



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Ηλεκτρολογίας

-Πτυχιακή Εργασία-

**ΣΧΕΔΙΑΣΗ,ΚΑΤΑΣΚΕΥΗ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΑΥΤΟΚΙΝΟΥΜΕΝΗΣ
ΡΟΜΠΟΤΙΚΗΣ ΠΛΑΤΦΟΡΜΑΣ**

Σπουδαστής: Γιώργος Σφυρογιαννάκης

Αριθμός Μητρώου:3578

Εισηγητής: Ιωάννης Στεφανάκης

Ηράκλειο 2009

Περιεχόμενα

Σύντομη Περιγραφή της Πτυχιακής Εργασίας.....	5
Εισαγωγή.....	6
Μικροελεγκτές	9
Οι μικροελεγκτές στον πραγματικό κόσμο.....	9
Απαιτήσεις – εφαρμογές.....	11
Δυνατότητες – Πλεονεκτήματα.....	12
Ο μικροελεγκτής ATMEGA16.....	14
Αρχιτεκτονική Harvard.....	17
Εσωτερικοί Καταχωρητές.....	19
Μνήμη EEPROM.....	20
Ρολόι συστήματος.....	21
Μνήμη Προγράμματος Flash.....	22
Περιφερειακές συσκευές.....	23
Διακοπές (Interrupts).....	24
Αρχικοποίηση (Reset).....	24
Χρονιστές/μετρητές.....	25
Θύρες εισόδου/εξόδου (I/O PORTS).....	27
Καταχωρητές εισόδου-εξόδου.....	29
Αυτόνομη ρομποτική πλατφόρμα.....	30
Αναπτυξιακό σύστημα STK200.....	30

Πλατφόρμα – σασί.....	30
Τροφοδοσία.....	30
Σερβοκινητήρες.....	31
Μετατροπή των σερβοκινητήρων.....	34
Έλεγχος των σέρβο-κινητήρων.....	37
Αισθητήρες.....	38
Διαμόρφωση εύρους παλμών.....	40
Ρομπότ διαφορικής κίνησης.....	42
Τροχοί.....	43
Προγραμματισμός του ρομπότ.....	45
Γιατί μια γλώσσα υψηλού επιπέδου.....	45
Η γλώσσα C.....	47
Προγραμματισμός του μικροελεγκτή.....	47
Προγραμματιστής STK200.....	48
Μεταφορά κώδικα στην μνήμη προγράμματος του AVR.....	50
Απαιτούμενος εξοπλισμός.....	50
Το λογισμικό του ρομπότ.....	52
Δημιουργώντας ένα νέο project.....	52
Εξομοίωση του προγράμματος.....	55
Εφαρμογή κίνησης με αποφυγή εμποδίων.....	60
Παράρτημα.....	61

Σύντομη Περιγραφή της Πτυχιακής Εργασίας

Η παρούσα πτυχιακή εργασία περιγράφει ένα ρομποτικό σύστημα το οποίο έχει τη δυνατότητα να πλοηγηθεί αυτόνομα αποφεύγοντας εμπόδια που τυχόν συναντά κατά την κίνησή του. Ένας κεντρικός μικροεπεξεργαστής, είναι υπεύθυνος για τις αποφάσεις που θα λάβει μέσω των αισθητηρίων και τον έλεγχο της κίνησης του συστήματος μέσω των σερβομηχανισμών. Η σχεδίαση και υλοποίηση είχε ως γνώμονα την απλότητα του συστήματος και την διαθεσιμότητα των εξαρτημάτων ώστε να αποτελέσει έναν απλοποιημένο οδηγό για όσους επιθυμούν να κάνουν τα πρώτα τους βήματα σε αυτή τη τόσο συναρπαστική τεχνολογία. Έγινε ιδιαίτερη προσπάθεια ώστε ο οδηγός αυτός να συμπεριλάβει όλα τα απαιτούμενα βήματα για την κατασκευή μιας μικρής αυτοκινούμενης ρομποτικής πλατφόρμας ικανή για κίνηση σε περιβάλλοντα με παρεμβολή εμποδίων, περιγράφοντας τις τυχόν αδυναμίες και σφάλματα που μπορεί να εμφανιστούν κατά την υλοποίηση. Οι επιμέρους τεχνολογίες που χρησιμοποιήθηκαν για να ολοκληρώσουν αυτή τη κατασκευή αλλά και ο τρόπος ενσωμάτωσης περιγράφονται με όσο γίνεται πιο σαφή τρόπο ώστε να μπορεί να τις ακολουθήσει και ο μη ειδικός του κλάδου της ρομποτικής .

Εισαγωγή

Η ρομποτική είναι ο κλάδος της μηχανικής που ασχολείται με τη σχεδίαση και κατασκευή και λειτουργία των ρομπότ. Τα ρομπότ είναι μηχανικές διατάξεις, συνήθως ηλεκτρομηχανικές, η χρήση των οποίων αποσκοπεί στην αντικατάσταση εκείνων των εργασιών που κρίνονται από τη φύση τους επαναληπτικές και χωρίς ενδιαφέρον, επικίνδυνες ή αδύνατες να τις εκτελέσει ένας άνθρωπος. Σε άλλες περιπτώσεις, χρησιμοποιούνται για να εκτελέσουν εργασίες ταχύτερα ή φθηνότερα απ' ό,τι ο άνθρωπος. Έτσι, μπορούν να χρησιμοποιηθούν στην αυτόματη παραγωγή μεγάλων ποσοτήτων κάποιου προϊόντος και με χαμηλότερο κόστος όπως για παράδειγμα, στις αλυσίδες παραγωγής των βιομηχανιών.

Η αντικατάσταση αφορά τόσο την φυσική διεργασία όσο και τη λήψη αποφάσεων που συνδέονται με αυτές. Σύμφωνα με το Robot Institute of America, η έννοια ρομπότ περιγράφει αυτούς τους μηχανισμούς που έχουν σχεδιαστεί ώστε μέσω προγραμματισμένων κινήσεων, να μεταφέρουν υλικά, τεμάχια, εργαλεία ή ειδικευμένες συσκευές με σκοπό την επιτέλεση εργασιών μεγάλου εύρους και ποικιλίας.

Ένας τέτοιος μηχανισμός περιλαμβάνει συνήθως τα εξής υποσυστήματα

- Ένα σύστημα από μηχανισμούς που επιτρέπει στο ρομπότ να εκτελεί το έργο που έχει ανατεθεί. Οι μηχανισμοί αυτοί επιτρέπουν στο ρομπότ να κινείται οπότε και μπορεί να περιλαμβάνουν αρθρώσεις, συστήματα μετάδοσης κίνησης, και διαφόρων ειδών επενεργητές.
- Ένα υποσύστημα αισθητηριακής αντίληψης, το οποίο χρησιμοποιείται από το ρομπότ για την συλλογή πληροφορίας σχετική τόσο με το ίδιο το ρομπότ όσο και με το περιβάλλον. Το σύστημα αυτό δέχεται τα εξωτερικά ερεθίσματα και τα μετατρέπει σε ηλεκτρικά σήματα κατάλληλα για την επεξεργασία τους από ένα υπολογιστικό σύστημα.
- Ένα σύστημα ελέγχου που είναι υπεύθυνο για τη λήψη αποφάσεων κάνοντας χρήση των αισθητηριακών ερεθισμάτων. Οι αποφάσεις

πρέπει να οδηγούν σε βέλτιστα αποτελέσματα σε όσα αφορούν τον τρόπο δράσης και λειτουργίας. Ο ελεγκτής του ρομπότ συντονίζει ολόκληρο το σύστημα. Επιπροσθέτως δεν είναι απαραίτητη η χρήση γνώσεων, για την υλοποίησή του, που να σχετίζονται με γνωστικές περιοχές όπως ο αυτόματος έλεγχος, η τεχνητή νοημοσύνη, η επιστήμη των υπολογιστών κ.α.

Οι ρίζες της ρομποτικής οδηγούν αρκετά πίσω στην ανθρώπινη ιστορία. Οι πρώτες μηχανές που δείχνουν ομοιότητα τόσο στη μορφή όσο και στη λειτουργία του ανθρώπου συναντώνται ως σύλληψη στην ελληνική μυθολογία. Ο Τάλως ήταν μυθικός φύλακας της Κρήτης. Ήταν γιγάντιος, ανθρωπόμορφος και με σώμα από χαλκό. Σχετικά με την προέλευσή του, υπάρχουν διαφορετικές εκδοχές. Η πιο γνωστή, από τον Απολλόδωρο λέει πως τον κατασκεύασε ο θεός Ήφαιστος και τον χάρισε στο βασιλιά Μίνωα για να φυλάει την Κρήτη. Ο Απολλώνιος ο Ρόδιος αναφέρει ότι ήταν δώρο του Δία στην Ευρώπη, η οποία μετά τον χάρισε στο γιο της Μίνωα.

Ο Τάλως ήταν επιφορτισμένος με το καθήκον να επιτηρεί την εφαρμογή των νόμων στην Κρήτη, κουβαλώντας τους μαζί του γραμμένους σε χάλκινες πλάκες. Οι περισσότερες πηγές αναφέρουν ότι ήταν άγρυπνος φύλακας της Κρήτης που γύριζε τις ακτές του νησιού τρεις φορές τη μέρα. Κατά άλλους ήταν φτερωτός και το καθήκον αυτό το εκτελούσε πετώντας. Κρατούσε σε απόσταση τα άγνωστα πλοία που πλησίαζαν την Κρήτη πετώντας τους τεράστιες πέτρες. Αν οι άγνωστοι είχαν ήδη αποβιβαστεί, τους έκαιγε με την ανάσα του ή πυράκτωνε το χάλκινο σώμα του, τους αγκάλιαζε σφιχτά πάνω του κι έτσι τους έκαιγε.

Η εισαγωγή της έννοιας ρομπότ έγινε στον εικοστό αιώνα από τον Τσέχο θεατρικό συγγραφέα Karel Capek με το θεατρικό έργο 'Rossum Universal Robots'. Σύμφωνα με την πλοκή του έργου, μία μηχανή με το όνομα ρόμπωτ είναι επιφορτισμένο με την εκτέλεση εκείνης της εργασίας που κρίνεται

καταναγκαστική. Η λέξη ρόμπωτ προέρχεται από την τσέχικη λέξη ρομπότα για την καταναγκαστική εργασία. Το κατασκεύασμα αυτό του Rossum στρέφεται τελικά ενάντια στην ανθρωπότητα.

Τη δεκαετία του '40 ο ρώσος συγγραφέας επιστημονικής φαντασίας Isaac Asimov περιέγραψε το ρομπότ ως μία μηχανή με εμφάνιση ανθρώπου απουσία όμως συναισθημάτων. Η συμπεριφορά του υπαγορευόταν από έναν εγκέφαλο προγραμματισμένο από ανθρώπους με ιδιαίτερο τρόπο ώστε να ακολουθεί βασικές ηθικές αρχές. Για πρώτη φορά επίσης χρησιμοποιήθηκε από τον Asimov ο όρος ρομπωτική για να περιγράψει τον τομέα της επιστήμης που ασχολείται με τα ρομπότ.

Διάσημοι είναι οι τρεις νόμοι που διέπουν τη συμπεριφορά των ρομπότ όπως τους συνέλαβε ο Asimov.

- Ένα ρομπότ δεν μπορεί να βλάψει ένα ανθρώπινο πλάσμα μέσω της εργασίας του ή της αδράνειάς του
- Ένα ρομπότ υπακούει πάντα στις εντολές που δίνονται από ανθρώπους εκτός και αν αυτές αντιτίθενται στον πρώτο νόμο
- Ένα ρομπότ πρέπει να προστατεύει την ύπαρξή του, εκτός και αν αυτό αντιτίθεται με τον πρώτο ή το δεύτερο νόμο.

Μικροελεγκτές

Οι μικροελεγκτές αποτελούν μέρος ενός εντυπωσιακού αριθμού προϊόντων τα οποία βρίσκονται γύρω μας. Το αυτοκίνητό μας, τα τηλεχειριστήριά μας, η τηλεόρασή μας, οι ψηφιακές κάμερες, τα κινητά τηλέφωνα, τα πλυντήριά μας είναι μερικά από αυτά. Στην ουσία δεν θα ήταν υπερβολή να πούμε ότι η χρήση μικροελεγκτών στις μέρες μας είναι καθολική και γενικά κάθε προϊόν το οποίο αλληλεπιδρά με ένα χρήστη περιλαμβάνει ένα μικροελεγκτή, ο οποίος παίζει το ρόλο του «εγκεφάλου» των ηλεκτρονικών κυκλωμάτων. Δεν είναι τυχαίο πλέον, ότι πολλές βιομηχανίες προσανατολίζονται σε εφαρμογές όπου αποτελούνται από ηλεκτρικά κυκλώματα τα οποία ελέγχονται από μικροελεγκτή σε αντίθεση με την πρακτική του παρελθόντος όπου χρησιμοποιούσαν ογκώδεις ηλεκτρονικούς υπολογιστές (PC) ή ηλεκτρονόμους και σύνθετη αλλά μόνιμη λογική.

Η χρήση μικροελεγκτών ξεκίνησε πριν από περίπου τριάντα πέντε χρόνια. Το πρώτο μοντέρνο ενσωματωμένο σύστημα (embedded system) ήταν ο υπολογιστής του διαστημοπλοίου Apollo, ο οποίος αναπτύχθηκε από τον Charles Stark Draper στο Εργαστήριο Instrumentation Laboratory του MIT. Λόγω της ραγδαίας ανάπτυξης της ηλεκτρονικής έγινε δυνατή η βιομηχανική παραγωγή τους με μικρό κόστος.

Οι μικροελεγκτές στον πραγματικό κόσμο

Ομως τι είναι ένας μικροελεγκτής; Ένας μικροελεγκτής είναι στην ουσία ένας υπολογιστής σε μια πιο συμπαγή μορφή. Έχει δηλαδή μία μονάδα μνήμης (CPU) στην οποία εκτελούνται τα προγράμματα, μία μνήμη στην οποία αποθηκεύονται και ανανεώνονται κατά την εκτέλεση του προγράμματος οι διάφορες μεταβλητές καθώς και θύρες εισόδων – εξόδων (I/O ports) με τις οποίες μπορεί να επικοινωνήσει διαδραστικά και αμφίδρομα με τους χρήστες.

Η κύρια διαφορά τους από τους σύγχρονους υπολογιστές έγκειται στο γεγονός ότι οι μικροελεγκτές έχουν περιορισμένη μνήμη (της τάξης μερικών Kbytes, τα οποία αρκούν για τις περισσότερες εφαρμογές, ακόμα και τις πιο απαιτητικές) ενώ δεν έχουν σκληρό δίσκο. Επιπλέον λειτουργούν με χαμηλή ισχύ (της τάξεως των 50mW σε σύγκριση με τα 50-400 W των ηλεκτρονικών υπολογιστών) και έχουν μικρότερη ταχύτητα επεξεργασίας δεδομένων (η οποία όμως σε μερικές περιπτώσεις φθάνει και τα 100 MIPS*, ταχύτητα αρκετά ικανοποιητική για τις περισσότερες απαιτητικές εφαρμογές).

Ο λόγος για τον οποίο οι μικροελεγκτές έχουν αντικαταστήσει τους ηλεκτρονικούς υπολογιστές είναι το αρκετά μικρό κόστος τους, το μεγάλο πεδίο περιβαλλοντικών συνθηκών στις οποίες μπορούν να αντεπεξέλθουν, πράγμα που τους καθιστά ιδανικούς για χρήση τους στο απαιτητικό βιομηχανικό περιβάλλον, καθώς και η απόκριση πραγματικού χρόνου (real time processing). Το τελευταίο χαρακτηριστικό εξασφαλίζει στους μικροελεγκτές τη δυνατότητα να αντιλαμβάνονται και να αποκρίνονται στον επιθυμητό για μας χρόνο με μεγάλη ακρίβεια.

Μία από τις πρώτες εταιρείες, η οποία σχεδίασε μικροελεγκτές με πολλά περιφερειακά ενσωματωμένα σε ένα μόνο chip, ήταν η Intel με τη σειρά 8051 που χρησιμοποιείται ακόμα και σήμερα στο σχεδιασμό νέων προϊόντων. Άλλες εταιρείες που ακολούθησαν και τροφοδοτούν σήμερα την παγκόσμια αγορά με μικροελεγκτές είναι κυρίως οι: Atmel, Microchip, Motorola, Hitachi, Toshiba, AMD, Zilog, National Semiconductor, Philips/Signetics. Από αυτές το μεγαλύτερο μερίδιο της αγοράς έχουν οι τρεις πρώτες μαζί με την Intel. Από αυτές η Atmel και η Microchip φαίνεται να έχουν επικρατήσει στη βιομηχανία λόγω της ύπαρξης δωρεάν εργαλείων και της γρήγορης υιοθέτησης μιας ειδικού τύπου μνήμης που ονομάζεται Flash. Η Microchip έχει επικρατήσει ανάμεσα στους «χομπίστες» με τους φθηνούς μικροελεγκτές PIC και μετά το 2004, με τη διάθεσή της στην αγορά των μικροελεγκτών 16-bit dspic, έχει κερδίσει ένα μεγάλο κομμάτι της αγοράς. Αρκετά γνωστοί και αξιόπιστοι είναι και οι μικροελεγκτές της εταιρείας Motorola, όπως τα μοντέλα MC68HC11 και M68HC12. Παρ' όλο που οι μικροελεγκτές άλλων εταιρειών δεν υστερούν σε

τίποτα, δεν είναι ευρέως γνωστοί καθώς χρησιμοποιούνται σε πιο εξειδικευμένες εφαρμογές.

Απαιτήσεις - εφαρμογές

Για να μπορέσει κάποιος να αναπτύξει μία βιομηχανική εφαρμογή (π.χ. οδήγηση ενός σερβο-κινητήρα) δεν χρειάζονται πολλά. Κατ' αρχήν απαιτείται η αγορά μικροελεγκτών με τις επιθυμητές ιδιότητες σύνδεσης και επεξεργασίας (συνήθως φθηνότεροι από 20 ευρώ) καθώς και μερικών ηλεκτρονικών μικροεξαρτημάτων αμελητέου κόστους (πυκνωτές, μπαταρίες 9V, LED κ.λπ.). Απαιτείται επίσης ένα ολοκληρωμένο περιβάλλον (IDE και μεταγλωττιστής) για τον προγραμματισμό τους. Σήμερα υποστηρίζονται γλώσσες προγραμματισμού όπως η C, η Basic, η Pascal κ.λπ. Αν και υπάρχουν πολλά εμπορικά πακέτα, μπορεί κανείς να βρει στο Internet δωρεάν ολοκληρωμένα περιβάλλοντα όπως ο avr-gcc για προγραμματισμό σε C των μικροελεγκτών της Atmel και το Mplab για τους μικροελεγκτές της εταιρείας Microchip.

Επιπλέον απαραίτητη είναι η ύπαρξη ενός προγραμματιστή (με τη μορφή ηλεκτρονικής πλακέτας), έτσι ώστε να μπορούμε να κατεβάσουμε στους μικροελεγκτές μας τα προγράμματα που φτιάχνουμε στο ολοκληρωμένο περιβάλλον. Πάλι το διαδίκτυο μας δίνει τη λύση. Σε αυτό υπάρχουν τόσο σχεδιαγράμματα πλακετών προγραμματιστών για να μπορέσουμε να τους φτιάξουμε μόνοι μας, όσο και έτοιμοι τους οποίους μπορούμε να παραγγείλουμε (όπως π.χ. ο ePIC για τους PIC με ενδεικτικό κόστος 50 ευρώ).

Τα ηλεκτρονικά κυκλώματα μπορούν να υλοποιηθούν είτε πάνω σε PCB (printed circuit boards) είτε σε breadboard (τι είναι?, φωτογραφία) (κόστος κάτω των 30 ευρώ). Τέλος, το πιο σημαντικό στις υλοποιήσεις έργων με τη βοήθεια μικροελεγκτών είναι η ύπαρξη του ανθρώπινου παράγοντα ο οποίος θα πρέπει να έχει στοιχειώδεις γνώσεις ηλεκτρονικής και προγραμματισμού. Με τον απεριόριστο όμως πλούτο γνώσεων που διατίθεται στο διαδίκτυο και

την πολύ μεγάλη βιβλιογραφία που υπάρχει, δεν απαιτούνται ιδιαίτερα εξειδικευμένες δεξιότητες.

Γενικά συνοψίζοντας θα μπορούσαμε να πούμε ότι για την υλοποίηση εφαρμογών με μικροελεγκτές απαιτείται ένα αρχικό κόστος της τάξης των 50-300 ευρώ (ιδιοκατασκευή ή αγορά προγραμματιστή, χρησιμοποίηση δωρεάν λογισμικού ή αγορά ολοκληρωμένου περιβάλλοντος, αγορά μικροελεγκτών, ηλεκτρονικών εξαρτημάτων) για να δημιουργηθεί μία αρχική υποδομή και στη συνέχεια το μόνο κόστος θα είναι η σχεδόν αμελητέα αντικατάσταση των χρησιμοποιούμενων ηλεκτρονικών εξαρτημάτων. Η δε απόσβεση της επένδυσης είναι άμεση από την πρώτη κιόλας εφαρμογή.

Δυνατότητες – Πλεονεκτήματα

Οι δυνατότητες των σύγχρονων μικροελεγκτών καλύπτουν το μεγαλύτερο μέρος των εφαρμογών της βιομηχανίας. Πιο αναλυτικά, οι περισσότεροι μικροελεγκτές έχουν δυνατότητες πολλαπλών αναλογικο-ψηφιακών μετατροπών (ADC είσοδοι) για λήψη μετρήσεων από όλων των ειδών αισθητήρων που υπάρχουν στην αγορά, έλεγχο κινητήρων (συνεχούς και εναλλασσόμενου ρεύματος, βηματικούς κ.λπ.) με χρήση διαμόρφωσης εύρους παλμού (Pulse-Width Modulation PWM) και δυνατότητα προγραμματισμού κατευθυντών PID. Επιπλέον μπορούν να συνδεθούν με υπολογιστές μέσω σειριακής θύρας (RS-232), παράλληλης θύρας (IEEE 1284), καθώς και των βιομηχανικών πρωτοκόλλων RS-422 και RS-485.

Επιπλέον, οι συνηθισμένοι μικροελεγκτές μπορούν να συνδεθούν με όλους τους αισθητήρες και υπολογιστές οι οποίοι είναι συμβατοί και όχι μόνο, με χρήση διαδεδομένων πρωτοκόλλων επικοινωνίας όπως τα I2C, CAN, SPI, RF κ.λπ. Εξειδικευμένοι μικροελεγκτές επιτρέπουν τη σύνδεσή τους μέσω Bluetooth, USB, ακόμα και σύνδεση με το διαδίκτυο μέσω Ethernet. Για ένα μέσο μικροελεγκτή, η ταχύτητά τους ξεκινά από το 1 MIPS και μπορεί να φτάσει και τα 100 MIPS, ταχύτητα αρκετά ικανοποιητική για απαιτητικές εφαρμογές όπως η μετάδοση βίντεο συνεχούς ροής (videostreaming), η

επεξεργασία εικόνας (image processing) και η ψηφιακή επεξεργασία σήματος (DSP).

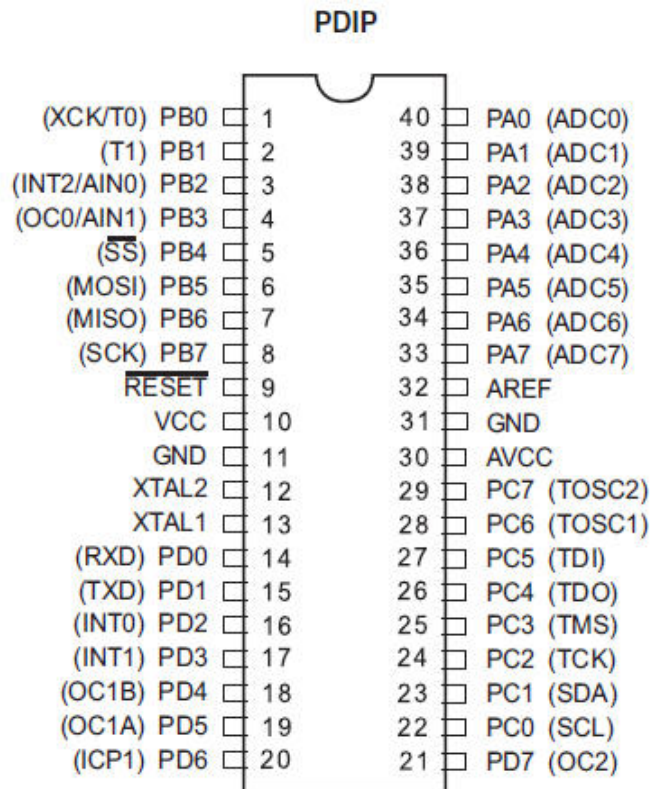
Ενας μέσος μικροελεγκτής έχει σχεδίαση 8-bit, ενώ υπάρχουν και μικροελεγκτές με αρχιτεκτονική 32-bit, γεγονός που τους δίνει την ικανότητα να εκτελούν γρήγορα πράξεις σε πραγματικό χρόνο. Με λίγα λόγια οι σύγχρονοι μικροελεγκτές μπορούν επάξια να αντικαταστήσουν σε ένα μεγάλο ποσοστό εργασιών τους υπολογιστές, τις ακριβές κάρτες οδήγησης-συλλογής δεδομένων και τα PLC. Είναι σαφώς φθηνότεροι, επομένως εάν καταστραφούν δεν δημιουργούν πρόβλημα, καταλαμβάνουν μικρότερο όγκο, δεν απαιτούν εξειδικευμένες γνώσεις (όπως π.χ. προγραμματισμό σε διαγράμματα ladder στα PLC), αλλά μόνο γενικές γνώσεις προγραμματισμού. Επιπλέον με τη χρήση μικροελεγκτών δεν έχουμε κανένα περιορισμό όσον αφορά το είδος του προγράμματος, σε αντίθεση με τα PLC ή με ακριβές κάρτες οδήγησης που έχουν συγκεκριμένες προ-πληρωμένες δυνατότητες (π.χ. μορφές ελέγχου από τις οποίες ο χρήστης επιλέγει).

Για την επίτευξη ελέγχου ή μετρήσεων σε πραγματικό χρόνο, οι ηλεκτρονικοί υπολογιστές πρέπει να τρέχουν λειτουργικά συστήματα πραγματικού χρόνου (όπως RT-Linux, QNX κ.ά.), ενώ οι μικροελεγκτές δεν απαιτούν επιπλέον λογισμικό. Τέλος, ένα άλλο πλεονέκτημα των μικροελεγκτών είναι η πολλή μικρή κατανάλωση ισχύος. Τόσο οι υπολογιστές όσο και τα PLC έχουν μικρή ενεργειακή αυτονομία σε σχέση με τους μικροελεγκτές. Έτσι για παράδειγμα ένα laptop μπορεί να εργαστεί για 2 ώρες χωρίς ρεύμα από το δίκτυο, ενώ ένας μέσος μικροελεγκτής μπορεί να εργάζεται για μέρες. Υπάρχουν δε και μικροελεγκτές χαμηλής κατανάλωσης οι οποίοι λειτουργούν με τάση κάτω των 2V.

Ο μικροελεγκτής ATMEGA16

Ένα χαρακτηριστικό παράδειγμα σύγχρονων μικροελεγκτών είναι εκείνοι της οικογένειας AVR της εταιρείας ATMEL. Οι μικροελεγκτές αυτοί προσφέρονται με ένα πλήθος εναλλακτικού αριθμού ακροδεκτών ξεκινώντας από μικρά και φτηνά ολοκληρωμένα των 8 ακροδεκτών για εφαρμογές πολύ χαμηλού κόστους με περιορισμένες απαιτήσεις σε πλήθος προγραμματιζόμενων ακροδεκτών γενικού σκοπού. Οι πιο εξελιγμένοι μικροελεγκτές της οικογένειας διαθέτουν περισσότερους από 60 προγραμματιζόμενους ακροδέκτες γενικού σκοπού. Επίσης πολλά μέλη της σειράς διατίθενται σε τρεις παραλλαγές: τους απλούς μικροελεγκτές που λειτουργούν στα 5V, τους χαμηλής κατανάλωσης στα 2.7V (κατάληξη L) και τους πολύ χαμηλής κατανάλωσης στα 1.8V (κατάληξη V).

Ο μικροελεγκτής ATMEGA16 ανήκει στην οικογένεια AVR των μικροελεγκτών της Atmel (σχήμα 1). Οι μικροελεγκτές AVR χρησιμοποιούν τροποποιημένη Αρχιτεκτονική Χάρβαρντ 8-bit RISC και αναπτύχθηκαν από την Atmel για πρώτη φορά το 1996. Η AVR ήταν μια από τις οικογένειες μικροελεγκτών που έκαναν χρήση της on-chip μνήμης flash για την αποθήκευση του προγράμματος, σε αντίθεση με τα Programmable ROM, EPROM ή EEPROM που χρησιμοποιούνται από άλλους μικροελεγκτές.



Σχήμα 1. Ακίδοσειρές του ATMEGA16

Η βασική αρχιτεκτονική των AVR επινοήθηκε από δύο μαθητές στο Νορβηγικό Ινστιτούτο Τεχνολογίας τους Alf-Bogen EGIL και Vegard Wollan. Αργότερα η πατέντα για τους AVR μικροελεγκτές αγοράστηκε από την εταιρία ATMEL και η εσωτερική αρχιτεκτονική τους αναπτύχθηκε περαιτέρω. Η θυγατρική της ATMEL στην Νορβηγία ιδρύθηκε από τους δύο φοιτητές. Το όνομα AVR δεν αποτελεί κάτι ιδιαίτερο όσον αφορά την ερμηνεία του. Απλά ονομάστηκε έτσι και ορίζει όλοι την οικογένεια των μικροελεγκτών τύπου 8-bit RISC.

Συνήθως οι ακροδέκτες γενικού σκοπού έχουν πολυπλεγμένες περισσότερες από μία λειτουργίες όπως για παράδειγμα είσοδοι με ικανότητα να προκαλούν διακοπή (interrupt) στον εσωτερικό επεξεργαστή, είσοδοι αναλογικών συγκριτών ή μετατροπέων αναλογικού σε ψηφιακό (ADC), είσοδοι κεντρικού ρολογιού (oscillator) ή ασύγχρονης οδήγησης μετρητών (counters), ακροδέκτες για σύνδεση με διάφορες διεπαφές όπως USART, SPI κ.α.

Στα πιο εξελιγμένα μέλη της οικογένειας διατίθενται ενσωματωμένα περιφερειακά ακόμα και για την οδήγηση LCD οθόνης ή τη σύνδεση με USB interface. Η διάταξη ακροδεκτών παρουσιάζεται στο Σχήμα 1. Από τα ονόματα των ακροδεκτών είναι προφανής η πολύπλεξη των λειτουργιών των εσωτερικών περιφερειακών με τη χρήση των ακροδεκτών ως γενικού σκοπού

Στο εσωτερικό ενός μικροελεγκτή όπως ο AVR υπάρχει ένας αριθμός από διαφορετικούς τύπους μνήμης όπως Flash για την εγγραφή του λογισμικού συστήματος (firmware), eeprom για την αποθήκευση διαφόρων παραμέτρων, καθώς και κάποιος αριθμός θέσεων μνήμης RAM για τις μεταβλητές του λογισμικού. Για το λόγο αυτό οι AVR δεν βγάζουν σε ακροδέκτες την εσωτερική αρτηρία διευθύνσεων ή δεδομένων παρά μόνο ακροδέκτες γενικού σκοπού. Με όλα τα παραπάνω περιφερειακά είναι φανερό ότι το πλήθος των εξωτερικών στοιχείων που απαιτούνται για τη δημιουργία ενός συστήματος με μικροελεγκτή AVR είναι ελάχιστο.

Το βασικό μειονέκτημα μιας τέτοιας αρχιτεκτονικής μικροελεγκτή είναι η δυσκολία επεκτασιμότητας. Πχ, αν οι απαιτήσεις σε μνήμη RAM είναι μεγάλες, ο μικροελεγκτής δεν είναι εύκολο να συνδεθεί με εξωτερική μνήμη μια και δεν έχει αρτηρία διευθύνσεων και δεδομένων. Για να γίνει αυτό θα πρέπει να υλοποιηθούν τέτοιες αρτηρίες με τη χρήση ακροδεκτών γενικού σκοπού οι οποίες ωστόσο θα ήταν αδύνατο να επιτύχουν γρήγορους χρόνους προσπέλασης της μνήμης. Επίσης, η συχνότητα ρολογιού στην οποία λειτουργούν τέτοιοι μικροελεγκτές δεν ξεπερνά τα 40 MHz στα πιο εξελιγμένα μοντέλα μιας σειράς όπως οι AVR.

Οι μικροελεγκτές AVR κατατάσσονται σε τέσσερις μεγάλες κατηγορίες. Απο αυτές ο μικροελεγκτής της τρέχουσας εφαρμογής ανήκει σε αυτήν με το κωδικό όνομα σειράς MEGA (megaAVR — Atmega). Τα χαρακτηριστικά που διακρίνουν αυτή την κατηγορία είναι

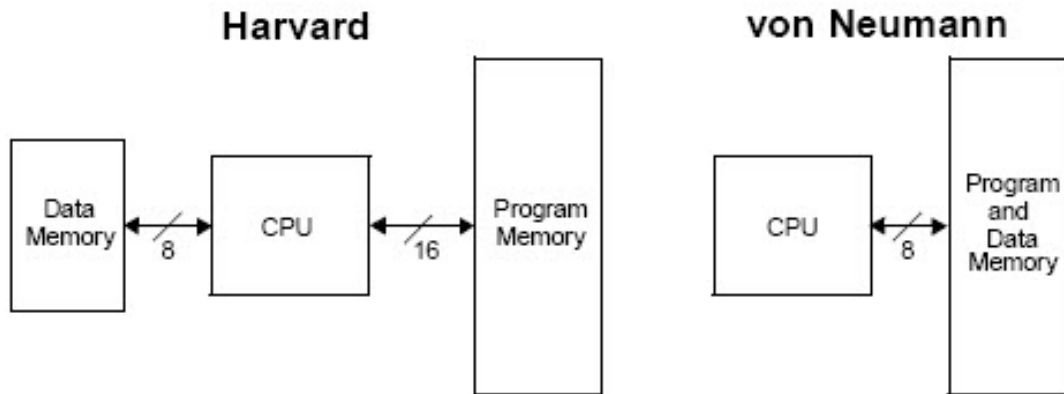
- Μνήμη προγράμματος 4-256 kB
- Αριθμός ακροδεκτών 28-100

- Εκτεταμένη σέτ εντολών (Εντολές Πολλαπλασιασμού και μεγαλύτερο χώρος για το πρόγραμμα μνήμης)
- Εκτεταμένο σετ εντολών χειρισμού περιφερειακών

Αρχιτεκτονική Harvard

Η αρχιτεκτονική Harvard έχει τη μνήμη προγράμματος και τη μνήμη δεδομένων ως χωριστές μνήμες που προσεγγίζεται από χωριστούς διαύλους (bus). Αυτό βελτιώνει το εύρος ζώνης πέρα από την παραδοσιακή αρχιτεκτονική Von Neumann στην οποία το πρόγραμμα και τα στοιχεία προσκομίζονται από την ίδια μνήμη χρησιμοποιώντας τον ίδιο δίαυλο. Για να εκτελέσει μια οδηγία, μια μηχανή Von Neumann πρέπει να κάνει γενικά περισσότερες προσβάσεις στον δίαυλο για να προσκομίσει την πληροφορία. Κατόπιν τα στοιχεία μπορεί να πρέπει να μεταφερθούν μέσω του διαύλου, να χρησιμοποιηθούν στην αριθμητική και λογική μονάδα, και ενδεχομένως να τοποθετηθούν σε μια νέα θέση μνήμης.

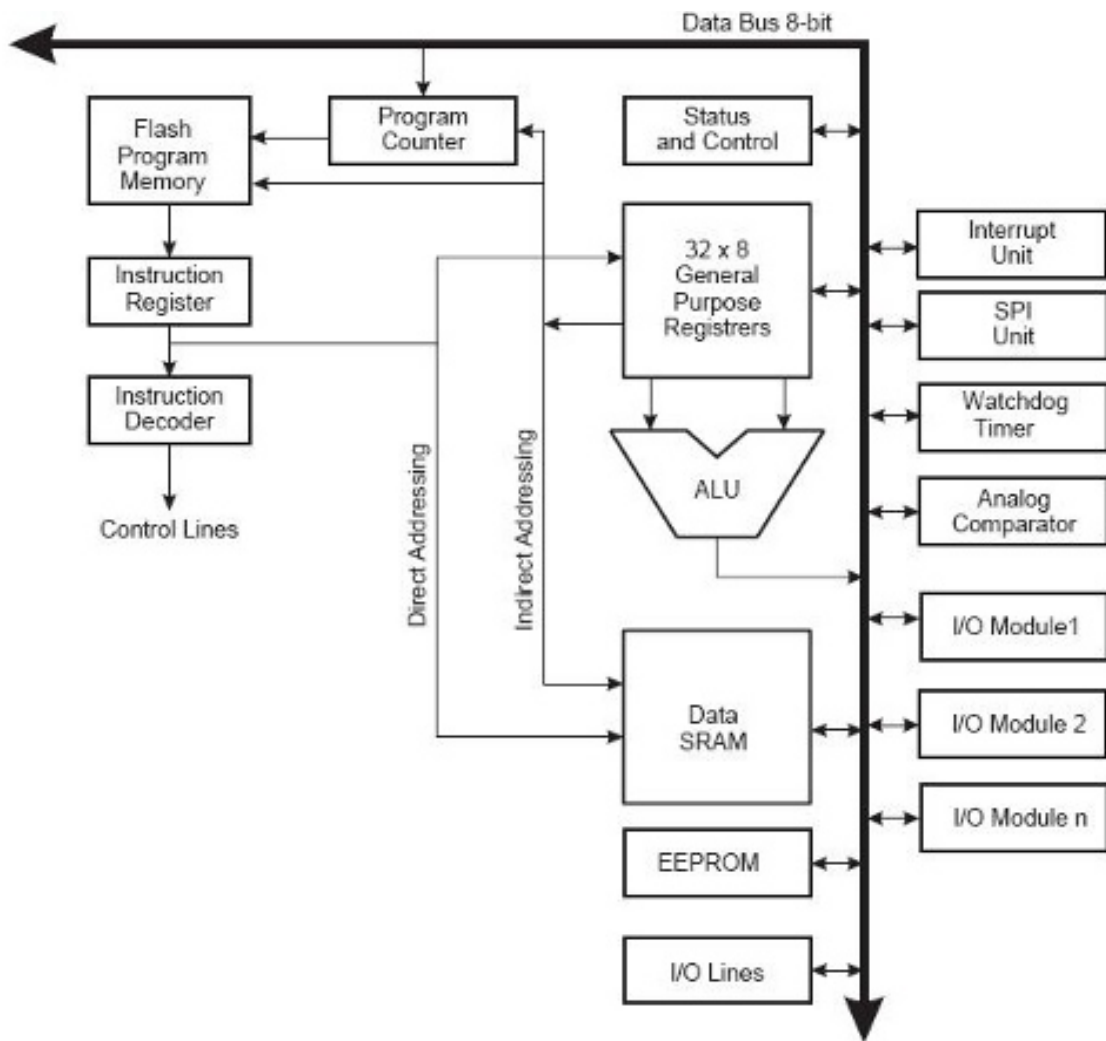
Όπως μπορεί κανείς να δει από αυτήν την περιγραφή, η αρχιτεκτονική αυτή μπορεί να δημιουργήσει κυκλοφοριακή συμφόρηση ή ακόμη και κορεσμό. Με την αρχιτεκτονική Harvard, η πληροφορία προσκομίζεται σε έναν απλό κύκλο ρολογιού. Ενώ η μνήμη προγράμματος προσπελαύνεται, η μνήμη δεδομένων είναι σε ένα ανεξάρτητο δίαυλο και μπορεί να διαβαστεί και να γραφτεί. Αυτοί οι χωρισμένοι δίαυλοι επιτρέπουν σε μια εντολή την εκτέλεση, ενώ η επόμενη εντολή προσκομίζεται. Μια σύγκριση των αρχιτεκτονικών Harvard και von Neumann παρουσιάζεται στο σχήμα 2.



Σχήμα 2. Σύγκριση των αρχιτεκτονικών

Τα γενικά γνωρίσματα των μικροελεγκτών αυτής της οικογένειας σε ότι αφορά την μνήμη δεδομένων και προγράμματος συνοψίζονται ως εξής (σχήμα 3)

- **Μνήμη δεδομένων.** Η Flash, η EEPROM και η SRAM είναι όλες ενσωματωμένες σε ένα και μοναδικό τσιπ, καταργώντας την ανάγκη για την εξωτερική μνήμη. Ορισμένες συσκευές έχουν έναν εξωτερικό διάλο ο οποίος επιτρέπει την προσθήκη συμπληρωματικών chip μνήμης .
- **Μνήμη Προγράμματος (Flash).** Οι εντολές του προγράμματος αποθηκεύονται σε μη πτητική μνήμη Flash. Αν και είναι μικροελεγκτές 8-bit, κάθε εντολή έχει εύρος ένα ή δύο 16-bit words. Το μέγεθος της μνήμης του προγράμματος αναφέρεται συνήθως στην ονοματοδοσία της ίδια της συσκευής (για παράδειγμα ο ATmega64x έχει Flash 64 KB). Δεν υπάρχει καμία πρόβλεψη για πρόγραμμα που βρίσκεται σε εξωτερική μνήμη. Οι κώδικες εκτελούνται από τον πυρήνα του AVR και μεταφέρονται στην ενσωματωμένη flash.
- **Εσωτερική Μνήμη Δεδομένων.** Η διευθυνσιοδότηση των δεδομένων αποτελείται από το αρχείο καταχωρητών, τους καταχωρητές I / O, και την SRAM.



Σχήμα 3 Μπλόκ διάγραμμα της οικογένειας μικροελεγκτών AVR

Εσωτερικοί Καταχωρητές

Οι AVR έχουν 32 single-byte καταχωρητές και έχουν ταξινομηθεί ως 8-bit RISC. Στις περισσότερες παραλλαγές της αρχιτεκτονικής AVR, η ομάδα εργασίας καταχωρητών χαρτογραφούνται σε 32 διευθύνσεις μνήμης (0000 16-001F 16), ακολουθούμενη από τους 64 I / O καταχωρητές (0020 16-005F 16). Η πραγματική SRAM αρχίζει μετά από το τμήμα καταχωρητών (διεύθυνση 0060 16). Σημειώστε ότι ο χώρος για καταχωρητές I / O μπορεί να

είναι μεγαλύτερος για πιο εκτεταμένες συσκευές. Ακόμη και αν υπάρχουν ξεχωριστά συστήματα διευθυνσιοδότησης και βελτιστοποιημένα opcodes για το αρχείο καταχωρητών (Register File) και για την πρόσβαση των καταχωρητών I / O, όλα μπορούν να διευθυνσιοδοτηθούν και να χειριστούν σαν να ήταν στην SRAM.

Μνήμη EEPROM

Σχεδόν όλοι οι AVR μικροελεγκτές έχουν εσωτερική EEPROM (Erasable Programmable Read Only Memory) για μόνιμη αποθήκευση δεδομένων. Όπως και η Flash Memory, η EEPROM μπορεί να διατηρήσει το περιεχόμενό της ακόμη και με απουσία τάσης. Στους περισσότερους AVR, αυτή η εσωτερική μνήμη EEPROM δεν διευθυνσιοδοτείται στο χώρο της μνήμης του μικροελεγκτή. Μπορεί μόνο να γίνει η προσπέλαση αυτής κατά τρόπο όμοιο με μια εξωτερική περιφερειακή συσκευή. Αυτό γίνεται με τη χρήση ειδικών καταχωρητών δείκτη (Pointer Registers) και εντολές ανάγνωσης / εγγραφής της EEPROM που κάνει πολύ την όλη διαδικασία σαφώς πιο αργή από την προσπέλαση της εσωτερικής RAM του μικροελεγκτή. Ωστόσο, μερικές συσκευές της τεχνολογίας SecureAVR χρησιμοποιούν μια ειδική χαρτογράφηση της EEPROM στα δεδομένα ή τον μνήμη του προγράμματος. Η οικογένεια των XMEGA επιτρέπει επίσης στην EEPROM να αντιστοιχίζεται με τον χώρο διευθύνσεων δεδομένων (data address space). Τέλος να αναφέρουμε ότι ο αριθμός εγγραφών της EEPROM δεν είναι απεριόριστος. Η Atmel στα datasheets των μικροελεγκτών της διευκρινίζει πως έχουν γύρω στους 100.000 κύκλους εγγραφής.

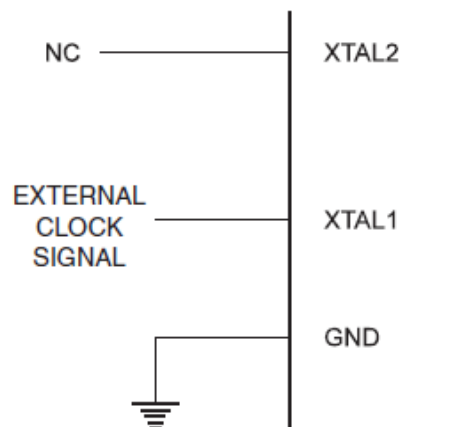
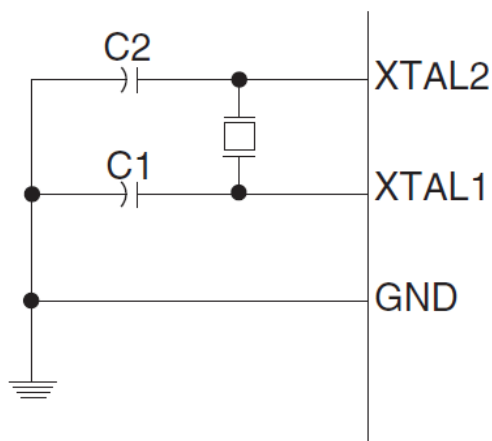
Εκτέλεση Προγράμματος

Η εκτέλεση του προγράμματος στους AVRs γίνεται με τον εξής τρόπο. Η επόμενη εντολή μηχανής φτάνει (fetched) ταυτόχρονα όσο εκτελείται η τρέχουσα. Οι περισσότερες εντολές απαιτούν έναν ή δύο κύκλους ρολογιού, καθιστώντας τους AVRs σχετικά γρήγορους μεταξύ άλλων 8-bit

μικροελεγκτών. Η οικογένεια των AVR επεξεργαστών έχουν σχεδιαστεί με στόχο την αξιόπιστη εκτέλεση κώδικα C και γιαυτό διαθέτει και ενσωματωμένους δείκτες για το έργο αυτό.

Ρολόι συστήματος

Αυτό το κύκλωμα δημιουργεί τους απαραίτητους παλμούς χρονισμού για να λειτουργήσει ο μικροεπεξεργαστής. Συνήθως συναντάται σαν ένα δίκτυωμα πυκνωτή - αντίσταση (R/C), ή με κρύσταλλο και δύο κεραμικούς πυκνωτές (Quartz), ή με ένα πακτωμένο μικροσκοπικό κύκλωμα κρυστάλλου - πυκνωτών το λεγόμενο Resonator (σχήμα 4).



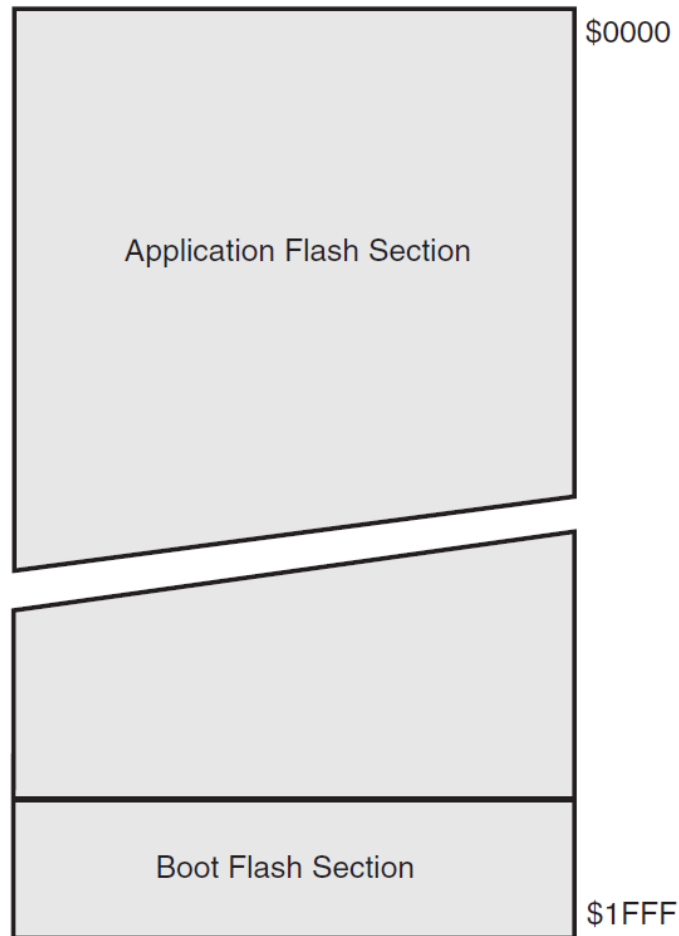
Σχήμα 4. Σύνδεση Ρολογιού

Ο ATMEGA16 εκτελεί μία εντολή (ένας κύκλος μηχανής) κάθε ένα παλμο χρονισμού, χρησιμοποιώντας την αρχιτεκτονική διασωλήνωσης (pipeline), εκτός από κάποιες εντολές όπως οι εντολές διακλάδωσης που για να εκτελεστούν χρειάζονται 2 κύκλους μηχανής. Για παράδειγμα, με κρύσταλλο 16MHz (16.000.000 παλμούς ανά δευτερόλεπτο) ο μικροελεγκτής αυτός μπορεί να εκτελέσει 16 εκατομύρια εντολές το δευτερόλεπτο.

Το ρολόι απαιτείται από τη συσκευή για να εκτελέσει τις εντολές προγράμματος και για να λειτουργήσουν οι περιφερειακές μονάδες. Το ρολόι συστήματος (CLK) προέρχεται από ένα εξωτερικό ρολόι αλλά μερικά μοντέλα παρέχουν ένα εσωτερικό ταλαντωτή περιορισμένης όμως συχνότητας. Το κύκλωμα χρονισμού μπορεί να παραχθεί με έναν από οκτώ διαφορετικούς τρόπους ταλαντωτών. Τα ψηφία διαμόρφωσης (configuration bits) της συσκευής επιλέγουν τον εκάστοτε ταλαντωτή. Τα configuration bits είναι θέσεις αμετάβλητης μνήμης και ο τρόπος λειτουργίας καθορίζεται από την τιμή που γράφεται σε αυτά κατά τη διάρκεια του προγραμματισμού της συσκευής.

Μνήμη Προγράμματος Flash

Στην μνήμη αποθηκεύεται το πρόγραμμα (κώδικας). Κάθε διεύθυνση απαρτίζεται από ψηφιολέξεις των 16 ψηφίων, και το σύνολό τους είναι πολλαπλάσιο του 1,024, ανάλογα με τον μικροελεγκτή της οικογένειας (σχήμα 5). Η μνήμη είναι αμετάβλητου τύπου (Non Volatile), κρατά τα περιεχόμενα της αναλλοίωτα ακόμα και όταν δεν υπάρχει τάση τροφοδοσίας. Το περιεχόμενο της μπορεί να γραφτεί ξανά μέσω του προγραμματιστή για περίπου 100,000 φορές.



Σχήμα 5. Χαρτογράφηση της μνήμης flash

Ο ATmega16 περιέχει 16 Kbytes επαναπρογραμματιζόμενης μνήμης προγράμματος. Η μνήμη αυτή είναι οργανωμένη σε λέξεις των 16 bit. Όπως φαίνεται και στο **σχήμα**, η μνήμη είναι χωρισμένη σε δύο ενότητες, το κυρίως κομμάτι προγράμματος και το κομμάτι εκκίνησης (bootloader)

Περιφερειακές συσκευές

Οι περιφερειακές συσκευές είναι αυτές που διαχωρίζουν έναν μικροελεγκτή από έναν επεξεργαστή. Βρίσκονται όλες μέσα στην ίδια συσκευασία και η ύπαρξη ή όχι μιας μονάδας είναι από τα κύρια κριτήρια επιλογής του κατάλληλου μικροελεγκτή. Οι περιφερειακές μονάδες διασυνδέονται με την κεντρική μονάδα επεξεργασίας και την μνήμη δεδομένων με τρόπο άμεσο ώστε η ανταπόκριση να είναι άμεση.

Διακοπές (Interrupts)

Οι διακοπές παρέχονται από διάφορες πηγές, εσωτερικές της συσκευής ή εξωτερικές. Η λειτουργία των διακοπών είναι να σταματούν την εκτέλεση του προγράμματος όταν αυτό ζητηθεί και η διακλάδωση της εκτελεσής του σε κάποιο άλλο σημείο του προγράμματος. Οι διακοπές εκτελούνται ανάλογα με την προτεραιότητα που έχει δηλωθεί στο πρόγραμμα ή στο υλικό της συσκευής. Αναφορικά μερικές ενδεικτικές πηγές μπορεί να είναι:

Εσωτερικές

- Υπερχείλιση μετρητή
- Τέλος μετατροπής στον αναλογικο-ψηφιακό μετατροπέα (για όσες συσκευές τον διαθέτουν)

Εξωτερικές

- Σήματα διακοπής από τις ακίδες INT, INT1, και INT2
- Μεταβολές της λογικής στάθμης των ακίδων

Αρχικοποίηση (Reset)

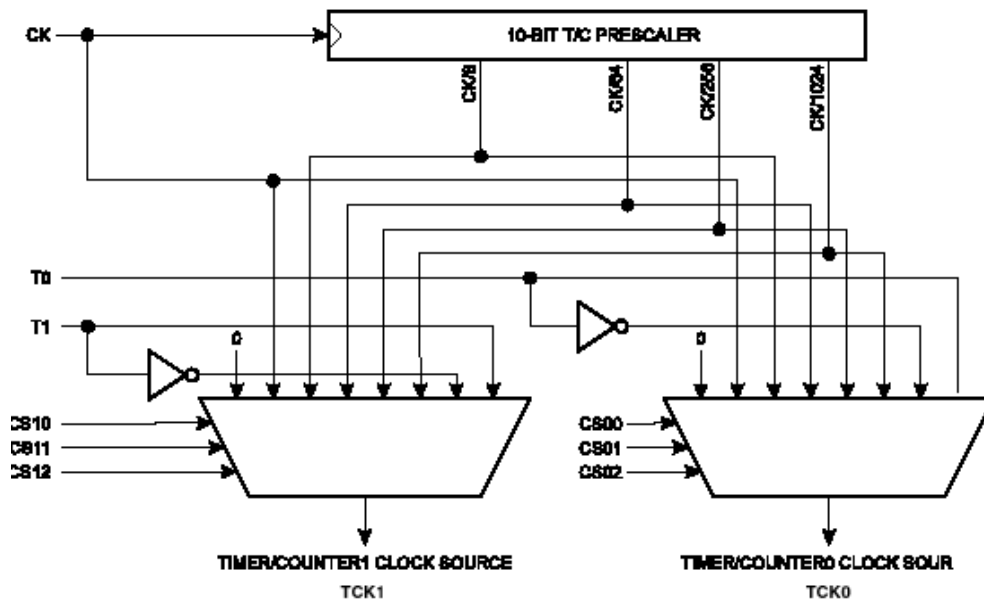
Η λογική αρχικοποίησης χρησιμοποιείται για να τοποθετήσει τη συσκευή σε μία γνωστή κατάσταση. Η πηγή του σήματος αρχικοποίησης μπορεί να καθοριστεί με τα ψηφία κατάστασης της συσκευής (Device Status Bits). Οι συσκευές διαφοροποιούνται μεταξύ τους με τα διάφορα είδη αρχικοποίησης όπως απεικονίζονται παρακάτω. Ένα απλοποιημένο διάγραμμα του κυκλώματος αρχικοποίησης φαίνεται στο σχήμα 6. Αυτό το κύκλωμα αποτελεί ένα υπερσύνολο όλων των σημάτων αρχικοποίησης στους μικροελεγκτές της σειράς AVR.

Ο ακροδέκτης Reset προκαλεί Επανεκκίνηση όταν πέσει σε λογικό 0. Κατά την έναρξη τροφοδότησης του ολοκληρωμένου αυτό γίνεται όταν η τάση ξεπεράσει ένα όριο και εφόσον το σήμα Reset είναι συνδεδεμένο στην τάση τροφοδοσίας. Το Reset συχνά συνδέεται μέσω μιας αντίστασης στην τροφοδοσία και μέσω ενός πυκνωτή προς τη γη. Αυτό συμβαίνει όταν η τάση τροφοδοσίας δεν φτάνει απότομα την τελική τιμή της. Επίσης η γραμμή Reset μπορεί να συνδεθεί και με κάποιο λογικό κύκλωμα που προκαλεί Επανεκκίνηση όταν συντρέχει κάποιος ειδικός λόγος για καθυστερημένη ολοκλήρωση της διαδικασίας Επανεκκίνησης.

Απο όλες τις περιφερειακές συσκευές που βρίσκονται στο κέλυφος του μικροελεγκτή ATMEGA16 αυτές που χρησιμοποιήθηκαν για την υλοποίηση της ρομποτικής πλατφόρμας είναι οι θύρες I/O, οι χρονιστές και οι διακοπές.

Χρονιστές/μετρητές

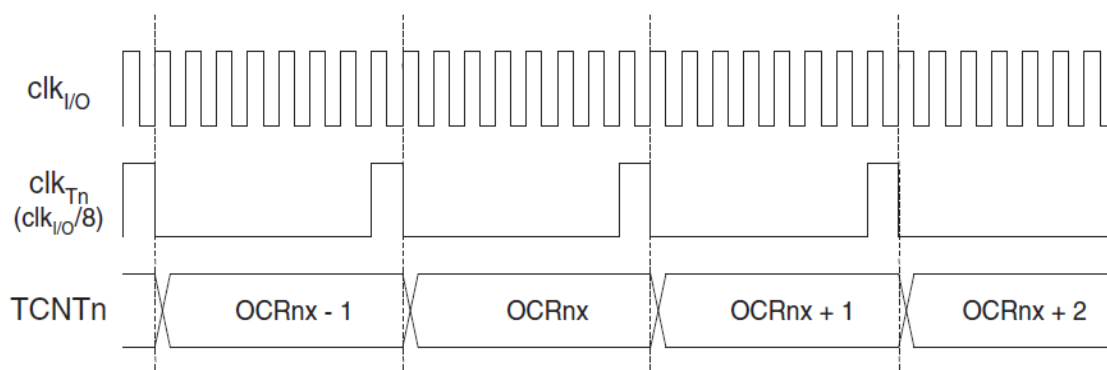
Ο ATmega16 διαθέτει 3 χρονιστές/μετρητές (timer/counter). Ο ένας είναι 8-bit και ονομάζεται T/C0 ενώ ο δεύτερος και ο τρίτος είναι 16-bits και ονομάζεται T/C1-2. Και οι δύο μπορούν να χρησιμοποιήσουν είτε εσωτερικό ρολόι οπότε λειτουργούν ως χρονιστές ή να χρησιμοποιήσουν κάποια εξωτερική πηγή παλμών των οποίων το πλήθος να μετράνε οπότε λειτουργούν ως μετρητές. Και στις δύο περιπτώσεις η συχνότητα της πηγής των παλμών μπορεί να υποδιαιρεθεί με τις τιμές /8, /64, /256, /1024 μέσω του κοινού prescaler ο οποίος απεικονίζεται στο Σχήμα 6,7.



Σχήμα 6. Χρήση του διαιρέτη prescaler για την επιλογή του χρονισμού των μετρητών

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0, falling edge
1	1	1	External Pin T0, rising edge

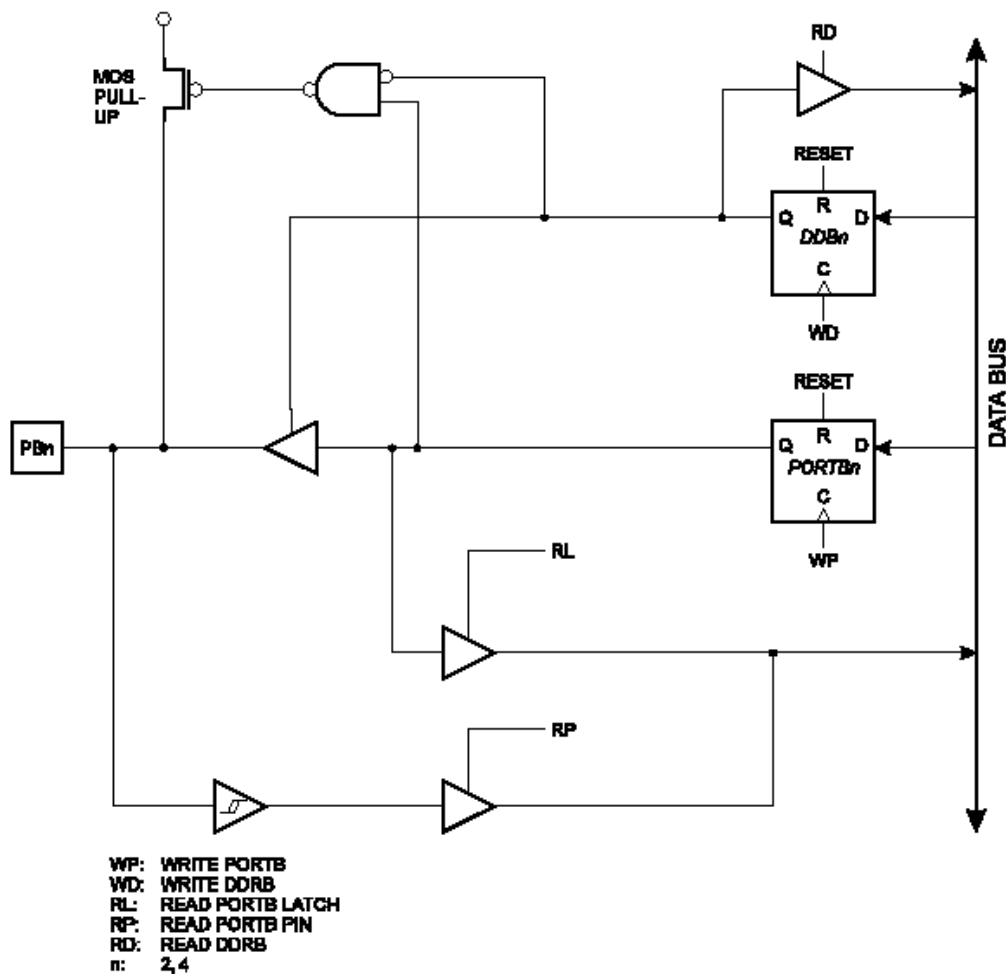
Σχήμα 7. Οι επιλογές που παρέχονται μέσω του prescaler. Οι δύο τελευταίες αφορούν την λειτουργία ως μετρητή.



Σχήμα 8. Ενδεικτική λειτουργία του χρονισμού ενός μετρητή. Φαίνονται τα σήματα του ρολογιού του συστήματος καθώς και το διαιρεμένο σήμα χρονισμού του prescaler.

Θύρες εισόδου/εξόδου (I/O PORTS)

Ο μικροελεγκτής έχει 32 ακίδες εισόδου - εξόδου. Πια θα είναι είσοδοι και πια έξοδοι δηλώνεται μέσα από το πρόγραμμα. Οι 32 αυτές ακίδες χωρίζονται σε τέσσερις ομάδες που ονομάζονται θύρες. Σε κάθε μια από τις θύρες αντιστοιχεί ένα όνομα που δηλώνεται με τα γράμματα A,B,C,D, παραδείγματος χάριν θύρα PORTB. Μέσω των ακίδων αυτών είναι δυνατόν η εισαγωγή ή εξαγωγή δεδομένων και για αυτό ονομάζονται θύρες εισόδου-εξόδου (I/O).



Σχήμα 9. Τυπική θύρα εισόδου - εξόδου

Οι ακίδες I/O γενικού σκοπού μπορούν να θεωρηθούν ως ο απλούστερος τύπος περιφερειακής μονάδας. Επιτρέπουν στον μικροελεγκτή την επιτηρήση και τον έλεγχο άλλων συσκευών. Για να προσθέσουν μεγαλύτερη ευελιξία στη λειτουργία του μικροελεγκτή ATMEGA16, η Atmel πρόσθεσε σε μερικές ακίδες και μια εναλλακτική λειτουργία. Αυτές οι λειτουργίες εξαρτώνται από ποια περιφερειακά συστήματα ενσωματώνονται στη συσκευή. Γενικά, όταν ένα περιφερειακό λειτουργεί, εκείνη η ακίδα δεν μπορεί να χρησιμοποιηθεί όπως μια I/O ακίδα γενικού σκοπού.

Για τους περισσότερες θύρες, η κατεύθυνση των δεδομένων στην ακίδα I/O ελέγχεται από τον καταχωρητή κατεύθυνσης δεδομένων, που ονομάζεται DDR. Ο καταχωρητής DDR ελέγχει την κατεύθυνση των δεδομένων της θύρας με τον εξής τρόπο. Λογικό ένα '1' σε ένα ψηφίο του καταχωρητή DDR αντιστοιχεί σε εκείνη την ακίδα που τα δεδομένα θα εξάγονται, ενώ '0' αντιστοιχεί σε εκείνη την ακίδα που εισάγει δεδομένα.

Ο καταχωρητής θύρας είναι αυτός που εφαρμόζει τα δεδομένα στις ακίδες της αντίστοιχης θύρας. Όταν η θύρα 'διαβάζεται', η συσκευή διαβάζει τα επίπεδα τάσης που είναι παρόντα στις ακίδες I/O. Αυτό σημαίνει ότι πρέπει να είμαστε προσεκτικοί με τις εντολές εγγραφής-ανάγνωσης-μεταφοράς δεδομένων στις θύρες και την αλλαγή της κατεύθυνσης μιας ακίδας από είσοδο σε έξοδο. Το σχήμα 10 παρουσιάζει μια χαρακτηριστική θύρα I/O. Το σχήμα αυτό δεν λαμβάνει υπόψη τις όποιες περιφερειακές λειτουργίες μπορεί να συνυπάρχουν επάνω στη ακίδα.

Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Σχήμα 10. Καταχωρητές της θύρας PORTA

Καταχωρητές εισόδου-εξόδου

Ο καταχωρητής PORTX είναι μια θύρα των 8-bit αμφίδρομης κατεύθυνσης δεδομένων. Ο αντίστοιχος καταχωρητής κατεύθυνσης είναι ο DDRX. Όλες οι ακίδες έχουν εισόδους τύπου TTL. Κάνοντας ένα bit μηδέν '0' στον καταχωρητή DDRX θέτει τον αντίστοιχο οδηγό εξόδου σε κατάσταση υψηλής σύνθετης αντίστασης. Εάν ένα ψηφίο γίνει '1' στον καταχωρητή DDRX τότε τοποθετεί το περιεχόμενο του μανδαλωτή εξόδου στην επιλεγμένη γραμμή. Όλες οι ακίδες δηλώνονται ως εισόδοι έπειτα από μια αρχικοποίηση του μικροελεγκτή.

Αυτόνομη ρομποτική πλατφόρμα

Αναπτυξιακό σύστημα STK200

Το αναπτυξιακό σύστημα της εφαρμογής είναι το STK200. Η πλακέτα αυτή μπορεί να φιλοξενήσει πλήθος μικροελεγκτών της οικογένειας AVR. Εκτός από τον μικροελεγκτή το αναπτυξιακό είναι εφοδιασμένο με συσκευές απεικόνισης τύπου LED όπως και Buttons για την εύκολη και γρήγορη σχεδίαση και υλοποίηση των εφαρμογών. Είναι επεκτάσιμο εφ' όσον παρέχει ακίδοσειρές για πρόσβαση σε όλες τις θύρες εισόδου εξόδου του μικροελεγκτή καθώς και μερικά υποστηρικτικά εξαρτήματα για τις εναλλακτικές λειτουργίες των εν λόγω θυρών. Προσφέρονται επίσης πολλές επιλογές επικοινωνίας όπως η ασύγχρονη σειριακή θύρα.

Πλατφόρμα – σασί

Μετά απο σχετική έρευνα αποφασίστηκε ότι το σασί του ρομπότ θα έχει σχήμα δίσκου αποφεύγοντας τις γωνίες που μπορούν να προκαλέσουν προβλήματα κατα την κίνησή του σε κάποιο περιβάλλον. Για να ενισχύσουμε αυτό το πλεονεκτημα, αποφασίσαμε επίσης να βάλουμε τους κινητήρες, επομένως και τις ρόδες, στο κέντρο της βάσης έτσι ώστε αυτό να δύναται να περιστραφεί γύρω απο τον εαυτό του αποφεύγοντας να κάνει όσο το δυνατόν μανούβρες αποφυγής δυσκολων καταστάσεων όπως εμπόδια ή κλειστούς χώρους.

Τροφοδοσία

Ένα μικρό πακέτο επαναφορτιζόμενων μπαταριών χρησιμοποιείται για να παρέχει την απαιτούμενη ισχύ τόσο στους κινητήρες όσο και στα ηλεκτρονικά κυκλώματα. Δύο αντι-διαμετρικά μηχανές κίνησης κινούν το ρομπότ. Μια τρίτη

ανενεργός ρόδα ελεύθερης κίνησης και περιστροφής (castor) χρησιμοποιείται για να παρέχει τη σταθερότητα που απαιτείται μιας και οι κινητήρες είναι τοποθετημένοι στο κέντρο του σασί.

Σερβοκινητήρες

Οι servos είναι κυρίως DC κινητήρες με ενσωματωμένο σύστημα γραναζιών (gearbox) και κύκλωμα ελέγχου ανάδρασης (feedback control loop circuitry). Ο πιο κοινός τύπος servo είναι αυτός ελέγχου θέσης (positional control). Οι servos είναι ηλεκτρικοί ή εν μέρει ηλεκτρονικοί, χρησιμοποιώντας έναν ηλεκτρικό κινητήρα ως βασικό μέσο για τη δημιουργία μηχανικής δύναμης. Τυπικοί servos δίνουν μια περιστροφική έξοδο. Πρισματικοί (linear) τύποι είναι επίσης συχνοί και δίνουν πρισματική (ευθεία) κίνηση. Άλλα είδη servos βασίζονται σε υδραυλικά, πνευματικά ή μαγνητικά συστήματα. Παρακάτω θα αναφερθούμε στους servos που χρησιμοποιήσαμε για την ρομποτική εφαρμογή.

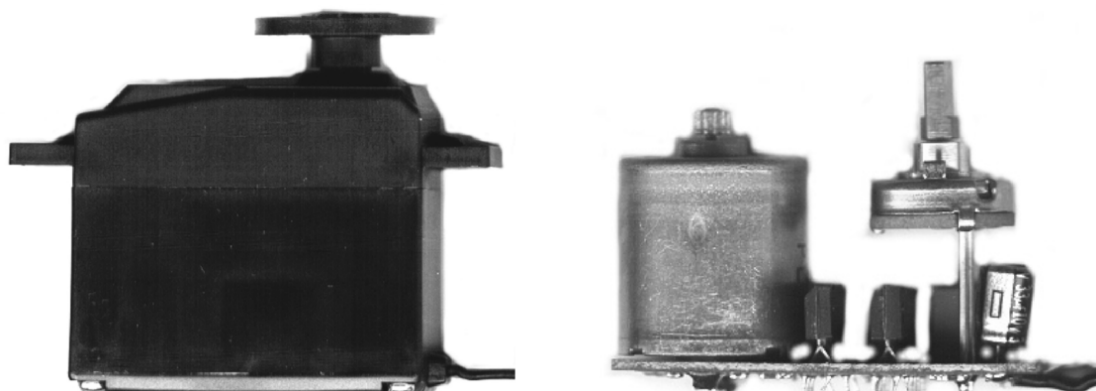
Οι R/C servos είναι πολύ δημοφιλείς στα τηλεκατευθυνόμενα μοντέλα όπως αεροπλάνα, πλοία και φυσικά τις μικρές ρομποτικές εφαρμογές. Οι περισσότεροι servos περιστρέφονται από 90 έως 180 μοίρες. Μερικοί περιστρέφονται 360 μοίρες ή περισσότερο. Αλλά οι servos είναι αδύνατο να περιστρέφονται συνέχεια (continuous rotation) δηλαδή δεν μπορούν να χρησιμοποιηθούν πχ για να κινούν ρόδες (εκτός και αν τροποποιηθούν), αλλά ο ακριβής καθορισμός θέσης που έχουν τους κάνει ιδανικούς για ρομποτικούς βραχίονες, πόδια κλπ.

Οι servos είναι ανεξάρτητοι, ο χειρισμός της ταχύτητας και γωνίας είναι εύκολο να υλοποιηθεί, και το κόστος τους είναι αρκετά χαμηλό. Για να χρησιμοποιηθεί ένα servo τέτοιου τύπου απαιτούνται τρεις μόλις γραμμές δύο εκ των οποίων αφορούν την τροφοδοσία του μηχανισμού και μία μόλις γραμμή τον έλεγχο της θέσης του (σχήμα 12). Η εισαγωγή τους στην εφαρμογή είναι πολύ εύκολη, απλά συνδέουμε το μαύρο καλώδιο του στη γείωση, το κόκκινο σε 4,8-6V τάση και το κίτρινο/άσπρο καλώδιο σε μία γεννήτρια παλμών (ο

μικροελεγκτής στη δική μας περίπτωση). Μεταβάλλοντας τον τετραγωνικό παλμό από 1-2ms ο servo είναι σε θέση να περιστραφεί μέχρι να φθάσει στην επιθυμητή θέση όπου και τερματίζεται η κίνησή του.

Το ρομπότ της εφαρμογής που παρουσιάζεται είναι μια χαμηλού κόστους κινούμενη πλατφόρμα ρομποτικής που σχεδιάστηκε για εκπαιδευτικούς κυρίως σκοπούς. Το ρομπότ αυτό έχει ως σκοπό να είναι ένας μικρός αυτόνομος φορέας που είναι προγραμματισμένος για να κινηθεί με απλά αισθητηριακά ερεθίσματα. Μια ευρεία γκάμα αισθητήρων μπορεί να συνδεθεί εύκολα στην πλατφόρμα για όποιον θελήσει να επεκτείνει την κατασκευή αυτή. Η στρογγυλή πλατφόρμα φτιάχτηκε από τυπικό πλαστικό το οποίο είναι εύκολο στη κοπή και επεξεργασία και άμεσα διαθέσιμο.

Το ρομπότ μπορεί να κινηθεί μπροστά, πίσω και να περιστραφεί γύρω από το κέντρο βάρους του. Δύο σχετικά ανέξοδα R/C servos ελέγχου χρησιμοποιούνται ως μηχανές κίνησης. Ο μικροελεγκτής του αναπτυξιακού είναι υπεύθυνος για τον έλεγχο τόσο των κινητήρων όσο και για την ανάγνωση και αποκωδικοποίηση της πληροφορίας που εισάγεται μέσω των αισθητηρίων. Τα servos R/C έχουν τροποποιηθεί για να λειτουργήσουν ως μηχανές κίνησης πλήρους περιστροφής. Τα servos ελέγχονται από τους παλμούς συγχρονισμού που παράγονται από τον μικροελεγκτή.



Σχήμα 12. Ένα τυπικό σέρβο τύπου μοντελισμού

Ένας χαρακτηριστικό R/C σερβο παρουσιάζεται στο σχήμα 12. Τα Servos έχουν ένα τροχό περιστροφικής κίνησης που ελέγχεται από ένα κωδικοποιημένο σήμα. Ο σερβο που παρουσιάζεται είναι ένα Futaba S3003 που είναι πολύ δημοφιλές και αντιστοιχεί σε μια πλειάδα μοντέλων σέρβο άλλων κατασκευαστών, οπότε ότι ισχύει για την μετατροπή του σε κινητήρα συνεχούς κίνησης θα ισχύει και για τα αντίστοιχα μοντέλα.

Οι ραδιο σερβοκινητήρες ελέγχου παράγονται μαζικά για την αγορά των εφαρμογών χόμπι και είναι επομένως σχετικά ανέξοδος και εύκολα διαθέσιμος. Είναι ταυτόχρονα ιδανικοί για τις εφαρμογές ρομποτικής μιάς κάποιας κλίμακας. Εσωτερικά, ο σερβο περιέχει μια κινητήρας συνεχούς κίνησης (που βλέπει επάνω το αριστερό στο σχήμα 13.2), τα ενσωματωμένα στοιχεία του κυκλώματος ελέγχου, και ένα σύστημα μειωτήρα στροφών. Είναι μικροί, παράγουν μια σχετικά μεγάλη ροπή για το μέγεθός τους, και τρέχουν με την κατάλληλη ταχύτητα για μια κινητήρας κίνησης ρομποτικής, δηλαδή με ταχύτητες της τάξης των 60-160 στροφών ανα λεπτό.

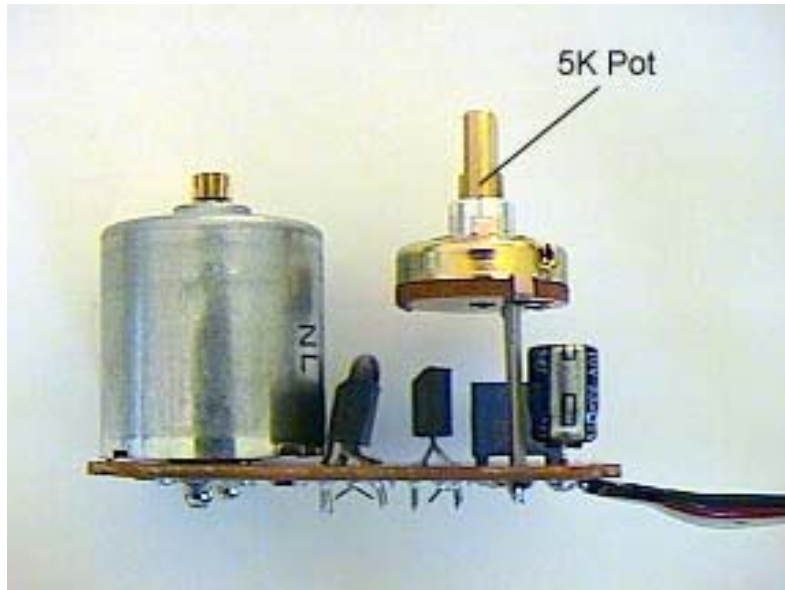
Το κύριο στοιχείο του κυκλώματος ελέγχου των σερβο είναι ένα ποτενσιόμετρο (μεταβλητός αντιστάτης) που χρησιμοποιείται για να αισθανθεί τη γωνιακή θέση του άξονα περιστροφής. Το ποτενσιόμετρο είναι το ψηλό εξάρτημα στο σχήμα 12. Ο άξονας περιστροφής ενός R/C σερβο κινείται κανονικά με 180-210 στροφές ανα λεπτό. Ένα ενιαίο κομμάτι ελέγχου χρησιμοποιείται για να διευκρινίσει τη γωνιακή θέση του άξονα. Ο συγχρονισμός αυτού του κομματιού δείχνει και τη γωνιακή θέση του άξονα. Το ποτενσιόμετρο μετρά τη γωνία περιστροφής, και εάν ο άξονας δεν είναι στη σωστή γωνία, το κύκλωμα εσωτερικού ελέγχου γυρίζουν τη κινητήρας στη σωστή κατεύθυνση έως ότου αισθανθεί την επιθυμητή γωνία.

Η επιθυμητή γωνία κωδικοποιείται χρησιμοποιώντας τετραγωνικούς παλμούς διαμόρφωσης πλάτους (PWM). Το πλάτος του ενεργού υψηλού παλμού ποικίλλει από 1-2ms. Ένας 1ms παλμός αντιστοιχεί σε 0 μοίρες, και 1.5ms είναι 90 μοίρες και ένας παλμός διάρκειας 2ms αντιστοιχεί σε περίπου 180 μοίρες και ούτω καθεξής. Οι νέοι παλμοί συγχρονισμού στέλνονται στο σερβο κάθε 20ms.

Μετατροπή των σερβοκινητήρων

Κανονικά, ένας σερβο έχει μια μηχανική στάση που το αποτρέπει από τη κίνηση πέρα της μίας πλήρους περιστροφής. Εάν αυτή η στάση αφαιρεθεί μαζί με άλλες τροποποιήσεις στο ποτενσιόμετρο, ένας σερβο μπορεί να μετατραπεί σε μια συνεχώς περιστρεφόμενη κινητήρας κίνησης. Οι τροποποιήσεις στο σερβο δεν είναι αντιστρέψιμες και παύουν την εγγύηση αγοράς τους. Μερικοί προμηθευτές εξαρτημάτων ρομπότ πωλούν τα servos ήδη τροποποιημένα.

Για να τροποποιήσετε το σερβο, ανοίξτε την συσκευασία με την αφαίρεση των βιδών και σημειώστε προσεκτικά τη θέση των στοιχείων, ιδιαίτερα των γραναζιών, έτσι ώστε να μπορούν να συγκεντρωθούν και συναρμολογηθούν εκ νέου αργότερα (σχήμα 13). Το ποτενσιόμετρο μπορεί να αντικατασταθεί με δύο αντιστάτες των $2.2K\Omega$ ($\frac{1}{4}$ Watt) ή απλώς να αποσυνδεθεί με την κοπή του άξονα και τη ρύθμιση του σε μία κεντρική θέση έτσι ώστε να ισορροπεί στη θέση που αντιστοιχεί στις 180 μοίρες. Μια ακριβέστερη ρύθμιση μπορεί να επιτευχθεί με την αποστολή στο σερβο ενός παλμού χρονικής διάρκειας 1.5ms και τη ρύθμιση του ποτενσιόμετρου μέχρι ο κινητήρας να σταματήσει. Το ποτενσιόμετρο μπορεί έπειτα να κολληθεί με κόλλα στιγμής για να μην μετακινηθεί από το σημείο ισορροπίας.



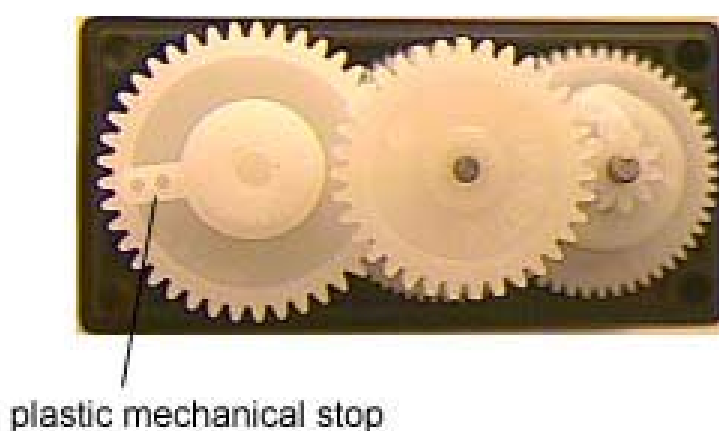
Σχήμα 13. Σερβοκινητήρας εκτός κελύφους

Στην κεντρική θέση το ποτενσιόμετρο θα έχει την ίδια αντίσταση από κάθε ένα από τους εξωτερικούς ακροδέκτες σε σχέση με τον κεντρικό ακροδέκτη (σχήμα 14). Εάν το ποτενσιόμετρο αντικαθίσταται με δύο αντιστάτες, ένας αντιστάτης συνδέεται μεταξύ κάθε μιας από τις δύο οπές που αντιστοιχούν στους εξωτερικούς ακροδέκτες του ποτενσιόμετρου ενώ η άλλη άκρη και των δύο αντιστατών συνδέονται μαζί στην κεντρική οπή που αντιστοιχεί στον κεντρικό ακροδέκτη του ποτενσιόμετρου.



Σχήμα 14. Δικτύωμα αντιστάσεων για την αντικατάσταση του ποτενσιόμετρου

Το μεγαλύτερο γρανάζι που οδηγεί απ' ευθείας τον άξονα περιστροφής, κανονικά έχει μια μικρή προεξοχή που χρησιμεύει ως μηχανική στάση (σχήμα 15). Μετά από την αφαίρεση της βίδας στον άξονα περιστροφής και την αφαίρεση του μεγάλου γραναζιού, η μηχανική στάση μπορεί να αφαιρεθεί προσεκτικά με ένα μικρό πριόνι, ένα μαχαίρι, ή ένα μικρό περιστροφικό εργαλείο τύπου Dremel.



Σχήμα 15. Μηχανική στάση στον σερβοκινητήρα

Ο σερβο έπειτα προσεκτικά συναρμολογείται εκ νέου. Μετά από τις τροποποιήσεις, εάν ένας παλμός πιά σύντομος από 1.5ms στέλνεται, ο κινητήρας θα περιστραφεί συνεχώς σε μια κατεύθυνση. Εάν ένας παλμός πιά μακροχρόνιος από την 1.5ms στέλνεται η κινητήρας θα περιστραφεί συνεχώς στην άλλη κατεύθυνση. Η θέση 1.5ms ή 90 μοιρών καλείται μερικές φορές ως ουδέτερη θέση ή νεκρή ζώνη.

Το σήμα κίνησης στο κινητήρα είναι ανάλογο, έτσι μεγαλύτερα ή μικρότερα σήματα κοντά στην ουδέτερη θέση αλλάζουν και την ταχύτητα του σέρβο αλλά σε περιορισμένο βαθμό. Αυτό μπορεί να χρησιμοποιηθεί για να ελέγξουμε την ταχύτητα του κινητήρα εάν η ουδέτερη θέση ρυθμίζεται προσεκτικά. Ένα πλάτος σφυγμού 0ms ή κανενός παλμού θα σταματήσει το σερβοκινητήρα. Ένας σερβο έχει τρία καλώδια, το συνεχές δυναμικό των +4 έως +6 βολτ, τη

γή, και το καλώδιο σημάτων. Η διάταξη των τριών σημάτων στο συνδετήρα ποικίλλει μεταξύ των διαφορετικών κατασκευαστών σερβο.

Για τα servos Futaba, το κόκκινο καλώδιο είναι +5, το Μαύρος η γή, και το άσπρο ή κίτρινο καλώδιο είναι η γραμμή σημάτων πλάτους παλμών. Για τα servos Hitec, το πορτοκαλί ή κίτρινο καλώδιο είναι η γραμμή σημάτων και το κόκκινο είναι +5, και το Μαύρο ή καφέ η γή.

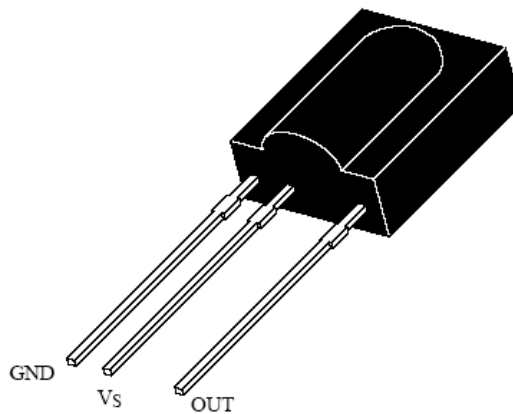
Έλεγχος των σέρβοκινητήρων

Για να οδηγήσει τους κινητήρες ενός σερβο το σήμα πρέπει να σταλεί κάθε 20ms με τα εξής πλάτη παλμού 0ms, 1ms, ή 2ms. Ο μικροελεγκτής είναι προγραμματισμένος για να παραγάγει τα σήματα συγχρονισμού που οδηγούν οι μηχανές. Εάν κανένας παλμός δεν στέλνεται, οι κινητήρες δεν περιστρέφονται. Εάν ένας παλμός 1ms στέλνεται, οι κινητήρες περιστρέφονται δεξιόστροφα και εάν παλμοί 2ms στέλνονται, οι κινητήρες περιστρέφονται με την αντίστροφη κατεύθυνση, αντίθετα προς τη φορά των δεικτων του ρολογιού. Για να προωθήσει το ρομπότ, ο ένας σέρβο κινείται δεξιόστροφα ενώ ο άλλος κινείται αντίθετα προς τη φορά των δεικτων του ρολογιού.

Αυτό δικαιολογείται λόγω του τρόπου που οι μηχανές τοποθετούνται στη βάση του ρομπότ. Στον κώδικα που παρατίθεται στο παράρτημα, διευκρινίζεται η κατεύθυνση για την αριστερή και δεξιά περιστροφή του κάθε κινητήρα. Εάν και τα δύο σήματα είναι 1ms η πλατφόρμα προωθείται προς μία κατεύθυνση (μπροστά). Ο κώδικας που είναι γραμμένος για τον μικροελεγκτή κινεί στην πραγματικότητα τον ένα κινητήρα στην αντίθετη κατεύθυνση για να κινηθεί το ρομπότ μπροστά. Εάν και τα δύο σήματα είναι διάρκειας 0ms το ρομπότ κινείται αντίστροφα δηλαδή προς τα πίσω. Εάν κάποιο σήμα είναι 1ms και άλλο είναι 0ms, το ρομπότ περιστρέφεται γύρω απο τον εαυτό του.

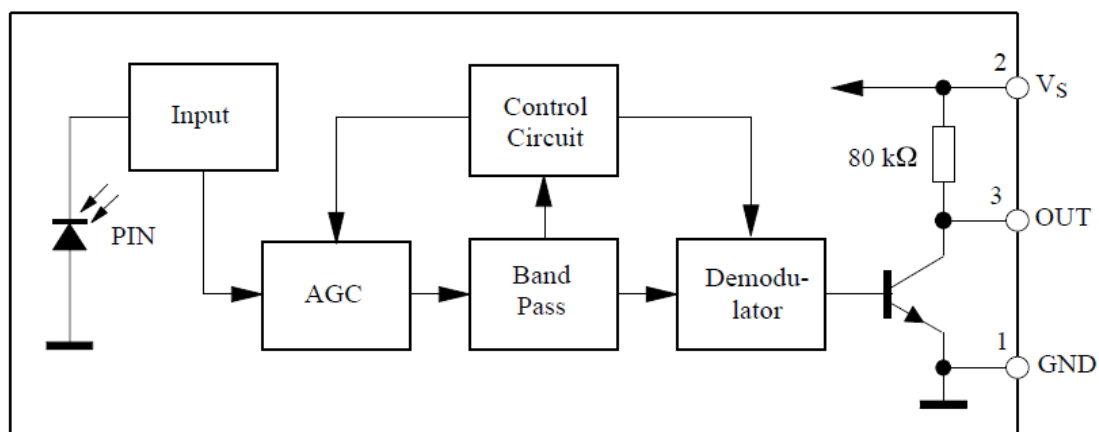
Αισθητήρες

Οι αισθητήρες της εφαρμογής είναι τύπου υπέρυθρων. Το σύστημα αποτελείται από δύο ξεχωριστές μονάδες: Έναν πομπό και ένα δέκτη. Ο πομπός εκπέμπει φως παλμών στην περιοχή μήκους κύματος υπέρυθρου ή κόκκινου φωτός. Όταν ένα εμπόδιο βρεθεί σε απόσταση δράσης τότε μέσω ανάκλασης μέρος του σήματος εξόδου κατευθύνεται στον αισθητήρα υπέρυθρων.



Σχήμα 16. ο αισθητήρας TSOP1736

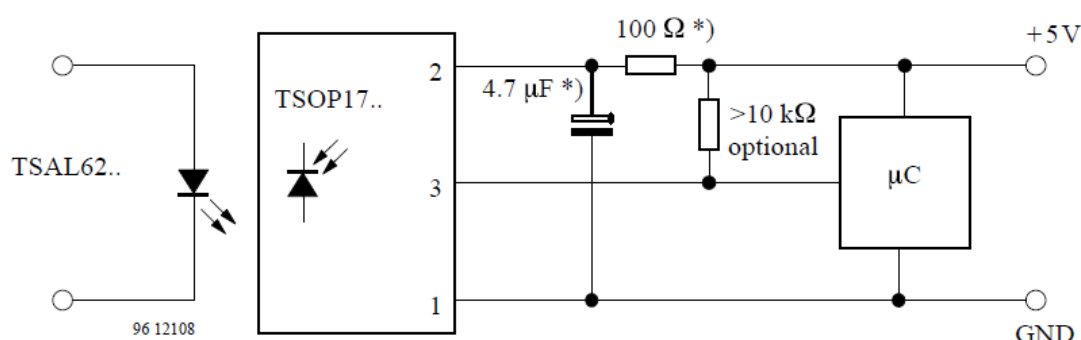
Οι αισθητήρες αυτοί δεν μετράνε απόσταση, αντιλαμβάνονται όμως αν υπάρχει εμπόδιο σε μία περιοχή της τάξης των μερικών εκατοστών έως μερικών δεκάδων εκατοστών. Οι δέκτες αυτοί βρίσκονται είτε σε μορφή phototransistors είτε σε μορφή ολοκληρωμένων κυκλωμάτων. Το πλεονέκτημα των τελευταίων είναι ότι δεν επηρεάζονται από το διάχυτο υπέρυθρο φως του περιβάλλοντος.



Για τους τελευταίους, ο κατασκευαστής δίνει την συχνότητα του εκπεμπόμενου φωτός για το οποίο έχουμε την μέγιστη απόκριση. Για την ρομποτική πλατφόρμα αυτής της εργασίας χρησιμοποιήθηκαν οι πολύ δημοφιλείς αισθητήρες TSOP 1736 (σχήμα 16). Το μπλόκ διάγραμμα αυτών των αισθητήρων φαίνεται στο παραπάνω σχήμα.

Απο τα δύο τελευταία ψηφία του μοντέλου μπορούμε να πληροφορηθούμε για την συχνότητα της μέγιστης απόκρισης για τους οποίους είναι τα 36KHz. Μέσα στη συσκευασία τους οι αισθητήρες αυτοί φέρουν προενισχυτή και η αποδιαμορφωμένη έξοδος τους μπορεί να οδηγήσει απ' ευθείας μικρουπολογιστικές διατάξεις.

Οι αισθητήρες αυτοί έχουν τρεις ακίδες. Οι δύο ακίδες αφορούν την τροφοδοσία του κυκλώματος ενώ ο τρίτος είναι η ψηφιακή έξοδος. Η έξοδος γίνεται λογικό '1' όταν υπέρυθρο σήμα διαμορφωμένο στα 36KHz πέσει πάνω στο εσωτερικό φωτοτρανζίστορ. Στην αντίθετη περίπτωση η έξοδος μένει σε λογικό '0'. Αυτό το χαρακτηριστικό το καθιστά ιδιαίτερα ευκολο στη χρήση όμως δεν μπορούμε να υπολογίσουμε την ποσότητα του φωτός που πέφτει στον αισθητήρα. Είναι ευκολα αντιληπτό ότι η υπέρυθρη ακτινοβολία πρέπει να περάσει μια τιμή κατωφλίου για να μπορέσει να διεγείρει την έξοδο.



Σχήμα 18. Συνδεσμολογία του αισθητήρα

Η σύνδεση του αισθητήρα TSOP 1736 μπορεί να γίνει όπως στο σχήμα 18. Το στοιχείο TSAL62 είναι μια κοινή δίοδος εκπομπός υπέρυθρης ακτινοβολίας τύπου LED. Είναι πολύ κοινή στο εμπόριο και μπορεί να

αντικατασταθεί με σχεδόν οποιαδήποτε άλλη δίοδο LED με παρόμοια χαρακτηριστικά.

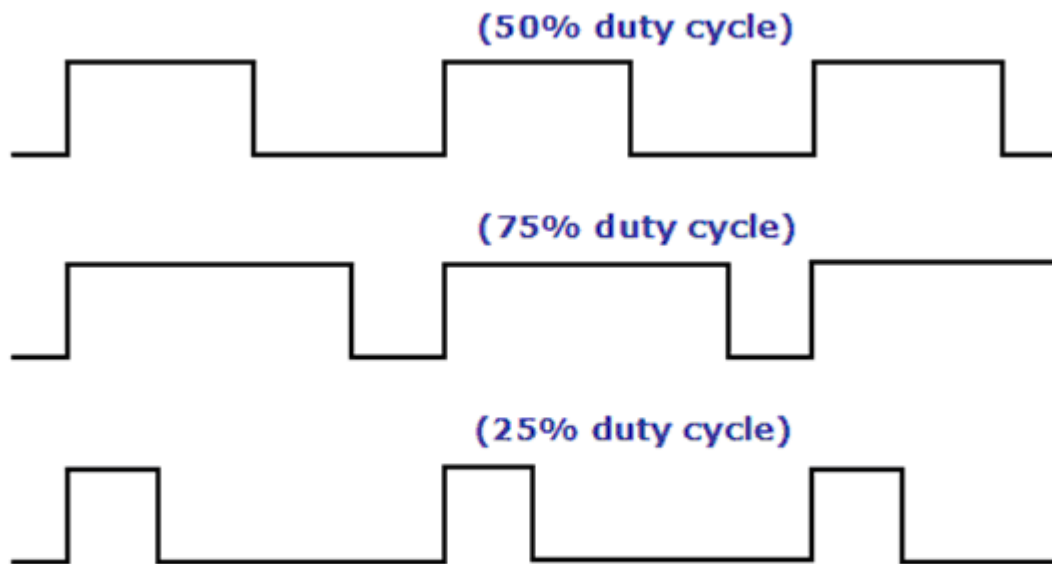
Ο ηλεκτρολυτικός πυκνωτής και η αντίσταση σχηματίζουν ένα δικτύωμα για το φιλτράρισμα αρμονικών που σχετίζονται με τα επίπεδα θορύβου στην τροφοδοσία του κυκλώματος. Πρόκειται για ένα προαιρετικό κύκλωμα το οποίο αν και δεν χρησιμοποιήθηκε στην εφαρμογή του ρομπότ, έδωσε ικανοποιητικά αποτελέσματα. Η τροφοδοσία που παρέχεται στο ρομπότ προέρχεται από συστοιχίες συσσωρευτών που από τη φύση τους παρέχουν σταθεροποιημένη τροφοδοσία.

Διαμόρφωση εύρους παλμών

Pulse Width Modulation, PWM συντομογραφία, είναι μια μέθοδος διαβίβασης πληροφοριών με μια σειρά παλμών. Τα δεδομένα που διαβιβάζονται είναι κωδικοποιημένα ανάλογα με το πλάτος των παλμών αυτών και χρησιμοποιείται κυρίως για τον έλεγχο του ποσού της ενέργειας που αποστέλλεται σε ένα φορτίο. Στην περίπτωση μας το φορτίο είναι οι κινητήρες των σέρβο. Με άλλα λόγια, η τεχνική αυτή της μεταβολής του πλάτους του παλμού, παράγει παλμούς με μεταβλητό πλάτος που αντιπροσωπεύουν το εύρος του αναλογικού σήματος εξόδου.

Πρόκειται για μια δημοφιλή τεχνική μέσω της οποίας ελέγχουμε ένα αναλογικό μέγεθος μέσω ενός ψηφιακού σήματος. Η διαμόρφωση πλάτους παλμού χρησιμοποιείται για να μειώσει τη συνολική ισχύ που παραδίδεται σε ένα φορτίο, χωρίς απώλειες, το οποίο συνήθως συμβαίνει όταν μια πηγή ενέργειας περιορίζεται μέσω μιας αντίστασης (σχήμα 19). Η βασική αρχή στην όλη διαδικασία είναι ότι η μέση ισχύ που παραδίδεται είναι ευθέως ανάλογη προς την διαφοροποίηση καθήκον κύκλου. Εάν το ποσοστό διαφοροποίησης είναι υψηλό, είναι δυνατόν να εξομαλύνει τον παλμό της αμαξοστοιχίας που

χρησιμοποιούν παθητικά ηλεκτρονικά φίλτρα και να ανακτήσει ένα μέσο κύμα αναλογική μορφή.



Σχήμα 19. διαφορετικές μορφές διαμόρφωσης πλάτους παλμών

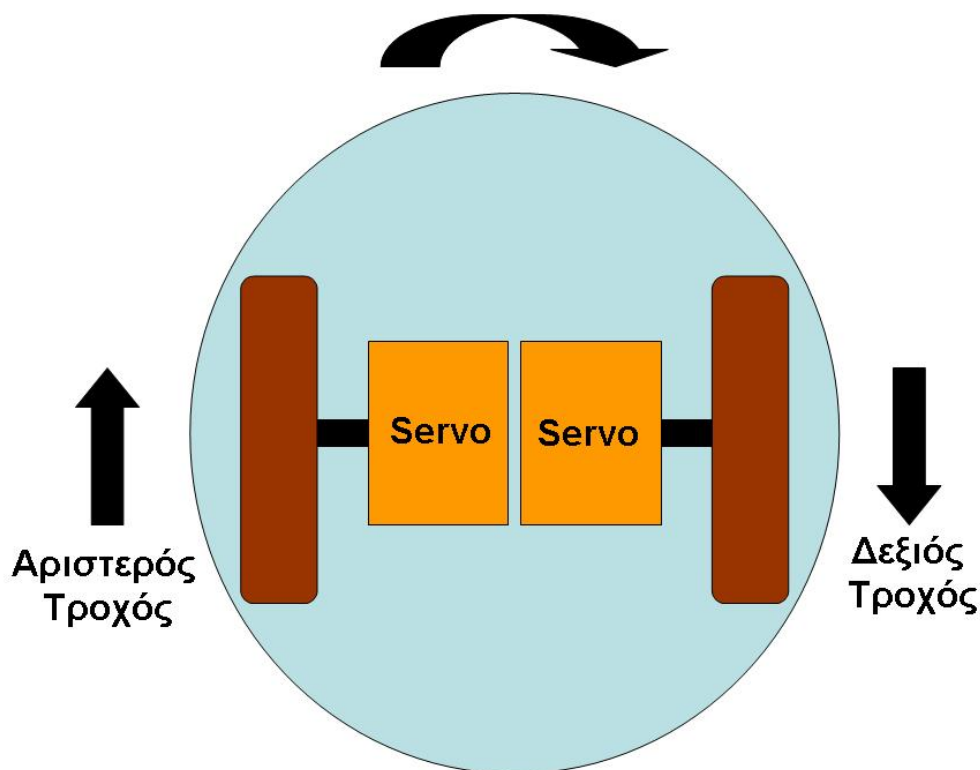
Ένα σήμα με Διαμόρφωση Πλάτους Παλμών είναι σήμα που αποτελείται από θετικούς παλμούς (τετραγωνική μορφή). Οι συχνότητα των παλμών είναι σταθερή, ενώ το πλάτος τους είναι μεταβλητό. Έτσι ανάλογα με το πλάτος των παλμών μεταβάλλεται και η ισχύς που εφαρμόζεται στον κινητήρα, επομένως και οι στροφές του. Ο κινητήρας οδηγείται από ένα συμβατικό αναλογικό ενισχυτή ισχύος που βρίσκεται μέσα στην συσκευασία του σέρβο.

Ρομπότ διαφορικής κίνησης

Το ρομπότ διαφορικής κίνησης αποτελεί την πιο κλασική επιλογή και ίσως την πιο δημοφιλή. Αποτελείται από δύο τροχούς αντιδιαμετρικά τοποθετημένους σε σχέση με το κέντρο βάρους της πλατφόρμας (σχήμα 20). Οι τροχοί αυτοί είναι συνδεδεμένοι σε δύο ανεξάρτητους κινητήρες, στην περίπτωση μας οι δύο σερβομηχανισμοί, έτσι ώστε κάθε τροχός να κινείται ανεξάρτητα.

Αυτή η τεχνική προσδίδει μεγάλη ευελιξία στην πλατφόρμα επειδή μπορεί εύκολα να αποφύγει εμπόδια ή να απομακρυνθεί από μια κλειστή περιοχή. Όταν οι δύο τροχοί κινούνται με την ίδια ταχύτητα και με αντίθετη φορά, το ρομπότ κινείται μπροστά ή πίσω ανάλογα με τη φορά κίνησης των τροχών.

Οποιοσδήποτε άλλος συνδυασμός μετακινεί την πλατφόρμα σε οποιαδήποτε κατεύθυνση εκτελώντας κυκλική κίνηση. Ενδεικτικά αναφέρουμε μια ακραία κατάσταση κίνησης, όταν οι φορές κίνησης και οι ταχύτητες είναι ίδιες, το ρομπότ περιστρέφεται γύρω από το κέντρο βάρους του.



Σχήμα 20. τυπική πλατφόρμα κίνησης διαφορικής μετάδοσης

Τροχοί

Οι τροχοί που χρησιμοποιήθηκαν στην ρομποτική πλατφόρμα είναι τυπικοί τροχοί μοντελισμού από αφρώδες υλικό (σχήμα 21). Οι τροχοί αυτοί κομπλάρονται εύκολα στους σερβο-κινητήρες και έχουν πολύ μικρό βάρος. Για την αντιστήριξη της κατασκευής χρησιμοποιήθηκε επίσης ένας τροχός ελεύθερης κίνησης (caster) ο οποίος είναι παντο-κατευθυντικός σε ότι αφορά την κίνησή του (εικόνα 22).

Ένα από τα πλεονεκτήματα που έγινε εμφανές κατά την σχεδίαση της εν λόγω πλατφόρμας είναι και οι ευκολίες που παρέχονται από τους σερβομηχανισμούς για την σύνδεση άλλων μηχανικών μερών. Τα σέρβο είναι εφοδιασμένα με κυκλικές βάσεις (horns) πάνω στις οποίες μπορούν να στηριχθούν εύκολα πολλά είδη τροχών.



Σχήμα 21. τροχοί αφρώδους υλικού



Σχήμα 22. παντοκατευθυντικός τροχός (caster)

Προγραμματισμός του ρομπότ

Γιατί μια γλώσσα υψηλού επιπέδου

Τα μικρουπολογιστικά συστήματα χρησιμοποιούν μια αρκετά εξειδικευμένη γλώσσα με δυαδικά ψηφία και ψηφιολέξεις που μόνο οι επαγγελματίες προγραμματιστές μπορούν να καταλάβουν. Η γλώσσα αυτή ονομάζεται γλώσσα μηχανής. Για τούς πολλούς ανθρώπους θα πρέπει με κάποιο τρόπο να μεταφράζεται το συγκεκριμένο πρόβλημα σε γλώσσα που μπορεί να καταλάβει ο υπολογιστής. Αυτό τον σκοπό έχουν οι γλώσσες προγραμματισμού, από την στιγμή που το πρόβλημα έχει μεταφραστεί σε κάποια γλώσσα που μπορεί να καταλάβει η μηχανή, ο υπολογιστής ακολουθεί τις οδηγίες για να επιλύσει το πρόβλημα. Το πρόγραμμα λοιπόν είναι μία σειρά από οδηγίες που ο υπολογιστής μπορεί να χρησιμοποιήσει για την επίλυση κάποιου προβλήματος.

Η γλώσσα μηχανής μπορεί να γίνει πιο κατανοητή εάν γίνει χρήση αντί για την δεκαεξαδική μίας άλλης γραφής λιγότερο σκοτεινής. Στο πρόγραμμα που είναι σε συμβολική γλώσσα οι κωδικοί των εντολών αντικαταστάθηκαν με συμβολικά ονόματα που είναι εύκολα στην απομνημόνευση, ενώ οι αριθμοί μπορούν να έχουν δεκαδική μορφή.

Στην συμβολική γλώσσα το πρόγραμμα διαβάζεται πιο εύκολα, μειώνονται τα λάθη προγραμματισμού και ελαττώνεται η εξάρτηση του προγράμματος από συγκεκριμένες θέσεις μνήμης. Ακόμα και έτσι όμως, η συμβολική γλώσσα έχει το μειονέκτημα ότι μοιάζει πολύ με γλώσσα μηχανής παρά με ανθρώπινη γλώσσα, εξαρτάται από τον επεξεργαστή, και απαιτεί πολλές γραμμές κώδικα ακόμη και για ένα απλό πρόγραμμα.

Η γλώσσα μηχανής και η συμβολική γλώσσα είναι και οι δύο γλώσσες χαμηλού επιπέδου δηλαδή μία απλή γραμμή οδηγίας γραμμένη από τον προγραμματιστή μεταφράζεται σε μια απλή οδηγία του υπολογιστή. Μέχρι πρόσφατα η ανάπτυξη εφαρμογών σε μικροελεγκτές επιτυγχάνονταν με την

χρήση σχεδόν αποκλειστικά της γλώσσας Assembly. Τα τελευταία χρόνια οι εφαρμογές έγιναν πιο απαιτητικές και σε συνδυασμό με τον συρρικνωμένο χρόνο ανάπτυξής τους, έγινε επιτακτική η χρήση των γλωσσών υψηλού επιπέδου.

Για να απαλλαγεί ο προγραμματιστής από όλες τις δυσκολίες της συμβολικής γλώσσας, δημιουργήθηκαν γλώσσες προγραμματισμού υψηλού επιπέδου, που είναι προσανατολισμένες προς την διευκόλυνση του ανθρώπου και όχι του υπολογιστή. Μια γλώσσα υψηλού επιπέδου χρησιμοποιεί μόνο μια απλή γραμμή οδηγιών για να αντιπροσωπεύσει πολλές γραμμές συμβολικής γλώσσας ή γλώσσας μηχανής. Αυτό ελαττώνει τον χρόνο ανάπτυξης του προγράμματος και το καθιστά λιγότερο εξαρτημένο από τον συγκεκριμένο επεξεργαστή.

Τα προγράμματα όμως που είναι γραμμένα σε γλώσσες υψηλού επιπέδου απαιτούν γενικά περισσότερη μνήμη και εκτελούνται με λιγότερη ταχύτητα από τα προγράμματα που είναι γραμμένα σε συμβολική γλώσσα. Δεν χρησιμοποιούν το σύστημα τόσο αποδοτικά όσο τα προγράμματα σε συμβολική γλώσσα. Μια από τις πλέον διαδεδομένες γλώσσες προγραμματισμού υψηλού επιπέδου είναι και η γλώσσα C που αναπτύχθηκε από τον Dennis Ritchie το 1972.

Μέχρι πρόσφατα σχεδόν όλες οι εφαρμογές που αναπτύσσονταν για μικρουπολογιστικά συστήματα ήταν βασισμένα σε συμβολική γλώσσα λόγω των περιορισμένων πόρων του μικροελεγκτή. Πρόσφατα πολλές εφαρμογές δημιουργούνται και σε γλώσσες υψηλού επιπέδου περιορίζοντας έτσι τον χρόνο ανάπτυξης και σχεδίασης.

Η γλώσσα C

Η C είναι μια γενικής χρήσης διαδικαστική γλώσσα προγραμματισμού η οποία αναπτύχθηκε στις αρχές της δεκαετίας 1970-1980 στα εργαστήρια Bell Labs για να χρησιμοποιηθεί για την ανάπτυξη του λειτουργικού συστήματος UNIX. Απο τότε χρησιμοποιείται ευρύτατα, και ιδιαίτερα για ανάπτυξη προγραμμάτων συστήματος (system software) αλλά και για απλές εφαρμογές. Οι λόγοι της ραγδαίας ανάπτυξης της συγκεκριμένης γλώσσας προγραμματισμού είναι η ταχύτητα της, καθώς και το γεγονός ότι είναι διαθέσιμη στα περισσότερα σημερινά λειτουργικά συστήματα.

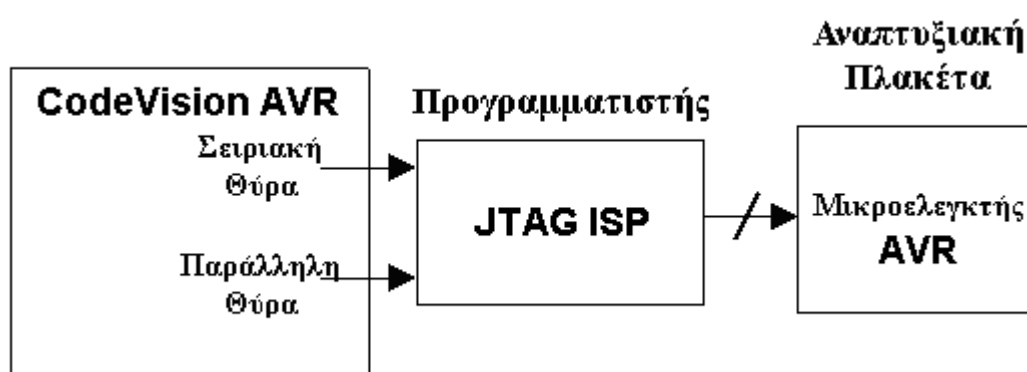
Η C είναι μια σχετικά μινιμαλιστική γλώσσα προγραμματισμού. Ανάμεσα στους σχεδιαστικούς στόχους που έπρεπε να καλύψει η γλώσσα περιλαμβανόταν το ότι θα μπορούσε να μεταγλωττιστεί (να γίνεται compile) άμεσα με τη χρήση single-pass compiler, με άλλα λόγια, ότι θα απαιτούνταν μόνο ένας μικρός αριθμός από εντολές (instructions) σε γλώσσα μηχανής (machine language) για κάθε βασικό στοιχείο της, χωρίς εκτεταμένη run-time υποστήριξη. Ως αποτέλεσμα, είναι δυνατό να γραφτεί κώδικας σε C σε χαμηλό επίπεδο προγραμματισμού με ακρίβεια ανάλογη της συμβολικής γλώσσας, στην πραγματικότητα η C ορισμένες φορές αποκαλείται (και χωρίς να υπάρχει πάντα αντιπαράθεση) "high-level assembly" ή "portable assembly." Επίσης, γίνονται αναφορές στη C ως mid-level γλώσσα προγραμματισμού.

Προγραμματισμός του μικροελεγκτή

Η συσκευή προγραμματισμού χρησιμοποιείται για να μεταφέρει "φορτώσει" τον μεταγλωττισμένο κώδικα (Αρχείο τύπου hex) στην μνήμη προγράμματος του AVR. Στην περίπτωση του ATMEGA16, η μεταφορά μπορεί να γίνει και επάνω στο κύκλωμα με με την λειτουργία ISP (In Circuit Programming). Το ISP είναι ένα πρωτόκολλο σειριακού προγραμματισμού εντός κυκλώματος, πάντοτε με την χρήση μιας συσκευής προγραμματισμού. Με αυτόν τον τρόπο είναι δυνατόν να προγραμματιστεί και η μνήμη EEPROM. Το σχηματικό για

αυτή τη λειτουργία φαίνεται στο σχήμα 23. Ο προγραμματισμός μπορεί να γίνει μέσω σειριακής θύρας ή παράλληλης θύρας ενός προσωπικού υπολογιστή. Τελευταία κυκλοφορούν στο εμπόριο και προγραμματιστές που κάνουν χρήση της θύρας USB του προσωπικού υπολογιστή. Το υλικό μέρος του προγραμματιστή συνδέεται με τον μικροελεγκτή μέσω τεσσάρων μόλις γραμμών (εικόνα 24).

Προσωπικός Υπολογιστής



Σχήμα 23. Διάταξη προγραμματισμού με το πρωτοκολλο ISP

Προγραμματιστής STK200

Ο προγραμματιστής αποτελείται από ελάχιστα εξαρτήματα και είναι ευκολος στην συναρμολόγηση. Η καρδιά του κυκλώματος αποτελεί ένα ολοκληρωμένο κύκλωμα τύπου 74HC244. Το ολοκληρωμένο αυτό εκτελεί χρέη ενίσχυσης του σήματος από και προς τον μικροελεγκτή. Είναι επίσης δυνατόν να παραληφθεί η χρήση του αλλά δεν ενδείκνυται γιατί εκτός των άλλων προστατεύει και τον προσωπικό υπολογιστή που θα χρησιμοποιηθεί για τον προγραμματισμό. Απαιτούνται μόλις τέσσερις γραμμές από τις οποίες δύο μεταφέρουν τα δεδομένα (MOSI, MISO), μία δίνει τον συγχρονισμό των

χρησιμοποιηθεί οποιοσδήποτε προγραμματιστής, ο οποίος είναι συμβατός με τη συσκευή της Kanda Systems STK200

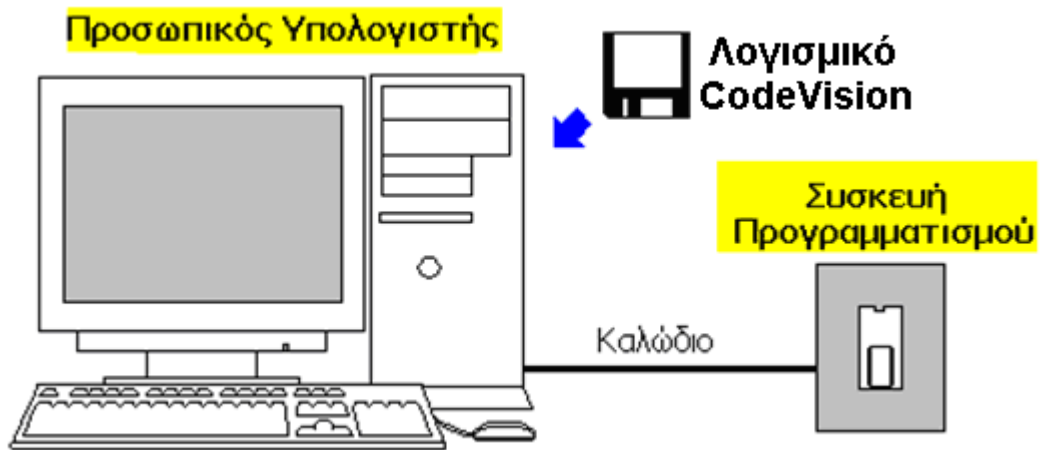
Μεταφορά κώδικα στην μνήμη προγράμματος του AVR

Μετά την μεταγλώττιση του κώδικα, δημιουργείται ένα αρχείο σε γλώσσα μηχανής (HEX file) για παράδειγμα (Test.hex). Το αρχείο αυτό "φορτώνεται" στην μνήμη προγράμματος του AVR. Για να πραγματοποιηθεί αυτή η μεταφορά γίνονται τα παρακάτω:

1. Διαβάζεται το περιεχόμενο του αρχείου HEX, αναγνωρίζονται οι διευθύνσεις μνήμης και τα δεδομένα.
2. Ο μικροεπεξεργαστής μεταβαίνει σε κατάσταση προγραμματισμού/επαλήθευσης.
3. Εφαρμόζεται η τάση προγραμματισμού / επαλήθευσης.
4. Μετατρέπονται τα δεδομένα σε ηλεκτρική μορφή.
5. Μεταφέρονται τα δεδομένα στον μικροεπεξεργαστή με την λειτουργία ISP.
6. Γίνεται επαλήθευση των δεδομένων της μνήμης προγράμματος του PIC με τα δεδομένα του αρχείου HEX

Απαιτούμενος εξοπλισμός

Ένας προσωπικός υπολογιστής, η συσκευή προγραμματισμού, και το λογισμικό (MPLAB) χρειάζονται για την μεταφορά του μεταγλωττισμένου κώδικα (Αρχείο.hex) στην μνήμη προγράμματος του μικροεπεξεργαστή (σχήμα 25).

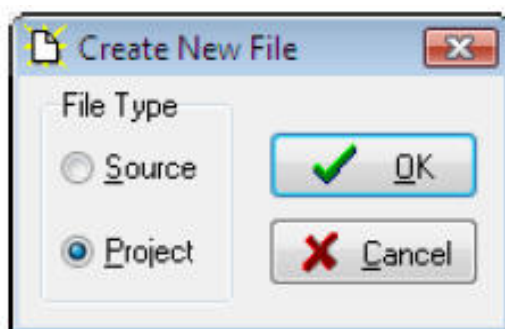


Σχήμα 25. απαιτούμενος εξοπλισμός για τον προγραμματισμό

Το λογισμικό του ρομπότ

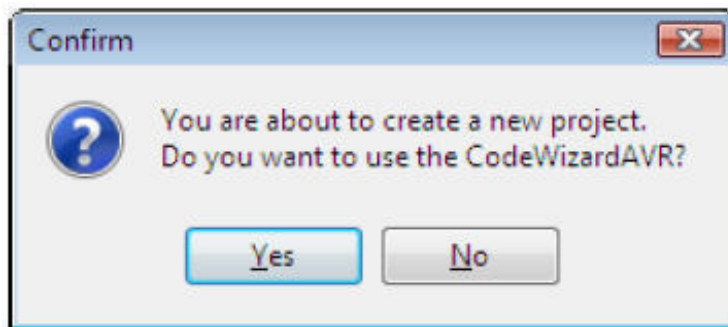
Δημιουργώντας ένα νέο project

Το πρώτο βήμα για την δημιουργία ενός project για μικροελεγκτές AVR με το Codevision, είναι πάντα η δημιουργία ενός αρχείου project. Ο ευκολότερος τρόπος για να το επιτύχει κάποιος αυτό είναι μέσω της υπηρεσίας του 'Project Wizard' που βρίσκεται στο μενού 'Project'. Οι παρακάτω εικόνες παρουσιάζουν βήμα – βήμα όλη τη διαδικασία για την δημιουργία ενός project με έναν μικροελεγκτή τύπου AVR. Με την δημιουργία ενός καινούριου project γίνεται η εκκίνηση της υπηρεσίας με την εμφάνιση του παραθύρου επιλογής όπως φαίνεται στην εικόνα 26.



Σχήμα 26. Δημιουργία ενός καινούριου project

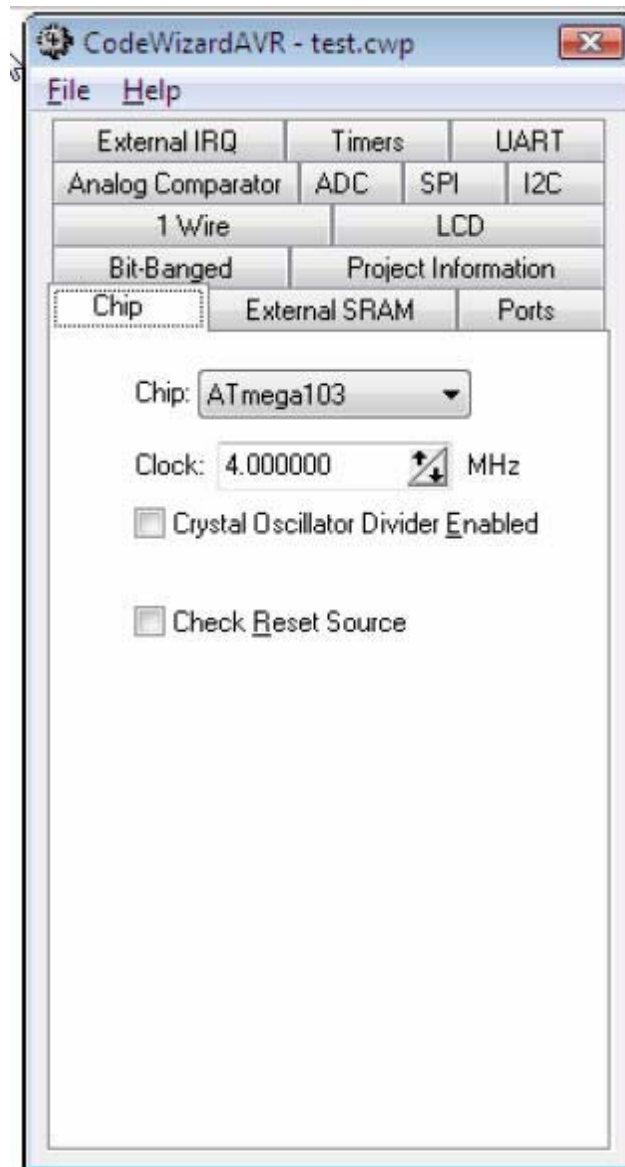
Επιλέγοντας ένα καινούριο project, εμφανίζεται ένα νέο παράθυρο που μας προτρέπει να χρησιμοποιήσουμε τον αυτόματο οδηγό (Wizard) . Για την εφαρμογή μας όπως και για τις περισσότερες εφαρμογές προτείνεται η χρήση του λόγω της ευκολίας στη χρήση και το πλήθος των επιλογών που παρέχει.



Σχήμα 27. Εκκίνηση του αυτόματου οδηγού (Wizard)

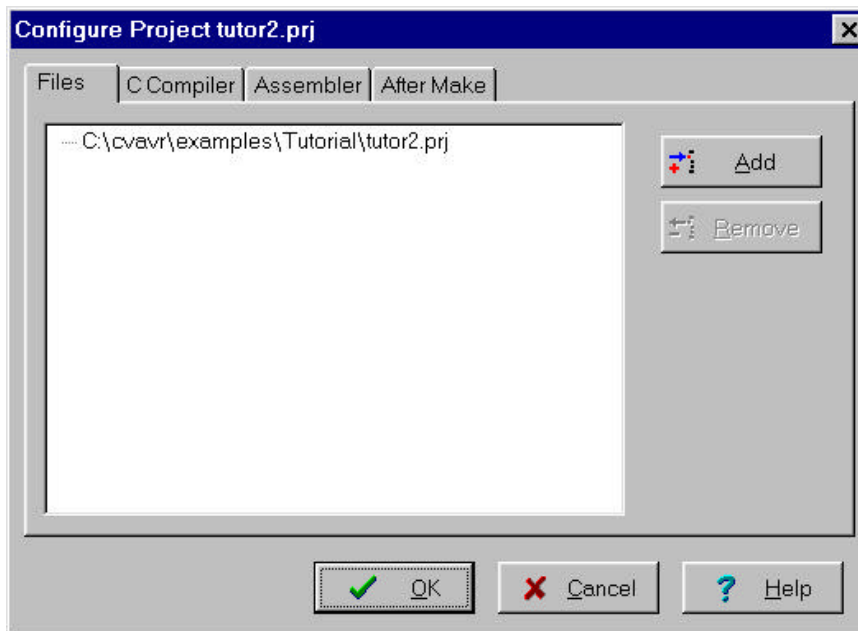
Στην καρτέλα που εμφανίζεται γίνεται η επιλογή του κατάλληλου μοντέλου μικροελεγκτή (σχήμα 28). Αν αυτός δεν είναι ο μικροελεγκτής που θα 'φορτώσει' τον κώδικα της εφαρμογής τότε η συμπεριφορά του μπορεί να γίνει απρόβλεπτη ή να μην δουλέψει καθόλου. Πολλοί προγραμματιστές διαβάζουν την υπογραφή του μικροελεγκτή και την συγκρίνουν με αυτή που αντιστοιχεί στον μικροελεγκτή που χρησιμοποιήθηκε για να γραφτεί ο κώδικας. Στην περίπτωση που δεν είναι ίδιες δεν είναι δυνατός ο προγραμματισμός και ένα μήνυμα σφάλματος εμφανίζεται στην οθόνη.

Εκτός από το μοντέλο του μικροελεγκτή μπορεί κανείς να επιλέξει και να αρχικοποιήσει τον παραγόμενο κώδικα με πολλές επιλογές. Μπορεί να ορίσει τη συχνότητα του ρολογιού που θα χρησιμοποιηθεί αλλά και να ορίσει τον τρόπο λειτουργίας των θυρών, των διακοπών και των μετρητών χρονιστών, όπως και άλλων περιφερειακών συσκευών που ενδέχεται να περικλείει το μοντέλο προς χρήση.



Σχήμα 28. Παράθυρο επιλογών του αυτόματου οδηγού.

Εάν προ-υπάρχουν αρχεία προς χρήση, μπορούν να αποθηκευτούν σε αυτό το στάδιο. Αυτό το στάδιο είναι προαιρετικό επειδή δεν έχουμε δημιουργήσει ακόμη αρχεία κώδικα (σχήμα 29).



Σχήμα 29. επιλογή αρχείων εφαρμογής

Μέσα από την επιλογή project|settings μπορούμε να προσθέσουμε εργαλεία τρίτων κατασκευαστών ώστε να συνεργάζονται με το CodeVision. Ένα από τα εργαλεία που προτείνεται στον αρχάριο χρήστη και όχι μόνο, είναι το AVR Studio, μια εφαρμογή που διανέμεται δωρεάν από την κατασκευάστρια εταιρία Atmel. Η εφαρμογή αυτή είναι ένα ολοκληρωμένο σύστημα ανάπτυξης εφαρμογών για τους μικροελεγκτές AVR και παρέχει εκτός από έναν κορυφαίο assembler για τη γλώσσα μηχανής των μικροελεγκτών, και ένα πολύ καλό περιβάλλον αποσφαλμάτωσης.

Το τελευταίο συνεργάζεται με τον C compiler του CodeVision και παρέχει πολλές δυνατότητες απεικόνισης του τρόπου λειτουργίας του μικροελεγκτή, των καταχωρητών του, της μνήμης δεδομένων και προγράμματος, τις θύρες εισόδου εξόδου κ.α.

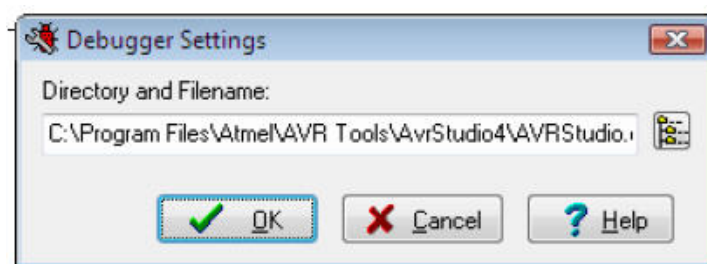
Εξομοίωση του προγράμματος

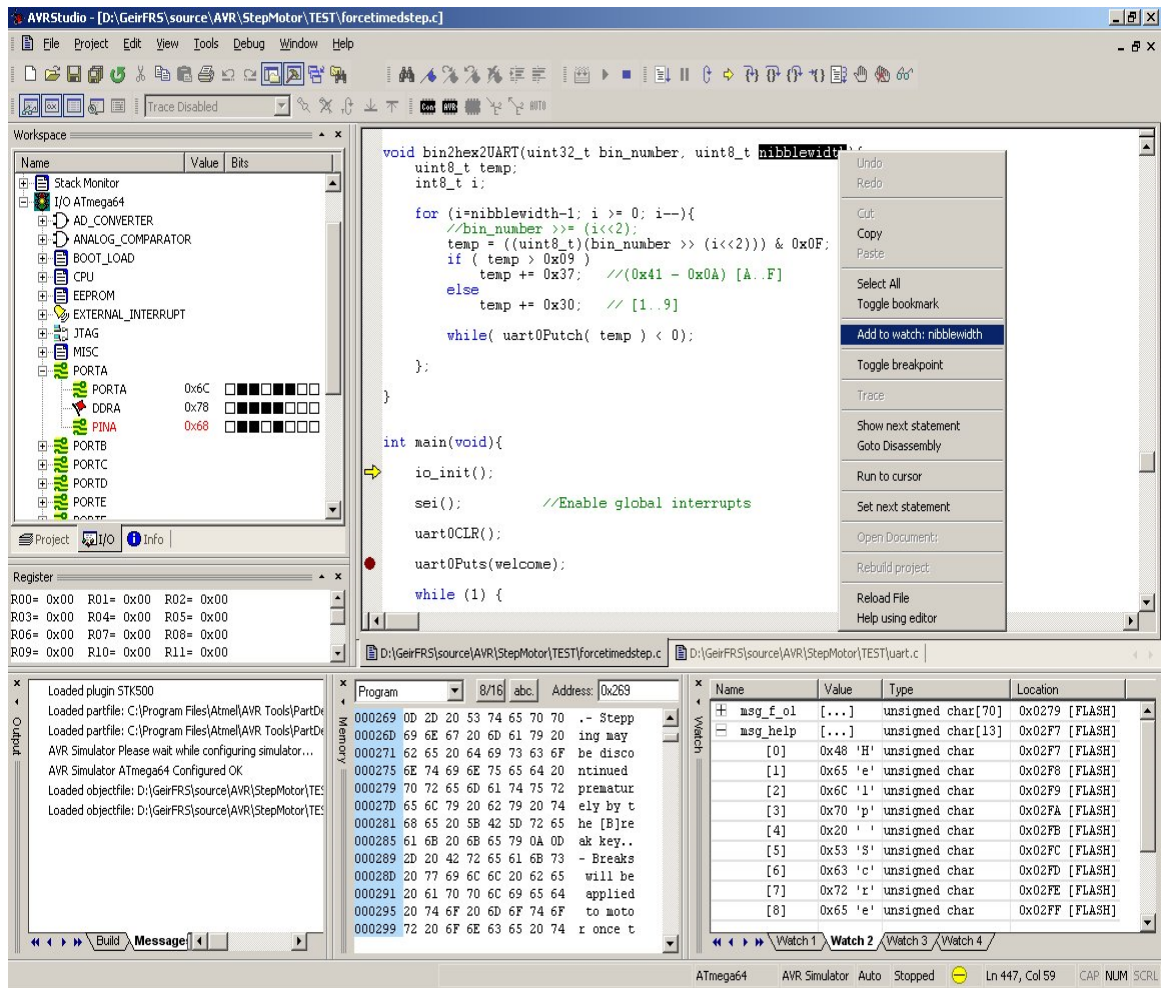
Η εξομοίωση χρησιμεύει στο να βρίσκονται τα σφάλματα του κώδικα και να διορθώνονται πριν τον προγραμματισμό του μικροεπεξεργαστή. Η εξομοίωση βήμα - βήμα είναι ο καλλίτερος τρόπος για να βρείτε και να διορθώσετε τα

λάθη του κώδικα που μόλις ανέπτυξε ένας μηχανικός λογισμικού. Θα μπορούσε κάποιος να σκεφτεί πως αν είναι έτοιμο το κύκλωμα μπορεί να προγραμματίζει τον AVR και να τον δοκιμάζει απευθείας.

Όμως αν υποθέσουμε πως ο μικροελεγκτής με χρονισμό 1MHz εκτελεί κάθε εντολή ανά 1 μ Sec, είναι κατανοητό ότι η συσκευή μπορεί να "κολλάει" ή να κάνει λάθη που δεν υπάρχει περίπτωση ποτέ να τα αντιληφθεί κάποιος με απλή παρατήρηση. Συνεπώς η εξομοίωση ό βήμα – βήμα με μια ροή εκτέλεσης ελεγχόμενη από τον χρήστη, δίνει την δυνατότητα ελέγχου, του τι συμβαίνει όταν εκτελείται κάθε εντολή.

Μέσω του λογισμικού AVR Studio (σχήμα 30), και αφού πρώτα γίνει η μεταγλώττιση με επιτυχία, δηλαδή το πρόγραμμα δεν περιέχει ορθογραφικά λάθη, μπορεί κάποιος να διαλέξει να παρατηρήσει μεταξύ όλων των περιφερειακών συσκευών, μνήμης δεδομένων ή προγράμματος, καταχωρητών ή της αριθμητικής και λογικής μονάδας. Αυτό επιτυγχάνεται με την προσθήκη ενός 'παρατηρητή' (Watch). Όλα τα βασικά παράθυρα που χρειάζονται για την εξομοίωση υπάρχουν στο χώρο εργασίας του λογισμικού.





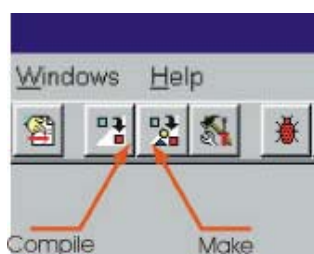
Σχήμα 30. το περιβάλλον AVR Studio

Τα breakpoints, είναι σημεία μέσα στον κώδικα που ο χρήστης δηλώνει την προσωρινή παύση εκτέλεσης του κώδικα. Η μέθοδος αυτή είναι η πλέον χρησιμοποιούμενη από τους μηχανικούς λογισμικού γιατί αποφεύγει τα χρονοβόρα βήματα, ιδιαίτερα αν ο κώδικας είναι πολύ μεγάλος ή χρησιμοποιούνται εκτεταμένα συναρτήσεις ή υποπρογράμματα. Βέβαια η μέθοδος αυτή συνεργάζεται πάντα με την μέθοδο 'βήμα – βήμα'.

Με την επιλογή Programmer → Select Programmer, επιλέγουμε τον προγραμματιστή του υλικού μέσω της καρτέλας επιλογών που εμφανίζεται στο περιβάλλον εργασίας. Στο αναπτυξιακό που επιλέξαμε για την ανάπτυξη των εφαρμογών ο προγραμματιστής πρέπει να είναι ο STK200. Η καρτέλα αυτή φαίνεται στο σχήμα 31 που ακολουθεί. Ο προγραμματιστής ‘κατεβάζει’ τον κώδικα της εφαρμογής στο chip και μόλις αυτό τεθεί υπο τάση λειτουργίας, αρχίζει αυτόματα η εκτέλεση του κώδικα. Για την εκτέλεση του κώδικα είναι απαραίτητο βέβαια να υπάρχει και κάποιο σήμα χρονισμού.

Ένα αρχείο πηγαίου κώδικα υψηλού επιπέδου γραμμένο σε κάποια γλώσσα μεταγλωττίζεται σε ένα αντίστοιχο αρχείο συμβολικού κώδικα (assembly) από τον κατάλληλο μεταγλωττιστή. Το προκύπτον αρχείο στη συνέχεια μετασχηματίζεται σε αντικειμενικό αρχείο γλώσσας μηχανής από έναν συμβολομεταφραστή (assembler), μία διαδικασία που μετατρέπει τον κώδικα σε άμεσα εκτελέσιμο από τον επεξεργαστή.

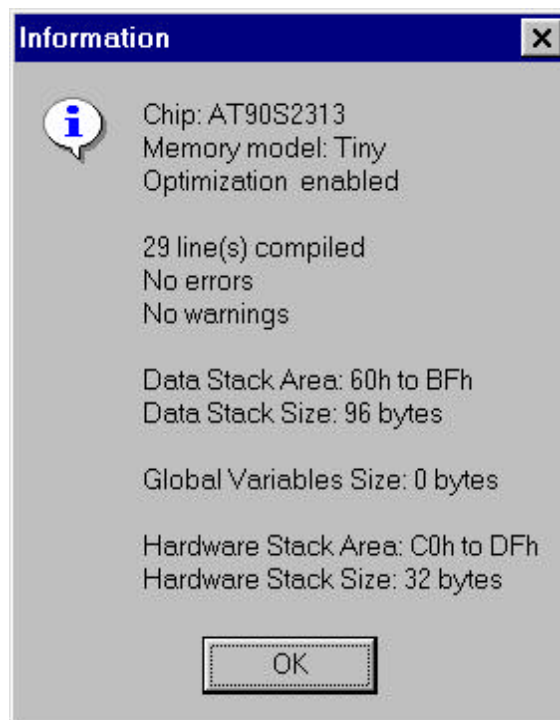
Η συμβολική γλώσσα και η γλώσσα μηχανής είναι ίδιου επιπέδου και υπάρχει αμφιμονοσήμαντη αντιστοιχία μεταξύ τους, απλώς η συμβολική αποτελείται από χαρακτήρες του λατινικού αλφαβήτου και δεκαεξαδικά ψηφία, ώστε να είναι κάπως κατανοητή από τον άνθρωπο, ενώ η γλώσσα μηχανής συντίθεται αποκλειστικά από bit. Έτσι τελικά σε κάθε αρχείο πηγαίου κώδικα, είτε αυτό είναι αυτόνομο (π.χ. main.c) είτε περιέχει περισσότερα αρχεία ή βιβλιοθήκες, αντιστοιχίζεται ένα αντικειμενικό αρχείο με κώδικα μηχανής. Για τη μεταγλώττιση και το χτίσιμο του προγράμματος χρησιμοποιούνται τα εικονίδια όπως απεικονίζονται στην εικόνα 32.



Σχήμα 32. κουμπιά επιλογής compile και make

Για να γίνει έλεγχος της καλής εκτέλεσης του προγράμματος στο περιβάλλον AVR Studio, πρέπει να χρησιμοποιήσουμε το εργαλείο του εξομοιωτή. Για να καταφέρει ο μεταγλωττιστής να αποδώσει την λειτουργία του παραπάνω προγράμματος πρέπει να είναι ενεργοποιημένος ο εξομοιωτής πριν το χτίσιμο του προγράμματος.

Αυτό είναι απαραίτητο γιατί ο μεταγλωττιστής παράγει επιπλέον πληροφορία (debugging information) που απαιτείται από το εργαλείο του εξομοιωτή. Αν είναι διαθέσιμος ένας εξομοιωτής υλικού, αυτή η ενέργεια δεν είναι απαραίτητη γιατί όλη η πληροφορία εξάγεται κατ' ευθείαν από τον μικροελεγκτή ενώ αυτός εκτελεί το δεδομένο πρόγραμμα. Μέσω της επιλογής Project → Build All ή μέσω του αντίστοιχου εικονιδίου, χτίζουμε το πρόγραμμα. Αν όλα πάνε καλά τότε εμφανίζεται ένα παράθυρο όπως στην εικόνα 33.



Σχήμα 33. παράθυρο μεταγλώττισης

Εφαρμογή κίνησης με αποφυγή εμποδίων

Για την επίδειξη της αυτοκινούμενης ρομποτικής πλατφόρμας επιλέχθηκε μια απλή εφαρμογή αποφυγής εμποδίων. Αυτή συνίσταται στην άμεση αποφυγή του αντικειμένου που θα βρεθεί εντός των ορίων των αισθητήρων υπέρυθρων. Ο τρόπος αποφυγής είναι άμεσος με στροφή της πλατφόρμας προς την αντίθετη κατεύθυνση από εκείνη που ανιχνεύθηκε το αντικείμενο.

Ο μικροελεγκτής του συστήματος χρησιμοποιεί δύο εξωτερικές διακοπές για να ανιχνεύσει τα σήματα από τους υπέρυθρους δέκτες TSOP1736. Ένα ζευγάρι τυπικών διόδων LED υπέρυθρου φάσματος στέλνουν συνεχώς παλμούς με συχνότητα 36KHz που είναι και η συχνότητα μέγιστης απόκρισης των δεκτών. Η υπέρυθη ακτινοβολία δεν είναι ορατή δια γυμνού οφθαλμού όμως για λόγους αποσφαλμάτωσης μπορεί να χρησιμοποιηθεί μια κάμερα με αισθητήρα CMOS ή CCD, παραδείγματος χάριν μια webcam. Τα IR-LED οδηγούνται μέσω του χρονιστή/μετρητή TCNT0 του μικροελεγκτή ο οποίος και ρυθμίζει το σήμα διαμόρφωσης πλάτους παλμών έτσι ώστε οι σερβο-κινητήρες να κινούνται με την επιθυμητή φορά.

Ταυτόχρονα όταν ενεργοποιείται κάποιος δέκτης ένα κόκκινο LED φωτοβολεί σαν ένδειξη ότι ένα αντικείμενο ανιχνεύτηκε. Δύο τέτοια LED είναι συνδεδεμένα με το αναπτυσιακό, ένα για κάθε δέκτη υπέρυθρων. Οι πομποί υπέρυθρης ακτινοβολίας βρίσκονται αριστερά και δεξιά του μπροστινού μέρους της πλατφόρμας και είναι τοποθετημένοι με τέτοιο τρόπο ώστε οι δέκτες να μην δέχονται την ακτινοβολία άμεσα παρα μόνο μέσω ανάκλασης με το αντικείμενο που βρίσκεται μπροστά τους. Θα πρέπει να αναφερθεί ότι το σύστημα αυτό δεν είναι δυνατόν να μετρήσει αποστάσεις. Το ποσό της ακτινοβολίας που επιστρέφει από την επιφάνεια ανάκλασης δεν είναι το ίδιο για όλα τα αντικείμενα.

Οι σερβο-κινητήρες κινούν τους ρότορες τους προς μία από τις δύο φορές περιστροφής, πάντα με την ίδια ταχύτητα. Η τροποποίηση που υπέστησαν τα σέρβο δεν επιτρέπουν τον έλεγχο της ταχύτητας παρα μόνο τη φορά περιστροφής.

Παράρτημα

```
#include <mega16.h>
```

```
#include <delay.h>
```

```
unsigned int ovf_counter;
```

```
boolean left_sensor;
```

```
boolean right_sensor;
```

```
Interrupt [TIM_OVF] void timer0_ovf_isr(void)
```

```
{
```

```
    ovf_counter++;
```

```
    if (left_sensor)
```

```
    {
```

```
        If (ovf_counter == 1)
```

```
        PORTB.3 = 1
```

```
        If (ovf_counter == 500)
```

```
        {
```

```
            PORTB.3 = 0;
```

```
            ovf_counter =0;
```

```
        }
```

```
}  
  
if(right_sensor)  
{  
  
    If (ovf_counter == 2)  
  
        PORTB.4 = 1  
  
    If (ovf_counter == 500)  
  
        {  
  
            PORTB.4 = 0;  
  
            ovf_counter =0;  
  
        }  
  
}  
  
else  
  
{  
  
    If (ovf_counter == 1)  
  
        PORTB.3 = 1;  
  
        PORTB.4 = 1;  
  
    If (ovf_counter == 500)  
  
        {  
  
            PORTB.3 = 0;  
  
            PORTB.4 = 0;
```

```

        ovf_counter =0;
    }
}

}

Interrupt [EXT_INT0] void ext_int0_isr(void)
{
    Left_sensor = true;

    PORTB.5 = 0;
}

Interrupt [EXT_INT1] void ext_int1_isr(void)
{
    Right_sensor = true;

    PORTB.6 = 0;
}

void main(void)
{
    DDRB = 0b00001111;

```

```
DDRC = 0b00000000;

PORTB = 0b11111111;

TCCR0 = 0x03;

TCNT0 = 0x06;

TIMSK = 0x01;

GICR = 0xC0;

MCUCR = 0x0F;

#asm("sei")

while(1)

{

    PORTB = 0b00000011;

    delay_us(17);

    PORTB = 0b00000000;

    delay_us(17);

}
```

Βιβλιογραφία

1. Ζ. Δουλγέρη, ΡΟΜΠΟΤΙΚΗ κινηματική, δυναμική και έλεγχος αρθρωτών βραχιόνων, εκδόσεις Κριτική, Θεσσαλονίκη, 2007
2. Συστήματα Μικροϋπολογιστών, Κ.Ζ. Πεκμεστζή
3. Εργαστήριο Μικροϋπολογιστών, Κ.Ζ. Πεκμεστζή
4. Ψηφιακή Σχεδίαση, Μ. Morris Mano
5. The Architecture of Small Computer Systems, Α.Γ. Lippiatt, G.G.L. Wright
6. Η γλώσσα προγραμματισμού C, Β.Β. Kernighan D.M. Ritchie
7. Μικροηλεκτρονικά Κυκλώματα, Sedra, Smith
8. Sensors – Electronic Components, Siemens
9. Τζαφέστας, Σπύρος Γ., «Ρομποτική. Τομ. 1: Ανάλυση και έλεγχος.» (629.892 TZA).
10. Yoshikawa, Tsuneo, "Foundations of robotics : analysis and control," The MIT Press, 1990. (629.892 YOS)
11. Mason, Matthew, "Mechanics of Robotic Manipulation," MIT Press, 2001.
12. Meystel, A., "Autonomous mobile robots : vehicles with cognitive control," World Scientific, 1991. (629.892 MEY)
13. Borenstein, Johann, "Navigating mobile robots : systems and techniques," Wellesley, MA.: : AK Peters, Ltd., 1996. (629.892)
14. Sheridan, Thomas B., "Telerobotics, automation, and human supervisory control," The MIT Press, 1992. (620.46 SHE)
15. Dudek, G., Jenkin, M., "Computational Principles of Mobile Robotics", Cambridge University Press, 2000.
16. Siegwart, R., Nourbakhsh, I. R., "Introduction to Autonomous Mobile Robots", The MIT Press, Cambridge, MA, 2004
17. Montemerlo, M., and Thrun, S., "FastSLAM - A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics", Springer Tracts in Advanced Robotics, 2007.
18. Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E., and Thrun, S., "Principles of Robot Motion - Theory, Algorithms and Implementations", The MIT Press, Cambridge, Massachusetts, 2005
19. Takashi Kenjo, "Stepping Motors And Their Microprocessor Controls", Oxford Science Publications, 1992.
20. Ηλεκτρικές Μηχανές DC - AC, Stephen J. Chapman ; μετάφραση Νίκος Μάργαρης, Νίκος Χατζηγιάννου. - Εκδόσεις Τζιόλα, 1993
21. Electrical Machines, Drives, and Power Systems, T. Wildi, Prentice Hall; 1999
22. Βασική Ηλεκτρολογία, Ν.Κολλιόπουλου, Εκδόσεις Ίων, Αθήνα 2001

23. Εμίρης, Δ. Μ. και Κουλουριώτης, Δ. Ε., Ρομποτική, 2η έκδοση, Τεκδοτική, Αθήνα 2004.
24. Τζαφέστας, Σ., Ρομποτική: Ανάλυση και Έλεγχος, ΕΜΠ, Αθήνα 2003.