότερο περίπλοκα ρομπότ.

II. (Lead-through)

Στη μέθοδο άμεσου προγραμματισμού, ο χειριστής χρησιμοποιεί το άκρο του βραχίονα για να οδηγήσει το ρομπότ στις επιθυμητές θέσεις. Καθώς το ρομπότ φθάνει σε κάθε επιθυμητό σημείο μέσω της ακολουθίας

κίνησης, το σημείο σώζεται στη μνήμη. Οι συντεταγμένες του κάθε σημείου αντιστοιχούν στη θέση των αρθρώσεων έναντι μιας θέσης αναφοράς. Τα σωζόμενα σημεία χρησιμοποιούνται για να παραγάγουν το μονοπάτι που το ρομπότ ακολουθεί κατά τη λειτουργία του προγράμματος.

III. (Walkthrough)

Στη μέθοδο προγραμματισμού με πέρασμα από τα σημεία, ένας χειριστής κινεί το ρομπότ με τις επιθυμητές κινήσεις. Ενώ ο χειριστής κινεί το ρομπότ μέσω του επιθυμητού μονοπατιού, τα διάφορα σημεία επιλέγονται και σώζονται στη μνήμη για να τα χρησιμοποιήσει την κατάλληλη στιγμή.

IV. (Software programming)

Στη μέθοδο προγραμματισμού μέσω λογισμικού, το ρομπότ είναι προγραμματισμένο με τη βοήθεια ενός υπολογιστή. Αυτή η μέθοδος είναι επίσης γνωστή όπως ασύρματος προγραμματισμός, από τότε ο προγραμματισμός πραγματοποιείται γενικά μακριά από τα ρομπότ.

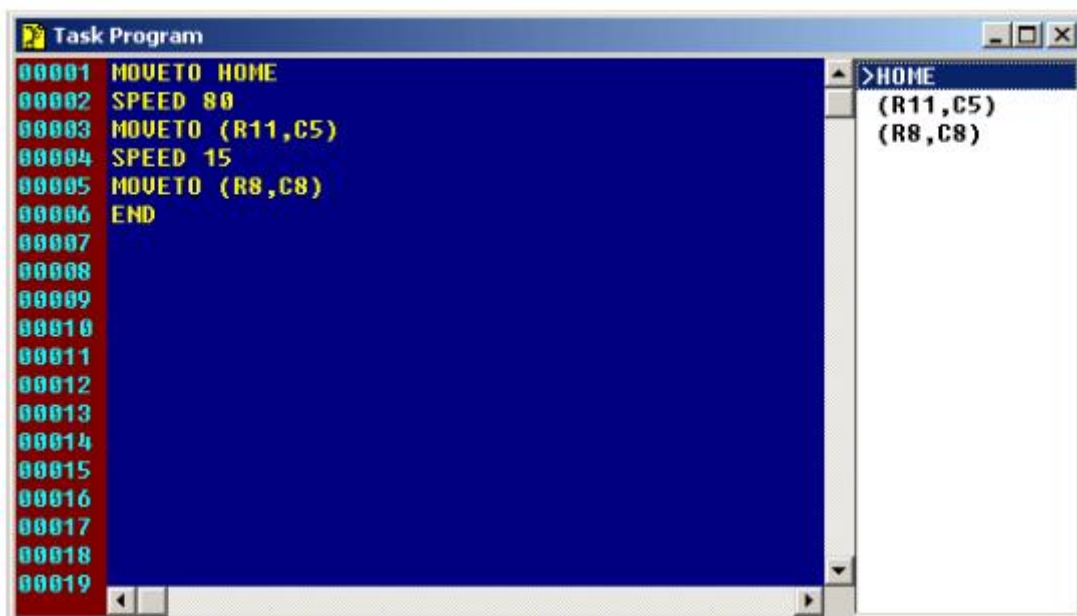
2.4.2 Από σημείο σε σημείο προγράμματα :

Τα από σημείο σε σημείο προγράμματα αποτελούνται από έναν κατάλογο σημείων θέσης και ελέγχου. Η ακολουθία με τα σημεία εκτελείται όπως εμφανίζεται στον κατάλογο σημείων. Τα από σημείο σε σημείο προγράμματα μπορούν να δημιουργηθούν χρησιμοποιώντας φορητά τερματικά και τη ρομποτική.

2.4.3 Στοιχειώδη προγράμματα :

Τα στοιχειώδη προγράμματα συνίστανται σε έναν κατάλογο γραμμών-οδηγιών που περιέχουν τα σημεία θέσης και στοιχειώδεις εντολές. Τα σημεία θέσης καθορίζονται χρησιμοποιώντας το φορητό τερματικό ή τη ρομποτική.

Κατά το γράψιμο ενός προγράμματος, τα σημεία επιλέγονται από τον κατάλογο και το αντίγραφο σημείου από την αριστερή στήλη του στοιχειώδη επεξεργαστή όπου συνδέονται με τις στοιχειώδεις εντολές.



Εικόνα 2-6: Παράθυρο ενός στοιχειώδους επεξεργαστή

Στην παραπάνω εικόνα βλέπουμε ένα παράθυρο ενός στοιχειώδους επεξεργαστή. *Σημειώνουμε ότι το όνομα ενός σημείου στην αριστερή στήλη πρέπει να είναι ακριβώς το ίδιο με την ορθογραφία από το σημείο στη δεξιά στήλη. Τα σημεία στη δεξιά στήλη μπορούν να χρησιμοποιηθούν όσο συχνά επιθυμούμαι, και δεν είναι υποχρεωτικό να χρησιμοποιηθούν εξολοκλήρου κάθε ένα από αυτά.

Οι στοιχειώδεις εντολές εισάγονται με την τοποθέτηση του δρομέα στον κατάλληλο αριθμό γραμμής και δακτυλογραφώντας το όνομα της εντολής. Στο στοιχειώδες πρόγραμμα που παρουσιάζεται στο σχήμα 3-1, τρεις στοιχειώδεις εντολές χρησιμοποιούνται: Moveto, Speed και End. Κατά την εκτέλεση, αυτό το στοιχειώδες πρόγραμμα θα προκαλέσει το Servo- ρομπότ κίνηση προς το homerosition (αρχική θέση), κίνηση προς το σημείο (R11, C5) με μια ταχύτητα 80, κίνηση προς το σημείο (R8, C8) με μια ταχύτητα 15, η εντολή End δηλώνει το τέλος του προγράμματος.

Ένα στοιχειώδες πρόγραμμα είναι πολύ περισσότερα από έναν κατάλογο σημείου. Περιλαμβάνει τις εντολές, τα στοιχεία, τα σχόλια, και τις πληροφορίες ελέγχου, ικανές να εκτελεστούν από το servo-ρομπότ και τις εξωτερικές συσκευές. Στο στοιχειώδη προγραμματισμό, το πρόγραμμα οργανώνεται από τον οικοδεσπότη υπολογιστή και το αρχείο είναι σωζόμενο στο σκληρό δίσκο του υπολογιστή.

2.4.4 Προγραμματισμός χρησιμοποιώντας το φορητό τερματικό ή τη ρομποτική.

Και το φορητό τερματικό και η ρομποτική μπορούν να χρησιμοποιηθούν για από σημείο σε σημείο προγραμματισμό. Το πλεονέκτημα στη χρησιμοποίηση του φορητού τερματικού είναι η κινητικότητα. Μπορούμε να κινηθούμε προς οποιαδήποτε θέση ώστε να την αποθηκεύσει και να βρεθεί εκεί ο εξοπλισμός με ακρίβεια.

Με τη ρομποτική, ένα από τα πλεονεκτήματα είναι η απλότητα για να ανεπτυχθεί ένα πρόγραμμα. Βρίσκουμε τα σημεία και δίνουμε την εντολή στο servo-ρομπότ ώστε να κινηθεί και να τα επιλέξει. Επιπλέον, οι ικανότητα της ρομποτικής να δημιουργήσει απλά ένα στοιχειώδες πρόγραμμα δεν είναι διαθέσιμη κατά την χρησιμοποίηση του φορητού τερματικού μόνο.

Η χρήση και των δυο συστημάτων στη δημιουργία ενός προγράμματος επιτρέπει την ακρίβεια στις μετακινήσεις και την απλότητα στη σχεδίαση του προγράμματος.

2.4.5 Βασικές εντολές

Ο προγραμματισμός λοιπόν γίνεται με την βοήθεια εντολών που θα παραθέσουμε παρακάτω.

Moveto (μετακίνηση σε)

Με αυτή την εντολή ο ρομποτικός ελεγκτής μπορεί να εκτελέσει, είτε κίνηση σε μια θέση είτε να μετακινηθεί σε ένα σημείο ελέγχου. Η σύνταξη και οι παράμετροι που συνδέονται με τη εντολή αυτή είναι οι εξής: **Moveto** <δίνουμε το όνομα του σημείου>. Για παράδειγμα: **Moveto POINT1**.

SPEED (ταχύτητα)

Η εντολή αυτή μας επιτρέπει να θέσουμε ένα όριο ταχύτητας σε κάθε οδηγία που εκτελείται από τον βραχίονα, η οποία περιέχει την εντολή αυτή. Όλες οι κινήσεις θα εκτελεστούν με την νέα ταχύτητα που έχουμε θέσει. Οι κινήσεις από σημείο σε σημείο στις οποίες έχουμε θέσει την εντολή ταχύτητας (**Speed command**) στο "0" θα έχουν προκαθορισμένη ταχύτητα. Η σύνταξη και οι παράμετροι που συνδέονται με τη εντολή ταχύτητας είναι: **Speed** < δίνουμε αριθμό από 0 έως 99>. Για παράδειγμα : **Speed 20**.

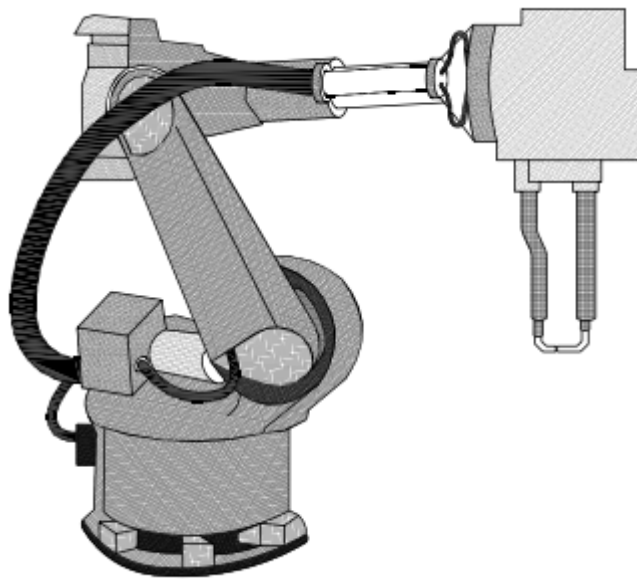
End (Τέλος)

Αυτή η εντολή τελειώνει την εκτέλεση μιας συγκεκριμένης οδηγίας του προγράμματος. Μπορεί να τοποθετείται οπουδήποτε εντός των εντολών του προγράμματος, αλλά στη συνέχεια όλες οι οδηγίες που έχουν δοθεί μετά την εντολή αυτή δεν θα εκτελεστούν ποτέ.

2.4.6 Τροποποίηση προγραμμάτων

Τα ρομπότ όπως έχουμε πει σε προηγούμενο κεφάλαιο χρησιμοποιούνται σε κατασκευαστικά περιβάλλοντα που οι άνθρωποι βρσκουν βαρετά. Τα ρομπότ μπορούν να προγραμματιστούν για να κάνουν επαναλαμβανόμενες στοιχειώδεις εργασίες, η ακρίβειά τους είναι υψηλή και μπορούν να λειτουργούν ασταμάτητα χωρίς λάθη εικοσιτέσσερις ώρες την ημέρα.

Έτσι ένα ρομπότ όπως αυτό που παρουσιάζεται στην εικόνα 2-7, μπορεί να κινήσει επανειλημμένα το πυροβόλο όπλο συγκόλλησης σε κάθε απαιτούμενη θέση και να το τοποθετεί στη ραφή συγκόλλησης. Ένας χειριστής πυροβόλου όπλου είναι λιγότερο αρεστός αφού θα αποδώσει πολύ λιγότερο από τον ρομποτικό βραχίονα λόγω του βάρους του πυροβόλου όπλου συγκόλλησης, για τον ίδιο λόγο είναι πιθανό η ραφή συγκόλλησης να γίνει λάθος. Οι ρομποτικοί βραχίονες έχουν πολλούς άξονες κίνησης για να πλησιάσουν τα σημεία εργασίας από κάθε κατάλληλη γωνία. Αυτό επιτρέπει το ρομπότ να είναι εύκαμπτο.



Εικόνα 2-7: Ρομπότ συγκόλλησης

Όταν λοιπόν ένα προϊόν τροποποιείται, το πρόγραμμα που ελέγχει το ρομπότ πρέπει να τροποποιηθεί κατάλληλα ώστε να λάβει υπόψη του τα νέα χαρακτηριστικά του προϊόντος.

2.4.7 Επιμελητής προγράμματος

Ο επιμελητής προγράμματος χρησιμοποιείται για να βελτιώσει την αποδοτικότητα ενός προγράμματος ή και να αποβάλει λάθη. Μας επιτρέπει να τροποποιήσουμε, να προσθέσουμε, και να διαγράψουμε τα σημεία θέσης και

τα μέσα ελέγχου σε ένα από σημείο σε σημείο πρόγραμμα, αλλά βοηθά και στην τροποποίηση ενός στοιχειώδους προγράμματος με την προσθήκη και την διαγραφή, κινούμενων σημείων και στοιχειωδών εντολών.

2.5 Είσοδοι (inputs) και έξοδοι (outputs)

Όπως είδαμε παραπάνω το servo-ρομπότ μπορεί να προγραμματιστεί και να εκτελέσει τις μετακινήσεις με την καταγραφή των συντεταγμένων σημείου. Χρησιμοποιήσαμε τις οδηγίες μετακίνησης για να κινηθεί η πένσα από μια θέση προς μια άλλη. Μερικές από αυτές τις εντολές χρησιμοποιούν τις εισόδους και τις εξόδους που βρίσκονται στον ελεγκτή-ρομπότ για να μπορέσουν να ανταποκριθούν, ώστε να προχωρήσει παρακάτω το πρόγραμμα. Στα από σημείο σε σημείο προγράμματα, οι εισοδοι και οι εξοδοι τίθενται μέσω των σημείων ελέγχου. Στα στοιχειώδη προγράμματα, τίθενται μέσω των στοιχειωδών εντολών.

2.5.1 Εντολές ελέγχου:

Οι παράμετροι που μπορούν να οριστούν σε σημεία ελέγχου είναι:

- είσοδοι TTL (TTL INPUTS)
- έξοδοι TTL (TTL OUTPUTS)
- επιβραδυντές (RELAYS)
- Και καθυστέρηση (DELAY)

Είσοδοι TTL (TTL INPUTS) :

Η οδηγία αυτή ορίζει το επίπεδο στο οποίο πρέπει να βρίσκεται μια είσοδος TTL πριν να προχωρήσει στην επόμενη γραμμή εντολής του προγράμματος. Η τιμή εισόδου και εξόδου TTL του ελεγκτή ρομπότ απεικονίζονται στην εικόνα 2-8.

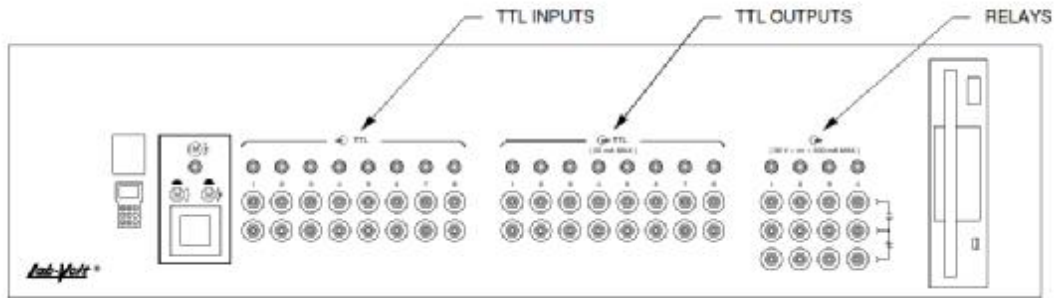
Στην ρομποτική, το επίπεδο TTL ορίζεται, επιλέγοντας το αντίστοιχο LED στο παράθυρο αποθήκευσης σημείου ελέγχου (save control point window). Η εντολή εισόδου μπορεί να χρησιμοποιηθεί επίσης για να οριστεί το επίπεδο των εισόδων TTL κατά τον προγραμματισμό μιας εργασίας.

Έξοδος TTL (TTL OUTPUTS) :

Η οδηγία αυτή ορίζει το επίπεδο που θα βρίσκεται μια έξοδος TTL μετά την εκτέλεση αυτής της γραμμής εντολής.

Στην ρομποτική, το επίπεδο TTL ορίζεται, επιλέγοντας το αντίστοιχο LED στο παράθυρο αποθήκευσης σημείου ελέγχου. Κατά τη χρήση του τερματικού σταθμού Hand-Held, το επίπεδο TTL έχει οριστεί στο παράθυρο εξόδου TTL του Teach μενού. Η εντολή εργασίας Έξοδος μπορεί να

χρησιμοποιηθεί επίσης για να ορίσετε το επίπεδο των εξόδων TTL κατά τον προγραμματισμό μιας εργασίας.

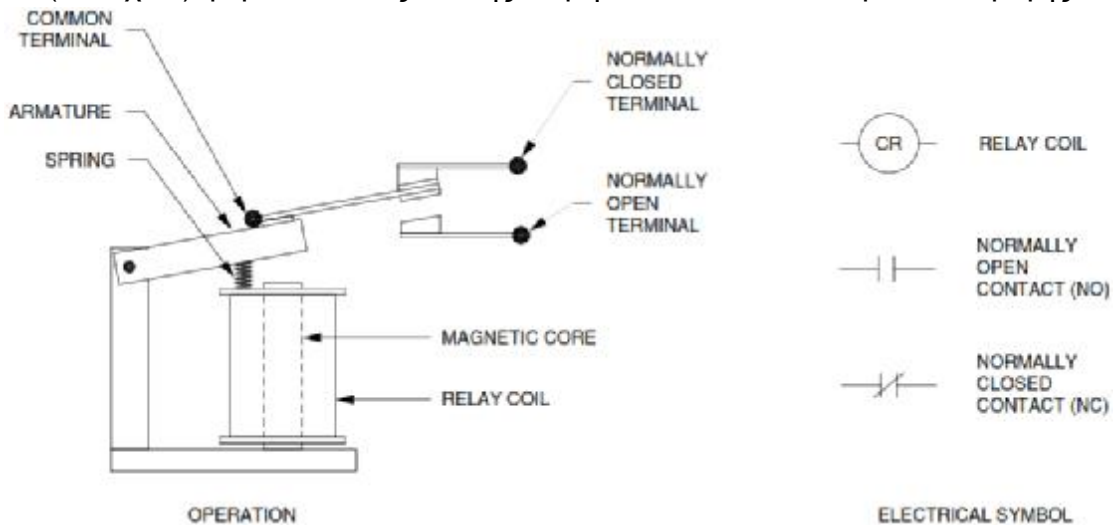


Εικόνα 2-8: Είσοδοι και έξοδοι του ελεγκτή ρομπότ

Επιβραδυντές (RELAYS) :

Η οδηγία αυτή ενεργοποιεί ή απενεργοποιεί, τα 4 πηνία αναμετάδοσης. Όπως φαίνεται στην εικόνα 2-9, ένα ρελέ αποτελείται από ένα πηνίο, ένα μαγνητικό πυρήνα, ένα πλαίσιο και ένα ή περισσότερα σύνολα από normally-open (συνήθως –ανοιχτές) (NO) και normally-closed (συνήθως-κλειστές) (NC) επαφές.

Όταν υπάρχει ροή ρεύματος μέσα στο πηνίο του ρελέ, ο μαγνητικός πυρήνας και το πλαίσιο έλκονται μεταξύ τους, προκαλώντας το πλαίσιο να προχωρήσει προς τον πυρήνα. Αυτή η ενέργεια αλλάζει τις επαφές του ρελέ στην ενεργοποιημένη κατάσταση. Η normally-open (NO) επαφή κλείνει, ενώ η normally-closed (NC) επαφή ανοίγει. Όταν το ρεύμα έχει καταργηθεί από το πηνίο του ρελέ, το πλαίσιο μετακινείται πίσω στην αρχική του θέση από ένα ελατήριο που επιτρέπει στις επαφές του ρελέ να απενεργοποιηθούν και να επιστρέψουν στην φυσιολογική τους κατάσταση. Οι επαφές ρελέ μπορούν να ελέγχουν τη λειτουργία των σημαντικών AC (εναλλασσόμενων) και DC(συνεχών) φορτίων όπως επίσης σειρήνων αλλά και στα φώτα διαφυγής.



Εικόνα 2-9: Σύστημα ρελέ

Στην ρομποτική, τα ρελέ (επιβραδυντές) ενεργοποιούνται ή απενεργοποιούνται, επιλέγοντας το αντίστοιχο LED στο παράθυρο αποθήκευσης σημείου ελέγχου. Η εντολή επιβράδυνσης (Relay) επίσης μπορεί να χρησιμοποιηθεί για τον έλεγχο των ρελέ κατά τον προγραμματισμό μιας εργασίας.

Καθυστέρηση(DELAY):

Η εντολή διακόπτει προσωρινά την εκτέλεση ενός προγράμματος. Στην ρομποτική, η καθυστέρηση έχει οριστεί στο παράθυρο αποθήκευσης σημείου ελέγχου. Η εντολή καθυστέρησης (Delay) μπορεί επίσης να χρησιμοποιηθεί για να ορίσετε μια καθυστέρηση κατά τον προγραμματισμό μιας εργασίας.

ΤΡΙΤΟ ΚΕΦΑΛΑΙΟ

ΤΡΙΤΟ ΚΕΦΑΛΑΙΟ

3.1 Περιγραφή εντολών:

3.1.1 CAROUSEL (Μύλος)

Σύνταξη: CAROUSEL(< ψηφίο 1-2 > , < ψηφίο 1-8 >)

Παράμετροι: < [HIGH | LOW] >

Περιγραφή: Η εντολή Carousel χρησιμοποιείται για τον έλεγχο της λειτουργίας του περιστροφικού μύλου. Προκαλεί το μύλο να περιστραφεί μια σειρά ή λιγότερο ανάλογα με την καθορισμένη τιμή εισόδου TTL που αντιστοιχεί σε ένα επιθυμητό επίπεδο. Οι τιμές σε παρένθεση δείχνουν το κανάλι (1 ή 2) όπου είναι συνδεδεμένος ο περιστροφικός μύλος και οι τιμές (1-8) την είσοδο TTL που διαβάζει. Η παράμετρος που ακολουθεί δείχνει το επιθυμητό επίπεδο της εισόδου TTL. Αυτή η εντολή θα αναστείλει την περιστροφή του μύλου αν ταιριάζει το επίπεδο που βρίσκεται με τις παράμετρους εντολής.

Σημείωση: Το <[HIGH | LOW]> μπορεί να αντικατασταθεί με <[ON | OFF]>, <[CLOSE| OPEN]>, ή <[1 | 0]>.

Παράδειγμα: 0001: CAROUSEL(1,1) HIGH
0002: MOVETO POINT1
0003: END

Η πρώτη γραμμή αναγκάζει το περιστρεφόμενο μύλο να περιστραφεί μια στροφή ή λιγότερο εάν το επίπεδο της εισόδου TTL 1 είναι υψηλό (HIGH). Τότε το πρόγραμμα εκτελεί την κίνησή του στο POINT1 και τερματίζει.

3.1.2 CNCINPUT (Είσοδος cnc)

Σύνταξη: CNCINPUT(< ψηφίο 1-2 > , < ψηφίο 1-4 >)

Παράμετροι: < [HIGH | LOW] >

Περιγραφή: Η εντολή αυτή διαβάζει την είσοδο του CNC 1 ή CNC 2. Ένα κανάλι συνήθως χρησιμοποιείται για να διασυνδεθεί άμεσα με έναν εξοπλισμό CNC η κάποια άλλη αυτοματοποίηση. Τα ψηφία στην παρένθεση δείχνουν το κανάλι

και την είσοδο αντίστοιχα: (1 ή 2), (1-4). Αυτή η εντολή θα εμποδίσει την εκτέλεση της εργασίας μέχρι να διαβάσει μια τιμή που να αντιστοιχεί στις παραμέτρους.

Σημείωση: Το <[HIGH | LOW]> μπορεί να αντικατασταθεί με <[ON | OFF]>, <[CLOSE | OPEN]>, ή <[1 | 0]>.

Παράδειγμα: 0001: CNCINPUT(1,2) HIG
0002: GOSUB LOAD_PART
0003: CNCOUTPUT(1,2) HIGH
0004: RESTART

Αυτό το πρόγραμμα παρουσιάζει έναν απλό βρόχο ελέγχου που μπορεί να χρησιμοποιηθεί με έναν εξοπλισμό CNC. Πρώτα, η γραμμή 1 περιμένει για το σήμα της CNC1, εισόδου 2 να γίνει υψηλό, τότε εκτελεί μια υπορουτίνα που φορτώνει ένα κομμάτι του αποθέματος στο μηχάνημα. Αφού φορτωθεί το απόθεμα, το CNC 1, εξόδου 2 (ή B) έχει οριστεί σε HIGH. Τέλος, πραγματοποιείται επανεκκίνηση της διαδικασίας.

3.1.3 CNCOUTPUT (Έξοδος cnc)

Σύνταξη: CNCOUTPUT (< ψηφίο 1 ή 2 > , < ψηφίο 1- 4 >)

Παράμετροι: < [HIGH | LOW] >

Περιγραφή: Η εντολή αυτή διαβάζει την έξοδο του CNC 1 ή CNC 2. Ένα κανάλι συνήθως χρησιμοποιείτε για να διασυνδεθεί άμεσα με έναν εξοπλισμό CNC η κάποια άλλη αυτοματοποίηση. Οι αριθμοί στην παρένθεση δείχνουν το κανάλι και την έξοδο αντίστοιχα: (1 ή 2), (1- 4 ή A μέχρι Δ). Η παράμετρος αυτή ελέγχει την έξοδο.

Σημείωση: <[HIGH | LOW]> μπορεί να αντικατασταθεί με <[ON | OFF]>, <[CLOSE|OPEN]>, ή <[1 | 0]>.

Παράδειγμα: 0001: CNCINPUT(1,2) HIGH
0002: GOSUB LOAD_PART
0003: CNCOUTPUT(1,2) HIGH
0004: RESTART

Αυτό το πρόγραμμα δείχνει έναν απλό βρόχο ελέγχου που μπορεί να χρησιμοποιηθεί με εξοπλισμό CNC. Πρώτα, η γραμμή 1 περιμένει το σήμα της CNC 1, εισόδου 2 να είναι υψηλή, τότε εκτελεί μια υπορουτίνα που φορτώνει ένα κομμάτι του αποθέματος στο μηχάνημα. Αφού φορτωθεί το απόθεμα, CNC 1, εξόδου 2 έχει οριστεί σε HIGH. Τέλος, πραγματοποιείται επανεκκίνηση της διαδικασίας.

3.1.4 DELAY (Καθυστέρηση)

Σύνταξη: DELAY

Παράμετροι: < αριθμός 0-999 >

Περιγραφή: Η εκτέλεση της εντολής αυτής εκφράζεται σε δευτερόλεπτα. Εάν η τιμή είναι 0 το πρόγραμμα σταματάει να λειτουργεί μέχρι να το ενεργοποιήσουμε εμείς από το αντίστοιχο κουμπί.

Παράδειγμα: 0001: MOVETO POINT1
0002: DELAY 10
0003: MOVETO POINT2
0004: DELAY 0
0005: HOME
0006: END

Το πρόγραμμα εκτελείται μέχρι το σημείο POINT1 και στη συνέχεια καθυστερεί για 10 δευτερόλεπτα. Στη συνέχεια εκτελείται μέχρι το σημείο POINT2 και σταματάει το πρόγραμμα. Τέλος ο χειριστής πατά το κουμπί έναρξης και το πρόγραμμα συνεχίζει στο HOME και σταματά τη λειτουργία του.

3.1.5 DO (Εκτέλεσε)

Σύνταξη: DO

Παράμετροι: < αριθμός 1-999>

Περιγραφή: Με αυτή την εντολή μπορούμε να εκτελέσουμε επαναλημμένα κάποιες λειτουργίες. Η εντολή αυτή δείχνει τον αριθμό των επαναλήψεων των εντολών που παρεμβαίνουν ανάμεσα στο DO και το UNTIL.

Παράδειγμα: 0001: DO 5
0002: IF INPUT(3) = LOW
0003: END
0004: ELSE
0005: MOVETO POINT1
0006: MOVETO POINT2
0007: MOVETO POINT3
0008: ENDIF
0009: UNTIL

Το πρόγραμμα αυτό εκτελεί μέχρι το POINT1, POINT2 και το POINT3, 5 φορές στη σειρά εκτός αν η έξοδος TTL3 έχει παράμετρο LOW, και στη συνέχεια εκτελείται η εντολή END. Αυτή η εντολή εκτελείται μέχρι ο χειριστής το θελήσει ή αν παρεμβαίνει μια εντολή END, RESTART ή κάποιο λάθος.

Βλέπε επίσης: εντολή UNTIL παρακάτω

3.1.6 DRIVETO (Πήγαινε προς)

Σύνταξη: DRIVETO

Παράμετροι: < Όνομα σημείου >

Περιγραφή: Η εντολή αυτή λειτουργεί ως συντόμευση αντί να επαναλαμβάνουμε την εντολή MOVETO. Για να εκτελέσουμε μια σειρά σημείων από τη λίστα, η εντολή αυτή θα εκτελέσει κάθε σημείο ανάμεσα στο τρέχον ενεργό σημείο και το σημείο με το όνομα της εντολής παραμέτρου. Τα σημεία θα εκτελεστούν με τη σειρά που υπάρχουν στη λίστα. Κανονικά, θα ξεκινήσει την εκτέλεση μιας μακράς ακολουθίας σημείων με την εντολή MOVETO από το πρώτο σημείο στη σειρά και στη συνέχεια θα ακολουθήσει η εντολή DRIVETO, που δηλώνει το τελευταίο σημείο της ακολουθίας. Αυτό μπορεί να απλοποιήσει το πρόγραμμα εργασιών, όταν έχουμε ένα μεγάλο αριθμό κινήσεων που πρέπει να εκτελεσθούν.

Προειδοποίηση: Η εκτέλεση των σημείων γίνεται με την σειρά που είναι στη λίστα. Αλλάζοντας τη λίστα των σημείων τροποποιείται ο τρόπος εκτέλεσης της εντολής.

Παράδειγμα: 0001: MOVETO POINT1

0002: DRIVETO POINT4

0003: END

Υποθέτοντας ότι το POINT1, POINT2, POINT3 και POINT4 είναι στη λίστα με αυτήν τη σειρά, το πρόγραμμα θα εκτελέσει πρώτα το POINT1. Στη συνέχεια η εντολή DRIVETO θα εκτελέσει κάθε σημείο μεταξύ του POINT1 μέχρι και του POINT4.

3.1.7 ELSE (Αλλιώς)

Σύνταξη: ELSE

Παράμετροι: Καμία

Περιγραφή: Η εντολή αυτή υποδηλώνει το τέλος των οδηγιών που εκτελούνται όταν μια συνθήκη είναι TRUE (αληθής) που αξιολογείται από μια προηγούμενη εντολή IF. Οι εντολές που ακολουθούν την εντολή ELSE εκτελούνται σε περίπτωση που η συσχετισμένη εντολή IF αξιολογείται ως FALSE (ψευδής).

Παράδειγμα: 0001: MOVETO POINT1

0002: MOVETO POINT2

0003: IF INPUT(1) = HIGH

0004: IF OUTPUT(2) = LOW

0005: MESSAGE Turn on the conveyor!

0006: ELSE

0007: GOSUB LOAD_FROM_CONVEYOR

0008: ENDIF

0009: ENDIF

0010: END

Αυτό το πρόγραμμα δείχνει τον τρόπο χρήσης της εντολής IF και επίσης δείχνει ότι μπορούμε να ενθέσουμε εντολές IF μέσα στις άλλες. Η Γραμμή 3 ελέγχει αν η είσοδος TTL 1 είναι υψηλή και εάν ναι, εκτελεί την επόμενη εντολή.

Η γραμμή 4 κατόπιν ελέγχει αν η έξοδος TTL 2 είναι χαμηλή. Εάν η εντολή IF εντοπίσει ότι η εξωτερική συσκευή της εξόδου TTL 2 είναι χαμηλή, εμφανίζεται το μήνυμα στη γραμμή 5. Αντίθετα, εάν η εντολή IF εντοπίσει ότι η εξωτερική συσκευή εξόδου TTL 2 είναι υψηλή, τότε το πρόγραμμα θα μεταπηδήσει στην γραμμή εντολής που ακολουθεί την εντολή ELSE, που ζητά την εκτέλεση στην πραγματική ρουτίνα φόρτωσης.

Βλέπε επίσης: εντολή IF, ENDIF

3.1.8 END (Τέλος)

Σύνταξη: END

Παράμετροι: Καμία

Περιγραφή: Αυτή η εντολή τερματίζει την εκτέλεση της τρέχουσας εργασίας. Εάν είναι επιλεγμένη η επανάληψη εργασίας στο παράθυρο εργασίας, το πρόγραμμα θα κάνει επανεκκίνηση εκτέλεσης της εργασίας κατά την έναρξη του προγράμματος. Αυτή η εντολή μπορεί να τοποθετηθεί οπουδήποτε στο πρόγραμμα εργασιών αλλά εάν υπάρχουν υπορουτίνες, οι εργασίες μετά την εντολή δεν θα εκτελεστούν ποτέ

Παράδειγμα: 0001: MOVETO POINT1

0002: MOVETO POINT2

0003: END

Βλέπε επίσης: MOVETO

3.1.9 ENDIF (Τέλος εάν)

Σύνταξη: ENDIF

Παράμετροι: Καμία

Περιγραφή: Η εντολή αυτή δηλώνει το τέλος μιας εργασίας IF-ELSE-ENDIF και η εκτέλεση συνεχίζεται με την επόμενη γραμμή εντολής. Για κάθε εντολή IF πρέπει να υπάρχει μία ακριβώς αντίστοιχη εντολή ENDIF. Εάν υπάρχουν πολλές εντολές IF, η καθεμία πρέπει να έχει μια αντίστοιχη εντολή ENDIF.

Παράδειγμα: 0001: MOVETO POINT1

0002: MOVETO POINT2

0003: IF INPUT(1) = HIGH

0004: IF OUTPUT(2) = LOW

0005: MESSAGE Turn on the conveyor!

0006: ELSE

0007: GOSUB LOAD_FROM_CONVEYOR

0008: ENDIF

0009: ENDIF

0010: END

Αυτό το πρόγραμμα δείχνει τον τρόπο χρήσης της εντολής IF, καθώς επίσης και ότι μπορείτε να ενθέσετε εντολές IF μέσα στις άλλες. Η γραμμή 3 ελέγχει αν η είσοδος TTL 1 είναι υψηλή και εάν ναι, εκτελεί την επόμενη γραμμή εντολής.

Η γραμμή 4 κατόπιν ελέγχει αν η έξοδος TTL 2 είναι χαμηλή. Εάν η εντολή IF εντοπίσει ότι η εξωτερική συσκευή της εξόδου TTL 2, εμφανίζεται το μήνυμα στη γραμμή 5. Αντίθετα, εάν η εντολή IF εντοπίσει ότι η εξωτερική συσκευή της εξόδου TTL 2 είναι υψηλή, τότε η εκτέλεση θα συνεχίσει στη γραμμή εντολής που ακολουθεί την εντολή ELSE, η οποία καλεί την εκτέλεση στην πραγματική ρουτίνα φόρτωσης.

Βλέπε επίσης: IF, ELSE

3.1.10 ERROR (Σφάλμα)

Σύνταξη: ERROR

Παράμετροι: < Συμβολοσειρά μηνυμάτων >

Περιγραφή: Αυτή η εντολή θα σταματήσει την εκτέλεση του προγράμματος και θα εμφανιστεί το μήνυμα της συμβολοσειράς ως παράμετρος στην περιοχή μηνυμάτων του παραθύρου εργασιών που εκτελούνται. Χρησιμοποιούμε αυτήν την εντολή, εάν θέλουμε να δημιουργήσουμε ένα σφάλμα στο πρόγραμμά ή εάν παρουσιαστεί μια απροσδόκητη συνθήκη.

Παράδειγμα: 0001: IF INPUT(1) = LOW
0002: ERROR No stock is available
0003: ELSE
0004: GOSUB LOAD_STOCK

0005: ENDIF

Αυτό το πρόγραμμα ελέγχει αν η τιμή εισόδου TTL 1 είναι χαμηλή αν η συνθήκη αληθής, το μήνυμα λάθους που εμφανίζεται στο πεδίο "μήνυμα" του παραθύρου εργασιών εκτελείτε και αναστέλλεται η εκτέλεση του προγράμματος.

3.1.11 GOSUB (Πήγαινε υπογείως)

Σύνταξη: GOSUB

Παράμετροι: < Όνομα σημείου >

Περιγραφή: Η εντολή αυτή καλεί μια επώνυμη υπορουτίνα. Η εκτέλεση θα επιστρέψει στην γραμμή εντολής που ακολουθεί την εντολή GOSUB όταν η εκτέλεση της υπορουτίνας ολοκληρωθεί. Μπορούμε να καλέσουμε υπορουτίνες μέσα από άλλες υπορουτίνες.

Παράδειγμα: 0001: IF INPUT(1) = HIGH

0002: MESSAGE Loading stock from conveyor.

0003: GOSUB LOAD_STOCK

0004: ENDIF

0005: END

0006: SUB LOAD_STOCK

0007: MOVETO POINT1

0008: MOVETO POINT2

0009: MOVETO POINT3

0010: RETURN

Αυτό το πρόγραμμα δείχνει πώς να ορίσετε και να καλέσετε υπορουτίνες. Εάν η τιμή εισόδου TTL 1 είναι υψηλή, τότε η ρουτίνα GOSUB μεταφέρει την εκτέλεση στη γραμμή εντολής που ακολουθεί την εντολή SUB. Κατόπιν εκτελούνται οι γραμμές 7 έως 9 και την εντολή RETURN της γραμμή 10 θα επιστρέψει την εκτέλεση πίσω στη γραμμή εντολής που ακολουθεί την εντολή GOSUB που ονομάζεται αυτή η υπορουτίνα.

Βλέπε επίσης: SUB, RETURN

3.1.12 HOME (Σπίτι)

Σύνταξη: HOME

Παράμετροι: Καμία

Περιγραφή: Η εντολή αυτή λέει στο Ρομποτικό βραχίονα να μετακινηθεί στην αρχική του θέση. Αυτή είναι η θέση που καθορίζεται από το σημείο που ονομάζεται HOME στη λίστα που σχετίζεται με αυτήν την εργασία. Είναι η θέση όπου το ρομπότ ξεκινά την κίνηση του και θα επιστρέψει να αναπαυτεί όταν δώσουμε την εντολή HOME. Αυτή η εντολή είναι ίδια με την εντολή MOVETO HOME.

Παράδειγμα: 0001: MOVETO POINT1

0002: HOME

0003: END

Το πρόγραμμα εκτελεί την εντολή POINT1 και στη συνέχεια επιστρέφει στην θέση HOME.

3.1.13 IF (Εάν)

Σύνταξη: IF

Παράμετροι: < Παράμετρος >

Περιγραφή: Αν η παράμετρος είναι TRUE (αληθής), η εκτέλεση συνεχίζεται κανονικά με την επόμενη γραμμή εντολής. Εάν η παράμετρος είναι FALSE (ψευδής), η εκτέλεση συνεχίζεται μετά την επόμενη εντολή ELSE (αν χρησιμοποιείται) ή την εντολή ENDIF.

Παράδειγμα: 0001: MOVETO POINT1

0002: MOVETO POINT2

0003: IF INPUT(1) = HIGH

0004: IF OUTPUT(2) = LOW

0005: MESSAGE Turn on the conveyer!

0006: ELSE

0007: GOSUB LOAD_FROM_CONVEYOR

0008: ENDIF

0009: ENDIF

0010: END

Αυτό το πρόγραμμα δείχνει τον τρόπο χρήσης της εντολής IF και επίσης δείχνει ότι μπορούμε να ενθέσουμε εντολές IF μέσα στις άλλες. Η Γραμμή 3 ελέγχει αν η είσοδος TTL 1 είναι υψηλή και εάν ναι, εκτελεί την επόμενη εντολή.

Η γραμμή 4 κατόπιν ελέγχει αν η έξοδος TTL 2 είναι χαμηλή. Εάν η εντολή IF εντοπίσει ότι η εξωτερική συσκευή της εξόδου TTL 2 είναι χαμηλή, εμφανίζεται το μήνυμα στη γραμμή 5. Αντίθετα, εάν η εντολή IF εντοπίσει ότι η εξωτερική συσκευή εξόδου TTL 2 είναι υψηλή, τότε το πρόγραμμα θα μεταπηδήσει στην γραμμή εντολής που ακολουθεί την εντολή ELSE, που ζητεί την εκτέλεση στην πραγματική ρουτίνα φόρτωσης.

Βλέπε επίσης: εντολή IF, ENDIF

3.1.14 INPUT (Είσοδος)

Σύνταξη: INPUT

Παράμετροι: INPUT (< Ψηφίο 1-8 >)

Περιγραφή: Αυτή η εντολή διαβάζει τις εισόδους TTL. Η τιμή στην παρένθεση δείχνει την είσοδο και η παράμετρος που ακολουθεί την εντολή, υποδεικνύει την επιθυμητή κατάσταση. Αυτή η εντολή θα μπλοκάρει και θα αναστείλει την εκτέλεση των εργασιών μέχρι η τιμή που θα αναγνωστεί να ταιριάζει με την εντολή παραμέτρου.

Σημείωση: Το <[HIGH | LOW]> μπορεί να αντικατασταθεί με <[ON | OFF]>, <[CLOSE| OPEN]>, ή <[1 | 0]>.

Παράδειγμα: 0001: MOVETO POINT1

0002: INPUT(4) LOW

0003: MOVETO POINT2

0004: END

Αυτό το πρόγραμμα εκτελεί την εντολή POINT1 και στη συνέχεια αναμένει μέχρι η τιμή εισόδου της TTL 4 είναι χαμηλή. Τότε εκτελείται η εντολή POINT2 και το πρόγραμμα τελειώνει.

3.1.15 LATHE (Διατομή)

Σύνταξη: LATHE

Παράμετροι: <συμβολοσειρά ονομάτων αρχείων>

Περιγραφή: Αυτή η εντολή εκτελεί το εξωτερικό πρόγραμμα που σχετίζεται με την εντολή της LATHE που ορίζεται στο παράθυρο επιλογές. Η παράμετρος των ονομάτων αρχείων είναι μια παράμετρος σε αυτό το πρόγραμμα, έτσι ώστε το λογισμικό CNC να κάνει αυτόματη λήψη του αρχείου σε ένα CNC LATHE.

Παράδειγμα: 0001: GOSUB LOAD_STOCK \Load stock into CNC Lathe

0002: LATHE sample_part.I4 \Download the part program to the Lathe

0003: CNCINPUT(1,1) HIGH \Wait for the CNC Lathe to signal completion

0004: END

Αυτό το πρόγραμμα δείχνει τον τρόπο χρήσης της εντολής LATHE για να κάνετε λήψη ενός CNC LATHE προγράμματος. Αυτή η εντολή χρησιμοποιείται κατά την ολοκλήρωση σύνδεσης του συστήματος σέρβο-ρομπότ με μηχανήματα CNC LATHE.

3.1.16 MESSAGE (Μήνυμα)

Σύνταξη: MESSAGE

Παράμετροι: < συμβολοσειρά μηνύματος >

Περιγραφή: Αυτή η εντολή θα εμφανίσει τα δεδομένα της συμβολοσειράς παραμέτρων που εκτελούνται στο πεδίο "μήνυμα" του παραθύρου εργασιών. Η εντολή αυτή δεν θα σταματήσει την εκτέλεση του προγράμματος ωστόσο αν χρειαστεί ο χρήστης να απαντήσει στο μήνυμά, χρησιμοποιούμε αυτή την εντολή με ΚΑΘΥΣΤΕΡΗΣΗ 0.

Παράδειγμα: 001: IF INPUT(1) = HIGH

002: MESSAGE Loading stock from conveyor
(φόρτωση αποθέματος από ιμάντες)

003: GOSUB LOAD_STOCK

004: ENDIF

Αυτό το πρόγραμμα ελέγχει εάν η τιμή της εισόδου TTL1 είναι υψηλή. Εάν ο όρος βρεθεί αληθής, αποστέλλεται ένα μήνυμα στο χρήστη στο πεδίο "μήνυμα" του παραθύρου εργασιών που εκτελούνται και εκτελείται η υπορουτίνα LOAD_STOCK.

3.1.17 MILL

Σύνταξη: MILL

Παράμετροι: < συμβολοσειρά ονομάτων φακέλων >

Περιγραφή: Αυτή η εντολή εκτελεί το εξωτερικό πρόγραμμα που σχετίζεται με την ίδια την εντολή MILL που έχει οριστεί στο παράθυρο επιλογών. Όταν εμφανιστεί η παράμετρος ονόματος αρχείου το λογισμικό CNC αυτόματα θα κατεβάσει το αρχείο σε ένα CNC MILL.

Παράδειγμα: 0001: GOSUB LOAD_STOCK \Load stock into CNC Mill.
0002: MILL sample_part.m4 \Download the part program to the Mill.
0003: CNCINPUT(1,1) HIGH \Wait for the Mill to signal completion.
0004: END

Αυτό το πρόγραμμα δείχνει τον τρόπο χρήσης της εντολής MILL για να κατεβάσουμε ένα πρόγραμμα CNC MILL. Αυτή η εντολή χρησιμοποιείται κατά την ολοκλήρωση σύνδεσης του συστήματος σέρβο-ρομπότ με μηχανήματα CNC MILL.

3.1.18 MOVETO (Πήγαινε προς)

Σύνταξη: MOVETO

Παράμετροι: < συμβολοσειρά ονόματος σημείου >

Περιγραφή: Η εντολή αυτή εκτελεί είτε ένα σημείο θέσης ή ένα σημείο ελέγχου.

Παράδειγμα: 0001: MOVETO POINT1
0002: END

Το πρόγραμμα εκτελεί το σημείο POINT1 και στη συνέχεια τερματίζει την λειτουργία του.

3.1.19 OUTPUT (Έξοδος)

Σύνταξη: OUTPUT (< ψηφίο 1-8 >)

Παράμετροι: < [HIGH | LOW] >

Περιγραφή: Η εντολή αυτή ελέγχει την κατάσταση των εξόδων TTL. Η τιμή στην παρένθεση υποδηλώνει την έξοδο για την αλλαγή και η παράμετρος που ακολουθεί το νέο καθεστώς της εξόδου.

Σημείωση: Η <[HIGH | LOW]> μπορεί να αντικατασταθεί με <[ON | OFF]>, <[CLOSE|OPEN]>, ή <[1 | 0]>.

Παράδειγμα: 0001: OUTPUT(1) HIGH
0002: DELAY 5
0003: OUTPUT(1) LOW
0004: END

Αυτό το πρόγραμμα ορίζει την έξοδο TTL 1 υψηλή, καθυστερεί για 5 δευτερόλεπτα και στη συνέχεια ρυθμίζει την έξοδο σε χαμηλή.

3.1.20 RELAY (Επιβράδυνση)

Σύνταξη: RELAY (< ψηφίο 1-4 >)

Παράμετροι: < [CLOSE | OPEN] >

Περιγραφή: Η εντολή αυτή ελέγχει την κατάσταση του ρελέ. Η τιμή στην παρένθεση δείχνει ποιο ρελέ ελέγχεται και ποια παράμετρος ακολουθεί: ενεργοποιημένη (κλείνει) ή απενεργοποιημένη (ανοίγει).

Σημείωση: Η <[HIGH | LOW]> μπορεί να αντικατασταθεί με <[ON | OFF]>, <[CLOSE|OPEN]>, ή <[1 | 0]>

Παράδειγμα: 0001: RELAY(1) CLOSE
0002: DELAY 5
0003: RELAY(1) OPEN
0004: END

Το πρόγραμμα αυτό ενεργοποιεί το Relay 1, καθυστερεί 5 δευτερόλεπτα και στη συνέχεια απενεργοποιεί το ρελέ.

3.1.21 REPEAT (Επανάληψη)

Σύνταξη: REPEAT

Παράμετροι: < Καμία >

Περιγραφή: Δηλώνει το τέλος ενός βρόχου WHILE. Όταν ο βρόχος φτάσει σε αυτήν την εντολή, η εκτέλεση θα μεταπηδήσει πίσω στο συσχετισμένο WHILE για να γίνει επανεκτίμηση της πρωτότυπης έκφρασης WHILE.

Παράδειγμα: 0001: MOVETO POINT1
0002: WHILE INPUT(4) = HIGH \ Stock is available from
feeder.
0003: GOSUB LOAD_STOCK
0004: OUTPUT(2) LOW
0005: GOSUB MOVE_STOCK
0006: OUTPUT(2) HIGH
0007: REPEAT
0008: END

Αυτό το πρόγραμμα δείχνει πώς μπορεί να γίνεται τροφοδοσία από ένα άλλο σημείο. η εργασία αυτή επαναλαμβάνεται μέχρις ότου η είσοδος TTL 4 είναι χαμηλή. Όσο η είσοδος TTL 4 παραμένει υψηλή, οι εντολές μεταξύ της WHILE και REPEAT θα εκτελούνται επανειλημμένα.

Βλέπε επίσης: WHILE

3.1.22 RESTART (Επανεκκίνηση)

Σύνταξη: RESTART

Παράμετροι: < Καμία >

Περιγραφή: Αυτή η εντολή καθοδηγεί το λογισμικό για να ματαιώσει την τρέχουσα εκτέλεση της εργασίας και να ξεκινήσει πάλι από την αρχή του προγράμματος. Σε αντίθεση με τις εντολές *END*, *RESTART* αγνοεί το *Repeat* κατά την εκτέλεση εργασιών.

Παράδειγμα: 0001: MOVETO POINT1
0002: MOVETO POINT2
0003: RESTART
0004: MOVETO POINT3
0005: END

Σε αυτό το πρόγραμμα οι εντολές των σειρών 4 και 5 δεν θα εκτελεστούν ποτέ.

3.1.23 RETURN (Επιστροφή)

Σύνταξη: RETURN

Παράμετροι: < Καμία >

Περιγραφή: Αυτή η εντολή ορίζει το τέλος μιας υπορουτίνας. Όταν εκτελεστεί η εντολή αυτή, εξέρχεται από την υπορουτίνα που εκτελείται και η εκτέλεση θα συνεχιστεί στη γραμμή εντολής που ακολουθεί την εντολή GOSUB. Επιτρέπεται μόνο μία εντολή RETURN σε μια υπορουτίνα και πρέπει να είναι η τελευταία εντολή στην υπορουτίνα.

Παράδειγμα: 0001: IF INPUT(1) = HIGH
0002: MESSAGE Loading stock from the conveyor.
0003: GOSUB LOAD_STOCK
0004: ENDIF
0005: END
0006: SUB LOAD_STOCK
0007: MOVETO POINT1
0008: MOVETO POINT2
0009: MOVETO POINT3
0010: RETURN

Αυτό το πρόγραμμα δείχνει πώς να ορίσουμε και να καλέσουμε υπορουτίνες. Εάν η τιμή εισόδου TTL 1 είναι υψηλή, τότε η ρουτίνα GOSUB μεταφέρει την εκτέλεση της γραμμής εντολής που ακολουθεί την εντολή SUB. Εκτελούνται κατόπιν οι εντολές των γραμμών 7 έως 9 και την εντολή RETURN στη γραμμή 10 και θα επιστρέψει την εκτέλεση πίσω στη γραμμή εντολής που ακολουθεί την αρχική εντολή GOSUB.

Βλέπε επίσης: GOSUB

3.1.24 RUNTO (Εκτέλεσε έως)

Σύνταξη: RUNTO

Παράμετροι: < συμβολοσειρά ονόματος σημείου >

Περιγραφή: Αυτή η εντολή λέει στο σέρβο-ρομπότ να εκτελέσει είτε ένα σημείο θέσης είτε ένα στοιχείο ελέγχου. Εάν το στοιχείο είναι ένα σημείο θέσης, η εντολή θα εκτελεστεί με τη μέγιστη ταχύτητα. Η τιμή της ταχύτητας που ορίστηκε για το σημείο παραβλέπεται.

Παράδειγμα: 0001: MOVETO POINT1
0002: RUNTO POINT2

0002: END

Αυτό το πρόγραμμα θα εκτελέσει την εντολή POINT1 και POINT2. Αν η εντολή POINT2 δεν είναι ένα σημείο ελέγχου, η ταχύτητα θα οριστεί στο 99 (μέγιστη τιμή), παρακάμπτοντας την τιμή ταχύτητας που έχει αποθηκευτεί στη θέση.

Βλέπε επίσης: MOVETO, END

3.1.25 SPEED (Ταχύτητα)

Σύνταξη: SPEED

Παράμετροι: < ψηφίο 0-99 >

Περιγραφή: Η εντολή αυτή μας επιτρέπει να ορίσουμε την τιμή της ταχύτητας για τις γραμμές οδηγίων που ακολουθούν τη γραμμή που περιέχει αυτήν την εντολή. Όλες οι κινήσεις θα εκτελεστούν σε αυτήν την νέα ταχύτητα. Οι εντολές που έχουν σαν τιμή ταχύτητας την τιμή 0 θα έχουν τη δικιά τους προκαθορισμένη ταχύτητα.

Παράδειγμα: 0001: MOVETO POINT1
0002: MOVETO POINT2
0003: SPEED 20
0004: MOVETO POINT1
0005: SPEED 0
0006: MOVETO POINT2
0007: END

Σε αυτό το πρόγραμμα, οι δύο πρώτες γραμμές οδηγίων μετακινούν το σέρβο-ρομπότ στο POINT1 και POINT2. Η ταχύτητα στη γραμμή 3 σημαίνει ότι η εντολή MOVETO στη γραμμή 4 θα προχωρήσει με ταχύτητα 20 αντί της προκαθορισμένης ταχύτητας. Η εντολή της γραμμής 5 θα απενεργοποιήσει την ταχύτητα που έχουμε θέσει στη γραμμή 3 και έτσι θα μετακινηθεί το MOVETO στη γραμμή 6 με την ταχύτητα του POINT2. Αυτό το παράδειγμα υποθέτει ότι το POINT1 και POINT2 δεν τα σημεία ελέγχου. Ο καθορισμός της τιμής της ταχύτητας δεν έχει καμία επίδραση στα σημεία ελέγχου.

Βλέπε επίσης: MOVETO, END

3.1.26 SUB (Υπό)

Σύνταξη: SUB

Παράμετροι: < όνομα συμβολοσειράς >

Περιγραφή: Αυτή η εντολή ορίζει το όνομα και τη θέση έναρξης μιας υπορουτίνας. Όλες οι γραμμές οδηγιών που ακολουθούν αυτή την εντολή ανήκουν στην επώνυμη υπορουτίνα μέχρι την εντολή RETURN που δηλώνει το τέλος της υπορουτίνας. Οι υπορουτίνες πάντα θα πρέπει να ορίζονται μετά την εντολή END και ξεχωριστά η μια από την άλλη. Οι υπορουτίνες μπορούν να καλέσουν άλλες υπορουτίνες, αλλά πρέπει να γνωρίζουμε ότι ένας ανεπιθύμητος βρόχος μπορεί να προκύψει.

Παράδειγμα: 0001: IF INPUT(1) = HIGH
0002: MESSAGE Loading stock from conveyor.
0003: GOSUB LOAD_STOCK
0004: ENDIF
0005: END
0006: SUB LOAD_STOCK
0007: MOVETO POINT1
0008: MOVETO POINT2
0009: MOVETO POINT3
0010: RETURN

Αυτό το πρόγραμμα δείχνει πώς να ορίσουμε και να καλέσουμε υπορουτίνες. Εάν η τιμή εισόδου TTL 1 είναι υψηλή, τότε η ρουτίνα GOSUB μεταφέρει την εκτέλεση της γραμμής εντολής που ακολουθεί την εντολή SUB. Εκτελούνται κατόπιν οι εντολές των γραμμών 7 έως 9 και την εντολή RETURN στη γραμμή 10 και θα επιστρέψει την εκτέλεση πίσω στη γραμμή εντολής που ακολουθεί την αρχική εντολή GOSUB.

Βλέπε επίσης: GOSUB, RETURN

3.1.27 UNTIL (Έως)

Σύνταξη: UNTIL

Παράμετροι: < Καμία >

Περιγραφή: Η εντολή αυτή δηλώνει το τέλος μιας ακολουθίας βρόχου και εάν ο αριθμός των περασμάτων μέσα στο βρόχο είναι μικρότερος από την παράμετρο εντολής. Η εκτέλεση μεταπηδά στην αρχική εντολή.

Εάν ο αριθμός των περασμάτων είναι μεγαλύτερος από τον αριθμό της εντολής, συνεχίζει την εκτέλεση με τη γραμμή εντολής που ακολουθεί την εντολή UNTIL.

Παράδειγμα: 0001: DO 5
0002: IF INPUT(3) = LOW
0003: END
0004: ELSE
0005: MOVETO POINT1
0006: MOVETO POINT2
0007: MOVETO POINT3
0008: ENDIF
0009: UNTIL

Το πρόγραμμα αυτό εκτελεί μέχρι το POINT1, POINT2 και το POINT3, 5 φορές στη σειρά εκτός αν η έξοδος TTL3 έχει παράμετρο LOW, και στη συνέχεια εκτελείται η εντολή *END*. Αυτή η εντολή εκτελείται μέχρι ο χειριστής το θελήσει ή αν παρεμβαίνει μια εντολή *END*, *RESTART* ή κάποιο λάθος.

Βλέπε επίσης: DO

3.1.28 WHILE (Μέχρι)

Σύνταξη: WHILE

Παράμετροι: <Παράμετρος >

Περιγραφή: Αυτή η εντολή αποτιμά μία παράμετρο. Αν είναι αληθής, εκτελεί τις γραμμές οδηγιών που ακολουθούν, μέχρι να την εντολή *REPEAT*. Όταν επιτευχθεί η εντολή *REPEAT*, η εκτέλεση θα επιστρέψει στην εντολή *WHILE* όπου η έκφραση θα επαναλαμβάνετε έως ότου αποτύχει.

Σημείωση: Η τιμή της έκφρασης αξιολογείται μόνο από την εντολή *WHILE*. Αλλαγή της κατάστασης των εισόδων και εξόδων μέσα σε αυτήν την εντολή, δεν θα προκαλέσει την άμεση έξοδο του βρόχου.

Παράδειγμα: 0001: MOVETO POINT1

0002: WHILE INPUT(4) = HIGH \Stock is available from feeder

0003: GOSUB LOAD_STOCK

0004: OUTPUT(2) LOW

0005: GOSUB MOVE_STOCK

0006: OUTPUT(2) HIGH

0007: REPEAT

0008: END

Αυτό το πρόγραμμα δείχνει πώς μπορεί να γίνετε τροφοδοσία από ένα άλλο σημείο. η εργασία αυτή επαναλαμβάνεται μέχρις ότου η είσοδος TTL 4 είναι χαμηλή. Όσο η είσοδος TTL 4 παραμένει υψηλή ενώ η εντολή WHILE εκτελείται, οι εντολές μεταξύ της WHILE και REPEAT θα εκτελούνται επανειλημμένα.

Βλέπε επίσης: REPEAT

3.2 ΕΝΤΟΛΕΣ

ΕΝΤΟΛΕΣ-ΣΥΝΤΟΜΟΣ ΠΙΝΑΚΑΣ		
Εντολή	Παράμετρος	Λειτουργία
CAROUSEL	<[0 1]>	Ξεκινά ο μύλος στο κανάλι 1 ή 2, περιμένει την είσοδο TTL αν η τιμή της είναι υψηλή (1) ή χαμηλή (0) και στη συνέχεια σταματά τον μύλο.
CNCINPUT(x,y)	<[0 1]>	Περιμένει για το CNC (1 ή 2) και για την γραμμή αν είναι υψηλή (1) ή χαμηλή (0).
CNCOUTPUT(x,y)	<[0 1]>	Θέτει το CNC (1 ή 2) και την γραμμή αν θα είναι υψηλή (1) ή χαμηλή (0).
DELAY	<0-999>	Καθυστερεί την εκτέλεση.
DO	<1-9999>	Επαναλαμβάνει μια σειρά από εντολές.
DRIVETO	<σημείο>	Εκτελεί μια ακολουθία σημείων.
ELSE	καμία	Εκτελεί μια άλλη παράμετρο αν μια συνθήκη είναι ψευδής.
END	καμία	Σταματάει την διαδικασία.
ENDIF	καμία	Σταματάει μια συνθήκη IF.
ERROR	<μήνυμα>	Σταματάει την εκτέλεση και εμφανίζει ένα μήνυμα λάθους.
GOSUB	<όνομα>	Η εκτέλεση μεταφέρεται σε μια υπορουτίνα.
HOME	καμία	Μετακινείται ο ρομποτικός βραχίονας στην αρχική του θέση.
IF	<Παράμετρος >	Εκτελεί μια εντολή υπό όρους.
INPUT(x)	<[0 1]>	Περιμένει αν η είσοδος TTL είναι υψηλή (1) ή χαμηλή (0).
LATHE	<ονόματα φακέλων>	Κατεβάζει ένα πρόγραμμα CNC LATHE.
MESSAGE	<μήνυμα>	Εμφανίζει ένα μήνυμα στο παράθυρο εντολών.
MILL	<ονόματα φακέλων>	Κατεβάζει ένα πρόγραμμα CNC MILL.
MOVETO	<σημείο>	Εκτελεί μια εντολή προς ένα σημείο.
OUTPUT(x)	<[0 1]>	Θέτει την έξοδο TTL υψηλή (1) ή χαμηλή (0).
RELAY(x)	<[0 1]>	Ενεργοποιεί ή απενεργοποιεί το ρελέ.
REPEAT	καμία	Τερματίζει τον βρόγχο WHILE.
RESTART	καμία	Ξεκινά από την αρχή το πρόγραμμα.
RETURN	καμία	Τερματίζει μια υπορουτίνα.
RUNTO	<σημείο>	Εκτελεί μια εντολή στη μέγιστη ταχύτητα.
SPEED	<0-99>	Θέτει μια γενική ταχύτητα για τις μετέπειτα εντολές.
SUB	<όνομα>	Καθορίζει την έναρξη μιας υπορουτίνας.
UNTIL	καμία	Τερματίζει έναν βρόγχο DO.
WHILE	<Παράμετρος >	Ξεκινά ένας βρόγχος όταν μια κατάσταση είναι αληθής.

ΤΕΤΑΡΤΟ ΚΕΦΑΛΑΙΟ

ΤΕΤΑΡΤΟ ΚΕΦΑΛΑΙΟ

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παραθέσουμε διάφορες μικρές εφαρμογές που εκτελούνται προγραμματίζοντας τον ρομποτικό μας βραχίονα, θα δούμε τις εντολές, που και πως χρησιμοποιούνται μέσα από τους τρόπους που μπορούμε να γράψουμε ένα πρόγραμμα.

4.2 Τρόποι γραφείς των προγραμμάτων

4.2.1 Εντολές κειμένου

Ο πρώτος και πιο σημαντικός τρόπος γραφής ενός προγράμματος. Οι εντολές γράφονται σαν κείμενο με λέξεις κλειδιά στο εκάστοτε πρόγραμμα και καθώς ο ελεγκτής διαβάζει τις εντολές, ο βραχίονας τις εκτελεί. Σε προηγούμενο κεφάλαιο έχουμε αναλύσει τις εντολές προγραμματισμού που υπάρχουν και πως συντάσσεται η κάθε μία. Παρακάτω παραθέτουμε ένα μικρό στοιχειώδες πρόγραμμα.

```
001: Home
002: Speed 20
003: Moveto point1
004: Open grip
005: Moveto point2
006: Close grip
007: Speed 10
008: Moveto point 3
009: Open grip
010: Moveto point 4
011: Close grip
012: Speed 30
013: Home
014: End
```

4.2.2 Γραφήματα ροής:

Τα γραφήματα ροής χρησιμοποιούνται για να απεικονίσουμε τις διάφορες διεργασίες. Στα γραφήματα παρουσιάζεται η ροή ή η μετακίνηση από το ένα βήμα στο άλλο και η σχέση μεταξύ των διαφόρων τμημάτων μιας διαδικασίας.

Σε ένα τυπικό διάγραμμα ροής όπως αυτό που φαίνεται στο σχήμα 5-3, η έναρξη δραστηριότητας είναι στην κορυφή, ή στην επάνω αριστερή γωνία και η ροή είναι προς τα κάτω ή προς τα δεξιά, τελειώνει στο κάτω μέρος ή στην κάτω δεξιά γωνία του γραφήματος ροής.

- Τα τετράγωνα και τα ορθογώνια συνήθως αντιπροσωπεύουν τις δραστηριότητες ή τα βήματα, μιας διαδικασίας.
- Τα διαμάντια υποδεικνύουν σημεία απόφασης που απαιτούν συνήθως μια απλή απάντηση Ναι ή όχι. Χρειάζονται τρεις ή τέσσερις γραμμές σύνδεσης, μια εισερχόμενη και δύο ή τρεις εξερχόμενες για ναι, όχι, ή άλλες κατάλληλες απαντήσεις.
- Οι κύκλοι είναι σημεία μεταφόρτωσης που δείχνει τη ροή από ένα τμήμα του γραφήματος σε ένα άλλο, ή σε άλλο γράφημα. Ένα σχήμα που μοιάζει με ένα ορθογώνιο με στρογγυλεμένες άκρες είναι μια αρχή ή ένας τερματισμός.

Για παράδειγμα, ας υποθέσουμε ότι είναι επιθυμητό να σχεδιάσουμε ένα διάγραμμα ροής για το ακόλουθο πρόγραμμα:

Ένα ρομπότ, παραλαμβάνει ένα αντικείμενο από ένα τροφοδοτικό και το τοποθετεί σε έναν ιμάντα μεταφοράς. Αφού το αντικείμενο βρίσκεται πάνω στον ιμάντα, αυτός αρχίζει να κινείται και συνεχίζει έως ότου το αντικείμενο να ανιχνευτεί από έναν αισθητήρα στην έξοδο του ιμάντα. Στη συνέχεια ο ιμάντας μεταφοράς σταματά.

Το πρόγραμμα περιλαμβάνει τα ακόλουθα βήματα:

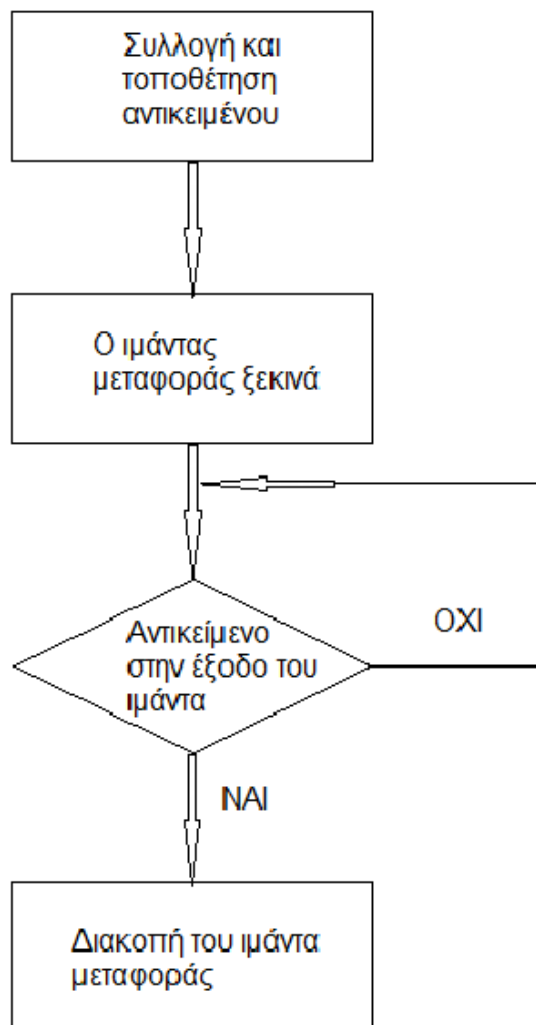
- Το ρομπότ παίρνει το αντικείμενο από το τροφοδοτικό και το τοποθετεί στο ιμάντα μεταφοράς.
- Η μεταφορά του αντικειμένου ξεκινά.
- Το ρομπότ ελεγκτής (αισθητήρας) ελέγχει εάν το αντικείμενο υπάρχει στην έξοδο του ιμάντα.
- Εάν εντοπιστεί το αντικείμενο, ο ιμάντας μεταφοράς σταματά.

Οι οδηγίες του προγράμματος που σχετίζεται με κάθε μπλοκ με το διάγραμμα ροής είναι ως εξής:

- [συλλογή και τοποθέτηση του αντικειμένου]: Θέση σημείου. Το ρομπότ έχει τοποθετηθεί στο σημείο που απαιτείται για τη συλλογή και την τοποθέτηση του αντικειμένου στον ιμάντα μεταφοράς. Καταγράφονται όλα τα σημαντικά σημεία.

- [Ο ιμάντας μεταφοράς ξεκινά]: Μια εντολή TTL εξόδου χρησιμοποιείται για να ορίσουμε την έξοδο TTL όπου ο ιμάντας είναι συνδεδεμένος, στο επίπεδο που απαιτείται ώστε να ξεκινήσει ο ιμάντας.
- [Αντικείμενο στην έξοδο του ιμάντα]: Μια εντολή TTL εισόδου χρησιμοποιείται. Ένας ανιχνευτής ελέγχει το επίπεδο που βρίσκετε μια είσοδο TTL στον ελεγκτή ρομπότ. Ανάλογα με το επίπεδο της εισόδου TTL, το πρόγραμμα εκτελεί την επόμενη εντολή (διακοπή της λειτουργίας του ιμάντα μεταφοράς) ή να περιμένει έως ότου εντοπιστεί ένα αντικείμενο.
- [Διακοπή του ιμάντα μεταφοράς]: Μια εντολή TTL εξόδου χρησιμοποιείται για να ορίσουμε το επίπεδο της εξόδου TTL, όπου ο ιμάντας είναι συνδεδεμένος, που απαιτείται για να σταματήσει.

Στην εικόνα4-1 φαίνεται το διάγραμμα ροής του παραπάνω προγράμματος.












Εικόνα 4-1: Διάγραμμα ροής προγράμματος

4.2.3 Εντολές με την χρήση εικονιδίων

Τέλος ο πιο πρόσφατος και πλέον διαδεδομένος τρόπος γραφής ενός προγράμματος είναι με την χρήση εικονιδίων. Είναι ένας απλός αλλά και πολύ γρήγορος τρόπος που μας βοηθάει στο προγραμματισμό εκμηδενίζοντας σχεδόν κάποια πιθανά λάθη που θα μπορούσαν να γίνουν φτιάχνοντας ένα πρόγραμμα με τις εντολές κειμένου.


Τέσσερα αντικείμενα βρίσκονται σε συγκεκριμένα σημεία στο εύρος του βραχίονα. Μόλις αρχίσει το πρόγραμμα ο βραχίονας παίρνει το πρώτο αντικείμενο και το τοποθετεί πάνω σε ένα στήριγμα συλλογής. Αυτό επαναλαμβάνεται και για τα άλλα τρία τουβλάκια. Μόλις ολοκληρωθεί η συλλογή τους στο στήριγμα, αυτό μετακινείται σε ένα επιθυμητό σημείο από τον βραχίονα. Τα παρακάτω σχήματα μας δείχνουν τον τρόπο γραφής του προγράμματος με την βοήθεια των εντολών εικονιδίων.

001		HOME
002		DELAY (s) = <input type="text" value="5"/>
003		SPEED = <input type="text" value="60"/>
004		GRIP = <input type="text" value="<-- -->"/>
005		MOVETO = <input type="text" value="pano_touvlak"/>
006		SPEED = <input type="text" value="20"/>
007		MOVETO = <input type="text" value="touvlaki_1"/>
008		SPEED = <input type="text" value="30"/>
009		GRIP = <input type="text" value="--> <--"/>




Ο βραχίονας έχει μεταφερθεί στην πρώτη θέση για να παραλάβει το αντικείμενο.

010		MOVETO = <input type="text" value="pano_touvlak"/>
011		SPEED = <input type="text" value="80"/>
012		MOVETO = <input type="text" value="koino_shmeid"/>
013		SPEED = <input type="text" value="20"/>
014		MOVETO = <input type="text" value="t1_vasi_pano"/>
015		MOVETO = <input type="text" value="t1_vasi"/>
016		SPEED = <input type="text" value="40"/>
017		GRIP = <input type="text" value="<-- -->"/>
018		MOVETO = <input type="text" value="t1_vasi_pano"/>












Το πρώτο αντικείμενο τοποθετείτε στην κατάλληλη θέση.

019		SPEED = <input type="text" value="80"/>
020		MOVETO = <input type="text" value="koino_shmeid"/>
021		MOVETO = <input type="text" value="touvlaki_2_pa"/>
022		SPEED = <input type="text" value="20"/>
023		MOVETO = <input type="text" value="touvlaki_2"/>
024		SPEED = <input type="text" value="30"/>
025		GRIP = <input type="text" value="--> <--"/>
026		MOVETO = <input type="text" value="touvlaki_2_pa"/>
027		SPEED = <input type="text" value="80"/>












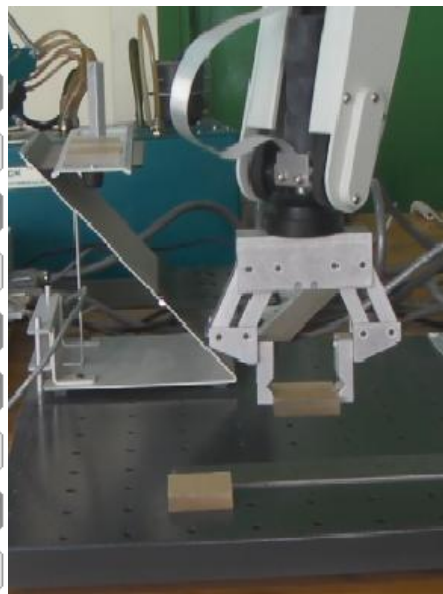
Ο ρομπότικος βραχίονας συλλέγει το δεύτερο αντικείμενο.

028		MOVETO = <input type="text" value="koino_shmeic"/>
029		SPEED = <input type="text" value="20"/>
030		MOVETO = <input type="text" value="t2_vasi_pano"/>
031		MOVETO = <input type="text" value="t2_vasi"/>
032		SPEED = <input type="text" value="40"/>
033		GRIP = <input type="text" value="<-- -->"/>
034		MOVETO = <input type="text" value="t2_vasi_pano"/>
035		SPEED = <input type="text" value="80"/>
036		MOVETO = <input type="text" value="koino_shmeic"/>



Ο βραχίονας βρίσκεται στην κατάλληλη θέση για να τοποθετήσει το δεύτερο αντικείμενο στην κατάλληλη θέση.

037		MOVETO = <input type="text" value="touvlaki_3_pa"/>
038		SPEED = <input type="text" value="20"/>
039		MOVETO = <input type="text" value="touvlaki_3"/>
040		SPEED = <input type="text" value="30"/>
041		GRIP = <input type="text" value="--> <--"/>
042		MOVETO = <input type="text" value="touvlaki_3_pa"/>
043		SPEED = <input type="text" value="80"/>
044		MOVETO = <input type="text" value="koino_shmeic"/>
045		SPEED = <input type="text" value="20"/>



Ο βραχίονας συνέλεξε το τρίτο αντικείμενο.

046		MOVETO = t3_vasi_pano	
047		MOVETO = t3_vasi	
048		SPEED = 40	
049		GRIP = <-- -->	
050		MOVETO = t3_vasi_pano	
051		SPEED = 80	
052		MOVETO = koino_shmeic	
053		MOVETO = touvlaki_4_pa	
054		SPEED = 20	

Το τρίτο αντικείμενο βρίσκεται στη θέση που έπρεπε.

055		MOVETO = touvlaki_4	
056		SPEED = 30	
057		GRIP = --> <--	
058		MOVETO = touvlaki_4_pa	
059		SPEED = 80	
060		MOVETO = koino_shmeic	
061		SPEED = 20	
062		MOVETO = t4_vasi_pano	
063		MOVETO = t4_vasi	




Το τέταρτο αντικείμενο μεταφέρεται στην κατάλληλη θέση.

064		SPEED = 40	
065		GRIP = <-- -->	
066		MOVETO = t4_vasi_pano	
067		SPEED = 80	
068		MOVETO = koino_shmeic	
069		SPEED = 20	
070		MOVETO = pano_basi	
071		MOVETO = basi	
072		SPEED = 30	

Ο βραχίονας συλλέγει τη βάση με τα αντικείμενα.

073		GRIP = --> <--	
074		MOVETO = pano_basi	
075		SPEED = 30	
076		MOVETO = pano_topothe	
077		SPEED = 15	
078		MOVETO = topothetisi	
079		SPEED = 40	
080		GRIP = <-- -->	

Η βάση τοποθετείται στο σημείο που έχει οριστεί ως τελικό σημείο μεταφοράς.

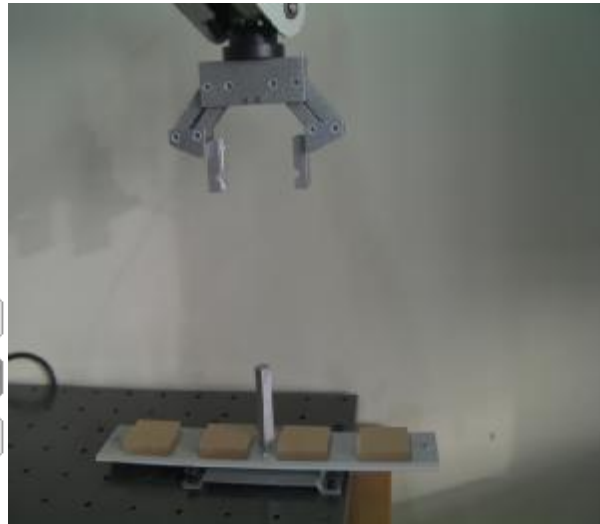
081	
082	
083	
084	

SPEED =

MOVETO =

SPEED =

HOME



Η βάση τοποθετήθηκε στο κατάλληλο σημείο και ο βραχίονας γυρίζει στην αρχική του θέση.

ΠΕΜΠΤΟ ΚΕΦΑΛΑΙΟ

ΠΕΜΠΤΟ ΚΕΦΑΛΑΙΟ

5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παραθέσουμε το κυρίως θέμα μας που είναι η δημιουργία της λειτουργίας εικονικού ρομποτικού πάρκινγκ με την βοήθεια ενός ρομποτικού βραχίονα.

Η ιδέα για την δημιουργία ενός ρομποτικού πάρκινγκ γεννήθηκε μέσα από την ανάγκη του ανθρώπου να απλοποιεί την ζωή του και να κερδίζει χρόνο στην καθημερινότητά του. Στο τέλος αυτού του κεφαλαίου αναλύονται τα θετικά και αρνητικά χαρακτηριστικά του ρομποτικού πάρκινγκ καθώς και τα συμπεράσματά μας.

Θα αναλύσουμε το θέμα σε κάθε φάση-βήμα με την βοήθεια φωτογραφιών αλλά και δείχνοντας το πρόγραμμα του ρομποτικού πάρκινγκ που κατασκευάσαμε με όλες τις μορφές:

- Πρόγραμμα γραμμών εντολών
- Πρόγραμμα εικονιδίων
- Και τέλος με γράφημα ροής για την γρήγορη και εύκολη κατανόηση του.

5.2 Περιγραφή του προγράμματος του ρομποτικού παρκινγκ.

Όταν εμφανιστεί στην είσοδο ένα όχημα το ρομπότ θα το μεταφέρει στην πρώτη ελεύθερη θέση που θα βρει στο πάρκινγκ. Το ρομπότ όταν του ζητηθεί θα μεταφέρει ένα όχημα από το πάρκινγκ σε ένα συγκεκριμένο σημείο, που θα είναι και το σημείο εξόδου. Όταν εμφανιστεί ξανά όχημα θα μεταφερθεί στη πρώτη κενή θέση.

Το πρόγραμμα περιέχει τα παρακάτω βήματα.

- Εμφανίζεται ένα όχημα στην είσοδο του πάρκινγκ
- Το ρομπότ πάει και το μεταφέρει σε μία θέση πάρκινγκ (την πρώτη ελεύθερη θέση η οποία πλέον θα φαίνεται κατειλημμένη).
- Ένα άλλο όχημα εμφανίζεται στην είσοδο του πάρκινγκ
- Το ρομπότ πάει και το μεταφέρει στην αμέσως επόμενη κενή θέση στο χώρο στάθμευσης η οποία πλέον θα φαίνεται κατειλημμένη.

- Στη συνέχεια έχουμε ένα άλλο όχημα στην είσοδο.
- Το ρομπότ πάει και το μεταφέρει στην αμέσως επόμενη κενή θέση στο χώρο στάθμευσης η οποία πλέον θα φαίνεται κατειλημμένη.
- Ένα ακόμα όχημα εμφανίζεται στην είσοδο
- Το ρομπότ πάει και το μεταφέρει στην αμέσως επόμενη κενή θέση στο χώρο στάθμευσης η οποία πλέον θα φαίνεται κατειλημμένη.
- Δίνετε εντολή ένα όχημα να μεταφερθεί στην έξοδο.
- Το ρομπότ μεταφέρει το όχημα στην έξοδο.
- Η θέση από την οποία πάρθηκε εμφανίζεται στο σύστημα κενή
- Ένα όχημα θα έρθει στην είσοδο
- Το ρομπότ θα το μεταφέρει στην αμέσως επόμενη κενή θέση στο χώρο στάθμευσης η οποία πλέον θα φαίνεται κατειλημμένη.

5.3 Πρόγραμμα γραμμών εντολών ρομποτικού πάρκινγκ.

Σε προηγούμενα κεφάλαια έχουμε αναλύσει την σύνταξη και το νόημα των εντολών που χρησιμοποιούνται στο παρακάτω πρόγραμμα.



ΠΡΟΓΡΑΜΜΑ :

001: HOME

(Ο ρομποτικός βραχίονας βρίσκεται στην αρχική του θέση)

002: INPUT(1) OFF

(Στέλνεται σήμα από την είσοδο TTL 1 ότι υπάρχει όχημα στη είσοδο)

003: GRIPOPEN

(Ανοίγει η αρπάγη)

004: MOVETO pano_touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

005: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

006: MOVETO touvlaki_1

(Μετακίνηση στο επιθυμητό σημείο)

007: GRIPCLOSE

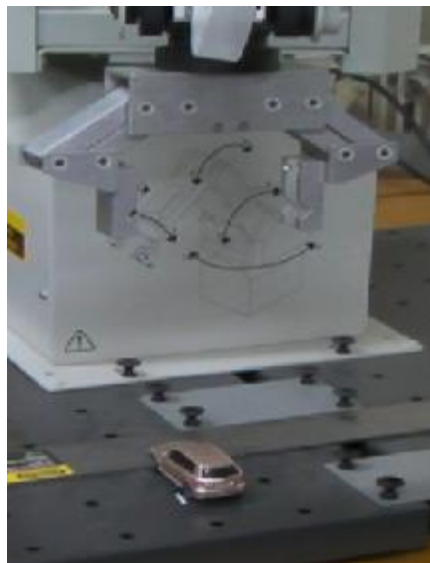
(Η αρπάγη κλείνει)

008: MOVETO pano_touvlaki_1

(Μετακίνηση στο επιθυμητό σημείο)

009: SPEED 50

(Η ταχύτητα ρυθμίζεται στο 50)



Εικόνα 5-1: Το όχημα παραλαμβάνεται από το σημείο εισόδου.

010: MOVETO koino_shmeio

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

011: INPUT(2) ON

012: INPUT(3) ON

013: INPUT(4) ON

014: INPUT(5) ON

(Οι γραμμές 011-014 δείχνουν ότι όλες οι θέσεις είναι ελεύθερες)

015: MOVETO koino_shmeio

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

016: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)



Εικόνα 5-2 Επεξεργάζεται από τα σήματα που λαμβάνει ότι όλες οι θέσεις πάρκινγκ είναι κενές.

017: MOVETO t1_vasi_pano
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

018: MOVETO t1_vasi
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

019: SPEED 40
(Η ταχύτητα ρυθμίζεται στο 40)

020: GRIPOPEN
(Η αρπάγη ανοίγει)

021: MOVETO t1_vasi_pano
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

022: SPEED 90
(Η ταχύτητα ρυθμίζεται στο 90)

023: HOME
(Ο ρομποτικός βραχίονας πηγαίνει στην αρχική του θέση)

024: INPUT(1) OFF
(Στέλνεται σήμα από την είσοδο TTL 1 ότι υπάρχει όχημα στη είσοδο)

025: GRIPOPEN
(Η αρπάγη ανοίγει)

026: MOVETO pano_touvlaki_1
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

027: SPEED 20
(Η ταχύτητα ρυθμίζεται στο 90)

028: MOVETO touvlaki_1
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

029: GRIPCLOSE
(Η αρπάγη ανοίγει)



Εικόνα 5-3 Τοποθέτηση του οχήματος στην πρώτη κενή θέση πάρκινγκ.

030: MOVETO pano_touvlaki_1
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

031: SPEED 50
(Η ταχύτητα ρυθμίζεται στο 50)

032: MOVETO koino_shmeio
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

033: INPUT(2) OFF

034: INPUT(3) ON

035: INPUT(4) ON

036: INPUT(5) ON

(Οι θέσεις 033 – 036 δείχνουν ποιες θέσεις είναι ελεύθερες)

037: MOVETO t2_vasi_pano
(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

038: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

039: MOVETO t2_vasi

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

040: SPEED 40

(Η ταχύτητα ρυθμίζεται στο 40)

041: GRIPOPEN

(Η αρπάγη ανοίγει)



Εικόνα 5-4 Ο ρομποτικός βραχίονας αναλύει τα δεδομένα από το σύστημα για να εντοπίσει την πρώτη κενή θέση που υπάρχει.

042: HOME

(Ο ρομποτικός βραχίονας πηγαίνει στην αρχική του θέση)

043: GRIPCLOSE

(Η αρπάγη ανοίγει)

044: INPUT(1) OFF

(Στέλνεται σήμα από την είσοδο TTL 1 ότι υπάρχει όχημα στη είσοδο)

045: GRIPOPEN

(Η αρπάγη ανοίγει)

046: MOVETO pano_touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

047: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

048: MOVETO touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

049: GRIPCLOSE

(Η αρπάγη κλείνει)

050: MOVETO pano_touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

051: SPEED 50

(Η ταχύτητα ρυθμίζεται στο 50)



Εικόνα 5-5 Η αρπάγη ανοίγει ώστε να παραλάβει το όχημα.

052: MOVETO koino_shmeio

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

053: INPUT(2) OFF

054: INPUT(3) OFF

055: INPUT(4) ON

056: INPUT(5) ON

(Οι θέσεις 053 – 056 δείχνουν ποιες θέσεις είναι ελεύθερες)

057: MOVETO t3_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

058: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

060: MOVETO t3_vasi

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

061: SPEED 40

(Η ταχύτητα ρυθμίζεται στο 40)

062: GRIPOPEN

(Η αρπάγη ανοίγει)

063: MOVETO t3_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

064: SPEED 90

(Η ταχύτητα ρυθμίζεται στο 90)



Εικόνα 5-6 Ο ελεγκτής στέλνει σήμα ότι οι θέσεις 4-5 είναι κενές.

065: HOME

(Ο ρομποτικός βραχίονας πηγαίνει στην αρχική του θέση)

066: INPUT(1) OFF

(Στέλνεται σήμα από την είσοδο TTL 1 ότι υπάρχει όχημα στη είσοδο)

067: GRIPOPEN

(Η αρπάγη ανοίγει)

068: MOVETO pano_touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

069: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

070: MOVETO touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

071: GRIPCLOSE

(Η αρπάγη κλείνει)

072: MOVETO pano_touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

073: SPEED 50

(Η ταχύτητα ρυθμίζεται στο 50)

074: MOVETO koino_shmeio

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

075: INPUT(2) OFF

076: INPUT(3) OFF

077: INPUT(4) OFF

078: INPUT(5) ON

(Οι θέσεις 075 – 078 δείχνουν ποιες θέσεις είναι ελεύθερες)

079: MOVETO t4_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

080: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

081: MOVETO t4_vasi

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

082: SPEED 40

(Η ταχύτητα ρυθμίζεται στο 40)

083: GRIPOPEN

(Η αρπάγη ανοίγει)

084: MOVETO t4_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

085: SPEED 90

(Η ταχύτητα ρυθμίζεται στο 90)



Εικόνα 5-7 Ακόμα ένα όχημα τοποθετείται στην κενή θέση πάρκινγκ.

086: HOME

(Ο ρομποτικός βραχίονας πηγαίνει στην αρχική του θέση)

087: OUTPUT(3) ON

(Στέλνεται σήμα από την έξοδο TTL 3 ότι το αντίστοιχο όχημα πρέπει να μετακινηθεί)

088: MOVETO koino_shmeio

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

089: GRIPOPEN

(Η αρπάγη ανοίγει)

090: MOVETO t3_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

091: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

092: MOVETO t3_vasi

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

093: GRIPCLOSE

(Η αρπάγη κλείνει)

094: MOVETO t3_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

095: SPEED 50

(Η ταχύτητα ρυθμίζεται στο 50)



Εικόνα 5-8 Ο βραχίονας επεξεργάζεται την εντολή που του δόθηκε ώστε να παραλάβει το όχημα από την κατάλληλη θέση.



Εικόνα 5-9 Ο βραχίονας παραλαμβάνει το επιθυμητό

096: MOVETO pano_topothesis

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

097: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

098: MOVETO topothesis

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

099: SPEED 40

(Η ταχύτητα ρυθμίζεται στο 40)

100: GRIOPEN

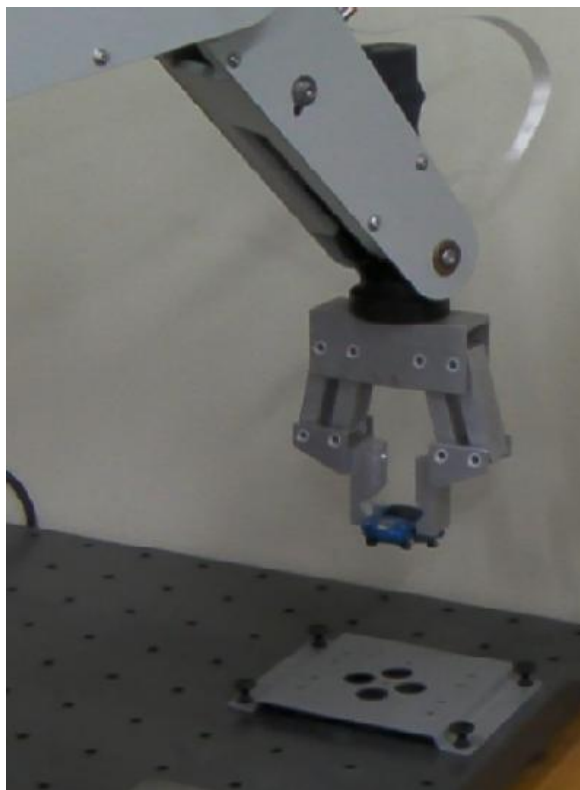
(Η αρπάγη ανοίγει)

101: MOVETO pano_topothesis

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

102: SPEED 90

(Η ταχύτητα ρυθμίζεται στο 90)



Εικόνα 5-10 Τοποθέτηση του οχήματος στο σημείο εξόδου οχημάτων.

103: HOME

(Ο ρομποτικός βραχίονας πηγαίνει στην αρχική του θέση)

104: INPUT(1) OFF

(Στέλνεται σήμα από την είσοδο TTL 1 ότι υπάρχει όχημα στη είσοδο)

105: GRIPOPEN

(Η αρπάγη ανοίγει)

106: MOVETO pano_touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

107: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

108: MOVETO touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

109: GRIPCLOSE

(Η αρπάγη κλείνει)

110: MOVETO pano_touvlaki_1

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

111: SPEED 50

(Η ταχύτητα ρυθμίζεται στο 50)

112: MOVETO koino_shmeio

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

113: INPUT(2) OFF

114: INPUT(3) OFF

115: INPUT(4) ON

116: INPUT(5) OFF

(Οι θέσεις 113 – 116 δείχνουν ποιες θέσεις είναι ελεύθερες)

117: MOVETO t3_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

118: SPEED 20

(Η ταχύτητα ρυθμίζεται στο 20)

119: MOVETO t3_vasi

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

120: SPEED 40

(Η ταχύτητα ρυθμίζεται στο 40)

121: GRIPOPEN

(Η αρπάγη ανοίγει)

122: MOVETO t3_vasi_pano

(Μετακίνηση βραχίονα στο επιθυμητό σημείο)

123: SPEED 90

(Η ταχύτητα ρυθμίζεται στο 90)

124: HOME

(Ο ρομποτικός βραχίονας πηγαίνει στην αρχική του θέση)

125: END

(Τέλος όλων των εργασιών)












Εικόνα 5-11 Ο ελεγκτής στέλνει σήμα στον βραχίονα ότι η μόνη κενή θέση παρκινγκ είναι η 4.












Εικόνα 5-12 Ο βραχίονας γυρίζει στην αρχική του θέση.

5-4 Πρόγραμμα εικονιδίων ρομποτικού παρκινγκ










Ο βραχίονας βρίσκεται στην αρχική του θέση. Μόλις του δοθεί το σήμα από τον εκλεκτή ότι στο σημείο εισόδου υπάρχει όχημα κινείται προς τα εκεί για να το παραλάβει.

001		HOME
002		INPUT 1 = 0
003		GRIP = <-- -->
004		MOVETO = pano_touvlak
005		SPEED = 20
006		MOVETO = touvlaki_1
007		GRIP = --> <--
008		MOVETO = pano_touvlak
009		SPEED = 50










Αφού ελέγξει ποιες θέσεις είναι κενές τοποθετεί το όχημα στην πρώτη κενή θέση.

010		MOVETO = koino_shmeic
011		INPUT 2 = 1
012		INPUT 3 = 1
013		INPUT 4 = 1
014		INPUT 5 = 1
015		MOVETO = koino_shmeic
016		SPEED = 20
017		MOVETO = t1_vasi_pano
018		MOVETO = t1_vasi










Αυξάνει την ταχύτητά του και γυρίζει στην αρχική του θέση. Παίρνει σήμα ότι άλλο ένα όχημα βρίσκεται στο σημείο εισόδου.

019		SPEED = <input type="text" value="40"/>
020		GRIP = <-- --> ▼
021		MOVETO = <input type="text" value="t1_vasi_pano"/> ▼
022		SPEED = <input type="text" value="90"/>
023		HOME
024		INPUT 1 ▼ = <input type="text" value="O"/> ▼
025		GRIP = <-- --> ▼
026		MOVETO = <input type="text" value="pano_touvlak"/> ▼
027		SPEED = <input type="text" value="20"/>










Κλείνει την αρπάγη και πηγαίνει στο σημείο ελέγχου για να εντοπίσει τις κενές θέσεις.

028		MOVETO = <input type="text" value="touvlaki_1"/> ▼
029		GRIP = --> <-- ▼
030		MOVETO = <input type="text" value="pano_touvlak"/> ▼
031		SPEED = <input type="text" value="50"/>
032		MOVETO = <input type="text" value="koino_shmeic"/> ▼
033		INPUT 2 ▼ = <input type="text" value="O"/> ▼
034		INPUT 3 ▼ = <input type="text" value="I"/> ▼
035		INPUT 4 ▼ = <input type="text" value="I"/> ▼
036		INPUT 5 ▼ = <input type="text" value="I"/> ▼










Πρώτη κενή θέση η 2 και το όχημα τοποθετείται εκεί.

037		MOVETO = t2_vasi_pano
038		SPEED = 20
039		MOVETO = t2_vasi
040		SPEED = 40
041		GRIP = <-- -->
042		HOME
043		GRIP = --> <--
044		INPUT 1 = O
045		GRIP = <-- -->










Παίρνει την εντολή ότι υπάρχει όχημα και πηγαίνει στο κατάλληλο σημείο για να το παραλάβει.

046		MOVETO = pano_touvlak
047		SPEED = 20
048		MOVETO = touvlaki_1
049		GRIP = --> <--
050		MOVETO = pano_touvlak
051		SPEED = 50
052		MOVETO = koino_shmeic
053		INPUT 2 = O
054		INPUT 3 = O










Μετά τον έλεγχο για κενές θέσεις το όχημα τοποθετείται στην πρώτη κενή.

055		INPUT 4 = I
056		INPUT 5 = I
057		MOVETO = t3_vasi_pano
058		SPEED = 20
059		MOVETO = t3_vasi
060		SPEED = 40
061		GRIP = <-- -->
062		MOVETO = t3_vasi_pano
063		SPEED = 90










Άλλη μία παραλάβει οχήματος από την θέση εισόδου.

064		HOME
065		INPUT 1 = O
066		GRIP = <-- -->
067		MOVETO = pano_touvlak
068		SPEED = 20
069		MOVETO = touvlaki_1
070		GRIP = --> <--
071		MOVETO = pano_touvlak
072		SPEED = 50










Έλεγχος κενής θέσης και τοποθέτηση στην πρώτη διαθέσιμη.

073		MOVETO = <input type="text" value="koino_shmeic"/>
074		INPUT 2 = <input type="text" value="O"/>
075		INPUT 3 = <input type="text" value="O"/>
076		INPUT 4 = <input type="text" value="O"/>
077		INPUT 5 = <input type="text" value="I"/>
078		MOVETO = <input type="text" value="t4_vasi_pano"/>
079		SPEED = <input type="text" value="20"/>
080		MOVETO = <input type="text" value="t4_vasi"/>
081		SPEED = <input type="text" value="40"/>










Αφού έχει επιστρέψει στην αρχική του θέση, λαμβάνει σήμα ότι ένα όχημα πρέπει να μεταφερθεί στο σημείο εξόδου.

082		GRIP = <input type="text" value="<-- -->"/>
083		MOVETO = <input type="text" value="t4_vasi_pano"/>
084		SPEED = <input type="text" value="90"/>
085		HOME
086		OUTPUT 3 = <input type="text" value="I"/>
087		MOVETO = <input type="text" value="koino_shmeic"/>
088		GRIP = <input type="text" value="<-- -->"/>
089		MOVETO = <input type="text" value="t3_vasi_pano"/>
090		SPEED = <input type="text" value="20"/>










Το επιθυμητό όχημα παραλαμβάνεται από την θέση παρκινγκ που βρίσκεται και τοποθετείτε στο σημείο εξόδου.

091		MOVETO = t3_vasi
092		GRIP = --> <--
093		MOVETO = t3_vasi_pano
094		SPEED = 50
095		MOVETO = pano_topothe
096		SPEED = 20
097		MOVETO = topothetisi
098		SPEED = 40
099		GRIP = <-- -->







Ένα καινούριο όχημα έχει έρθει στο σημείο εισόδου του παρκινγκ και ο βραχίονας το παραλαμβάνει

100		MOVETO = pano_topothe
101		SPEED = 90
102		HOME
103		INPUT 1 = O
104		GRIP = <-- -->
105		MOVETO = pano_touvlak
106		SPEED = 20
107		MOVETO = touvlaki_1
108		GRIP = --> <--

Μετά τον έλεγχο για κενή θέση το όχημα τοποθετείται στην πρώτη ελεύθερη θέση στάθμευσης.

109		MOVETO = pano_touvlak
110		SPEED = 50
111		MOVETO = koino_shmeic
112		INPUT 2 = O
113		INPUT 3 = O
114		INPUT 4 = I
115		INPUT 5 = O
116		MOVETO = t3_vasi_pano
117		SPEED = 20

Αφού τελειώσει η τοποθέτηση και του τελευταίου οχήματος ο βραχίονας γυρίζει στο αρχικό του σημείο.

118		MOVETO = t3_vasi
119		SPEED = 40
120		GRIP = <-- -->
121		MOVETO = t3_vasi_pano
122		SPEED = 90
123		HOME

5.5 Πρόγραμμα γραφημάτων ροής ρομποτικού πάρκινγκ.

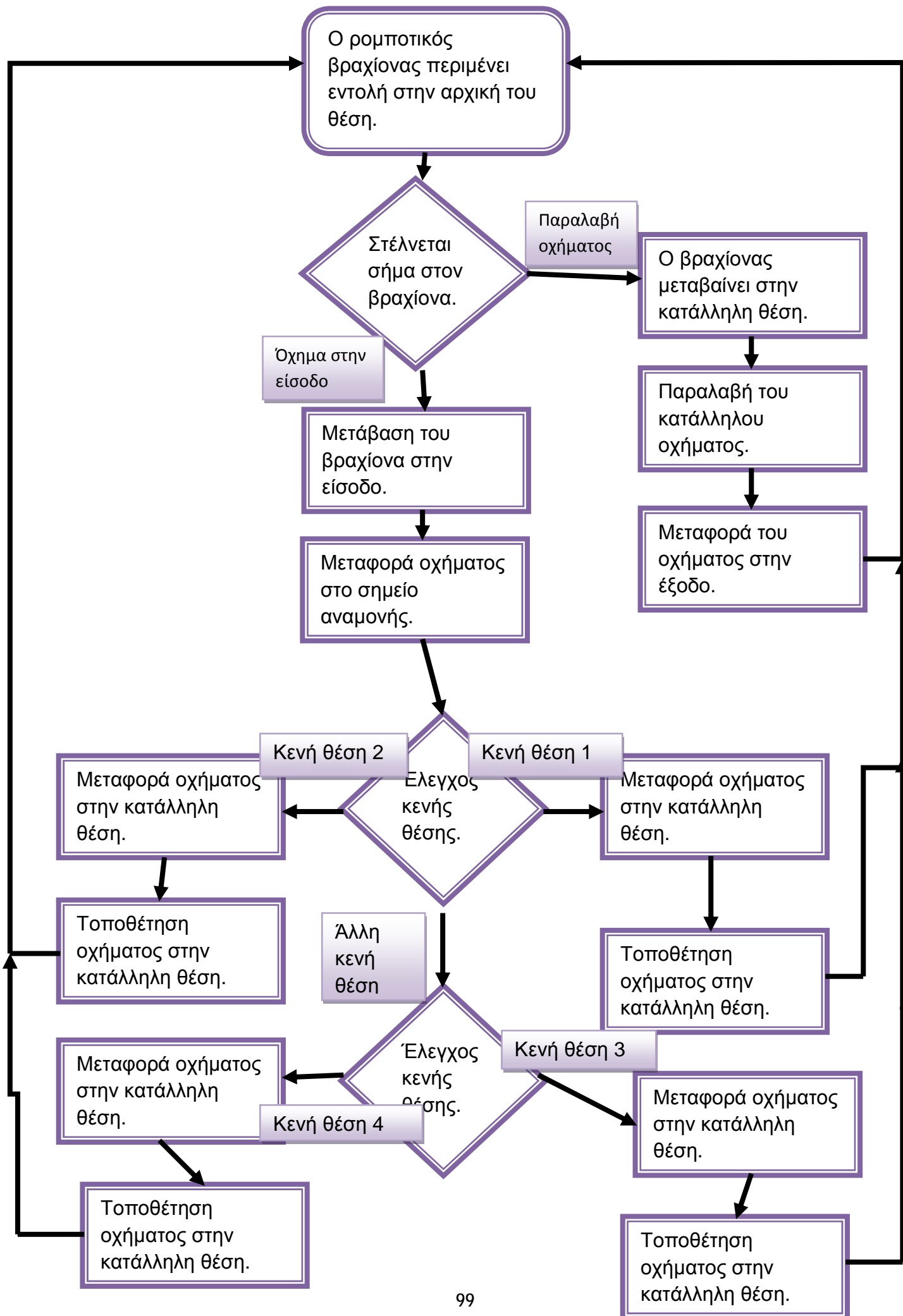
- ✚ Ο ρομποτικός βραχίονας βρίσκεται στην αρχική του θέση.
- ✚ Ο ελεγκτής στέλνει σήμα:
 - i. Για την παραλαβή οχήματος από την είσοδο.
 - ii. Για την παράδοση οχήματος στην έξοδο.

Για την παραλαβή οχήματος από την είσοδο:

- ✚ Εάν υπάρχει όχημα ο βραχίονας το παραλαμβάνει.
- ✚ Το μεταφέρει σε ένα σημείο αναμονής.
- ✚ Ελέγχει για κενή θέση στάθμευσης και το μεταφέρει στην πρώτη κενή.
- ✚ Γυρίζει πάλι στην αρχική του θέση.
- ✚ Κάνει έλεγχο για επόμενο όχημα στο σημείο εισόδου.
- ✚ Εάν δεν υπάρχει παραμένει στην αρχική του θέση.

Για την παράδοση οχήματος στην έξοδο:

- ✚ Ο βραχίονας μεταβαίνει στο επιθυμητό σημείο.
- ✚ Παραλαμβάνει το κατάλληλο όχημα.
- ✚ Το μεταφέρει στο σημείο εξόδου.
- ✚ Γυρίζει πάλι στην αρχική του θέση.



ΣΥΜΠΕΡΑΣΜΑΤΑ

ΣΥΜΠΕΡΑΣΜΑΤΑ

Με την εκπόνηση αυτής της πτυχιακής εργασίας καταφέραμε να αποκομίσουμε σημαντικές γνώσεις καθώς:

- ∅ Μάθαμε για την ιστορική εξέλιξη των αυτόματων συστημάτων και ειδικότερα των ρομποτικών βραχιόνων.
- ∅ Κατανοήσαμε τον τρόπο λειτουργίας ενός ρομποτικού βραχίονα.
- ∅ Διδαχθήκαμε τις βασικές αρχές του προγραμματισμού και τον τρόπο σύνταξης της κάθε εντολής.
- ∅ Προγραμματίσαμε ένα ρομποτικό βραχίονα να εκτελεί κινήσεις για την δημιουργία ρομποτικού πάρκινγκ.

Γενικά καταλάβαμε την σημασία της εξέλιξης της τεχνολογίας στη ζωή μας και πόσο βοηθάει αυτή την καθημερινότητά μας.

Κάνοντας αυτή την εργασία είδαμε ότι υπάρχουν πλεονεκτήματα, μειονεκτήματα αλλά και προοπτικές.

Πλεονεκτήματα:

- ∅ Κατασκευάζοντας ένα τέτοιου είδους πάρκινγκ αυξάνουμε τον χώρο στάθμευσης των οχημάτων. Αυτό προκύπτει από το γεγονός ότι σε ένα συμβατικό στεγασμένο πάρκινγκ το αυτοκίνητο οδηγείται στη θέση του από τον εκάστοτε χειριστή. Έτσι ως αποτέλεσμα είναι να χρειάζεται περισσότερος χώρος ανάμεσα στα οχήματα για να γίνεται εύκολα η είσοδος και η έξοδος από τις διάφορες θέσεις. Όταν όμως αυτό γίνει με την βοήθεια του εκάστοτε ρομποτικού βραχίονα ο χώρος ανάμεσα στα οχήματα μπορεί να είναι αρκετά λιγότερος.
- ∅ Ένα άλλο θετικό στοιχείο είναι και η κατακόρυφη αύξηση του ποσοστού ασφάλειας του οχήματος από κλοπές, φθορές από τρίτους αλλά και από τις καιρικές συνθήκες. Είναι λογικό ένα κλειστό πάρκινγκ να προστατεύει το όχημα από τον καιρό είτε αυτό είναι ένα απλό στεγασμένο πάρκινγκ είτε είναι κατασκευασμένο να λειτουργεί με την βοήθεια της ρομποτικής. Σίγουρα όμως η ασφάλεια είναι μεγαλύτερη σε έναν ρομποτικό χώρο στάθμευσης καθώς στο χώρο φύλαξης των οχημάτων πρόσβαση έχει μόνο το εργαζόμενο προσωπικό και μόνο όταν αυτό είναι απαραίτητο και αναγκαίο αφού την παραλαβή και την παράδοση των οχημάτων την κάνει ο ρομποτικός βραχίονας.

- Ø Ακόμα υπάρχει αύξηση των θέσεων εργασίας παρά την χρήση μηχανημάτων, που συνήθως μειώνουν το ανθρώπινο δυναμικό. Η αύξηση αυτή οφείλεται στην ιδιαιτερότητά του και αυτό γιατί εκτός των υπαλλήλων που θα είναι αναγκαίοι για την παρακολούθηση της σωστής διαδικασίας και λειτουργίας του πάρκινγκ, θα χρειάζονται και μόνιμοι μηχανικοί οι οποίοι θα ελέγχουν τη σωστή λειτουργία των μηχανισμών αλλά και πραγματοποιούν και επισκευές όταν και όπου χρειαστεί, καθώς και συνεχείς ελέγχους ανάλογα με τις ανάγκες του συστήματος.
- Ø Τέλος ένα ακόμα στοιχείο το οποίο δεν μπορούμε να παραλείψουμε είναι ο χρόνος εξυπηρέτησης, δηλαδή η παράδοση-παραλαβή του οχήματος είναι σύντομη και αυτό γιατί ο χρόνος μειώνεται σημαντικά καθώς δεν χρειάζεται η άμεση μετάβασή μας στο χώρο στάθμευσης αφού το όχημα μεταφέρεται από και προς εμάς γρήγορα λόγω του αυτόματου μηχανισμού μετακίνησης ουσιαστικά με το πάτημα ενός κουμπιού.

Μειονεκτήματα:

- Ø Αυξημένο κόστος κατασκευής σε σχέση με ένα συμβατικό στεγασμένο πάρκινγκ, λόγω της χρήσης των αυτόματων-μηχανικών αλλά και ηλεκτρικών μηχανημάτων.
- Ø Αυξημένο κόστος συντήρησης.
- Ø Αυξημένο κόστος χρήσης ενός τέτοιου χώρου.

Η σύγκριση όμως των θετικών και αρνητικών στοιχείων που αναφέρονται παραπάνω μας οδηγούν στο συμπέρασμα ότι μια τέτοια κατασκευή έχει περισσότερα οφέλη παρά ζημίες.

Τέλος παραθέτουμε και κάποιες προοπτικές εξέλιξης που θα μπορούσαν να συμπεριληφθούν σε ένα πραγματικό ρομποτικό πάρκινγκ για την καλύτερη λειτουργία του.

Προοπτικές:

- Ø Στο χώρο στάθμευσης θα μπορούσαν να υπάρχουν αισθητήρες βάρους για την καταγραφή σε χρονοδιακόπτη της διάρκειας της στάθμευσης έτσι ώστε να υπάρχει και η κατάλληλη χρέωση.
- Ø Σε ειδικά διαμορφωμένους χώρους θα μπορούσε να υπάρχει χώρος για την εξωτερική καθαριότητα του οχήματος εάν αυτό είναι επιθυμητό.
- Ø Επίσης θα μπορούσε να υπάρχει και κάποιος άλλος ρομποτικός βραχίονας του οποίου η αρπάγη να αντικατασταθεί από κάποιο ειδικό εξάρτημα το οποίο να διαθέτει αισθητήρες θερμότητας και τραχύτητας επιφανείας συλλέγοντας αποτελέσματα και διακρίνοντας την φθορά του ελαστικού ειδοποιώντας τον οδηγό κατά την παραλαβή του οχήματος στο σημείο εξόδου.

- Ø Τέλος ο βραχίονας θα ,μπορούσε να μεταφέρει σε ανάλογες κενές θέσεις διάφορα άλλα αντικείμενα είτε μιας γραμμής παραγωγής ενός εργοστασίου είτε τα αποθέματα μιας αποθήκης π.χ. ανταλλακτικών (άξονες, έδρανα, ζάντες κ.α.) οχημάτων βαρέου τύπου που θα έπρεπε ανάλογα με την πληρότητας ενός υλικού στον εκάστοτε αποθηκευτικό χώρο να τον ανεφοδιάζει έτσι ώστε να κρατάει ένα σταθερό αριθμό ανταλλακτικών στο σημείο αποθήκευσης του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΒΙΒΛΙΑ:

- 1) ΝΕΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΠΑΡΑΓΩΓΗ (Δημήτριος Ε. Κοδολάκης)
(Εκδόσεις Α. Σταμούλης, Αθήνα 1988)
- 2) ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ (Φώτης Ν. Κουμπούλης – Βασίλης Γ. Μέρτζιος) (Εκδόσεις Παπασωτηρίου)
- 3) ROBOT ANALYSIS (LUNG-WEN TSAI) (Εκδόσεις Wiley-Interscience)
- 4) Servo Robot System Model 5250 (User's manual) (Lab-Volt)
- 5) Servo Robot System Model 5250 (Student's manual) (Lab-Volt)

ΠΤΥΧΙΑΚΕΣ:

1. ΚΑΤΑΣΚΕΥΗ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΑΥΤΟΝΟΜΟΥ ΤΗΛΕΟΔΗΓΟΥΜΕΝΟΥ ΡΟΜΠΟΤ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΚΕΤΟΥ LEGO ROBOTICS (Αλατατσάνος Εμμανουήλ – Βασιλικόπουλος Βασίλειος, Πάτρα 2005)
2. ΡΟΜΠΟΤΙΚΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΗΣ (Παπαδούρης Παναγιώτης – Παπάλης Σπυρίδων)
3. Η ΡΟΜΠΟΤΙΚΗ ΚΑΙ ΟΙ ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΣΤΙΣ ΣΥΓΚΟΛΛΗΣΕΙΣ ΜΕΤΑΛΛΩΝ (Γκιζιώτης Αλέξανδρος – Μουτσάτσος Αντώνιος, Πάτρα 2003)

ΙΣΤΟΤΟΠΟΣ:

1. <http://edurobotics.weebly.com/chirho942sigmaiotamuomicron-epsilonlambdaiotakappa972.html>
2. <http://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CF%8C%CF%84>

3. <http://www.grobot.gr/>
4. <http://www.mythologia.8m.com/robot.html>
5. http://www.on-news.gr/2011/06/blog-post_7483.html
6. <http://vivliothmmy.ee.auth.gr/183/>
7. http://courseware.mech.ntua.gr/ml23419/robotics_pdf/intro.pdf
8. <http://www.robotics.com>