

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ



**ΑΝΑΠΤΥΞΗ ΚΑΙ ΥΛΟΠΟΙΣΗ ΑΝΑΣΤΡΟΦΟΥ ΕΚΚΡΕΜΟΥΣ ΜΕ
ΤΗΝ ΧΡΗΣΗ ΥΔΙΣΜΙΚΟΥ ΑΝΟΙΧΤΗΣ ΠΗΓΗΣ**

ΣΠΥΡΟΣ ΤΣΟΠΑΝΑΚΗΣ

ΔΗΜΗΤΡΙΟΣ ΠΥΡΟΜΑΛΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΙΑΝΟΥΑΡΙΟΣ 2019

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

Π. Ράλλη & Θηβών 250, 12244 Αιγάλεω, Αθήνα – Ελλάδα
Τηλ. 210-5381488

Εγκρίθηκε από την τριμελή επιτροπή στις

.....

(Υπογραφή)

.....

(Υπογραφή)

.....

(Υπογραφή)

Πίνακας Περιεχομένων

Κεφαλαιο1:Εισαγωγή.....	10
1.1:Βιβλιογραφική Ανασκόπηση.....	10
1.2:Αρχη Λειτουργίας ανάστροφου εκκρεμούς.....	11
1.3:Μοντέλο ανάστροφου εκκρεμούς.....	12
Κεφαλαιο2:Υλικά και εξαρτήματα εκκρεμούς.....	13
2.1: Arduino.....	13
2.1.1:Η πλακέτα του Arduino.....	14
2.1.2: Είσοδοι – έξοδοι.....	16
2.1.3:Τροφοδοσία.....	18
2.1.4 Διακόπτες και ενδικτεικές φωτοδύοδοι της πλακέτας..	19
2.1.5:Το περιβάλλον arduino ide.....	20
2.1.6:Γλώσσα προγραμματισμού.....	22
2.2:Κινητήρες.....	24
2.2.1:Εισαγωγή.....	24
2.2.2:Αρχή Λειτουργίας.....	25
2.2.3:Κατηγορίες.....	26
2.2.4:Κινητήρες DC.....	27
2.3: Μπαταρίες.....	29
2.3.1:Εισαγωγή.....	29
2.3.2:Τα είδη των μπαταριών.....	29
2.3.1:Powerbank.....	30
2.4: Adafruit LSM9DS1 Αξελερόμετρο- γυροσκόπιο..	33
2.4.1:Γενικά.....	33
2.5: Drivers.....	37
2.5.1:Περίληψη.....	37
2.6:Ρόδες.....	40
2.6.1Εισαγωγή.....	40
2.6.2:Λειτουργία και χρήσεις.....	41
2.7:Η ξύλινη κατασκευή.....	42
2.7.1:Breadboard.....	43
Κεφάλαιο 3:Σχεδίαση του ελεγκτή.....	44
3.1 PID ελεγκτής.....	44
3.2.Γέννοια της ανάδρασης.....	44
3.3.1Τι είναι η ανατροφοδότηση.....	44
3.4.ΓΈλεγχος - PID ελεγκτής.....	45
Κεφαλαιο4:Προγραμματισμός.....	52
4.1:Κώδικας.....	52
Κεφαλαιο5:Αποτελέσματα.....	62

5.1 : Γραφήματα.....	62
5.2: Συμπεράσματα-μελλοντική έρευνα.....	66
Κεφάλαιο6: Βιβλιογραφία.....	69

Ευχαριστίες

Θέλω να δώσω τις ευχαριστίες στον κύριο Δημήτριο Πυρομάλη διότι με βοήθησε και με υποστήριξε σε αυτήν την πτυχιακή. Ακόμα θα ήθελα να ευχαριστήσω τους συγγενείς μου και τον αδελφό μου σ'αυτήν την διάρκεια υλοποίησης της πτυχιακής.

Περίληψη

Η πτυχιακή εργασία είναι το τελικό αποτέλεσμα των σπουδών μου στο Πανεπιστήμιο Δυτικής Αττικής. Οπότε με τις γνώσεις μου που αποκόμισα και την βοήθεια του κύριου Δημήτριου Πυρομάλη υλοποίησα το ανάστροφο εκκρεμές. Το ανάστροφο εκκρεμές είναι ένα όχημα με δύο ρόδες που ισορροπεί. Για να ισορροπήσει χρειάζεται να εκτελεστεί πρόγραμμα στον μικροεπεξεργαστή. Επίσης χρειάζεται και το υλισμικό κομμάτι ώστε να πετύχουμε ισορροπία στο σύστημα μας. Όπως ξύλινη βάση, ρόδες, καλώδια, servos, bradboard αυτά βοηθούν ώστε να πετύχουμε ισορροπία.

Abstract

The thesis is the final result of my studies at the University of Western Attica. So with my knowledge that was obtained and assisted by Mr. Dimitrios Pyromalis the reverse pendulum. The reverse cam is a two-wheeled vehicle balancing wheels. To balance it needs to be executed program in microprocessor. Hardware is also needed piece to bring balance to our system. Like wooden base, wheels, cables, servers, bradboard they help in order to strike a balance.

Κεφαλαίο1

Εισαγωγή

Στόχος της πτυχιακής είναι να μπορέσουμε να ισορροπήσει το όχημα στις δύο ρόδες. Παρακάτω θα δούμε ότι χρειάζονται κάποιες διαδικασίες . Όπως να προσαρμόσουμε διάφορα εξαρτήματα ώστε να μπορέσει να ισορροπήσει. Επιπρόσθετα για να μπορέσει να ισορροπήσει χρειάζεται και προγραμματιστικό κομμάτι που θα δούμε παρακάτω.

1.1 Βιβλιογραφική ανασκόπηση

Η κατασκευή αυτή θα μπορούσε να γίνει σε μεγαλύτερη κλίμακα . Όμως δεν υλοποιήθηκε λόγω κόστους και τεχνογνωσίας. Θα είχα φτιάξει ένα πατίνι με δύο ρόδες ώστε ο επιβάτης να μπορεί να ισορροπήσει στα δύο πατάκια του οχήματος. Ο έλεγχος του οχήματος θα γινόταν κουνώντας τα τακάκια του οχήματος και η ταχύτητα θα γινόταν με το χέρι αυξομειώνοντας τον λεβιέ που είναι τοποθετημένος στο τιμόνι

1.2 Αρχή λειτουργίας του ανάστροφου εκκρεμούς

Για να ισορροπήσει ένα σώμα θα πρέπει από το σημείο Α να μπορέσει να ισορροπήσει σε ένα σημείο Β. Το εκκρεμές θα πρέπει να είναι σε ισορροπία έτσι ώστε να δημιουργήσουμε τόξο εφαπτομένης και αδράνεια. Η ταλάντωση που θα του δώσουμε θα πρέπει να ξεπερνάει το σημείο ΑΟ και αντίστοιχα το σημείο ΒΟ για να πετύχουμε την ταλάντωση που θέλουμε. Έτσι η ταλάντωση έχει περίοδο Τ. Έτσι φαίνεται ο τύπος παρακάτω[1]:

$$T_0 = 2\pi\sqrt{\frac{l}{g}}$$

Η μάζα του εκκρεμές καθορίζει την ταλάντωση του. Έτσι με αυτόν τον τρόπο μπορούμε βρούμε την περίοδο δηλαδή το πλήθος των ταλαντώσεων[2]

$$g = \frac{4\pi^2 L}{T^2}$$

1.3 Μοντέλο ανάστροφου εκκρεμούς

Ο PID έλεγχος καθορίζει την ισορροπία του οχήματος γιατί το σύστημα είναι μη γραμμικό. Έτσι δημιουργήσαμε ένα ελεγκτή ανάδρασης ώστε το σύστημα να γίνει γραμμικό. Έχουμε παρακάτω κάποιες εξισώσεις του PID ώστε το σύστημα να γίνει γραμμικό.

$$h_1 = \frac{3}{4l}g\sin\theta \quad (\text{Εξ.1}) \qquad h_2 = \frac{3}{4l}\cos\theta \quad (\text{Εξ.2})$$

$$f_1 = m \left(l\sin\theta\dot{\theta}^2 - \frac{3}{8}g\sin 2\theta \right) - f\dot{x} \quad (\text{Εξ. 3})$$

$$f_2 = M + m \left(1 - \frac{3}{4}\cos^2\theta \right) \quad (\text{Εξ. 4})$$

$$u = \frac{f_2}{h_2} \left[h_1 + k_1(\theta - \theta_d) + k_2\dot{\theta} + c_1(x - x_d) + c_2\dot{x} \right] - f_1 \quad (\text{Εξ. 5})$$

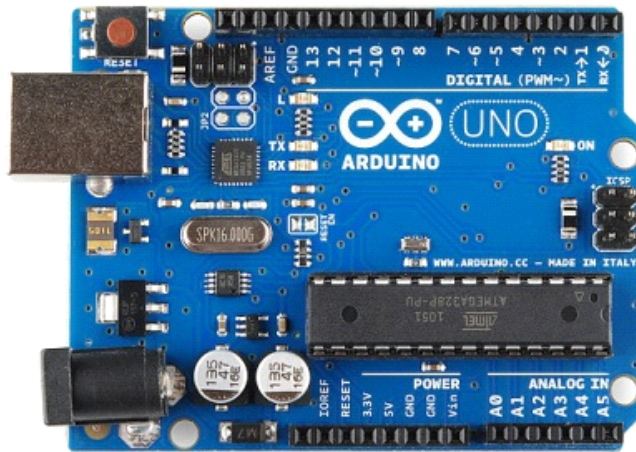
Για τις προσομοιώσεις δίνονται οι τιμές M, μ, l, g του μοντέλου εκκρεμούς. Οι παρακάτω αριθμητικές τιμές είναι: $M=1,2\text{kg}, l=0,4\text{ m}, g=9,81\text{m/s}^2, m=0,1\text{kg}, k_1=25, k_2=10, c_1=1, c_2=2,6$. [3]

Κεφάλαιο 2:

Υλικά και εξαρτήματα εκκρεμούς

2.1 Arduino

Η επιλογή του Arduino Uno έγινε διότι υπάρχει μια αναπτυξιακή μονάδα όπου μπορείς να ανεβάσεις πρόγραμμα. Επίσης είναι η πιο γνωστή πλακέτα με ποικίλες εφαρμογές. Παρακάτω θα αναφερθούμε αναλυτικά για το Arduino Uno.



Η πλακέτα του arduino

Το Arduino είναι ένας single-board τύπου μικροελεκτης, με λίγα λόγια αποτελείται από μια απλή μητρική πλακέτα με έναν ενσωματωμένο μικροελεκτη, ο οποίος προγραμματίζεται με τη χρήση της γλώσσας wiring. Ουσιαστικά η γλώσσα wiring πρόκειται για τη γνωστή γλώσσα προγραμματισμού C++ και αρκετές βιβλιοθήκες, οι οποίες έχουν υλοποιηθεί σε αυτή.

Ο μικροελεκτης από τον οποίο αποτελείται ένα Arduino όπως αναφέρθηκε παραπάνω, ανήκει στη σειρά των μικροελεκτών της εταιρίας Atmel, AVR. Ανάλογα με το μοντέλο του Arduino, χρησιμοποιείται και διαφορετικό μοντέλο μικροελεκτη, κατάλληλο για την εκάστοτε εφαρμογή.

Κάθε μητρική πλακέτα του Arduino ενσωματώνει έναν voltage regulator στα 5 Volt, για τη σωστή λειτουργία του μικροελεκτη και όλων των υπόλοιπων υποσυστημάτων, καθώς επίσης και έναν κρυσταλλικό ταλαντωτή στα 16MHz.

Τα I/O pins του Arduino μπορούν να λάβουν ψηφιακά, αλλά και αναλογικά δεδομένα, καθώς επίσης μπορούν να παράγουν και pulse-width διαμορφωμένα σήματα. Αυτά τα pins είναι τοποθετημένα περιμετρικά της μητρικής πλακέτας του Arduino, καθώς η σύνδεση τους υλοποιείται με τη χρήση female headers 2.2mm.

Όλα τα Arduino boards προγραμματίζονται με τη χρήση μίας σειριακής σύνδεσης με έναν ηλεκτρονικό υπολογιστή, καθώς οι παλιότερες εκδόσεις χρησιμοποιούσαν μία RS-232 σειριακή σύνδεση, τα τωρινά Arduino boards προγραμματίζονται με τη χρήση μιας USB θήρας.

Το περιβάλλον ανάπτυξης του Arduino είναι σχεδιασμένο με τέτοιο τρόπο, ούτως ώστε να μπορεί να εισάγει στον προγραμματισμό άτομα που δεν είναι εξοικειωμένα με την ανάπτυξη λογισμικού. Για το λόγο αυτό, το πρόγραμμα επεξεργασίας κώδικα περιλαμβάνει επισήμανση σύνταξης, καθώς επίσης μεταγλωττίζει και φορτώνει τα προγράμματα στον μικροελεκτη, τα οποία

ονομάζονται sketch, με ένα απλό κλικ του ποντικιού. Επίσης το Arduino έρχεται με αρκετές έτοιμες βιβλιοθήκες λογισμικού, κάτι το οποίο καθιστά αρκετές κοινές λειτουργίες εισόδου και εξόδου του μικροελεγκτή πάρα πολύ πιο εύκολες στη διαχείριση.

Τέλος, όλα τα Arduino, καθώς επίσης και όλα τα υπόλοιπα συμβατά με το Arduino boards χρησιμοποιούν την τεχνολογία γνωστή ως shields. Τα shields είναι τυπωμένα επεκτάσιμα κυκλώματα σε μορφή board, τα οποία συνδέονται στα ήδη διαθέσιμα pins του Arduino. Τα επεκτάσιμα shields μπορεί να είναι είτε motor drivers, είτε GPS, είτε Ethernet, ή ακόμα και μία οθόνη LCD, καθώς επίσης με τη χρήση συγκεκριμένων prototyping shields, μπορούν υλοποιηθούν αρκετά ενδιαφέροντα DIY shields .

Το Arduino Uno έχει έναν μικροελεγκτή όπου είναι το βασικό στοιχείο της πλακέτας. Επίσης αποτελείται από leds, pins κ.α. που θα τα αναφέρουμε παρακάτω. Η πλακέτα καταναλώνει υπολογιστική ισχύς και μπορεί να συνδεθεί με τον υπολογιστή με καλώδιο usb. Ωστε να μεταφέρουμε το πρόγραμμα στην πλακέτα.



Σχήμα 2.2 - Η πλακέτα του μικροελεγκτή Arduino

Στην πλακέτα υπάρχει ένας Atmel AVR με 5v ρυθμιστή τάσης και bootloader. Επίσης συνδέεται με τον υπολογιστή μέσω θύρας usb. Υπάρχει ένας εναλλάκτης σύνδεσης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
TTL-FTDI όπου ρυθμίζει την τάση. Ο προγραμματισμός σήμερα γίνεται από τον
χειριστή με την βοήθεια του IDE περιβάλλοντος.

Είσοδοι – Έξοδοι

Έχουμε 14 ψηφιακές θύρες στο Arduino Uno εισόδου και εξόδου, επίσης έχουμε έξι αναλογικές εισόδους (analog input pins). Από 0 έως 13 ονομάζουμε τις 14 ψηφιακές θύρες με όνομα A ακολουθούμενο από ένα νούμερο από το 0 μέχρι το 5. Τα pins της εξόδου, επίσης μπορούν να μας δώσουν 0 έως 5 Volt τάση. Οι υπόλοιπες 14 ψηφιακές θύρες η 6 και οι παρακάτω ειδικότερα οι 3,5,6,9,10,11 είναι PWM θύρες, οι οποίες προσομοιώνουν αναλογικές εξόδους. Επομένως έχουμε για εισόδους και εξόδους θα δούμε παρακάτω:

Η ψηφιακή είσοδος, χρησιμοποιούνται οι 14 ψηφιακές 0...13. Οι οποίες δουλεύουν ψηφιακά, η τάση εισόδου μπορεί να είναι ή 0 ή 5v.

Η ψηφιακή έξοδος, χρησιμοποιείτε οι 14 ψηφιακές 0...13. Οι οποίες δουλεύουν ψηφιακά, τάση εξόδου μπορεί να είναι 0 ή 5 volt.

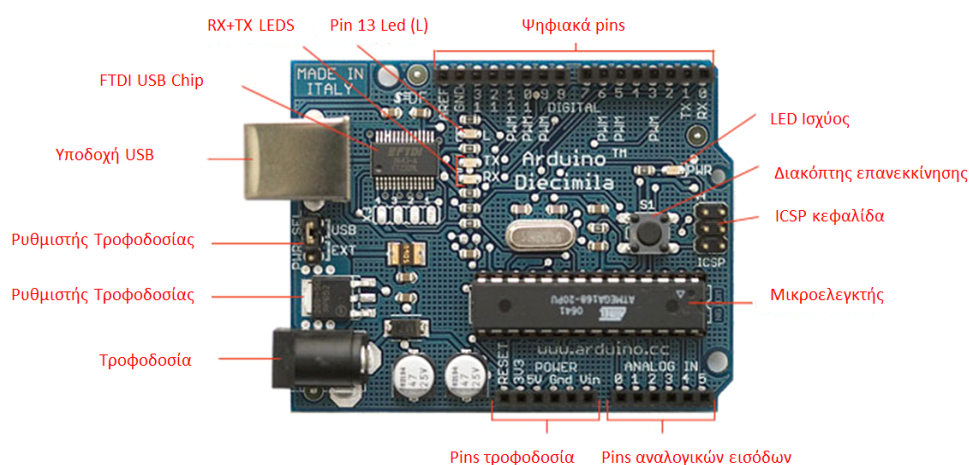
Οι αναλογικοί είσοδοι, χρησιμοποιούνται οι τιμές ρεύματος στο διάστημα 0 έως 5v.

Οι αναλογική έξοδος χρησιμοποιείτε για τις έξι PWM ψηφιακές θύρες (3,5,6,9,10,11), οι οποίες έχουν ρεύμα εξόδου τιμές 0 έως 5v.

Όταν γράφουμε κώδικα θα πρέπει να αποθηκεύουμε σε ποιες θύρες όπου θέλουμε με την συνάρτηση pinMode(), όπου δίνουμε μια πληροφορία για τις οποίες δουλεύουν αν είναι για είσοδο ή για έξοδο. Επίσης στη σειριακή είσοδο χρησιμοποιούνται εφαρμογές επικοινωνίας με τον υπολογιστή. Δηλαδή τα pins 0 και 1, όπου στις εφαρμογές, αντί να είναι απαραίτητο είναι υποχρεωτικό να αποθηκεύονται και να φτάνουν τα υπόλοιπα 12 pins στην

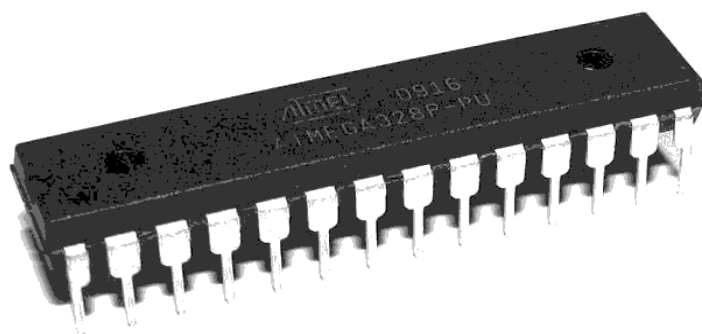
εφαρμογή μας. Επιπρόσθετα υπάρχει συνδεδεμένο ήδη ένα Led και χρησιμοποιείτε σε διάφορες λειτουργίες .

Η βασική έξοδος του μικροελεγκτή είναι θήρα USB. Ο χρήστης – προγραμματιστής στέλνει μέσω της θύρας πρόγραμμα έτοιμο προς χρήση. Επιπρόσθετα έχει και τα υπόλοιπα pins που συνδέονται είτε ως είσοδοι , είτε ως έξοδοι στο Arduino.



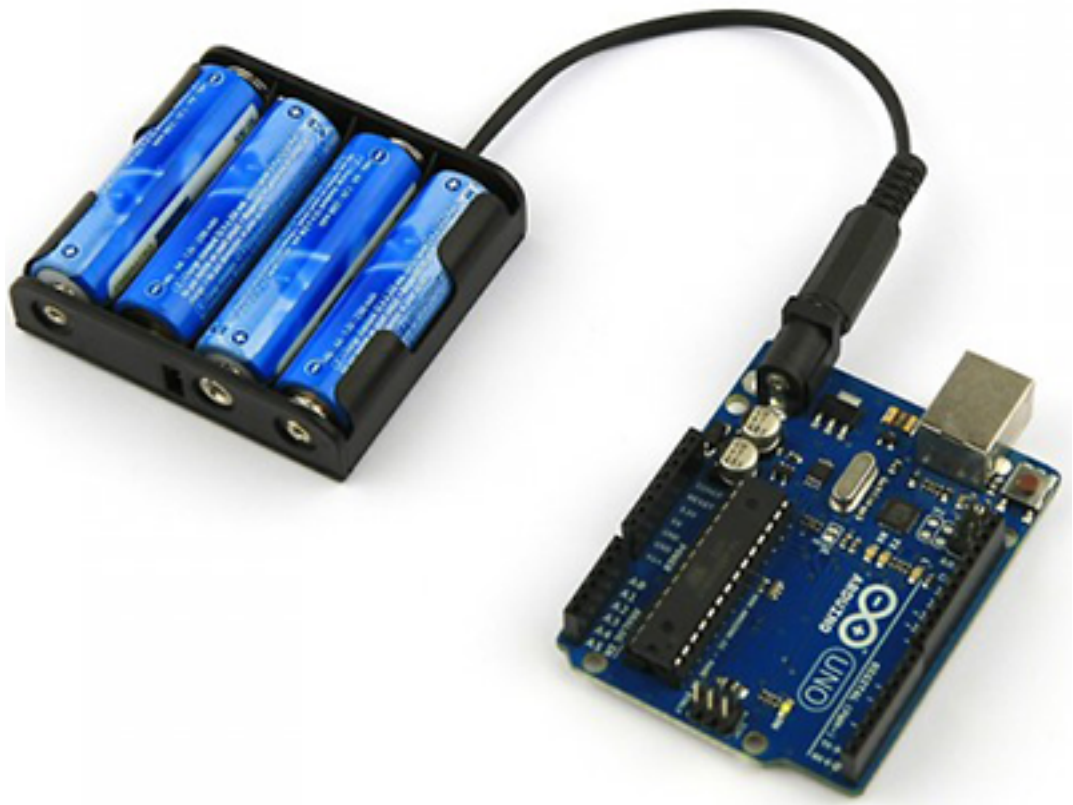
Σχήμα 2.3 - Η πλακέτα του μικροελεγκτή Arduino και με τα στοιχεία.

Ακόμα υπάρχουν στο Arduino 14 Pins όπου τα εξής Pins (0-13) είναι είσοδοι-έξοδοι. Λειτουργούν στα 40mA και έχουν μέγιστη τάση 5v. Επίσης υπάρχει η κατάσταση HIGH και LOW όπου ο προγραμματιστής μπορεί να ενεργοποιήσει τα Pins που θέλει. Έτσι μπορεί ο χειριστής να ανάψει ένα LED ή να ενεργοποιήσει ένα αισθητήριο όπου είναι συνδεδεμένο στο pin αυτό.



Τροφοδοσία

Η τροφοδοσία του Arduino βρίσκεται πάνω αριστερά μεριά. Τροφοδοτείται με 9-12Volts με την υπάρχουσα εσοχή .Επίσης έχει και άλλη μια εσοχή ,πέρα από αυτήν με την usb ,υπάρχει και μια άλλη που τροφοδοτείται με τροφοδοτικό .



Σχήμα 2.5 - Εδώ το Arduino τροφοδοτείτε με μπαταρίες

Επιπρόσθετα η τροφοδοσία μπορεί να γίνει είτε με μπαταρίες είτε με τροφοδοτικό ή με άλλες διαδικασίες. Εξίσου υπάρχουν και κάποια Pins που τροφοδοτούν το Arduino τα οποία είναι 6 σε αριθμό με ετικέτα Power:

Το πρώτο pin , είναι για την επανεκκίνηση του Arduino και έχει ετικέτα RESET.

Το δεύτερο pin, μπορεί να δώσει μέχρι 50mA όπου δίνετε από εξωτερική τροφοδοσία.

Το τρίτο pin, μπορεί να δώσει 5v είτε από τη τροφοδοσία είτε από τα pins με ρύθμιση να μειώνετε στα 5v.

Το τέταρτο pin, είναι οι γειώσεις με την ετικέτα GND.

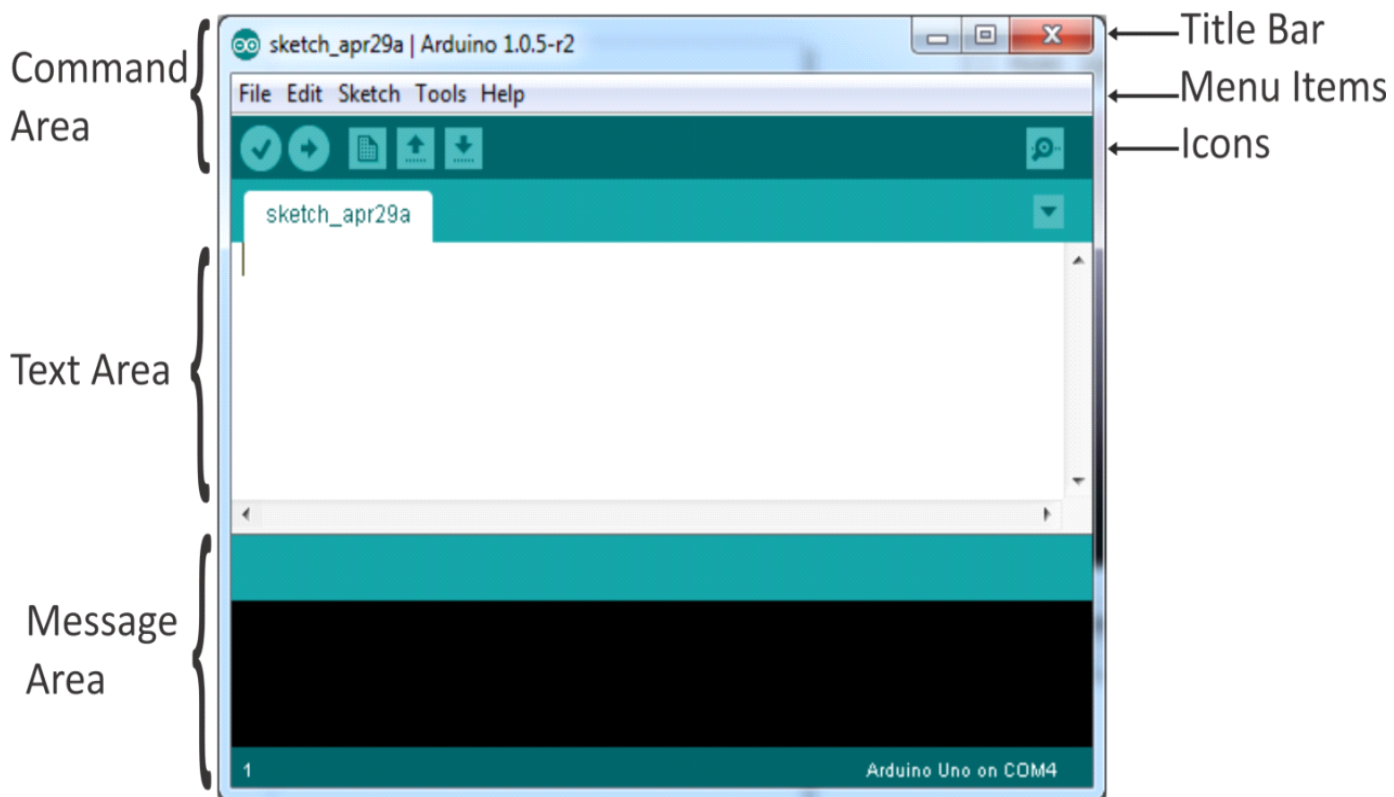
Το πέμπτο pin, είναι με την ετικέτα Vin, λειτουργεί με δύο τρόπους ,με την μέθοδο τροφοδοσίας και ως φως 2.1mm

Διακόπτες και ενδεικτικές φωτοдиодι της πλακέτας

Στην πλακέτα του Arduino υπάρχουν δύο LEDs .Τα LEDs αυτά είναι κατά βάση για εκπαιδευτικούς λόγους και είναι εσωτερικά συνδεδεμένα με το pin 13.Αλλιώς τα LEDs ονομάζονται και Serial-Over-Usb με την σύνδεση της θύρας Usb. Αυτά τα LEDs είναι στην πλακέτα ενσωματωμένα με ετικέτα POWER. Επίσης τα LEDs ονομάζονται Tx και Rx.

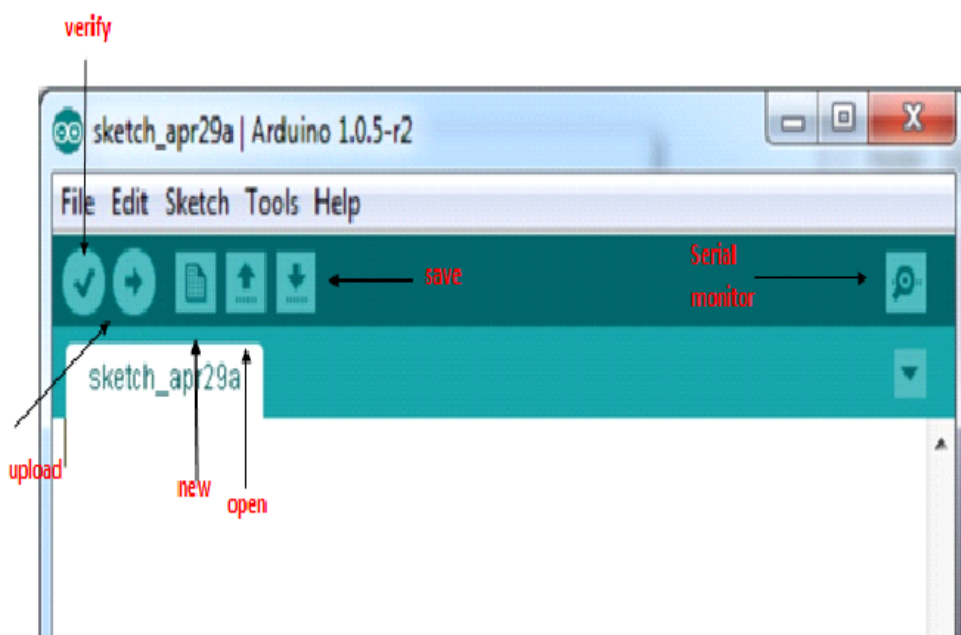
Το περιβάλλον arduino ide

Το περιβάλλον της IDE συνοδεύετε από το πρόγραμμα της Java.Όπου εκεί γράφτηκε το πρόγραμμα της πτυχιακής. Μέσα στο περιβάλλον της Arduino IDE υπάρχουν έτοιμες βιβλιοθήκες έτσι ώστε να κάνουν πιο εύκολο τον προγραμματισμό. Επίσης ο compiler χρησιμοποιείται για να γίνει η μεταγλώττιση. Ακόμα υπάρχει η σειριακή (Usb) που αναλαμβάνει τον προγραμματισμό στο Arduino για το debugging. Είναι ένα από τα απλούστερα περιβάλλοντα του Arduino IDE.



Σχήμα 2.6 - Περιβάλλον Arduino IDE

Παρακάτω βλέπουμε την εργαλειοθήκη του Arduino IDE όπου με αυτήν μπορούμε να χειριστούμε διάφορες λειτουργίες του προγράμματος. Επίσης μπορεί να γίνει ποιο εύκολος ο προγραμματισμός του Arduino με την βοήθεια της εργαλειοθήκης (toolbar).



Σχήμα 2.7 - Το toolbar του Arduino IDE

Ακόμα υπάρχουν και άλλα κουμπιά όπως το New. Όπου δημιουργεί νέο φάκελο αποθήκευσης ενός προγράμματος και σβήνει ότι άλλο έχει ο editor. Το Open όπου ανοίγουμε και άλλα προγράμματα. Το Save όπου με το κουμπί αυτό σώζουμε το πρόγραμμα. Το κουμπί Upload όπου ανεβάζουμε το πρόγραμμα στο Arduino Uno αφού ελεγχθεί μήπως υπάρχουν λάθη. Η

σύνδεση του Arduino Uno γίνεται μέσω της σειριακής σύνδεσης. Το κουμπί Verify όπου ο χειριστής μπορεί να κατασκευάσει το πρόγραμμα και να το επεξεργαστεί ανάλογα. Το Serial Monitor όπου υπάρχουν υπολογιστικά δεδομένα. Το κουμπί Import Library όπου επιλέγουμε βιβλιοθήκη. Το κουμπί Tools έχει την δυνατότητα να διαλέξει εργαλεία όπως θύρα και board. Επομένως έχουμε το Auto Format όπου μας δίνει τη δυνατότητα να διαβάσουμε καλύτερα τον κώδικα. Τέλος το κουμπί Burning Bootloader είναι κατά βάση για εξειδικευμένους προγραμματιστές που χρησιμοποιούν μνήμη στο bootloader.

Γλώσσα Προγραμματισμού

Το Arduino είναι βασισμένο σε ευέλικτο και κατανοητό σε χρήση του και το λογισμικό του. Αναφέρετε σε σχεδιαστές, υλοποίηση χόμπι. Για την ακρίβεια ,υπάρχει ένας μικροελεγκτής ,ο οποίος έχει πύλες οι οποίες λειτουργούν είτε ως είσοδοι είτε ως έξοδοι. Οι έξοδοι και οι είσοδοι , γράφοντας κώδικα στο περιβάλλον του Arduino IDE. Πέρα από το περιβάλλον του Arduino IDE που βασιστεί στην γλώσσα C/++ υπάρχει και μια παραλλαγή την Scratch , όπου μπορεί να γραφεί πρόγραμμα για το Arduino. Στο διαδίκτυο υπάρχουν και άλλες εκδόσεις όπως ο οπτικός προγραμματισμός. Επίσης το BlocklyDuino ή το ArduinoMio είναι άλλες δύο οπτικές εκδόσεις στο διαδίκτυο. Στο μικροελεγκτή θα αναφερθούμε στις επόμενες ενότητες αφού γίνει κατανοητή η λειτουργία του.

Η γλώσσα προγραμματισμού του Arduino είναι παρόμοια με την Writing εξίσου με την C++ . Στην γλώσσα αυτή έχουμε ίδιες όπως εντολές, συναρτήσεις και σταθερές. Επίσης στην γλώσσα του Arduino με την C++

έχουνε ίδιες εντολές ,συναρτήσεις και σταθερές. Ο compiler του Arduino είναι ο AVRgcc επομένως με βιβλιοθήκη C δίνετε η AVRlib. Οι εντολές που χρησιμοποιούνται πιο πολύ οι εξής παρακάτω:

Για να γίνει η αρχή του προγράμματος υπάρχουν δύο βασικές ρουτίνες όπου μπορεί να ξεκινήσει η βασική δομή στο περιβάλλον του IDE.

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

Σχημα 2.8 - Οι δυο βασικές ρουτίνες του Arduino IDE

2.3 Κινητήρες

Εισαγωγή

Έχουμε τρία είδη λειτουργίας ενός κινητήρα όπως, ως κινητήρας, ως πέδες και ως γεννήτρια. Θα αναφερθούμε παρακάτω για τα τρία είδη λειτουργίας. Η λειτουργία ως γεννήτρια είναι η αντίστροφη κίνηση από τη κανονική. Όπου αποθηκεύει με την κίνηση του, ηλεκτρική ενέργεια. Η ηλεκτρομαγνητική επαγωγή οφείλετε, η κύρια λειτουργία των ηλεκτρικών γεννητριών. Στους κινητήρες παρουσιάζεται ηλεκτρομαγνητική επαγωγή. Καθώς έχουμε διαρροή ρεύματος τότε παρατηρούμε κίνηση που οφείλετε στη δύναμη που ασκήθηκε στον ρότορα. Αν αλλάξουμε την φορά του ρεύματος τότε θα αλλάξει και η φορά της κίνησης του κινητήρα.

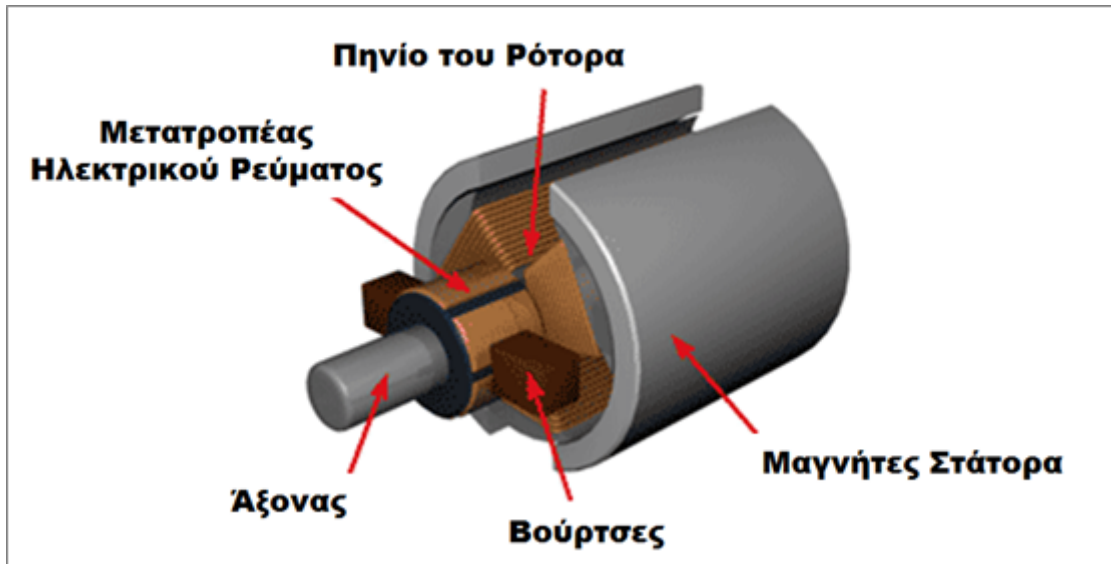
Ο κινητήρας αποτελείται από δίφορα μέρη:

Ρότορας: Καθώς γυρίζει ο ρότορας υπάρχει διαρροή ρεύματος μέσα στις ψήκτρες. Επίσης έχει διάφορα πηνία

Εναλλάκτης: Διαθέτει πολλά μεταλλικά ελάσματα που αποτελούνται από έναν κύλινδρο. Σε κάθε σύνδεση υπάρχουν ψήκτρες.

Στάτορας: Είναι δύο πηνία που διαρρέονται από ηλεκτρικό ρεύμα. Με αποτέλεσμα να έχουμε κίνηση στον ρότορα.

Ψήκτρα: είναι κατασκευασμένες από γραφίτη και καθορίζουν την σύνδεση του εναλλάκτη.



Αρχή λειτουργίας

Με την εξίσωση κίνησης μπορούμε να την χρησιμοποιήσουμε, σε προβλήματα σε στερεά σώματα που γυρνάνε γύρω από ένα σταθερό άξονα στο χώρο, με στόχο να προσδιορίσουμε την ποιότητα της περιστροφής σε σχέση με τον χρόνο και με τις ροπές που ασκούνται στο σώμα. Η ροπή αδράνειας ενός σωματιδίου, όπου εκτελεί κυκλική κίνηση ως προς ένα ακίνητο άξονα περιστροφής. Ένα άλλο αρκετά χρήσιμο θεώρημα τον υπολογισμό της ροπής αδράνειας επιπέδων σχημάτων, είναι το θεώρημα των κάθετων αξόνων. Βρίσκει εφαρμογή μόνο σε σώματα μορφής λεπτού φύλλου. Έστω J_x και J_y , οι ροπές αδράνειας του λεπτού φύλλου ως προς τους άξονες Ox και Oy αντίστοιχα. Ορίσαμε τη δύναμη ως το χρονικό ρυθμό μεταβολής της ορμής. Για σώματα με σταθερή μάζα, προέκυψε η γνωστή έκφραση $F=m \cdot a$. Όμως η ορμή ενός σώματος είναι το γινόμενο της μάζας του επί την ταχύτητα του. Επομένως η γενική έκφραση της δύναμης που δρα επάνω στο

σώμα, στην περίπτωση που υπάρχει χρονική μεταβολή της μάζας του είναι η μεταβολή της ταχύτητας από την αρχική στην τελική κατάσταση.

Κατηγορίες

Έχουμε δύο κατηγορίες κινητήρων τους κινητήρες εναλλασσόμενου ρεύματος, όπου χρησιμοποιούνται περισσότερο στην αγορά. Μονοφασικούς και πολυφασικούς ονομάζονται οι κινητήρες εναλλασσόμενου ρεύματος. Σε σύγχρονους και ασύγχρονους χωρίζονται οι μονοφασικοί και οι πολυφασικοί. Τέσσερις κατηγορίες κινητήρων εναλλασσόμενου ρεύματος είναι κατά βάση οι πολυφασικοί κινητήρες. Η ροπή είναι μια συνθήκη με τύπο (DESIGN A,B), στο δεύτερο τύπο (DESIGN C) έχουμε κανονική ένταση. Στο τρίτο (DESIGN D) δημιουργούμε υψηλή τάση παρόλα αυτά έχουμε χαμηλό ρεύμα εκκίνησης. Τέλος στο τέταρτο (DESIGN F) δημιουργήσαμε ρεύμα και ροπή αλλά μερική ολίσθησή.

Έχουμε δύο διαφορετικά ήδη κινητήρων στην παραγωγή μαγνητικού πεδίου, όπου έχει παραγωγή μόνιμου μαγνήτη και στην παραγωγή πεδίου από διεγέρσεως τυλίγμα. Η πρώτη τάξης, κινητήρες η ροή των μαγνητών παραμένει ίδια και χωρίζονται σε όλες τις ταχύτητες με καμπύλες ροπής-ταχύτητας και ροπής-έντασης. Έχουμε δύο κατηγορίες που χωρίζονται σε τρεις υποκατηγορίες. Στους κινητήρες όπου τα τυλίγματα συνδέονται παράλληλα. Στους κινητήρες που έχει πόλους σε δύο τυλίγματα.

Κινητήρες DC

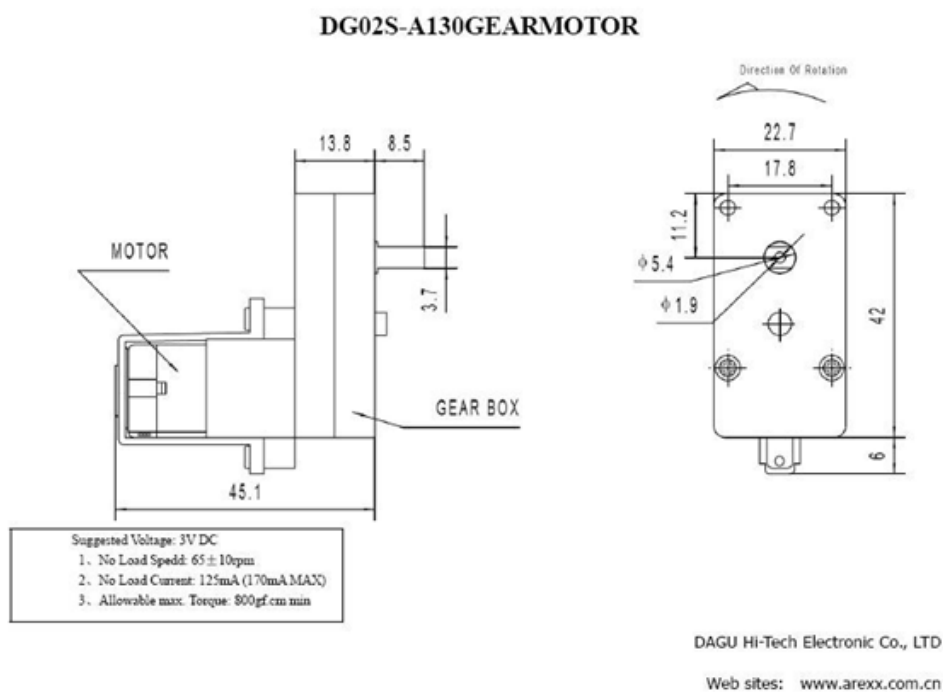
Η τάση των κινητήρων είναι μέχρι 6V. Έχουν ονομαστική τάση 3-9V. Λειτουργούν σε χαμηλότερες τάσεις που θα μπορούσαν να βοηθήσουν αρνητικά την διάρκεια της ζωής του κινητήρα.

Με βάση την παρούσα πτυχιακή, οι τροχοί είναι αυτοί που βοηθούν την κίνηση του εκκρεμούς. Οπότε χρειαζόμαστε κινητήρες συνεχούς ρεύματος και τάση λειτουργίας 6V και ροπή ώστε το βάρος του εκκρεμούς χωρίς να υπάρξει επιβάρυνση των power banks. Οι κινητήρες που χρησιμοποιήσαμε είναι με ταχύτητες για να βοηθάει το εκκρεμές στην ισορροπία.



Σχήμα 4.4 – Κινητήρας

Κάθε ένας από τους τέσσερεις κινητήρες που χρησιμοποιήσαμε απαιτεί τάση τροφοδοσίας από 3 έως 6 Volts και ρεύμα 200mA απουσία φορτίου, με μέγιστη ταχύτητα 65rpm.



Εικόνα 4.2 – Σχηματικό κινητήρα

2.4 Μπαταρίες

Εισαγωγή

Η μπαταρία είναι φτιαγμένη από χημικά όπου μέσα σε αυτήν διοχετεύετε ηλεκτρική ενέργεια. Είναι ένα δοχείο κατασκευασμένο από μονωτικό υλικό. Με την ένωση των καλωδίων με το κύκλωμα μειώνετε η εκφόρτωση του συσσωρευτή . Με αποτέλεσμα να έχουμε χημική μετατροπή ενέργειας σε ηλεκτρική. Η φόρτιση γίνεται με την τροφοδοσία ενός τροφοδοτικού ή πηγής όπου διαρρέετε από ρεύμα από την άλλη πηγή. Η χωρητικότητα δηλαδή η ποσότητα του ηλεκτρισμού σε αμπερώρια , από την τάση σε Volt. Έχουμε δύο είδη μπαταριών, τις μπαταρίες ως πρωτογενείς και ως δευτερογενείς. Αναλυτικά θα δούμε παρακάτω.

Οι πρωτογενείς χρησιμοποιούνται στους υπολογιστές ,σε φωτοτυπικά, σε ηλεκτρικές συσκευές και άλλες χρήσεις. Η μορφή τους είναι κυλινδρική με διαθέσιμο χώρο κάτω από 20mA Όπου τις χρησιμοποιούμε στο σπίτι περισσότερο σε αντίθεση με τις δευτερογενείς που είναι για την βιομηχανία. Οι βασικοί τύποι πρωτογενών είναι οι παρακάτω:

Ψευδαργύρου: χρησιμοποιούνται όπου έχουμε χρήση ενέργειας.

Αργύρου: χρησιμοποιούνται στα ρολόγια χειρός.

Λιθίου: χρησιμοποιούνται σε φωτοτυπικά μηχανήματα και στα κινητά.

Υδραργύρου: χρησιμοποιούνται σε ακουστικά βαρηκοΐας και είναι επικίνδυνο για το περιβάλλον.

Αλκαλικές: έχουν διάρκεια λειτουργίας και δεν μολύνουν το περιβάλλον.

Οι δευτερογενείς μπαταρίες είναι επαναφορτιζόμενες και έχουν τρία βασικά συστήματα :

Είναι κατασκευασμένες από νικέλιο-καδμίου: είναι επίσης επαναφορτιζόμενες ,ακόμα είναι οι πρώτες μπαταρίες που φτιαχτήκαν ποτέ. Χρησιμοποιούνται στα κινητά τηλέφωνα και τους υπολογιστές. Αυτό το είδος μπαταρίας δεν προτιμάτε γιατί είναι βλαβερό για το περιβάλλον και θέλουν να τις αντικαταστήσουν .

Οι μόλυβδου (Pb) : Ανακαλύφθηκε και έκανε επανάσταση στην αυτοκινητοβιομηχανία. Όμως ο μόλυβδος είναι βλαβερός για το περιβάλλον, γι' αυτό συλλέγονται άδειες από το συνεργείο των αυτοκινήτων.

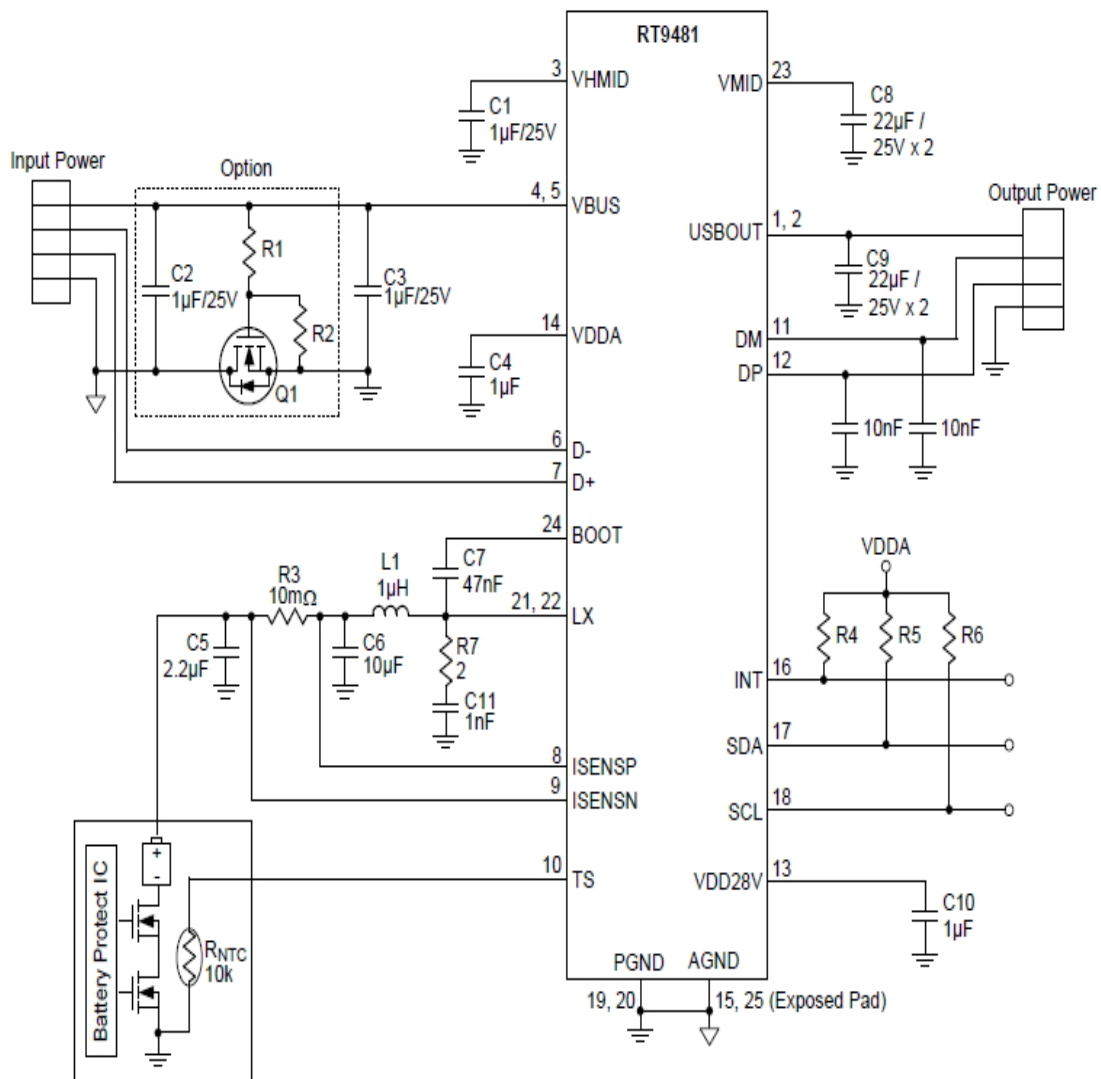
Το σύστημα νικελίου-μετάλλου υδριδίου (NiMH): Είναι πιο φιλικές προς το περιβάλλον από τις άλλες μπαταρίες και έχουν μεγάλη διάρκεια ζωής.

Power Bank

Το Power Bank είναι μια φορητή συσκευή που όταν επιθυμούμε να φορτίσουμε μια συσκευή την χρησιμοποιούμε. Επίσης σε σημεία που δεν υπάρχει άλλη πηγή φόρτισης χρησιμοποιούμε την συσκευή. Τα Power Bank έχουν μπαταρίες Li-Lon όπου είναι ο ποιο γνωστός τύπος τέτοιων πηγών ενέργειας. Όπως είναι το αντίστοιχο αυτό με μπαταρίες όλων των γκάτζετ. Η εκφόρτιση είναι ελάχιστη ,το βάρος είναι ελαφρύ και ικανοποιητικό στο να παρέχει το κατάλληλο ρεύμα. Το Power Bank έχει μεγάλο μειονέκτημα το οποίο είναι η ευκολία καταστροφής του όταν υπερθερμανθεί . Παρόλα αυτά το πρόβλημα αυτό δεν θα πρέπει να απασχολεί κάποιον ,αν η κατασκευάστρια εταιρία έχει κάνει πρόβλεψη για την αυτόματη λειτουργία απενεργοποίησης του ορίου επιτρεπόμενης θερμοκρασίας. Σε όλες τις περιπτώσεις αναφέρονται στις συσκευές ,η τάση εξόδου 5V όπου αυτή είναι η επιτρεπόμενη τάση που χρησιμοποιείτε σήμερα ώστε να φορτιστεί μια μικρή συσκευή. Η χωρητικότητα ενός Power Bank είναι περίπου 2000mAh έως 30000mAh .Η μπαταρία μπορεί να επαναφορτιστεί 500-1000 φορές. Το βάρος του είναι από 100gr ανα 5000mAh.



To power bank



Κύκλωμα power bank

2.4 Adafruit LSM9DS Αξελερόμετρο-γυροσκόπιο

Γενικά

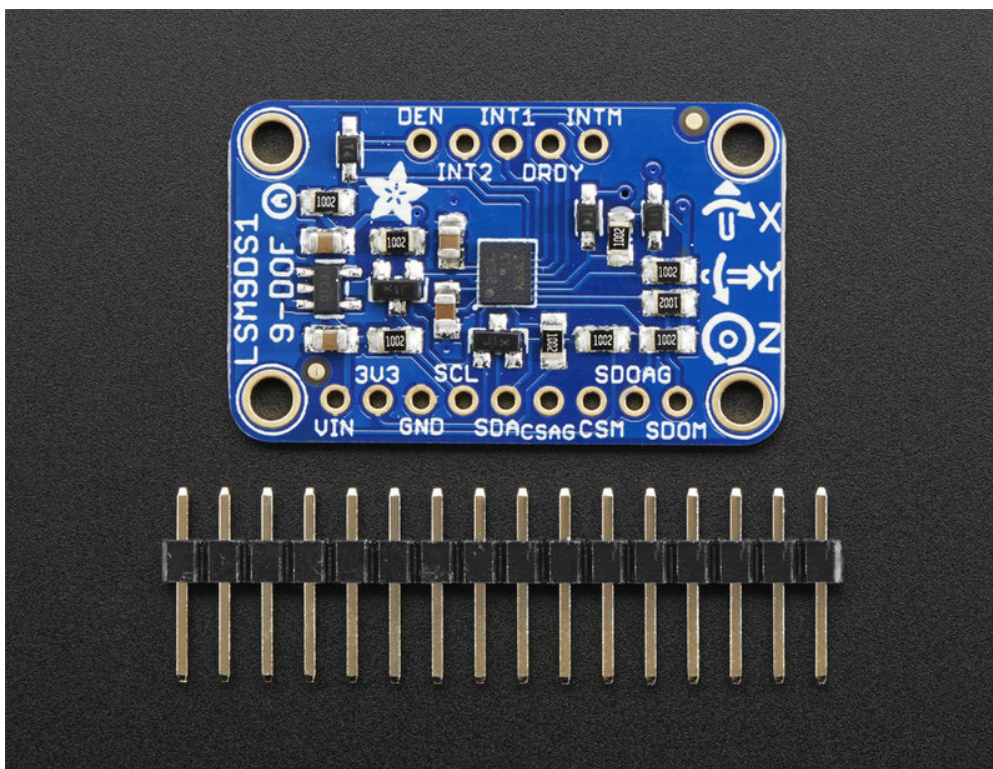
Στην πτυχιακή αυτή έχουμε χρησιμοποιήσει ένα τσίπ (LSM9DS1). Όπου το βάλουμε για να μπορέσουμε να βαλούμε σε ισοροπία το εκκρεμές. Το Όποιο τσίπ έχει τρία αισθητήρια το ένα είναι το επιταχυσσιόμετρο , τα άλλα είναι ένα μαγνητόμετρο και το τρίτο ένα γυροσκόπιο 3 αξόνων. Έχει διάφορα pins επάνω στο αισθητήριο. Το ξεπλοκάρισμα του με I2C και SPI , η τοποθέτηση του με δύναμη Vin, GND και SCL και SDA. Τέσσερα pins για να δημιουργήσουν μια σύνδεση διακοπής στην μια πλευρά του pin και από την άλλη πλευρά ένα συμπαγές ξεμπλοκάρισμα.

Μια αδρανειακή μονάδα μέτρησης, κοινώς γνωστή ως IMU, είναι μια ηλεκτρονική συσκευή που μετράει και αναφέρει τον προσανατολισμό, την ταχύτητα και τις βαρυτικές δυνάμεις μέσω της χρήσης επιταχυνσιόμετρων και γυροσκοπίων και συχνά μαγνητόμετρων. Τα IMU αποτελούν βασικό συστατικό των συστημάτων αδρανειακής πλοήγησης που χρησιμοποιούνται στα αεροσκάφη, τα μη επανδρωμένα εναέρια οχήματα (UAV) και άλλα μη επανδρωμένα συστήματα, καθώς και στους πυραύλους και ακόμη και στους δορυφόρους. Τα δεδομένα που συλλέγονται από μια IMU υποβάλλονται σε επεξεργασία από τους υπολογιστές για να παρακολουθήσουν τη θέση τους μέσω της νεκράς αναμέτρησης. Οι κοινές εφαρμογές για IMU περιλαμβάνουν έλεγχο και σταθεροποίηση, πλοήγηση και διόρθωση, μέτρηση και δοκιμή, έλεγχο μη επανδρωμένων συστημάτων και κινητή χαρτογράφηση.

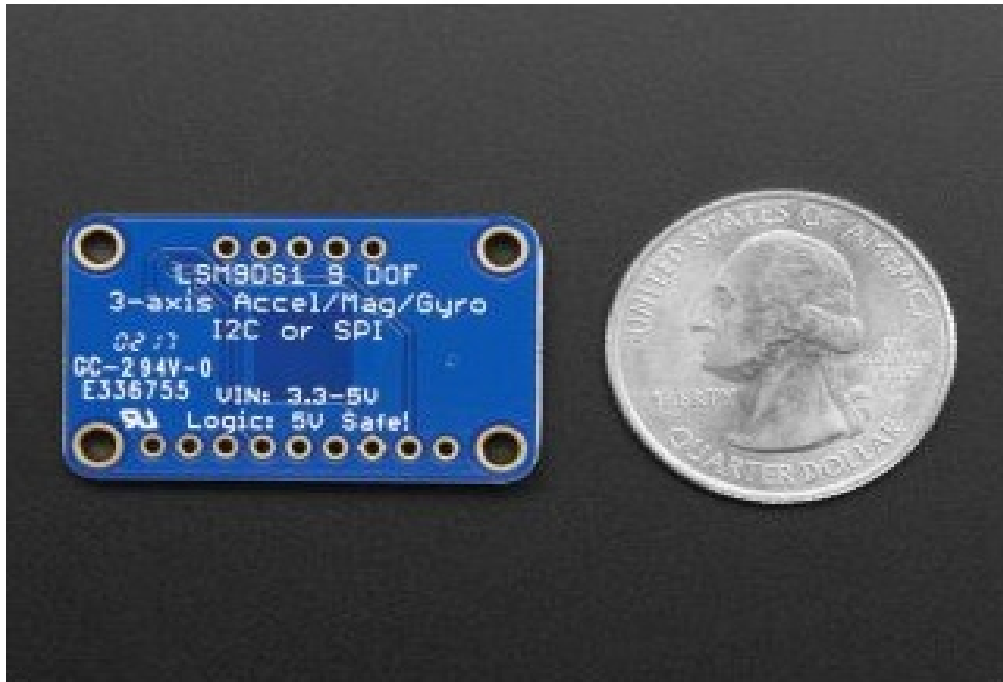
Σε ένα σύστημα αδρανειακής πλοήγησης, τα δεδομένα που συλλέγονται και αναγγέλλονται από ένα IMU υποβάλλονται σε επεξεργασία μέσω υπολογιστή, προκειμένου να υπολογιστεί η τρέχουσα θέση με βάση την ταχύτητα και το χρόνο. Οι υπολογισμοί ταχύτητας και χρόνου μπορούν να χρησιμοποιηθούν για την παροχή νεκρού αναλογισμού. Για παράδειγμα, εάν ένα σύστημα (ένα αεροσκάφος) ταξιδεύει βόρεια με ταχύτητα 100 μίλια ανά ώρα, μπορεί να συναχθεί ότι το σύστημα είναι τώρα 100 μίλια βόρεια από την αρχική του θέση. Ο συνδυασμός αυτής της "απλής" λογικής σε ένα σύστημα χαρτών μπορεί να δείξει πού ένα σύστημα είναι στο χάρτη παρόμοιο με το GPS, αλλά χωρίς να χρειάζεται να συνδεθεί με ή να επικοινωνήσει με εξωτερικά συστήματα. Αυτή είναι η διαδικασία που ονομάζεται νεκρός υπολογισμός.

Τα συστήματα IMU, στην πραγματικότητα όλα τα συστήματα πλοήγησης, απαιτούν πολύ ακριβή γυροσκόπια, καθώς η ποιότητα αυτών των συσκευών επηρεάζει σημαντικά τη συνολική απόδοση του συστήματος αισθητήρων αδρανείας. Ένα παράδειγμα γυροσκοπίου πολύ υψηλού επιπέδου είναι το γυροσκόπιο λέιζερ δακτυλίου (RLG). Οι RLG θεωρούνται γενικά η πιο ακριβής επιλογή, αλλά είναι επίσης πολύ ακριβές. Μια εναλλακτική λύση χαμηλότερου κόστους για το ακριβότερο RLG είναι το γυροσκόπιο οπτικών ινών (FOG). Τα FOG παρέχουν εξαιρετικά ακριβείς πληροφορίες περιστροφικού ρυθμού εξαιτίας της έλλειψης κινούμενων μερών. Ωστόσο, τα FOGs έχουν ένα μεγάλο μέρος των εγγενών εξελίξεων ανάπτυξης και κατασκευής, καθώς και ένα μεγαλύτερο μορφοποιητικό παράγοντα και υψηλότερες απαιτήσεις ισχύος.

Καθώς βελτιώνεται η τεχνολογία, τα FOGs έχουν ακόμα χαμηλότερο ανταγωνισμό κόστους. Τα γυροσκόπια βασισμένα σε μικροηλεκτρομηχανικά συστήματα (MEMS) έχουν κλείσει το χάσμα επιδόσεων σε μερικά FOGs και όταν παραγοντοποιούν τις χαμηλότερες απαιτήσεις κόστους και ισχύος τους, οι συσκευές MEMS παρέχουν μια εξαιρετική απάντηση για την ανάγκη ακρίβειας σε γυροσκόπια και συστήματα πλοήγησης, όπως IMUs. Τα συστήματα που βασίζονται σε MEMS δεν είναι μόνο λιγότερο δαπανηρά, είναι γενικά μικρά, ανθεκτικά και καταναλώνουν λιγότερη ενέργεια - παρέχοντας μεγάλες δυνατότητες για μικροσκοπικές λύσεις.



Η πρόσοψη του Adafruit LSM9DS1 αξελερόμετρου- γυροσκόπιου



Η κάτωψη του Adafruit LSM9DS1 αξελερόμετρου- γυροσκόπιου

2.5 Drivers

Περίληψη

Η κίνηση των κινητήρων γίνεται με την οδήγηση ενός ολοκληρωμένου L293D H-Bridge Motor. Είναι ένας τετραπλός οδηγός τάσης και με ρεύμα 1A και τάσεις 4.5V ως 36V.

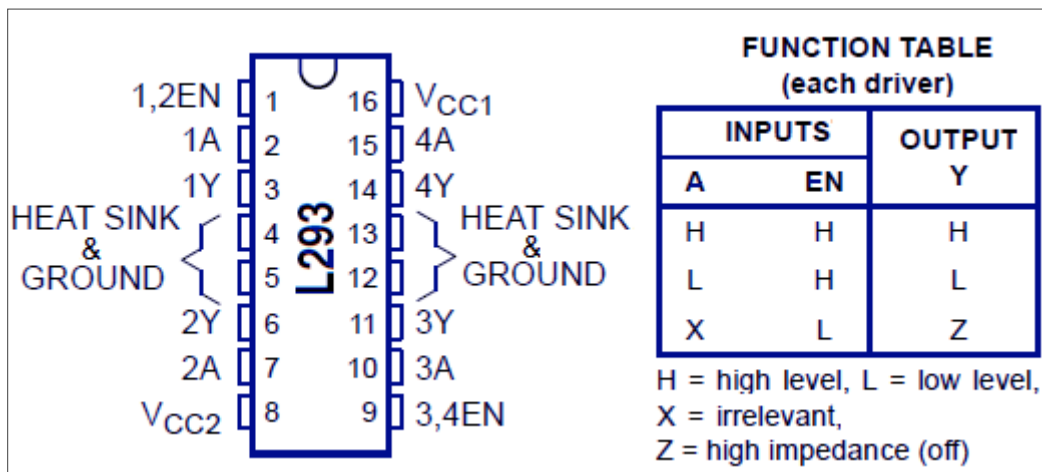
Έχουμε τις εξόδους-εισόδους για κάθε ζευγάρι ,η γέφυρά η οποία είναι για την εγκατάσταση κινητήρων είναι κατάλληλη. Η παροχή για το κύκλωμα εξόδου χρησιμοποιείται (VCC2).



Σχήμα 4.2 - Το ολοκληρωμένο κύκλωμα L293D που χρησιμοποιήσαμε στην κατασκευή

Ένας ελεγκτής ηλεκτροκινητήρα είναι μια συσκευή, η οποία καθορίζει την απόδοση ενός ηλεκτροκινητήρα. Ο ελεγκτής περιλαμβάνει τη δυνατότητα χειροκίνητης ή αυτόματης εκκίνησης και τερματισμού του κινητήρα, την δυνατότητα περιστροφής του προς τα εμπρός ή προς τα πίσω, την δυνατότητα ρύθμισης της ταχύτητας του, τη δυνατότητα ρύθμισης ή και περιορισμού της ροπής, και τέλος την προστασία του ηλεκτροκινητήρα από υπερφόρτωση. Κάθε ηλεκτροκινητήρας θα πρέπει να έχει κάποιο είδος ελεγκτή. Ο ελεγκτής του κινητήρα έχει διαφορετικά χαρακτηριστικά και πολυπλοκότητα, ανάλογα με την εργασία που καλείτε να εκτελέσει ο κινητήρας, όμως το βασικό υποσύστημα του είναι μία Η γέφυρα.

Η απλούστερη περίπτωση ενός ελεγκτή είναι ένας διακόπτης για τη σύνδεση του κινητήρα σε μια πηγή ενέργειας, όπως για παράδειγμα σε μικρές συσκευές, ή και σε ηλεκτρικά εργαλεία. Ο διακόπτης μπορεί να λειτουργεί χειροκίνητα, ή μπορεί να είναι ένας ηλεκτρονόμος ή μια επαφή, που συνδέεται σε κάποιον αισθητήρα, για να ξεκινήσει και να σταματήσει αυτόματα τον κινητήρα. Οι πιο πολύπλοκοι ελεγκτές ηλεκτροκινητήρων παρέχουν τη δυνατότητα του ακριβή ελέγχου της ταχύτητας και της ροπής του συνδεδεμένου κινητήρα ή των κινητήρων, και ενδέχεται να αποτελούν μέρος των συστημάτων ελέγχου κλειστού βρόχου, για την ακριβή τοποθέτηση της θέσης του ηλεκτροκινητήρα. Για παράδειγμα, ένας CNC τόρνος θα τοποθετήσει με ακρίβεια το εργαλείο κοπής για να διατηρηθεί η επιθυμητή θέση του εργαλείου



Σχήμα 4.3 - Πίνακας λειτουργίας για κάθε οδηγό του H-Bridge Motor Driver

2.6 Ρόδες

Εισαγωγή

Ο τροχός είναι το βασικό εξάρτημα της πτυχιακή. Ο τροχός ξέρουμε έχει μεγάλη ποικιλία εφαρμογών. Η θεωρία του τροχού είναι η εξής: ο νοητός άξονας περιστροφής θεωρείτε ακίνητος ,περνά από το κέντρο του και είναι κάθετος στο επίπεδο του τροχού. Ο πραγματικός άξονας θα έχει μία από τις ακόλουθες δύο διαρρυθμίσεις 1) ο τροχός θα είναι ελεύθερος να περιστραφεί γύρο από αυτόν ,2) ο τροχός θα είναι στέρα συνδεδεμένος με αυτόν.[4]



Λειτουργία και χρήσεις

Ο τροχός σε εφαρμογές μεταφοράς του δίνει την δυνατότητα να εκμηδενίσει τις τριβές, έτσι διευκολύνει την κίνηση μεγάλων αντικειμένων .Επιπρόσθετα η μετατροπή επιτρέπει η ολίσθηση των τριβών σε τριβές κύλισης οι οποίες είναι αρκετά μικρότερες

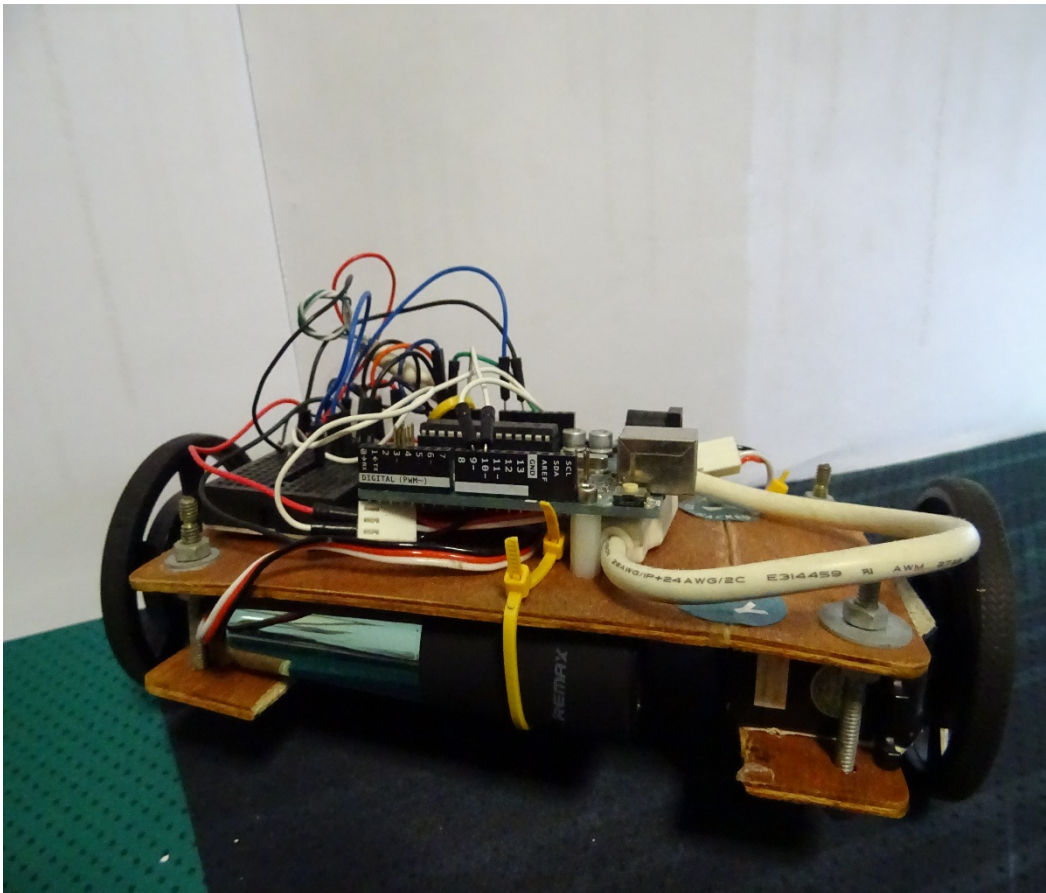
Για παράδειγμα ο περιστρεφόμενος τροχός του αγγειοπλάστη γύρο από την γραμμή του χρησιμοποιήθηκε για να μπορεί να επεξεργαστεί γύρο από το πήλινο αντικείμενο

Σε διάφορες εφαρμογές ο τροχός μπορεί να χρησιμοποιηθεί για την περιστροφική κίνηση των ή αντίθετα. Ακόμα ,για παράδειγμα η παλινδρομική κίνηση των εμβόλων ενός κινητήρα του αυτοκινήτου μετατρέπεται σε περιστροφική κίνηση του στρόφαλου ,μέσω της μπιέλας η οποία εν τέλη μετατρέπεται σε γραμμική κίνηση στους τροχούς .Επίσης ένα άλλο παράδειγμα είναι το βαρούλκο ,όπου γύρο από το τύμπανο τυλίγεται το συρματόσχοινο ,όπου είναι ένας απλός τροχός , η κίνηση του τύμπανου γίνεται γραμμική κίνηση του φόρτου που υπάρχει στην άκρη του.[5]

Δύο τροχοί διαμέτρου διαφορετικές σε μηχανικές εφαρμογές και μηχανήματα γίνεται να χρησιμοποιηθούν ως μειωτήρας .Γνωστό παράδειγμα είναι ο οδοντωτός (γρανάζι)

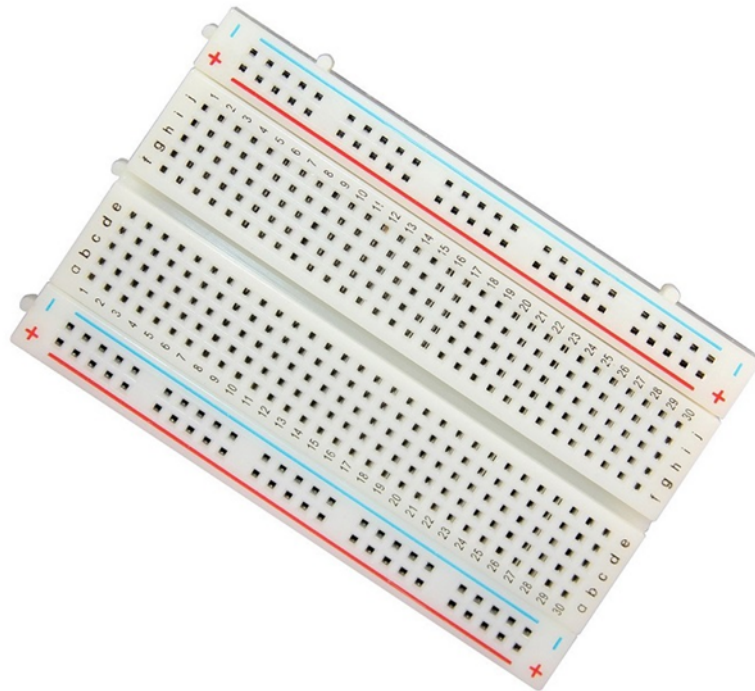
2.7 Η ξύλινη κατασκευή

Η ξύλινη κατασκευή συμπεριλαμβάνει διάφορα εξαρτήματα όπου έχει αρκετή αντοχή . Επίσης είναι τα εξαρτήματα έτσι διαμορφωμένα έτσι ώστε να μπορεί να ισορροπεί. Επιπρόσθετα η κατασκευή περιέχει το Arduino ,Power Bank, dc motor και καλώδιο σύνδεσης με το Laptop,για να κάνουμε διάφορες ρυθμίσεις.



Breadboard

Για τη συνδεσμολογία όλων των παραπάνω υλικών που απαρτίζουν το κατασκευαστικό κομμάτι της παρούσας εργασίας, χρησιμοποιήθηκε ένα breadboard το οποίο είναι χρησιμεύει για την τροφοδοσία, και όλες τις υπόλοιπες απαραίτητες καλωδιώσεις των επιμέρους υποσυστημάτων.



Κεφαλαίο3:

Σχεδίαση του ελεγκτή

3.1 PID Ελεγκτής

Ο έλεγχος που υπάρχει στο εκκρεμές είναι ο PID ένας τριπλός ελεγκτής όπου χρησιμοποιείτε στην βιομηχανία και σε άλλες εφαρμογές. Ένας PID ελεγκτής κάνει υπολογισμούς την τιμή του σφάλματος $e(t)$ ως διαφορά της επιθυμητής τιμής ρύθμισης. Στην ουσία κάνει διόρθωση βασισμένη σε αναλογικές. Όμως πρακτικά δίνει διορθωμένη και ακριβή διόρθωση σε μια άλλη λειτουργία. Για παράδειγμα ο έλεγχος της ταχύτητας σε οδικό όχημα. Ανάλογα με την κλίση του οχήματος και το πόσο πατάει το γκάζι, έχουμε ως έξοδο την ταχύτητα του οχήματος. Η πρώτη πρακτική εφαρμογή υπήρξε στα πλοία, όπου εφαρμόστηκε το 1920 και μετά. Τα επόμενα χρόνια χρησιμοποιήθηκε για τον αυτοματοποιημένο έλεγχο και στην συνέχεια στους ηλεκτρονικούς ελεγκτές.

Έννοια της ανάδρασης

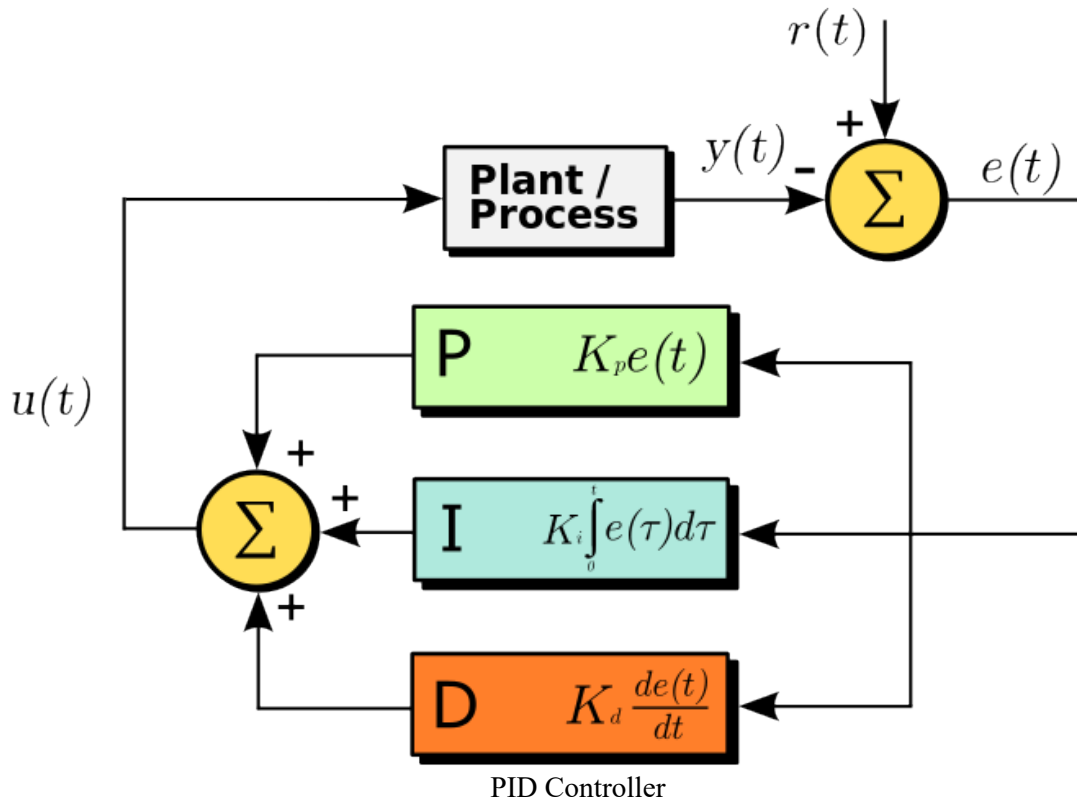
Όπως είδη αναφέραμε το ανάστροφο εκκρεμές είναι ένα σύστημα ασταθές έτσι λοιπόν η ανατροφοδότηση (feedback) είναι αναγκαία προκειμένου να δημιουργήσουμε ένα χρήσιμο ευσταθές σύστημα.[6]

Τι είναι η ανατροφοδότηση;

Όλα τα συστήματα αυτομάτου ελέγχου μπορούν να διακριθούν σε δύο κατηγορίες στα ανοικτά και τα κλειστά συστήματα. Η ανατροφοδότηση αναφέρεται στα κλειστά συστήματα. Κλειστό σύστημα είναι ένα σύστημα όπου η είσοδος του $u(t)$ είναι συνάρτηση της εξόδου του $y(t)$. Σε ένα σύστημα αυτομάτου ελέγχου το σήμα εισόδου $u(t)$ δεν παράγεται απ' ευθείας από μία γεννήτρια, αλλά είναι έξοδος ενός πρόσθετου συστήματος που ονομάζουμε ελεγκτή. Ας δούμε ένα καθημερινό παράδειγμα του ανθρώπου που οδηγεί, το αυτοκίνητο είναι το σύστημα και η έξοδος είναι η ταχύτητα. Ο οδηγός παρακολουθεί τη συμπεριφορά του αυτοκινήτου και ρυθμίζει τη διέγερση του

συστήματος, π.χ. πατάει γκάζι, στρίβει το τιμόνι πατάει φρένο. Η οδήγηση δηλαδή έχει τη δομή ενός κλειστού συστήματος όπου ο οδηγός είναι ο ελεγκτής. Από το παραπάνω παράδειγμα είναι φανερό ότι τα κλειστά συστήματα διαφέρουν στην αρχή λειτουργίας τους από τα ανοικτά συστήματα. Η διαφορά αυτή οφείλεται στην τροφοδοσία πληροφοριών από την έξοδο στην είσοδο του συστήματος. Η διαδικασία αυτή ονομάζεται ανατροφοδότηση και παίζει βασικότατο ρόλο στον αυτοματισμό. Εδώ θα πρέπει να αναφέρουμε και μία βασική διαφορά του ανοικτού συστήματος και του κλειστού συστήματος όσον αφορά την είσοδο, σαν είσοδο στο ανοικτό σύστημα είχαμε απλώς το (voltage) του motor ενώ στο κλειστό σύστημα σαν είσοδο έχουμε την επιθυμητή θέση η οποία αφαιρείται από την πραγματική θέση που βρίσκεται το βαγόνι. Γενικά τα ο σχεδιασμός ελέγχου με ανατροφοδότηση χρησιμοποιείται εκεί που οι απαιτήσεις αυτοματισμού είναι μεγάλες ενώ πολύ σπάνια βρίσκουμε ελεγκτές χωρίς ανατροφοδότηση.[7]

Έλεγχος - PID ελεγκτής



Ο έλεγχος της θέσης κάθε άρθρωσης του Delta Robot, αποτελεί το κυριότερο ζήτημα ελέγχου. Έτσι είναι αναγκαία η χρήση ενός συστήματος ελέγχου, ούτως ώστε να δέχεται την απαραίτητη πληροφορία από τα εκάστοτε αισθητήρια, και με βάση την μέτρηση που θα λαμβάνει, αλλά και με τη χρήση μαθηματικών εξισώσεων να οδηγεί τους ηλεκτρικούς κινητήρες στην επιθυμητή για κάθε κινητήρα ξεχωριστά θέση αναφοράς με τη βέλτιστη δυνατή ταχύτητα εκτέλεσης, και με το μικρότερο δυνατό σφάλμα. Το πιο κοινό, αλλά και πιο γνωστό σύστημα ελέγχου που χρησιμοποιείται κατά κόρον σε τέτοιου είδους εφαρμογές, είναι ο PID ελεγκτής.

Ο PID ελεγκτής ο οποίος είναι τα αρχικά των λέξεων proportional, integral και derivative, που μεταφράζεται ως αναλογικός-ολοκληρωτικός-διαφορικός ελεγκτής, είναι ένας σύνδεσμος βρόγχων ελέγχου ανατροφοδότησης που η κύρια χρήση του επικεντρώνεται σε βιομηχανικές εφαρμογές αυτομάτου ελέγχου.

Οι PID ελεγκτές χρησιμοποιούνται ούτως ώστε να μηδενίσουν το σφάλμα μεταξύ μιας επιθυμητής και μιας μετρούμενης μεταβλητής, κάνοντας τους κατάλληλους υπολογισμούς και ενεργώντας ανάλογα έτσι ώστε να επιτευχθεί η σωστή ρύθμιση της εκτελούμενης διεργασίας.

Για τη σωστή ρύθμιση και τον υπολογισμό ενός PID ελεγκτή, χρειάζεται να ορίσουμε τρεις διαφορετικούς παραμέτρους, τον αναλογικό τον ολοκληρωτικό και τον διαφορικό όρο. Ο πρώτος όρος, ο αναλογικός, διαμορφώνει το πως θα αντιδράσει το σύστημα στο σφάλμα της παρούσας μέτρησης, ο δεύτερος όρος ο ολοκληρωτικός, διαμορφώνει το πως θα αντιδράσει το σύστημα σε σχέση με το σύνολο των

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

πρόσφατων σφαλμάτων, και τέλος ο τρίτος όρος, ο διαφορικός, διαμορφώνει το πως θα αντιδράσει το σύστημα σε σχέση με το πόσο μεταβλήθηκε η τιμή του σφάλματος.

Η συνολική τιμή και των τριών παραπάνω όρων ρυθμίζει μέσω ενός στοιχείου ελέγχου τη διεργασία, όπως για παράδειγμα η ταχύτητα ενός ηλεκτροκινητήρα ελέγχεται από τη τάση που εφαρμόζεται στον κινητήρα. Καθορίζοντας τα κατάλληλα κέρδη και για τους τρεις όρους του PID ελεγκτή, είναι εφικτό ο PID να αντιδρά με βάση τις σχεδιασμένες απαιτήσεις που έχουν οριστεί για κάθε διαφορετική διεργασία.

Το κατά πόσο ο ελεγκτής είναι αποδοτικός καθορίζεται από το πώς ανταποκρίνεται σε παρουσία σφάλματος στο σύστημα, με λίγα λόγια με την υπερύψωση που εισάγετε στο σύστημα. Χρησιμοποιώντας τον PID ελεγκτή δεν είναι σίγουρο ότι θα επιτευχθεί ο βέλτιστος έλεγχος του συστήματός μας, όμως ο αλγόριθμος του PID είναι αρκετός έτσι ώστε να επιτευχθεί η εκτέλεση της διεργασίας με αρκετά μεγάλη ακρίβεια.

Υπάρχουν εφαρμογές όπου δεν είναι αναγκαία η ρύθμιση και των τριών κερδών, καθώς για τον κατάλληλο έλεγχο του συστήματος επαρκούν μόνο ένα ή και δύο κέρδη, ενώ τα κέρδη που δεν χρησιμοποιούνται παίρνουν αυτομάτως τη τιμή μηδέν. Έτσι ο PID ελεγκτής ανάλογα με το ποιους όρους δεν χρησιμοποιεί ονομάζεται είτε P, I, PI, ή PD, δεδομένου των κερδών που χρησιμοποιούνται.

Η συνολική συνεισφορά της εξόδου του PID ορίζεται από το άθροισμα των τριών όρων διόρθωσης του σφάλματος, την οποία ονομάζουμε MV. Η μαθηματική διατύπωση της εξόδου του PID ορίζεται από τον παρακάτω τύπο, καθώς το P_{out}, το I_{out}, και το D_{out} είναι οι συνεισφορές εξόδου του καθενός όρου του PID ελεγκτή.

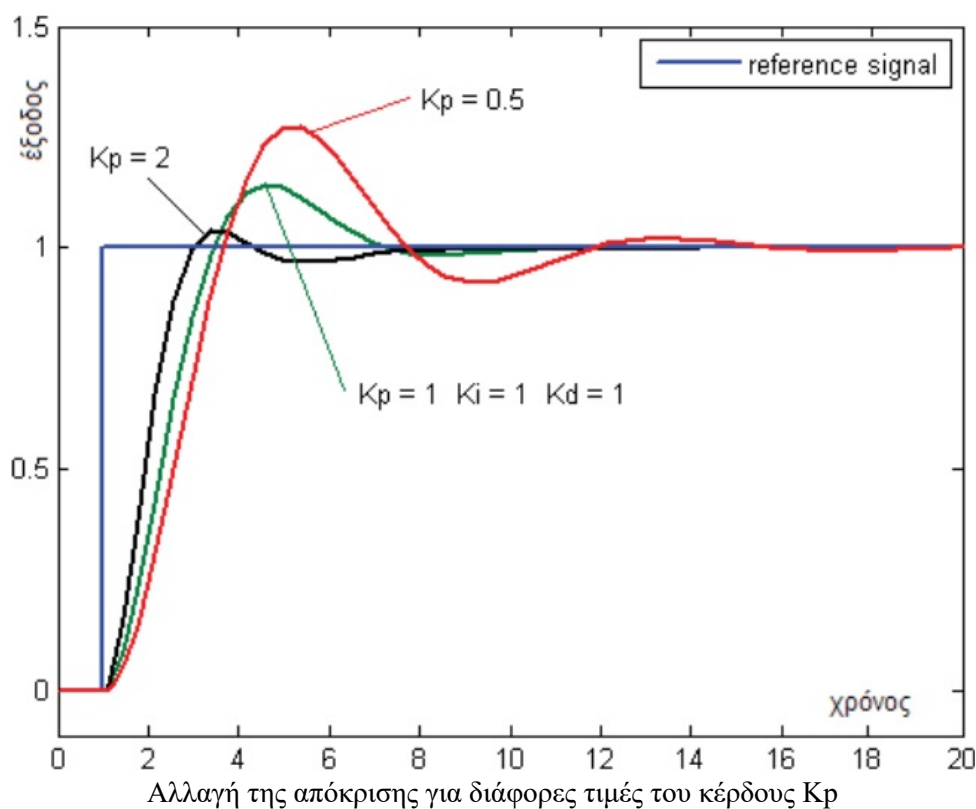
$$MV(t) = P_{out} + I_{out} + D_{out}$$

Η έξοδος του αναλογικού P όρου του PID είναι ανάλογη του σφάλματος, καθώς το πόσο επιδρά στο σύστημα ρυθμίζεται άμεσα από το αναλογικό κέρδος K_p , το οποίο πολλαπλασιάζεται με το σφάλμα. Η έξοδος του αναλογικού όρου δίνεται από τη παρακάτω σχέση, όπου P_{out} η έξοδος, K_p το αναλογικό κέρδος, και e το σφάλμα.

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

Η αρκετά μεγάλη συνεισφορά της αναλογικής εξόδου, συνεπάγεται σε μεγάλη και απότομη αλλαγή της εξόδου του συστήματος σε κάθε αλλαγή του σφάλματος, καθώς το σύστημα μπορεί να οδηγηθεί και σε αστάθεια.

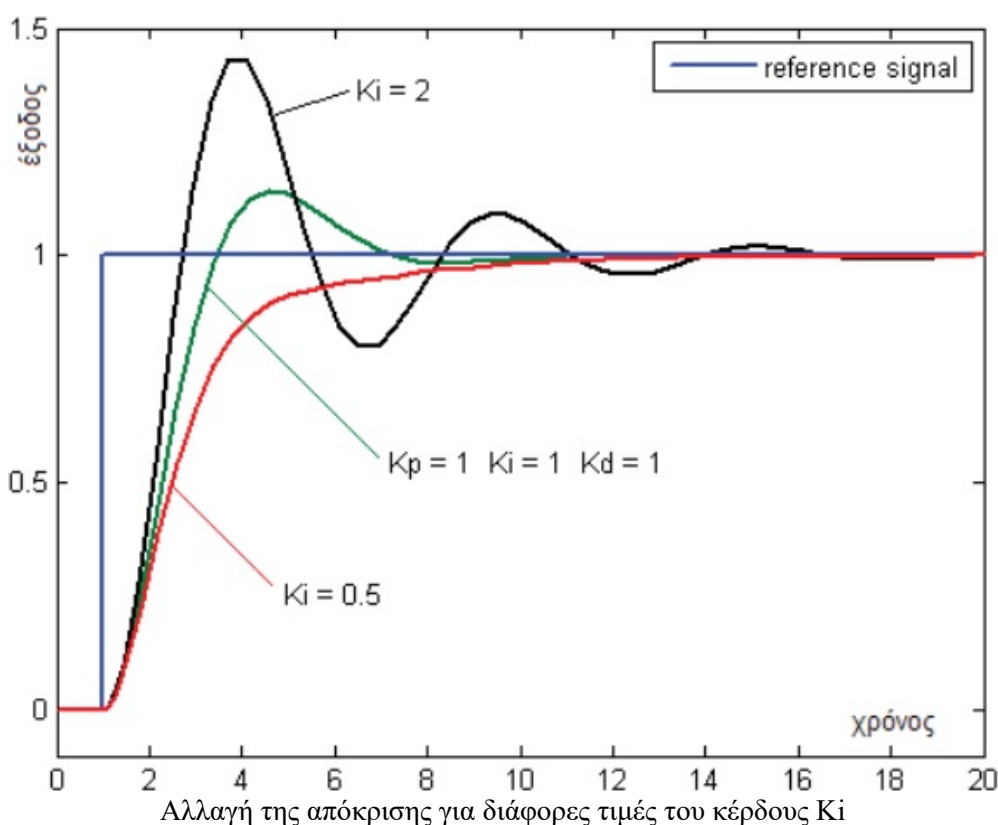
Όμως η μικρή συνεισφορά της αναλογικής εξόδου, συνεπάγεται σε μικρή αλλαγή της εξόδου του συστήματος σε κάθε αλλαγή του σφάλματος, καθώς επίσης η ευαισθησία του ελεγκτή μειώνεται.



Η έξοδος του ολοκληρωτικού I όρου του PID είναι ανάλογη του συσσωρευμένου σφάλματος, καθώς το πόσο επιδρά στο σύστημα ρυθμίζεται άμεσα από το ολοκληρωτικό κέρδος K_i , το οποίο πολλαπλασιάζεται με το συσσωρευμένο σφάλμα. Η έξοδος του αναλογικού όρου δίνεται από τη παρακάτω σχέση, όπου I_{out} η έξοδος, K_i το ολοκληρωτικό κέρδος, και e το σφάλμα.

$$I_{out} = K_i \int_0^t e(\tau) d\tau$$

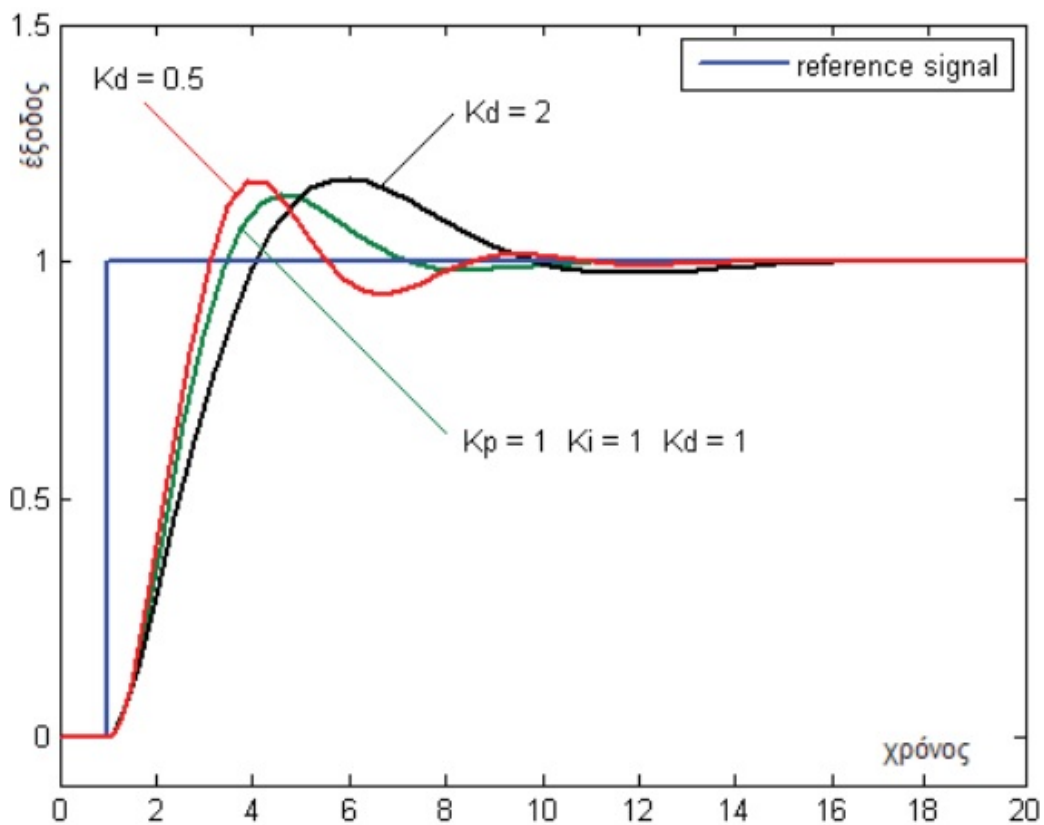
Ο I όρος οδηγεί αρκετά πιο γρήγορα την έξοδο του συστήματος στην επιθυμητή κατάσταση, μηδενίζοντας επίσης το σφάλμα κατάστασης. Όμως λόγω του συσσωρευμένου σφάλματος, μια μεγάλη τιμή του κέρδους μπορεί να οδηγήσει την έξοδο του συστήματος σε υπερύψωση. Αυτό σημαίνει ότι το σύστημα θα ξεπεράσει την επιθυμητή κατάσταση, δημιουργώντας έτσι μία απόκλιση του συστήματος.



Η έξοδος του διαφορικού D όρου του PID είναι ανάλογη της κλίσης του σφάλματος, καθώς το πόσο επιδρά στο σύστημα ρυθμίζεται άμεσα από το διαφορικό κέρδος K_d , το οποίο πολλαπλασιάζεται με τη κλίση του σφάλματος. Η έξοδος του διαφορικού όρου δίνεται από τη παρακάτω σχέση, όπου D_{out} η έξοδος, K_d το διαφορικό κέρδος, και e το σφάλμα.

$$D_{out} = K_d \frac{de}{dt}$$

Ο D όρος μειώνει την απότομη μεταβολή της εξόδου του συστήματος. Για το λόγω αυτό ο D όρος χρησιμοποιείται για να μειώσει τις τυχόν υπερυψώσεις του συστήματος, καθώς επίσης αναβαθμίζει το συνδυασμό της σταθερότητας ανάμεσα στον PID ελεγκτή και της συνολικής διαδικασίας του ελέγχου του συστήματος.



Αλλαγή της απόκρισης για διάφορες τιμές του κέρδους K_d

Το άθροισμα του αναλογικού, του ολοκληρωτικού και του διαφορικού όρου ορίζεται ως η έξοδος του PID ελεγκτή. Ορίζοντας ως MV την ολοκληρωμένη συνολική έξοδο του PID ελεγκτή, η εξίσωση που τον ορίζει είναι η εξής:

$$MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}$$

Τα κέρδη του PID ελεγκτή τα οποία ορίζονται και ως παράμετροι συντονισμού είναι τα εξής:

1) Το αναλογικό κέρδος K_p , ή και αναλογική ενίσχυση. Όσο μεγαλύτερη είναι η τιμή του κέρδους K_p , τόσο ταχύτερη απόκριση έχει το σύστημά μας, καθώς όσο αυξάνεται η τιμή του σφάλματος τόσο αυξάνεται και η αναλογική έξοδος του PID ελεγκτή. Επίσης μια αρκετά υψηλή τιμή του κέρδους K_p μπορεί να οδηγήσει σε αστάθεια του συστήματος.

2) Το ολοκληρωτικό κέρδος K_i , ή και ολοκληρωτική ενίσχυση. Όσο μεγαλύτερη είναι η τιμή του κέρδους K_i , τόσο πιο γρήγορα το σφάλμα της τελικής κατάστασης μηδενίζεται, καθώς ενδέχεται να υπάρξουν φαινόμενα υπερύψωσης στο σύστημα.

3) Το διαφορικό κέρδος K_d , ή και διαφορική ενίσχυση. Όσο μεγαλύτερη είναι η τιμή του κέρδους K_d τόσο μειώνεται και η υπερύψωση του συστήματος, καθώς όμως μπορεί να οδηγήσει το σύστημα σε αστάθεια

Εύρεση Κερδών του PID Ελεγκτή

Για το συντονισμό και την επιλογή των κερδών ενός PID ελεγκτή υπάρχουν αρκετές μέθοδοι. Αρχικά υπάρχουν οι χειρωνακτικοί μέθοδοι, οι οποίες βασίζονται καθαυτού στην εμπειρία του χειριστή της διεργασίας. Όμως τέτοιου είδους μέθοδοι καθίσταται ανεπαρκείς για τη βέλτιστη και σωστή λειτουργία του συστήματος.

Οι πιο αποτελεσματικές μέθοδοι που χρησιμοποιούνται για την επιλογή των κατάλληλων κερδών ενός PID ελεγκτή εμπεριέχουν κάποια συγκεκριμένη αλληλουχία πρότυπων διαδικασιών, με τις οποίες επιτυγχάνονται οι βέλτιστες επιλογές των τριών όρων του PID ελεγκτή.

Η πιο γνωστή μέθοδος, αλλά επίσης και η πιο διαδεδομένη, είναι η μέθοδος Ziegler-Nichols, η οποία εισάχθηκε στη θεωρία του αυτομάτου ελέγχου από τον John G. Ziegler, και τον Nathaniel B. Nichols.

Η αρχή λειτουργίας της συγκεκριμένης μεθόδου είναι ως εξής, αρχικά και τα τρία κέρδη του PID ελεγκτή P I και D, παίρνουν τη τιμή μηδέν. Στη συνέχεια το κέρδος P αυξάνεται σταδιακά μέχρι να αρχίσει το σύστημα να ταλαντώνεται. Τότε η τιμή του κέρδους P ορίζεται ως κρίσιμο κέρδος K_C . Τέλος η τιμή του κρίσιμου κέρδους K_C και η περίοδος της ταλάντωσης P_C , χρησιμοποιούνται για την εύρεση των κερδών με βάση τον τύπο του ελεγκτή που θα χρησιμοποιήσουμε με βάση τον παρακάτω πίνακα.

Μέθοδος Ziegler-Nichols			
Τύπος Ελέγχου	K_p	K_i	K_d
P	$0.5 * K_c$	-	-
PI	$0.45 * K_c$	$1.2K_p / P_c$	-
PID	$0.6 * K_c$	$2K_p / P_c$	$K_p P_c / 8$

Κέρδη συντονισμού PID Ziegler-Nichols

4.1 Κώδικας

Μπορούμε να γράψουμε κώδικα με την εγκατάσταση του Arduino IDE. Ξεκινώντας με δύο βασικές εντολές στο περιβάλλον Arduino IDE είναι η Setup και η loop οι οποίες έχουν ως εξής: setup() όπου ενεργοποιούν τις εντολές μία φορά. Ακόμα οι μεταβλητές μπαίνουν και οι χαρακτήρες οπωσδήποτε των εξόδων που το πρόγραμμα θα χρησιμοποιήσει. loop() το πρόγραμμα μας γράφουμε εδώ. Η loop() για να ξανακαλεστεί θα τρέξουν οι εντολές που υπάρχουν, συνεχίζοντας από την αρχή της. Το reset μηδενίζει τα πάντα, όσο ρεύμα έχει στο Arduino, υπάρχει. Το Arduino με το setup() που τρέχει το πρόγραμμα και με το loop() να τρέχει ξανά και ξανά και την reset που μηδενίζει.

```
#include <SPI.h>

#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <Adafruit_LSM9DS0.h>

#include <Servo.h> // Comes with Arduino IDE

Adafruit_LSM9DS0 lsm = Adafruit_LSM9DS0(1000); // Use I2C, ID #10

#define LSM9DS0_XM_CS 10

#define LSM9DS0_GYRO_CS 9

#define ServoPIN 9 // Can be changed 3,5,6,9,10,11 (NOW can be any pin
including A0..A5)

#define ServoMIN 180 // Don't go to end of servo travel

#define ServoMAX 180 // which may be 0 to 180.

#define ServoPIN2 10

#define LSM9DS0_SCLK 13

#define LSM9DS0_MISO 12

#define LSM9DS0_MOSI 11

Servo myservo;

Servo myservo2;

int pos=0;

double outMax = 5;
```

```

double outMin = -outMax;

float lastInput = 0;

double ITerm =0;

// PID constants

// You can change this values to adjust the control

double kp =0.8;    // Proportional value

double ki = 1.5;    // Integral value

double kd = 4.6;    // Derivative value

double Setpoint = 0; // Initial setpoint is 0

double comp;

int move1=0,move2=0,start=0;

int degree=2;

int Delay=15;

int lim=10;

double yd=7;

double yr,y0=0,iT,yr0;

float max_val=1.5,dT;

float MAX_VAL=2,error,PID;

sensors_event_t accel, mag, gyro, temp;

void displaySensorDetails(void)

{

sensor_t accel, mag, gyro, temp;

lsm.getSensor(&accel, &mag, &gyro, &temp);

Serial.println(F("-----"));

```

```

Serial.print (F("Sensor:  ")); Serial.println(accel.name);

Serial.print (F("Driver Ver:  ")); Serial.println(accel.version);

Serial.print (F("Unique ID:  ")); Serial.println(accel.sensor_id);

Serial.print (F("Max Value:  ")); Serial.print(accel.max_value);

Serial.println(F(" m/s^2"));

Serial.print (F("Min Value:  ")); Serial.print(accel.min_value);

Serial.println(F(" m/s^2"));

Serial.print (F("Resolution:  ")); Serial.print(accel.resolution);

Serial.println(F(" m/s^2"));

Serial.println(F("-----"));

Serial.println(F(""));

Serial.println(F("-----"));

Serial.print (F("Sensor:  ")); Serial.println(mag.name);

Serial.print (F("Driver Ver:  ")); Serial.println(mag.version);

Serial.print (F("Unique ID:  ")); Serial.println(mag.sensor_id);

Serial.print (F("Max Value:  ")); Serial.print(mag.max_value);
Serial.println(F(" uT"));

Serial.print (F("Min Value:  ")); Serial.print(mag.min_value);
Serial.println(F(" uT"));

Serial.print (F("Resolution:  ")); Serial.print(mag.resolution);

Serial.println(F(" uT"));

Serial.println(F("-----"));

Serial.println(F(""));

Serial.println(F("-----"));

Serial.print (F("Sensor:  ")); Serial.println(gyro.name);

Serial.print (F("Driver Ver:  ")); Serial.println(gyro.version);

```

```

Serial.print (F("Unique ID:  ")); Serial.println(gyro.sensor_id);

Serial.print (F("Max Value:  ")); Serial.print(gyro.max_value);

Serial.println(F(" rad/s"));

Serial.print (F("Min Value:  ")); Serial.print(gyro.min_value);

Serial.println(F(" rad/s"));

Serial.print (F("Resolution:  ")); Serial.print(gyro.resolution);

Serial.println(F(" rad/s"));

Serial.println(F("-----"));

Serial.println(F(""));

Serial.println(F("-----"));

Serial.print (F("Sensor:    ")); Serial.println(temp.name);

Serial.print (F("Driver Ver:  ")); Serial.println(temp.version);

Serial.print (F("Unique ID:  ")); Serial.println(temp.sensor_id);

Serial.print (F("Max Value:  ")); Serial.print(temp.max_value);

Serial.println(F(" C"));

Serial.print (F("Min Value:  ")); Serial.print(temp.min_value);

Serial.println(F(" C"));

Serial.print (F("Resolution:  ")); Serial.print(temp.resolution);

Serial.println(F(" C"));

Serial.println(F("-----"));

Serial.println(F(""));

delay(500);

}void configureSensor(void)
{

```



```

lsm.setupAccel(lsm.LSM9DS0_ACCEL_RANGE_16G);
lsm.setupMag(lsm.LSM9DS0_MAG_GAIN_2GAUSS);
lsm.setupGyro(lsm.LSM9DS0_GYRO_SCALE_245DPS);
}

void setup(void)
{
  #ifndef ESP8266
  while (!Serial); // will pause Zero, Leonardo, etc until serial console
  opens
  #endif

  Serial.begin(9600);

  Serial.println(F("LSM9DS0 9DOF Sensor Test")); Serial.println("");

  /* Initialise the sensor */

  if(!lsm.begin())
  {
    /*
    There was a problem detecting the LSM9DS0 ... check your connections
    */

    Serial.print(F("Oops, no LSM9DS0 detected ... Check your wiring or
    I2C ADDR!"));

    while(1);
  }

  Serial.println(F("Found LSM9DS0 9DOF"));

  displaySensorDetails();

```

```
configureSensor();

Serial.println("");

myservo.attach(ServoPIN);

myservo2.attach(ServoPIN2);

}

// Calculates the PID output
double Compute(double input)
{

    double error = Setpoint - input;

    //return error;

    ITerm+= (ki * error);

    if(ITerm > outMax) ITerm= outMax;

    else if(ITerm < outMin) ITerm= outMin;

    double dInput = (input - lastInput);

    // Compute PID Output

    double output = kp * error + ITerm + kd * dInput;

    if(output > outMax) output = outMax;

    else if(output < outMin) output = outMin;

    // Remember some variables for next time
```

```
lastInput = input;
```

```
return output;
```

```
}
```

```
int i=0;
```

```
void loop(void)
```

```
{
```

```
DelayOnce(); //delay first time
```

```
//ChangePos();
```

```
lsm.getEvent(&accel, &mag, &gyro, &temp);
```

```
yd=0.9;
```

```
yr=accel.acceleration.y-yd;
```

```
i++;
```

```
if(i>4)
```

```
{
```

```
Serial.print("\nY:");
```

```
Serial.print(yr);
```

```
i=0;
```

```
}
```

```
comp=Compute(yr); //PID
```

```
//Serial.print("PID output=");Serial.print(PID);Serial.print("\n");
```

```
error=yr-y0;
```

```
dT=yr-yr0;
```

```
iT+=error;
```

```
PID_fn();
```

```
double angle=0.4;
```

```
if(PID>angle) Forw();
```

```
else if(PID<-angle) Back();
```

```
else
```

```
Stop();
```

```
yr0=yr;
```

```
}
```

```
//----- new
```

```
void Forw()
```

```
{
```

```
myservo.write(ServoMAX-pos);          // tell servo to go to
position in variable 'pos'
```

```
myservo2.write(pos);                  // tell servo to go to position in
variable 'pos'
```

```
ChangePos();
```

```
delay(Delay); // waits 15ms for the servo to reach the position
```

```
}
```

```
//-----
```

```
void Back()
```

```
{
```

```
myservo2.write(ServoMAX-pos);          // tell servo to go to
```

position in variable 'pos'

```
myservo.write(pos); // tell servo to go to position in
```

```
variable 'pos'
```

```
ChangePos();
```

```
delay(Delay); // waits 15ms for the servo to reach the position
```

```
}
```

```
//-----
```

```
void Stop()
```

```
{
```

```
myservo2.write(95); // tell servo to go to position in variable
```

```
'pos'
```

```
myservo.write(90); // tell servo to go to position in variable
```

```
'pos'
```

```
delay(Delay); // waits 15ms for the servo to reach the position
```

```
}
```

```
//-----
```

```
void PID_fn()
```

```
{
```

```
PID=1.45*error+0.05*iT+0.1*dT; //PID CONTROL
```

```
if(PID>MAX_VAL) PID=MAX_VAL;
```

```
if(PID<-MAX_VAL) PID=-MAX_VAL;
```

```
}
```

```
void DelayOnce()
```

```
{  
if(start=0)  
{  
pos = ServoMAX;  
delay(1000);  
start=1;  
}  
}  
  
//-----  
  
void ChangePos()  
{  
pos-=degree;  
if(pos<ServoMIN) pos=ServoMAX;  
}  

```

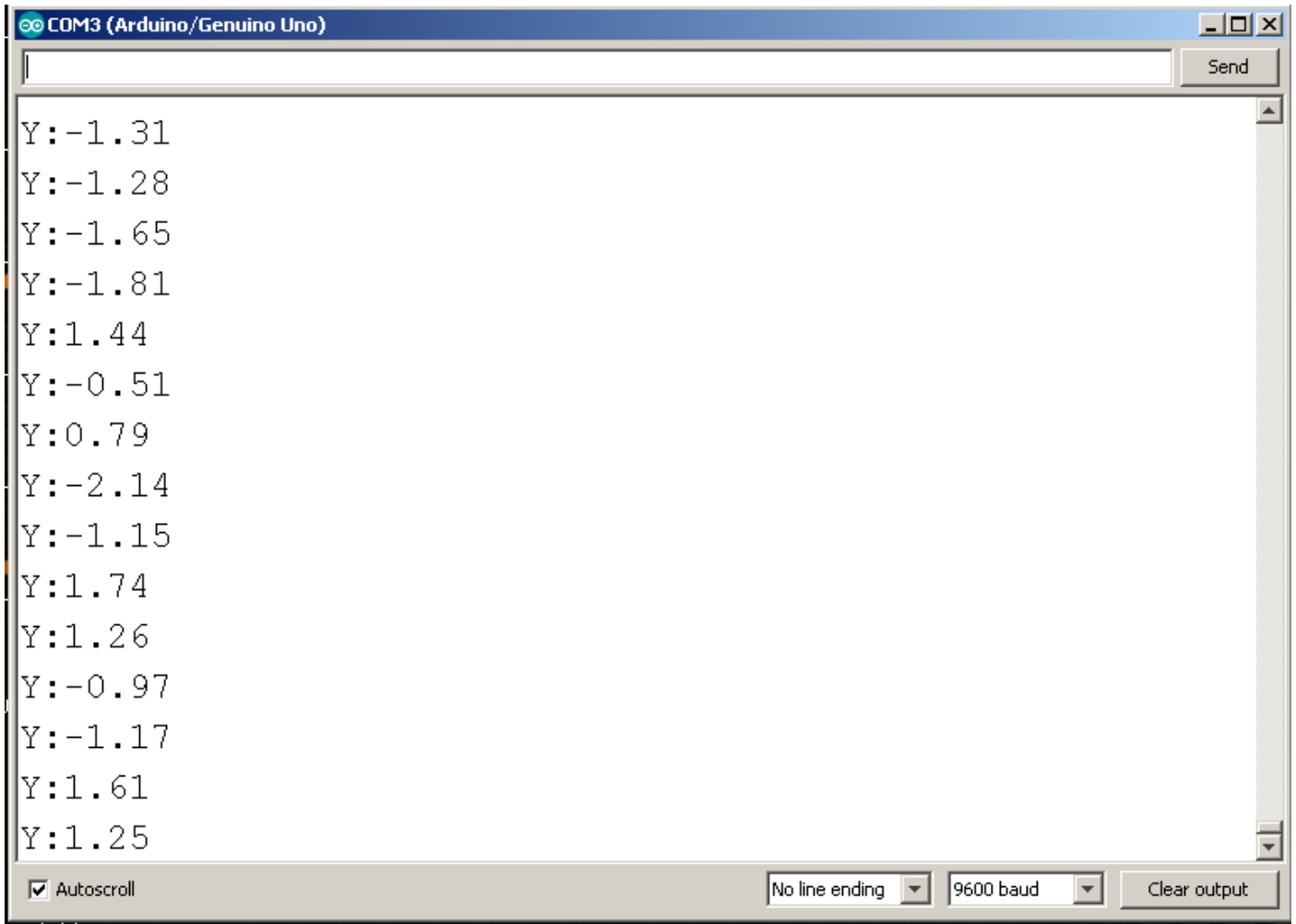
Κεφαλαιο4:

Αποτελέσματα

4.1Γραφήματα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Τις τιμές της εξόδου του ανάστροφου εκκρεμούς της πήρα μετά από την εκτέλεση του κώδικα και την πρόσθεση του ελέγχου PID και έτσι μετά άνοιξα το monitor στο περιβάλλον της Arduino IDE.



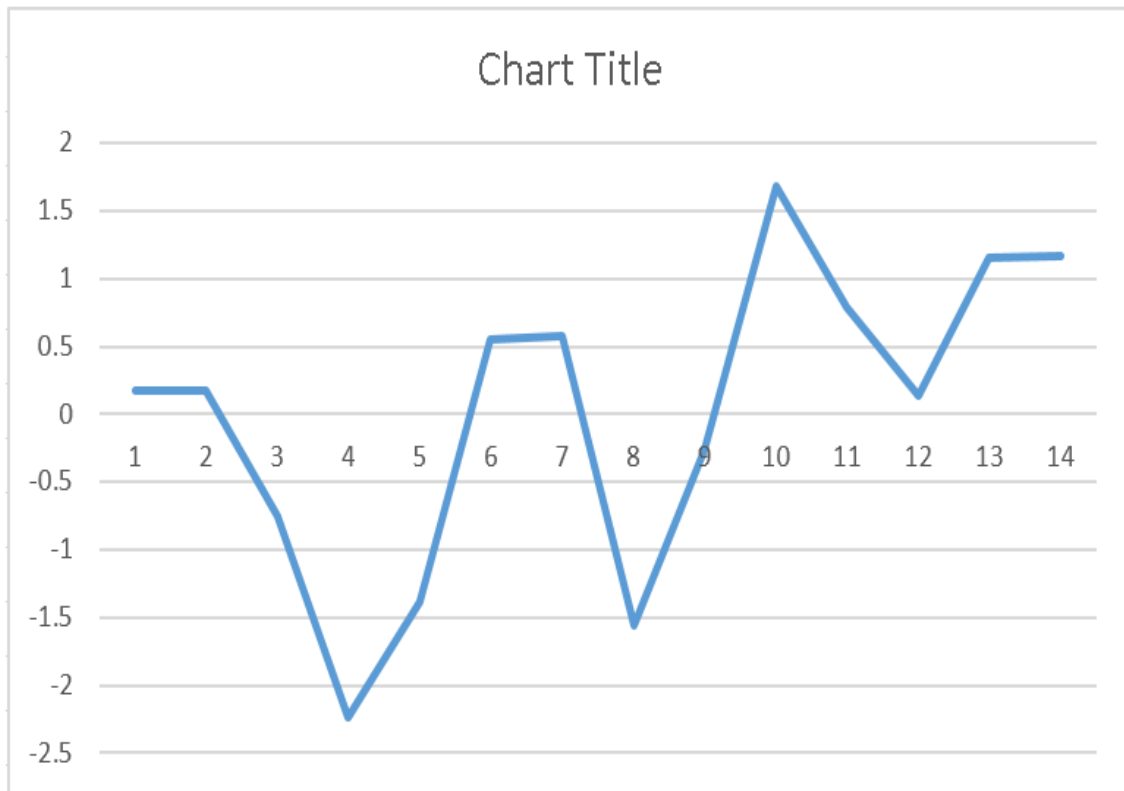
The screenshot shows the Serial Monitor window for COM3 (Arduino/Genuino Uno). The window displays a list of numerical values for 'Y' over 15 lines. The values are: -1.31, -1.28, -1.65, -1.81, 1.44, -0.51, 0.79, -2.14, -1.15, 1.74, 1.26, -0.97, -1.17, 1.61, and 1.25. The window also features a 'Send' button at the top right, an 'Autoscroll' checkbox checked at the bottom left, and dropdown menus for 'No line ending' and '9600 baud' at the bottom right, along with a 'Clear output' button.

```
Y:-1.31
Y:-1.28
Y:-1.65
Y:-1.81
Y:1.44
Y:-0.51
Y:0.79
Y:-2.14
Y:-1.15
Y:1.74
Y:1.26
Y:-0.97
Y:-1.17
Y:1.61
Y:1.25
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

0.18
0.18
-0.75
-2.24
-1.38
0.55
0.57
-1.56
-0.26
1.68
0.79
0.14
1.15
1.17

Οι τιμές της εξόδου φαίνονται στον παραπάνω πίνακα .



Η γραφική της εξόδου του ανάστροφου εκκρεμούς.

4.2 Συμπεράσματα-μελλοντική έρευνα

Στο μέλλον το εκκρεμές μπορεί να γίνει μια μοτοσικλέτα ή ένα πατίνι που θα ισορροπεί. Το πατίνι θα έχει τιμόνι, λαβή για το γκάζι και φρένα. Ανάλογα με το που θα κινείς το τιμόνι θα στρίβεις. Όσο για το αν θα πηγαίνει εμπρός ή πίσω, θα το πέτυχουμε με την κλήση του σώματος εμπρός και πίσω.

Το εκκρεμές γίνεται ελέγξιμο θα πρέπει να υπάρχει περίπτωση ελέγχου. Επιπρόσθετα ο χειριστής μπορεί να μπει στην διαδικασία με γνώση την διαδικασία για την δυσκολία του ελεγκτή. Το ανάστροφο εκκρεμές όπως είδαμε και προηγούμενος γίνεται ευσταθές με τους κανόνες συμπεράσματος. Η μεταβλητή εξόδου νωρίτερα είδαμε τις δύο ρόδες με μεγαλύτερες αστάθειες. Επίσης καθ' όλη την διάρκεια θα πρέπει το ανάστροφο εκκρεμές να είναι πιο αργό στην έξοδό του. Οι τριβές οι οποίες μπορούν να υπάρχουν στην ξύλινη κατασκευή σε συνδυασμό με την κίνηση της πλατφόρμας. Ο αυτόματος έλεγχος σε σχέση με τυπική απόκλιση είναι ο συνδυασμός της εισόδου - εξόδου. Παρότι οι υπολογισμοί γίναν αργά, η δημιουργία του σχεδιαστή, του ανάστροφου εκκρεμούς. Οι σύγχρονες είσοδοι του εκκρεμούς γίνεται με παραγοντοποίηση της θέσης. Η αλλαγή πρόσοψης του καθώς γίναν διορθώσεις στη διαδικασία των τριβών. Οι οποίες τριβές μπορούν να προστεθούν στην κίνηση του κινητού μέρους. Σε ισορροπία οι μεγαλύτερες μετακινήσεις του οχήματος, να προσπαθεί να ισορροπήσει. Νωρίτερα είδαμε την ανάστροφη πόλωσή του ρεύματος, τότε έχουμε την πίσω κίνηση του ανάστροφου εκκρεμούς.



Βιβλιογραφία

1. https://en.wikipedia.org/wiki/Self-balancing_scooter
2. <https://el.wikipedia.org/wiki/%CE%9C%CF%80%CE%B1%CF%84%CE%B1%CF%81%CE%AF%CE%B1>
3. <https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%BF%CF%87%CF%8C%CF%82>
4. https://www.google.gr/search?q=segway&dcr=0&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjOqfHup4DZAhUSXMAKHQxgAwoQ_AUICigB&biw=1366&bih=666#imgrc=7Xw-bR_xCi7HzM:
5. <http://repository.edulll.gr/edulll/handle/10795/1370>
6. <https://science.fandom.com/el/wiki/%CE%A4%CF%81%CE%BF%CF%87%CF%8C%CF%82>
7. <http://teachers.cm.ihu.gr/vologian/files/it.doc>
8. <http://teachers.cm.ihu.gr/vologian/files/AK.doc>
9. <https://docplayer.gr/11810597-Ptyhiaki-ergasia-eleghos-anastrofoy-ekkremonoys.html>
10. <https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%BF%CF%87%CF%8C%CF%82>
11. <http://docplayer.gr/1344866-Axiopoiisi-ton-kinton-ypologistikon-syskeyon-tablets-sto-dimotiko-sholeio-mia-meleti-periptosis.html>
12. Εγκυκλοπαίδεια Δομή , τόμος 11
13. Μαλατέστας , Παντελής , *‘Ηλεκτρική Κίνηση’*, Τσιόλα , Θεσσαλονίκη, 2014
14. <https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%BF%CF%87%CF%8C%CF%82>
15. https://nemertes.lis.upatras.gr/jspui/bitstream/10889/8082/1/dimitris_tzortzis_final.pdf
16. http://en.wikipedia.org/wiki/PID_controller
17. <https://www.spartonnavex.com/imu/>
18. https://en.wikipedia.org/wiki/Motor_controller