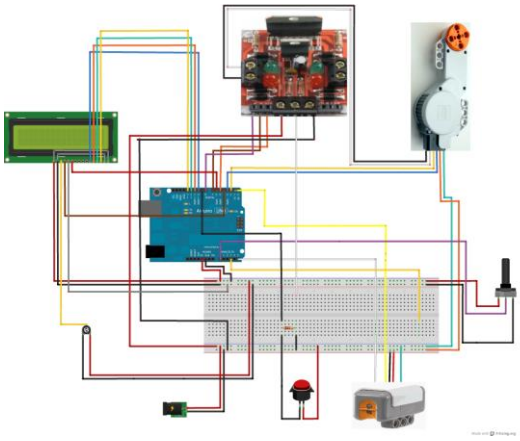
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανολόγων Μηχανικών Τ.Ε

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επέκταση των δυνατοτήτων του εκπαιδευτικού πακέτου «LEGO-MINDSTORMS», με την χρήση του μικροελεγκτή Arduino, για μηχατρονικές εφαρμογές



Σπουδαστής
Χαρούπιας Α. Ανδρέας Α.Μ. 4290

Επιβλέπων
Δρ. Φασουλάς Ι. Ιωάννης
Επίκουρος Καθηγητής σχολής ΣΤΕΦ
τμήματος Μηχανολόγων Μηχανικών Τ.Ε. Τ.Ε.Ι. Κρήτης

Ηράκλειο 2013

η σελίδα αυτή είναι σκόπιμα λευκή

Περιεχόμενα

Περίληψη και σκοπός της εργασίας 7

Κεφάλαιο 1 - Η υπολογιστική πλατφόρμα Arduino 8

1.1 Εισαγωγή 8
1.2 Ιστορικά στοιχεία 8
1.3 Αναφορά στις υπολογιστικές πλατφόρμες Arduino 9
1.4 Εφαρμογές με την υπολογιστική πλατφόρμα Arduino 9
1.4.1 Arduino Laser Harp 10
1.4.2 Unmanned Aerial Vehicle (UAV) Spyplane 10
1.4.3 The Soundmachine 11
1.4.4 Hexacopter and Hexapod 11
1.5 Περιγραφή της υπολογιστικής πλατφόρμας Arduino 12
1.6 Διαφορετικά είδη των υπολογιστικών πλατφορμών Arduino 14
1.6.1 Υπολογιστική πλατφόρμα Arduino Mini 14
1.6.2 Υπολογιστική πλατφόρμα Arduino Pro Mini 14
1.6.3 Υπολογιστική πλατφόρμα Arduino Micro 14
1.6.4 Υπολογιστική πλατφόρμα Arduino Nano 15
1.6.5 Υπολογιστική πλατφόρμα Arduino Frio 15
1.6.6 Υπολογιστική πλατφόρμα Arduino Pro 15
1.6.7 Υπολογιστική πλατφόρμα Arduino Leonardo 16
1.6.8 Υπολογιστική πλατφόρμα Arduino Due 16
1.6.9 Υπολογιστική πλατφόρμα Arduino Mega 2560 16
1.6.10 Υπολογιστική πλατφόρμα Arduino ADK 17
1.6.11 Υπολογιστική πλατφόρμα Arduino Ethernet 17
1.6.12 Υπολογιστική πλατφόρμα Arduino BT 17

1.6.13 Υπολογιστική πλατφόρμα Arduino Lilypad Usb 18
1.6.14 Υπολογιστική πλατφόρμα Arduino Esplora 18
1.7 Μονάδες επέκτασης υπολογιστικής πλατφόρμας Arduino (Arduino Shields) 19
1.8 Αναλυτική περιγραφή της υπολογιστικής πλατφόρμας Arduino Uno 19
1.8.1 Θύρες εισόδου και θύρες εξόδου της υπολογιστικής πλατφόρμας Arduino Uno 20
1.8.2 Υπόλοιπα μέρη της υπολογιστικής πλατφόρμας Arduino Uno 22
1.8.3 Μνήμες που χρησιμοποιούνται από όλες τις υπολογιστικές πλατφόρμες Arduino 23
1.8.3.1 Μνήμη «Flash memory» 23
1.8.3.2 Μνήμη «SRAM memory» 24
1.8.3.3 Μνήμη «EEPROM memory» 24
1.8.4 Φυσικά χαρακτηριστικά της υπολογιστικής πλατφόρμας Arduino Uno 24
1.9 Το περιβάλλον προγραμματισμού Arduino IDE 25
1.10 Αναφορά στην «γλώσσα» της υπολογιστικής πλατφόρμας Arduino 26
1.10.1 Βασική δομή του κώδικα 27
1.10.2 Δομές ελέγχου (Control Structures) 27
1.10.3 Μεταβλητές της υπολογιστικής πλατφόρμας Arduino 27
1.10.4 Βασικές συναρτήσεις της υπολογιστικής πλατφόρμας Arduino 28
1.11 Το παράδειγμα «BLINK» 28
1.12 Οι Βιβλιοθήκες της υπολογιστικής πλατφόρμας Arduino 29
1.12.1 Βιβλιοθήκη της οθόνης LCD 29
1.12.1.1 Παράδειγμα χρήσης της βιβλιοθήκης 30
1.12.2 Βιβλιοθήκη των κινητήρων τύπου «RC-SERVO» 31
1.12.3 Βιβλιοθήκη των κινητήρων τύπου «STEPPER» 31
1.12.4 Βιβλιοθήκη του ελεκτή PID 32
1.13 Διαμόρφωση Εύρους Παλμών (P.W.M) 33

Κεφάλαιο 2 - Το εκπαιδευτικό πακέτο Lego Mindstorms NXT 36

2.1 Εισαγωγή 36
2.2 Ιστορική αναδρομή 36
2.3 Μέρη του εκπαιδευτικού πακέτου Lego Mindstorms NXT 37
2.3.1 Το NXT Brick 38

2.3.2 Οι κινητήρες που χρησιμοποιεί το εκπαιδευτικό πακέτο Lego Mindstorms NXT 2.0 39
2.3.3 Αισθητήρες που χρησιμοποιεί το εκπαιδευτικό πακέτο Lego Mindstorms NXT 2.0 39
2.3.3.1 Αισθητήρας αφής 40
2.3.3.2 Αισθητήρας ήχου 40
2.3.3.3 Αισθητήρας φωτός 41
2.3.3.4 Αισθητήρας υπερήχων 41
2.3.4 Λοιπά μέρη του εκπαιδευτικού πακέτου 42

Κεφάλαιο 3 - Κατασκευή εκπαιδευτικής διάταξης με χρήση του κινητήρα «NXT» και της υπολογιστικής πλατφόρμας Arduino 43

3.1 Εισαγωγή 43
3.2 Παρουσίαση της εκπαιδευτικής διάταξης 43
3.3 Βάσεις στήριξης 45
3.4 Ο κινητήρας «NXT» της Lego 46
3.5.1 Σύνδεση της «Η-Γέφυρας»(H-Bridge) 50
3.5.2 Παράδειγμα χρήσης της «Η-Γέφυρας» σε κώδικα 51
3.6 Σύνδεση του αναλογικού περιστροφικού ποτενσιόμετρου 53
3.7 Σύνδεση του μπουτόν 53
3.8 Σύνδεση της οθόνης LCD 54
3.9 Σύνδεση του αισθητήρα φωτός 55
3.10 Σύνδεση της πλατφόρμας «Arduino Uno» και τα εξωτερικά «Interrupt»(«Interrupt Service Routine») 58
3.11 Παράδειγμα χρήσης των εξωτερικών «Interrupt»(«Interrupt Service Routine») 59
3.12 Ηλεκτρολογική σύνδεση των παραπάνω εξαρτημάτων 59
3.13 Ο ελεκτής «PID» και ο έλεγχος κλειστού βρόγχου με την χρήση του ελεκτή 60
3.14 Λειτουργία του «PID» 61
3.15 Η μέθοδος συντονισμού του ελεκτή PID «Ziegler-Nichols» («Ziegler-Nichols» tuning method) 63
3.16 Υπολογισμός του ελεκτή «PID» με τη μέθοδο συντονισμού «Ziegler and Nichols» («Ziegler-Nichols» tuning method) 64
3.17 Πειραματική διαδικασία με τα κέρδη του «PID» ελεκτή 65

3.17.1 Πειραματική διαδικασία 1 70
3.17.2 Πειραματική διαδικασία 2 71
3.17.3 Πειραματική διαδικασία 3 72
3.17.4 Πειραματική διαδικασία 4 74
3.17.5 Πειραματική διαδικασία 5 74
3.17.6 Πειραματική διαδικασία 6 75

Κεφάλαιο 4 - Κατασκευή εκπαιδευτικής διάταξης με χρήση των κινητήρων τύπου «Servo» - «Stepper» 77

4.1 Εισαγωγή 77
4.1.1 Οι κινητήρες τύπου «RC SERVO» 77
4.1.2 Συνδεσμολογία του κινητήρα τύπου «RC SERVO» 78
4.1.3 Βασική δομή ενός σερβοκινητήρα 78
4.1.4 Έλεγχος του κινητήρα τύπου «RC SERVO» 79
4.2 Κατασκευή της θήκης του «RC Servo» 79
4.3 Σύνδεση της υπολογιστικής πλατφόρμας Arduino Uno με τον κινητήρα τύπου «Stepper» 80
4.4 Ο κώδικας για την λειτουργία όλων των παρελκόμενων της κατασκευής με τον κινητήρα τύπου «RC Servo» 81
4.5.1 Βηματικοί κινητήρες («Stepper motors») 83
4.5.2 Βασική δομή ενός βηματικού κινητήρα 83
4.5.3 Έλεγχος ενός βηματικού κινητήρα 84
4.6 Το ολοκληρωμένο κύκλωμα «ULN2803»(«Integrated circuit») 84
4.7.1 Μονοπολικοί («Unipolar») βηματικοί κινητήρες 85
4.7.2 Διπολικοί («Bipolar») βηματικοί κινητήρες 86
4.8 Σύνδεση της υπολογιστικής πλατφόρμας Arduino Uno με τον κινητήρα τύπου «Stepper» 87
4.9 Παράδειγμα κώδικα για την λειτουργία ενός βηματικού κινητήρα 87
4.10 Ο κώδικας για την λειτουργία όλων των παρελκόμενων της διάταξης με τον κινητήρα τύπου «Stepper» 89

Κεφάλαιο 5 - Συμπεράσματα και μελλοντική επέκταση 91

Βιβλιογραφία 92

Περίληψη και σκοπός της εργασίας

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι η επέκταση των δυνατοτήτων του εκπαιδευτικού πακέτου **«Lego Mindstorms»** με την χρήση της υπολογιστικής πλατφόρμας **«Arduino»** και η δημιουργία εκπαιδευτικών εφαρμογών που θα αναδεικνύουν τα πλεονεκτήματα και τα μειονεκτήματα στη χρήση τους σε ρομποτικές και γενικά μηχατρονικές εφαρμογές. Εμβαθύνοντας στη μελέτη του εκπαιδευτικού πακέτου **«Lego Mindstorms»** και της υπολογιστικής πλατφόρμας **«Arduino»**, κατασκευάσαμε τρεις διαφορετικές εκπαιδευτικές διατάξεις, στις οποίες εξετάζεται ο έλεγχος μίας περιστροφικής άρθρωσης ενός βαθμού ελευθερίας με την βοήθεια μικροελεγκτή. Η διαφοροποίηση μεταξύ των διατάξεων έγινε για να χρησιμοποιηθούν τρείς διαφορετικοί ηλεκτρικοί επενεργητές (κινητήρες) για την κίνηση και τον έλεγχο της άρθρωσης, ώστε να αντιληφθεί ο χρήστης τις διαφορές στην λειτουργία, τον έλεγχο και τον προγραμματισμό μεταξύ α) του ηλεκτρικού κινητήρα **«NXT» του εκπαιδευτικού πακέτου «Lego Mindstorms»** β) του ηλεκτρικού κινητήρα τύπου **«RC-Servo** και γ) του ηλεκτρικού κινητήρα τύπου «**Stepper»**. Η υπολογιστική πλατφόρμα **«Arduino»** αποτέλεσε τον βασικό μικροελεγκτή για τον έλεγχο των διατάξεων. Τέλος με τις τρείς αυτές διατάξεις θα μπορεί ο κάθε σπουδαστής να έρθει σε επαφή, να εκπαιδευτεί πάνω τους, και μελλοντικά ακόμα και να επεκτείνει τις δυνατότητες τους. Ποιο συγκεκριμένα, **στο Κεφάλαιο 1** θα ασχοληθούμε με τα είδη των υπολογιστικών πλατφόρμων **«Arduino»** τις λειτουργίες τους, τις δυνατότητές τους και τον προγραμματισμό τους. **Στο κεφάλαιο 2** παρουσιάζεται το εκπαιδευτικό πακέτο **«Lego Mindstorms»** με όλα τα μέρη του και την λειτουργία τους. **Στο κεφάλαιο 3** γίνεται εκτενείς αναφορά στη πρώτη εκπαιδευτική διάταξη, που αποτελείται από μέρη του εκπαιδευτικού πακέτου **«Lego Mindstorms**» και της υπολογιστικής πλατφόρμας **«Arduino»**, όπως επίσης και ο έλεγχος της μέσω του ελεκτή **«PID»**. Τέλος **στο Κεφάλαιο 4** παρουσιάζεται η εκπαιδευτική διάταξη με τη χρήση α) του ηλεκτρικού κινητήρα τύπου **«RC-Servo** και β) του ηλεκτρικού κινητήρα τύπου «**Stepper»**, αναλύοντας την λειτουργία του κάθε επενεργητή (κινητήρα) ξεχωριστά και παραθέτοντας τα συμπεράσματα μας κλείνουμε το τελευτάιο κεφάλαιο της εργασίας.

Κεφάλαιο 1

Η υπολογιστική πλατφόρμα Arduino

1.1 Εισαγωγή

Στο κεφάλαιο αυτό θα γίνει αναλυτική αναφορά στην **υπολογιστική πλατφόρμα Arduino**, τα χαρακτηριστικά , τις δυνατότητες και τις λειτουργίες της. Μέσω τις βραχείας τεχνολογικής εξέλιξης οι πλατφόρμες Arduino αποτελούν ένα πανίσχυρο εργαλείο για απλές και πολύπλοκες εφαρμογές σε διαφορετικά πεδία της ανθρώπινης εξέλιξης, για αυτό θα προσπαθήσουμε να αναλύσουμε όσο πιο απλά γίνεται τα πλεονεκτήματα και τα μειονεκτήματα τους έχοντας ως γνώμονα τα θετικά ενός «ανοιχτού τύπου» εγχειρήματος όπως είναι το εγχείρημα του Arduino.

1.2 Ιστορικά στοιχεία

Το 2005, στην Ivrea της Ιταλίας (στην ίδια περιοχή με την εταιρεία υπολογιστών Olivetti), ένα έργο άρχισε να δημιουργείτε, μια συσκευή για τον έλεγχο σχεδίων, χτισμένο από μαθητές με λιγότερα έξοδα από ό, τι με άλλα πρωτότυπα συστήματα που ήταν διαθέσιμα εκείνην τη περίοδο, Massimo Banzi και David Cuartielles, ονόμασαν το έργο τους Arduin of Ivrea, «Arduino» είναι επίσης ιταλικό όνομα, που μεταφράζεται ελεύθερα ως «γενναίος φίλος» και έτσι σιγά-σιγά ξεκίνησαν την παραγωγή του σ ένα μικρό εργοστάσιο της περιοχής . Το Arduino είναι μια παραγόμενη έκδοση της πλατφόρμας ανοικτού κώδικα

«Wiring Platform». Ο Κολομβιανός καλλιτέχνης και προγραμματιστής Hernando Barragán δημιούργησε τη συρμάτωση «Wiring» ως μια Μεταπτυχιακή διπλωματική εργασία στο Interaction Design Institute Ivrea υπό την εποπτεία του Massimo Banzi και του Casey Reas. Η Καλωδίωση «Wiring» βασίστηκε στην επεξεργασία και το ολοκληρωμένο περιβάλλον ανάπτυξης που είχε δημιουργηθεί από τον Casey Reas και τον Ben Fry [2].

Εικόνα 1.2.1 Το «Logo» της εταιρείας

1.3 Αναφορά στην υπολογιστική πλατφόρμα Arduino

Όπως το περιγράφει ο δημιουργός του, το Arduino είναι μία open-source (ανοικτού κώδικα) πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών κυκλωμάτων βασισμένη σε ευέλικτο και εύκολο στη χρήση «hardware» και «software» που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.

Εικόνα 1.2.2 Η πρώτη υπολογιστική πλατφόρμα Arduino τέλη του 2005

Το Arduino αποτελείται από δύο κύρια μέρη ,την υπολογιστική πλατφόρμα Arduino, η οποία είναι το κομμάτι του hardware πάνω στο οποίο εργάζεται ο κατασκευαστής όταν πραγματοποιεί μία κατασκευή, ενώ το δεύτερο τμήμα είναι το Arduino IDE, το κομμάτι του λογισμικού που τρέχει στον υπολογιστή. Το IDE χρησιμοποιείται για να δημιουργηθεί ένα «sketch» (ένα μικρό πρόγραμμα στον υπολογιστή) που φορτώνεται στον μικροελεγκτή της υπολογιστικής πλατφόρμας Arduino [2].

1.4 Εφαρμογές με την υπολογιστική πλατφόρμα Arduino

Πριν αναφερθούμε αναλυτικά στις **υπολογιστικές πλατφόρμες Arduino**, τις δυνατότητες και τα πλεονεκτήματα τους, σκόπιμο είναι να παρουσιάσουμε τέσσερα από τα πιο σημαντικά και εντυπωσιακά project που έχουν δημιουργηθεί ώστε να αντιληφθούμε πόσο ισχυρές είναι οι πλατφόρμες αυτές.

1.4.1 Άρπα με την χρήση λέιζερ («Arduino Laser Harp»)

Ίσως ένα από τα πιο εντυπωσιακά «project» με τη χρήση του Arduino είναι η **«Laser Harp»** άρπα με την χρήση λέιζερ, που πρωτοχρησιμοποίηθηκε από τον μουσικό Jean Michel Jarre στης συναυλίες του. Ουσιαστικά η συσκευή παράγει μία σειρά κάθετων φωτεινών γραμμών («laser») που ξεκινούν από το επίπεδο και διακόπτοντας τις δέσμες φωτός παράγεται μία ποικιλία μουσικών ήχων, φυσικά ρόλο για την παραγωγή του ήχου δε παίζει μόνο η διακοπή της δέσμης αλλά και το ύψος που θα τοποθετηθεί το εμπόδιο από το επίπεδο. Η παραγωγή ήχου γίνεται μέσω ενός synthesizer ώστε να λαμβάνει από το Arduino σειριακά τα MIDI («Musical Instrument Digital Interface») δεδομένα που παράγει [4].

Εικόνα 1.4.1 Arduino Laser Harp

1.4.2 Μη επανδρωμένο κατασκοπευτικό αεροσκάφος («Unmanned Aerial Vehicle Spyplane»)

Σε αυτό το «project» το εντυπωσιακό είναι ότι όπως λέει και το όνομα του για την πλοήγηση του δεν χρειάζεται η ανθρώπινη παρουσία μόνο ο προγραμματισμός του, αφού απαιτείται ένας μικροελεγκτής που θα είναι υπεύθυνος για τον έλεγχο του οχήματος (Arduino) και ένα σύστημα για την ασύρματη επικοινωνία του Arduino με την συσκευή που θα το τηλεκατευθύνει (όπως για παράδειγμα η shield x-bee). Η δουλειά του μικροελεγκτή είναι να ελέγχει τους απαραίτητους κινητήρες και σερβοκινητήρες για την κατεύθυνση του σκάφους. Η τηλεκατεύθυνση μπορεί να προσαρμοστεί σε οποιαδήποτε μορφή σκάφους-οχήματος (αυτοκίνητο, πλοίο, αεροπλάνο) και με την ενσωμάτωση του Arduino GPS shield που ενημερώνει την υπολογιστική πλατφόρμα Arduino με τις συντεταγμένες του σημείου που βρίσκεται, το όχημα γνωρίζοντας τον προορισμό του βάση των συντεταγμένων που του δόθηκαν, θα μπορεί να κινηθεί προς αυτόν χωρίς να χρειάζεται κάποιος να το κατευθύνει [5].

Εικόνα 1.4.2 Arduino Spyplane

1.4.3 Ηλεκτρονικό ηχοσύστημα («The Soundmachine»)

Η εξωτερική εμφάνιση της κατασκευής μοιάζει αρκετά με ένα κλασικό πικάπ, η λειτουργία της όμως είναι εντελώς διαφορετική. Η βελόνα του πικάπ έχει αντικατασταθεί με ένα LED το οποίο φωτίζει την περιστρεφόμενη πλατφόρμα και έναν αισθητήρα φωτός που ανιχνεύει το αντανακλώμενο φως. Η κυκλική περιστρεφόμενη πλατφόρμα η οποία στην πραγματικότητα έχει αντικαταστήσει το βινύλιο αποτελείται από διάφορες χρωματιστές ακτίνες οι οποίες ανάλογα με το χρώμα που έχουν, αντανακλούν το φως στον αισθητήρα ο οποίος μετατρέπει το φως σε ηλεκτρικά σήματα και τα στέλνει στον Arduino. Ο μικροελεγκτής είναι υπεύθυνος για την μετατροπή των σημάτων στην κατάλληλη μορφή και στη συνέχεια για την αποστολή τους στον υπολογιστή όπου ανάλογα με το φως που αντανακλάται από την πλατφόρμα, ο υπολογιστής αναπαράγει τα ανάλογα μουσικά «loops».

Εικόνα 1.4.3 Arduino Soundmachine

1.4.4 Εξάποδο και εξάπτερο («Hexacopter and Hexapod»)

Το συγκεκριμένο «project» ουσιαστικά είναι η ένωση δυο διαφορετικών σε ένα αφού είναι ένα όχημα με 6 πόδια σε σχήμα αράχνης και ένα ελικόπτερο με 6 έλικες. Ο μικροελεγκτής είναι υπεύθυνος τόσο για την κίνηση όσο βρίσκεται στο έδαφος όσο και στον αέρα. Η κατασκευή έχει γίνει όλη με ανθρακονήματα για εμφανής λόγους και χρησιμοποιήθηκαν 6 κινητήρες για καλύτερη σταθερότητα κατά την διάρκεια της πτήσης [6].

Εικόνα 1.4.4 Arduino Hexacopter and Hexapod

1.5 Περιγραφή της υπολογιστικής πλατφόρμας Arduino

Συγκεκριμένα το **Arduino** είναι μια **υπολογιστική πλατφόρμα** βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή, εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί και να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων και προγραμμάτων. Είναι ουσιαστικά ένα εργαλείο για να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό Ηλεκτρονικό Υπολογιστή. Είναι ανοιχτού υλικού και λογισμικού και βασίζεται σε μια αναπτυξιακή, μιας υπολογιστικής πλατφόρμας, που ενσωματώνει επάνω έναν μικροελεγκτή και συνδέεται με τον Η/Υ για να τον προγραμματίσουμε μέσα από ένα απλό περιβάλλον ανάπτυξης. Ένα Arduino μπορεί να χρησιμοποιηθεί για να αναπτύξουμε διαδραστικά αντικείμενα, να δεχτούμε εισόδους από πληθώρα αισθητηρίων οργάνων και διακόπτες, αλλά και να ελέγχουμε διάφορα φώτα, κινητήρες και άλλες συσκευές εξόδου του φυσικού κόσμου. Ο **μικροελεγκτής** («microcontroller») είναι ένας τύπος επεξεργαστή, μια παραλλαγή ουσιαστικά του μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια),ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

Εικόνα 1.5.1 Χειροποίητη κατασκευή της υπολογιστικής πλατφόρμας Arduino

Η **υπολογιστική πλατφόρμα Arduino** αποτελείται από ένα μικροελεγκτή Atmel AVR και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό

και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλατφόρμες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα «bootloader» δηλαδή ένα εσωτερικό κώδικα , έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Το Arduino διατίθεται σε πολλές εκδόσεις ανάλογα με τις ανάγκες και τις δυνατότητες του κάθε προγραμματιστή γενικά όμως όλες οι πλακέτες είναι προγραμματισμένες μέσω μιας σειριακής σύνδεσης RS-232, αλλά ο τρόπος με τον οποίο αυτό υλοποιείται ποικίλλει ανάλογα με την έκδοση. Οι σειριακές πλακέτες Arduino περιέχουν ένα απλό κύκλωμα αντιστροφής για την μετατροπή ανάμεσα στα σήματα των επιπέδων «RS-232» και «TTL». Οι υπολογιστικές πλατφόρμες **Arduino** που κυκλοφορούν σήμερα στην αγορά, προγραμματίζονται μέσω USB, εφαρμόζοντας ένα τσίπ προσαρμογέα «usb-to-serial» όπως το «FTDI FT232». Όσον αφορά την αρχιτεκτονική του τώρα, το Arduino είναι μια πλατφόρμα βασισμένη στον ATMEGA328. Περιέχει 14 ψηφιακούς ακροδέκτες εισόδου-εξόδου(«I/O»), 6 αναλογικές εισόδους και 6 ψηφιακές εξόδους και έναν ταλαντωτή των 16 MHz. Η σύνδεση γίνεται με «usb» καλώδιο, ενώ υπάρχει και υποδοχή σύνδεσης με ρεύμα, καθώς και δυνατότητα εντός του κυκλώματος, σειριακού προγραμματισμού-ICSP («In-Circuit Serial Programming») και ένα κουμπί επανεκκίνησης (reset) σε περίπτωση που βραχυκυκλώσει η πλατφόρμα χωρίς την θέλησή μας .

Μια **υπολογιστική πλατφόρμα Arduino** εμπεριέχει όλα όσα χρειάζονται για την υποστήριξη ενός μικροελεγκτή. Το μόνο που απαιτείται είναι η σύνδεσή του με έναν ηλεκτρονικό υπολογιστή μέσω ενός καλωδίου usb ή η τροφοδοσία του μέσω ενός μετασχηματιστή τάσης για τη μετατροπή μεταξύ εναλλασσόμενου και ασυνεχούς ρεύματος («adapter AC-to-DC-Alternating Current-to-Direct Current») ή μιας μπαταρίας.

Εικόνα 1.5.2 Υπολογιστική πλατφόρμα Arduino Mega ADK

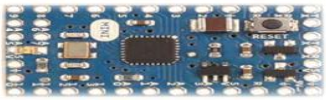
Τόσο η μεγάλη επιτυχία που έδειξε το πρώτο καιρό η **υπολογιστική πλατφόρμα Arduino** όσο η ανάγκη δημιουργίας διαφορετικών πλακετών ανάλογα με τις ανάγκες και δυνατότητες του κάθε χρήστη καθώς και η πληθώρα χρηστών που ασχολήθηκαν με την

ανάπτυξη της σε επίπεδο προγραμματισμού ώθησε τους κατασκευαστές του να δημιουργήσουν μια σειρά από διαφορετικές εκδόσεις της υπολογιστικής πλατφόρμας Arduino με διαφορετικές δυνατότητες και χαρακτηριστικά η κάθε μία αλλά έχοντας ως γνώμονα την αρχική ιδέα.

1.6 Διαφορετικά είδη της υπολογιστικής πλατφόρμας Arduino

1.6.1 Υπολογιστική πλατφόρμα Arduino Mini

Το **Arduino Mini** είναι η πιο μικρή σε μέγεθος έκδοση που χρησιμοποιεί τον Atmega328 έχει 14 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 6 μπορούν να παράγουν P.W.M, 8 αναλογικές θύρες, 16MHz ταλαντωτή κρυστάλλου και κουμπί «reset».



Εικόνα 1.6.1 Arduino Mini

1.6.2 Υπολογιστική πλατφόρμα Arduino Pro Mini

Το **Arduino Pro Mini** χρησιμοποιεί τον Atmega168 έχει 14 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 6 μπορούν να παράγουν P.W.M, 6 αναλογικές θύρες, και η βασική του διαφορά με τον Pro πέρα του ότι είναι ελαφρά μεγαλύτερος είναι ότι έχει προ-συγκολλημένες θύρες (Pin). Σχεδιάστηκε και παράγεται από την εταιρεία «SparkFun Electronics».

Εικόνα 1.6.2 Arduino Pro Mini

1.6.3 Υπολογιστική πλατφόρμα Arduino Micro

Το **Arduino Micro** χρησιμοποιεί τον ATmega32u4 έχει 20 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 7 μπορούν να παράγουν P.W.M, 12 αναλογικές θύρες, 16MHz ταλαντωτή κρυστάλλου, κουμπί «reset» κεφαλή «ICSP». Το βασικό του πλεονέκτημα είναι το ενσωματωμένο(«built-in») usb με το οποίο μπορεί να χρησιμοποιηθεί ως πληκτρολόγιο και ποντίκι. Ο σχεδιασμός και η παραγωγή του γίνεται σε συνεργασία με την εταιρεία «Adafruit».

Εικόνα 1.6.3 Arduino Micro

1.6.4 Υπολογιστική πλατφόρμα Arduino Nano

Το **Arduino Nano** χρησιμοποιεί τον Atmega328 ουσιαστικά βασίζεται στον Arduino Duemilanove με τη διαφορά ότι είναι μικρότερος σε μέγεθος και στην απώλεια εξωτερικής τροφοδοσίας. Επίσης συνδέεται μέσω Mini-B USB και σχεδιάστηκε και παράγεται από την εταιρεία «Gravitech».

Εικόνα 1.6.4 Arduino Nano

1.6.5 Υπολογιστική πλατφόρμα Arduino Frio

Το **Arduino Frio** χρησιμοποιεί τον ATmega328P η τροφοδοσία του γίνεται με 3.3V και δουλεύει στα 8 Mhz, έχει 14 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 6 μπορούν να παράγουν P.W.M, καθώς και 8 αναλογικές θύρες. Το σημαντικότερο πλεονέκτημα του σε σχέση με τις υπόλοιπες εκδόσεις είναι ότι σχεδιάστηκε για εφαρμογές που χρησιμοποιούν ασύρματα πρωτόκολλα επικοινωνίας γι'αυτό και έχει ενσωματωμένη θύρα για μπαταρία και υποδοχή «usb-to-Xbee».

Εικόνα 1.6.5 Arduino Frio

1.6.6 Υπολογιστική πλατφόρμα Arduino Pro

Το **Arduino Pro** χρησιμοποιεί τον Atmega328 ή τον Atmega168 η τροφοδοσία του γίνεται με 3.3V ή 5V ,δουλεύει στα 8 Mhz ή στα 16MHz ανάλογα με την έκδοση, έχει 14 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 6 μπορούν να παράγουν P.W.M, καθώς και 6 αναλογικές θύρες .Σχεδιάστηκε για σταθερές εφαρμογές για αυτό και οι υποδοχές του δεν είναι συγκολλημένες. Σχεδιάστηκε και παράγεται από την εταιρεία «SparkFun Electronics».

Εικόνα 1.6.6 Arduino Pro

1.6.7 Υπολογιστική πλατφόρμα Arduino Leonardo

Το **Arduino Leonardo** χρησιμοποιεί τον ATmega32u4 έχει 20 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 7 μπορούν να παράγουν P.W.M, 12 αναλογικές θύρες, 16MHz ταλαντωτή κρυστάλλου και κουμπί «reset». Επίσης έχει ενσωματωμένο («built-in») usb με το οποίο μπορεί να χρησιμοποιηθεί ως πληκτρολόγιο και ποντίκι.

Εικόνα 1.6.7 Arduino Leonardo

1.6.8 Υπολογιστική πλατφόρμα Arduino Due

Το **Arduino Due** χρησιμοποιεί τον ATMEL SAM3X8E ARM Cortex-M3 CPU και είναι η πρώτη υπολογιστική πλατφόρμα Arduino που έχει ARM πυρήνα 32-bit μικροελεγκτή. Έχει 54 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 12 μπορούν να παράγουν P.W.M και άλλες δύο θύρες μπορούν να χρησιμοποιηθούν για «I2C» επικοινωνία, 12 αναλογικές θύρες, 84MHz ταλαντωτή κρυστάλλου, κουμπί «reset», 4 σειριακές θύρες για «hardware micro-usb», δύο θύρες μετατροπής ψηφιακού σήματος σε αναλογικό, «SPI» και «JTAG» κεφαλή, ενώ τροφοδοτείται μόνο με 3.3V.

Εικόνα 1.6.8 Arduino Due

1.6.9 Υπολογιστική πλατφόρμα Arduino Mega 2560

Το **Arduino Mega 2506** είναι η πιο ισχυρή και πιο πρόσφατη υπολογιστική πλατφόρμα Arduino με πάρα πολλές δυνατότητες και είναι η αναβάθμιση του Arduino Mega. Χρησιμοποιεί τον Atmega2560 μικροελεγκτή της ATMEL. Έχει 54 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 15 μπορούν να παράγουν «8-bit» P.W.M και άλλες 2 θύρες μπορούν να χρησιμοποιηθούν για «I2C» επικοινωνία καθώς και 6 εξωτερικά interrupt, 16 αναλογικές θύρες, 16MHz ταλαντωτή κρυστάλλου, κουμπί «reset», 4

σειριακές θύρες για «hardware» κεφαλή, και 4 θύρες «SPI» για την σύνδεση με άλλα περιφερειακά ή πλακέτες.

Εικόνα 1.6.9 Arduino Mega 2506

1.6.10 Υπολογιστική πλατφόρμα Arduino ADK

Το **Arduino ADK** είναι ουσιαστικά ακριβώς η ίδια υπολογιστική πλατφόρμα με τον Arduino Mega 2506 με τη σημαντική διαφορά ότι η διασύνδεση «usb» γίνεται από τον ελεγκτή «MAX3421E IC» και επιτρέπει στην υπολογιστική πλατφόρμα να επικοινωνήσει και να αλληλεπιδρά με οποιαδήποτε συσκευή έχει θύρα «usb», για παράδειγμα κινητά τηλέφωνα, φωτογραφικές μηχανές, πληκτρολόγια, ποντίκια, παιχνιδομηχανές κ.α.

Εικόνα 1.6.10 Arduino AD

1.6.11 Υπολογιστική πλατφόρμα Arduino Ethernet

Το **Arduino Ethernet** χρησιμοποιεί τον ATmega328 έχει 14 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 4 μπορούν να παράγουν P.W.M, 6 αναλογικές θύρες, 16MHz ταλαντωτή κρυστάλλου και κουμπί «reset». Κύριο χαρακτηριστικό του η απουσία σειριακής θύρας για «usb» όπου αντικαθίσταται με ένα «Ethernet» ελεγκτή της «Wiznet» απ'όπου γίνεται και η εξωτερική σύνδεση του με οποιοδήποτε δίκτυο. Επίσης έχει ενσωματωμένο ένα αναγνώστη καρτών «micro-SD» για την αποθήκευση αρχείων.

Εικόνα 1.6.11 Arduino Ethernet

1.6.12 Υπολογιστική πλατφόρμα Arduino BT

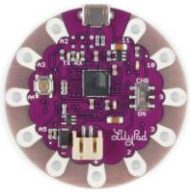
Το **Arduino BT** χρησιμοποιεί τον Atmega328 μικροελεγκτή της ATMEL. Έχει 14 ψηφιακές θύρες εισόδου και εξόδου (I/O) εκ των οποίων οι 6 μπορούν να παράγουν P.W.M και άλλη μία θύρα για επαναφορά του bluetooth, 6 αναλογικές θύρες και 16MHz ταλαντωτή. Το πιο σημαντικό όμως αυτής της υπολογιστικής πλατφόρμας Arduino είναι το

«bluetooth chip WT11» της «Bluegiga», το οποίο επιτρέπει την ασύρματη επικοινωνία της πλατφόρμας μέσω «bluetooth» με οποιαδήποτε συσκευή χρησιμοποιεί την αντίστοιχη τεχνολογία εκτός από ακουστικά και συσκευές ήχου.

Εικόνα 1.6.12 Arduino BT

1.6.13 Υπολογιστική πλατφόρμα Arduino Lilypad Usb

Η πιο μοντέρνα έκδοση της υπολογιστικής πλατφόρμας Arduino, σχεδιασμένη να ράβεται εύκολα πάνω σε ρούχα αλλά και εξοπλισμένη με τον Atmega32u4 έχει όλα τα χαρακτηριστικά και τις δυνατότητες των άλλων εκδόσεων όπως ενσωματωμένο (built-in) usb με το οποίο μπορεί να χρησιμοποιηθεί ως πληκτρολόγιο και ποντίκι συγκεκριμένη έκδοση έχει δυνατότητα σύνδεσης με usb σε αντίθεση με το Lilypad Simple, Lilypad Simple Snap και το Lilypad που είναι και η βασική έκδοση της συγκεκριμένης σειράς.



Εικόνα 1.6.13 Arduino Lilypad Usb

1.6.14 Υπολογιστική πλατφόρμα Arduino Esplora

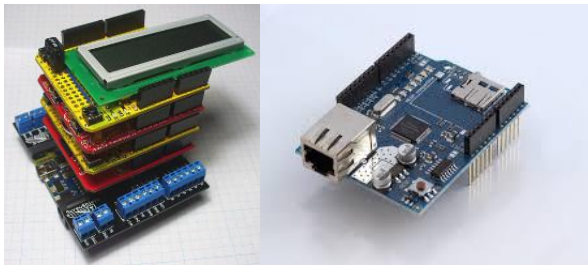
Η συγκεκριμένη υπολογιστική πλατφόρμα **Arduino** βασίζεται πάνω στο Arduino Leonardo και έχει το μοναδικό έως τώρα πλεονέκτημα, σε σχέση με τις υπόλοιπες, να έχει ενσωματωμένα και έτοιμα για χρήση διάφορους αισθητήρες, χειριστήρια διακόπτες, LED, μικρόφωνο κ.α. Απευθύνεται βασικά σε χρήστες που δεν είναι αρκετά εξοικειωμένη με τα ηλεκτρολογικά και ηλεκτρονικά κυκλώματα.

Εικόνα 1.6.14 Arduino Esplora

1.7 Μονάδες επέκτασης υπολογιστικής πλατφόρμας Arduino («Arduino Shields»)

Πέραν όμως της μεγάλης ποικιλίας των υπολογιστικών πλατφορμών Arduino, υπάρχει και μία μεγάλη ποικιλία από πλακέτες οι οποίες μπορούν να προσαρμοστούν και να συνδεθούν με την **υπολογιστική πλατφόρμα Arduino**, με σκοπό την προέκταση των δυνατοτήτων τους, οι οποίες ονομάζονται «shields». Κάποιες από αυτές είναι οι εξής:

- **Arduino GSM Shield:** Συνδέει το Arduino με το internet μέσω GPRS ασύρματου δικτύου.
- **Arduino Ethernet Shield:** Συνδέει το Arduino με το internet μέσω ενός καλωδίου ethernet.
- **Arduino WIFI Shield:** Συνδέει το Arduino με το internet ασύρματα.
- **Arduino WIFI SD Shield:** Συνδέει το Arduino με άλλη συσκευή όπως το Xbee της DIGI ή οποιοδήποτε άλλο παρόμοιο chip ασύρματα μέχρι και σε απόσταση 100 μέτρων.
- **Arduino Motor Shield:** Το Arduino μπορεί να ελέγξει ως και 2 κινητήρες ελέγχοντας την ταχύτητα και την κατεύθυνση του ανεξάρτητα στο καθένα.
- **Arduino GPS Shield:** Συνδέει το Arduino με τουλάχιστον 3 δορυφόρους και επιστρέφει στο Arduino έναν αριθμό μεταβλητών που αντιστοιχούν σε συντεταγμένες.



Εικόνα 1.7 Διάφορα είδη "Arduino Shield"

1.8 Αναλυτική περιγραφή της υπολογιστικής πλατφόρμας Arduino Uno

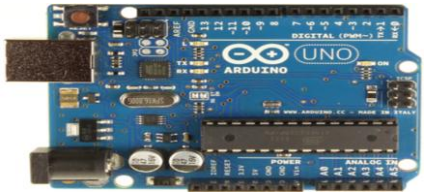
Η δημοφιλέστερη έκδοση του Arduino είναι η Duemilanove/Uno που βασίζεται στο ολοκληρωμένο ATmega 328, έναν 8-bit RISC μικροελεγκτή, ο οποίος χρονίζει στα 16MHz. Εμείς θα ασχοληθούμε συγκεκριμένα με την έκδοση Uno που στα Ιταλικά σημαίνει «ένα» και όπως αναφέρουν οι κατασκευαστές του μαζί με την «1.0» είναι οι εκδόσεις αναφοράς για τις υπόλοιπες εκδόσεις.

1.8.1 Θύρες εισόδου και εξόδου της υπολογιστικής πλατφόρμας Arduino Uno

Η υπολογιστική πλατφόρμα του **Arduino Uno** διαθέτει :

• 14 ψηφιακές θύρες εισόδου και εξόδου («I/O»). Ανάλογα με το πρόγραμμα που θα φορτωθεί στον μικροελεγκτή αυτές οι θύρες μπορούν να εργαστούν σαν είσοδοι ή έξοδοι ψηφιακών σημάτων.

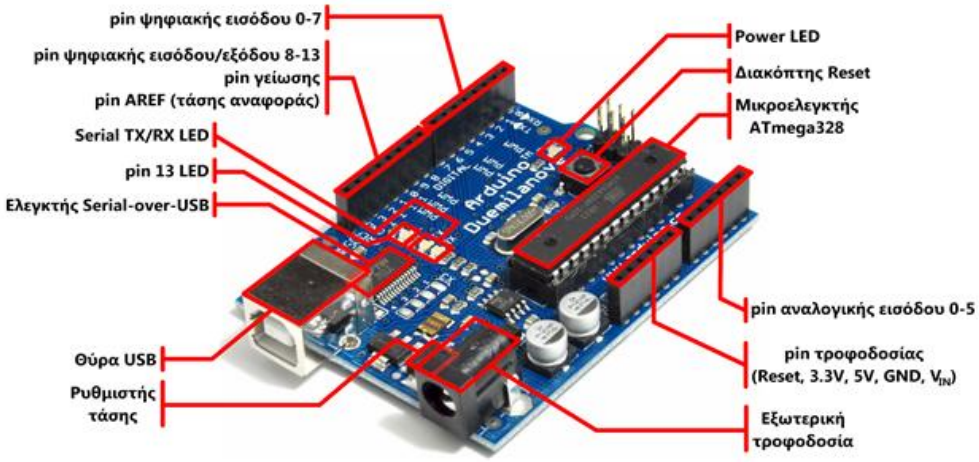
• Οι ψηφιακές θύρες 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές θύρες εξόδου με το σύστημα PWM («Pulse Width Modulation»), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Το PWM παίρνει ένα εύρος τιμών από το 0 έως το 255. Δεν είναι πραγματικά αναλογικό σύστημα, έτσι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα παρέχει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει έναν παλμό που η τάση του θα εναλλάσσεται με μεγάλη συχνότητα και για ίσα χρονικά διαστήματα μεταξύ των τιμών 0V και 5V με σκοπό η μέση τιμή να ισούται με 2,5V.



Εικόνα 1.8.1.2 Arduino Uno

- Οι θύρες 0 και 1 χρησιμοποιούνται επίσης και για να λαμβάνουν (RX) και να μεταδίδουν (TX) TTL σειριακά δεδομένα. Έτσι, όταν για παράδειγμα το πρόγραμμα στέλνει δεδομένα σειριακά, τότε αυτά προωθούνται στην θύρα USB μέσω του ελεγκτή «Serial-Over-Usb» όπως επίσης και στο pin 0 για να τα διαβάσει ενδεχομένως μία άλλη συσκευή (π.χ. ένα δεύτερο Arduino στη δικιά του θύρα 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιηθεί το σειριακό «interface», καταλαμβάνονται δύο ψηφιακές θύρες εισόδου/εξόδου.
- Οι θύρες 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορούν να ρυθμιστούν μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές τάσης, η κανονική ροή του προγράμματος να σταματάει άμεσα και να εκτελείται μία συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας. Έξη αναλογικές θύρες εισόδου αριθμημένες από το 0 έως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC («Analog to Digital Converter»). Για παράδειγμα, αν

τροφοδοτηθεί ένα από αυτά τα pin με μία τάση η οποία μπορεί να κυμανθεί με ένα ποτενσιόμετρο από 0V ως μία τάση αναφοράς Vref (η οποία αν δεν γίνει κάποια αλλαγή είναι προρυθμισμένη στα 5V), τότε μέσα από το πρόγραμμα μπορεί να «διαβαστεί» η τιμή της θύρας ως ένας ακέραιος αριθμός χωρητικότητας 10-bit, από το 0 (όταν η τάση στο pin είναι 0V) μέχρι το 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μία εντολή όπως για παράδειγμα στα 1.1V. Ένας άλλος τρόπος όπου η τάση αναφοράς μπορεί να δηλωθεί από τον προγραμματιστή είναι τροφοδοτώντας με μία εξωτερική τάση αναφοράς τη θύρα με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της υπολογιστικής πλατφόρμας Arduino. Έτσι, αν τροφοδοτηθεί η θύρα AREF με 3.3V και στην συνέχεια εκτελεστή η εντολή να διαβαστεί κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512.



*Εικόνα 1.8.1.3 Θύρες εισόδου και εξόδου της υπολογιστικής πλατφόρμας **Arduino Uno***

Δίπλα από της θύρες αναλογικής εισόδου, υπάρχει μία ακόμα συστοιχία από 6 pin με την σήμανση **«POWER»**. Η λειτουργία του καθενός Pin έχει ως εξής:

1) Το πρώτο, με την ένδειξη **«RESET»**, όταν γειωθεί (με οποιοδήποτε από τα 3 pin με την ένδειξη «GND» που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση της υπολογιστικής πλατφόρμας Arduino.

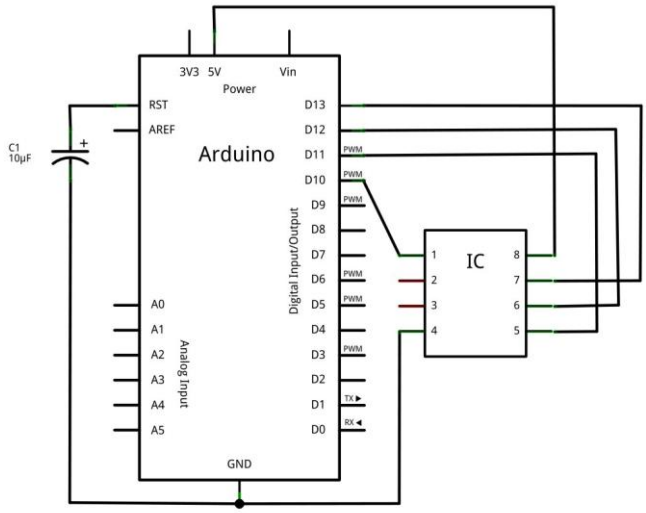
2) Το δεύτερο με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει διατάξεις, συσκευές ή αισθητήρες με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή «Serial-over-Usb» και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.

3) Η τρίτη θύρα με την ένδειξη **«5V»**, μπορεί να χρησιμοποιηθεί και αυτή για την τροφοδότηση διαφόρων εξαρτημάτων, συσκευών ή αισθητήρων με τάση 5V. Ανάλογα με

τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως παρέχει τάση 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «σταθεροποιήσει» στα 5V.

4) Το τέταρτο και το πέμπτο pin με την ένδειξη **«GND»** είναι οι γειώσεις.

5) Το έκτο και τελευταίο pin, με την ένδειξη **«Vin»** έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino ή μπορεί να χρησιμοποιηθεί αυτό το pin για να τροφοδοτήσει εξαρτήματα και συσκευές με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.



Εικόνα 1.8.1.4 Σχηματικό διάγραμμα θυρών υπολογιστικής πλατφόρμας Arduino Uno

1.8.2 Υπόλοιπα μέρη της υπολογιστικής πλατφόρμας Arduino Uno

Πάνω στην υπολογιστική πλατφόρμα του **Arduino Uno** υπάρχει ένας διακόπτης «micro-switch» και 4 SMD LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση «RESET») και του ενός LED με την σήμανση «POWER» είναι προφανής. Τα δύο LED με τις σημάνσεις «TX» και «RX», χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει αντίστοιχα δεδομένα μέσω της «Usb». Τα LED αυτά ελέγχονται από τον ελεγκτή «Serial-over-Usb» και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin «0» και «1». Τέλος, υπάρχει το LED με τη σήμανση «L». Η βασική λειτουργία του LED στην υπολογιστική πλατφόρμας είναι για να αναβοσβήνει συνήθως για δοκιμαστικό σκοπό. Οι κατασκευαστές σκέφτηκαν να ενσωματώσουν ένα LED, το οποίο το σύνδεσαν στη ψηφιακή θύρα 13. Έτσι ακόμα και αν δεν έχει συνδεθεί τίποτα πάνω στο

φυσικό pin 13, αναθέτοντας του την τιμή «HIGH» μέσα από το πρόγραμμα, θα ενεργοποιηθεί το ενσωματωμένο LED «L».

Όσον αφορά την τροφοδοσία της **υπολογιστικής πλατφόρμας Arduino**, αυτή μπορεί να τροφοδοτηθεί από μία «Usb» θύρα ενός υπολογιστή ή από μία εξωτερική πηγή τροφοδοσίας (9 volt) συνεχούς ρεύματος με βύσμα 2.1mm «barrel tip». Ο θετικός πόλος θα πρέπει να βρίσκεται στη εσωτερική πλευρά και ο αρνητικός στην εξωτερική πλευρά του βύσματος. Η φόρτωση του «sketch»(κώδικα) πραγματοποιείται μέσω μίας «Usb» θύρας που διαθέτει η υπολογιστική πλατφόρμα Arduino. Έτσι οι πληροφορίες που προέρχονται από την «Usb» θύρα του υπολογιστή εισέρχονται στην «Usb» θύρα του Arduino και στην συνέχεια οδηγούνται στο FDTI ολοκληρωμένο για να διαμορφωθούν σε μία κατάλληλη μορφή ώστε ο μικροελεγκτής να μπορέσει να τις διαβάσει. Οι μικροελεγκτές ATMega προγραμματίζονται χρησιμοποιώντας σειριακή επικοινωνία με τους υπολογιστές, έτσι το «FDTI» αναλαμβάνει την εργασία της μετατροπής της σειριακής θύρας σε «Usb».

1.8.3 Μνήμες που χρησιμοποιούνται από τις υπολογιστικές πλατφόρμες Arduino

Όπως σε όλες τις υπολογιστικές πλατφόρμες έτσι και σε αυτή του Arduino χρησιμοποιούνται διάφορες μνήμες για την προσπέλαση των δεδομένων, ο ATmega328 που χρησιμοποιείται στον **Arduino Uno** διαθέτει ενσωματωμένη μνήμη **τριών τύπων**:

1.8.3.1 Μνήμη «Flash memory»

Η μνήμη **Flash** έχει χωρητικότητα 32Kb, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware των πλατφόρμων που έχει εγκαταστήσει ήδη ο κατασκευαστής. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται **bootloader** είναι αναγκαίο για την εγκατάσταση των προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash δεν χάνει τα περιεχόμενά της με την απώλεια της τροφοδοσίας ή κάνοντας reset το μικροελεγκτή. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση αποθήκευσης κατά την διάρκεια εκτέλεσης του προγράμματος, μέσα από τα προγράμματα λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μία βιβλιοθήκη που επιτρέπει την εγγραφή στον χώρο που περισσεύει από την αποθήκευση των sketch (30Kb μείον το μέγεθος του προγράμματος σε μεταγλωττισμένη μορφή).

1.8.3.2 Μνήμη «SRAM memory»

Η μνήμη **SRAM** («static random access memory») είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το «runtime». Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος σταματήσει ή αν γίνει reset. Στο ATmega328 η **SRAM** μνήμη καταλαμβάνει χώρο 2048 bytes κατά την διάρκεια μίας κανονικής λειτουργίας και όλες οι μεταβλητές φορτώνονται σε αυτή καθ' όλη την διάρκεια της λειτουργίας του μικροελεγκτή.

1.8.3.3 Μνήμη «EEPROM memory»

Το τελευταίο μέρος της μνήμης είναι η **EEPROM** και καταλαμβάνει 1024 bytes, αρκετά μικρή για μνήμη που χρησιμοποιείται μόνο για ανάγνωση (read-only). Η **EEPROM** έχει όριο ζωής καθώς δε μπορεί να επαναπρογραμματιστεί για περισσότερες από 100.000 φορές. Είναι μία «byte addressable» μνήμη, γεγονός που καθιστά λίγο δυσκολότερο να τεθεί σε χρήση αφού απαιτείται ειδική βιβλιοθήκη ώστε να μπορέσει κάποιος να έχει πρόσβαση σε αυτή.

1.8.4 Φυσικά χαρακτηριστικά της υπολογιστικής πλατφόρμας Arduino Uno

Παρακάτω παρατίθενται συμπτυγμένα τα φυσικά χαρακτηριστικά της πλατφόρμας **Arduino Uno** σε μορφή πίνακα:

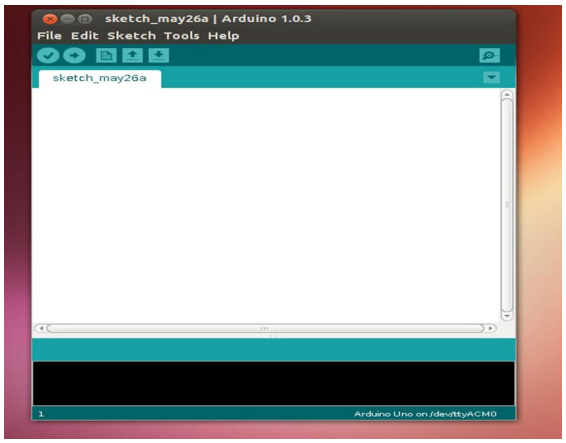
Μικροεπεξεργαστής	**ATmega328**
Τάση Λειτουργίας	**5V**
Συνιστώμενη Τάση Εισαγωγής	**7-12V**

Τάση Εισαγωγής(όρια)	**6-20V**
Ψηφιακές I/O Είσοδοι	**14(εκ των οποίων οι 6 μπορούν να παράγουν P.W.M)**
Αναλογικές Είσοδοι	**6**
Συνεχές Ρεύμα(ανά είσοδο)	**40 mA**
Συνεχές Ρεύμα Εισόδου 3.3V	**50 mA**
Μνήμη	**32 KB (ATmega328 εκ των οποίων 0.5 KB χρησιμοποιεί ο bootloader)**
SRAM	**2 KB(ATmega328)**
EEPROM	**1 KB(ATmega328)**
Ταχύτητα Ρολογιών	**16 MHz**

1.9 Το περιβάλλον προγραμματισμού Arduino IDE (Integrated Development Environment)

Το περιβάλλον προγραμματισμού του **Arduino (IDE)** είναι υπερβολικά απλό σε σχέση με αντίστοιχα περιβάλλοντα ανάπτυξης όπως το Eclipse , το Xcode και το Visual Studio, περιέχει έναν κειμενογράφο («editor»), έναν μεταγλωττιστή («compiler»), ένα «loader» και μια σειριακή οθόνη («serial monitor») παρόλα αυτά δεν περιέχει προχωρημένες λειτουργίες όπως αποσφαλμάτωση («debugging»). Στο **Arduino IDE** υπάρχουν οι επιλογές του compile δηλαδή της μεταγλώττισης του κώδικα, του «upload» δηλαδή της φόρτωσης του κώδικα και οι επιλογές «new», «open», «save» με τις οποίες ο χρήστης μπορεί να δημιουργήσει ανοίξει ή να σώσει ένα «σχέδιο». Μέσα στο μενού

υπάρχουν οι επιλογές για την σειριακή θύρα («serial port») με την οποία επικοινωνεί το Arduino με τον Η/Υ και η καρτέλα επιλογής της υπολογιστικής πλατφόρμας Arduino που χρησιμοποιείται κάθε φορά («board»).



Εικόνα 1.9 Απεικόνιση του Arduino IDE

Άλλες επιλογές του **Arduino IDE** είναι το «Archive Sketch», το οποίο μετατρέπει την εφαρμογή που αναπτύσσεται σε ένα αρχείο «zip» και το αποθηκεύει, η επιλογή εμφάνισης του «Serial Monitor», όπου βλέπουμε σε ένα νέο παράθυρο τα δεδομένα και το «Import Library» με το οποίο ανεβάζει ο χρήστης μια συγκεκριμένη βιβλιοθήκη είτε έτοιμη από την ομάδα του Arduino είτε μία δικιά του. Αρκετές από αυτές βρίσκονται προεγκατεστημένες στο περιβάλλον προγραμματισμού **Arduino IDE** και κάποιες από τις πιο σημαντικές είναι:

- **Liquid Crystal** :εφαρμόζεται για τον έλεγχο επιδείξεων υγρών κρυστάλλων.
- **Servo**: εφαρμόζεται για τον έλεγχο Servo κινητήρων.
- **Stepper**: εφαρμόζεται για τον έλεγχο Stepper κινητήρων.
- **Wire**: η διεπαφή δύο καλωδίων (TWI/I2C) έχει την δυνατότητα αποστολής και λήψης δεδομένων εκτός των συσκευών και των αισθητηρίων.

Το **Arduino IDE** όπως και οτιδήποτε σχετίζεται με την υπολογιστική πλατφόρμα Arduino είναι «freeware» λογισμικό πράγμα που βοηθά τόσο στη χρήση του αφού υποστηρίζεται σε όλα τα διαθέσιμα λειτουργικά Η/Υ Windows, MAC OS, Linux, όσο και στην ανάπτυξη του μέσω του διαδικτύου με σκοπό την απλούστευση του περιβάλλοντος και την φιλικότητα προς το χρήστη.

1.10 Αναφορά στη «γλώσσα» της υπολογιστικής πλατφόρμας Arduino

Στη «γλώσσα» του **Arduino** το κάθε πρόγραμμα ακολουθεί μια συγκεκριμένη δομή καθώς και συγκεκριμένες συναρτήσεις που βασίζονται στη γλώσσα προγραμματισμού «C»

ή στη «C++», και με τη χρήση της βιβλιοθήκης "Wiring" γίνεται εφικτός ο προγραμματισμός της κάθε υπολογιστικής πλατφόρμας.

1.10.1 Βασική δομή του κώδικα

Η δομή των προγραμμάτων στηρίζεται στις δύο βασικές συναρτήσεις **setup()** και **loop()** ο ρόλος των οποίων παρουσιάζεται παρακάτω:

- **setup() :** Μέσα στην συνάρτηση αυτή καθορίζουμε τις μεταβλητές που χρησιμοποιεί το πρόγραμμα, τις θύρες εισόδου εξόδου και τις βιβλιοθήκες που πρόκειται να χρησιμοποιήσουμε
- **loop():** Η συνάρτηση αυτή αποτελεί τον κύριο ατέρμονα βρόγχο του κώδικα μας στον οποίο γράφουμε το κυρίως πρόγραμμα

1.10.2 Δομές Ελέγχου (Control Structures)

Οι **δομές ελέγχου** («Control Structures») είναι κάποιες συναρτήσεις οι οποίες χρησιμοποιούνται σε διάφορες γλώσσες προγραμματισμού και μας βοηθούν να κάνουμε τους απαραίτητους ελέγχους στο κώδικα μας για το αν μια κατάσταση είναι αληθής η ψευδής και ανάλογα να δώσουμε την κατάλληλη εντολή για τις επικείμενες ενέργειες μέσα στο πρόγραμμα.

- **if:** συνάρτηση που χρησιμοποιείται σε συνδυασμό με έναν τελεστή σύγκρισης, ελέγχει αν μια συγκεκριμένη κατάσταση έχει επιτευχθεί, όπως μιας εισόδου εάν είναι πάνω από ένα ορισμένο αριθμό και ανάλογα εάν είναι, ακολουθείται μια συγκεκριμένη ενέργεια, αλλιώς μια διαφορετική.
- **if / else:** συνάρτηση που μοιάζει με την «if» με την διαφορά ότι επιτρέπει μεγαλύτερο έλεγχο της ροής του κώδικα, επιτρέποντας πολλαπλές δοκιμές
- **while():** βρόγχος που εκτελείται συνεχώς και απείρως, έως ότου η έκφραση μέσα στην παρένθεση γίνει ψευδής.

1.10.3 Μεταβλητές της υπολογιστικής πλατφόρμας Arduino

Οι σταθερές είναι προκαθορισμένες μεταβλητές στη «γλώσσα» του **Arduino** και χρησιμοποιούνται για να απλοποιήσουν τα προγράμματα και διακρίνονται σε λογικά επίπεδα, αληθή και ψευδή (σταθερές Boolean) και συνήθως συμβολίζονται με «0» για ψευδή και «1» για αληθή. Κάθε ψηφιακή θύρα του Arduino μπορεί να δεχθεί δύο πιθανές τιμές, «HIGH» ή «LOW» δηλαδή ελέγχουμε μια είσοδο ή έξοδο ανάλογα εάν περνά ή όχι ρεύμα από τι θύρα που έχουμε δηλώσει. Η κάθε θύρα του Arduino μπορεί να δηλωθεί είτε ως είσοδος είτε ως έξοδος πράγμα που αλλάζει την ηλεκτρική συμπεριφορά της θύρας. Στο κώδικα χρησιμοποιούντα διάφορες μεταβλητές από άλλες γλώσσες προγραμματισμού, από τις πιο συνήθεις είναι οι ακέραιες σταθερές («Integer constants») είναι οι αριθμοί που

χρησιμοποιούνται απευθείας σε ένα πρόγραμμα, όπως το 123. Από προεπιλογή, οι αριθμοί αυτοί συμβολίζονται με το «int».

1.10.4 Βασικές συναρτήσεις της υπολογιστικής πλατφόρμας Arduino

Κάποιες από τις πιο βασικές συναρτήσεις που χρησιμοποιούνται στη σύνταξη ενός απλού κώδικα της υπολογιστικής πλατφόρμας Arduino είναι οι εξής:

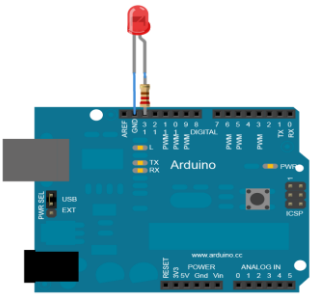
- **pinMode() :** κάθε θύρα μπορεί να δηλωθεί είτε ως είσοδος είτε ως έξοδος και συντάσσεται **pinMode (αριθμός θύρας, είσοδος/έξοδος)**
- **digitalWrite() :** κάθε ψηφιακή θύρα μπορεί να δεχθεί δύο πιθανές τιμές, «HIGH» ή «LOW» και συντάσσεται **digitalWrite (αριθμός θύρας, HIGH/LOW)**
- **digitalRead() :** «διαβάζει» την τιμή από μια ψηφιακή θύρα και συντάσσεται **digitalRead(αριθμός θύρας)**

1.11 Το παράδειγμα «BLINK»

Όπως σε όλες τις γλώσσες προγραμματισμού έτσι και στη γλώσσα του **Arduino** το πιο εύκολο και απλό εισαγωγικό παράδειγμα είναι το “Hello World”, δυστυχώς δεν υπάρχει ενσωματωμένη οθόνη στην υπολογιστική πλατφόρμα Arduino οπότε χρησιμοποιούμε το LED που βρίσκεται πάνω στην υπολογιστική πλατφόρμα Arduino για την αλληλεπίδραση του Arduino με τον «κόσμο». Στο συγκεκριμένο παράδειγμα το LED της υπολογιστικής πλατφόρμας Arduino θα αναβοσβήνει κάθε ένα δευτερόλεπτο μέχρι να αποσυνδέσουμε τον Arduino και βρίσκεται εγκατεστημένο μέσα στα παραδείγματα που έχει η εφαρμογή του Arduino (Arduino IDE) όταν την εγκαθιστούμε σε έναν Η/Υ.

```
// Στο Pin 13 σε όλες σχεδόν τις πλακέτες ARDUINO βρίσκεται το LED
// το οποίο και ονομάζουμε:

int led = 13;              // Δηλώνουμε τη μεταβλητή led ως integer
void setup() {
   pinMode(led, OUTPUT);   // Δηλώνουμε το LED ως έξοδο
             }
void loop() {               // Ατέρμον βρόχος
  digitalWrite(led, HIGH);  // Δηλώνουμε το LED ως  HIGH
  delay(1000);              // Περιμένουμε 1 δευτερόλεπτο
  digitalWrite(led, LOW);   // Δηλώνουμε το LED ως  LOW
  delay(1000);}             //  Περιμένουμε 1 δευτερόλεπτο
```

Με βάση το παραπάνω παράδειγμα μπορούμε να δημιουργήσουμε εύκολα διάφορες εφαρμογές χρησιμοποιώντας διάφορα περιφερειακά εξαρτήματα όπως LED, RC servo, DC και Stepper κινητήρες, πληκτρολόγια, αισθητήρες, οθόνες κ.α.



Εικόνα 1.11 Απεικόνιση της συνδεσμολογίας του παραδείγματος «Blink»

1.12 Οι Βιβλιοθήκες της υπολογιστικής πλατφόρμας Arduino

Όλες οι βιβλιοθήκες της **υπολογιστικής πλατφόρμας Arduino** βρίσκονται στο διαδίκτυο και μπορεί να τις κατεβάσει ο καθένας ελεύθερα και σκοπός τους είναι η ευκολία στη χρήση διαφόρων παρελκόμενων μαζί τους. Το περιβάλλον Arduino (IDE) μπορεί να επεκταθεί με τη χρήση βιβλιοθηκών, ακριβώς όπως οι περισσότερες πλατφόρμες προγραμματισμού, αφού οι βιβλιοθήκες παρέχουν επιπλέον λειτουργίες για τη χρήση σε σχέδια όπου χωρίς αυτές η δημιουργία τους θα ήταν περίπλοκη. Μια σειρά από βιβλιοθήκες υπάρχουν προεγκατεστημένες, και είναι πολύ εύκολο να τις κατεβάσετε και να τις επεξεργαστείτε ή ακόμα και να δημιουργήσετε τη δική σας.

1.12.1 Βιβλιοθήκη της οθόνης LCD

Η **βιβλιοθήκη LCD** βρίσκεται προεγκατεστημένη στο περιβάλλον προγραμματισμού του Arduino (IDE) και για την ορθή της λειτουργία ακολουθούμε την παρακάτω διαδικασία:

Εικόνα 1.12.1 Οθόνη LCD

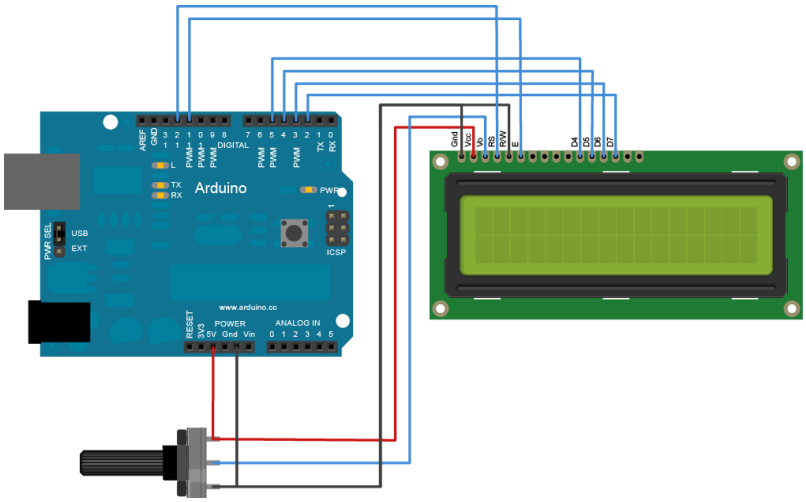
- Δηλώνουμε τη χρήση της βιβλιοθήκης στην αρχή του κώδικα π.χ. **#include <LiquidCrystal.h>**
- Δηλώνουμε τις θύρες (pin) στις οποίες θα συνδέσουμε την οθόνη π.χ. **LiquidCrystal lcd(RS_pin, Enable_pin, Dpin4, Dpin5, Dpin6, Dpin7)**
  για το συγκεκριμένο παράδειγμα έχουμε **LiquidCrystal lcd(12, 11, 5, 4, 3, 2)** (η δήλωση αυτή γίνεται πριν την συνάρτηση **setup()**).

- Χρησιμοποιώντας την εντολή **lcd.begin();** στην συνάρτηση **setup()** ορίζουμε τις γραμμές και τις στήλες της οθόνης που χρησιμοποιούμε, π.χ. **lcd.begin(αριθμός στηλών, αριθμός γραμμών),** η οθόνη που χρησιμοποιείται ως παράδειγμα μέσα στην βιβλιοθήκη έχει **16** στήλες και **2** γραμμές.
- Με την εντολή **lcd.print();** στην συνάρτηση **loop()** μπορούμε να εκτυπώσουμε οποιοδήποτε μήνυμα στην οθόνη π.χ. **lcd.print("hello, world!");**
- Τέλος με την εντολή **lcd.setCursor();** στην συνάρτηση **loop()** ορίζουμε την θέση του κέρσορα π.χ. **lcd.setCursor(0, 1);** θέτουμε τον κέρσορα στην πρώτη στήλη της οθόνης και στην δεύτερη γραμμή αφού η αρίθμηση ξεκινά από το μηδέν**.**

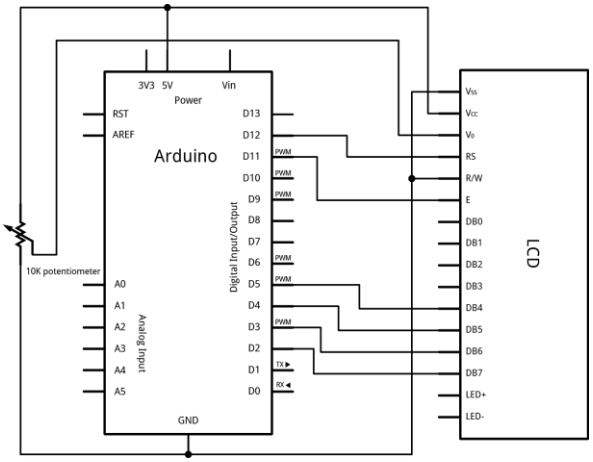
1.2.1.1 Παράδειγμα της χρήσης της βιβλιοθήκης της οθόνης «LCD»

Παρακάτω παραθέτουμε ένα απλό παράδειγμα της χρήσης της βιβλιοθήκης της **οθόνης LCD**, ώστε να γίνει αντιληπτός ο τρόπος με τον οποίο δηλώνουμε της θύρες του Arduino και της οθόνης αλλά και οι συναρτήσεις της βιβλιοθήκης.

```
#include <LiquidCrystal.h>       //  δήλωση της βιβλιοθήκης
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // δηλώνουμε τα PIN της οθόνης
 void setup() {
  lcd.begin(16, 2); // ορίζουμε τις γραμμές και τις στήλες της οθόνης
  lcd.print("hello,world!");} // εκτυπώνουμε το μήνυμα «hello,world»
 void loop() {
  lcd.setCursor(0, 1); } //θέτουμε τον κέρσορα στην επιθυμητή θέση
  lcd.print(millis()/1000); // εκτυπώνουμε τα δευτερόλεπτα από την

              }                  τελευταία επαναφορά
```



Εικόνα 1.12.2 Συνδεσμολογία της οθόνης LCD



Εικόνα 1.12.3 Σχηματικό διάγραμμα της οθόνης LCD

1.12.2 Βιβλιοθήκη των DC κινητήρων τύπου «RC-SERVO»

Όπως έχει προαναφερθεί και στις προηγούμενες βιβλιοθήκες έτσι και η βιβλιοθήκη των κινητήρων τύπου **«RC-Servo»** υπάρχει προ-εγκατεστημένη στο περιβάλλον προγραμματισμού του Arduino (IDE). Οι περισσότεροι κινητήρες τύπου «servo» κινούνται μεταξύ 0 και 180 μοιρών με καλή ακρίβεια και σε διαφορετικές ταχύτητες. Η βιβλιοθήκη των κινητήρων τύπου «servo» υποστηρίζει έως 12 κινητήρες για τις περισσότερες πλακέτες Arduino ενώ για τον Mega έως 48. Με την ενεργοποίηση της βιβλιοθήκης στην υπολογιστική πλατφόρμα Arduino, εκτός του Arduino Mega, απενεργοποιείται η λειτουργία «PWM» στις θύρες 9 και 10 ακόμα και αν δεν υπάρχει κινητήρας συνδεδεμένος, ενώ στον Mega μέχρι και 12 κινητήρες μπορούν να δουλέψουν χωρίς αυτή την δυσλειτουργία, εάν όμως χρησιμοποιηθούν 12 έως 23 κινητήρες απενεργοποιείται η λειτουργία «PWM» στις θύρες 11 και 12. Για την σωστή χρήση της βιβλιοθήκης ακολουθούμε την εξής διαδικασία:

- Δηλώνουμε την βιβλιοθήκη στην αρχή του κώδικα ως **#include <Servo.h>**
- Δηλώνουμε το όνομα του κινητήρα σαν μία μεταβλητή τύπου **Servo π.χ. Servo myservo_name;**
- Καθορίζουμε μέσα στην συνάρτηση **setup()** τη θύρα που είναι συνδεδεμένος ο κινητήρας π.χ. **myservo.attach(9);**
- Καθορίζουμε μέσα στην συνάρτηση **loop()** την επιθυμητή τιμή για την γωνία του κινητήρα π.χ. **myservo.write(180);**

1.12.3 Βιβλιοθήκη βηματικών κινητήρων τύπου «STEPPER»

Υπάρχουν δύο είδη βηματικών κινητήρων τύπου **«Stepper»**, οι μονοπολικοί («Unipolar») και οι διπολικοί («Bipolar»), όπως οι προηγούμενες βιβλιοθήκες έτσι και η

βιβλιοθήκη αυτών των κινητήρων υπάρχει προεγκατεστημένη στο περιβάλλον προγραμματισμού του Arduino (IDE). Η βιβλιοθήκη και για τους δύο τύπους κινητήρων είναι ίδια με μοναδική διαφορά τα τέσσερα καλώδια σύνδεσης που χρησιμοποιεί ο διπολικός κινητήρας σε αντίθεση με τα έξη που χρησιμοποιεί ο μονοπολικός. Για την σωστή χρήση της βιβλιοθήκης ακολουθούμε την εξής διαδικασία:

- Δηλώνουμε την βιβλιοθήκη στην αρχή του κώδικα ως **#include <Stepper.h>**
- Ορίζουμε τα βήματα (steps) του κινητήρα και τις θύρες που συνδέουμε τον κινητήρα στην υπολογιστική πλατφόρμα Arduino πριν την συνάρτηση **setup()**
  Π.χ. για τον μονοπολικό : **Stepper(steps, pin1, pin2)**
  Π.χ. για τον διπολικό : **Stepper(steps, pin1, pin2, pin3, pin4)**
- Ορίζουμε μέσα στην συνάρτηση **setup()** την ταχύτητα κίνησης του κινητήρα σε στροφές ανά λεπτό (rpm) π.χ. **setSpeed(rpms)**
- Τέλος στην συνάρτηση **loop()** δίνουμε τη επιθυμητή τιμή βημάτων ώστε να κινηθεί ο κινητήρας (για θετική τιμή ο κινητήρας στρέφεται αντίθετα από ότι για αρνητική τιμή βημάτων), π.χ. **step(steps)** π.χ. **Stepper: step(100**) ώστε ο κινητήρας να κάνει 100 βήματα [1,2]

1.12.4 Βιβλιοθήκη του ελεκτή PID

Η βιβλιοθήκη του ελεγκτή **PID** δεν υπάρχει προ-εγκατεστημένη στο περιβάλλον προγραμματισμού του Arduino (IDE) αλλά υπάρχει διαθέσιμη στο διαδίκτυο σε διάφορες εκδόσεις και ακολουθώντας μια απλή διαδικασία μπορούμε να τη χρησιμοποιήσουμε σε οποιαδήποτε υπολογιστική πλατφόρμα Arduino. Αφού κατεβάσουμε το φάκελο της βιβλιοθήκης από την ηλεκτρονική διεύθυνση του Arduino (*http://playground.arduino.cc*) τον αποσυμπιέζουμε και αντιγράφουμε τα αρχεία στο φάκελο «libraries» του Arduino. Για την σωστή χρήση της βιβλιοθήκης ακολουθούμε την εξής διαδικασία:

- Αρχικά να δηλωθεί η χρήση της βιβλιοθήκης στην αρχή του κώδικα **#include <PID_v1.h>**

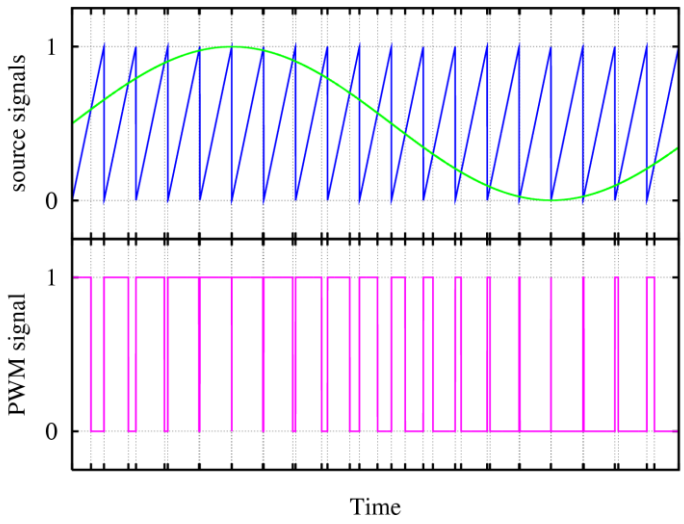
- **PID(&Input, &Output, &Setpoint, Kp, Ki, Kd, Direction)**
  Ορίζουμε την είσοδο(**Input**), την έξοδο(**Output**), την επιθυμητή τιμή(**Setpoint**), τα κέρδη **Kp,Ki,Kd** και τη λειτουργία του PID, στην οποία υπάρχουν δύο επιλογές, η επιλογή **DIRECT** ή **REVERSE** ανάλογα με το σφάλμα της εξόδου, επίσης επιλέγουμε **DIRECT** ή **REVERSE**, εάν η είσοδος είναι μεγαλύτερη από την επιθυμητή τιμή η έξοδος πρέπει να αυξάνεται ή να μειώνεται, ανάλογα με την χρήση του ελεγκτή, εάν π.χ. ελέγχουμε ένα αυτοκίνητο θα χρησιμοποιήσουμε την επιλογή «DIRECT», ενώ για ένα ψυγείο την επιλογή «REVERSE» με την εντολή

SetControllerDirection(Direction); π.χ. **PID(&Input, &Output, &Setpoint, 10, 5, 3, DIRECT)**

- Καθορίζουμε μέσα στην συνάρτηση **setup()** εάν ο PID θα βρίσκεται σε κατάσταση Automatic ή Manual δηλαδή ενεργοποιημένος ή απενεργοποιημένος αντίστοιχα, χρησιμοποιώντας το **SetMode(mode),** όπου **«mode»** αντικαθιστούμε με Automatic ή Manual
- Καθορίζουμε μέσα στην συνάρτηση **loop()** την εντολή **Compute()** η οποία υλοποιεί τον αλγόριθμο του PID και πρέπει να καλείται σε κάθε βρόχο, συνήθως δεν επιστρέφει τίποτα εκτός και αν υπάρχει στο κώδικα η συνάρτηση **SetSampleTime** όπου και υπολογίζει μια νέα έξοδο: **SetSampleTime(SampleTime)** καθορίζει το χρόνο που απαιτεί, για την υλοποίησή του, ο αλγόριθμος του PID, ο προκαθορισμένος χρόνος είναι τα 200 ms που για ρομποτικές εφαρμογές είναι αργός οπότε πρέπει να αλλαχθεί, παρόλα αυτά για αρκετές εφαρμογές είναι αρκετός π.χ. **SetSampleTime(100)**
- **SetOutputLimits(min, max)** Δηλώνουμε τα όρια στην συνάρτηση **loop()** για την τιμή τις εξόδου, αρχικά είναι ορισμένη από 0 έως 255 π.χ. **SetOutputLimits(0, 100)** [3]

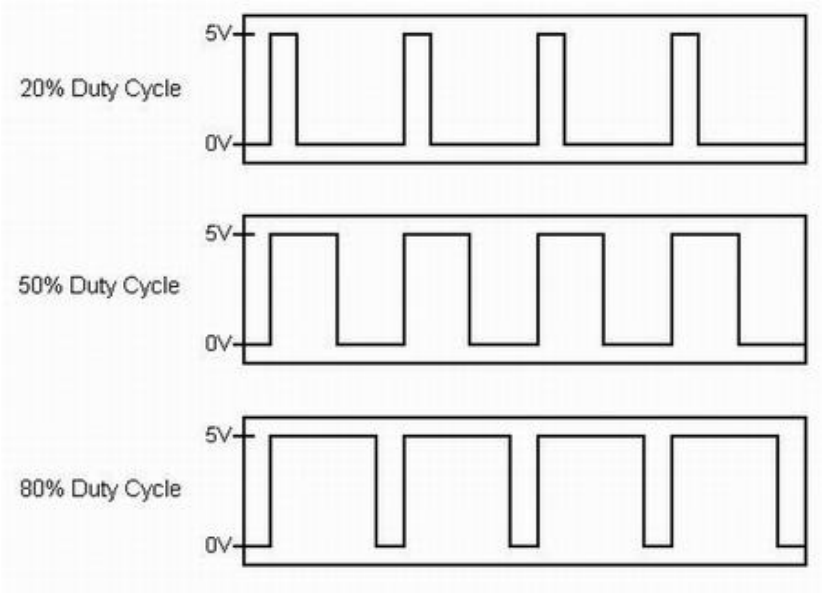
1.13 Διαμόρφωση Εύρους Παλμών (P.W.M)

Η **Διαμόρφωση Εύρους Παλμών** γνωστή και ως **«PWM»** («Pulse Width Modulation») ή αλλιώς Pulse Duration Modulation («PDM») είναι μια τεχνική διαμόρφωσης που συμμορφώνει το σήμα του παλμού και χρησιμοποιεί ένα ψηφιακό κύκλωμα για να δημιουργήσει ένα μεταβλητό αναλογικό σήμα. Το **P.W.M** ουσιαστικά ανοίγει και κλείνει ένα διακόπτη με ομοιόμορφα, επαναλαμβανόμενα χρονικά διαστήματα, η τεχνική **P.W.M** είναι ένας πανίσχυρος τρόπος στο να ελέγχουμε το αναλογικό σήμα του κυκλώματος και τα συστήματά του, χρησιμοποιώντας την ψηφιακή έξοδο των μικροεπεξεργαστών (microprocessors). Μέσω της χρήσης μετρητών υψηλής ανάλυσης, ο κύκλος λειτουργίας ενός τετραγωνικού παλμού διαμορφώνεται για να κωδικοποιεί μία συγκεκριμένη στάθμη αναλογικού σήματος. Το σήμα P.W.M είναι αναλογικό διότι σε κάθε χρονική στιγμή η πλήρης παροχή είναι είτε ολοσχερώς on, είτε off. Η τάση ή η πηγή ρεύματος παρέχεται στο αναλογικό φορτίο με την έννοια των επαναλαμβανομένων σειρών on και off παλμών. Ο ενεργός χρόνος (on-time) είναι ο χρόνος κατά τη διάρκεια του οποίου η παροχή εφαρμόζεται στο φορτίο και ο ανενεργός χρόνος (off- time) είναι ο χρόνος όπου η παροχή έχει κλείσει. Δεδομένου του επαρκούς εύρους ζώνης, κάθε αναλογική τιμή μπορεί να κωδικοποιηθεί με **P.W.M.** Ένα σήμα **P.W.M** χαρακτηρίζεται από τον κύκλο και χρόνο λειτουργίας του. Ο κύκλος λειτουργίας μετριέται σε επί τις εκατό (%) και η συχνότητα διαμόρφωσης σε Hz.



Εικόνα 1.13.1 Γραφική απεικόνιση του P.W.M

Στην Εικόνα 1.13.2 εμφανίζονται τρία διαφορετικά **P.W.M** σήματα το πρώτο έχει 20% «on» της περιόδου κύκλου λειτουργίας, το δεύτερο έχει το 50% «on» και το τρίτο το 80%. Και τα τρία σήματα είναι ταλαντώσεις τετραγωνικών παλμών, διαμορφωμένα ως προς το πλάτος των ταλαντώσεων τους, πράγμα το οποίο ονομάζεται κύκλος λειτουργίας. Τα σήματα αυτά έχουν την ίδια συχνότητα, αλλά διαφέρουν ως προς το πλάτος της θετικής κατάστασης. Ο κύκλος λειτουργίας είναι το επί τις εκατό της θετικής κατάστασης της περιόδου του σήματος.



Εικόνα 1.13.2 Παράδειγμα του «duty cycle»

Άρα, 20% on του κύκλου λειτουργίας σημαίνει ότι η θετική κατάσταση παραμένει για 20% της περιόδου του σήματος. Ο όρος «κύκλος λειτουργίας» περιγράφει την αναλογία του

χρόνου on ανά τακτά χρονικά διαστήματα. Ένας χαμηλός κύκλος λειτουργίας αντιστοιχεί σε χαμηλή ισχύ, διότι η ισχύς είναι off για μεγαλύτερο διάστημα. Σήμερα η τεχνική **P.W.M.** έχει μεγάλο εύρος εφαρμογών, όπως τον έλεγχο τάσης ρεύματος, ηλεκτροκινητήρα ισχύος στα «U.P.S», στους «Inverters» κ.α[9].

Κεφάλαιο 2

Το εκπαιδευτικό πακέτο Lego Mindstorms NXT

2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναφερθούμε στο εκπαιδευτικό πακέτο **Lego Mindstorms NXT**, σε κάθε μέρος της ξεχωριστά αναλύοντας τις δυνατότητες τους χρησιμοποιώντας τα και ως εκπαιδευτικά εργαλεία και ως προσομοιωτές τεχνολογικών και ρομποτικών εφαρμογών.

2.2 Ιστορική αναδρομή

Το εκπαιδευτικό πακέτο της Δανέζικης εταιρείας **Lego** αποτελεί μια σειρά προϊόντων από προγραμματιζόμενα «τούβλα», κινητήρες, αισθητήρες, απλά τουβλάκια Lego και διάφορα άλλα μηχανικά μέρη, όπως άξονες, ακτίνες αλυσίδες κ.α. και κρατώντας ως βασική αρχή της εταιρείας τα γνωστά της τούβλα με τα οποία είναι δυνατές εκπληκτικές κατασκευές εκπαιδευτικού και όχι μόνο χαρακτήρα και σε συνδυασμό με τα υπόλοιπα μέρη του πακέτου μπορεί να δοθεί στις κατασκευές κίνηση στο χώρο με διάφορους τρόπους και απόλυτη αλληλεπίδραση με το περιβάλλον καθιστώντας τα χρήσιμα εργαλεία

αυτοματισμού και ρομποτικού έλεγχου. Η αρχή είχε γίνει με την χρηματοδότηση της **Lego** σε ένα ερευνητικό πρόγραμμα του πανεπιστημίου «MIT»(«Massachusetts Institute of Technology») δημιουργώντας το «Media Lab» και ακολούθως το «Robotics Invention System»(RIS) το οποίο ήταν εξελιγμένο γύρο απ το RCX «τούβλο» που είχε ενσωματωμένο έναν 8-bit μικροελεγκτή της Renesas με 32K RAM. Με την μεγάλη εμπορική επιτυχία του «RIS» το 2006 κυκλοφόρησε το NXT με πιο εξελιγμένα αισθητήρια κινητήρες και λογισμικό και τέλος το 2009 το NXT 2.0 που είναι και η πιο εξελιγμένη μορφή της πλατφόρμας μέχρι σήμερα[8].

Εικόνα 2.1 Το NXT brick συνδεδεμένο με διάφορα μέρη της πλατφόρμας

2.3 Μέρη του εκπαιδευτικού πακέτου Lego Mindstorms NXT 2.0

Η εκπαιδευτική έκδοση του **Lego Mindstorms** αποτελείται από 1 «τούβλο»(«brick») NXT, 3 σερβοκινητήρες NXT, 2 αισθητήρες αφής, 1 αισθητήρα ήχου, 1 αισθητήρα φωτός,1 αισθητήρα υπερήχων και μηχανικά μέρη για την δημιουργία κατασκευών και φυσικά καλώδια σύνδεσης και φόρτισης της μπαταρίας. Αναλυτικά τα πιο σημαντικά μέρη:

Εικόνα 2.3 Διαφορές δημιουργίες με τη χρήση του εκπαιδευτικού πακέτου Lego Mindstorms NXT

2.3.1 Το NXT Brick

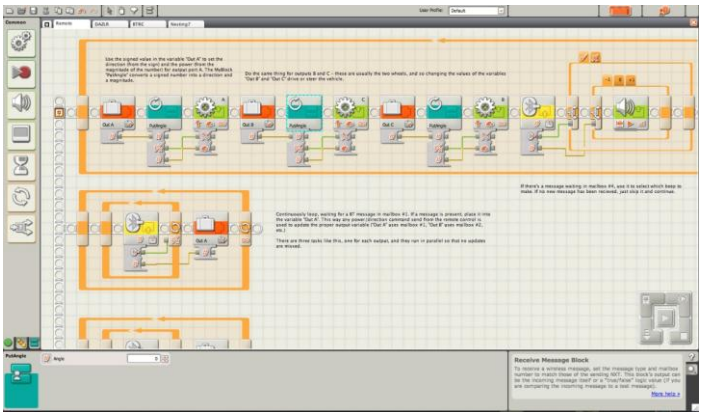
Το **NXT Brick** είναι η καρδιά της πλατφόρμας αφού μέσω αυτού γίνεται η επεξεργασία των δεδομένων και ο προγραμματισμός της κάθε κατασκευής. Λειτουργεί με 6 επαναφορτιζόμενες αλκαλικές μπαταρίες και έχει έναν 32-bit μικροεπεξεργαστή της Atmel, της σειράς ARM7 ο οποίος είναι χρονισμένος στα 48Mhz και έχει στη διάθεσή του 256KB Flash μνήμης και 64KB RAM. Για τον έλεγχο των κινητήρων έχει έναν ακόμη μικροεπεξεργαστή της Atmel αυτή τη φορά στα 8-bit με 4KB flash μνήμης και 512Byte RAM. Το «τούβλο» έχει τέσσερις εισόδους, αριθμημένες από το 1 μέχρι το 4, στις οποίες μπορούν να συνδεθούν οι αισθητήρες και άλλες 3 θύρες επικοινωνίας, από A μέχρι C, στις οποίες συνδέονται οι σερβοκινητήρες, σύνδεση Bluetooth, 100 x 64 pixel LCD οθόνη και ηχείο 8kHz. Η σύνδεση γίνεται με το καλώδιο RJ12 το οποίο είναι αντίστοιχο με εκείνο των τηλεφωνικών γραμμών.

Εικόνα 2.3.1 Το "NXT brick"

Το **NXT Brick** έχει τη δυνατότητα να προγραμματιστεί με διάφορους τρόπους καθώς η εταιρεία Lego θέλησε να δώσει τη δυνατότητα στους χρήστες του να μπορούν να πειραματιστούν με την πλατφόρμα με διάφορους τρόπους. Επιγραμματικά οι σημαντικότερες «γλώσσες» προγραμματισμού που υποστηρίζει η συσκευή είναι:

- **RCX Code** και **Labview** οι οποίες παρέχονται με την αγορά της πλατφόρμας
- **ROBOLAB** η οποία βασίζεται στο Labview και αναπτύχθηκε στο πανεπιστήμιο του Tufts
- **NXC,NQC** που βασίζονται στην C
- **RobotC** που βασίζεται στην C με ποιο φιλικό περιβάλλον
- **Lejos** η οποία βασίζεται στην Java

Το περιβάλλον προγραμματισμού είναι αρκετά εύχρηστο και φιλικό προς τον χρήστη και έχει ως αρχή λειτουργίας τα μπλοκ εντολών πράγμα που φαίνεται στην παρακάτω εικόνα[8]:



Εικόνα.2.3.2 Περιβάλλον προγραμματισμού του NXT Brick

2.3.2 Οι κινητήρες που χρησιμοποιεί το εκπαιδευτικό πακέτο Lego Mindstorms NXT 2.0

Οι **Interactive Servomotors του NXT** (αλληλεπιδραστικοί σερβοκινητήρες) λειτουργούν στα 9V και έχουν ενσωματωμένο αισθητήρα περιστροφής ο οποίος επιτρέπει στο NXT να ελέγχει τις κινήσεις τους με μεγάλη ακρίβεια χάρη στην περιστροφική ανατροφοδότηση («feedback»). Ο ενσωματωμένος αισθητήρας περιστροφής μετρά τις στροφές του κινητήρα σε βαθμούς (ακρίβεια + / - ένα βαθμό) ή πλήρεις περιστροφές. Μία περιστροφή είναι 360 μοίρες, οπότε αν έχετε ρυθμίσει τον κινητήρα για να γυρίσει 180 μοίρες, το κέντρο θα κάνει μισή στροφή. [7,8]

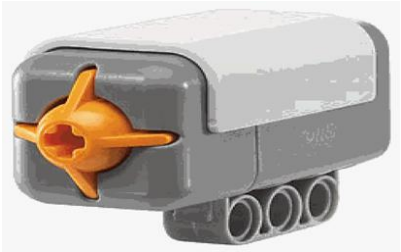
Εικόνα 2.3.3 Ο NXT κινητήρας

2.3.3 Αισθητήρες που χρησιμοποιεί το εκπαιδευτικό πακέτο Lego Mindstorms NXT 2.0

Το εκπαιδευτικό πακέτο **Lego Mindstorms NXT** περιέχει διάφορα είδη αισθητηρίων με τα οποία μπορεί ο χρήστης να δημιουργήσει πολλές διαφορετικές εφαρμογές και κατασκευές και αυτά που υπάρχουν στην εκπαιδευτική έκδοση είναι τα εξής:

2.3.3.1 Αισθητήρας αφής

Ο **αισθητήρας αφής** είναι ο πιο απλός αισθητήρας της σειράς αφού ουσιαστικά λειτουργεί σαν ένας απλός διακόπτης δίνοντας «HIGH» ή «LOW» ανάλογα με το αν βρίσκεται μπροστά του κάποιο αντικείμενο ή όχι.



Εικόνα 2.3.3.1 Αισθητήρας αφής

2.3.3.2 Αισθητήρας ήχου

Ο **αισθητήρας ήχου** ανιχνεύει το επίπεδο των ντεσιμπέλ δηλαδή την απαλότητα ή ένταση ενός ήχου, ανιχνεύει τόσο dB και dBA. DBA είναι οι ήχοι που τα ανθρώπινα αυτιά μπορούν να αναγνωρίσουν ενώ dB όλοι οι υπόλοιποι πραγματικοί ήχοι, συμπεριλαμβανομένων των ήχων σε πολύ υψηλή ή χαμηλή συχνότητα για το ανθρώπινο αυτί. Ο αισθητήρας ήχου μπορεί να μετρήσει επίπεδα ηχητικής πίεσης μέχρι και 90 dB - σχετικά με το επίπεδο της μηχανής. Τα ηχητικά επίπεδα των αισθητήρων εμφανίζεται σε ποσοστό [%] του ήχου που ο αισθητήρας είναι σε θέση για ανάγνωση. Για λόγους σύγκρισης, το 4-5% είναι σαν ένα σιωπηλό καθιστικό και 5-10% είναι το επίπεδο κάποιος λέει κάτι σε κάποιον από κάποια απόσταση. Από το 10-30% είναι κανονική συνομιλία κοντά στον αισθητήρα ή μουσική που παίζεται σε κανονικό επίπεδο και το 30-100% αντιπροσωπεύει μια σειρά από ανθρώπους που φωνάζουν ή μουσική που παίζεται σε πολύ υψηλή ένταση. Αυτές οι υποθέσεις είναι για μια απόσταση περίπου 1 μέτρο μεταξύ της ηχητικής πηγής και του αισθητήρα[7].

Εικόνα 2.3.3.2 Αισθητήρας ήχου

2.3.3.3 Αισθητήρας φωτός

Ο **αισθητήρας φωτός** ανιχνεύει το επίπεδο φωτός σε μία κατεύθυνση, και περιλαμβάνει επίσης ένα κόκκινο LED για το φωτισμό του αντικειμένου ώστε να μπορεί να ανιχνεύει το φως που αντανακλάται σε τιμές (χρησιμοποιώντας το ενσωματωμένο κόκκινο LED), ή το φως του περιβάλλοντος. Στο λογισμικό προγραμματισμού ο αισθητήρας ανιχνεύει φως σε μια κλίμακα από 0 έως 100, 100 είναι πολύ φωτεινό και 0 σκοτεινό. Εάν βαθμονομηθεί, ο αισθητήρας μπορεί επίσης να χρησιμοποιηθεί ως αισθητήρας απόστασης[7].

Εικόνα 2.3.3.3 Αισθητήρας φωτός

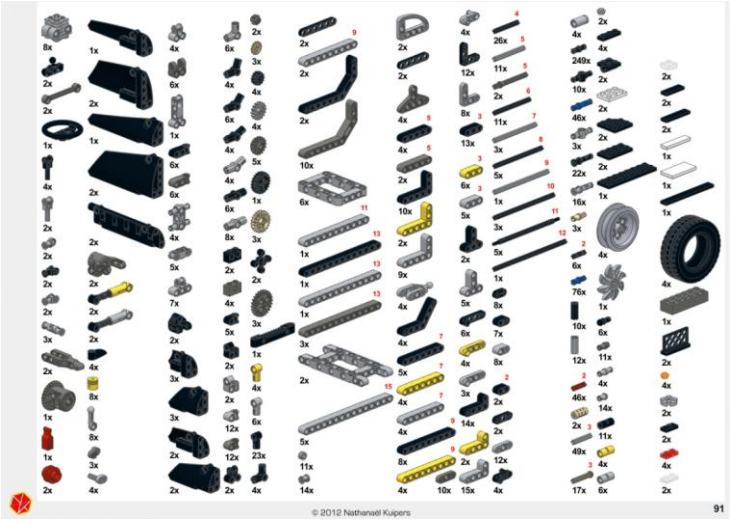
2.3.3.4 Αισθητήρας υπερήχων

Ο **αισθητήρας υπερήχων** μπορεί να μετρήσει την απόσταση από τον αισθητήρα σε κάτι που αντιμετωπίζει όπως επίσης ανιχνεύει και την κίνηση. Μπορεί να δείξει την απόσταση σε εκατοστά αλλά και σε ίντσες. Η μέγιστη απόσταση που μπορεί να μετρήσει είναι 0 έως 2,5 μέτρα με ακρίβεια 3 εκατοστών. Ο υπερηχητικός αισθητήρας λειτουργεί με την αποστολή υπερηχητικών κυμάτων που ανακλώνται από ένα αντικείμενο μπροστά του και στη συνέχεια πίσω δηλαδή μετρά την απόσταση από τον υπολογισμό του χρόνου που χρειάζεται για ένα ηχητικό κύμα για να χτυπήσει ένα αντικείμενο και να επιστρέψει, ακριβώς σαν ηχώ. Αυτός ο αισθητήρας είναι ακριβής μόνο στην ανίχνευση επίπεδων επιφανειών άρα αντικείμενα μεγάλου μεγέθους με σκληρές επιφάνειες παρέχουν τις καλύτερες αναγνώσεις[7].

Εικόνα 2.4.3.4 Αισθητήρας υπερήχων

2.3.4 Λοιπά μέρη του εκπαιδευτικού πακέτου

Το εκπαιδευτικό πακέτο πέρα από τους κινητήρες και τους αισθητήρες περιέχει και διάφορα άλλα εξαρτήματα για την διευκόλυνση του χρήστη στη δημιουργία διαφόρων εφαρμογών και κατασκευών. Η λίστα των εξαρτημάτων αυτών είναι αρκετά μεγάλη και όλα τα μέρη είναι έτσι κατασκευασμένα ώστε να συναρμολογούνται πολύ εύκολα μεταξύ τους και σχεδιασμένα με τον γνωστό σχέδιο «τούβλου» της **Lego**. Άξονες, γρανάζια, ασφάλειες ακόμα και ρόδες ή και ερπύστριες υπάρχουν μέσα στο εκπαιδευτικό πακέτο της **Lego** και υπάρχουν αναλυτικά σχέδια με τα οποία μπορεί ο καθένας να δημιουργήσει το δικό του ρομπότ.



Εικόνα 2.3.4 Εξαρτήματα του πακέτου Lego Mindstorms NXT

Κεφάλαιο 3

Κατασκευή εκπαιδευτικής διάταξης με χρήση του κινητήρα «NXT» και της υπολογιστικής πλατφόρμας Arduino

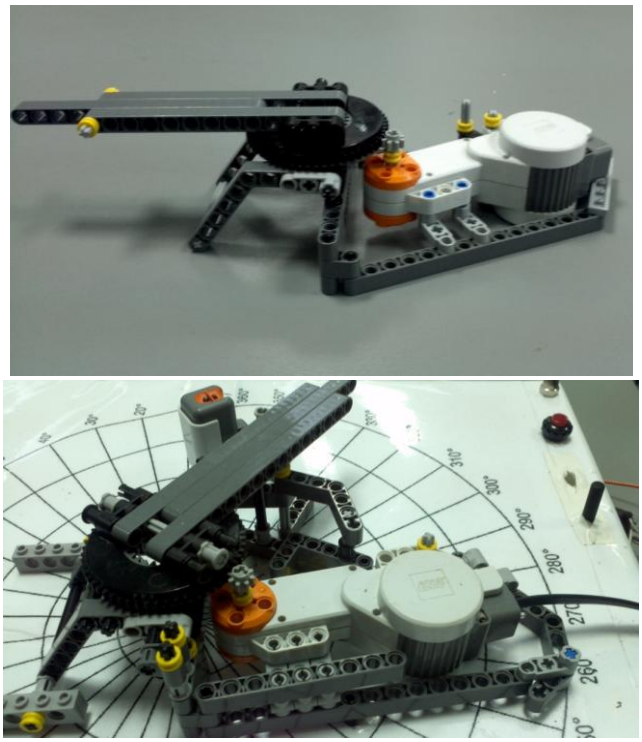
3.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται ο τρόπος σχεδιασμού και κατασκευής της μια εκ των τριών κατασκευών, συγκεκριμένα της διάταξης με χρήση του κινητήρα **«NXT»** και της **υπολογιστικής πλατφόρμας Arduino**. Στις τρείς αυτές κατασκευές χρησιμοποιήθηκε η ίδια βάση στήριξης με μικρές μετατροπές και τα ίδια παρελκόμενα, ώστε να γίνει πιο εύκολη η σύγκριση μεταξύ των τριών ηλεκτρικών κινητήρων. Επίσης αναλύεται ο **ελεκτής PID** και παρουσιάζεται η πειραματική διαδικασία που ακολουθήθηκε και τα συμπεράσματα τα οποία καταλήξαμε. Σκοπός είναι η χρησιμοποίηση τους ως εκπαιδευτικό αντικείμενο στα εργαστήρια του τμήματος.

3.2 Παρουσίαση της εκπαιδευτικής διάταξης

Η βασική κατασκευή αποτελείται εξ ολοκλήρου από μέρη του εκπαιδευτικού πακέτου της **Lego** με λιτό σχεδιασμό ώστε να είναι εύκολο και από έναν μη εξοικειωμένο χρήστη να δημιουργήσει μια παρόμοια διάταξη και να αντιληφθεί την λειτουργία των ηλεκτρικών

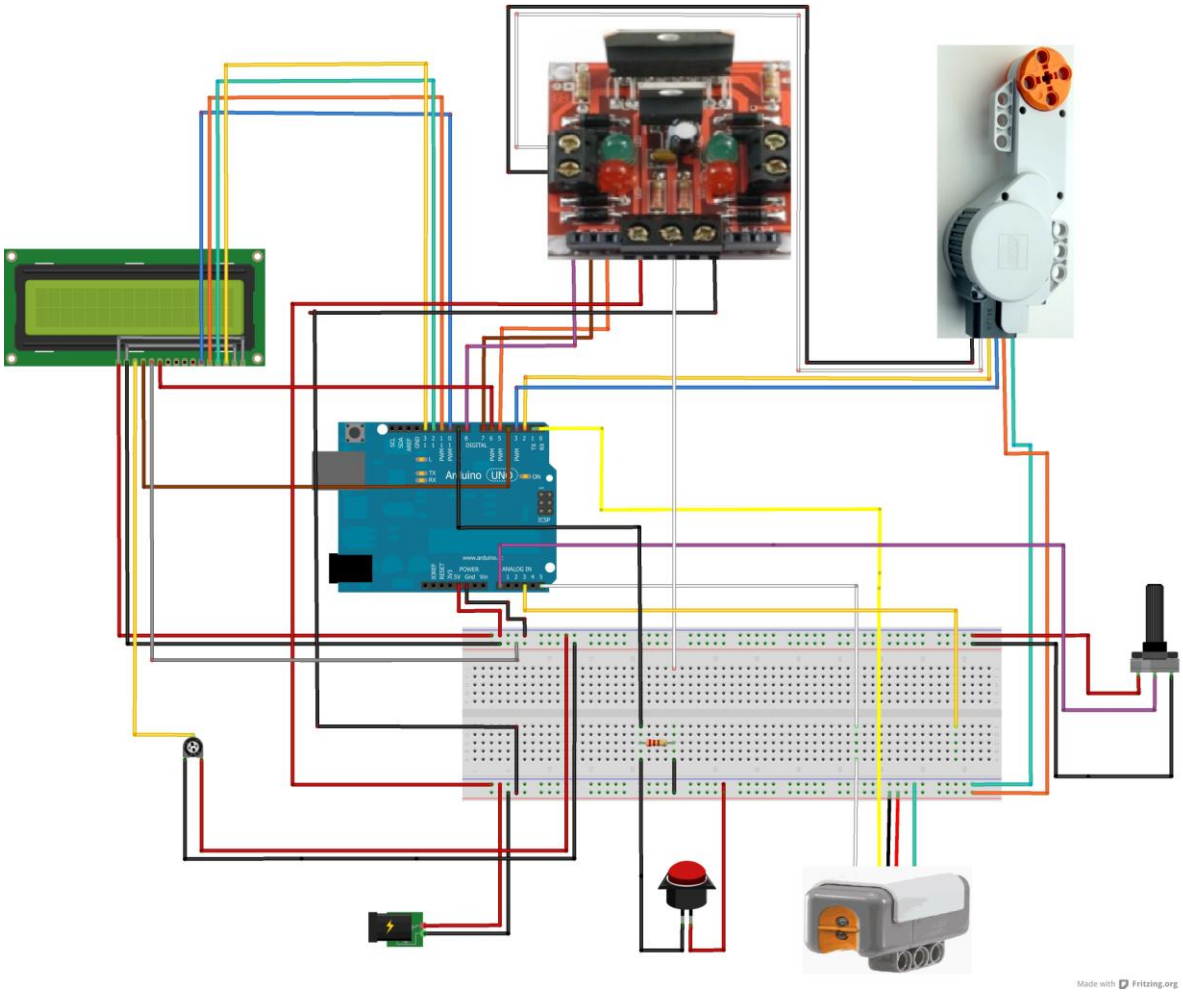
κινητήρων σε συνδυασμό με τον έλεγχο τους, μέσω ενός μικροελεγκτή και να πειραματιστεί με μοναδικές μηχανικές δημιουργίες, μέσω του πακέτου **Lego Mindstorms NXT**. Η διάταξη αποτελείται από τον κινητήρα NXT, την βάση στήριξης του, η οποία είναι κατασκευασμένη από μέρη του εκπαιδευτικού πακέτου Lego Mindstorms (ράγες, άξονες, ασφάλειες κ.α) και διατηρεί τον κινητήρα NXT σε σταθερή θέση. Η βάση στήριξης του κινητήρα NXT τοποθετείται σε μια κεντρική βάση στήριξης όλης της διάταξης, ώστε να επιτυγχάνεται σταθερότητα όλης της κατασκευής. Για τον έλεγχο της διάταξης χρησιμοποιείται ο μικροελεγκτής **Arduino Uno** και για την οδήγηση του **κινητήρα NXT** μια **Η-Γέφυρα** (H-Bridge). Για την επικοινωνία του χρήστη με την διάταξη χρησιμοποιήσαμε ένα μπουτόν και ένα αναλογικό ποτενσιόμετρο, με τα οποία ο χρήστης μπορεί να δίνει την επιθυμητή τιμή της γωνίας, να την εναλλάσσει σε πραγματικό χρόνο και να βλέπει τα αποτελέσματα αυτά σε μια οθόνη **«LCD»** μαζί με άλλες χρήσιμες πληροφορίες. Τέλος χρησιμοποιήθηκε ο αισθητήρας φωτός του εκπαιδευτικού πακέτου **Lego Mindstorms** για το «calibration» του κινητήρα NXT για το οποίο θα αναφερθούμε αναλυτικά παρακάτω.



Εικόνα 3.2.1 Αρχική (πάνω) και τελική (κάτω) όψη της κατασκευής

Η βασική εφαρμογή των κατασκευών είναι να δίνει ο χρήστης μια τιμή σε μοίρες (0-360) μέσω ενός ποτενσιόμετρου και με το πάτημα ενός μπουτόν ο βραχίονας να κινείται στην επιθυμητή τιμή, έτσι δημιουργήσαμε μια κατασκευή, όπου αναπαριστάται η λειτουργία ενός ρομποτικού βραχίονα ενός βαθμού ελευθερίας και απεικονίσαμε στην βάση του της μοίρες τις οποίες κινείται (0-360). Κάθε αλλαγή στη θέση του βραχίονα (σε

μοίρες) απεικονίζεται σε μια οθόνη LCD σε πραγματικό χρόνο όπως επίσης απεικονίζεται και η επιθυμητή τιμή που ορίζει ο χρήστης.



Εικόνα 3.2.2 Συγκεντρωτική ηλεκτρική διάταξη της κατασκευής

3.3 Βάσεις στήριξης

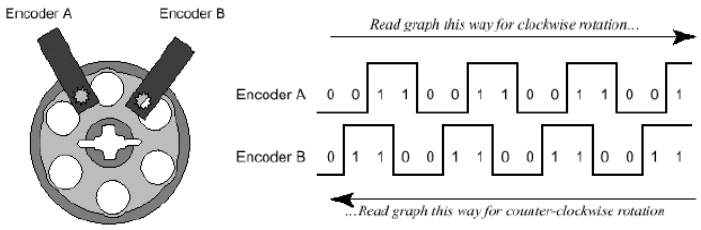
Η **βάση στήριξης του κινητήρα NXT** σχεδιάστηκε έτσι ώστε ο κινητήρας να παραμένει σταθερός όταν μεταδίδει κίνηση στο βραχίονα, και τα γρανάζια που μεταδίδουν την κίνηση να συμπλέκονται παράλληλα. Η **βάση στήριξης του κινητήρα NXT** αποτελείται από ράγες άξονες και ασφάλειες του εκπαιδευτικού πακέτου Lego Mindstorms. Η **βάση στήριξης της διάταξης** αποτελείται από ελαφρύ και λεπτό υλικό. Επιλέξαμε το υλικό της βάσης να είναι «PVC» και φέρνοντας το στις κατάλληλες διαστάσεις δημιουργήσαμε μια κυκλική κλίμακα μοιρών (0 έως 360) ώστε να είναι πιο εύκολη η απεικόνιση των κινήσεων του βραχίονα με «γυμνό» μάτι. Δημιουργώντας τέσσερις τρύπες περιμετρικά της βάσης προσθέσαμε τα «πόδια», τα οποία υπερυψώνουν τη βάση ελαφρά ώστε να περνάν κάποια καλώδια κάτω

από τη βάση και κάποιες πιο μικρές τρύπες για τον ίδιο σκοπό. Επίσης δημιουργήσαμε άλλες τρείς τρύπες με συγκεκριμένη διάμετρο ώστε να «κουμπώνουν» δύο ποτενσιόμετρα και ένα μπουτόν με τα οποία θα λειτουργεί ο χρήστης την διάταξη. Η βάση στήριξης της διάταξης δημιουργήθηκε με τέτοιο τρόπο ώστε αν πάσα στιγμή να μπορούμε να συναρμολογήσουμε πάνω της έναν από τους τρεις κινητήρες (DC, Rc Servo, Stepper) θέλουμε, χωρίς να επηρεάζονται τα υπόλοιπα περιφερειακά μέρη της κατασκευής (οθόνη, ποτενσιόμετρα, μπουτόν κ.τ.λ.)

Εικόνα 3.3 Η βάση στήριξης της κατασκευής, μόνη και μαζί με τον κινητήρα "NXT"

3.4 Ο κινητήρας «NXT» της Lego

Ο κινητήρας **NXT** της Lego έχει ενσωματωμένο περιστροφικό κωδικοποιητή («rotary encoder») δύο καναλιών που αποτελείται από ένα στρεφόμενο δίσκο που είναι διαιρεμένος σε τομείς οι οποίοι είναι εναλλάξ διαφανείς και μη-διαφανείς. Στη μια πλευρά του δίσκου υπάρχει μια φωτεινή πηγή και στην άλλη ένα φωτοκύτταρο. Όταν ο δίσκος περιστρέφεται, κάθε μεταβολή του φωτός που προσπίπτει στο φωτοκύτταρο παράγει ένα παλμό εξόδου. Το πλήθος των παλμών αυτών ανά μονάδα χρόνου είναι ανάλογο προς τη γωνιακή ταχύτητα του άξονα και το ολικό πλήθος παλμών σε κάθε χρονική στιγμή είναι ανάλογο προς την ολική γωνιακή μετατόπιση του μετρούμενου άξονα. Ο κωδικοποιητής έχει δύο φωτοκύτταρα ανίχνευσης στον ίδιο δίσκο, τα οποία είναι τοποθετημένα κατά τέτοιο τρόπο, ώστε οι έξοδοι να έχουν διαφορά φάσης 90° η μία, σε σχέση με την άλλη. Η φορά περιστροφής προσδιορίζεται με τη βοήθεια κατάλληλου λογικού κυκλώματος, το οποίο δέχεται σαν εισόδους τις δυο ακολουθίες παλμών, όπως φαίνεται και στο παρακάτω σχήμα (Εικ.3.4.1).

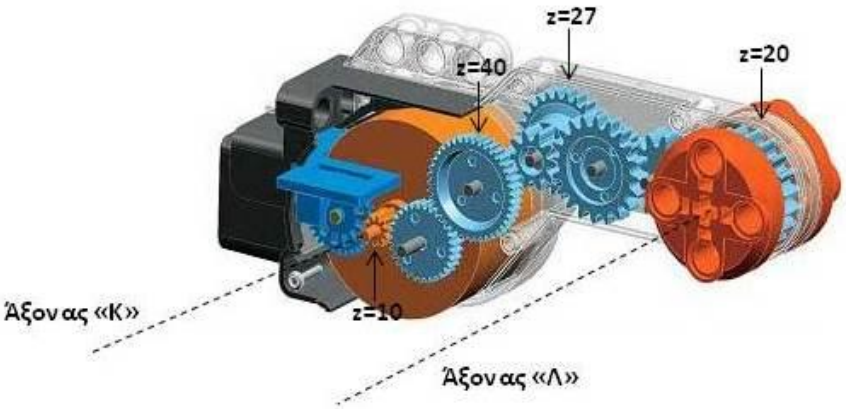


Εικόνα 3.4.1 Απεικόνιση των καναλιών (A, B) του περιστροφικού κωδικοποιητή του κινητήρα NXT

Από την έξοδο του κινητήρα η **σχέση μετάδοσης**, δηλαδή ο λόγος των διαμέτρων συνεργαζόμενων γραναζιών εσωτερικά του κινητήρα μπορεί να βρεθεί με βάση την εικόνα (Εικ.3.3.2.3) και τον παρακάτω πίνακα ο οποίος περιγράφει την σχέση των γραναζιών.

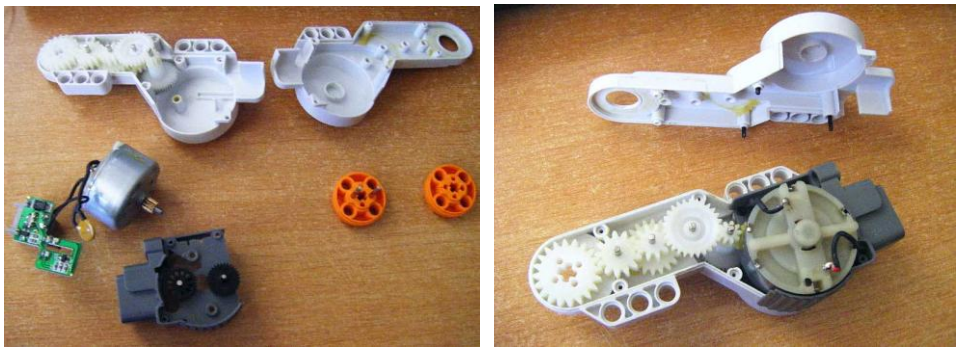
```
10:30:40             = 1:4
       9:27          = 1:3
         10:20       = 1:2
            10:13:20 = 1:2
Overall               1:48
```

Πίνακας 3.4.2



Εικόνα 3.4.3 Εσωτερική όψη του κινητήρα "NXT"

Προκύπτει από τον παραπάνω πίνακα (Πίν.3.4.2) ότι η **σχέση μετάδοσης** είναι 1:48 δηλαδή για μια περιστροφή του άξονα «Λ» (hub) ο άξονας «Κ» πρέπει να κάνει 48 περιστροφές. Η **σχέση μετάδοσης** μεταξύ του γραναζιού στο οποίο βρίσκεται ο κωδικοποιητής (z=32) και του γραναζιού(z=10) που βρίσκεται πάνω στον άξονα «Κ» είναι 10:32, άρα για μία περιστροφή του άξονα «Λ»(hub) ο κωδικοποιητής κάνει 48*10:32=15 περιστροφές.

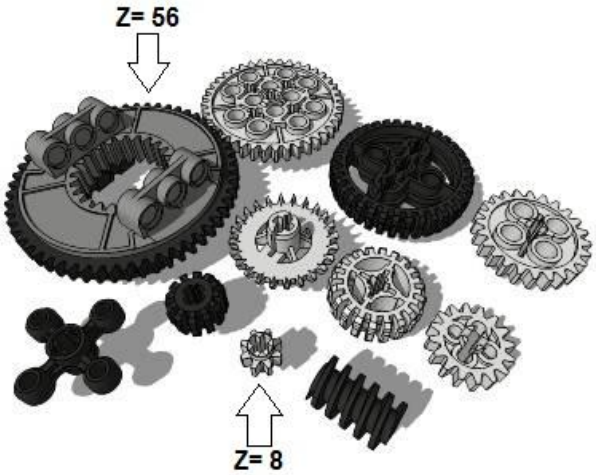


Εικόνα 3.4.4 Αποσυναρμολογημένη όψη του κινητήρα NXT

Ο κωδικοποιητής του **NXT** έχει 12 σχισμές, οπότε οι οπτικοί ανιχνευτές «βλέπουν» 12*15=180 παλμούς, χρησιμοποιώντας και τα δύο κανάλια του κωδικοποιητή έχουμε 360 παλμούς ανά περιστροφή και καθώς ο κωδικοποιητής του **NXT** είναι τετραπεριοδικής διεύθυνσης («quadrature encoder») οι μέγιστοι παλμοί που μπορεί να παράγει ο κινητήρας είναι 720 παλμοί ανά περιστροφή. Χρησιμοποιώντας το γρανάζι των 8 δοντιών στην έξοδο του κινητήρα (άξονας «Λ»(hub)) του εκπαιδευτικού πακέτου Lego, και στον βραχίονα αυτό των 56 δοντιών, η μεταξύ τους σχέση μετάδοσης είναι η εξής:

$$i = \frac{z_1}{z_2} = \frac{8}{56} = \frac{1}{7} = 0.14$$

*όπου **i** η σχέση μετάδοσης των γραναζιών, **z1** ο αριθμός δοντιών του κινητήριου τροχού και **z2** ο αριθμός δοντιών του κινούμενου τροχού*[14].



Εικόνα 3.4.5 Τα γρανάζια του εκπαιδευτικού πακέτου Lego Mindstorms NXT

Όπως και όλα τα υπόλοιπα μέρη του πακέτου **Lego Mindstorms NXT**, ο κινητήρας χρησιμοποιεί για την σύνδεση του το καλώδιο «RJ12», το οποίο είναι ένα διεθνές πρότυπο σύνδεσης σταθερών τηλεφώνων και άλλων παρόμοιων εφαρμογών. Έτσι για την σύνδεση του με την **υπολογιστική πλατφόρμα Arduino Uno** χρειάστηκε να αφαιρέσουμε την μια άκρη του καλωδίου ώστε να συνδέσουμε το κάθε καλώδιο στις θύρες (pin) του Arduino. Την ίδια διαδικασία ακολουθήσαμε σε όλα τα μέρη του πακέτου της Lego που χρησιμοποιούν το συγκεκριμένο καλώδιο(«RJ12»). Παραθέτουμε παρακάτω τον πίνακα με τις πληροφορίες για το καλώδιο «RJ12»[12].

Επεξήγηση των θυρών(Pin) του καλωδίου «RJ12»				
Θύρες/ Pin	**Ονομασία**	**Χαρακτηριστικά καλωδίου**	**Χρώμα καλωδίου**	**Αρίθμηση Θυρών/Pin**
1	ANALOG	+9V Τροφοδοσία	Άσπρο	
2	GND	Γείωση	Μαύρο	
3	GND	Γείωση	Κόκκινο	
4	IPOWERA	+5V Τροφοδοσία	Πράσινο	*Εικόνα 3.4.6 Καλώδιο «RJ12»*
5	DIGIAI0	Κανάλι Α οπτικού κωδικοποιητή	Κίτρινο	
6	DIGIAI1	Κανάλι Β οπτικού κωδικοποιητή	Μπλε	

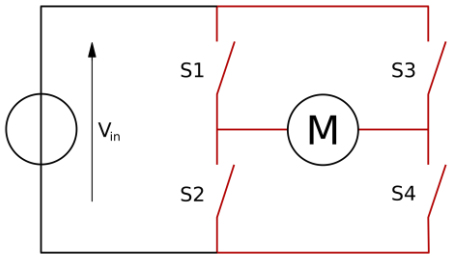
Όπως φαίνεται στη παραπάνω πίνακα η συνδεσμολογία για το καλώδιο «RJ12» είναι η εξής:

- Το **άσπρο** καλώδιο(Pin 1) είναι η τροφοδοσία (9V)

- Το **μαύρο** καλώδιο(Pin 2) η γείωση
- Το **κόκκινο** καλώδιο(Pin 3) η γείωση του αισθητηρίου(εάν υπάρχει)
- Το **πράσινο** καλώδιο(Pin 4) η τροφοδοσία (4.3V) του αισθητηρίου(εάν υπάρχει)
- Το **κίτρινο** καλώδιο(Pin 5) το κανάλι (B)
- Το **μπλε** καλώδιο(Pin 6) το κανάλι (A)

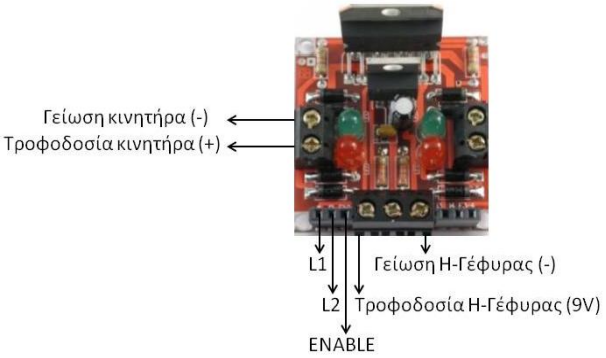
3.5.1 Σύνδεση της «Η-Γέφυρας» (H-Bridge)

Ο κινητήρας **NXT** λειτουργεί με μέγιστη τάση 9 Volt ενώ η υπολογιστική πλατφόρμα Arduino Uno μπορεί να τροφοδοτήσει μέχρι 5 Volt οπότε για την σύνδεση τους χρησιμοποιήσαμε μια **Η-Γέφυρα (H-bridge)** η οποία είναι ένα ηλεκτρονικό κύκλωμα που επιτρέπει σε μια τάση να εφαρμοστεί σε ένα φορτίο. Η **Η-γέφυρα** χρησιμοποιείται συχνά στην ρομποτική και σε άλλες εφαρμογές και επιτρέπει στους κινητήρες συνεχούς ρεύματος να κινηθούν προς τα εμπρός και προς τα πίσω ανάλογα με το σήμα που θα της δοθεί. Η γενική λειτουργία της είναι απλή καθώς αποτελείται από τέσσερις διακόπτες S1,S2,S3,S4 όταν οι διακόπτες S1 και S4 είναι κλειστές (και S2 και S3 είναι ανοικτές) μια θετική τάση θα εφαρμοστεί στον κινητήρα, με το άνοιγμα του S1 και S4 διακόπτη και κλείνοντας τον S2 και S3 διακόπτη, αυτή η τάση αντιστρέφεται, επιτρέποντας ανάστροφη λειτουργία του κινητήρα. Επίσης οι διακόπτες S1 και S2 δεν πρέπει ποτέ να είναι κλειστοί κατά την ίδια στιγμή, καθώς αυτό θα προκαλέσει βραχυκύκλωμα στην πηγή τάσης εισόδου, το ίδιο ισχύει και για τους S3 και S4.



Εικόνα 3.5.1 Ηλεκτρική διάταξη «Η-Γέφυρας»

Για την συγκεκριμένη πτυχιακή εργασία χρησιμοποιήσαμε μια **Η-Γέφυρα** της «Solarbotics», η οποία περιέχει το ολοκληρωμένο κύκλωμα L298. Η συγκεκριμένη **Η-Γέφυρα** μπορεί να οδηγήσει μέχρι και δύο κινητήρες, λειτουργεί από 6 Volt έως 26 Volt, με συνολικό ρεύμα οδήγησης 2 Ampere ο κάθε κινητήρας .



Εικόνα 3.5.2 « Η- Γέφυρα» Solarbotics L298

Η λειτουργία της συγκεκριμένης **Η-Γέφυρας** βασίζεται στον παρακάτω πίνακα αληθείας όπου ανάλογα με το σήμα που λαμβάνει στις θύρες (PIN) Enable, L1, L2, L3 και L4, (HIGH ή LOW) δίνει το ακόλουθο αποτέλεσμα, επίσης στην είσοδο Enable η Η-Γέφυρα μπορεί να οδηγηθεί και με σήμα P.W.M.

Logic Table

ENABLE	L1	L2	Result
L	L	L	OFF
L	L	H	OFF
L	H	L	OFF
L	H	H	OFF
H	L	L	BRAKE
H	L	H	FORWARD
H	H	L	BACKWARD
H	H	H	BRAKE
PWM	L	L	PULSE-BRK
PWM	L	H	FWD-SPD
PWM	H	L	BCK-SPD
PWM	H	H	PULSE-BRK

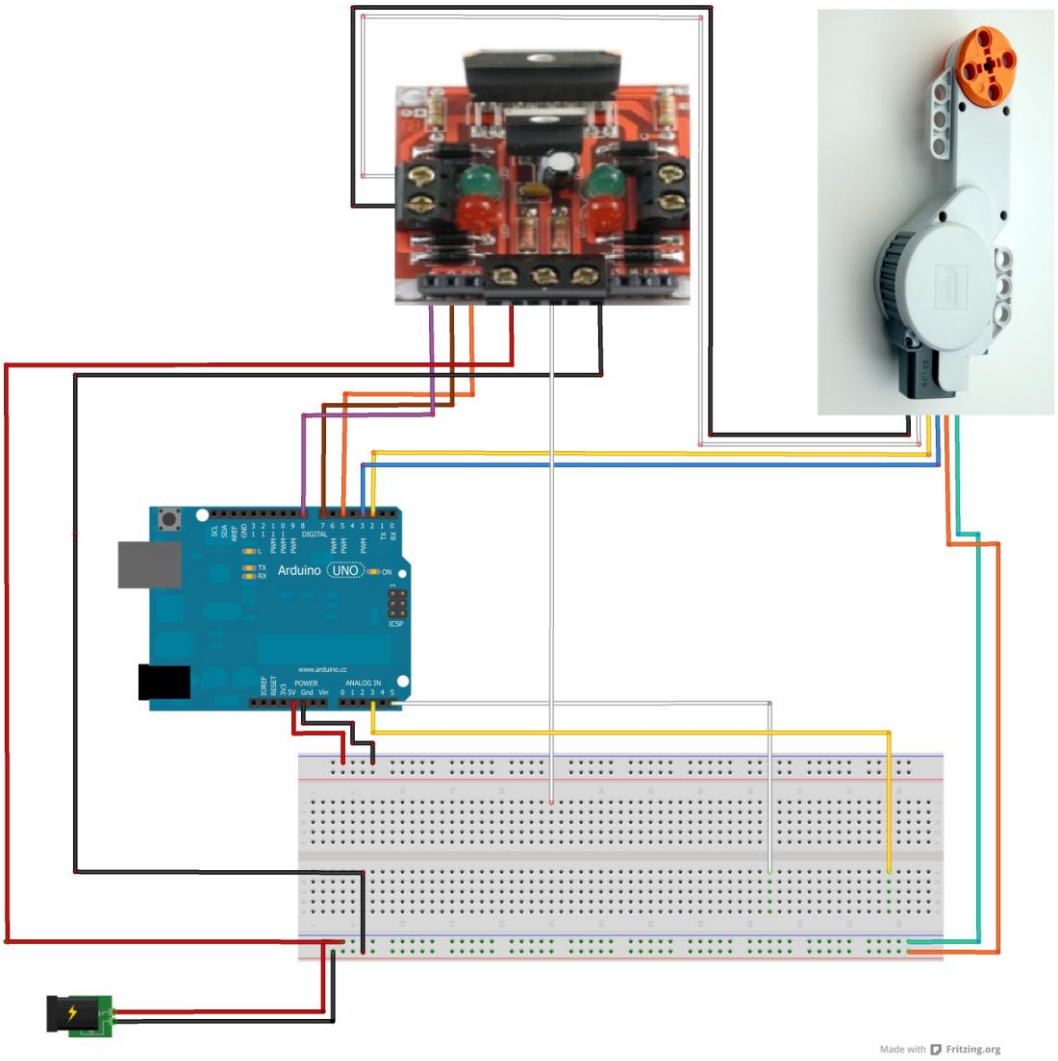
ENABLE	L3	L4	RESULT
L	L	L	OFF
L	L	H	OFF
L	H	L	OFF
L	H	H	OFF
H	L	L	BRAKE
H	L	H	FORWARD
H	H	L	BACKWARD
H	H	H	BRAKE
PWM	L	L	PULSE-BRK
PWM	L	H	FWD-SPD
PWM	H	L	BCK-SPD
PWM	H	H	PULSE-BRK

Εικόνα 3.5.3 Πίνακας αληθείας Solarbotics L298

3.5.2 Παράδειγμα χρήσης της Η-Γέφυρας σε κώδικα

Σύμφωνα με το παραπάνω πίνακα αληθείας παραθέτουμε ένα απλό παράδειγμα χρήσης της **Η-Γέφυρας** σε κώδικα ώστε να γίνει κατανοητή η χρήση της για την κίνηση ενός κινητήρα συγκεκριμένα του κινητήρα **NXT**. Ο κινητήρας συνδέεται στους ακροδέκτες της Η-Γέφυρας όπως φαίνεται στην Εικόνα 3.3.3.2, και η **Η-Γέφυρα** ελέγχεται από τον μικροελεγκτή από τις θύρες (Pin) L1, L2, και Enable. Με το συγκεκριμένο πρόγραμμα ο κινητήρας θα κινείται συνεχόμενα μπροστά (FORWARD) καθώς το PIN 5 (Enable) το ορίζουμε HIGH, το PIN 7 (L1) το ορίζουμε LOW και το PIN 8 το ορίζουμε HIGH.

```
int motorPin0 = 7; //δήλωση της θύρας που είναι συνδεδεμένο το
                   // L1 της γέφυρας
int motorPin1 = 8; //δήλωση της θύρας που είναι συνδεδεμένο το
                   // L2 της γέφυρας
int motorPin2 = 5; //δήλωση της θύρας που είναι συνδεδεμένο το
                   // ENABLE της γέφυρας
void setup() {
pinMode(motorPin0, OUTPUT);
pinMode(motorPin1, OUTPUT);
pinMode(motorPin2, OUTPUT); }
Void loop (){
digitalWrite(motorPin0,LOW);//δήλωση του PIN 7 του κινητήρα ως LOW
digitalWrite(motorPin1,HIGH);//δήλωση του PIN 8 του κινητήρα ως HIGH
digitalWrite(motorPin2,HIGH);//δήλωση του PIN 5 του κινητήρα ως HIGH
            }
```



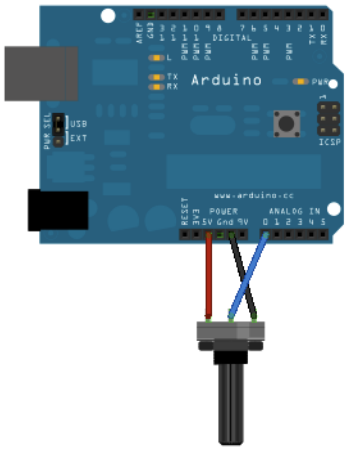
Εικόνα 3.5.4 Συνδεσμολογία της «Η-Γέφυρας» με τον κινητήρα NXT και τον Arduino Uno

3.6 Σύνδεση του αναλογικού περιστροφικού ποτενσιόμετρου

Για να δώσουμε την επιθυμητή τιμή της γωνίας χρησιμοποιήσαμε ένα **αναλογικό περιστροφικό ποτενσιόμετρο** που είναι μια συσκευή μεταβλητής αντίστασης που χρησιμοποιείται για την μέτρηση της γωνιακής θέσης (προσανατολισμού). Το ποτενσιόμετρο αποτελείται από αγώγιμη πλάκα σχήματος Ω, πάνω στην οποία γυρίζει ένας δρομέας (wiper) που βρίσκεται σε επαφή με ένα ωμικό στοιχείο και καθώς μετακινείται η αντίσταση μεταξύ του δρομέα και των άκρων μεταβάλλεται ανάλογα με τη γωνιακή μετατόπιση.

Εικόνα 3.6.1 Αναλογικό περιστροφικό ποτενσιόμετρο

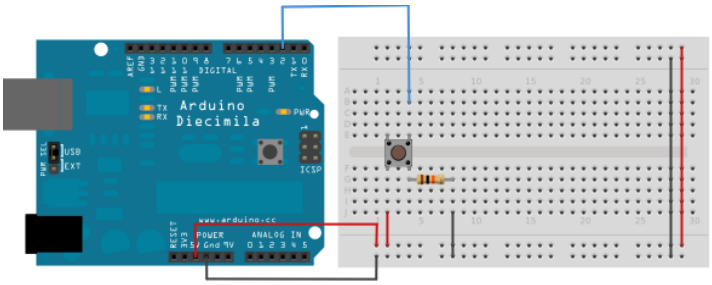
Το **ποτενσιόμετρο** έχει τρεις ακίδες (pin) και για την σύνδεση του με την πλατφόρμα του Arduino αρκεί να συνδέσουμε τη μεσαία ακίδα με μια αναλογική θύρα του Arduino και τα άλλα δυο άκρα σε μια γείωση και μια τροφοδοσία ρεύματος[1].



Εικόνα 3.6.2 Συνδεσμολογία αναλογικού περιστροφικού ποτενσιόμετρου με τον Arduino Uno

3.7 Σύνδεση του μπουτόν

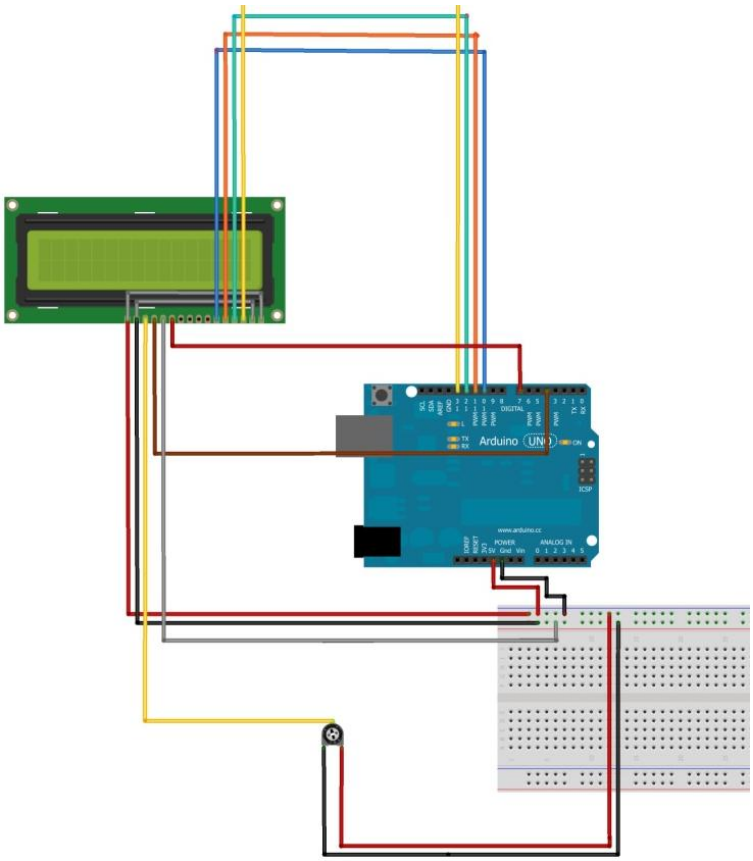
Το **μπουτόν** λειτουργεί ως διακόπτης και έχει σημεία, τα οποία συνδέονται με το κύκλωμα και ονομάζονται ακροδέκτες. Έχει δύο καταστάσεις, την κατάσταση που είναι κλειστό και την κατάσταση που είναι ανοιχτό. Όταν ένα **μπουτόν** είναι ανοιχτό δεν επιτρέπει τη διέλευση ηλεκτρικού ρεύματος μεταξύ των ακροδεκτών του, ενώ όταν είναι κλειστό την επιτρέπει[1].



Εικόνα 3.7 Συνδεσμολογία ενός μπουτόν με τον Arduino Uno

3.8 Σύνδεση της οθόνης τύπου LCD

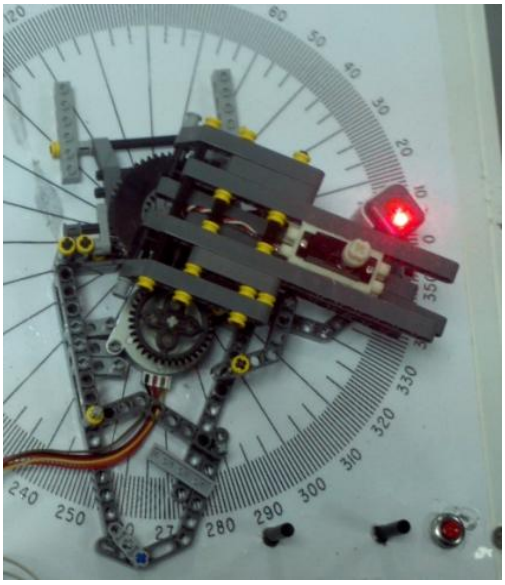
Οι οθόνες τύπου **LCD** διαθέτουν παράλληλη διασύνδεση, πράγμα που σημαίνει ότι ο μικροελεγκτής πρέπει να χειριστεί διάφορα PIN (θύρες) ταυτόχρονα για να ελέγξει την οθόνη. Για την εύρυθμη λειτουργία της LCD έχει δημιουργηθεί από την ομάδα του Arduino μια βιβλιοθήκη στο κάλεσμα τις οποίας επιτρέπει σε μια υπολογιστική πλατφόρμα Arduino τον έλεγχο οποιασδήποτε οθόνης (LCD) η οποία είναι συμβατή με την «Hitachi HD44780». Η βιβλιοθήκη λειτουργεί με είτε 4 -bit ή 8-bit (δηλ. χρησιμοποιώντας 4 ή 8 γραμμές δεδομένων). Παρακάτω δείχνουμε την συνδεσμολογία της οθόνης στη τελική διάταξη[1].



Εικόνα 3.8 Συνδεσμολογία οθόνης LCD με την υπολογιστική πλατφόρμα Arduino Uno

3.9 Ο αισθητήρας φωτός

Ο **αισθητήρας φωτός** είναι ένας αισθητήρας που δουλεύει με τάση 5 Volt, όπως έχει προαναφερθεί στο Κεφάλαιο 2 (2.3.3.3). Εάν χρησιμοποιηθεί από το λογισμικό της **Lego** διαβάζει μια κλίμακα από το 0 έως το 100, όσο πιο κοντά στο μηδέν βρίσκεται η τιμή τόσο πιο σκοτεινό είναι το οπτικό περιβάλλον του αισθητήρα, συνδέοντας τον όμως με την υπολογιστική πλατφόρμα **Arduino** το αναλογικό σήμα που διαβάζει το **Arduino** είναι μια τάση η οποία κυμαίνεται από 0 έως 1023, όσο πιο κοντά στο μηδέν βρίσκεται η τιμή τόσο πιο σκοτεινό ενώ όσο πιο κοντά στο 1023 τόσο πιο φωτεινό είναι το οπτικό περιβάλλον του αισθητήρα. Βάση των παλμών μπορούμε να κινήσουμε τον κινητήρα σε οποία θέση θέλουμε αρκεί να γνωρίζουμε την αρχική του θέση, για αυτό και πριν από οποιαδήποτε μετάβαση της αρχικής θέσης του κινητήρα κάνουμε έναν έλεγχο («calibration») ώστε ο κινητήρας να βρίσκεται στη «θέση μηδέν».



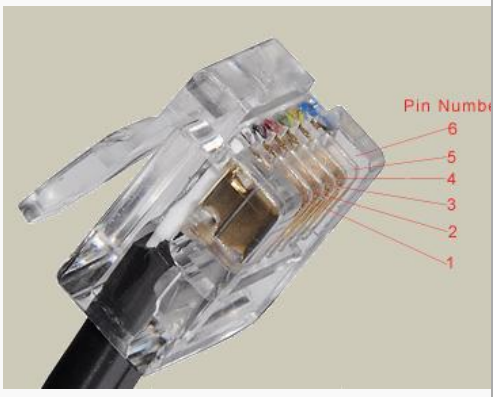
Εικόνα 3.9.1 Ο αισθητήρας φωτός σε λειτουργία

Για να βρίσκεται ο «βραχίονας» στη θέση μηδέν χρησιμοποιήσαμε τον αισθητήρα φωτός του εκπαιδευτικού πακέτου της πλατφόρμας **Lego Mindstorms NXT** έτσι ώστε όταν ο βραχίονας βρεθεί ακριβώς πάνω από τον αισθητήρα φωτός ο κινητήρας να σταματήσει θέτοντας το σημείο αυτό, αφού ο αισθητήρας φωτός βρίσκεται σταθερός στην κατασκευή, ως σημείο μηδέν («θέση μηδέν»). Για να το πετύχουμε αυτό μετρήσαμε της τιμές που επιστρέφει ο αισθητήρας σε συνθήκες φωτισμού δωματίου όταν βρίσκεται ο βραχίονας ακριβώς πάνω από τον αισθητήρα φωτός και όταν ο βραχίονας βρίσκεται σε διαφορετική θέση. Γνωρίζοντας τις τιμές αυτές δημιουργούμε μια συνθήκη με την οποία ο κινητήρας θα κινείται διαρκώς μέχρις ότου η συνθήκη δεν είναι αληθής, δηλαδή ο βραχίονας βρεθεί

ακριβώς πάνω από τον αισθητήρα φωτός, οπότε και θα σταματήσει («θέση μηδέν») περιμένοντας την επόμενη «εντολή» του χρήστη.

Επεξήγηση των θυρών(Pin) του καλωδίου «RJ12»

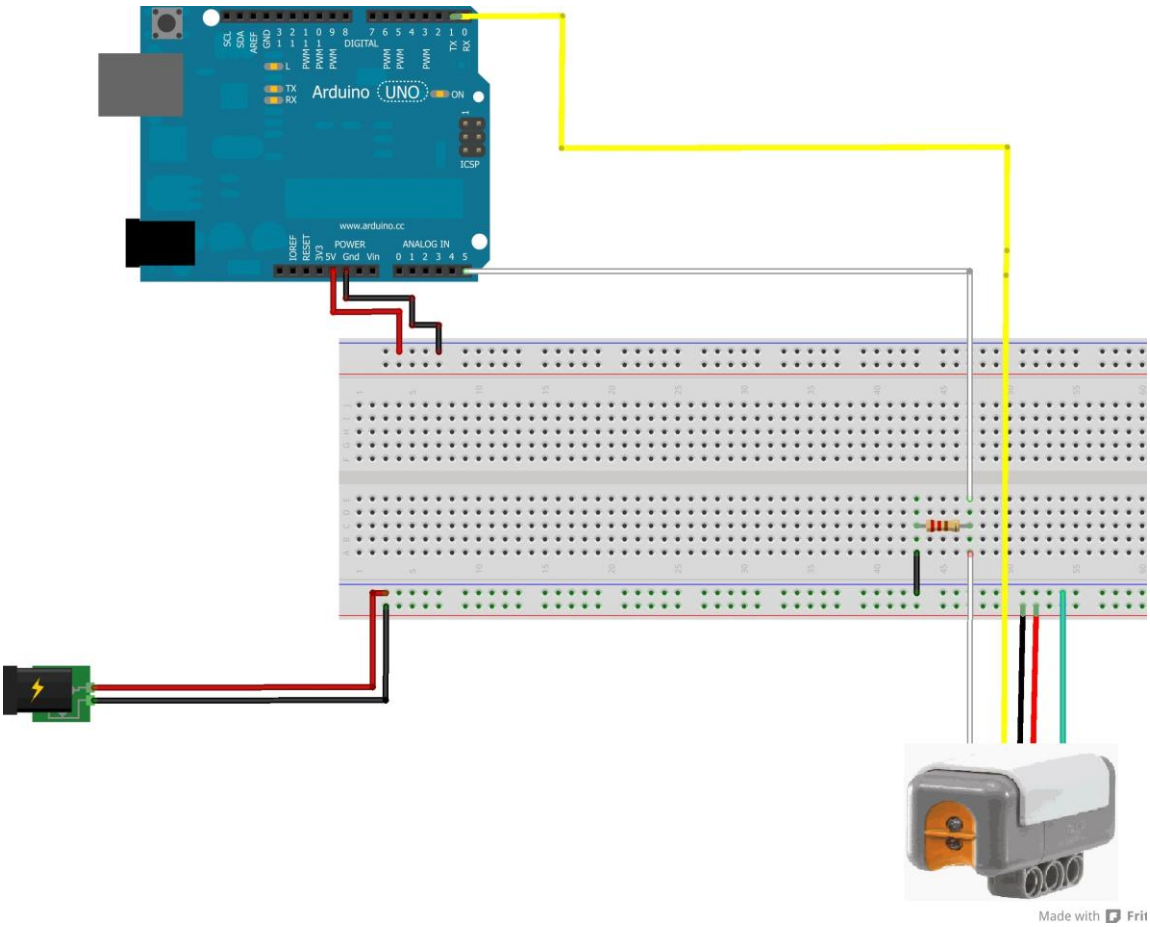
Θύρες/ Pin	Ονομασία	Χαρακτηριστικά καλωδίου	Χρώμα καλωδίου	Αρίθμηση Θυρών/Pin
1	ANALOG	Αναλογική διεπαφή, +9V Τροφοδοσία	Άσπρο	
2	GND	Γείωση	Μαύρο	
3	GND	Γείωση	Κόκκινο	
4	IPOWERA	+5V Τροφοδοσία	Πράσινο	
5	DIGIAI0	Κανάλι Α οπτικού κωδικοποιητή	Κίτρινο	
6	DIGIAI1	Κανάλι Β οπτικού κωδικοποιητή	Μπλε	



Εικόνα 3.9.2 Καλώδιο «RJ12»

Η συνδεσμολογία του **αισθητήρα φωτός** είναι αρκετά απλή όπως φαίνεται και στο παρακάτω σχήμα. Η μοναδική διαφορά με τα υπόλοιπα μέρη του εκπαιδευτικού πακέτου **Lego** είναι ότι το πρώτο Pin (λευκό καλώδιο) είναι ένα τρανζίστορ τύπου ανοικτού συλλέκτη (open-collector) οπότε χρειάζεται μια αντίσταση «pull-up» για την εύρυθμη λειτουργία του αισθητήρα. Τα pin του «RJ12» καλωδίου συνδέονται ως εξής[7.12]:

- Το **λευκό** καλώδιο(Pin 1) σε μια αναλογική θύρα του Arduino
- Το **κίτρινο** καλώδιο(Pin 5) σε μια ψηφιακή θύρα του Arduino για το φωτισμό του αντικειμένου
- Το **μαύρο** και το **κόκκινο**(Pin 2,3) καλώδιο είναι οι γειώσεις
- Το **πράσινο**(Pin 4) η τροφοδοσία του αισθητήρα



Εικόνα 3.9.3 Συνδεσμολογία της υπολογιστικής πλατφόρμας Arduino Uno με τον αισθητήρα φωτός

3.10 Σύνδεση της υπολογιστικής πλατφόρμας «Arduino Uno» και τα εξωτερικά «Interrupt».

Η υπολογιστική πλατφόρμα Arduino Uno χρησιμοποιεί μέχρι και δύο εξωτερικά interrupt, τα **INT0** και **INT1**, τα οποία εκ κατασκευής είναι ορισμένα στις θύρες (PIN) 2 και 3 αντίστοιχα. Τα εξωτερικά interrupt είναι ένα σήμα που δέχεται ο μικροελεγκτής οποιαδήποτε χρονικά στιγμή, για ένα γεγονός το οποίο χρειάζεται άμεση προσοχή και όταν ένα interrupt ενεργοποιηθεί ο μικροελεγκτής σταματά οποιαδήποτε εργασία πράττει και εκτελεί την ρουτίνα του interrupt (interrupt service routine).Μετά την εκτέλεση της ρουτίνας του interrupt μόλις αυτή τελειώσει επιστρέφει στην προηγούμενη εργασία του. Τα interrupt είναι πολύ χρήσιμα καθώς μας επιτρέπουν να εκτελούμε μια συνάρτηση άμεσα και ταχύτατα. Για να χρησιμοποιήσουμε ένα εξωτερικό interrupt αρκεί να χρησιμοποιήσουμε την συνάρτηση π.χ. attachInterrupt στην οποία πρέπει να δηλώσουμε τον αριθμό του interrupt ,την συνάρτηση του και τον τρόπο λειτουργίας του, δηλαδή **attachInterrupt(interrupt, function, mode).** Προφανώς στη θέση του interrupt επιλέγουμε ανάμεσα από το **INT0** και **INT1**, η συνάρτηση του interrupt (interrupt service routine) καλείται κάθε φορά που ενεργοποιείται το interrupt, δεν χρειάζεται κάποια παράμετρο και δεν επιστρέφει τίποτα. Στη θέση του **function** το όνομα της συνάρτησης του interrupt, ενώ όσον αφορά τον τρόπο λειτουργίας (**mode**) του interrupt υπάρχουν τέσσερις προεγκατεστημένες επιλογές οι οποίες είναι:

- **LOW** : το interrupt ενεργοποιείται όταν στη θύρα (PIN) που είναι συνδεδεμένο το interrupt υπάρξει σήμα LOW.
- **CHANGE** : το interrupt ενεργοποιείται όταν στη θύρα (PIN) που είναι συνδεδεμένο το interrupt υπάρξει αλλαγή το σήμα οποιαδήποτε και αν είναι αυτή.
- **RISING** : το interrupt ενεργοποιείται όταν έχουμε ανερχόμενο μέτωπο δηλαδή όταν στη θύρα (PIN) που είναι συνδεδεμένο το interrupt υπάρξει σήμα που μετατρέπεται από LOW σε HIGH.
- **FALLING** : το interrupt ενεργοποιείται όταν έχουμε κατερχόμενο μέτωπο δηλαδή όταν στη θύρα (PIN) που είναι συνδεδεμένο το interrupt υπάρξει σήμα που μετατρέπεται από HIGH σε LOW .

Τέλος, μέσα στη συνάρτηση του interrupt η εντολή **delay()** δεν λειτουργεί, οι τιμές της εντολής **millis()** δεν αυξάνονται και η μεταβλητές που χρησιμοποιούνται μέσα στη συνάρτηση του interrupt πρέπει να είναι μορφής volatile.

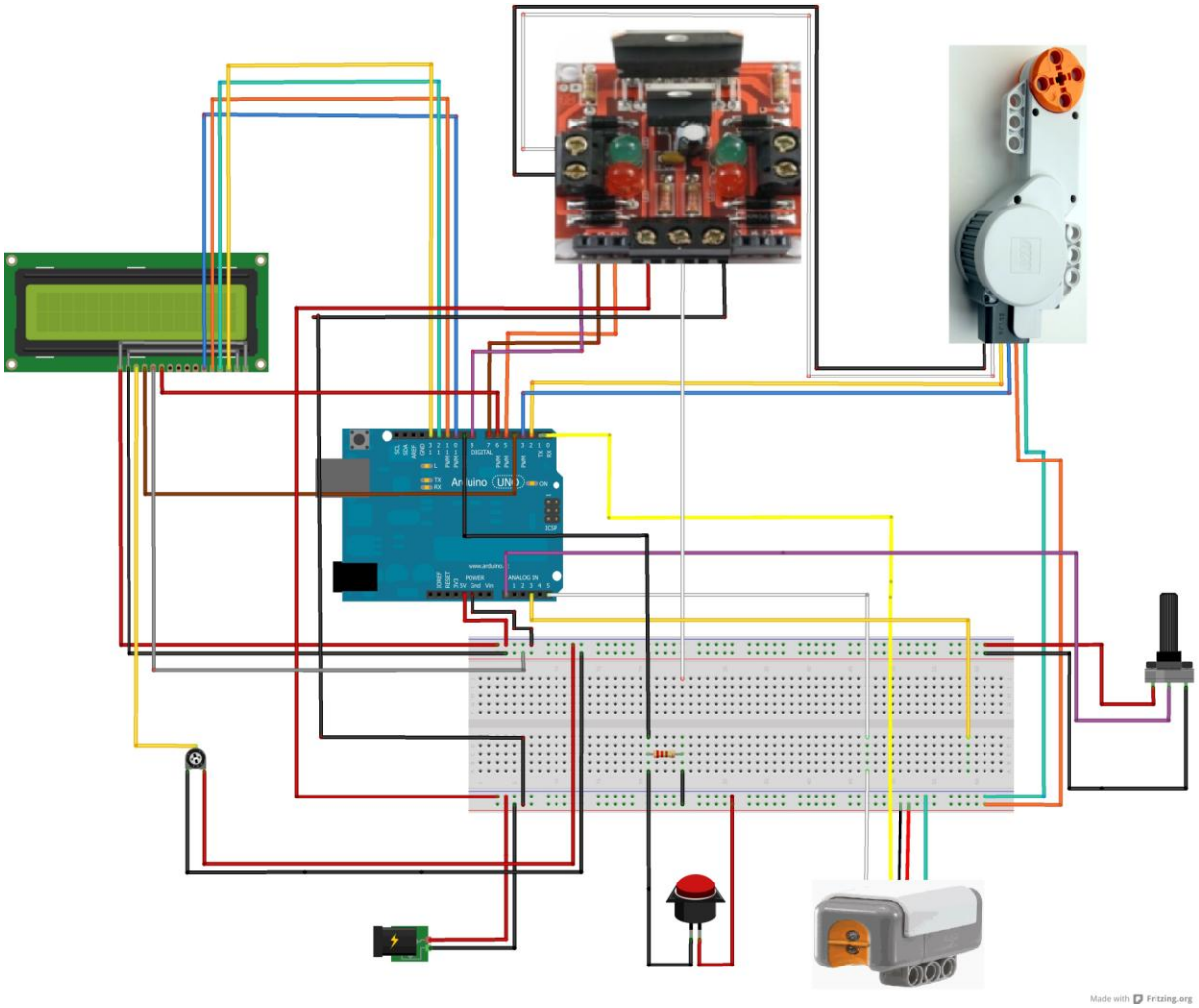
3.11 Παράδειγμα χρήσης των εξωτερικών Interrupt («Interrupt Service Routine»)

Η χρήση των εξωτερικών Interrupt («Interrupt Service Routine») γίνεται αντιληπτή με το παρακάτω παράδειγμα, με το οποίο το LED 13, που βρίσκεται πάνω στην **υπολογιστική πλατφόρμα του Arduino**, εναλλάσσεται η κατάσταση του («HIGH-LOW») κάθε φορά που ενεργοποιείται το Interrupt 0 (INT0) [1].

```
int pin = 13;//δήλωση του LED

volatile int state = LOW;//δήλωση της μεταβλητής state

void setup(){

pinMode(pin, OUTPUT);//δήλωση του LED ως έξοδο

attachInterrupt(0, blink, CHANGE);}//δήλωση της συνάρτησης του
                                   //interrupt (INT0)
void loop()

{digitalWrite(pin, state);}//εγγραφή της κατάστασης του LED

void blink()

{state = !state;}//αναστροφή της κατάστασης του LED
```

3.12 Ηλεκτρολογική σύνδεση των παραπάνω εξαρτημάτων

Βάση των προηγούμενων κεφαλαίων παρουσιάζεται παρακάτω η ηλεκτρολογική σύνδεση ολόκληρης της διάταξης, δηλαδή η σύνδεση του κινητήρα NXT της Lego με την Η-Γέφυρα, το ποτενσιόμετρο, τον αισθητήρα φωτός, την LCD οθόνη και το μπουτόν. Με αυτό τον τρόπο κινούμε τον κινητήρα σε ένα εύρος τιμών από 0 έως 360 χωρίς όμως να τον ελέγχουμε, να γνωρίζουμε δηλαδή την θέση του. Για να το επιτύχουμε αυτό κάνουμε χρήση ενός ελεκτή, συγκεκριμένα του ελεκτή «PID» υλοποιώντας ένα σύστημα κλειστού βρόγχου, που θα αναλύσουμε στο επόμενο κεφάλαιο, όπου θα παραθέσουμε και τον πλήρη κώδικα για την λειτουργία της διάταξης.



Εικόνα 3.12 Συγκεντρωτική ηλεκτρική διάταξη της κατασκευής

3.13 Ο ελεκτής «PID» και ο έλεγχος κλειστού βρόγχου με την χρήση του ελεκτή.

Παρακάτω θα αναλύσουμε την λειτουργία του **ελεκτή «PID»** και την χρησιμότητα του την σημερινή εποχή, στην μηχατρονική αλλά και γενικά στην μηχανική για τον ακριβή έλεγχο διαφόρων υπολογιστικών συστημάτων που είτε βασίζονται στην λειτουργία του ελεκτή, είτε τον χρησιμοποιούν. Ένα τυπικό σύστημα **κλειστού βρόχου ελέγχου** ταχύτητας ή θέσης χρησιμοποιείται συνήθως σε κινητήρες συνεχούς ρεύματος σε διάφορες σύγχρονες εφαρμογές και ουσιαστικά χρησιμοποιεί τη μέτρηση του σήματος και την ανατροφοδοτεί για να συγκριθεί με το σήμα αναφοράς (είσοδος - επιθυμητή έξοδος). Ο όρος κλειστός βρόγχος αναφέρεται στον βρόγχο που δημιουργείται από τη διαδρομή της ανάδρασης («feedback») και μετράει τη διαφορά ανάμεσα στην πραγματική τιμή της μεταβλητής ελέγχου και την τιμή της επιθυμητής τιμής («setpoint») και χρησιμοποιεί τη

διαφορά αυτή για να οδηγεί την πραγματική τιμή, προς το επιθυμητό αποτέλεσμα.

3.14 Λειτουργία του PID

Το σύστημα ελέγχου **PID** πήρε το όνομά του μετά από την διόρθωση τριών όρων του, των οποίων το άθροισμα αποτελεί την ζητούμενη μεταβλητή. Οι τιμές των, αναλογικός όρος (**P**) ολοκληρωτικός όρος (**I**) διαφορικός όρος (**D**), αθροίζονται για τον υπολογισμό της εξόδου του ρυθμιστή **PID**. Ορίζοντας το U(t) ως την τιμή της εξόδου η τελική μορφή του **PID** αλγορίθμου είναι:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(T)dT + K_d \frac{d}{d_t} e(t)$$

K_p : Αναλογικό κέρδος, ρυθμιζόμενη παράμετρος

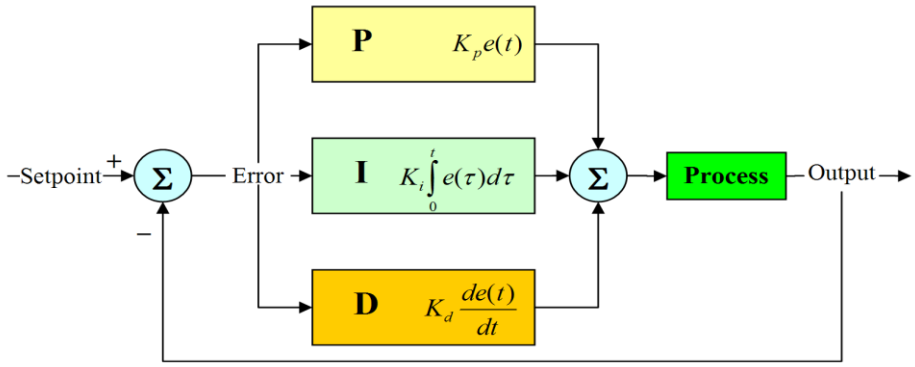
K_i : Ολοκληρωτικό κέρδος, ρυθμιζόμενη παράμετρος

K_d : Διαφορικό κέρδος, ρυθμιζόμενη παράμετρος

e : Error,σφάλμα παρακολούθησης

t : Χρόνος ή στιγμιαίος χρόνος(το παρόν)

T : Μεταβλητή ολοκληρώσεως παίρνει τιμές από το μηδέν έως το παρών t



Εικόνα 3.14 Η λειτουργία του PID γραφικά

Ο αναλογικός όρος (P)

Ο **αναλογικός όρος** («Proportional») ελαττώνει το χρόνο ανύψωσης σε πολύ μεγάλο ποσοστό αλλά πότε δεν εξαλείφει το μόνιμο σφάλμα.

$$P_{out}=K_p e(t)$$

Ο ολοκληρωτικός όρος (I)

Ο **ολοκληρωτικός όρος** («Integral») εξαλείφει το μόνιμο σφάλμα αλλά κάνει χειρότερη την μεταβατική απόκριση δηλαδή της ταλαντώσεις μέχρι την σταθεροποίηση στην επιθυμητή τιμή.

$$I_{out} = K_i \int_0^t e(T)dT$$

Ο διαφορικός όρος (D)

Ο **διαφορικός όρος** («Differential») πλεονεκτεί στο γεγονός ότι αυξάνει την σταθερότητα του συστήματος τόσο στη μείωση της υπερύψωσης όσο και στην βελτίωση της μεταβατικής απόκρισης.

$$D_{out} = K_d \frac{d}{d_t} e(t)$$

ΕΠΙΔΡΑΣΕΙΣ ΣΤΗΝ ΑΥΞΗΣΗ ΤΗΣ ΚΑΘΕ ΠΑΡΑΜΕΤΡΟΥ ΑΝΕΞΑΡΤΗΤΑ

ΠΑΡΑΜΕΤΡΟΣ	ΧΡΟΝΟΣ ΑΝΥΨΩΣΗΣ	ΥΠΕΡΥΨΩΣΗ	ΧΡΟΝΟΣ ΑΠΟΚΑΤΑΣΤΑΣΗΣ	ΜΟΝΙΜΟ ΣΦΑΛΜΑ	ΣΤΑΘΕΡΟΤΗΤΑ
K_p	ΜΕΙΩΣΗ	ΑΥΞΗΣΗ	ΜΙΚΡΗ ΑΛΛΑΓΗ	ΜΕΙΩΣΗ	ΜΕΙΩΣΗ
K_i	ΜΕΙΩΣΗ	ΑΥΞΗΣΗ	ΑΥΞΗΣΗ	ΕΞΑΛΕΙΨΗ	ΜΕΙΩΣΗ
K_d	ΜΙΚΡΗ ΑΛΛΑΓΗ	ΜΕΙΩΣΗ	ΜΕΙΩΣΗ	ΚΑΜΙΑ ΕΠΙΔΡΑΣΗ(ΣΤΗ ΘΕΩΡΙΑ)	ΒΕΛΤΙΩΣΗ ΕΑΝ ΤΟ K_d ΕΙΝΑΙ ΜΙΚΡΟ

Στο συγκεκριμένο πίνακα οι συσχετισμοί που γίνονται μεταξύ της κάθε παραμέτρου δεν γίνεται να είναι αρκετά ακριβείς καθώς ο κάθε όρος αλληλεξαρτάται από τον άλλο και έτσι

η αλλαγή σε κάποια από αυτές τις μεταβλητές, μπορεί να αλλάξει την επίδραση και των άλλων δύο μεταβλητών για αυτό τον δεχόμαστε ως ενδεικτικό σημείο αναφοράς[10.13].

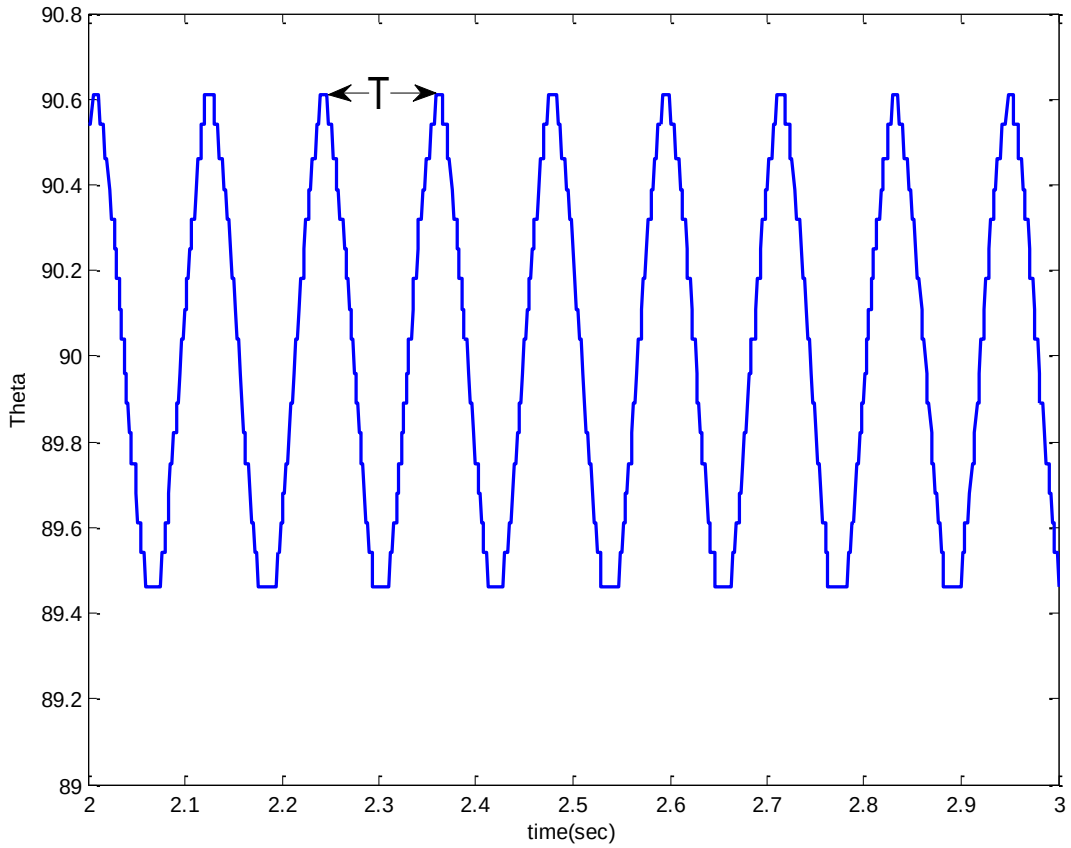
3.15 Η μέθοδος συντονισμού του ελεκτή PID «Ziegler-Nichols» («Ziegler-Nichols» tuning method)

Ο **John G. Ziegler** και ο **Nathaniel B. Nichols** δημιούργησαν μια ευρετική μέθοδο συντονισμού για την ρύθμιση (εύρεση κερδών) ενός αναλογικού - ολοκληρωτικού - διαφορικού ελεγκτή (PID ελεγκτή). Ο συντονισμός γίνεται με τη ρύθμιση του ολοκληρωτικού όρου (I) που καθορίζει την αντίδραση βασισμένη στο άθροισμα των λαθών και της παραγώγου (D) όταν τα κέρδη φτάσουν στο μηδέν. Ο αναλογικός όρος (P), τότε αυξάνεται (από το μηδέν) έως ότου φθάσει στην τελική τιμή, κατά την οποία η έξοδος του βρόχου ελέγχου ταλαντώνεται με ένα σταθερό πλάτος και η περίοδος ταλάντωσης χρησιμοποιείτε για να ρυθμιστεί ο αναλογικός όρος (P), και ο διαφορικός όρος (D), ανάλογα με το χρησιμοποιούμενο τύπο του ελεγκτή[11].

«Ziegler-Nichols» μέθοδος συντονισμού (tuning method)			
Control Type	K_p	K_i	K_d
P	$K_u / 2$	-	-
PI	$K_u / 2.2$	$1.2K_p / T_u$	-
classic PID	$0.60K_u$	$2K_p / T_u$	$K_pT_u / 8$

3.16 Υπολογισμός του ελεκτή «PID» με τη μέθοδο συντονισμού «Ziegler and Nichols» («Ziegler-Nichols» tuning method)

Αυξάνοντας τον **αναλογικό όρο (P)** σε μία τιμή όπου ο κινητήρας αρχίζει να εμφανίζει ταλαντώσεις, με τους υπόλοιπους όρους να είναι μηδέν, βρίσκουμε πειραματικά ότι στον κινητήρα της **Lego (NXT)** η τιμή του κέρδους Kp είναι 540. Η περίοδος ταλάντωσης (Tu) υπολογίζεται από την γραφική παράστασης γωνίας θ σε συνάρτηση με το χρόνο, όπως φαίνεται στην παρακάτω γραφική παράσταση.



Εικόνα 3.16.1 Γραφική παράσταση της γωνίας ϑ ως προς τον χρόνο

Αντικαθιστώντας τα παραπάνω δεδομένα στο πίνακα του **«Ziegler-Nichols»** και αφού υπολογίσουμε τα δεδομένα καταλήγουμε στο παρακάτω συγκεντρωτικό πίνακα αποτελεσμάτων με τα κέρδη του PID:

«Ziegler-Nichols» μέθοδος συντονισμού (tuning method)			
Control Type	K_p	K_i	K_d
P	**270**	-	-
PI	**245.5**	**2678.2**	-
classic PID	**324**	**5891**	**4.455**

Άρα για εύρυθμη λειτουργία του κινητήρα που χρησιμοποιήθηκε στη παρούσα πτυχιακή εργασία τα κέρδη που θα χρησιμοποιήσουμε θα είναι τα υπολογισμένα:

Kp : 324 , Ki : 5891 , Kd : 4.455

3.17 Πειραματική διαδικασία με χρήση κώδικα για τα κέρδη του ελεκτή PID

Παρακάτω θα δούμε την πειραματική διαδικασία που ακολουθήσαμε με τον ελεκτή PID στον κινητήρα **NXT**, θα αναλυθεί η διαδικασία, ο σκοπός της και τέλος τα συμπεράσματα στα οποία καταλήξαμε μέσω αυτής. Ο κώδικας χρησιμοποιεί τον ελεκτή PID και ανάλογα με την κάθε πειραματική διαδικασία αλλάζει η συνάρτηση void loop(). Στο παρακάτω πρόγραμμα ο χρήστης δίνει την επιθυμητή τιμή (γωνία θ) σε μοίρες μέσω του ποτενσιόμετρου και ο βραχίονας αφού βρεθεί στη θέση μηδέν («calibrating») εκτελεί τον αλγόριθμο του ελεκτή PID, και κινείται στην επιθυμητή τιμή (γωνία θ).

```
#include <PID_v1.h>       //δήλωση της βιβλιοθήκης PID
#include <LiquidCrystal.h>//δήλωση της βιβλιοθήκης LCD
#include <Servo.h>        //δήλωση της βιβλιοθήκης Servo
#include <Bounce.h>   //δήλωση της βιβλιοθήκης Bounce
#define BUTTON 9     //δήλωση του μπουτόν

#define encoder0PinA 2 // δήλωση της θύρας για το 1ο interrupt(PIN2)
#define encoder0PinB 3 // δήλωση της θύρας για το 2ο interrupt(PIN3)

#define legoSensorPin 1//δήλωση του πρώτου pin του αισθητήρα φωτός
#define legoSensor A5//δήλωση του δεύτερου pin του αισθητήρα φωτός

volatile unsigned int encoder0Pos = 32767.5; //δήλωση της
                                             //μεταβλητής encoder0Pos
unsigned int Aold = 0;     //δήλωση της μεταβλητής Aold
unsigned int Bnew = 0;     //δήλωση της μεταβλητής Bnew
volatile double angle = 0; //δήλωση της μεταβλητής angle

int motorPin0 = 7; //δήλωση της θύρας που είναι συνδεδεμένο το
                   //ένα PIN του κινητήρα (PIN7)
int motorPin1 = 8; //δήλωση της θύρας που είναι συνδεδεμένο το
                   //άλλο PIN του κινητήρα (PIN8)
int potPin = 5;    //δήλωση του ποτενσιόμετρου
int sensorPin = A0;//δήλωση της θύρας σύνδεσης του ποτενσιόμετρου
int val = 0;       //δήλωση της μεταβλητής που αποθηκεύεται η τιμή
                   //του ποτενσιόμετρου

int lightvalue;//δήλωση της μεταβλητής lightvalue

double Setpoint,Input,Output;//δήλωση των μεταβλητών

                             //Setpoint,Input,Output

double err, Pwm1_val;//δήλωση των μεταβλητών err,Pwm1_val

double QPWM;//δήλωση της μεταβλητής QPWM

PID myPID(&Input, &Output, &Setpoint,324, 5891.4.455, DIRECT);//δήλωση-

                                          //ρύθμιση των κερδών του PID

LiquidCrystal lcd(4, 6, 10, 11, 12, 13); //δήλωση των pin της οθόνης


void setup()
        {
pinMode(encoder0PinA, INPUT); //δήλωση του καναλιού A του encoder
pinMode(encoder0PinB, INPUT); // δήλωση του καναλιού B

pinMode(motorPin0, OUTPUT); // δήλωση της εξόδου σύνδεσης του κινητήρα
pinMode(motorPin1, OUTPUT); //δήλωση της δεύτερης εξόδου

attachInterrupt(0, doEncoderA, CHANGE);// δήλωση της συνάρτησης του
                                       //interrupt (INT0)
attachInterrupt(1, doEncoderB, CHANGE);// δήλωση της δεύτερης
```

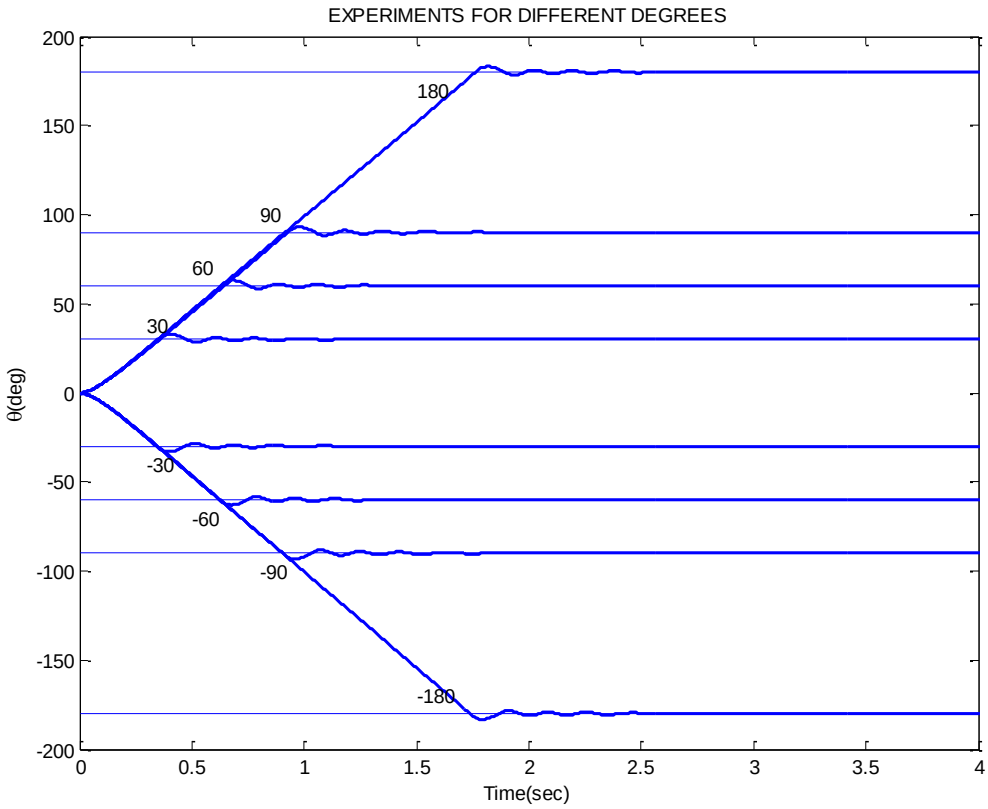
```
                                        //συνάρτησης του interrupt (INT1)
        }
myPID.SetMode(AUTOMATIC);//ενεργοποίηση του PID
Bounce bouncer = Bounce(BUTTON,5);//ρύθμιση του debounce στα 5ms
pinMode(BUTTON,INPUT); //δήλωση του μπουτόν ως είσοδο

lcd.begin(16, 2);// ορίζουμε τις γραμμές και τις στήλες της οθόνης
pinMode(legoSensor,INPUT);//δήλωση του PIN του αισθητήρα φωτός ως


                          //είσοδο
pinMode(legoSensorPin, OUTPUT); //δήλωση του PIN του αισθητήρα φωτός ως


                                //έξοδο
digitalWrite(legoSensorPin, HIGH);//δήλωση του PIN του αισθητήρα φωτός


                                    //ως HIGH

delay(500);
pinMode(legoSensorPin, INPUT);//δήλωση του PIN του αισθητήρα φωτός ως

                              //είσοδο
lightvalue=analogRead(A5);//ισότητα της μεταβλητής lightvalue με την


                         //είσοδο A5

delay(1000);
lcd.setCursor(0, 0);//ορίζουμε τις γραμμές και τις στήλες της οθόνης


                   //στη θέση 0,0
lcd.print("CALIBRATING...");//εκτύπωση μηνύματος στην οθόνη
lcd.setCursor(0, 1);//ορίζουμε τις γραμμές και τις στήλες της οθόνης


                   //στη θέση 0,1


lcd.print("PLEASE WAIT!!!"); //εκτύπωση μηνύματος στην οθόνη

lightvalue=analogRead(A5);//ισότητα της μεταβλητής lightvalue με την


                         //είσοδο A5

do{ // βρόγχος ελέγχου do…while
digitalWrite(motorPin0, 1);//δήλωση της θύρας που είναι συνδεδεμένο το

                          //ένα PIN του κινητήρα (PIN7)ως HIGH(1)
digitalWrite(motorPin1, 0);// δήλωση της θύρας που είναι συνδεδεμένο το
```

```
                          //ένα PIN του κινητήρα (PIN8)ως LOW(0)
analogWrite(5,100);//δήλωση του PIN 5 στην τιμή 100

lightvalue=analogRead(A5); //ισότητα της μεταβλητής lightvalue με την


                          //είσοδο A5

}while(lightvalue<=600);// βρόγχος ελέγχου do…while

 digitalWrite(motorPin0, 0); //δήλωση της θύρας που είναι συνδεδεμένο το
                            //ένα PIN του κινητήρα (PIN7)ως LOW(0)

 digitalWrite(motorPin1, 0); //δήλωση της θύρας που είναι συνδεδεμένο το

                             //ένα PIN του κινητήρα (PIN8)ως LOW(0)
 encoder0Pos=0;//δήλωση της μεταβλητής encoder0Pos ως μηδέν
 delay(1000);
 lcd.clear();}}//εκκαθάριση της οθόνης


void loop()
           {
lcd.setCursor(0, 0); //ορίζουμε τις γραμμές και τις στήλες της οθόνης

                    //στη θέση 0,0
lcd.print("THETA");//εκτύπωση μηνύματος στην οθόνη

lcd.setCursor(0, 1); //ορίζουμε τις γραμμές και τις στήλες της οθόνης
                    //στη θέση 0,1
lcd.print(angle); //εκτύπωση της τιμής της μεταβλητής angle στην
                 //οθόνη
lcd.setCursor(8, 0); //ορίζουμε τις γραμμές και τις στήλες της οθόνης

                    //στη θέση 8,0
lcd.print("SETPOINT");//εκτύπωση μηνύματος στην οθόνη

lcd.setCursor(8, 2); //ορίζουμε τις γραμμές και τις στήλες της οθόνης

                    //στη θέση 8,2
lcd.print(Setpoint);//εκτύπωση της τιμής της μεταβλητής Setpoint στην

                   //οθόνη
delay(50);

lcd.clear();//εκκαθάριση της οθόνης
{

Setpoint = map(analogRead(A0), 0, 1023, 0, 360);//δήλωση της μεταβλητής


                     //Setpoint στη θύρα A0 και αντιστοίχηση των τιμών
```

```
                    //της από 0-1023 σε 0-360
Input=angle;//ισότητα της μεταβλητής Input με την μεταβλητή angle

err=Setpoint-Input;//ισοδυναμία της μεταβλητής err με την μεταβλητή


                  //Setpoint-Input

Pwm1_val=Output;//ισοδυναμία της μεταβλητής Pwm1_val με την μεταβλητή

                //Output
myPID.Compute();//υλοποίηση του αλγόριθμου του PID

  if (Pwm1_val>=0) //

  {

  digitalWrite(motorPin0,1);

  digitalWrite(motorPin1,0);

  }

  if (Pwm1_val<0)

  {

  digitalWrite(motorPin0,0);

  digitalWrite(motorPin1,1);

  Pwm1_val=abs(Pwm1_val);
  }
  QPWM=Pwm1_val;//ισότητα της μεταβλητής Input με την μεταβλητή angle
  analogWrite(5,QPWM);//δήλωση του PIN 5 στην τιμή της μεταβλητής QPWM
bouncer.update ( ); //ενημέρωση του debouncer
}}

void doEncoderA(){  // Ονομασία της συνάρτησης του interrupt (INT0)
       Bnew^Aold ? encoder0Pos++:encoder0Pos--; // Η συνάρτηση του INT0
            Aold=digitalRead(encoder0PinA);
            angle=(encoder0Pos*0.5-16383.75)/7;
                 }

void doEncoderB(){  //  Ονομασία της συνάρτησης του interrupt (INT1)
           Bnew=digitalRead(encoder0PinB);  // Η συνάρτηση του INT1
           Bnew^Aold ? encoder0Pos++:encoder0Pos--;
           angle=(encoder0Pos*0.5-16383.75)/7;
                 }
```

3.17.1 Πειραματική διαδικασία 1

Χρησιμοποιώντας τα κέρδη που υπολογίσαμε από τι μέθοδο συντονισμού **«ZIEGFLER-NICHOLS»** στη βιβλιοθήκη του PID, **Kp=324 Ki=5891** και **Kd=4,455**, ελέγχουμε αν όντως τα κέρδη αυτά ανταποκρίνονται σε εύρυθμη λειτουργιά του κινητήρα. Έτσι θέσαμε ως επιθυμητή τιμή στα 8 πειράματα τις εξής μοίρες: **30,45,90,180,-30,-45,-90,-180** και καταγράφοντας τα αποτελέσματα στο «Serial Monitor» του Arduino και μετά από κατάλληλη επεξεργασία στο πρόγραμμα «MATLAB» (matrix laboratory), παίρνουμε την παρακάτω συγκεντρωτική γραφική παράσταση.



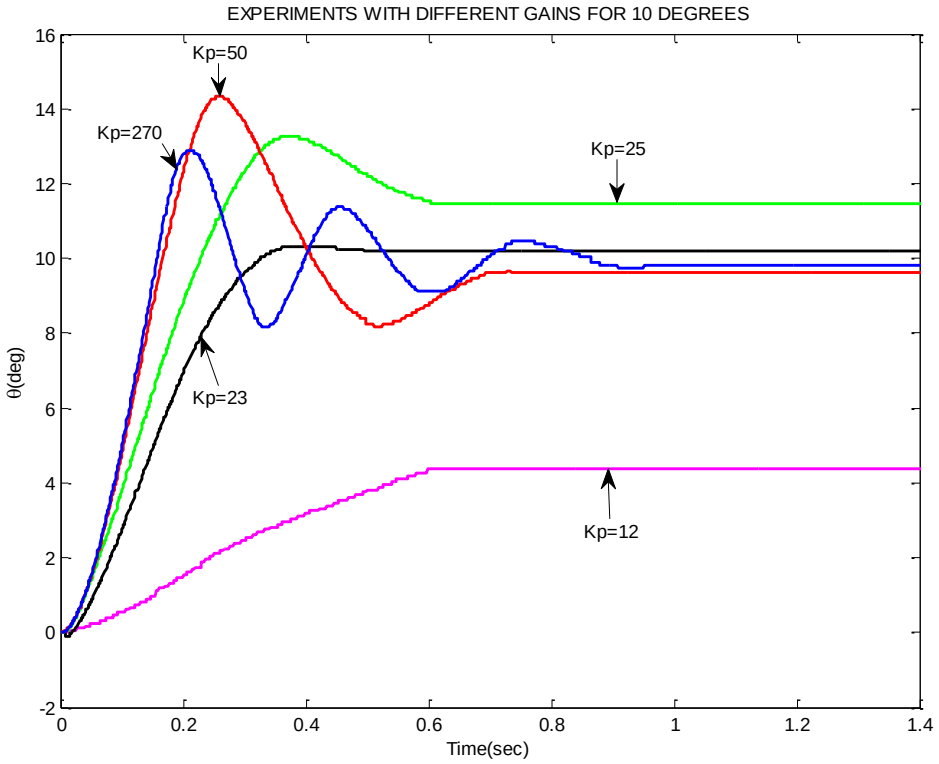
Εικόνα 3.17.1 Συγκεντρωτική γραφική παράσταση πειραμάτικης διαδικαίας

Συμπεραίνουμε ότι η συμπεριφορά του κινητήρα τόσο στις αρνητικές όσο και στις θετικές τιμές είναι εξίσου ομαλή χωρίς ανυψώσεις και υπερυψώσεις και με την πάροδο του χρόνου το μόνιμο σφάλμα εξαλείφεται και ο βραχίονας παραμένει στην επιθυμητή τιμή. Παρόλα αυτά η μέθοδος συντονισμού **«ZIEGFLER-NICHOLS»** βασίζεται στην πρακτική ανάλυση των δεδομένων για αυτό και παρατηρείται υπερύψωση στην επιθυμητή τιμή (Setpoint) και ταλαντώσεις, πράγμα που δεν είναι αποδεκτό σε εφαρμογές όπου χρειάζεται μεγάλη ακρίβεια. Για αυτό χρησιμοποιώντας ως σκεπτικό την αρχή της μεθόδου **«ZIEGFLER-NICHOLS»** και κάνοντας κάποια πειράματα με διαφορετικά κέρδη καταλήξαμε σε κάποια

διαφορετικά κέρδη για την ρύθμιση του PID ελεκτή καταγράφοντας σταδιακά τα εξής αποτελέσματα:

3.17.2 Πειραματική διαδικασία 2

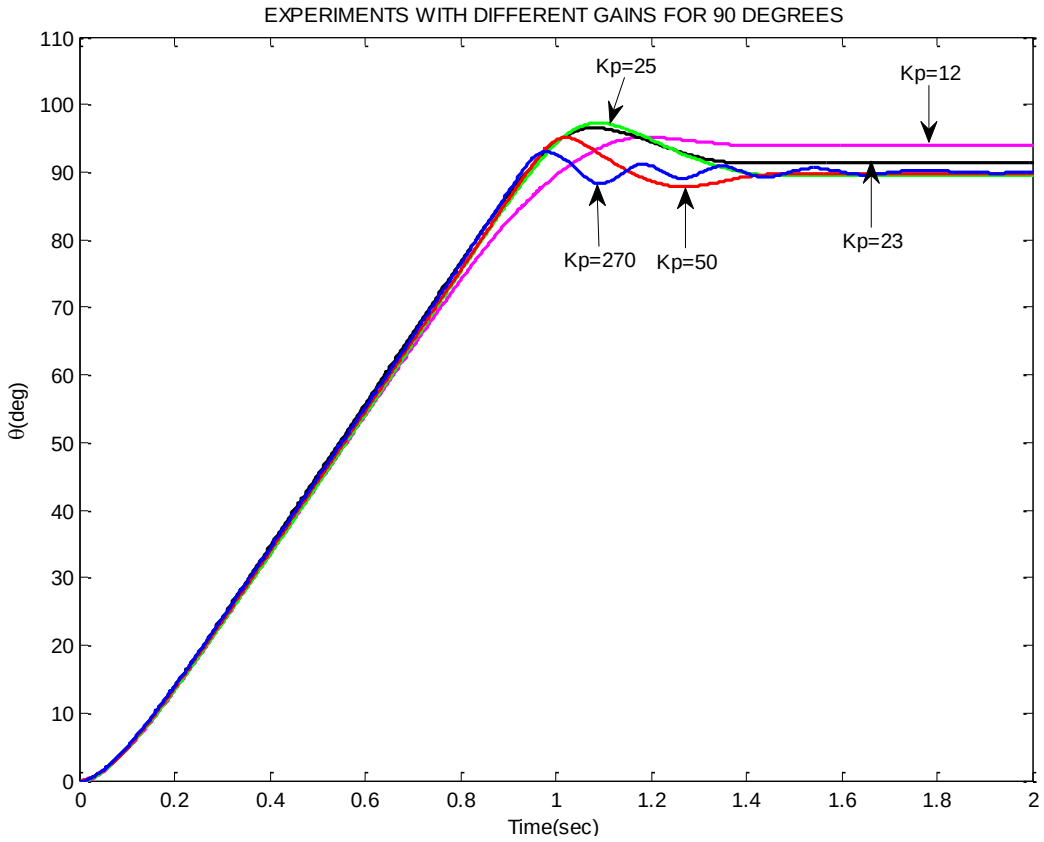
1) Ρυθμίζοντας την επιθυμητή τιμή (Setpoint) στης **10** μοίρες και αυξάνοντας μόνο το κέρδος **Kp (Ki=0, Kd=0)**, δηλαδή παρατηρούμε τον τρόπο που επιδρά το σύστημα στην αλλαγή μόνο του όρου Kp, παίρνουμε για το ίδιο χρονικό διάστημα αρκετά δείγματα και παρατηρούμε ότι για τον όρο P υπάρχει μεγάλη απόκλιση στο κέρδος Kp (Kp=270). Η υπερύψωση είναι αυξημένη για όλες τις τιμές του Kp όπως και ο χρόνος ανύψωσης πράγμα που δεν συμβαίνει για Kp=23. Όπως φαίνεται στην παρακάτω γραφική παράσταση, για Kp=23, παρατηρούμε την καλύτερη συμπεριφορά του βραχίονα αλλά με εσφαλμένη απόκριση του συστήματος.



Εικόνα 3.17.2 Απόκριση του συστήματος (θ) *για επιθυμητή τιμή ϑd =10 για διαφορετικά* κέρδη Kp

2) Ακολουθώντας την ίδια διαδικασία με τη διαφορά ότι αλλάζουμε την επιθυμητή τιμή (Setpoint) στης **90** μοίρες και αυξάνοντας μόνο το κέρδος **Kp (Ki=0, Kd=0)** παρατηρούμε μια διαφορετική συμπεριφορά στο βραχίονα καθώς οι ταλαντώσεις

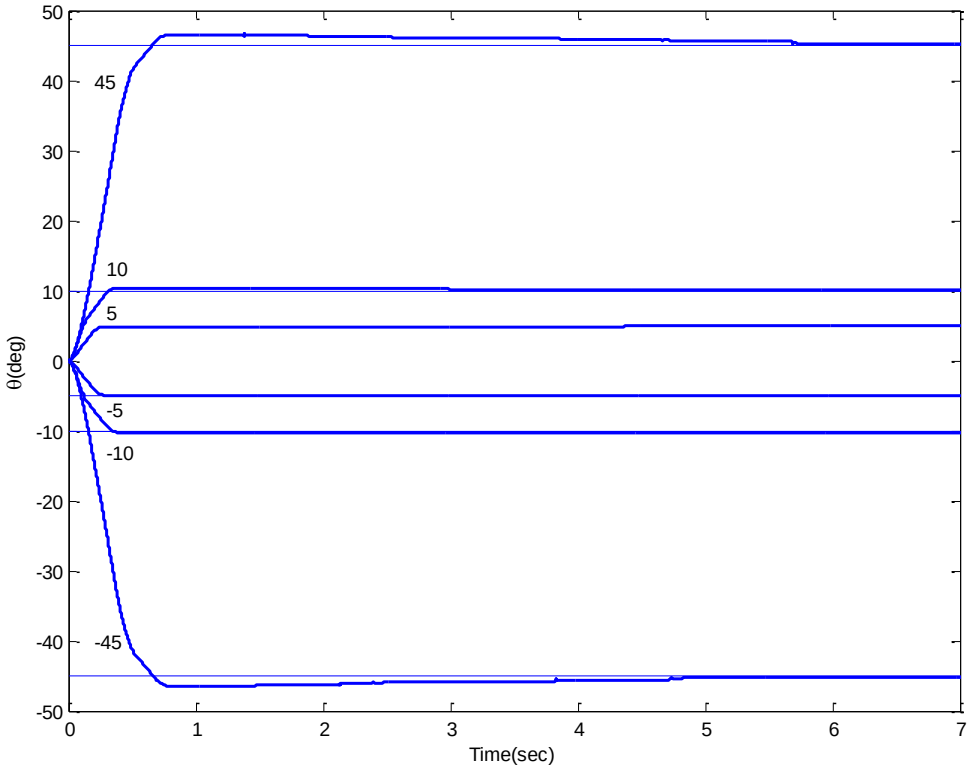
είναι πολύ μικρές σε σχέση το πείραμα 1 και υπερύψωση εμφανίζεται μόνο για Kp=25 και Kp=23. Τα κέρδη με χαμηλές τιμές (Kp 12, 23) εμφανίζουν σφάλμα μόνιμης κατάστασης. Τέλος η απόκριση του συστήματος στα κέρδη με υψηλές τιμές (Kp 270, 50) είναι ακριβής αλλά εμφανίζονται μικρές ταλαντώσεις.



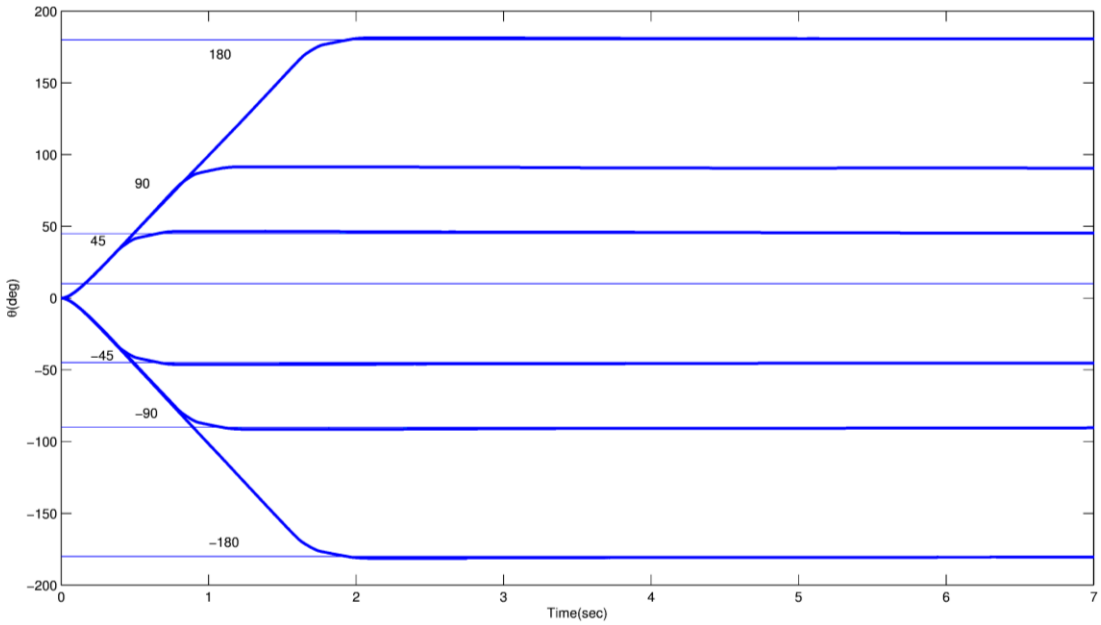
Εικόνα 3.17.3 Απόκριση του συστήματος (θ) για επιθυμητή τιμή θd =90 για διαφορετικά κέρδη Kp

3.17.3 Πειραματική διαδικασία 3

Κάνοντας κάποια πειράματα με διαφορετικά κέρδη, καταλήξαμε σε διαφορετικά κέρδη για την ρύθμιση του PID ελεκτή με τα οποία ο κινητήρας παρουσίασε ομαλή συμπεριφορά και η απόκριση του συστήματος είναι ακριβής. Τα κέρδη αυτά για το κάθε όρο του ελεκτή PID είναι: **Kp: 150, Ki: 70, Kd: 15.** Χρησιμοποιώντας τις νέες αυτές τιμές διευρύνουμε το φάσμα των επιθυμητών τιμών (Setpoint) που μέχρι τώρα χρησιμοποιούσαμε, ώστε να αντιληφθούμε την απόκριση του συστήματος σε αυτές. Θέτουμε την επιθυμητή τιμή (Setpoint) στις εξής μοίρες: **5, 10, 45, 90, 180, -5, -10, -45, -90, -180** και καταγράφοντας τα αποτελέσματα στο «Serial Monitor» του Arduino παίρνουμε τις παρακάτω συγκεντρωτικές γραφικές παραστάσεις:



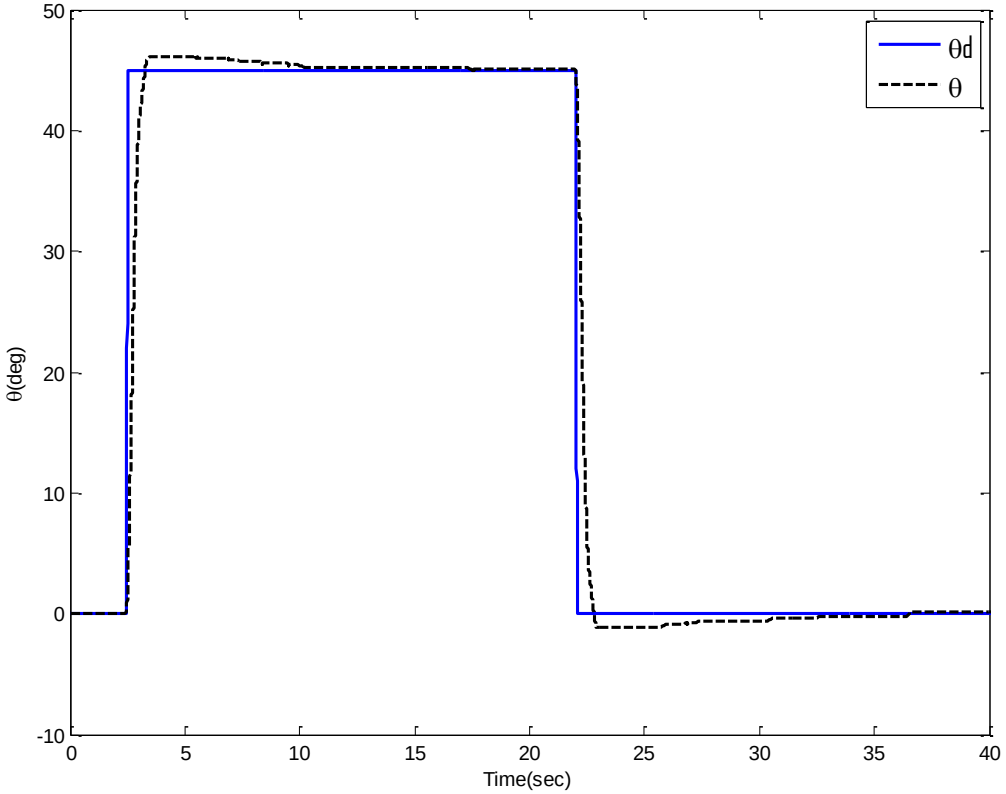
Εικόνα 3.17.3 Απόκριση του συστήματος (ϑ) για επιθυμητή τιμή ϑd = **5, 10, 45, -5, -10, -45** *μοίρες*



Εικόνα 3.17.4 Απόκριση του συστήματος (ϑ) για επιθυμητή τιμή ϑd = 45, 90, 180, -45, -90, -180 μοίρες

3.17.5 Πειραματική διαδικασία 4

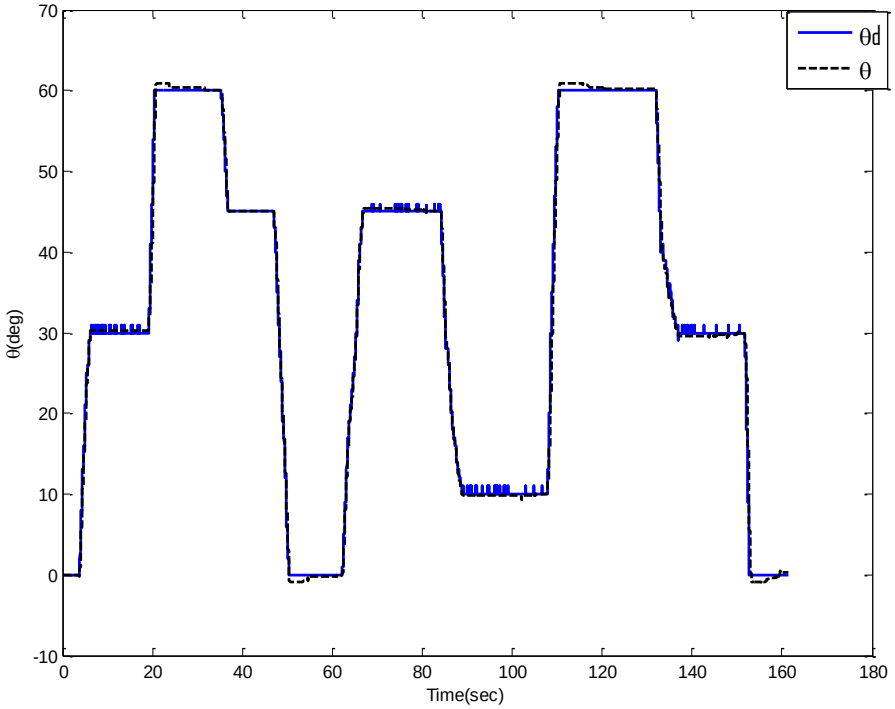
Ακολουθώντας την ίδια διαδικασία με την μόνη διαφορά ότι η επιθυμητή τιμή (Setpoint) δίνεται μέσα από τον κώδικα και ορίζεται στις **45** μοίρες και μετά από 15 δευτερόλεπτα επανέρχεται στην αρχική θέση 0 παρατηρούμε ότι σε αυτό το χρονικό διάστημα το σφάλμα μεταξύ επιθυμητής και πραγματικής τιμής της γωνίας φθίνει και τελικά μηδενίζεται.



Εικόνα 3.17.5 Διάγραμμα με τη χρήση κώδικα για την επιθυμητή τιμή ϑd

3.17.4 Πειραματική διαδικασία 5

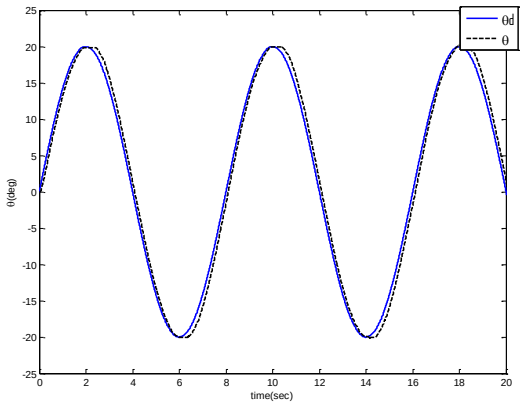
Με τα ίδια κέρδη **Kp: 150, Ki: 70, Kd: 15** χρησιμοποιούμε ένα ποτενσιόμετρο και ορίζουμε η επιθυμητή τιμή (Setpoint) να μεταβάλλεται να μεταβάλλεται από αυτό τυχαία μεταξύ 0 και 60 μοιρών. Παρατηρούμε ότι η επιθυμητή τιμή της γωνίας παρότι μεταβάλλεται αρκετές φορές σε πολύ μικρό χρονικό διάστημα σχεδόν ταυτίζεται με την πραγματική τιμή της γωνίας.

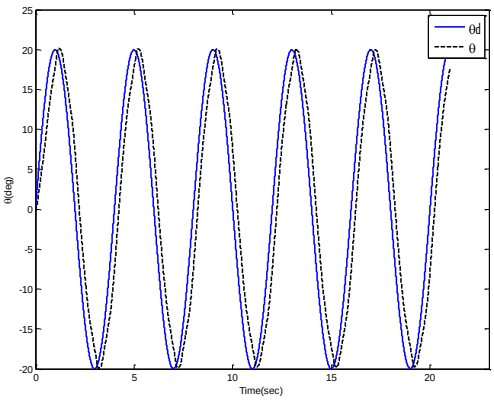


Εικόνα 3.17.5 Διάγραμμα με τη χρήση ποτενσιόμετρου για την επιθυμητή τιμή *ϑd*

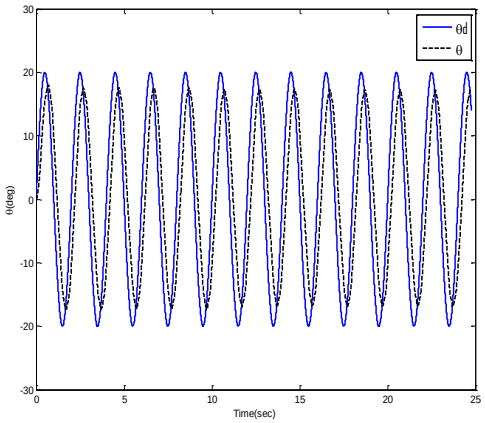
3.17.6 Πειραματική διαδικασία 6

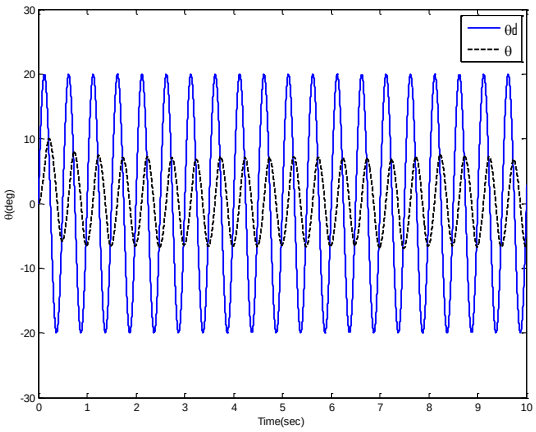
Τέλος θέτουμε στον κώδικα ως επιθυμητή τιμή (Setpoint) την συνάρτηση **f(x)=A0 sin (2π/T*t)** και αλλάζοντας κάθε φορά την περίοδο του σήματος παρατηρούμε την συμπεριφορά του κινητήρα ο οποίος στις αργές εναλλαγές του χρόνου ακολουθεί την επιθυμητή τιμή ενώ όσο αυξάνουμε την περίοδο του σήματος η απόκριση του συστήματος μειώνεται.





Εικόνα 3.17.6.1 Περίοδος επιθυμητής τιμής: 8sec *Εικόνα 3.17.6.2 Περίοδος επιθυμητής τιμής: 4sec*





Εικόνα 3.17.6.3 Περίοδος επιθυμητής τιμής : 2sec *Εικόνα 3.17.6.4 Περίοδος επιθυμητής τιμής : 1sec*

Κεφάλαιο 4

Κατασκευή εκπαιδευτικής διάταξης με χρήση των κινητήρων τύπου «Servo» - «Stepper»

4.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφονται οι κινητήρες τύπου **«SERVO»**, **«STEPPER»** όσον αφορά τον τρόπο λειτουργίας τους, τις δυνατότητες τους και τον έλεγχο τους. Επίσης παρουσιάζεται ο τρόπος σύνδεσης τους με την **υπολογιστική πλατφόρμα Arduino** και οι κώδικες για την λειτουργία των δυο διατάξεων.

4.1.1 Οι κινητήρες τύπου «RC SERVO»

Η λέξη «servo» προέρχεται από την λατινική λέξη «servus» που σημαίνει υπηρέτης, μπορεί να θεωρηθεί η λειτουργία του κινητήρα έτσι αφού ο σερβοκινητήρας είναι ένας κινητήρας ο οποίος μπορεί να κινηθεί με τεράστια ακρίβεια ακολουθώντας ακριβώς τις εντολές που θα του δοθούν. Οι σερβοκινητήρες είναι μια εξελιγμένη μορφή των απλών κινητήρων με βελτιωμένα λειτουργικά χαρακτηριστικά, την πυκνότητα ισχύος, τη ροπή αδράνειας, και τον βαθμό απόδοσης. Είναι κινητήρες συνεχούς ρεύματος με ένα

ενσωματωμένο σύστημα γραναζιών και ένα κύκλωμα ανάδρασης και χρησιμοποιούνται σε συστήματα υψηλών απαιτήσεων όπου η ταχύτητα απόκρισης του συστήματος ως προς τον έλεγχο της ταχύτητας, της ροπής, και της θέσης πρέπει να γίνεται με μεγάλη ακρίβεια και ταχύτατα. Σε γενικές γραμμές το μέγεθος τους είναι μικρό ειδικά σε σχέση με την ροπή που μπορούν να μεταφέρουν και την ταχύτητα που μπορούν να περιστραφούν (60-180rpm) και σε συνδυασμό με το χαμηλό τους κόστος αποτελούν την καλύτερη λύση για την χρήση σε ρομποτικά συστήματα.

Εικόνα 4.1.1 Κινητήρας τύπου «RC Servo»

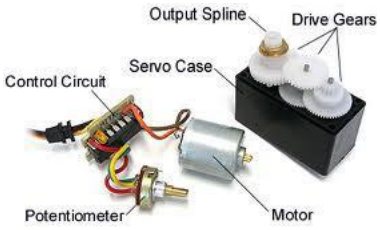
4.1.2 Συνδεσμολογία του κινητήρα τύπου «RC SERVO»

Η συνδεσμολογία τους είναι απλή αφού αποτελείται από **τρία** καλώδια, το πρώτο είναι η γείωση (-), το δεύτερο η τάση τροφοδοσίας (+), και το τρίτο το σήμα PWM με το οποίο ελέγχουμε την θέση. Διακρίνονται σε δύο βασικές κατηγορίες τους τυπικούς σερβοκινητήρες και τους σερβοκινητήρες ταχύτητας, οι μεν κινούνται σε μια επιθυμητή θέση που την ορίζει ο χρήστης από 0 έως 180 μοίρες, οι δε κινούνται με ταχύτητα που ορίζει ο χρήστης από 0 έως 360 μοίρες.

4.1.3 Βασική δομή ενός σερβοκινητήρα

Ένας σερβοκινητήρας αποτελείται από έξη βασικά μέρη:

- το κύκλωμα ελέγχου («Control Circuit»), το οποίο διαβάζει τα σήματα PWM που δέχεται ο κινητήρας και αναλόγως τον περιστρέφει στην επιθυμητή θέση
- τον κινητήριο άξονα («Output Spline»)
- το σύστημα γραναζιών («Drive Gears»), το οποίο μεταφέρει την κίνηση στον άξονα
- το ποτενσιόμετρο («Potentiometer»), το οποίο συνδέεται με τον κινητήριο άξονα και περιστρέφεται μ’ αυτόν έτσι ώστε να μεταβάλλει η αντίσταση του
- τον κινητήρα («Motor»)



Εικόνα 4.1.3 Εσωτερικό ενός κινητήρα τύπου «RC SERVO»

4.1.4 Έλεγχος του κινητήρα τύπου «RC SERVO»

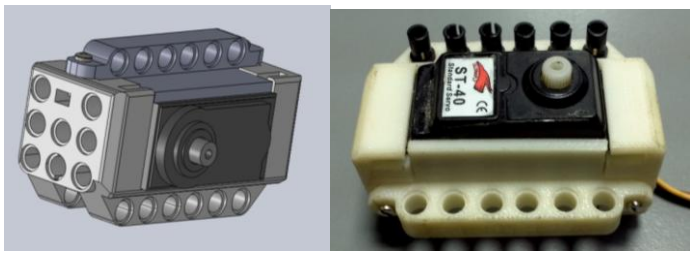
Ο έλεγχος του σερβοκινητήρα γίνεται με τη χρήση της τεχνολογίας «Pulse Width Modulation» (PWM),η λειτουργία της οποίας έχει αναφερθεί λεπτομερώς στο Κεφάλαιο 1 (1.4.8.3), το σήμα τετραγωνικών παλμών είναι 50Hz άρα περίοδο 20ms και ανάλογα με το εύρος του προκύπτει η επιθυμητή γωνία. Στους περισσότερους σερβοκινητήρες το εύρος του παλμού κυμαίνεται από 1ms έως 2ms, με δεδομένο ότι η γωνία μπορεί να είναι από 0 έως 180 μοίρες προκύπτει ότι αν το εύρος του παλμού είναι 1ms τότε η γωνία θα αντιστοιχεί σε 0 μοίρες, 1,5ms σε 90 μοίρες και 2ms σε 180 μοίρες[15].

Εικόνα 4.1.4 Εύρος παλμών του κινητήρα τύπου «RC SERVO»

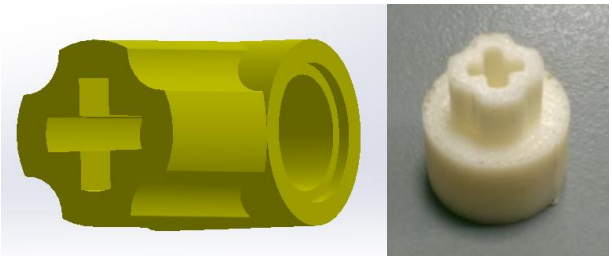
4.2 Κατασκευή της θήκης και του εξαρτήματος συγκράτησης του άξονα του «RC Servo»

Για να συνδέσουμε έναν κινητήρα **«RC Servo»** με το εκπαιδευτικό πακέτο Lego Mindstorms NXT δημιουργήσαμε μια θήκη στην οποία τοποθετείται ο κινητήρας και γίνεται και αυτός μέρος του εκπαιδευτικού πακέτου της **Lego**. Χρησιμοποιώντας το «3D» σχεδιαστικό πρόγραμμα Solidworks της Dassault Systèmes σχεδιάσαμε την θήκη και το εξαρτήματος συγκράτησης του άξονα για τον κινητήρα, ώστε να μπορούμε εύκολα να τον τοποθετούμε σε οποιαδήποτε κατασκευή και εφαρμογή δημιουργούμε με το εκπαιδευτικό πακέτο της **Lego**. Έπειτα μέσω της μηχανής τρισδιάστατης εκτύπωσης “Dimension elite” εκτυπώσαμε τη θήκη και το εξάρτημα συγκράτησης του άξονα. Τα δύο εξαρτήματα

σχεδιάστηκαν ακριβώς όπως η **Lego** έχει σχεδιάσει όλα τα μέρη του εκπαιδευτικού πακέτου Lego Mindstorms NXT, ως «τούβλα» δηλαδή, χρησιμοποιώντας τις ίδιες διαστάσεις ώστε η θήκη να «κουμπώνει» σε όλα τα εξαρτήματα της υπόλοιπης πλατφόρμας της Lego.



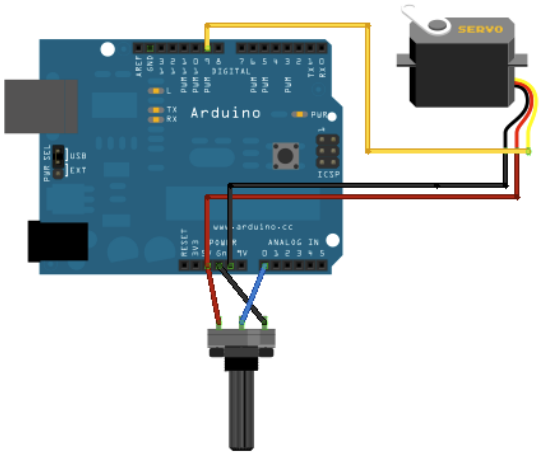
Εικόνα 4.2.1 Όψη της θήκης μέσα από το περιβάλλον του" Solidworks" και στην πραγματική της μορφή



Εικόνα 4.2.2 Όψη του εξαρτήματος συγκράτησης του άξονα μέσα από το περιβάλλον του "Solidworks" και στην πραγματική του μορφή

4.3 Σύνδεση υπολογιστικής πλατφόρμας Arduino Uno με τον κινητήρα τύπου «RC Servo»

Η σύνδεση της **υπολογιστικής πλατφόρμας Arduino Uno** με τον κινητήρα τύπου **«RC Servo»** μπορεί να πραγματοποιηθεί αρκετά εύκολα χρησιμοποιώντας τη βιβλιοθήκη του Arduino, για την οποία έχουμε αναφερθεί αναλυτικά στο Κεφάλαιο 1(1.12.2), κάνοντας φυσικά τις απαραίτητες αλλαγές.



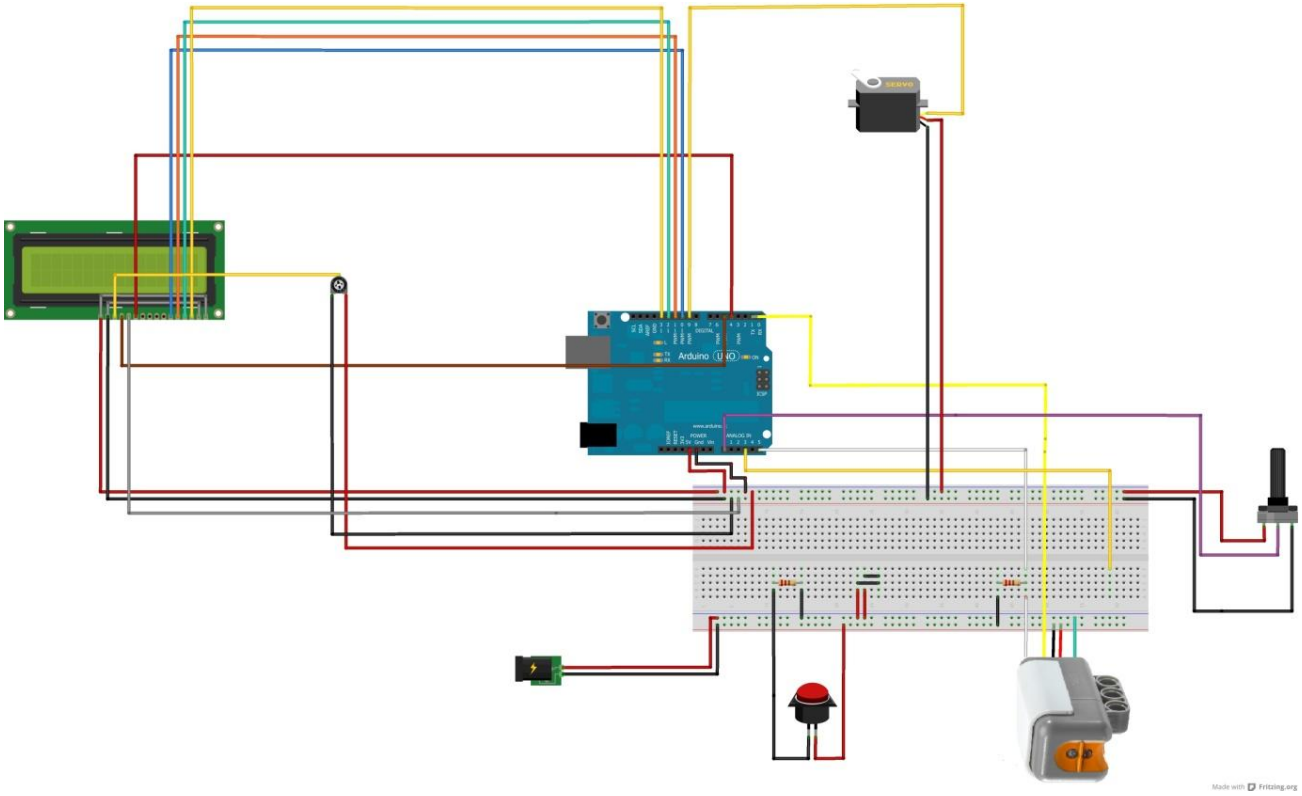
Εικόνα 4.3 Συνδεσμολογία του κινητήρα "RC Servo"

4.4 Ο κώδικας για την λειτουργία όλων των παρελκόμενων της κατασκευής με τον κινητήρα τύπου «RC Servo»

Παρακάτω παραθέτουμε τον κώδικα για την λειτουργία της διάταξης με τον κινητήρα τύπου **«RC Servo»** και όλα τα παρελκόμενα, δηλαδή την οθόνη LCD, τον αισθητήρα φωτός, το αναλογικό περιστροφικό ποτενσιόμετρο και το μπουτόν. Ο χρήστης δίνει μια τιμή σε μοίρες (0-360) μέσω ενός ποτενσιόμετρου και με το πάτημα του μπουτόν ο βραχίονας κινείται στην επιθυμητή τιμή και τα αποτελέσματα εμφανίζονται σε πραγματικό χρόνο στην LCD οθόνη.

```
#include <Servo.h>      //δήλωση της βιβλιοθήκης του «RC Servo»
#include <LiquidCrystal.h>    //δήλωση της βιβλιοθήκης LCD
#include <Bounce.h>   //δήλωση της βιβλιοθήκης Bounce
#define BUTTON 2     //δήλωση του μπουτόν
#define legoSensorPin1    //δήλωση του πρώτου pin του αισθητήρα φωτός
#define legoSensor A5    //δήλωση του δεύτερου pin του αισθητήρα φωτός

int ReadLegoSensor, BUTTON, lightvalue;  //δήλωση των μεταβλητών

                                //ReadLegoSensor, BUTTON, lightvalue
int potpin = A0;    //δήλωση της θύρας  του ποτενσιόμετρου

int val;           //δήλωση της μεταβλητής val
LiquidCrystal lcd(4, 6, 10, 11, 12, 13);  //δήλωση των θυρών της
Bounce bouncer = Bounce(BUTTON,5);//ρύθμιση του debounce στα 5ms
                                          //βιβλιοθήκης LCD
Servo myservo;    //ονομασία του Servo που χρησιμοποιούμε
void setup()
      {
myservo.attach(9);   //δήλωση της θύρας  που συνδέουμε τον Servo
lcd.begin(16, 2);   //δήλωση των  στηλών και γραμμών της οθόνης
pinMode(legoSensor,INPUT);   //δήλωση του αισθητήρα φωτός ως

                            //είσοδο
pinMode(legoSensorPin, OUTPUT);// δήλωση του αισθητήρα φωτός ως έξοδο
digitalWrite(legoSensorPin, HIGH); //δήλωση της εξόδου του αισθητήρα

                                  //φωτός ως HIGH
pinMode(BUTTON,INPUT); //δήλωση του μπουτόν ως είσοδο
pinMode(legoSensorPin, INPUT); //δήλωση του αισθητήρα φωτός ως είσοδο
lightvalue=analogRead(A5); //διαβάζουμε την είσοδο του αισθητήρα φωτός
delay(2000);
lcd.setCursor(0, 0); //θέτουμε τον κέρσορα της οθόνης στη θέση 0,0
lcd.print("CALIBRATING..."); //εκτύπωση μηνύματος στην οθόνη
lcd.setCursor(0, 1);   //θέτουμε τον κέρσορα της οθόνης στη θέση 0,1
lcd.print("PLEASE WAIT!!!");   //εκτύπωση μηνύματος στην οθόνη

do//βρόγχος ελέγχου «do…while»
                    {
analogWrite(val,50);//δίνουμε την τιμή 50 στη μεταβλητή val
lightvalue=analogRead(A5);//διαβάζουμε την είσοδο του αισθητήρα φωτός
```

```
}while(lightvalue<=600);//βρόγχος ελέγχου «do…while»
analogWrite(val, 0);//δίνουμε την τιμή 0 στην μεταβλητή val
lcd.clear();}  // εκκαθάριση της οθόνης
void loop()
{lcd.setCursor(0, 0);  //θέτουμε τον κέρσορα της οθόνης στη θέση 0,0
lcd.print("ANGLE");   //εκτύπωση μηνύματος στην οθόνη
lcd.setCursor(0, 1); //θέτουμε τον κέρσορα της οθόνης στη θέση 0,1
lcd.print(val);     //εκτυπώνουμε  την τιμή της μεταβλητής val
lcd.setCursor(8, 0);    //θέτουμε τον κέρσορα της οθόνης στη θέση 8,0
lcd.print("SETPOINT"); //εκτυπώνουμε  την έκφραση μέσα στην παρένθεση
lcd.setCursor(8, 2);  //θέτουμε τον κέρσορα της οθόνης στη θέση 8,2
lcd.print(Setpoint); //εκτυπώνουμε  την τιμή της μεταβλητής Setpoint
delay(100);
lcd.clear();  //εκκαθάριση της οθόνης
val = analogRead(potpin);  //διαβάζουμε την τιμή του ποτενσιόμετρου

val = map(val, 0, 1023, 0, 180);  //αντιστοιχούμε την τιμή του

                                  //ποτενσιόμετρου από 0-1023 σε 0-180
bouncer.update ( ); //ενημέρωση του debouncer
myservo.write(val); //η θέση του Servo σε μοίρες
delay(15);  //μικρή καθυστέρηση μέχρι να φθάσει ο κινητήρας στην
           //επιθυμητή τιμή
}
```



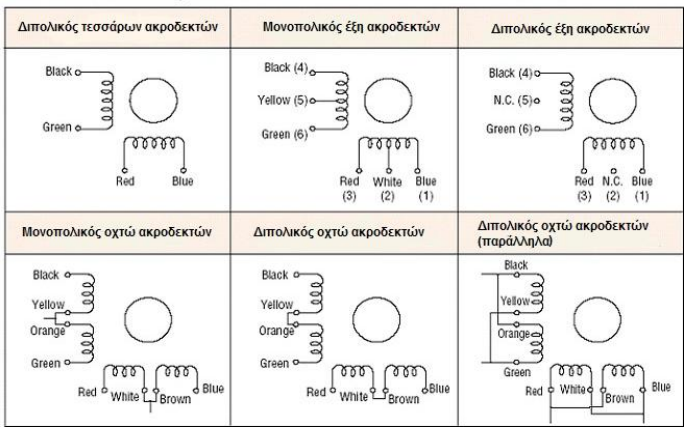
Εικόνα 4.4 Συνδεσμολογία της κατασκευής με τον κινητήρα τύπου "RC Servo"

4.5.1 Βηματικοί κινητήρες («Stepper motors»)

Οι **βηματικοί κινητήρες** είναι κινητήρες συνεχούς ρεύματος μόνιμου μαγνήτη ή μεταβλητής αντίστασης και το βασικό τους χαρακτηριστικό είναι ότι μπορούν να κινηθούν με πολύ μεγάλη ακρίβεια σε συγκεκριμένες γωνίες όπως επίσης να κρατούν τη ροπή συγκράτησης υπό μηδενική ταχύτητα. Η ονομασία τους προέρχεται από την ακριβή κίνηση κατά τμήματα (βήματα/steps) συγκεκριμένης γωνίας , πράγμα που γίνεται με την εφαρμογή ψηφιακών παλμών σε ένα ηλεκτρικό κύκλωμα οδήγησης. Γενικά οι **βηματικοί κινητήρες** κατασκευάζονται με συγκεκριμένα όρια στα βήματα που μπορούν να κάνουν σε μία πλήρη περιστροφή, συνήθως αυτά είναι 12, 24, 72, 144, 180 και 200 οπότε το κάθε βήμα αντιλαμβανόμαστε ότι αντιστοιχεί σε συγκεκριμένες μοίρες οι οποίες είναι αντίστοιχα 30, 15, 5, 2.5, 2, και 1.8. Η τροφοδοσία που δέχονται είναι συνήθως μεγαλύτερη από αυτή που μπορεί να τους δώσει ένας μικροελεγκτής, δίνεται από τον κατασκευαστή και είναι συνήθως 5V, 9V, 12V, και 24V. Διακρίνονται σε διπολικούς και μονοπολικούς και χρειάζονται απαραιτήτως την παρουσία ηλεκτρικού κυκλώματος για την εκκίνηση των πηνίων.

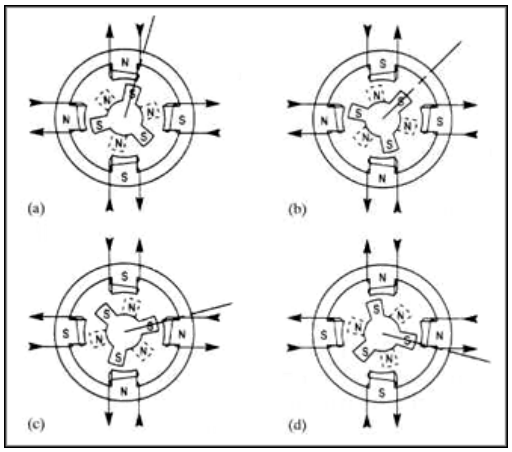
4.5.2 Βασική δομή ενός βηματικού κινητήρα

Ένας **βηματικός κινητήρας** είναι ένας κινητήρας που ελέγχεται από μια σειρά από ηλεκτρομαγνητικά πηνία. Ο άξονας κέντρου έχει μια σειρά από μαγνήτες που είναι στερεωμένοι σε αυτόν, και τα πηνία γύρω από την άτρακτο εναλλάξ δίνουν συνεχές ρεύμα ή όχι, και δημιουργούν μαγνητικά πεδία τα οποία απωθούν ή έλκουν τους μαγνήτες πάνω στον άξονα, προκαλώντας τον κινητήρα να περιστρέφεται. Οι **βηματικοί κινητήρες** λειτουργούν είτε με ρότορα μόνιμου μαγνήτη είτε με μεταβλητή αντίσταση. Στους πρώτους ο στάτορας αποτελείται από πόλους με περιέλιξη, ενώ οι πόλοι του ρότορα είναι μόνιμοί μαγνήτες και η διέγερση διαφορετικών συνδυασμών περιελίξεων μετακινεί και σταθεροποιεί τον ρότορα σε διαφορετικές θέσεις. Αντίστοιχα ένας **βηματικός κινητήρας** μεταβλητής αντίστασης έχει φερρομαγνητικό ρότορα, αντί για ρότορα μονίμων μαγνητών και ανάλογα με την έλξη μεταξύ ρότορα και στάτορα είτε ο κινητήρας κινείται είτε σταθεροποιείται.



Εικόνα 4.5.2 Παραδείγματα διάταξης των πηνίων στους βηματικούς κινητήρες

Για την κατά βήματα περιστροφή ενός **βηματικού κινητήρα**, θεωρούμε μια απλή διάταξη από τέσσερις πόλους στάτορα και έναν ρότορα μόνιμου μαγνήτη. Όπως είναι εμφανές στο παρακάτω σχήμα (Εικ.4.5.2.2) στην αρχική θέση (a) ο ρότορας είναι σε ισορροπία, γιατί οι αντίθετοι πόλοι του στάτορα και του ρότορα είναι απέναντι και έλκονται. Για την μετάβαση στο επόμενο βήμα (b), αλλάζει η πολικότητα του στάτη, εφαρμόζεται ροπή στον ρότορα και κινείται ωρολογιακά κατά 90 μοίρες, σε μια νέα θέση ισορροπίας (c). Όταν ξανά μεταβληθούν οι πολικότητες του στάτορα, ο ρότορας υπόκειται σε ροπή που τον μεταφέρει στο επόμενο βήμα (d).



Εικόνα 4.5.2.2 Ακολουθία βηματικού κινητήρα

4.5.3 Έλεγχος ενός βηματικού κινητήρα

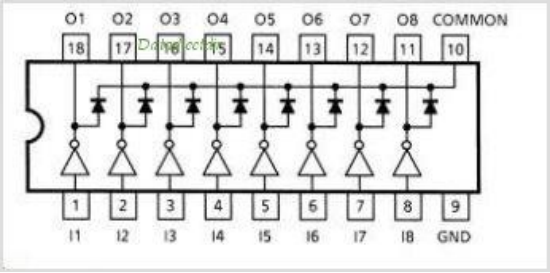
Οι **βηματικοί κινητήρες** χρειάζονται ψηφιακά κυκλώματα για την εκκίνηση των διαδικασιών ενεργοποίησης των πηνίων ώστε να περιστραφεί ο κινητήρας, η ανάδραση δεν είναι πάντα απαραίτητη, με τη χρήση όμως ενός κωδικοποιητή ή κάποιου αισθητήρα σε εφαρμογές που ο έλεγχος θέσης είναι βασικός εξασφαλίζει ακρίβεια. Το πλεονέκτημα στη λειτουργία χωρίς ανάδραση είναι ότι δε χρειάζεται το σύστημα ελέγχου κλειστού βρόγχου και καθώς αυτού του είδους οι κινητήρες αποδίδουν λιγότερο από 1 hp (746 W) χρησιμοποιούνται μόνο σε εφαρμογές ελέγχου θέσης χαμηλής ισχύος. Ο ρυθμός και ο αριθμός των ψηφιακών παλμών είναι οι παράμετροι που ελέγχουν την ταχύτητα και τη θέση του κινητήρα.

4.6 Το ολοκληρωμένο κύκλωμα «ULN2803» («Integrated circuit»)

Το **ολοκληρωμένο κύκλωμα «ULN2803»** χρησιμοποιείται για την οδήγηση των πηνίων του βηματικού κινητήρα παρέχοντας το απαραίτητο ρεύμα. Περιέχει μια σειρά από τρανζίστορ, όπου το καθένα μπορεί να λειτουργήσει με μέγιστο ρεύμα 500mA. Στο παρακάτω σχήμα (Εικ.4.6.2) φαίνετε το σχηματικό του **ολοκληρωμένου κυκλώματος**

«ULN2803» της εταιρείας «Toshiba». Ο τρόπος συνδεσμολογίας του **ολοκληρωμένου κυκλώματος «ULN2803»** με τον κινητήρα φαίνετε στην εικόνα 4.7.2.

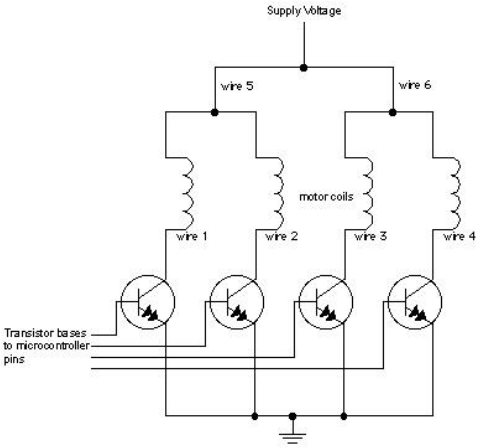
Εικόνα 4.6.1 Ολοκληρωμένο κύκλωμα "Toshiba ULN2803"



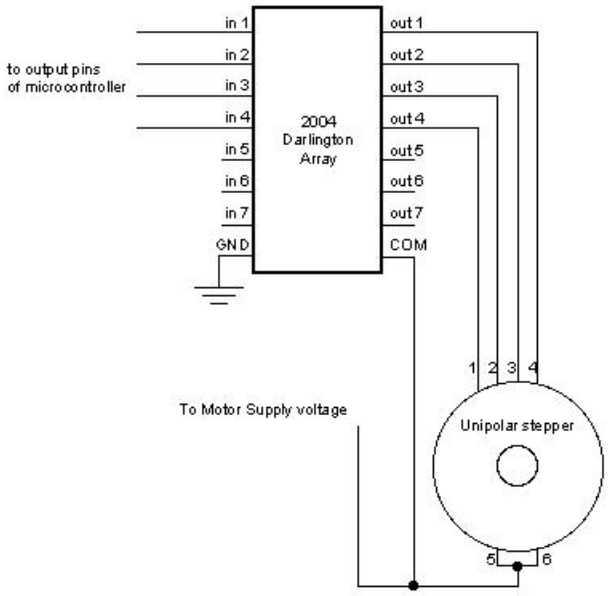
Εικόνα 4.6.2. Σχηματικό διάγραμμα ολοκληρωμένου κυκλώματος "Toshiba ULN2803"

4.7.1 Μονοπολικοί («Unipolar») βηματικοί κινητήρες

Οι **μονοπολική** («unipolar») βηματικοί κινητήρες απαιτούν μόνο μία πηγή τάσης, έχουν πέντε ή έξι καλώδια, τέσσερα πηνία (στην πραγματικότητα δύο πηνία διαιρούμενα με συνδέσεις στο κέντρο του κάθε πηνίου). Οι συνδέσεις βρίσκονται στο κέντρο των πηνίων τα οποία είναι δεμένα μεταξύ τους ώστε να χρησιμοποιηθούν ως σύνδεση ισχύος.



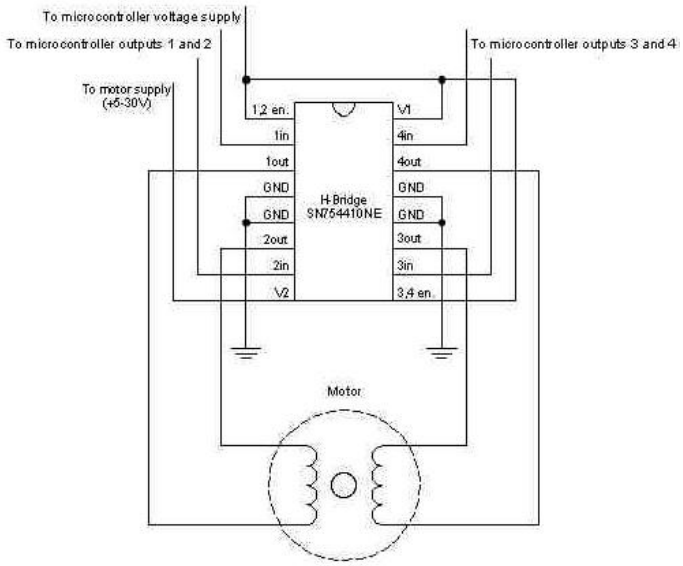
Εικόνα 4.7.1 Διάταξη των σπειρών σε ένα μονοπολικό βηματικό κινητήρα



Εικόνα 4.7.2 Σχηματικό διάγραμμα σύνδεσης ενός μονοπολικού (Unipolar) βηματικού κινητήρα με ένα ολοκληρωμένο κύκλωμα

4.7.2 Διπολικοί («Bipolar») βηματικοί κινητήρες

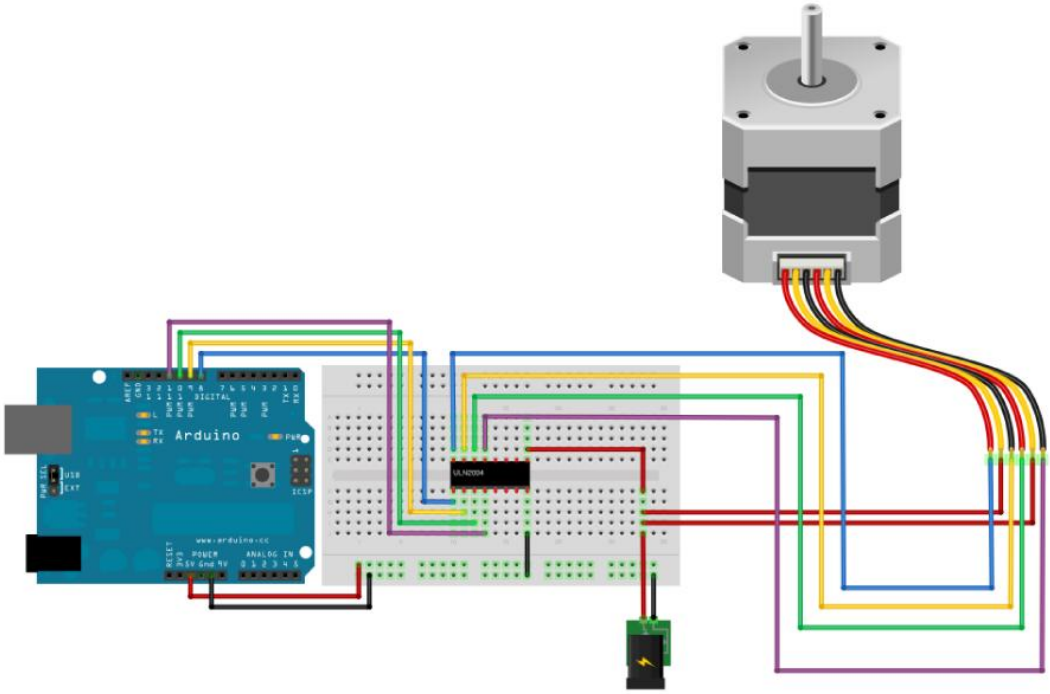
Οι **διπολικοί** («bipolar») βηματικοί κινητήρες απαιτούν δύο πηγές τάσης ή μία πηγή τάσης. Έχουν συνήθως τέσσερα καλώδια και σε αντίθεση με τους μονοπολικούς βηματικους κινητήρες, δεν έχουν κοινή σύνδεση στο κέντρο αλλά έχουν δύο ανεξάρτητα πηνία αντ'αυτού. Η διάκριση μεταξύ τους γίνεται μετρώντας την αντίσταση μεταξύ των συρμάτων[14].



Εικόνα 4.7.3 Σχηματικό διάγραμμα σύνδεσης ενός διπολικού (Bipolar) βηματικού κινητήρα με ένα ολοκληρωμένο κύκλωμα

4.8 Σύνδεση υπολογιστικής πλατφόρμας Arduino Uno με τον κινητήρα τύπου «Stepper»

Όπως θα γίνει αντιληπτό παρακάτω η συνδεσμολογία και ο κώδικας των παρελκόμενων πέρα του κινητήρα (οθόνη, αισθητήρας φωτός) είναι ακριβώς ίδια, για αυτό θα επικεντρωθούμε στην σύνδεση του βηματικού κινητήρα. Στην διάταξη μας χρησιμοποιήσαμε έναν μονοπολικό βηματικό κινητήρα («Unipolar Stepper»), έχοντας ως βάση τη βιβλιοθήκη των βηματικών κινητήρων του Arduino, για την οποία έχουμε αναφερθεί αναλυτικά στο Κεφάλαιο 1(1.12.3), και χρησιμοποιώντας ένα αναλογικό περιστροφικό ποτενσιόμετρο για να δώσουμε την επιθυμητή τιμή της γωνίας καταλήξαμε στην παρακάτω συνδεσμολογία:



Εικόνα 4.8 Σύνδεση ενός μονοπολικού («Unipolar») βηματικού κινητήρα με την υπολογιστική πλατφόρμα Arduino Uno(Εικ.4.7.2)

4.9 Παράδειγμα κώδικα για την λειτουργία ενός βηματικού κινητήρα

Το παρακάτω παράδειγμα περιστρέφει ένα βηματικό κινητήρα **100 βήματα** προς μια κατεύθυνση, έπειτα **100 βήματα** προς την αντίθετη κατεύθυνση και απευθύνεται και σε μονοπολικούς και σε διπολικούς βηματικούς κινητήρες. Χρησιμοποιεί την βιβλιοθήκη των κινητήρων «Stepper» που είναι προεγκατεστημένη στο **Arduino IDE**.

```
Stepper myStepper(motorSteps, motorPin1,motorPin2,motorPin3,motorPin4);

//δήλωση των PIN που συνδέεται ο Stepper στον Arduino(τα
motorPin3,motorPin4 απευθύνονται για μονοπολικούς βηματικούς κινητήρες)

#include <Stepper.h>//δήλωση της βιβλιοθήκης Stepper

#define motorSteps 200 //τιμή των βημάτων ανάλογα με τον κινητήρα

#define motorPin1 8//καθορισμός του PIN 1 του κινητήρα στο PIN 8 του
Arduino

#define motorPin2 9//καθορισμός του PIN 2 του κινητήρα στο PIN 9 του
Arduino

#define motorPin3 10//καθορισμός του PIN 3 του κινητήρα στο PIN 10 του
Arduino(για μονοπολικούς βηματικούς κινητήρες)

#define motorPin4 11//καθορισμός του PIN 4 του κινητήρα στο PIN 11 του
Arduino(για μονοπολικούς βηματικούς κινητήρες)

#define ledPin 13//δήλωση του LED 13

Stepper myStepper(motorSteps, motorPin1,motorPin2);//ρύθμιση της
βιβλιοθήκης Stepper

void setup() {

myStepper.setSpeed(60);//δήλωση της ταχύτητας του κινητήρα στις 60rpm

Serial.begin(9600);//εκκίνηση της σειριακής επικοινωνίας

pinMode(ledPin, OUTPUT);// δήλωση του LED 13 ως έξοδο

blink(3); }//αναβοσβήνει το LED

void loop() {

Serial.println("Forward");//εκτύπωση μηνύματος στη σειριακή οθόνη

myStepper.step(100);//δήλωση των θετικών επιθυμητών βημάτων

delay(500);

Serial.println("Backward");// εκτύπωση μηνύματος στη σειριακή οθόνη

myStepper.step(-100);// δήλωση των αρνητικών επιθυμητών βημάτων

delay(500); }
```

```
void blink(int howManyTimes) {

int i;//δήλωση της μεταβλητής int

for (i=0; i< howManyTimes; i++) { //βρόγχος ελέγχου for

digitalWrite(ledPin, HIGH);//δήλωση του LED HIGH

delay(200);

digitalWrite(ledPin, LOW); // δήλωση του LED LOW

delay(200);

}}
```

4.10 Ο κώδικας για την λειτουργία όλων των παρελκόμενων της κατασκευής με τον κινητήρα τύπου «Stepper».

Το παρακάτω παράδειγμα αναφέρεται στο κώδικα που δημιουργήσαμε για την λειτουργία όλων των παρελκόμενων της διάταξης με τον κινητήρα τύπου «Stepper» ώστε ο χρήστης να δίνει μια επιθυμητή θέση σε μοίρες (0-360), μέσω του αναλογικού περιστροφικού ποτενσιόμετρου, και πατώντας το μπουτόν ο βραχίονας να κινείται σε αυτήν. Η επιθυμητή τιμή ορίζεται στον κώδικα ως «theta» και για τον υπολογισμό της πολλαπλασιάζουμε το πλήθος των βημάτων του κινητήρα με το 7.5, που είναι η αντιστοιχία του ενός βήματος σε μοίρες (σύμφωνα με τον κατασκευαστή), και το διαιρούμε με το 1.56, που είναι η σχέση μετάδοσης των γραναζιών.

```
#include <Stepper.h>//δήλωση της βιβλιοθήκης Stepper

#include <Bounce.h>//δήλωση της βιβλιοθήκης Bounce

#define motorSteps 48//τιμή των βημάτων ανάλογα με τον κινητήρα

#define motorPin1 8//καθορισμός του PIN 1 του κινητήρα στο PIN 8

#define motorPin2 9//καθορισμός του PIN 2 του κινητήρα στο PIN 9

#define motorPin3 10//καθορισμός του PIN 3 του κινητήρα στο PIN 10

#define motorPin4 11//καθορισμός του PIN 4 του κινητήρα στο PIN 11

int stepamount;//δήλωση της μεταβλητής stepamount

int e,newPos;//δήλωση των μεταβλητών e, newPos
```

```
float theta; //δήλωση της μεταβλητής theta

int oldPos=0; //δήλωση της μεταβλητής oldPos

volatile int state = LOW;//δήλωση της μεταβλητής state

Stepper myStepper(motorSteps, motorPin1,motorPin2,motorPin3,motorPin4);
//δήλωση των PIN που συνδέεται ο Stepper στον Arduino

Bounce bouncer = Bounce( state,5 );//αρχικοποίηση της βιβλιοθήκης Bounce

void setup(){

myStepper.setSpeed(60);//δήλωση της ταχύτητας του κινητήρα στις 60rpm

attachInterrupt(0, stepping, RISING);} //δήλωση της συνάρτησης του
                                        //interrupt (INT0)
void loop(){

if (state==HIGH){//συνθήκη if

newPos = map(analogRead(A0), 0, 1023, 0, 75);//αντιστοίχηση της

                                      //μεταβλητής newPos από 0-1023 σε 0-75
stepamount = newPos-oldPos;

oldPos=newPos; //ισότητα των μεταβλητών oldPos και newPos

myStepper.step(stepamount);  //δήλωση των βημάτων

state=LOW;    //δήλωση της μεταβλητής state ως LOW

bouncer.update ( );  //ενημέρωση του debouncer

theta = oldPos* 7.5/1.56; // υπολογισμός της επιθυμητής τιμής

}}

void stepping () //συνάρτηση του interrupt

{state=!state;}
```

Κεφάλαιο 5

Συμπεράσματα και μελλοντική επέκταση

Στην παρούσα πτυχιακή εργασία αναλύθηκε η λειτουργία της υπολογιστικής πλατφόρμας Arduino, για την επέκταση των δυνατοτήτων του εκπαιδευτικού πακέτου «LEGO-MINDSTORMS» χρησιμοποιώντας παράλληλα άλλα δυο διαφορετικά ειδή επενεργητών (RC-Servo, Stepper) για τη δημιουργία τριών διαφορετικών εκπαιδευτικών διατάξεων. Παρότι η λειτουργία και των τριών διατάξεων είναι απόλυτα ακριβείς, η απουσία ελέγχου στις διατάξεις με τον κινητήρα «RC-Servo» και «Stepper» καθιστά τη διάταξη με τον DC κινητήρα του «NXT» ως την ιδανική για περαιτέρω επέκταση και εκπαιδευτικούς πειραματισμούς. Αυτό ενισχύεται με την χρήση του ελεκτή PID για την απαλοιφή των σφαλμάτων και των ταλαντώσεων στο κινητήρα. Έχοντας ως βάση την υπολογιστική πλατφόρμα Arduino της οποίας οι δυνατότητες πλεονάζουν, μπορούν να δημιουργηθούν διάφορες παραλλαγές και βελτιώσεις στις διατάξεις, όπως:

- Προσθήκη διάφορων αισθητηρίων (αισθητήρας απόστασης, υπερήχων κ.α)
- Η είσοδος της επιθυμητής τιμής να δίνεται είτε μέσω ενός πληκτρολογίου είτε μέσω ασύρματης επικοινωνίας («bluetooth»)
- Προσθήκη ενός ακόμα βαθμού ελευθερίας στο βραχίονα
- Προσθήκη κωδικοποιητών στις διατάξεις με τον κινητήρα RC-Servo και Stepper
- Αλλαγή του μικροελεγκτή χρησιμοποιώντας κάποιον με πιο πολλές δυνατότητες
- Προσθήκη ενός ακόμα κινητήρα ή τον DC του NXT ή τον RC-Servo ή τον Stepper

Βιβλιογραφία

[1] web: http://arduino.cc/

[2] web: http://en.wikipedia.org/wiki/Arduino

[3] web: http://playground.arduino.cc/

[4] web: http://en.wikipedia.org/wiki/Laser_harp

[5] web: http://diydrones.com/profiles/blogs/ardupilot-main-page

[6] web: http://madlabindustries.com/hackerspace/projects-2/hexapod-quadcopter/

[7] Mindstorms Education, NXT user guide (www.MINDSTORMSeducation.com)

[8] web : http://en.wikipedia.org/wiki/Lego_Mindstorms

[9] web: http://en.wikipedia.org/wiki/Pulse-width_modulation

[10] web: http://en.wikipedia.org/wiki/PID_controller

[11] web: http://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method

[12]web: http://www.philohome.com/

[13] Δρ. Φασουλάς Ιωάννης, Σημειώσεις Ρομποτικής, σχολή Σ.Τε.Φ. τμήμα Μηχανολογίας Τ.Ε.Ι. Κρήτης, Ηράκλειο 2010

[14] Δρ. Αχιλλέας Βαΐρης,
Σημειώσεις Στοιχεία Μηχανών, Τμήμα Μηχανολογίας Τ.Ε.Ι. Κρήτης, Ηράκλειο.

[15] Καλαντζής Σ. Σπυρίδων
Πτυχιακή εργασία: Σχεδιασμός, κατασκευή και έλεγχος μικρής ρομποτικής πλατφόρμας "Stewart"