

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ



«ΚΑΤΑΣΚΕΥΗ ΟΧΗΜΑΤΟΣ ΔΥΟ ΑΞΟΝΩΝ ΜΕ ΔΥΝΑΤΟΤΗΤΑ ΙΣΟΡΡΟΠΙΑΣ ΚΑΙ ΜΗΔΕΝΙΣΜΟΥ ΣΦΑΛΜΑΤΟΣ ΜΕ ΧΡΗΣΗ ARDUINO»

ΑΛΕΞΙΟΣ ΠΟΥΛΗΣ Α.Μ.: 44252

ΒΑΣΙΛΙΚΗ ΜΑΚΡΥΓΙΑΝΝΗ Α.Μ.:44259

Επιβλέπων καθηγητής: Δρ. Γρηγόρης Νικολάου

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΜΑΡΤΙΟΣ 2017 (Ημερομηνία Υποβολής)

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ
Π. Ράλλη & Θεβών 250, 12244 Αιγάλεω,

Πρόλογος

Η παρούσα εργασία αποτελεί την πτυχιακή μας και αναφέρεται στην κατασκευή ενός αυτόνομου οχήματος δύο αξόνων, το οποίο έχει τη δυνατότητα ισορροπίας με τη χρήση του μικροεπεξεργαστή Arduino..

Η επιλογή του θέματος έγινε σε συνεννόηση με τον επόπτη καθηγητή μας, κύριο Γρηγόρη Νικολάου, τον οποίο και ευχαριστούμε θερμά για τη συνεργασία και τις πολύτιμες συμβουλές του καθ όλη τη διάρκεια της ενασχόλησής μας με το συγκεκριμένο αντικείμενο . Αφορμή για την εκπόνηση αυτού του θέματος στάθηκε το προσωπικό μας ενδιαφέρον για τους ελεγκτές PID και τα συστήματα αυτόματου ελέγχου.

Η εργασία μας χωρίζεται σε δύο μέρη. Το πρώτο μέρος περιλαμβάνει γενικές πληροφορίες γύρω από τον μικροεπεξεργαστή Arduino και το ανάστροφο εκκρεμές και το δεύτερο αναφέρεται στο σχεδιασμό και την υλοποίηση της κατασκευής.

Προσπαθήσαμε, στηριζόμενοι σε έγκυρες επιστημονικές πηγές και ενδελεχή αναζήτηση, να είμαστε περιεκτικοί και κατανοητοί, έτσι ώστε ο μη ειδικός αναγνώστης να κατανοήσει το αντικείμενο και ο ειδικός να αντλήσει επιστημονικά χρήσιμες και ουσιαστικές πληροφορίες για το θέμα.

Περίληψη

Σκοπός του οχήματος είναι η πλήρης ισορροπία στους δύο τροχούς και η ομαλή πορεία του. Το όχημα καταβάλει συνεχή προσπάθεια, για μηδενισμό των πιθανών σφαλμάτων. Χρησιμοποιήθηκε ένα γυροσκόπιο (MPU 6050), το οποίο αποτέλεσε το αισθητήριό μας για τον προσδιορισμό της θέσης του οχήματος. Μέσω του αισθητηρίου λαμβάνονταν οι απαραίτητες πληροφορίες σχετικά με την γωνία του οχήματος, συγκρίνοντας αυτή τη γωνία με την επιθυμητή γωνία, στην οποία το όχημα ισορροπεί. Η χρήση του μικροεπεξεργαστή Arduino σε συνδυασμό με τα ηλεκτρονικά εξαρτήματα που χρησιμοποιήθηκαν είναι υπεύθυνα για την ισορροπία του οχήματος. Βασικό κομμάτι για την επίτευξη της εργασίας μας, ήταν η χρήση PID ελεγκτή, με τον οποίο θέσαμε το όχημα μας σε έναν κλειστό βρόχο με ανάδραση. Ήμαστε σε θέση να γνωρίζουμε την απόκλιση της τρέχουσας γωνίας από την επιθυμητή γωνία και προγραμματιστικά να μηδενίσουμε το σφάλμα.

Abstract

Main purpose of the vehicle, is balancing in two wheels and the smooth motion of it. The vehicle tries to balance constantly by eliminating the errors. It succeeds this by using a gyroscope (MPU 6050), which was our main sensor for defining vehicle's position. Gyroscope gave us information about the vehicle's angle and then we compared it with the desire angle of the vehicle, in which the vehicle balances. The usage of the micro controller Arduino with combination of some electronics devices make our vehicle to balance. Main component of our project is the use of a PID controller. With PID controller we put our vehicle in a close loop with feedback, so we were able to know every second the error and eliminate it.

Περιεχόμενα

Πρόλογος.....	2
Περίληψη.....	3
Abstract.....	3
ΚΕΦΑΛΑΙΟ 1	
1.1 Ιστορική Αναδρομή Arduino.....	9
1.2 Η πλακέτα Arduino	9
1.3 Hardware.....	10
1.4 Ανάλυση ATmega328P.....	13
1.4.1 Flash Memory.....	13
1.4.2 SRAM Memory.....	13
1.4.3 EEPROM Memory.....	14
1.4.4 FTDI.....	14
1.5 Arduino Shields.....	14
1.6 Software – Λογισμικό.....	16
1.7 Σύνδεση Arduino με τον υπολογιστή.....	17
ΚΕΦΑΛΑΙΟ 2	
2.1 Θεωρία Ανάστροφου εκκρεμούς.....	21
2.2 Μαθηματική Ανάλυση του Ανάστροφου εκκρεμούς.....	21
2.2.1 Τυχαία Ασταθή Κατάσταση.....	22
2.2.2. Λύση χωρίς Ανάδραση.....	23
2.2.3 Λύση με Ανάδραση.....	25
ΚΕΦΑΛΑΙΟ 3	
3.1 Τι είναι το γυροσκόπιο.....	27
3.1.1 Που χρησιμοποιείται το γυροσκόπιο.....	28
3.1.2 Πως λειτουργεί το γυροσκόπιο.....	28
3.1.3 Ανάλυση τρόπου λειτουργίας γυροσκοπίου.....	29
3.2 Τι είναι το επιταχυνσιόμετρο.....	29
3.2.1 Που χρησιμοποιείται το επιταχυνσιόμετρο.....	29
3.2.2 Πως λειτουργεί το επιταχυνσιόμετρο.....	30
3.2.3 Ανάλυση τρόπου λειτουργίας.....	30

ΚΕΦΑΛΑΙΟ 4

4.1 Περιγραφή MPU 6050.....	31
4.2 Εφαρμογές των MPU 6050.....	34
4.3 Επιπλέον χαρακτηριστικά του MPU 6050.....	34
4.4 Motion Processing.....	35
4.5 Γυροσκόπιο στο MPU 6050.....	36
4.6 Επιταχυνσιόμετρο στο MPU 6050.....	36

ΚΕΦΑΛΑΙΟ 5

5.1 Bluetooth.....	38
5.1.1 Εφαρμογή.....	38
5.1.2 Λειτουργία Bluetooth.....	38
5.2 Τι είναι το BluetoothModule.....	39
5.2.1 Συνδεσμολογία Arduino με HC-05.....	40
5.3 Εφαρμογή Android.....	40
5.4 I2C Protocol.....	42
5.4.1 Λογικές στάθμες των γραμμών SCL και SDA.....	42
5.4.2 Το φυσικό πρωτόκολλο του διαύλου I2C.....	42
5.4.3 Διευθυνσιοδότηση συσκευών I2C.....	43
5.4.4 Εφαρμογές του διαύλου I2C.....	43

ΚΕΦΑΛΑΙΟ 6

6.1 Συμπληρωματικό φίλτρο.....	44
6.2 Atan2.....	44

ΚΕΦΑΛΑΙΟ 7.....

7.1 Ελεγκτής PID.....	45
7.2 Βρόχος ανάδρασης.....	45
7.3 Οι όροι PID.....	47
7.4 Χαρακτηριστικά των όρων PID.....	49

ΚΕΦΑΛΑΙΟ 8.....

8.1 Κινητήρες.....	50
8.2 Χαρακτηριστικά κινητήρα.....	50
8.3 Συνδεσμολογία με Arduino.....	50
8.4 Ρόδες.....	51

ΚΕΦΑΛΑΙΟ 9

9.1 Τι είναι ο 3Dεκτυπωτής.....	53
9.2 Ψηφιακό σχέδιο.....	53
9.3 Παρουσίαση κατασκευής.....	55

ΚΕΦΑΛΑΙΟ 10

10.1 Συνδεσμολογία κατασκευής.....	57
------------------------------------	----

ΚΕΦΑΛΑΙΟ 11

11.1 Κώδικας.....	58
11.2 Σχόλια Κώδικα.....	63

ΚΕΦΑΛΑΙΟ 12

12.1 Συμπεράσματα.....	67
------------------------	----

ΒΙΒΛΙΟΓΡΑΦΙΑ.....	69
--------------------------	-----------

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτοχ(η) υπογεγραμμένος / η
..... ΠΑΧΛΗΣ ΑΝΔΡΕΑΣ

του ΙΩΑΝΝΗ....., με αριθμό μητρώου 44252
φοιτητής / τρια του Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν
αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα
παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο
περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή
μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν
λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα.
Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των
συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του
έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η
Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή.
Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού
δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο
άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Επίσης δηλώνω υπεύθυνα ότι έχω παρακολουθήσει το σεμινάριο συγγραφής και
εκπόνησης πτυχιακής εργασίας που διοργανώνεται από το Τμήμα Μηχανικών Αυτοματισμού
Τ.Ε. κατά το Χειμερινό/Εαρινό Εξάμηνο του Ακ. Έτους 2017.

Ο Δηλών



Ημερομηνία

9/9/2019

ΚΕΦΑΛΑΙΟ 1

1.1 Ιστορική Αναδρομή

Η πλακέτα Arduino κατασκευάστηκε το 2005 στην Ιταλία από τους Massimo Banzi, David Cuartielles, Tom Igoe, David Mellis και Gianluca Martino με κύριο σκοπό τη δημιουργία μιας συσκευής η οποία θα έχει τη δυνατότητα να αλληλεπιδρά με το περιβάλλον και ταυτόχρονα να έχει χαμηλό κόστος.

Το Arduino αναπτύχθηκε σύμφωνα με την πλατφόρμα Wiring και είχε ως στόχο να αναπτύξει ένα δικό της περιβάλλον προγραμματισμού, αλλά σχεδιαστικά και συντακτικά να μοιάζει με την εκδοχή Processing. Η πλακέτα Arduino δημιουργήθηκε με την εμφάνιση της ψηφιακής τεχνολογίας και των μικροεπεξεργαστών.

Όλες οι λειτουργίες που μέχρι τότε έπρεπε να γίνουν ενσύρματα με πολύπλοκες και χρονοβόρες διατάξεις μπορούσαν πλέον να αντικατασταθούν από λογισμικά προγράμματα. Πρακτικά το λογισμικό είναι πιο εύκολο να τροποποιηθεί από ότι το hardware με αποτέλεσμα να μπορεί να αλλάξει όλη η λογική της πλακέτας πολύ πιο γρήγορα συγκριτικά με το χρόνο που απαιτεί μια τροποποίηση στον σχεδιασμό κυκλώματος ή στην συγκόλληση διατάξεων. Αποτέλεσμα αυτού ήταν το κόστος της πλακέτας να μειωθεί αρκετά.

Τα δύο κύρια μέρη του Arduino είναι η πλακέτα, δηλαδή το κομμάτι hardware, με το οποίο συνήθως ασχολείται ο κατασκευαστής και το κομμάτι Arduino IDE δηλαδή το software, με το οποίο ασχολείται ο χρήστης κατά τη διάρκεια του προγραμματισμού δημιουργώντας ένα sketch, δηλαδή ένα μικρό πρόγραμμα που φορτώνει εντολές στον μικροελεγκτή της πλακέτας Arduino.

1.2 Η πλακέτα Arduino

Το Arduino αποτελεί μια πλακέτα ανοικτού κώδικα με ενσωματωμένο τον μικροελεγκτή ATmega της Atmel. Είναι εύχρηστος τόσο στο κομμάτι του software όσο και του hardware. Με τον κατάλληλο προγραμματισμό μπορεί να χρησιμοποιηθεί ως ένας μικρός υπολογιστής, αφού δίνει τη δυνατότητα στο χρήστη να συνδέσει πολλές μονάδες εισόδου/εξόδου.

Η γλώσσα προγραμματισμού Wiring, θεωρείται μια εύκολη γλώσσα στη σύνταξη και διατίθεται σε πλατφόρμες Windows, MAC και Linux.

Ο λόγος που επιλέχθηκε ο μικροελεγκτής Arduino είναι η μεγάλη βάση δεδομένων από την οποία μπορεί ο καθένας να αντλήσει τις κατάλληλες πληροφορίες και να βρει τεκμηριωμένες απαντήσεις σε πολλά ερωτήματα. Επίσης οι πλακέτες Arduino κυκλοφορούν στην αγορά προκατασκευασμένες, δηλαδή δεν χρειάζονται συναρμολόγηση και σε αρκετά προσιτές τιμές. Επιπλέον ενώ τα περισσότερα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows, το Arduino τρέχει σε Machinstoh OSX και Linux. Τέλος η πλακέτα Arduino διαθέτει πληθώρα δυνατοτήτων με βάση τον αυτοματισμό όπως την κίνηση servo, stepper και DC κινητήρων, τη λήψη πληροφοριών από πολλά αισθητήρια που ικανοποιούν ένα μεγάλο εύρος εφαρμογών. Η δημοφιλέστερη εκδοχή του Arduino είναι η Duemilanove/UNO που βασίζεται στο ολοκληρωμένο ATmega328, έναν 8-bit RICS μικροελεγκτή ο οποίος χρονίζει στα 16MHz.

1.3 Hardware

Ο επεξεργαστής Arduino αποτελείται από έναν μικροελεγκτή Atmel AVR, ειδικότερα ο Arduino UNO αποτελείται από το ATmega328P.

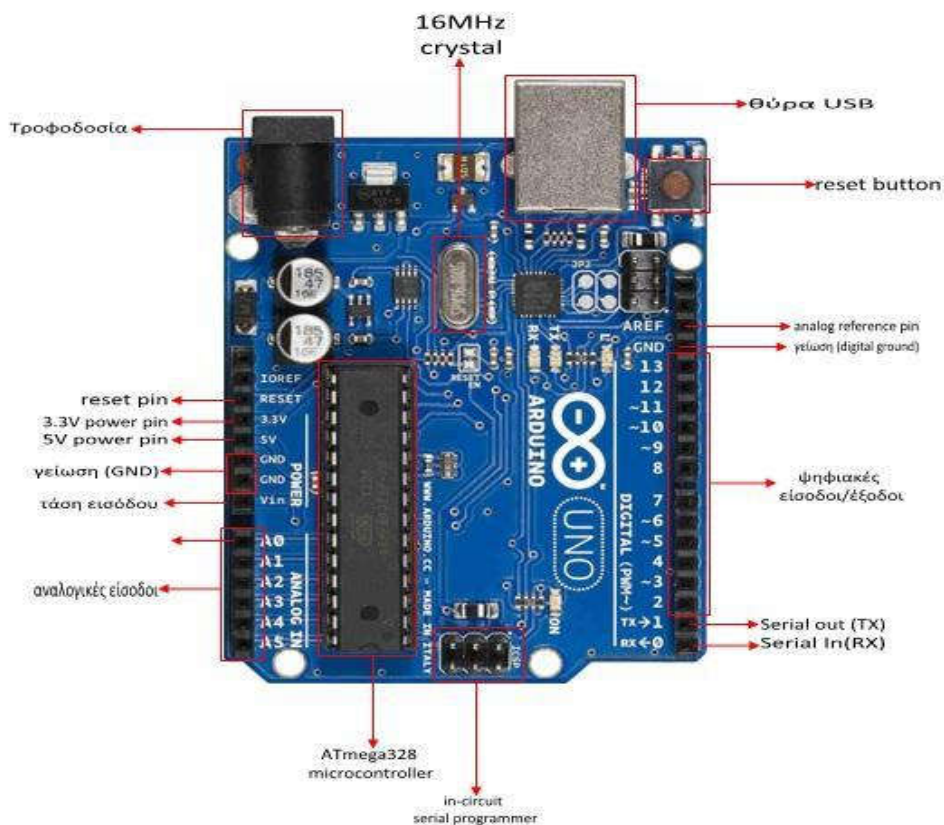
Διαθέτει :

- δεκατέσσερις (14) ψηφιακές εισόδους/εξόδους οι οποίες καθορίζονται από το εκάστοτε πρόγραμμα αν θα χρησιμοποιηθούν ως είσοδοι ή ως εξοδοι.
- έξι (6) ψευδοαναλογικές εισόδους, τις ψηφιακές θύρες 3,5,6,9,10 και 11 οι οποίες λειτουργούν με το σύστημα PWM (Pulse Width Modulation), το οποίο παίρνει ένα εύρος τιμών από το 0 έως το 255. Στην πραγματικότητα δεν είναι αναλογικό σύστημα, έτσι θέτοντας στην έξοδο την τιμή 127 για παράδειγμα δεν σημαίνει ότι η έξοδος θα έχει την τιμή 2.5V αντί της κανονικής των 5V, αλλά 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.
- Οι θύρες 0 (RX) και 1 (TX) χρησιμοποιούνται για να λαμβάνουν και να μεταδίδουν σειριακά δεδομένα. Όταν δηλαδή 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 μια άλλη συσκευή. Απαραίτητη προϋπόθεση είναι η ενεργοποίηση του serial interface.
- Οι θύρες 2 και 3 μπορούν αν λειτουργήσουν και ως εξωτερικά interrupt 0 και 1 αντίστοιχα. Μπορούν να ρυθμιστούν μέσω προγράμματος έτσι ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι. Χρησιμοποιούν πολύ σε περιπτώσεις όπου συμβαίνουν αλλαγές στην τάση, τότε η κανονική ροή του προγράμματος σταματάει και εκτελείται μια συγκεκριμένη συνάρτηση, κυρίως σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.
 - Οι θύρες A0 έως A5 (6 θύρες) αποτελούν αναλογικές εισόδους, κάνοντας χρήση του ADC δηλαδή analog to digital. Εάν ένα από αυτά τα pin τροφοδοτηθεί με μια τάση η οποία μπορεί να κυμανθεί από 0V έως μια τάση αναφοράς έως 5V, τότε μπορεί να μέσα από το πρόγραμμα να διαβαστεί η αντίστοιχη ψηφιακή τιμή ως ένας αριθμός 10-bit από το 0 για 0V μέχρι το 1023 ($2^{10}-1$ bit) για τα 5V.
 - Η θύρα με ένδειξη AREF χρησιμεύει για να οριστεί η τάση αναφοράς εξωτερικά, χωρίς να απαιτείται κάποιος προγραμματισμός.
 - Η θύρα με ένδειξη RESET , όταν γειωθεί με κάποιο από τα 3 pin που φέρουν την ένδειξη GND έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
 - Η θύρα με ένδειξη 3.3V, τροφοδοτεί διατάξεις/συσκευές ή αισθητήρες με τάση 3.3V και μέγιστη ένταση 50mA τα οποία προέρχονται από τον ελεγκτή Serial-over-USB και όχι από κάποια εξωτερική τροφοδοσία.
 - Η θύρα με ένδειξη 5V, τροφοδοτεί και εκείνη συσκευές ή οτιδήποτε άλλο με τάση 5V. Η τάση αυτή προέρχεται είτε από τη θύρα USB ή από εξωτερική τροφοδοσία, η οποία περνάει από ρυθμιστή τάσης για να σταθεροποιηθεί στην τιμή των 5V.
 - Οι θύρες με ένδειξη GND αποτελούν τις γειώσεις της πλακέτας.
 - Η θύρα με ένδειξη Vin σε συνδυασμό με μια από τις θύρες γείωσης μπορεί να λειτουργήσει ως εξωτερική τροφοδοσία του Arduino. Αν όμως υπάρχει ήδη εξωτερική τροφοδοσία τότε μπορεί από μόνη της να χρησιμεύσει για να τροφοδοτήσει συσκευές και εξαρτήματα με την πλήρη τάση εξωτερικής τροφοδοσίας, πριν αυτή περάσει από τον ρυθμιστή τάσης, δηλαδή από 7V έως 12V.
 - Η θύρα USB χρησιμεύει για την φόρτωση των προγραμμάτων από τον υπολογιστή ύστερα από τον κατάλληλο προγραμματισμό. Στη συνέχεια οδηγούνται στο ολοκληρωμένο FDTI στο οποίο διαμορφώνονται στην

κατάλληλη μορφή έτσι ώστε να μπορεί ο μικροελεγκτής να τις διαβάσει και να τις εκτελέσει.

- Ο διακόπτης micro-switch χρησιμεύει για την επανεκκίνηση του Arduino.
- Τέλος πάνω στην πλακέτα υπάρχουν τέσσερα σχεδόν επιφανειακά LED. Το LED με την ένδειξη POWER ενεργοποιείται όταν τεθεί σε λειτουργία το Arduino. Τα LED με ενδείξεις TX και RX αντίστοιχα χρησιμοποιούνται ως ένδειξη λειτουργίας του serial interface, όταν το Arduino λαμβάνει ή στέλνει δεδομένα. Ο έλεγχος αυτών των LED γίνεται μέσω της θύρας USB επομένως τα συγκεκριμένα δεν ενεργοποιούνται όταν η επικοινωνία γίνεται μέσω των ψηφιακών θυρών 0 και 1. Τέλος υπάρχει το LED με την ένδειξη L, του οποίου ο ρόλος είναι κυρίως δοκιμαστικός, δηλαδή ακόμα και αν δεν συνδεθεί τίποτα στην πλακέτα και η πλακέτα λάβει την τιμή HIGH μέσω ενός προγράμματος το συγκεκριμένο LED θα ενεργοποιηθεί.



Εικόνα 1: Άνοψη του ελεγκτή Arduino

1.4 Ανάλυση ATmega328P

Τα βασικά χαρακτηριστικά του ολοκληρωμένου ATmega 328 είναι τα παρακάτω:

- Υψηλή απόδοση, χαμηλή ισχύς,
- Διαθέτει προηγμένη αρχιτεκτονική RISC
- Υψηλή αντοχή στη μνήμη
- I/O: 23 προγραμματιζόμενα I/O
- Τάση συστήματος από 2.7V έως 5.5V
- Εύρος θερμοκρασίας : αυτόματο εύρος από -40°C έως +125 °C
- Ταχύτητα
 - από 0 έως 8MHz στα 2.7V έως 5.5V
 - από 0 έως 16MHz στα 4.5V έως 5.5V
- Χαμηλή κατανάλωση ενέργειας
 - ενεργή κατάσταση: 1.5mA στα 3V – 4MHz
 - κατάσταση διακοπής ρεύματος: 1μΑ στα 3V

Το ολοκληρωμένο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων, τη flash memory, την SRAM memory και την EEPROM memory.

1.4.1 Flash memory

Έχει χωρητικότητα 32Kb εκ των οποίων τα 2Kb χρησιμοποιούνται από το firmware του Arduino, το οποίο είναι ενσωματωμένο από τον κατασκευαστή, στην ορολογία του Arduino ονομάζεται bootloader και είναι απαραίτητο για την εγκατάσταση προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 30Kb της μνήμης χρησιμοποιούνται για την αποθήκευση των προγραμμάτων αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Χαρακτηριστικά της μνήμης είναι πως δεν χάνει τα δεδομένα της με την απώλεια της τροφοδοσίας ή κάνοντας reset τον μικροελεγκτή ενώ δεν προορίζεται για χρήση runtime.

1.4.2 SRAM memory

SRAM memory δηλαδή static random access memory είναι η μνήμη που χρησιμοποιούν τα προγράμματα κατά τη διάρκεια του runtime για να αποθηκεύσουν μεταβλητές, πίνακες κ.α. Λειτουργεί σαν τη μνήμη του υπολογιστή, δηλαδή χάνει τα δεδομένα της όταν η τροφοδοσία του Arduino σταματήσει ή όταν γίνει reset της

συσκευής. Η μνήμη καταλαμβάνει χώρο 2048bytes κατά τη διάρκεια μιας κανονικής λειτουργίας.

1.4.3 EEPROM memory

Το τελευταίο μέρος της μνήμης είναι η EEPROM και καταλαμβάνει 1024bytes και η συγκεκριμένη μνήμη χρησιμοποιείται μόνο για ανάγνωση (readonly).

Χαρακτηριστικό της μνήμης EEPROM είναι πως δεν μπορεί να επαναπρογραμματιστεί για περισσότερες από 100.000 φορές, δηλαδή έχει όριο ζωής.

1.4.4 FTDI

Το Arduino χρησιμοποιεί ένα FTDI ολοκληρωμένο, το οποίο είναι απαραίτητο για την σειριακή επικοινωνία των μικροελεγκτών ATmega και των υπολογιστών, μπορεί δηλαδή να χαρακτηριστεί και ως μετατροπέας σειριακής θύρας σε USB.

1.5 Arduino Shields

Στο εμπόριο υπάρχουν διαθέσιμες πολλές πλακέτες Arduino που καλύπτουν μεγάλο είδος εφαρμογών, καθώς και πολλά εξαρτήματα που είναι κατασκευασμένα με τρόπο τέτοιο έτσι ώστε να “συνεργάζονται” άψογα με τις πλακέτες Arduino και να διευρύνουν τις δυνατότητες τους. Παρακάτω παρουσιάζεται μια λίστα με τα λεγόμενα Arduino Shields και τις βασικές τους λειτουργίες.

a/a	Ονομασία	Εφαρμογή
1	EthernetShield	Δυνατότητα σύνδεσης των Arduino με το internet.
2	RelayShield	Αποτελεί μια πλακέτα με τέσσερα μηχανικά ρελέ, που δίνει τη δυνατότητα ελέγχου της υψηλής τάσης.
3	ProtoShield	Επιτρέπει την εύκολη σύνδεση μεταξύ breadboard και Arduino
4	Motor Shield	Δυνατότητα ελέγχου κίνησης και κατεύθυνσης των κινητήρων σε οποιαδήποτε εφαρμογή Arduino
5	LCD Shield	Συνδέεται στα I2C pins του Arduino και δίνει τη δυνατότητα ελέγχου μιας οθόνης 16x2 χαρακτήρων, έως 3 ακίδες φωτισμού και 5 ακίδες πληκτρολογίου.
6	CapacityTouchpadShield	Επιτρέπει τη δημιουργία interfaces

7	CAN-BUS Shield	Χρησιμοποιεί ένα MCP2515 CAN buscontroller με interface SPI και πομποδέκτη MCP2551 CAN, δίνοντας έτσι τη δυνατότητα CAN-BUS στο Arduino.
8	SmokeDetectorShield	Δίνει τη δυνατότητα ανίχνευσης αερίου στον αέρα και το διαβάζει σαν αναλογική τιμή, χρησιμεύει σε εφαρμογές για συστήματα ανίχνευσης καπνού.
9	NegativeVoltageGenerationShield	Χρησιμεύει σε περίπτωση που θέλουμε να δημιουργήσουμε αρνητική τάση στο Arduino.
10	WaveShield	Χρησιμοποιείται σε εφαρμογές ήχου (audio) και βελτιώνει την χαμηλή ποιότητα ήχου σε υψηλή. Μπορεί να αναπαραγάγει μέχρι και 22KHz 12 μη αποσυμπιεσμένα αρχεία ήχου οποιουδήποτε μήκους. Τα αρχεία ήχου διαβάζονται από μια κάρτα SD/MMC
11	CISECO ProtoXShield	Είναι ένα prototyping board με SMT pads και serial breakout.
12	64-Button Shield	Με τη συγκεκριμένη πλακέτα μπορούμε να συνδέσουμε έως και 64 buttons στο Arduino.
13	JoystickShield	Παρέχει απλές αναλογικές εισόδους και τέσσερα ξεχωριστά κουμπιά και ένα button κάτω από το joystick.
14	GSM/GPRS Shield	Επιτρέπει τη σύνδεση της πλακέτας Arduino με δίκτυο κινητής τηλεφωνίας GSM/GPRS, μας δίνει τη δυνατότητα να καλούμε ένα αριθμό τηλεφώνου ή να στέλνουμε μήνυμα (SMS) σε οποιαδήποτε επαφή.
15	GameduinoShield	Αποτελεί προσαρμογέα παιχνιδιών για Arduino και διαθέτει βύσμα για οθόνη VGA και ηχεία
16	MicroSDShield	Δίνει τη δυνατότητα μαζικής αποθήκευσης και καταγραφής δεδομένων.
17	NFC/RFID Shield	Χρησιμοποιείται σε εφαρμογές RFID ή NFC στα 13.56MHz
18	AdafruitNeoPixelShields	Μπορούμε να ελέγχουμε ένα matrix των 5x8 LED RGB χρησιμοποιώντας μόνο έναν ακροδέκτη.
19	MP3 Player Shield	Δίνει τη δυνατότητα να προσθέσουμε ηχητικά εφέ ή μουσική σε ένα project.
20	Camera Shield	Χρησιμεύει σε εντολές που αφορούν την κάμερα.
21	GPS LoggerShield	Αποτελείται από μια μονάδα GPS και έχει σχεδιαστεί για την καταγραφή δεδομένων σε μια κάρτα SD.
22	Wireless SD Shield	Επιτρέπει την απομακρυσμένη επικοινωνία Arduino.
23	cc3000 Wi-FiShield	Απλοποιεί τη διαδικασία σύνδεσης στο δίκτυο και χρησιμεύει σε εφαρμογές χαμηλού κόστους και χαμηλής κατανάλωσης
24	ESP8266 Wi-FiShield	Αποτελεί μια μονάδα που χρησιμεύει σε έργα σχετικά με το Internet of Things (IoT)
25	HC-05 BluetoothShield	Χρησιμεύει για ασύρματη σειριακή σύνδεση με Bluetooth SPP (Serial Port Protocol)

1.6 Λογισμικό – Software

Το Arduino IDE (Integrated Development Environment – Ολοκληρωμένο Περιβάλλον ανάπτυξης) είναι ένα λογισμικό πρόγραμμα που βασίζεται σε γλώσσα Java η οποία διαμορφώθηκε μέσα από τη γλώσσα Processing.

Η processing αποτελεί μια γλώσσα προγραμματισμού ανοιχτού κώδικα, δημιουργήθηκε το 2001 από τους Benjamin Fry και Casey Reas με σκοπό να χρησιμοποιείται για προγραμματισμό εικόνων, animation και ήχο, όχι μόνο από επαγγελματίες χρήστες, αλλά και από αρχάριους. Η ανάπτυξη της γλώσσας βασίστηκε στη γλώσσα Java, όμως ο τρόπος γραφής της και ο προγραμματισμός της διαφέρουν από αυτή.

Ουσιαστικά το λογισμικό IDE αποτελεί ένα περιβάλλον μέσα στο οποίο υπάρχει η δυνατότητα εγγραφής κώδικα, παρακολούθησης της σειριακής επικοινωνίας του υπολογιστή με το Arduino και λειτουργίες εργαλείων για εφαρμογές. Τα προγράμματα που δημιουργούνται μέσα στο IDE ονομάζονται sketches και αποθηκεύονται με την μορφή αρχείου “.ino”. Χαρακτηριστικό του λογισμικού είναι η δυνατότητα εμφάνισης σφάλματος κατά τη μεταφόρτωση ενός κώδικα από τον υπολογιστή στην πλακέτα, αλλά και άλλων πληροφοριών που σχετίζονται με το κομμάτι του προγραμματισμού. Συγκεκριμένα υπάρχει και η επιλογή της επικύρωσης του προγράμματος με την οποία χωρίς να χρειάζεται να μεταφορτώσουμε το πρόγραμμά μας στην πλακέτα του Arduino, ακόμα και χωρίς να έχουμε συνδέσει το Arduino στον ηλεκτρονικό υπολογιστή μας, μπορούμε να συντάξουμε κώδικα και να δούμε τα τυχόν συντακτικά του λάθη. Τα πιθανά λάθη στη σύνταξη του κώδικα, υπογραμμίζονται στην σειρά που βρίσκεται η λανθασμένη εντολή και συνοδεύονται από μήνυμα σφάλματος. Το μήνυμα σφάλματος εμφανίζεται στο κάτω μέρος του προγραμματιστικού περιβάλλοντος και μας ενημερώνει για περισσότερες πληροφορίες σχετικά με το σφάλμα του κώδικα.

Η πλακέτα Arduino, έχει ένα ακόμα πλεονέκτημα όσο αναφορά τα σφάλματα, το οποίο σχετίζεται με το hardware κομμάτι. Για παράδειγμα, έστω ότι έχουμε υλοποιήσει ένα απλό κύκλωμα στο οποίο θέλουμε να αναβοσβήνουμε ένα λαμπάκι LED. Αν έχουμε συνδέσει λανθασμένα την τάση με την οποία θα τροφοδοτήσουμε το LED με την γείωση του κυκλώματος, η πλακέτα δεν αφήνει να ολοκληρώσουμε το ανέβασμα του προγράμματος. Εμφανίζει δηλαδή λάθος στο compile του προγράμματός μας, προστατεύοντας με αυτόν τον τρόπο την ίδια την πλακέτα από πιθανές βλάβες.

1.7 Σύνδεση Arduino με τον υπολογιστή

Για την σύνδεση του ηλεκτρονικού υπολογιστή/φορητού υπολογιστή με το Arduino, εκτός από την πλακέτα του Arduino Uno, θα χρειαστούμε και ένα καλώδιο USB και ακολουθούμε την παρακάτω διαδικασία.



Εικόνα 2

Στη συνέχεια από την επίσημη ιστοσελίδα του Arduino, <https://www.arduino.cc/en/Main/Software>, επιλέγουμε το κατάλληλο Arduino IDE πρόγραμμα ανάλογα με το λειτουργικό μας σύστημα (Windows, Mac, Linux) και το μεταφορτώνουμε δωρεάν.

Download the Arduino IDE



ARDUINO 1.8.5
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.7 Lion or newer

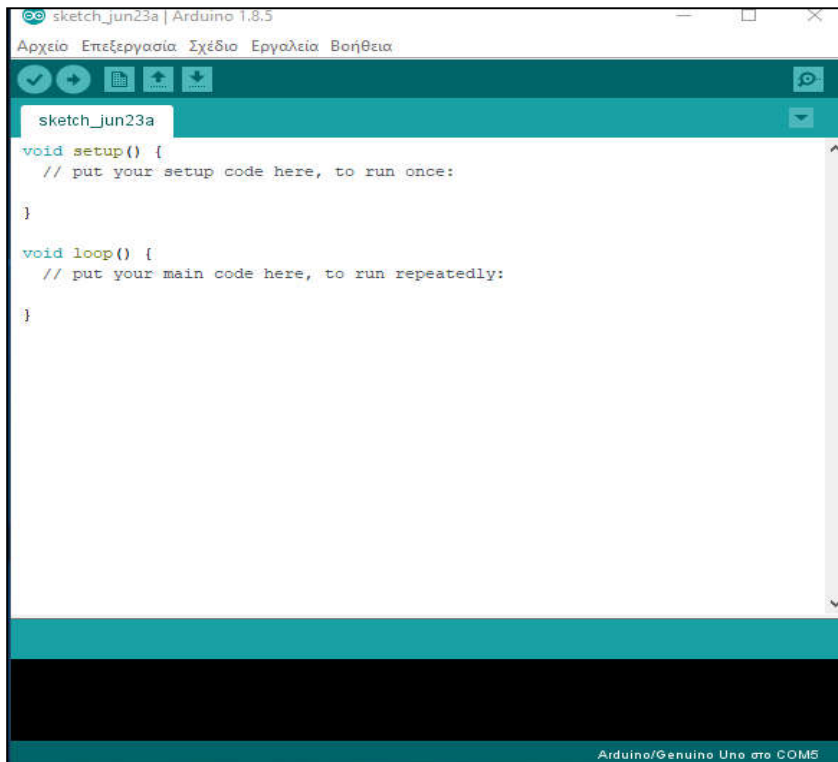
Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Εικόνα 3

Συνεχίζουμε, επιλέγοντας το αρχείο που μεταφορτώσαμε και ακολουθούμε τα βήματα εγκατάστασης. Επιλέγουμε τον επιθυμητό προορισμό, όπου θέλουμε να εγκαταστήσουμε το πρόγραμμα στον υπολογιστή μας και την γλώσσα του προγραμματιστικού περιβάλλοντος. Το Arduino IDE είναι διαθέσιμο σε πολλές γλώσσες, μεταξύ των οποίων και η ελληνική.

Εφόσον εγκαταστήσουμε το Arduino IDE, εκκινούμε το πρόγραμμα πατώντας το εικονίδιο “Arduino.exe” στην επιφάνεια εργασίας ή στο menu έναρξης.



Εικόνα 4

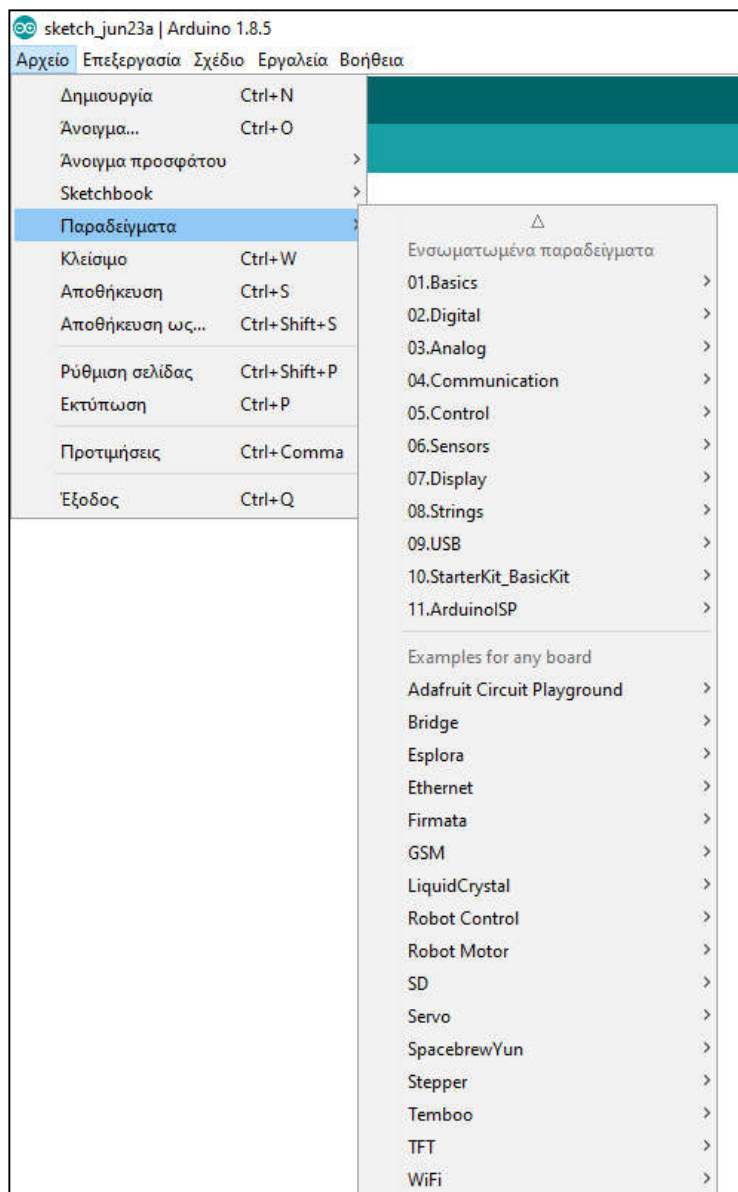
Στην εικόνα 4 παρουσιάζεται η πλατφόρμα IDE. Στην γραμμή εργαλείων υπάρχουν οι επιλογές:

- Αρχείο για δημιουργία, αποθήκευση, κλείσιμο ενός προγράμματος αλλά και βασικά παραδείγματα.
- Επεξεργασία που αφορά κυρίως την επεξεργασία του κειμένου, ενός κώδικα αλλά και τις επιλογές αντιγραφή/επικόλληση.
- Σχέδιο όπου μας δίνονται οι επιλογές για το «ανέβασμα» ενός προγράμματος στην πλακέτα αλλά και για μεταγλώττιση/επικύρωση.
- Εργαλεία με τις αντίστοιχες επιλογές για αυτόματη διόρθωση, αρχειοθέτηση σχεδίου αλλά παρακολούθηση σειριακής πόρτας. Επιπλέον στην επιλογή «Πλακέτα» γίνεται η σύνδεση/ επικοινωνία της εκάστοτε πλακέτας Arduino με τον υπολογιστή για να γίνει η μεταφόρτωση του κώδικα.
- Τέλος η επιλογή βοήθεια, στην οποία υπάρχουν πληροφορίες για το περιβάλλον IDE, το Arduino αλλά και επιλογές όπως επίλυση προβλήματος.

Ακριβώς κάτω από τη γραμμή εργαλείων, εμφανίζονται με τη σειρά τα εικονίδια από τα οποία μπορούμε να κάνουμε τις λειτουργίες επικύρωση/ ανέβασμα/ δημιουργία/ άνοιγμα/ αποθήκευση.

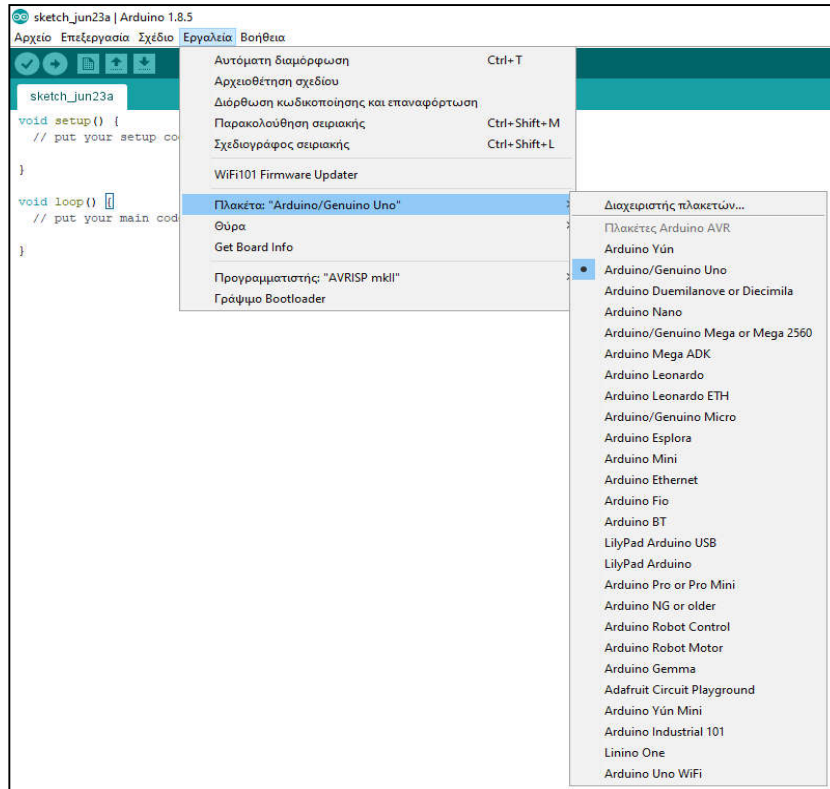
Στο μέρος του κώδικα “ voidsetup” συντάσσουμε τις εντολές που θα εκτελεστούν μόνο μια φορά στο πρόγραμμα, ενώ στο “voidloop” συντάσσουμε το κύριο μέρος του προγράμματός μας.

Επιλέγοντας το «Αρχείο» από την γραμμή εργαλείων, και στην συνέχεια το «Παραδείγματα» μπορούμε να επιλέξουμε διάφορα βασικά παραδείγματα, που θα βοηθήσουν στην πρώτη επαφή με την πλατφόρμα.



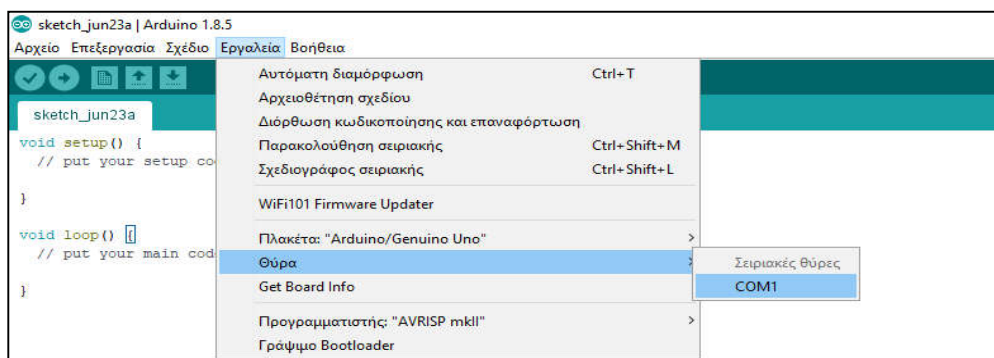
Εικόνα 5

Πριν «ανεβάσουμε το πρόγραμμα μας, στην πλακέτα του Arduino, πρέπει να βεβαιωθούμε ότι έχουμε συνδέσει το Arduino με τον υπολογιστή μέσω του καλωδίου usb. Τότε από την γραμμή εργαλείων επιλέγουμε το «εργαλεία» και μετά επιλέγουμε το «πλακέτα».



Εικόνα 6

Στη συνέχεια, από τη λίστα με τις πλακέτες του Arduino, επιλέγουμε αυτή που χρησιμοποιούμε. Στην προκειμένη περίπτωση το Arduino Uno. Τελευταίο βήμα πριν ανεβάσουμε το πρόγραμμα μας, είναι να επιλέξουμε την θύρα (PORT) στην οποία έχουμε συνδέσει την πλακέτα μας μέσω usb. Επιλέγουμε από την γραμμή εργαλείων το «Εργαλεία», μετά το «Θύρα» και μετά το COMx, όπου x ο αριθμός της πόρτας που έχουμε συνδέσει το Arduino.



Εικόνα 7

ΚΕΦΑΛΑΙΟ 2

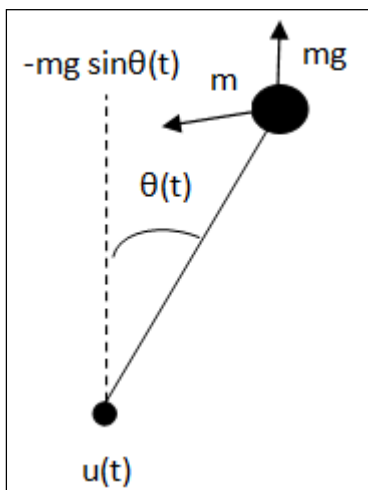
Ανάστροφο εκκρεμές

2.1 Θεωρία ανάστροφου εκκρεμής

Η αρχή της θεωρίας στην οποία βασίζεται η εργασία μας είναι το ανεστραμμένο ή ανάστροφο εκκρεμές. Το ανάστροφο εκκρεμές είναι ένα εκκρεμές το οποίο έχει το κέντρο της μάζας πάνω στο σημείο περιστροφής του. Χωρίς πρόσθετη βοήθεια δεν είναι ικανό να ισορροπήσει γι' αυτό είναι απαραίτητη η χρήση ενός συστήματος ελέγχου το οποίο θα του δίνει τη δυνατότητα να διατηρεί σταθερή τη γωνία του και να μετακινεί το σημείο περιστροφής γύρω από το κέντρο μάζας του, όταν εκείνο χάνει την ισορροπία του. Αποτελεί ένα σταθερό πρόβλημα στη θεωρία της δυναμικής και του ελέγχου και χρησιμοποιείται ως σημείο αναφοράς για τη δοκιμή στρατιωτικών ελέγχων. Συχνά εφαρμόζεται με το σημείο περιστροφής τοποθετημένο σε ένα αντικείμενο που μπορεί να κινείται οριζόντια υπό τον έλεγχο ενός σερβοσυστήματος. Οι περισσότερες εφαρμογές περιορίζουν το εκκρεμές σε έναν βαθμό ελευθερίας, τοποθετώντας τον πόλο σε έναν άξονα περιστροφής. Η διαφορά με το σταθερό εκκρεμές είναι ότι το σταθερό κρέμεται προς τα κάτω, ενώ ένα ανεστραμμένο εκκρεμές είναι εγγενώς ασταθές και πρέπει να είναι ενεργά ισορροπημένο ώστε να παραμένει όρθιο. Αυτό μπορεί να γίνει εφαρμόζοντας ροπή στρέψης στο σημείο περιστροφής οριζόντια ως μέρος ενός συστήματος ανάδρασης, αλλάζοντας τον ρυθμό περιστροφής της μάζας.

2.2 Μαθηματική ανάλυση του ανάστροφου εκκρεμούς.

Γεωμετρική απεικόνιση του ανάστροφου εκκρεμούς:



Η μη γραμμική διαφορική εξίσωση που ορίζει το εκκρεμές είναι η εξής:

$$mL^2\ddot{\theta}(t) + mgL\sin\theta(t) = u(t)$$

Εικόνα 8

2.2.1 Τυχαία ασταθή κατάσταση

Παρακάτω θα δούμε ένα παράδειγμα για μια τυχαία ασταθή κατάσταση, η οποία έστω ότι είναι η $\theta(t)=\pi$, $\dot{\theta}(t)=0$

Θεωρούμε πως η κατάσταση διαταράσσεται με αποτέλεσμα να ισχύει η εξίσωση $\theta(t)=\pi +\varphi(t)$, όπου η γωνία $\varphi(t)=\theta-\pi$, να είναι μικρή, δηλαδή κάτω από 90° ώστε να ισχύει ότι:

$$\sin\theta(t)=\sin(\pi+\varphi(t))=-\sin(\varphi(t))=-\varphi(t) \text{ και } \dot{\theta}(t) = \dot{\varphi}(t), \ddot{\theta}(t)=\ddot{\varphi}(t)$$

Άρα η γωνία $\varphi(t)$ ικανοποιεί τη διαφορική εξίσωση:

$$\ddot{\varphi}(t) - \frac{g}{L}\varphi(t) = \frac{1}{mL^2} u(t) \quad (1)$$

Από τον μετασχηματισμό Laplace, έχουμε ότι:

$$L\{\varphi(t)\} = \Phi(s) \text{ και } L\{u(t)\} = U(s)$$

Επομένως:

$$s^2\Phi(s) - s\varphi(0) - \dot{\varphi}(0) - \frac{g}{L}\Phi(s) = \frac{1}{mL^2}U(s)$$

$$\Phi(s) = \frac{s\varphi(0) + \dot{\varphi}(0)}{s^2 - \frac{g}{L}} + \frac{1}{(s^2 - \frac{g}{L}) mL^2} U(s)$$

Αν $\frac{g}{L}=1$, $mL^2=1$, $u(t)=0$ και $t \geq 0$ τότε,

$$\Phi(s) = \frac{s\varphi(0) + \dot{\varphi}(0)}{s^2 + 1} = \frac{A}{s-1} + \frac{B}{s+1}$$

Όπου:

$$A = \frac{\varphi(0) + \dot{\varphi}(0)}{2}$$

$$B = \frac{-\varphi(0) + \dot{\varphi}(0)}{-2}$$

Άρα έχουμε:

$$\Phi(t) = L^{-1}\{\Phi(s)\}$$

$$= \frac{\varphi(0) + \dot{\varphi}(0)}{2} = L^{-1}\left\{\frac{1}{s-1}\right\} + \frac{-\varphi(0) + \dot{\varphi}(0)}{-2} L^{-1}\left\{\frac{1}{s+1}\right\}$$

$$= \frac{\varphi(0) + \dot{\varphi}(0)}{2} = e^t + \frac{-\varphi(0) + \dot{\varphi}(0)}{-2} e^{-t}$$

Λόγω του όρου e^t , όταν :

$t \rightarrow \infty$ τότε $\varphi(t) \rightarrow \infty$

Άρα χωρίς είσοδο(ροπή στρέψης) ($u(t)=0, t \geq 0$), η κατάσταση ισορροπίας είναι:

$$\theta(t)=\pi, \dot{\theta}(t)=0, t \geq 0$$

ή ισοδύναμα η κατάσταση ισορροπίας :

$$\varphi(t)=0, \dot{\varphi}(t)=0, t \geq 0, \text{ είναι ασταθής.}$$

Στόχος μας είναι η επιλογή της εισόδου $u(t)=0, t \geq 0$ έτσι ώστε για κάθε μικρή διαταραχή:

$$\varphi(0) \neq 0, \dot{\varphi}(t) \neq 0$$

η κατάσταση ισορροπίας

$$\varphi(t)=0, \dot{\varphi}(t) = 0, t \geq 0$$

να είναι τέτοια ώστε

$$(2) \lim_{t \rightarrow \infty} \varphi(t)=0, \lim_{t \rightarrow \infty} \dot{\varphi}(t) =0$$

2.2.2. Λύση χωρίς ανάδραση

Έστω ότι γνωρίζουμε ακριβώς την αρχική κατάσταση και έστω ότι είναι:

$$(3) \varphi(0) = 1, \dot{\varphi}(0) = -2$$

Αν επιλέξουμε είσοδο χωρίς ανάδραση ή open loop control:

$$(4) u(t)=3e^{-2t}$$

Η (1) εξίσωση θα γράφεται :

$$(5) \ddot{\varphi}(t) - \varphi(t) = 3e^{-2t}$$

της οποίας η λύση για αρχικές συνθήκες τις $\varphi(0)=1, \dot{\varphi}(t) = -2$ είναι η:

$$\varphi(t) = 3e^{-2t} \Rightarrow \dot{\varphi}(t) = -2e^{-2t}$$

έτσι ώστε η (2) να ικανοποιείται:

$$\lim_{t \rightarrow \infty} \varphi(t)=0, \lim_{t \rightarrow \infty} \dot{\varphi}(t) =0$$

Αν όμως υπάρχει σφάλμα στον προσδιορισμό των αρχικών συνθηκών και οι αρχικές συνθήκες δεν είναι οι (3) αλλά οι

$$\varphi(0)=1, \dot{\varphi}(t) = -2 + \varepsilon, \varepsilon \neq 0$$

τότε η λύση της (5) είναι η

$$\varphi(t) = \frac{1}{2} e^t \varepsilon + e^{-2t} - \frac{1}{2} e^{-t} \varepsilon$$

η οποία, λόγω του όρου $\frac{1}{2} e^t \varepsilon$, ικανοποιεί τις

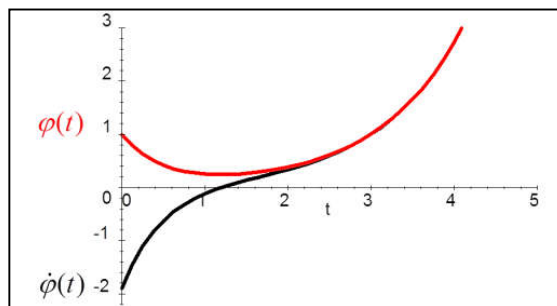
$$\lim_{t \rightarrow \infty} \varphi(t) \rightarrow \infty, \lim_{t \rightarrow \infty} \dot{\varphi}(t) \rightarrow \infty$$

Αν για παράδειγμα $\varepsilon = \frac{1}{10}$

$$(6) \varphi(t) = \frac{1}{20} e^t + e^{-2t} - \frac{1}{20} e^{-t}$$

$$(7) \dot{\varphi}(t) = \frac{1}{20} e^t - 2e^{-2t} + \frac{1}{20} e^{-t}$$

Γραφική παράσταση των $\varphi(t)$, $\dot{\varphi}(t)$ στις (6) και (7)



Βλέπουμε από την γραφική παράσταση, ότι μια προγραμματισμένη είσοδος όπως ήταν η (4), δεν μπορεί να σταθεροποιήσει το σύστημα, γιατί μια τέτοια είσοδος εξαρτάται από τις αρχικές συνθήκες $\varphi(0)$ και $\dot{\varphi}(0)$ και η ακριβής γνώση των αρχικών συνθηκών δεν είναι εφικτή.

2.2.3 Λύση με ανάδραση

Έστω ότι υπάρχει αισθητήριο όργανο μέσω του οποίου μπορούμε να προσδιορίζουμε τη γωνία περιστροφής $\varphi(t)$ του άξονα του ανάστροφου εκκρεμούς γύρω από την κατακόρυφο, τη γωνιακή ταχύτητα $\dot{\varphi}(t)$ και έστω ότι επιλέγουμε την είσοδο $u(t)$ (ροπή στρέψης του άξονα) έτσι ώστε να είναι ανάλογη της γωνίας $\varphi(t)$ και της γωνιακής ταχύτητας $\dot{\varphi}(t)$, δηλαδή:

$$u(t) = -a\varphi(t) \quad (8), \text{ όπου } a \text{ επιλέξιμη σταθερά έτσι ώστε αν } \varphi(t) = \theta(t) - \pi > 0 \text{ ή } \theta(t) > \pi$$

η φορά στρέψης να είναι σύμφωνα με την κίνηση των δεικτών του ρολογιού και αν

$\varphi(t) = \theta(t) - \pi < 0$ ή $\theta(t) < \pi$ η φορά στρέψης να είναι αντίθετη από την κίνηση των δεικτών του ρολογιού.

Αντικαθιστώντας την (8) στην (1), παίρνουμε τη διαφορική εξίσωση που διέπει το κλειστό σύστημα $\ddot{\varphi}(t) - \varphi(t) = -a\varphi(t)$ ή ισοδύναμα τη διαφορική εξίσωση

$$\ddot{\varphi}(t) + (a - 1)\varphi(t) = 0 \quad (9)$$

Το χαρακτηριστικό πολυώνυμο της (9) είναι το $s^2 + a - 1 = 0$ (10)

Αν $a - 1 > 0$, οι λύσεις της (10) είναι καθαρά μιγαδικές και συζυγείς:

$$\lambda_1 = j\sqrt{a-1}, \lambda_2 = -j\sqrt{a-1}$$

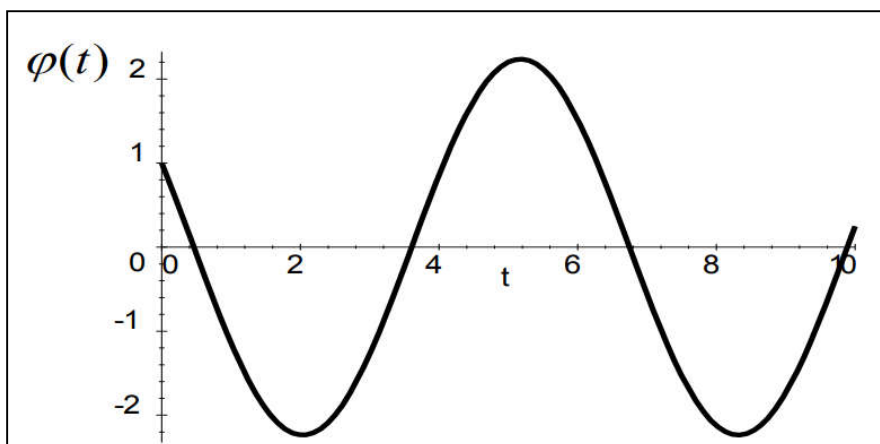
Επομένως η λύση της (9) είναι η

$$\varphi(t) = \varphi(0) \cos \sqrt{a-1}t + \frac{\dot{\varphi}(0)}{\sqrt{a-1}} \sin \sqrt{a-1}t$$

Έχουμε δηλαδή ταλάντωση ανάστροφου εκκρεμούς γύρω από την κατάσταση ισορροπίας $\varphi(0)=0$, $\dot{\varphi}(0) = 0$

Αν επιλέξουμε ότι έστω $a=2$, τότε η εξίσωση γίνεται:

$\varphi(t) = \varphi(0)\cos t + \dot{\varphi}(0)\sin t$, της οποίας η γραφική παράσταση διαμορφώνεται ως εξής για τις τιμές $\varphi(0)=1$, $\dot{\varphi}(0) = -2$



Αν $\alpha-1 < 0$, οι ρίζες του χαρακτηριστικού πολυωνύμου είναι πραγματικές και είναι οι:

$$\lambda_1 = \sqrt{1-\alpha} > 0, \lambda_2 = -\sqrt{1-\alpha} < 0$$

και η λύση της διαφορικής εξίσωσης (9) διαμορφώνεται ως εξής:

$$\varphi(t) = c_1 e^{(\sqrt{1-\alpha})t} + c_2 e^{-(\sqrt{1-\alpha})t}$$

για την οποία, λόγω του όρου $c_1 e^{(\sqrt{1-\alpha})t}$ έχουμε ότι

$$\lim_{t \rightarrow \infty} \varphi(t) = 0, \quad \lim_{t \rightarrow \infty} \dot{\varphi}(t) = 0$$

Επομένως η ανάδραση εξόδου δεν σταθεροποιεί το σύστημα.

Κεφάλαιο 3

Γυροσκόπιο

3.1 Τι είναι το γυροσκόπιο;

Το γυροσκόπιο είναι μια συσκευή η οποία μπορεί να διατηρεί σταθερό τον προσανατολισμό της μέσω της περιστροφής των μερών της και της αρχής της διατήρησης της στροφορμής. Εφευρέθηκε από τον Ζαν Μπερνάρ Λεόν Φουκώ το 1852, ο οποίος προσπάθησε με αυτή να αποδείξει την περιστροφή της Γης. Πρόκειται για μια διάταξη όμοια με εκείνη που φέρει η σχολική υδρόγειος σφαίρα. Αντί όμως της υδρογείου φέρεται μια μεταλλική στεφάνη που μπορεί να περιστρέφεται δεξιά ή αριστερά. Αυτή η στεφάνη φέρει δεύτερη εσωτερική που στηρίζεται με συνδέσμους σε οριζόντια διάταξη, ως προς την εξωτερική, δυνάμενη έτσι να περιστρέφεται ελεύθερα με φορά πάνω ή κάτω. Στην εσωτερική αυτή στεφάνη συγκρατείται εσωτερικά σε κάθετη διάταξη σε σχέση με την προηγούμενη ο "σφόνδυλος" που αποτελεί μια μικρή σφαίρα που περιστρέφεται υπό μορφή σβούρας. Τα σημεία έδρασης της κάθε στεφάνης καθώς και του σφονδύλου εξασφαλίζουν την ελεύθερη περιστροφή όλων των τμημάτων της διάταξης, δηλαδή του γυροσκοπίου ως σύνολο.



Εικόνα 9

3.1.1 Που χρησιμοποιείται το γυροσκόπιο

Εφαρμογή του γυροσκοπίου αποτελεί η λεγόμενη "γυροσκοπική πυξίδα" της οποίας οι ενδείξεις, μετά από κάποιο χρόνο αφού τεθεί σε λειτουργία, θεωρούνται αληθείς με συνέπεια να μη χρήζουν διορθώσεων, όπως συμβαίνει στη μαγνητική πυξίδα. Άλλη σημαντική εφαρμογή είναι η χρήση του γυροσκοπίου για τη διατήρηση και αλλαγή της πορείας των πυραύλων και η χρήση του σε συστήματα αδρανειακής πλοήγησης σε αεροσκάφη και πυραύλους. Η διαστημική αποστολή Gravity B της NASA του 2005 χρησιμοποίησε τέσσερα γυροσκόπια με τις πιο τέλειες σφαίρες που έχουν κατασκευαστεί ποτέ, προκειμένου να μετρήσει τη δημιουργία δινών στο χωρόχρονο από το βαρυτικό πεδίο της Γης, επαληθεύοντας την ισχύ της Γενικής Σχετικότητας.

3.1.2 Πως λειτουργεί το γυροσκόπιο

Τα γυροσκόπια λειτουργούν με βάση δύο αρχές:

A. Σύμφωνα με την αρχή της διατήρησης της στροφορμής. Μια περιστρεφόμενη μάζα διατηρεί σταθερή την κατεύθυνση του άξονα περιστροφής. Ένα περιστρεφόμενο γυροσκόπιο κάνει ένα δορυφόρο να είναι σταθερός και πάντοτε στραμμένος προς τη Γη. Μια περιστρεφόμενη μάζα επίσης αντιστέκεται στις δυνάμεις που προσπαθούν να αλλάξουν την κατεύθυνση του άξονα περιστροφής. Στα μέσα του περασμένου αιώνα, τεράστια γυροσκόπια, που ζύγιζαν τόνους, βιδώνονταν στην άτρακτο των πλοίων και περιστρέφονταν από κινητήρες ώστε να «πιέζουν αντίθετα» όταν τα κύματα έτειναν να ανατρέψουν τα σκάφη.



Εικόνα 10

B. Σύμφωνα με την αρχή της μετάπτωσης, Ένα περιστρεφόμενο γυροσκόπιο με ελευθερία μετακίνησης κινείται κάθετα ως προς τις εξωτερικές δυνάμεις που του ασκούνται. Μια παιδική σβούρα δεν ανατρέπεται όταν τη σπρώξουμε ελαφρά, αλλά μετακινείται στο πλάι. Ένα γυροσκόπιο σε αεροπλάνο που πετάει παράλληλα με το έδαφος θα παρουσιάσει μετάπτωση όταν τα φτερά αρχίσουν να παίρνουν κλίση. Ανιχνεύοντάς την τα όργανα δείχνουν στον πιλότο τη γωνία που έχουν τα φτερά ως προς τον ορίζοντα, στοιχείο καθοριστικής σημασίας ιδίως όταν η πτήση γίνεται «στα τυφλά». Ένα δεύτερο γυροσκόπιο μπορεί να δείξει την κλίση της ατράκτου του αεροπλάνου προς τα μπρος ή προς τα πίσω. Συνδεδεμένα με επιταχυνσιόμετρα τα γυροσκόπια μπορούν να πετάξουν σταθερά το αεροπλάνο, λειτουργία γνωστή ως «αυτόματος πιλότος».

3.2 Τι είναι το επιταχυνσιόμετρο

Το επιταχυνσιόμετρο αποτελεί μια συσκευή η οποία είναι ικανή να μετράει δυνάμεις επιτάχυνσης, όπως για παράδειγμα δυνάμεις επιτάχυνσης της βαρύτητας, στατικές ή οποιοσδήποτε δυνάμεις προέρχονται από αλλαγές στην ταχύτητα ή την κίνηση.

3.2.1 Που χρησιμοποιείται το επιταχυνσιόμετρο

Χρησιμοποιείται κυρίως σε συστήματα INS (inertial navigation system) δηλαδή αδρανειακά συστήματα πλοήγησης. Σε συνδυασμό με το γυροσκόπιο και άλλα ηλεκτρονικά κυκλώματα το επιταχυνσιόμετρο μπορεί να υπολογίζει τη θέση ενός οχήματος στον χώρο καθώς και τη στάση και την ταχύτητα αυτού με πολύ μεγάλη ακρίβεια. Η ικανότητα αυτή χρησιμεύει συνήθως σε λειτουργίες αυτόματου πιλότου, στη ναυσιπλοΐα αεροπλάνων, πλοίων και υποβρυχίων. Επιπλέον αποτελεί σημαντικό εργαλείο για τη μέτρηση και καταγραφή των επιταχύνσεων στους τρεις άξονες των αεροσκαφών από τον καταγραφέα των στοιχείων της πτήσης, για την ενεργοποίηση του αερόσακου στα αυτοκίνητα και σε οποιαδήποτε άλλη εφαρμογή απαιτείται μέτρηση της επιτάχυνσης.

3.2.2 Πως λειτουργεί το επιταχυνσιόμετρο

Υπάρχουν πολλοί τρόποι οι οποίοι ορίζουν πως λειτουργεί ένα επιταχυνσιόμετρο. Ένας τρόπος είναι η αξιοποίηση του πιεζοηλεκτρικού φαινομένου, χρησιμοποιείται δηλαδή πιεζοκρύσταλλο το οποίο πιέζεται από μάζα ανάλογη της επιτάχυνσης που δέχεται αυτή και παράγει τάση λόγω πιεζοηλεκτρικού φαινομένου ανάλογη της επιτάχυνσης. Ο δεύτερος τρόπος είναι η μέτρηση αλλαγών στην χωρητικότητα ενός πυκνωτή. Ο πυκνωτής αποτελεί μέρος της ανάρτησης μάζας και η τιμή του μεταβάλλεται με την κίνηση αυτής της μάζας. Η μεταβολής της χωρητικότητας του πυκνωτή παρέχει μέτρηση της επιτάχυνσης.

3.2.3 Ανάλυση τρόπου λειτουργίας

Παρακάτω παρουσιάζεται ο τρόπος λειτουργίας του επιταχυνσιομέτρου. Σε αυτή την περίπτωση το επιταχυνσιόμετρο λειτουργεί κάνοντας χρήση της διάταξης πυκνωτή και μάζας αδράνειας και είναι μέρος του συστήματος αδρανειακής πλοήγησης ή αδρανειακού συστήματος αναφοράς (IRS) επιβατικών αεροσκαφών. Υπάρχει αναρτημένη μάζα ανάμεσα στους πόλους δύο μαγνητών (άνω κάτω). Γύρω από την μεταλλική μάζα υπάρχει τύλιγμα πηνίου. Το έλασμα ανάρτησης της μεταλλικής μάζας σε συνδυασμό με δυο αγωγίμες πλάκες άνω και κάτω, σχηματίζει πυκνωτές.

Επιτάχυνση κατά τον διαμήκη άξονα της μάζας, μετακινεί τη μάζα υπερνικώντας την αντίσταση του ελατηρίου της ανάρτησης, μετακινώντας τη από το σημείο μηδέν η σημείο ισορροπίας. Η μετακίνηση αυτή προκαλεί αλλαγή στις χωρητικότητες των δυο πυκνωτών που στο σημείο ισορροπίας της μάζας είναι ίσες. Ο ενισχυτής σήματος αντιλαμβάνεται την διαφοροποίηση αυτή σαν σφάλμα και στέλνει κατάλληλο ηλεκτρικό ρεύμα στο πηνίο που περιβάλλει τη μάζα με σκοπό να την επαναφέρει στο σημείο ισορροπίας και να ακυρώσει το σήμα σφάλματος. Η τιμή του ρεύματος που στέλνεται στο πηνίο για να ακυρώσει το σήμα σφάλματος και να επαναφέρει τη μάζα στο σημείο ισορροπίας είναι η αναλογική έξοδος της συσκευής που αντιστοιχεί στην επιτάχυνση. Το αναλογικό αυτό σήμα ολοκληρώνεται συναρτήσει του χρόνου, μια φορά για να δώσει ταχύτητα και μια δεύτερη φορά για να μας δώσει απόσταση. Ένας αισθητήρας θερμοκρασίας είναι εγκατεστημένος μαζί με το επιταχυνσιόμετρο με σκοπό την βελτίωση της ακρίβειας του επιταχυνσιόμετρου.

Το σήμα από τον αισθητήρα θερμοκρασίας χρησιμοποιείται για την διόρθωση του σφάλματος θερμοκρασίας του επιταχυνσιόμετρου.

Κεφάλαιο 4

Ανάλυση MPU 6050

Για την εξισορρόπηση του οχήματος χρησιμοποιήθηκε το MPU 6050. το οποίο αποτελείται από 3 άξονες γυροσκοπίου και 3 άξονες επιταχυνσιομέτρου. Η πλήρης ονομασία του είναι MPU 6060- GY21.

4.1 Περιγραφή MPU 6050

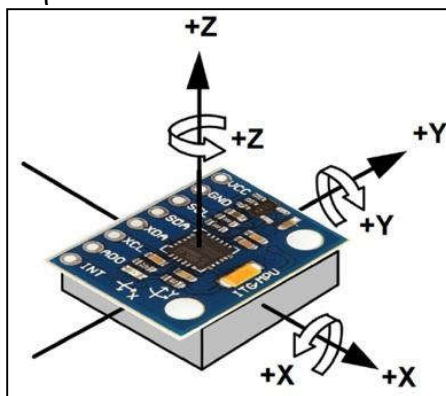
Το MPU 6050 αποτελεί την πρώτη μονάδα επεξεργασίας κινήσεων στον κόσμο με ενσωματωμένους 9 άξονες συγχρονισμού αισθητήρων, χρησιμοποιώντας τον κινητήρα “MotionFusion™”, ο οποίος μπορεί να χρησιμοποιηθεί σε εφαρμογές παιχνιδιών, τηλεχειριστήρια και διάφορες συσκευές εμπορίου. Το MPU 6050 διαθέτει ενσωματωμένο γυροσκόπιο MEMS με 3 άξονες, επιταχυνσιόμετρο MEMS με 3 άξονες και ψηφιακή κίνηση με επιταχυντή υλικού Processor με βοηθητική θύρα I2C. Το MPU 6050 συνδυάζει πληροφορίες επιτάχυνσης και περιστροφικής κίνησης σε μια ενιαία ροή δεδομένων και έχει σχεδιαστεί έτσι ώστε να συνδέεται με πολλαπλούς μη αδρανειακούς ψηφιακούς αισθητήρες πίεσης και αποτελούν επεξεργαστές δεύτερης γενιάς.

Το MPU 6050 διαθέτει τρεις αναλογικούς μετατροπείς 16 ψηφίων (ADC) για ψηφιοποίηση των εξόδων γυροσκοπίου και τρεις ADC 16 ψηφίων για ψηφιοποίηση των εξόδων επιταχυνσιομέτρου. Για την ακριβή παρακολούθηση των κινήσεων, το επιταχυνσιόμετρο και το γυροσκόπιο διαθέτει εμβέλεια που ορίζεται στις τιμές + 250 500 1000 2000 ° /s ενώ του επιταχυνσιομέτρου στις αντίστοιχες τιμές $\pm 2 \pm 4 \pm 8 \pm 16$ g. Στο MPU 6050 υπάρχει ενσωματωμένο ένα buffer FIFO 1023 byte, το οποίο βοηθάει στη μείωση της κατανάλωσης ενέργειας του συστήματος επιτρέποντας στον επεξεργαστή του συστήματος να διαβάζει τα δεδομένα του αισθητήρα και στη συνέχεια να ενσωματώνει μια λειτουργία χαμηλής κατανάλωσης όσο το MPU θα συλλέγει περισσότερα δεδομένα.

Με όλα τα απαραίτητα στοιχεία επεξεργασίας και τα κατάλληλα εξαρτήματα, το MPU 6050 είναι ικανό να υποστηρίξει μεγάλο εύρος εφαρμογών μειώνοντας έτσι την κατανάλωση σε φορητές εφαρμογές που δεν απαιτούν μεγάλες απαιτήσεις επεξεργασίας. Η επικοινωνία με όλους τους καταχωρητές της συσκευής πραγματοποιείται χρησιμοποιώντας το I2C στα 400kHz. Τα πρόσθετα χαρακτηριστικά περιλαμβάνουν ενσωματωμένο αισθητήρα θερμοκρασίας και έναν ταλαντωτή με εύρος $\pm 1\%$.

Το μέγεθος του MPU 6050 είναι 4x4x0,9 mm (QFN), παρέχει υψηλό ποσοστό απόδοσης, χαμηλό ποσοστό θορύβου αλλά και χαμηλό κόστος κατασκευής. Επιπλέον η πλακέτα διαθέτει αντοχή έως 10.000g.

Το MPU 6050 λειτουργεί υπό τροφοδοσία τάσης VDD με εύρος 2.375V – 3.46V και διαθέτει και έναν ακροδέκτη αναφοράς VLOGIC (εκτός από την αναλογική πηγή τροφοδοσίας VDD) ο οποίος καθορίζει τα λογικά επίπεδα του I2C. Η τάση VLOGIC μπορεί να είναι $1,8V \pm 5\%$ ή η τάση VDD.



Εικόνα 11

Παρακάτω παρουσιάζεται ο πίνακας με τα χαρακτηριστικά του MPU 6050:

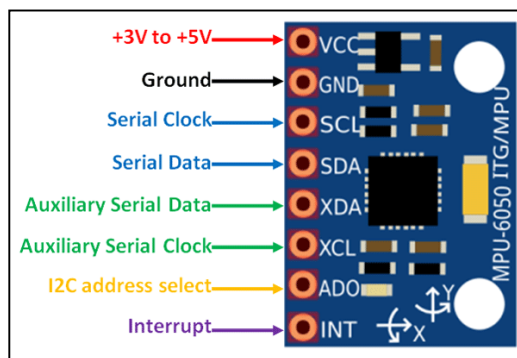
Part/ Item	MPU 6050
VDD	2.375V - 3.46V
VLOGIC	1/71V to VDD
Serial interfacessupported	I ² C
Pin 8	VLOGIC
Pin 9	AD0
Pin 23	SCL
Pin 24	SDA

Αναλυτικά τα PINSτου MPU 6050:

Αριθμός PIN	Όνομα PIN	Περιγραφή
1	VCC	Τροφοδοσία του αισθητήρα
2	GND	Συνδέεται με την γείωση του κυκλώματος
3	Serial Clock (SCL)	Χρησιμοποιείται για clockpulseγια επικοινωνία με I ² C
4	Serial Data (SDA)	Χρησιμοποιείται για μεταφορά δεδομένων μέσω I ² C
5	Auxiliary Serial Data (XDA)	Μπορεί να χρησιμοποιηθεί για να επικοινωνήσει με

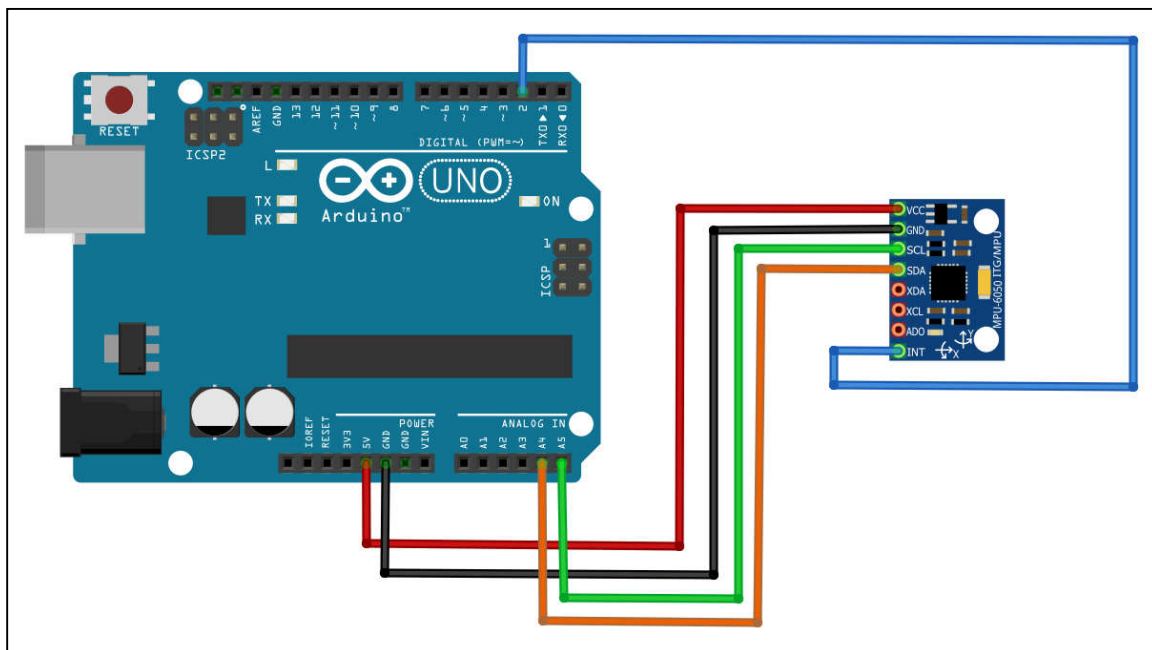
		άλλες συσκευές που έχουν το πρωτόκολλο I ² C
6	Auxiliary Serial Clock (XCL)	Μπορεί να χρησιμοποιηθεί για να επικοινωνήσει με άλλες συσκευές που έχουν το πρωτόκολλο I ² C
7	AD0	Προσδιορισμός της διεύθυνσης αν χρησιμοποιούνται και άλλα MPU 6050
8	Interrupt (INT)	ΠINγια ένδειξη ότι τα δεδομένα είναι διαθέσιμα προς ανάγνωση

Απεικόνιση των pin του MPU 6050:



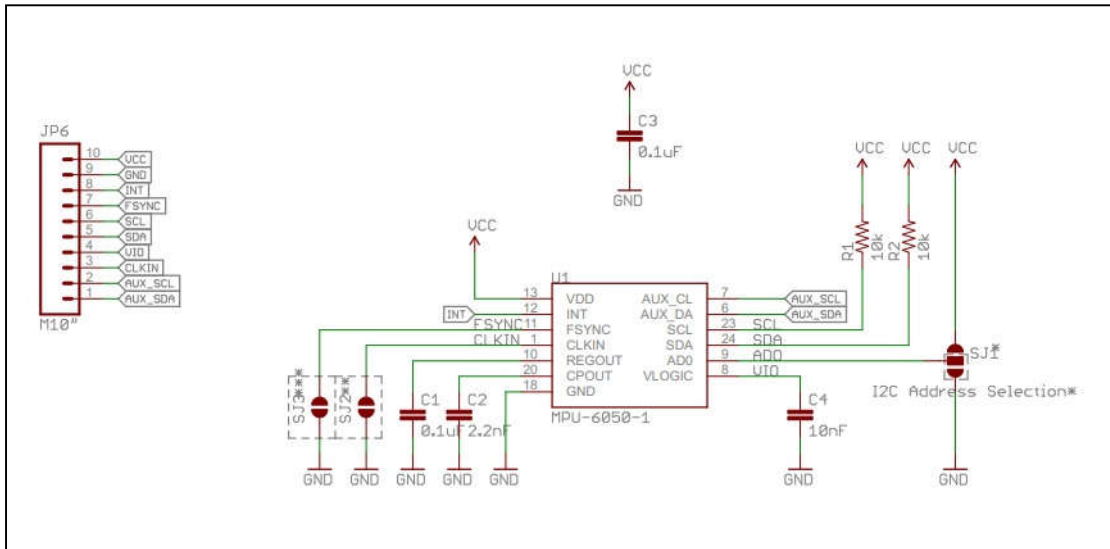
Εικόνα 12

Συνδεσμολογία του MPU 6050:



Εικόνα 13

Σχηματικό διάγραμμα του MPU 6050:



Εικόνα 14

4.2 Εφαρμογές του MPU 6050

- Τεχνολογία Blur Free™, για σταθεροποίηση βίντεο και εικόνας
- Τεχνολογία Air Sign™, για ασφάλεια και έλεγχο ταυτότητας
- Τεχνολογία Touch Anywhere™, για έλεγχο/ πλοήγηση εφαρμογών UI ‘‘χωρίς επαφή’’
- Τεχνολογία Motion Command™, για κινήσεις Short cuts
- Παιχνίδια και εφαρμογές με δυνατότητα κίνησης
- Αναγνώριση κίνησης Instant Gesture™iG™
- Υπηρεσίες που βασίζονται στην τοποθεσία
- Ακουστικά και φορητά παιχνίδια
- Ελεγκτές παιχνιδιών με κίνηση
- Τρισδιάστατα τηλεχειριστήρια για συνδεδεμένα στο διαδίκτυο δορυφορικά τηλεοπτικά κανάλια και αποκωδικοποιητές
- Ευέλικτοι αισθητήρες που να ασχολούνται με υγεία, γυμναστήριο κ.ά.

4.3 Επιπλέον χαρακτηριστικά του MPU 6050

- Εννιά (9) άξονες Motion Fusion από τον επεξεργαστή ψηφιακών κινήσεων (DMP, DigitalMotionProcessor).
- Βοηθητικό I2C για την αναγνώριση δεδομένων από τους εξωτερικούς αισθητήρες

- Ρεύμα λειτουργίας 3.9mA, όταν είναι ενεργοποιημένοι και οι 6 άξονες ανίχνευσης κίνησης και το DMP
- Εύρος τάσης τροφοδοσίας VDD 2.375V- 3.46V
- Η ευέλικτη τάση τροφοδοσίας υποστηρίζει πολλαπλά I2C
- Λεπτή και μικρή συσκευή QFN με διαστάσεις 4x4x0,9mm
- Ελάχιστη ευαισθησία σε εγκάρσιο άξονα μεταξύ των αξόνων επιταχυνσιομέτρου και γυροσκοπίου
- 1024 byte FIFO buffer μειώνει την κατανάλωση ρεύματος επιτρέποντας στον κεντρικό επεξεργαστή να διαβάζει τα δεδομένα και να μεταβαίνει σε λειτουργία χαμηλής κατανάλωσης καθώς το MPU 6050 συλλέγει περισσότερα δεδομένα
- Αισθητήρας θερμοκρασίας ψηφιακής εξόδου
- Ψηφιακά φίλτρα προγραμματιζόμενα από τον χρήστη για το γυροσκόπιο και το επιταχυνσιόμετρο
- 10.000g αντοχή σε κραδασμούς
- 400kHz fast mode I2C για επικοινωνία με όλους τους καταχωρητές

4.4 MotionProcessing

- Η εσωτερική μηχανή DMP (Digital Motion Processing) υποστηρίζει 3D κινήσεις αλγορίθμων
- Το MPU 6050 συλλέγει δεδομένα γυροσκοπίου και επιταχυνσιομέτρου, ενώ συγχρονίζει τη δειγματοληψία δεδομένων σε καθορισμένο από τον χρήστη ρυθμό. Το σύνολο των δεδομένων που λαμβάνεται από το MPU 6050 περιλαμβάνει στοιχεία από τους 3 άξονες γυροσκοπίου και τους 3 άξονες επιταχυνσιομέτρου καθώς και δεδομένα θερμοκρασίας. Ο επεξεργαστής μπορεί επιπλέον να λάβει δεδομένα από ένα ψηφιακό μαγνητόμετρο.
- Το FIFO αναστέλλει το πλήρες σύνολο δεδομένων, μειώνοντας τις απαιτήσεις χρονισμού στον επεξεργαστή του συστήματος, έτσι ο επεξεργαστής μπορεί να εξοικονομήσει ενέργεια εισάγοντας μια κατάσταση χαμηλής κατανάλωσης κατά τη διάρκεια λειτουργίας.
- Το πρόγραμμα μπορεί να υποστηρίζει χαρακτηριστικά όπως η αναγνώριση κίνησης, η μετατόπιση, η μεγέθυνση, η κύλιση και η ανίχνευση μηδενικής κίνησης (ακινησία)
- Η λειτουργία χαμηλής κατανάλωσης «βηματόμετρο» επιτρέπει στον επεξεργαστή να είναι σε αναμονή (ή sleep mode) ενώ το DMP να μετρά βήματα.

4.5 Γυροσκόπιο στο MPU 6050

Το γυροσκόπιο MEMS τριπλού άξονα στο MPU 6050 περιλαμβάνει τα εξής χαρακτηριστικά:

- Ψηφιακοί αισθητήρες γωνιακού ρυθμού αξόνων X, Y και Z με πλήρες φάσμα το οποίο μπορεί να προγραμματιστεί από τον χρήστη και εμβέλεια ± 250 , ± 500 , ± 1000 και ± 2000 ° / δευτερόλεπτο.
- Το εξωτερικό σήμα συγχρονισμού που είναι συνδεδεμένο με τον ακροδέκτη FSYNC υποστηρίζει συγχρονισμό εικόνα, βίντεο και GPS
- Οι ενσωματωμένοι ADC 16-bit επιτρέπουν ταυτόχρονη δειγματοληψία γυροσκοπίων
- Δυνατότητα σταθερότητας θερμοκρασίας
- Βελτιωμένη απόδοση θορύβου χαμηλής συχνότητας
- Ψηφιακό προγραμματιζόμενο φίλτρο χαμηλής διέλευσης
- Ρεύμα λειτουργίας γυροσκοπίου: 3,6Ma
- Ρεύμα αναμονής: 5μΑ
- Σταθμισμένος συντελεστής κλίμακας ευαισθησίας
- Αυτοέλεγχος συστήματος.

4.6 Επιταχυνσιόμετρο στο MPU 6050

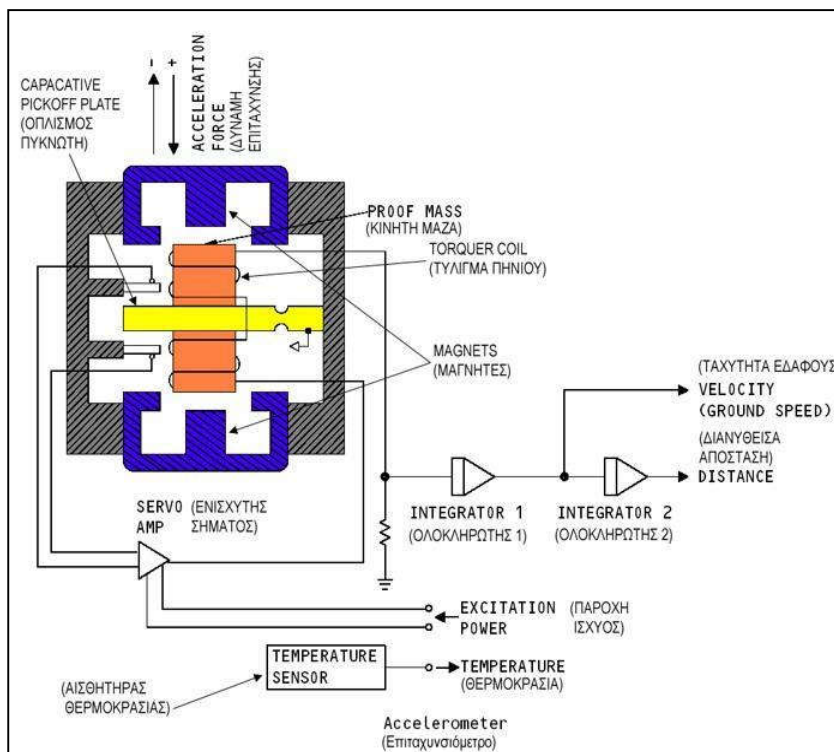
Το επιταχυνσιόμετρο MEMS με τριπλό άξονα στο MPU 6050 περιλαμβάνει τα παρακάτω χαρακτηριστικά:

- Επιταχυνσιόμετρο τριών αξόνων ψηφιακής εξόδου με προγραμματιζόμενη κλίμακα $\pm 2 \pm 4 \pm 8 \pm 16$ g.
- Οι ενσωματωμένοι ADC 16-bit επιτρέπουν την ταυτόχρονη δειγματοληψία των επιταχυνσιομέτρων χωρίς να απαιτείται εξωτερικός πολυπλέκτης.
- Κανονικό ρεύμα λειτουργίας: 500μΑ
- Ρεύμα λειτουργίας επιταχυνσιομέτρου χαμηλής ισχύος: 10μΑ στα 1,2Hz, 20μΑ στα 5Hz, 60μΑ στα 20Hz, 110μΑ στα 40Hz
- Ανίχνευση προσανατολισμού και σηματοδότησης
- Προγραμματισμένες από τον χρήστη διακοπές

- Ανίχνευση χτυπήματος
- Διακοπή ελεύθερης πτώσης
- Διακοπή High – G
- Διακοπή μηδενικής κίνησης
- Αυτοέλεγχος συστήματος
- Στο MPU 6050 υπάρχει γεννήτρια χρονισμού $\pm 1\%$ με μεταβολή συχνότητας
- Προαιρετικές εξωτερικές εισοδοι ρολογιού : 32,768kHz ή 19,2MHz

Το επιταχυνσιόμετρο MEMS αποτελεί μια ηλεκτρομηχανική συσκευή που σκοπό έχει να μετρά δυνάμεις επιτάχυνσης. Οι δυνάμεις αυτές μπορεί να είναι στατικές όπως είναι η επιτάχυνση της βαρύτητας ή δυναμικές οι οποίες να προέρχονται από αλλαγές στην ταχύτητα ή στην διεύθυνση της κίνησης, δηλαδή επιτάχυνση, επιβράδυνση και στροφές.

Εσωτερικό του MPU 6050:



Εικόνα 15

Κεφάλαιο 5

5.1 Bluetooth

Το Bluetooth είναι ένα βιομηχανικό πρότυπο για ασύρματα προσωπικά δίκτυα υπολογιστών WPAN, (Wireless Personal Area Networks). Αφορά μια ασύρματη τηλεπικοινωνιακή τεχνολογία μικρών αποστάσεων, η οποία μπορεί να μεταδώσει σήματα μέσω μικροκυμάτων σε ψηφιακές συσκευές. Επομένως το Bluetooth είναι ένα πρωτόκολλο το οποίο παρέχει ασύρματη επικοινωνία ανάμεσα σε κινητά τηλέφωνα, φορητούς υπολογιστές, προσωπικούς υπολογιστές, PDA(προσωπικός ψηφιακός βοηθός), εκτυπωτές, ψηφιακές κάμερες, ηχεία μέσω μιας ασφαλούς, φθηνής και παγκοσμίως διαθέσιμης χωρίς ειδική άδεια ραδιοσυχνότητας μικρής εμβέλειας. Από τεχνικής άποψης το Bluetooth είναι ένα πρωτόκολλο ασύρματης δικτύωσης σε φυσικό επίπεδο.

5.1.1 Εφαρμογή

Με τη χρήση του Bluetooth, καταργήσαμε την ενσύρματη επικοινωνία μεταξύ συσκευών, η οποία ήταν απαραίτητη για να υπάρχει σύνδεση. Το Bluetooth επιτρέπει επίσης την μεταφορά δεδομένων όπως είναι οι εικόνες, μουσική και επαφές από κινητό προς κινητό, χωρίς σύνθετες διαδικασίες.

Μερικές από τις εφαρμογές του bluetooth:

- Ασύρματα περιφερειακά, όπως είναι τα χειριστήρια, τα ποντίκια υπολογιστών, οι εκτυπωτές και γενικά τα πολυμηχανήματα.
- Ασύρματη μεταφορά ψηφιακών αρχείων ανάμεσα σε κινητά.
- Ασύρματα ακουστικά για κινητά τηλέφωνα.
- Ασύρματη επικοινωνία στο αυτοκίνητο, σε οχήματα που έχουν τις προδιαγραφές για σύνδεση Bluetooth.
- Απομακρυσμένος έλεγχος συσκευών.
- Σε ορισμένες ιατρικές εφαρμογές.

5.1.2 Λειτουργία Bluetooth

Το Bluetooth λειτουργεί στο φάσμα συχνότητας 2.4GHz, κάτι που το καθιστά λειτουργικό σε οποιοδήποτε μέρος σε εμβέλεια όμως εντός 10 μέτρων. Ο μέγιστος αριθμός συσκευών που μπορούν να συνδεθούν σε μια συγκεκριμένη συχνότητα είναι

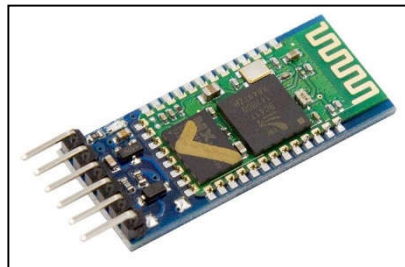
εφτά. Ένας περιορισμός που παρουσιάζεται στο Bluetooth είναι πως οι συσκευές που τα υποστηρίζουν δεν είναι δυνατό να υποστηρίζουν ταυτόχρονα πρωτόκολλα IEEE 802.11 (πρότυπα για ασύρματα δίκτυα) καθώς θα υπήρχαν σοβαρά προβλήματα παρεμβολών. Στα Bluetooth υπάρχουν διαφόρων ειδών εκδοχές, οι οποίες ονομάζονται προφίλ. Κάθε προφίλ έχει ένα διαφορετικό ρόλο, για παράδειγμα υπάρχει το Bluetooth προφίλ εκτύπωσης, προφίλ ασύρματου τηλεφώνου κ.α.

5.2 Τι είναι το Bluetooth module

Για να μπορέσουμε να παρέχουμε ένα είδος αυτονομίας στο ρομπότ, επιλέξαμε το Bluetooth module HC-05. Έτσι έχουμε τη δυνατότητα να ελέγχουμε την εκκίνηση και τη διακοπή των motor μέσω ενός κινητού Smartphone. Η εφαρμογή που χρησιμοποιήθηκε είναι η Arduino Bluetooth Controller. Ουσιαστικά, οι συσκευές Bluetooth είναι μονάδες οι οποίες έχουν ενσωματωμένο ένα πρωτόκολλο σειριακής επικοινωνίας δίνοντας έτσι τη δυνατότητα σε δύο συσκευές να επικοινωνούν.

Το HC-05 ρυθμίζεται ως master module, σε σύγκριση με το HC-06 το οποίο μπορεί να ρυθμιστεί μόνο ως slave.

Η εφαρμογή Arduino Bluetooth Controller έχει σχεδιαστεί για να στέλνει σειριακά δεδομένα στη Bluetooth module όταν πατηθεί ένα κουμπί στην εφαρμογή. Η μονάδα Bluetooth λαμβάνει τα δεδομένα μέσω του Transmitter (TX).

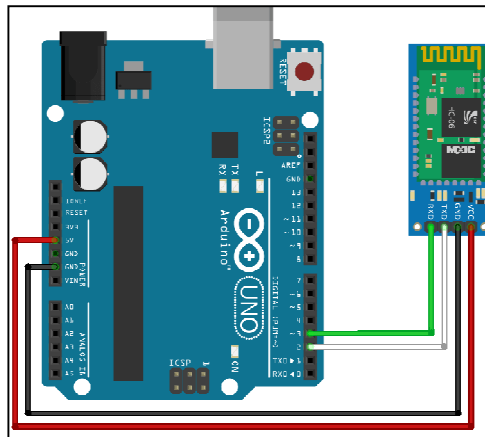


Εικόνα 16

5.2.1 Συνδεσμολογία Arduino με HC-05

Συνδέουμε τα pin για το ρεύμα και τη γείωση στις αντίστοιχες πόρτες του Arduino. Στη συνέχεια συνδέουμε τα pin του Transmitter (TXD) και του Receiver (RXD) σε δύο digital pin αντίστοιχα.

Για να μπορέσουμε να φορτώσουμε πρόγραμμα στο Arduino θα πρέπει να αποσυνδέσουμε τα pin Tx και Rx.

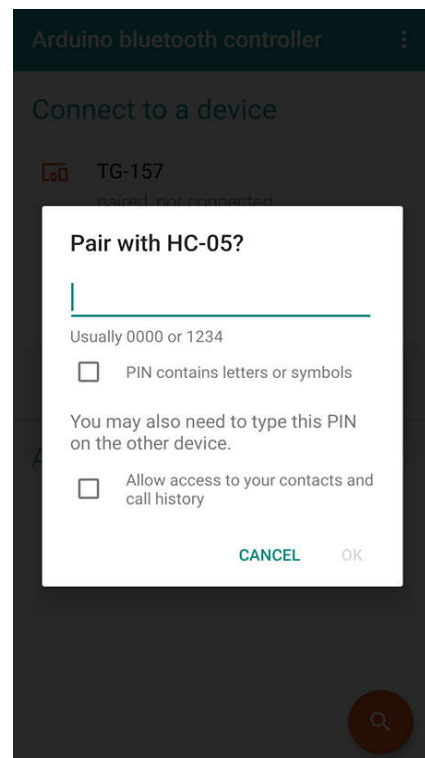


Εικόνα 17

5.3 Εφαρμογή Android

Για να μπορέσει να υπάρξει επικοινωνία του Bluetooth module με μια συσκευή Android χρησιμοποιήσαμε την εφαρμογή Arduino Bluetooth Controller. Ο τρόπος που μπορούμε να ξεκινήσουμε μια επικοινωνία με το Bluetooth module περιγράφεται παρακάτω.

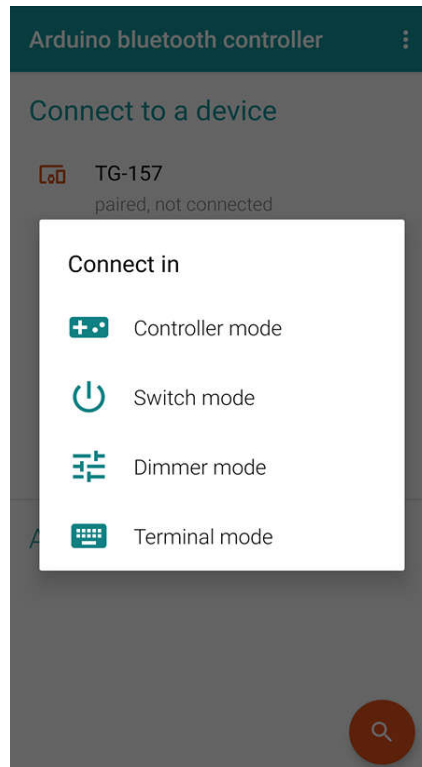
Αρχικά ανοίγουμε το Bluetooth του κινητού και την εφαρμογή για να μπορέσει το κινητό να κάνει ανίχνευση των συσκευών Bluetooth. Στη συνέχεια πληκτρολογούμε τα ψηφία που υπάρχουν πάνω στο HC-05 για να μπορέσει να γίνει σύζευξη. Μετά από αυτό, έχουμε τον έλεγχο του Bluetooth.



Οι επιλογές που μας δίνουν τέτοιου είδους

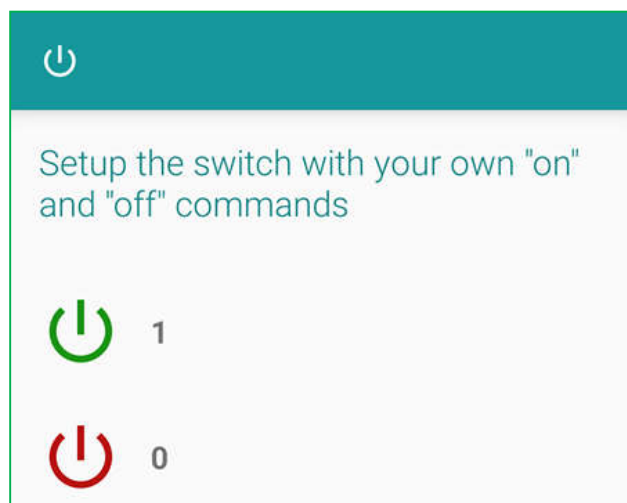
εφαρμογές είναι οι εξής:

- Controller mode
- Switch mode
- Dimmer mode
- Terminal mode



Εμείς χρησιμοποιήσαμε το switch mode γιατί μας ενδιαφέρει μόνο το on/off της εφαρμογής.

Αφού επιλέχθηκε η κατάλληλη λειτουργία, είχαμε τον έλεγχο της συσκευής με με βάση τον προγραμματισμό που είχε γίνει. Επιλέξαμε να ονομάσουμε το Power On με την κατάσταση «1» και το Power Off με την κατάσταση «0»



5.4 I2C Protocol

Ο διάυλος I2C αποτελεί τρόπο επικοινωνίας συσκευών, είτε αυτές βρίσκονται πάνω σε ένα τυπωμένο κύκλωμα, είτε σε συσκευές που συνδέονται με καλώδια. Χρησιμοποιείται για τη σύνδεση περιφερειακών μικρής ταχύτητας σε μητρικές πλακέτες, ενσωματωμένα συστήματα, κινητά τηλέφωνα ή άλλες ηλεκτρονικές συσκευές. Στη μεταφορά δεδομένων με καλώδια είναι απαραίτητο να χρησιμοποιηθούν τέσσερα καλώδια. Η γραμμή SCL είναι η γραμμή ρολογιού, η γραμμή SDA είναι η γραμμή δεδομένων. Οι γραμμές αυτές συνδέονται με τις αντίστοιχες γραμμές SDA και SCL που υπάρχουν στο κύκλωμα. Επιπλέον χρειάζονται άλλα δύο καλώδια η τάση (V_{cc} ή V_{dd}) και η γείωση (GND). Οι γραμμές SCL και SDA είναι τύπου ανοικτού συλλέκτη, δηλαδή πρέπει να συνδεθούν με μια αντίσταση στη γραμμή τροφοδοσίας η οποία ονομάζεται αντίσταση τερματισμού. Οι αντιστάσεις είναι απαραίτητες, σε διαφορετική περίπτωση οι γραμμές SCL και SDA θα είναι μονίμως σε κατάσταση λογικού 0 και ο διάυλος δε θα δουλεύει. Ο αριθμός των κόμβων, δηλαδή συσκευών που μπορούν να συνδεθούν στο διάυλο, περιορίζεται από τον αριθμό διαθέσιμων διευθύνσεων αλλά και από τη συνολική χωρητικότητα του διαύλου.

5.4.1 Λογικές στάθμες των γραμμών SCL και SDA

Σε ένα διάυλο I2C μπορεί να χρειαστεί να συνδεθούν συσκευές με διαφορετικές τάσεις λειτουργίας με αποτέλεσμα οι λογικές τιμές 0 (Low) και 1 (High) να μην είναι σταθερές αλλά να εξαρτώνται από την τάση τροφοδοσίας. Έτσι το κατώφλι του λογικού 0 ορίζεται στην τιμή του 30% της τάσης τροφοδοσίας ενώ το κατώφλι του λογικού 1 ορίζεται στην τιμή του 70% της τάσης τροφοδοσίας.

5.4.2 Το φυσικό πρωτόκολλο του διαύλου I2C

Ο διάυλος αποτελείται από Master και Slave συσκευές. Στο διάυλο υπάρχουν δύο ειδών ακολουθίες, η ακολουθία έναρξης και η ακολουθία λήξης. Όταν μια συσκευή Master επιθυμεί να επικοινωνήσει με μια Slave, ξεκινάει στέλνοντας στον διάυλο μια ακολουθία έναρξης. Οι ακολουθίες έναρξης και λήξης διαφέρουν επειδή είναι οι μοναδικές στις οποίες επιτρέπεται να αλλάζει η γραμμή δεδομένων (SDA) όσο η γραμμή

ρολογιού (SCL) είναι σε κατάσταση λογικού 1. Κατά τη διάρκεια της ακολουθίας έναρξης και λήξης ο διάυλος είναι απασχολημένος.

Τα δεδομένα μεταφέρονται σε ακολουθίες των 8 bits, τα οποία τοποθετούνται στη γραμμή SDA, ξεκινώντας από το πιο σημαντικό. Για κάθε 8 bit δεδομένων που μεταφέρονται, η συσκευή που λαμβάνει στέλνει πίσω ένα bit επιβεβαίωσης. Εάν η συσκευή που λαμβάνει τα δεδομένα, στείλει πίσω ένα low bit επιβεβαίωσης, τότε έχει λάβει όλα τα δεδομένα και είναι έτοιμη να λάβει το επόμενο byte δεδομένων. Εάν όμως στείλει πίσω ένα high bit επιβεβαίωσης, τότε δεν μπορεί να λάβει περαιτέρω δεδομένα και η master συσκευή πρέπει να τερματίσει την αποστολή δεδομένων στέλνοντας μια ακολουθία λήξης.

5.4.3 Διευθυνσιοδότηση συσκευών του I2C διαύλου

Ένας διάυλος I2C μπορεί να είναι 7 bit, που πρακτικά σημαίνει πως στο διάυλο μπορούν να συνδεθούν 128 συσκευές, 10 bit όπου μπορούν να συνδεθούν έως 1024 συσκευές, ή 16 bit που θεωρητικά μπορούν να συνδεθούν έως 65536 συσκευές.

Οι master συσκευές δεν είναι απαραίτητο να έχουν μια συγκεκριμένη διεύθυνση, εκτός και αν υπάρχουν πολλές master συσκευές στο διάυλο. Για τις slaves συσκευές είναι διαθέσιμες όλες οι υπόλοιπες διευθύνσεις, εκτός από 8 διευθύνσεις οι οποίες χρησιμοποιούνται για ειδικούς σκοπούς.

5.4.4 Εφαρμογές του διαύλου I2C

Συνηθισμένες εφαρμογές που χρησιμοποιείται ο διάυλος:

- Ανάγνωση-εγγραφή σειριακών μνημών EEROM
- Πρόσβαση σε χαμηλής ταχύτητας μετατροπείς αναλογικού σήματος σε ψηφιακό ή ψηφιακό σε αναλογικό
- Ανάγνωση αισθητήρων με σύνδεση I2C
- Ανάγνωση ρολογιών πραγματικού χρόνου
- Ανάγνωση επιτηρητών Hardware και διαγνωστικών αισθητήρων, όπως πχ θερμοστατών CPU
- Ενεργοποίηση – απενεργοποίηση τροφοδοσίας τμημάτων συστημάτων.

Κεφάλαιο 6

6.1 Συμπληρωματικό φίλτρο

Τα περισσότερα IMU (Inertial Measurement Units) έχουν έξι άξονες ελευθερίας - DOF (degrees of freedom). Πρακτικά αυτό σημαίνει πως διαθέτουν τρεις άξονες για αξελερόμετρο και 3 άξονες για γυροσκόπιο μέσα στην ίδια συσκευή.

Τόσο τα δεδομένα από το αξελερόμετρο όσο και τα δεδομένα από το γυροσκόπιο χρησιμοποιούνται για τον ίδιο σκοπό, για να ανακτήσουμε τη θέση του αντικειμένου (angular position), δηλαδή τη σωστή γωνία.

Το γυροσκόπιο μπορεί να βρίσκει τη γωνιακή θέση ενσωματώνοντας τη γωνιακή ταχύτητα στο χρόνο. Στο αξελερόμετρο όμως πρέπει να καθορίσουμε το διάνυσμα της βαρύτητας (g-force), το οποίο είναι πάντα εμφανές. Για να μπορέσουμε να το πετύχουμε αυτό χρησιμοποιούμε τη συνάρτηση atan2.

Ο κύριος λόγος που χρησιμοποιούμε φίλτρο είναι για να επιλύσουμε τα προβλήματα που παρουσιάζουν το αξελερόμετρο και το γυροσκόπιο, τα οποία είναι τα εξής:

- Πρόβλημα αξελερόμετρου: καθώς το επιταχυνσιόμετρο μετρά όλες τις δυνάμεις που ασκούνται πάνω στο αντικείμενο, συλλέγει και άλλες πληροφορίες πέρα από το διάνυσμα της βαρύτητας, με αποτέλεσμα κάθε μικρή δύναμη να αλλοιώνει τη μέτρηση. Τα αποτελέσματα του επιταχυνσιομέτρου είναι αξιόπιστα μόνο σε ένα μεγάλο χρονικό διάστημα και για το λόγο αυτό χρειάζεται ένα low pass φίλτρο.
- Πρόβλημα γυροσκοπίου: με την πάροδο του χρόνου, τα αποτελέσματα μιας μέτρησης τα οποία δεν επηρεάζονται από εξωτερικές δυνάμεις, όπως συμβαίνει στο αξελερόμετρο, έχουν την τάση να «παρασύρονται» με αποτέλεσμα να μην επιστρέφουν στην τιμή 0. Άρα τα αποτελέσματα του γυροσκοπίου είναι αξιόπιστα μόνο σε μικρό διάστημα, γιατί αλλιώς αρχίζουν να παρασύρονται.

6.2 Atan2

Στη γλώσσα τον υπολογιστών η λειτουργία της αντίστροφης εφαπτομένης παρουσιάζεται με το όνομα atan2 (x,y) ή arctan2 (x,y). Αυτή η συνάρτηση χρησιμοποιεί δύο μεταβλητές, την x και την y οι οποίες αντιστοιχούν στις συντεταγμένες ενός αυθαίρετου σημείου στο καρτεσιανό σύστημα και διαφέρουν από το μηδέν. Ο λόγος x/y αντιστοιχεί στη εφαπτομένη γωνία η οποία εκτείνεται από τον άξονα X και την ακτίνα από την αρχή σε αυτό το σημείο.

Κεφάλαιο 7

7.1 Ελεγκτής PID

Ένα από τα πιο βασικά στοιχεία στην επιστήμη του αυτόματου ελέγχου είναι ο PID ελεγκτής. Πρόκειται για ένα γενικό μηχανισμό βρόχου ανάδρασης που χρησιμοποιείται εκτεταμένα στα συστήματα αυτόματου ελέγχου. Αποτελεί έναν ελεγκτή ο οποίος προσπαθεί συνεχώς να ελαχιστοποιήσει το σφάλμα προσαρμόζοντας τις εισόδους της διαδικασίας ελέγχου. Ο PID ελεγκτής ακολουθεί έναν αλγόριθμο που αφορά τρεις ξεχωριστές παραμέτρους και γι' αυτό πολλές φορές καλείται ελεγκτής τριών όρων: οι όροι αυτοί είναι το αναλογικό (P), το ολοκληρωτικό (I) και το διαφορικό (D). Στην παρούσα εργασία εστιάζουμε στη σχεδίαση και την προσομοίωση PID ελεγκτών σε ένα πραγματικό σύστημα. Χρησιμοποιούμε και τους τρεις όρους για να πετύχουμε το επιθυμητό αποτέλεσμα. Στο κεφάλαιο αυτό θα γίνει μια εισαγωγή στο θεωρητικό κομμάτι που ακολουθήσαμε για να πετύχουμε την ισορροπία του οχήματος. Η σχεδίαση και ο προγραμματισμός του PID ελεγκτή επιλέξαμε να γίνει στον επεξεργαστή Arduino. Ωστόσο, υπάρχουν και άλλες λύσεις όπως για παράδειγμα η σχεδίαση στο πρόγραμμα Matlab.

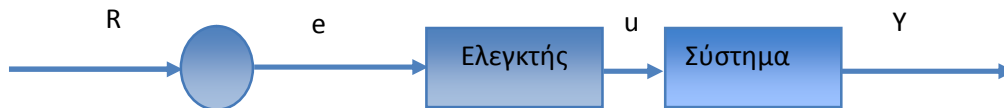
Με την έννοια ισορροπία οχήματος ή αλλιώς self-balancing εννοούμε το μηδενισμός σφάλματος, δηλαδή το μηδενισμό του παράγοντα που προκαλεί μια διαταραχή στην ομαλή λειτουργία της διαδικασίας. Πρακτικά αυτό επιτυγχάνεται μέσω του ελεγκτή PID, ο οποίος παρουσιάζεται παρακάτω.

Ο ελεγκτής όπως αναφέρθηκε και παραπάνω αποτελείται από τρεις ελέγχους, τον αναλογικό (P), τον ολοκληρωτικό (I) και τον διαφορικό (D) κάθε ένας από τους οποίους είναι υπεύθυνος για μια διαφορετική λειτουργία. Με τον συνδυασμό και των τριών επιτυγχάνεται η επιθυμητή απόκριση.

7.2 Βρόχος ανάδρασης

Η πιο απλή διάταξη ελέγχου εικονίζεται στο παραπάνω σχήμα και πρόκειται για μια απλή επιλογή της επιθυμητής εξόδου (Desired output response) η οποία αποτελεί την είσοδο. Το συγκεκριμένο σήμα περνάει από έναν ελεγκτή (Controller) και από ένα δευτερογενή μηχανισμό (Actuator) προκειμένου να «μεταφραστεί» για να το «κατανοήσει» το βασικό σύστημα της διαδικασίας (Process). Με αυτή τη διαδικασία ελπίζουμε ότι το σύστημα θα συμπεριφερθεί σωστά και θα παράξει μία έξοδο (Output) η

οποία θα είναι αρκετά κοντά στην επιθυμητή εκδοχή της. Με αυτό τον τρόπο όμως αφήνουμε το σύστημα να λειτουργήσει ελεύθερα και δεν παρακολουθούμε τη δράση του. Πιθανότατα καθώς ο χρόνος περνάει θα αποκλίνει σημαντικά από την επιθυμητή εκδοχή δεδομένου πως θα συσσωρεύονται σφάλματα.



Το πρόβλημα αυτό επιλύθηκε με την εισαγωγή ενός εργαλείου που παίζει πολύ σπουδαίο ρόλο στην επιστήμη του ελέγχου των συστημάτων: του βρόχου ανατροφοδότησης. Η εισαγωγή αυτού του εξαρτήματος φαίνεται στο παρακάτω σχήμα.



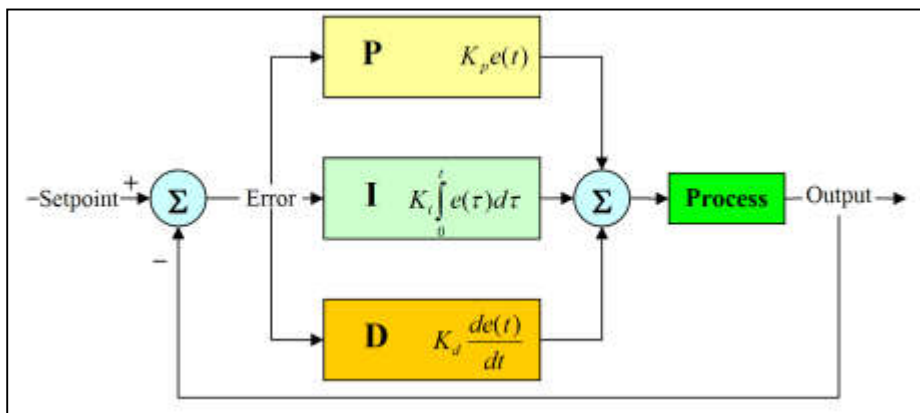
Παρατηρούμε ότι το σήμα της εξόδου Y (Actual output) περνάει από ένα αισθητήριο (Sensor) το οποίο μετράει την τιμή της και την επιστρέφει στην είσοδο του ολικού συστήματος όπου και αφαιρείται από την επιθυμητή έξοδο. Αυτή η διαφορά μεταξύ της ιδανικής εξόδου και της πραγματικής εξόδου αποτελεί την είσοδο στο εσωτερικό σύστημα (σύστημα ανοικτού βρόχου) και συνιστά το σφάλμα. Αν αυτό το σφάλμα έχει μικρή τιμή, σημαίνει πως η κατάσταση είναι κοντά στην επιθυμητή και άρα η αντίδραση από το εσωτερικό σύστημα δε θα είναι σημαντική. Από την άλλη πλευρά αν το σφάλμα έχει μεγάλη τιμή, αυτό δρα ως «συναγερμός» για το σύστημα το οποίο δέχεται ως είσοδο ένα σήμα μεγάλης ισχύος και άρα πρέπει να δράσει έτσι ώστε να το ουδετεροποιήσει. Με αυτό τον τρόπο, έχουμε διαρκώς στραμμένη την προσοχή μας στην έξοδο του συστήματος και όταν η κατάσταση πάει να παρεκτραπεί, η αντίδραση του συστήματος είναι ανάλογη αυτής της παρεκτροπής.

Στα συστήματα αυτομάτου ελέγχου ξεχωριστό ρόλο παίζουν κάποια συγκεκριμένα εξαρτήματα που ονομάζονται ελεγκτές. Αυτά τα υπό-συστήματα ως κύρια δραστηριότητα έχουν να τροποποιούν το σφάλμα μεταξύ της ιδεατής και της

πραγματικής εξόδου έτσι ώστε να γίνει περισσότερο αντιληπτό από το κεντρικό σύστημα και να μηδενίσει (ή να ελαχιστοποιήσει) το τελικό σφάλμα γρηγορότερα.

7.3 Οι όροι PID

Τα αρχικά PID αντιπροσωπεύουν τρεις λέξεις: Proportional (αναλογικός), Integral (ολοκληρωτικός) και Derivative (διαφορικός). Με άλλα λόγια το σφάλμα εισόδου πολλαπλασιάζεται με μία σταθερά (Proportional), ολοκληρώνεται (Integral) και παραγωγίζεται (Derivative). Στη συνέχεια, οι τρεις αυτές ποσότητες αθροίζονται και αποτελούν την πραγματική είσοδο του κυρίως συστήματος.



Στο παραπάνω διάγραμμα η μεταβλητή Σ αντιστοιχεί στην πραγματική τιμή εισόδου, ο όρος error στο σφάλμα, και ο όρος Output στην τιμή της πραγματικής εξόδου. Ως σφάλμα ορίζεται η διαφορά ανάμεσα στην τιμή της επιθυμητής εισόδου και της πραγματικής εξόδου. Το σήμα του σφάλματος θα σταλεί στον ελεγκτή PID ο οποίος θα υπολογίσει την παράγωγο και το ολοκλήρωμα του σήματος. Το σήμα που θα προκύψει ονομάζεται u ή process και μετά τον ελεγκτή είναι ίσο με το αναλογικό κέρδος (K_p) επί την τιμή του σφάλματος, συν το ολοκληρωτικό κέρδος (K_i) επί το ολοκλήρωμα του σφάλματος, συν το διαφορικό κέρδος (K_d) επί την παράγωγο του σφάλματος, δηλαδή:

$$u = K_p e + K_i \int e dt + K_d \frac{de}{dt}$$

Το σήμα αυτό θα σταλεί στο σύστημα προς έλεγχο και στη συνέχεια θα λάβουμε ένα νέο σήμα εξόδου. Η νέα έξοδος θα σταλεί ξανά πίσω στο αισθητήριο για να ανιχνεύσει και αυτό με τη σειρά του το νέο σήμα σφάλματος. Ο ελεγκτής θα πάρει το νέο σήμα σφάλματος και θα υπολογίσει ξανά την παράγωγο και το ολοκλήρωμα και η ίδια διαδικασία θα επαναλαμβάνεται συνέχεια.

Χαρακτηριστικό γνώρισμα του ελεγκτή PID είναι η δυνατότητα χρήσης των τριών όρων ελέγχου, του αναλογικού, του ολοκληρωτικού και του παραγωγικού με τελικό αποτέλεσμα την επίδραση στην έξοδο του ελεγκτή για την εφαρμογή του βέλτιστου ελέγχου. Η παραπάνω εικόνα αποτελεί ένα μπλοκ διάγραμμα όπου παρουσιάζει έναν ελεγκτή PID ο οποίος υπολογίζει συνεχώς μια τιμή σφάλματος ως τη διαφορά μεταξύ μιας επιθυμητής τιμής ρύθμισης και μιας μεταβλητής και έτσι ο ελεγκτής επιχειρεί να ελαχιστοποιήσει το σφάλμα με την πάροδο του χρόνου ρυθμίζοντας μια μεταβλητή ελέγχου.

Στο παραπάνω μοντέλο οι όροι P, I και D έχουν τα εξής χαρακτηριστικά:

- Ο όρος P είναι αναλογικός ως προς το τρέχον σφάλμα. Πρακτικά ένα μεγαλύτερο σφάλμα σημαίνει πιο επιθετική διορθωτική ενέργεια του ελεγκτή. Αν σε ένα σύστημα εφαρμοσθεί μόνο ο όρος P, τότε το σύστημα δεν καταφέρνει να διορθώσει το πρόβλημα με αποτέλεσμα να εμφανίζεται το χαρακτηριστικό της υπέρβασης το οποίο όμως διορθώνεται με τη χρήση των άλλων δύο όρων.
- Ο όρος I, αποτελεί το προηγούμενο σφάλμα το οποίο ενσωματώνεται με την πάροδο του χρόνου. Εάν υπάρξει υπολειπόμενο σφάλμα μετά την εφαρμογή του ο αναλογικός όρος επιδιώκει να το εξαλείψει.
- Ο όρος D είναι αυτός ο οποίος προσδιορίζει μια τιμή σφάλματος που μπορεί να προκύψει στο μέλλον, με βάση τον τρέχοντα ρυθμό μεταβολής. Συνήθως ονομάζεται και «προληπτικός έλεγχος» λόγω της ιδιότητας του.

Η ισορροπία όλων των παραπάνω επιδράσεων επιτυγχάνεται με τη ρύθμιση βρόγχου «loop tuning» και έχει ως σκοπό τη βέλτιστη παραγωγή λειτουργίας ελέγχου. Οι σταθερές συντονισμού παρουσιάζονται με το σύμβολο «K» και πρέπει να εξάγονται για κάθε εφαρμογή ελέγχου καθώς εξαρτώνται από τα χαρακτηριστικά του βρόγχου.

Η συνάρτηση μεταφοράς για τον ελεγκτή τριών όρων είναι η παρακάτω:

$$Kp + \frac{Ki}{s} + Kds = \frac{KDs^2 + KPs + Ki}{s}$$

Όπου οι όροι Kp, Ki και Kd είναι το αναλογικό, το ολοκληρωτικό και το διαφορικό κέρδος.

7.4 Χαρακτηριστικά των όρων P, I και D

Ο κάθε ένας από τους τρεις όρους παίζει διαφορετικό ρόλο σε ένα σύστημα. Ο αναλογικός ελεγκτής (Kp), έχει ως αποτέλεσμα την ελάττωση του χρόνου ανύψωσης και τη μείωση, αλλά ποτέ την εξάλειψη, του μόνιμου σφάλματος. Ο ολοκληρωτικός έλεγχος (Ki) εξαλείφει το μόνιμο σφάλμα, αλλά θα χειροτερέψει την μεταβατική απόκριση (αριθμός των ταλαντώσεων μέχρι την τελική ισορροπία του συστήματος). Ο διαφορικός έλεγχος (Kd) έχει ως αποτέλεσμα την αύξηση της σταθερότητας του συστήματος, μειώνοντας την υπερύψωση και βελτιώνοντας την μεταβατική απόκριση. Τα αποτελέσματα της επίδρασης καθενός από τους ελεγκτές Kp, Kd και Ki σε ένα σύστημα κλειστού βρόγχου, συνοψίζονται στο παρακάτω πίνακα.

Αντίδραση ελεγκτή	Χρόνος ανύψωσης	Υπερύψωση	Χρόνος αποκατάστασης	Μόνιμο σφάλμα
Kp	Μείωση	Αύξηση	Μικρή αλλαγή	Μείωση
Ki	Μείωση	Αύξηση	Αύξηση	Εξάλειψη
Kd	Μικρή αλλαγή	Μείωση	Μείωση	Μικρή αλλαγή

Ας σημειωθεί ότι αυτοί οι συσχετισμοί μπορεί να μην είναι πολύ ακριβείς, επειδή οι ελεγκτές Kp, Kd και Ki αλληλεξαρτώνται. Στην πραγματικότητα, η αλλαγή μίας από αυτές τις μεταβλητές, μπορεί να αλλάξει την επίδραση και των άλλων δύο μεταβλητών. Για το λόγο αυτό, ο παραπάνω πίνακας θα πρέπει να χρησιμοποιείται μόνο ως σημείο αναφοράς όταν προσδιορίζουμε τις τιμές για τους ελεγκτές Kp, Kd, και Ki.

Κεφάλαιο 8

8.1 Κινητήρες

Οι κινητήρες που χρησιμοποιήθηκαν στο όχημα είναι servo κινητήρες (FS90R) συνεχούς περιστροφής της εταιρίας FEETECH. Σε πλήρες φορτίο (6V) έχουν ροπή έως 1,5kg και είναι ικανοί να επιτύχουν πλήρη περιστροφή 130 rpm (rounds per minute). Τα χαρακτηριστικά τους παρουσιάζονται στον παρακάτω πίνακα.



Εικόνα 18

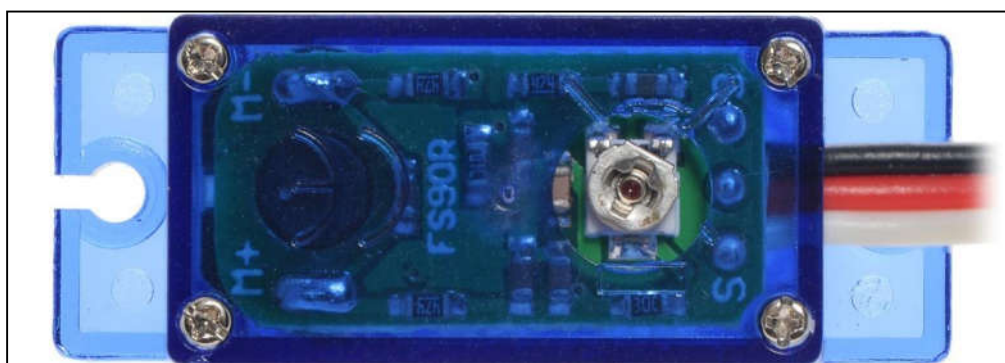
8.2 Χαρακτηριστικά κινητήρα

Πίνακας τεχνικών χαρακτηριστικών Servo Continuous Rotation

α/α	Servo Micro Continuous Rotation (Feetech FS90R)	
1	Ταχύτητα λειτουργίας	110RPM (4.8V)130RPM (6V)
2	Ροπή (StallTorque)	1.3kg.cm/18.09oz.in(4.8V)1.5kg.cm/20.86oz.in(6V)
3	Τάση συστήματος	4.8V~6V
4	Σύστημα ελέγχου	Αναλογικό
5	Κατεύθυνση	CCW
6	Γωνία λειτουργίας	360° μοίρες
7	Παλμός	900us-2100us
8	Τύπος Ρουλεμάν (BearingType)	Κανένα
9	GearType	Πλαστικός
10	Τύπος κινητήρα	Μεταλλικό
11	Μήκος καλωδίου σύνδεσης	20cm
12	Διαστάσεις	23.2 x 12.5 x 22 mm
13	Βάρος	9g

8.3 Συνδεσμολογία με Arduino

Η συνδεσμολογία των servo κινητήρων είναι απλή. Διαθέτουν ένα θηλυκό βύσμα με τρεις υποδοχές. Το ένα καλώδιο (καφέ) συνδέεται στη γείωση (GND), το μεσαίο (κόκκινο) καλώδιο με την τάση του arduino (5V) ή σε μια άλλη παροχή και το τελευταίο καλώδιο (πορτοκαλί) με το σήμα του arduino. Μπορούμε να συνδέσουμε τους κινητήρες και με τις θύρες του arduino, οι οποίες υποστηρίζουν PWM. Δε χρειάζονται κάποιοι driver ή κάποια ενίσχυση σήματος για να λειτουργήσουν. Οι κινητήρες μετατρέπουν τους παλμούς θέσεις σε ταχύτητα συνεχούς περιστροφής. Στο κάτω μέρος του servo κινητήρα, υπάρχει ένα ποτενσιόμετρο με το οποίο μπορούμε να αλλάζουμε το σημείο τερματισμού του κινητήρα.



Ποτενσιόμετρο

8.4 Ρόδες

Οι ρόδες επιλέχθηκαν με βάση τους κινητήρες που χρησιμοποιήσαμε, τις ανάγκες της κατασκευής μας και το μέγεθος του οχήματός μας. Είναι κατασκευασμένες από πλαστικό, ειδικές για τους servo-κινητήρες FS90R. Τα λάστιχα που διαθέτουν είναι κατασκευασμένα από σιλικόνη.

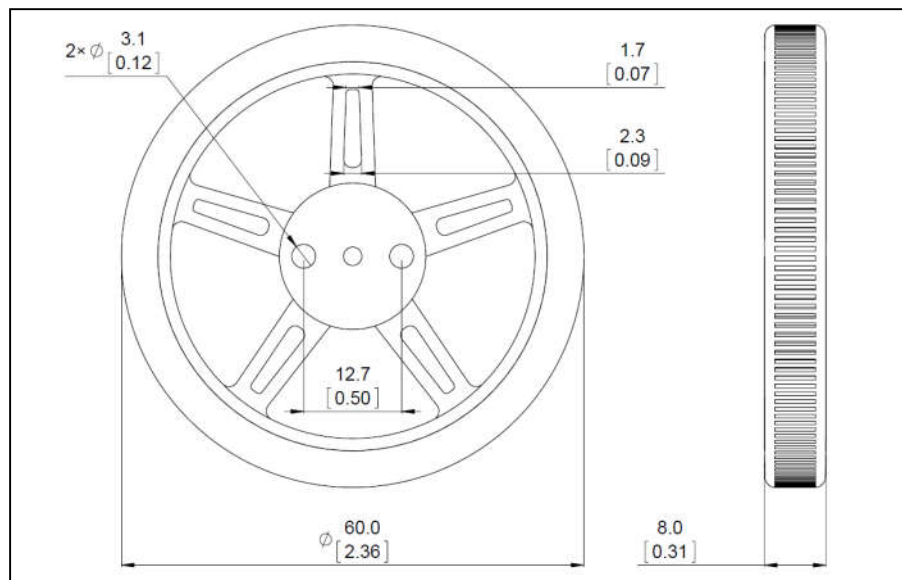


Εικόνα 19

Τα λάστιχα δεν έχουν λεία επιφάνεια, για να μην υπάρχει πρόβλημα ολίσθησης κατά την κίνηση του οχήματος σε λείες επιφάνειες.

Οι ζάντες τους έχουν τρεις υποδοχές για βίδες. Οι υποδοχές είναι δύο μεγεθών, στις δύο πλαϊνές μπορούμε να προσθέσουμε κάποιο σύστημα encoder και στη μεσαία να βιδώσουμε τη ρόδα πάνω στον κινητήρα. Έχουν διάμετρο 2.36cm και πλάτος 0.31cm.

Οι διαστάσεις:



Εικόνα 20

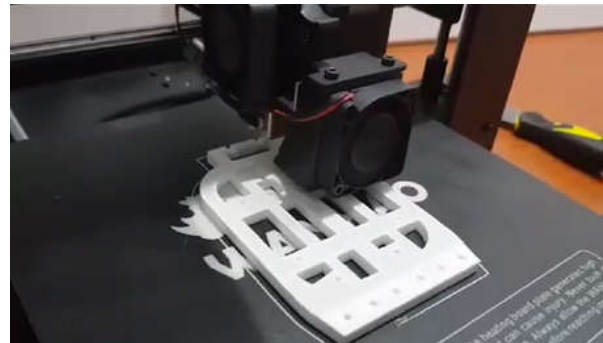
Κεφάλαιο 9

Επιλέξαμε να δημιουργήσουμε μια κατασκευή με 3Dεκτυπωτή λόγω της μορφής του ανάστροφου εκκρεμούς που θέλαμε να παρουσιάσουμε.

9.1 Τι είναι ο 3D εκτυπωτής

Ο 3Dεκτυπωτής χρησιμοποιείται για την κατασκευή τρισδιάστατων στερεών αντικειμένων, τα οποία προέρχονται από ένα ψηφιακό αρχείο. Ο τρόπος εκτύπωσης που επιλέχθηκε ήταν μέθοδος κατασκευής λιωμένου νήματος ή αλλιώς εναπόθεσης υλικού σε διαδοχικές στρώσεις. Το υλικό το οποίο επιλέξαμε για την συγκεκριμένη κατασκευή είναι PLA χρώματος λευκού. Το 3DPLA είναι ένα πολυμερές υλικό το οποίο παράγεται από ανανεώσιμες πηγές όπως το καλαμπόκι. Είναι μη τοξικό, βιοδιασπώμενο και φιλικό προς το περιβάλλον.

Φωτογραφίες κατά τη διάρκεια της εκτύπωσης.



9.2 Ψηφιακό σχέδιο

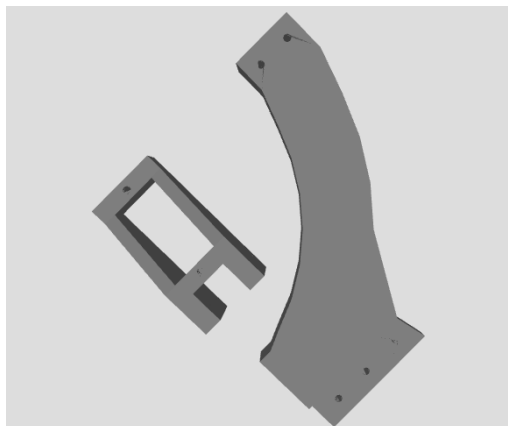
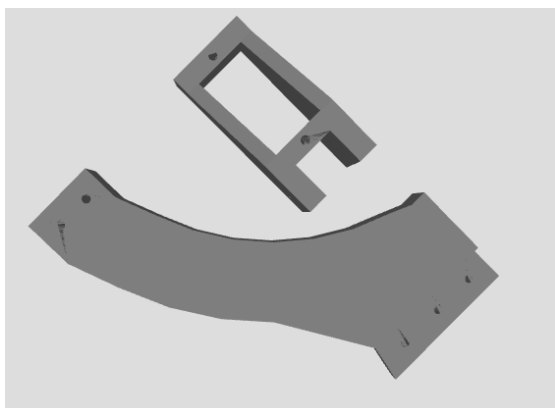
Το σχέδιο επιλέχθηκε με βάση τις ανάγκες της κατασκευής, οι οποίες ήταν οι παρακάτω:

- Να είναι ελαφριά για να μπορεί να κινείται.
- Να είναι διαμορφωμένη έτσι ώστε όλο το βάρος της να βρίσκεται μοιρασμένο στο πάνω μέρος της για να μπορεί να ισχύει η λογική του ανάστροφου εκκρεμούς.

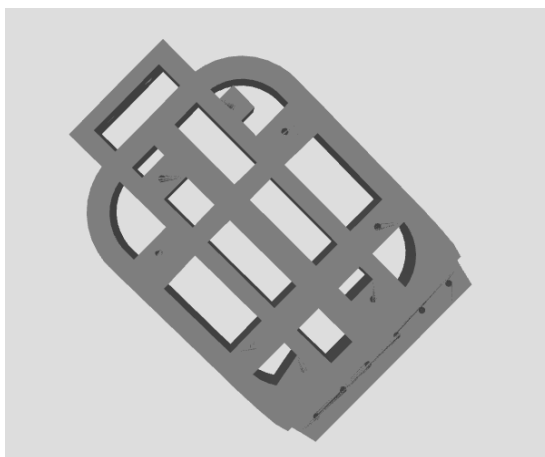
Η κατασκευή χωρίζεται σε τρία κύρια κομμάτια, το δεξί, το αριστερό πόδι και το κύριο σώμα. Τυπώθηκαν επιπλέον δυο κομμάτια τα οποία χρησιμοποιήθηκαν για να μπορέσουν να κουμπώσουν οι κινητήρες.

Παρακάτω παρουσιάζονται τα ψηφιακά αρχεία.

Το δεξί και το αριστερό πόδι της κατασκευής με τις αντίστοιχες βάσεις.



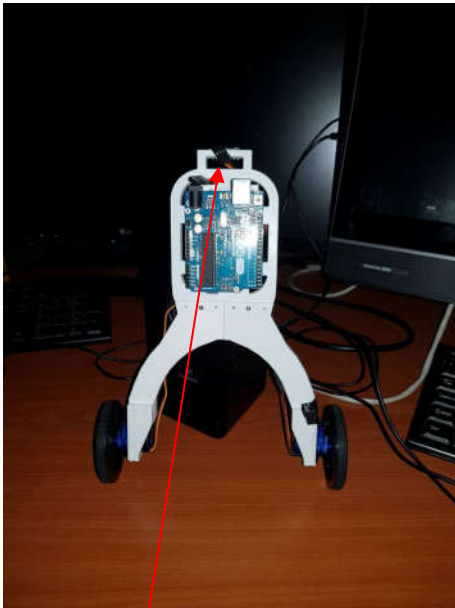
Κύριο σώμα



Το κύριο μέρος της κατασκευής, ή αλλιώς το σώμα επιλέξαμε να εκτυπωθεί με διάτρητη μορφή και όχι συμπαγές για να είναι πιο εύκολη η συνδεσμολογία των εξαρτημάτων που θα τοποθετηθούν πάνω αλλά και για να δημιουργούνται κανάλια για την συνδεσμολογία των καλωδίων. Το μέγεθος το οποίο τυπώθηκε είναι ανάλογο με το μήκος του μικροεπεξεργαστή Arduino.

Όπως παρατηρείτε, στο ψηφιακό σχέδιο υπάρχουν οι απαραίτητες τρύπες για να μπορέσουμε μετά την κατασκευή να βιδώσουμε τα μέλη μεταξύ τους αλλά και οποιοδήποτε άλλο εξάρτημα είναι απαραίτητο.

9.3 Παρουσίαση κατασκευής

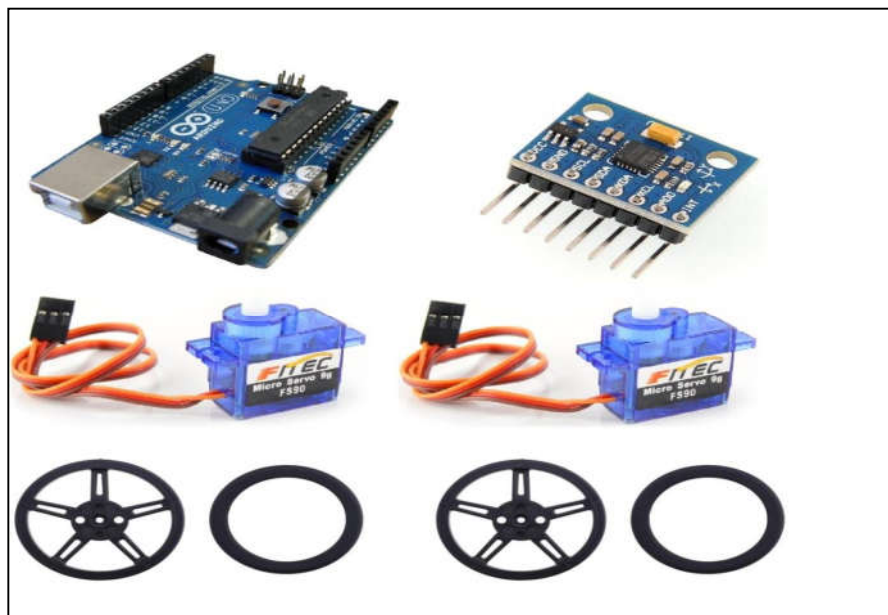


MPU 6050

Στην κορυφή της κατασκευής τοποθετήσαμε το MPU 6050, διότι αυτό παρέχει τη μετάδοση της θέσης στον μικροεπεξεργαστή δίνοντας έτσι ισορροπία.

Στη συνέχεια, από τη μια πλευρά τοποθετήσαμε μια πλακέτα και από πάνω το Arduino, ενώ από την άλλη πλευρά τοποθετήσαμε μια μπαταρία έτσι ώστε να υπάρχει αυτονομία στο ρομπότ. Στο κάτω μέρος έχουν τοποθετηθεί οι κινητήρες του οχήματος.

Εξαρτήματα που τοποθετήθηκαν στην πρώτη φάση της κατασκευής.



Στη δεύτερη φάση της κατασκευής, εφόσον είχαν τοποθετηθεί όλα τα απαραίτητα εξαρτήματα έγινε η επιλογή της μπαταρίας για να μπορεί να κινηθεί με αυτονομία το ρομπότ. Η επιλογή της μπαταρίας έγινε με βάση δύο παραμέτρους, τις ανάγκες φόρτισης των εξαρτημάτων αλλά και το μέγεθος αυτής καθώς ο χώρος που θα μπορούσε να τοποθετηθεί είναι συγκεκριμένος. Η μπαταρία που επιλέχθηκε είναι επαναφορτιζόμενη.



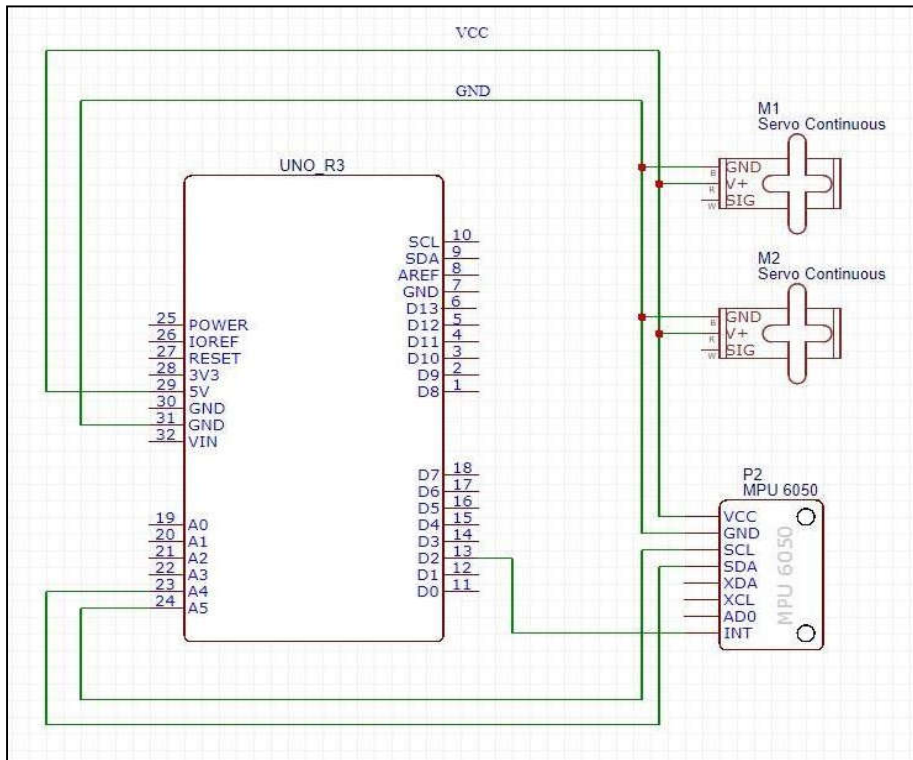
Κεφάλαιο 10

10.1 Συνδεσμολογία κατασκευής

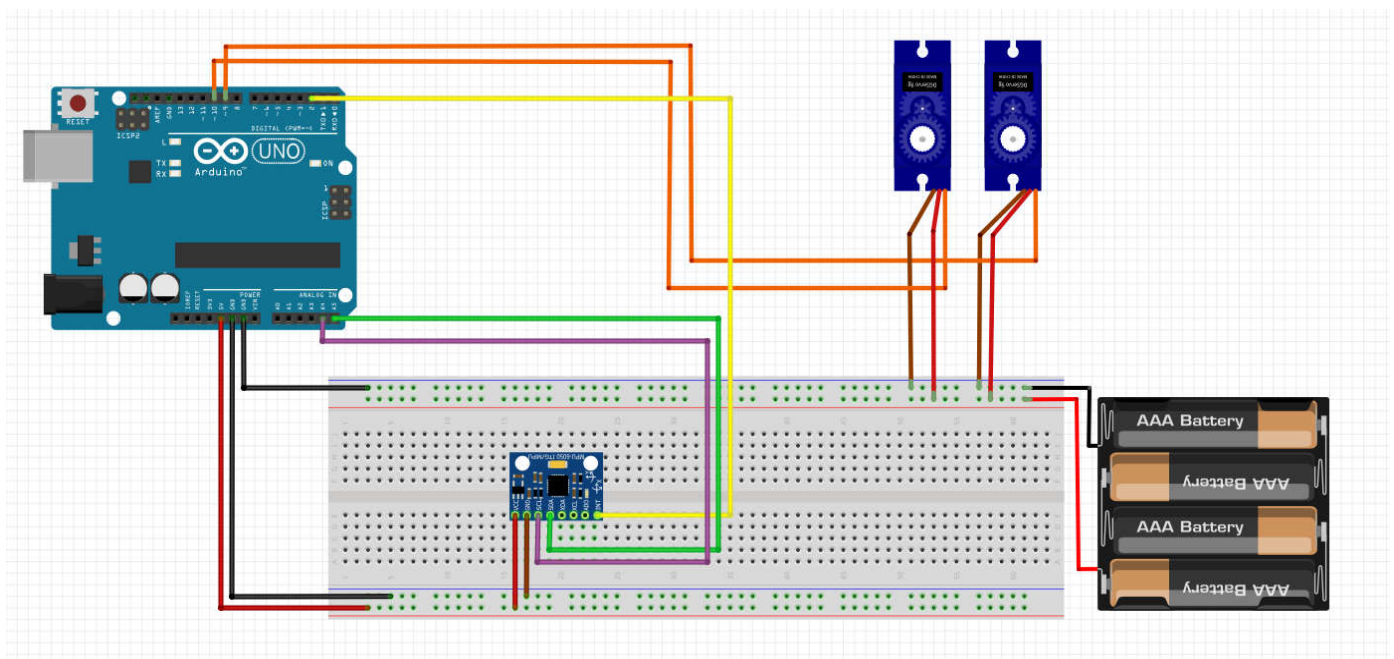
Πραγματοποιήσαμε την συνδεσμολογία της κατασκευής στο διαδικτυακό πρόγραμμα easy Dea και στο circuit.io.

Παρακάτω ακολουθούν το σχηματικό διάγραμμα και το σχέδιο της κατασκευής.

Σχηματικό Διάγραμμα



Σχέδιο κατασκευής



11.1 Κώδικας

```
#include<Wire.h> //επικοινωνία με I2C συσκευές
#include<Servo.h> //βιβλιοθήκη servo κινητήρων
#include "MPU6050_6Axis_MotionApps20.h" //βιβλιοθήκη για το MPU6050
```

```
#define STD_LOOP_TIME 10
#define MAXPID 6000
```

} Ορίζουμε τιμές στον χρόνο του βρόγχου και στον PIDελεγκτή

```
Float delta Time;
unsigned long loop Start Time = 0;
unsigned long start Time;
```

```
float Kp = 385;
float Ki = 43;
float Kd = 670;
```

} Τιμές του PID

```
float target Angle = 1.7;
float angle Offset To Start = 3;
float previous Error = 0;
float pid Value = 0;
float i Limit = 2000;
```

} Ορίζουμε επιθυμητή γωνία ισορροπίας, αρχικοποιούμε κάποιες τιμές

```
Servo wheel Left;
Servo wheel Right;
MPU6050 mpu;
```

```
Bool dmp Ready = false; // TRUEόταν η σύνδεση με τοMPU6050 είναι πετυχημένη
uint8_t mpuInt Status; // κρατάει interrupt status byte από το MPU
uint8_t devStatus; // επιστρέφει την κατάσταση της συσκευής (0 = success, !0 = error)
uint16_t packet Size; // DMP(digital motion processor) μέγεθος (by default είναι 42
```

```

bytes)
uint16_t fifoCount; // μετράειόλατα bytes του FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// μεταβλητέςκίνησης

Quaternionq; // [w, x, y, z] ορίζουμε τέσσερις μεταβλητές
VectorFloatgravity; // [x, y, z] διάνυσμα βαρύτητας
floateuler[3]; // [psi, theta, phi] στοιχεία Euler γωνίας
floatypr[3]; // [yaw, pitch, roll] yaw/pitch/roll και διάνυσμα βαρύτητας

enumRobotState { rsInit, rsReady, rsBalancing };
RobotStatestate = rsInit;
floatcurrentAngle;

charbtData = 0; //μεταβλητή για αποθήκευση των δεδομένων που λαμβάνουμε
boolbtOn = false;

voidsetup() {
startTime = millis(); //χρόνοςσεmilliseconds
Serial.begin(115200); // baudrate σειριακής επικοινωνίας
SetupMpuDMP();

void Print(){
Serial.print("Angle: ");
Serial.print(currentAngle);
Serial.print(" | ");
Serial.print("PID: ");
Serial.print(pidValue);
Serial.print(" | ");
Serial.print("DT: ");
Serial.print(deltaTime);
Serial.print(" | ");
Serial.print("state: ");
Serial.print(state);
Serial.println();
}

```

Εμφάνιση τιμών όπως γωνία, τιμές PID, χρόνος delta και κατάσταση συσκευής

```
}
```

```
volatile bool mpuInterrupt = false; // ανιχνεύει αν το interrupt pin του MPU έχει τιμή HIGH
```

```
void dmpDataReady()
```

```
{
```

```
  mpuInterrupt = true;
```

```
}
```

```
void setupMpuDMP(){
```

```
  Wire.begin(); // επικοινωνία με I2C bus (η βιβλιοθήκη δεν το κάνει αυτόματα)
```

```
  TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz)
```

```
  while (!Serial); // ξεκινάει η σειριακή επικοινωνία
```

```
  // αρχικοποίηση συσκευής
```

```
  Serial.println(F("Initializing I2C devices..."));
```

```
  mpu.initialize();
```

```
  // επιβεβαίωση σύνδεσης
```

```
  Serial.println(F("Testing device connections..."));
```

```
  Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") :
```

```
  F("MPU6050 connection failed"));
```

```
  Serial.println(F("Initializing DMP..."));
```

```
  devStatus = mpu.dmpInitialize();
```

} φόρτωση και αρχικοποίηση του επεξεργαστή

```
  mpu.setXGyroOffset(65);
```

```
  mpu.setYGyroOffset(-32);
```

```
  mpu.setZGyroOffset(98);
```

```
  mpu.setXAccelOffset(-912);
```

```
  mpu.setYAccelOffset(2817);
```

```
  mpu.setZAccelOffset(1171);
```

} Ορίζουμε τις ανάλογες τιμές που λάβαμε από το calibration του MPU

```

// αν η παραπάνω διαδικασίες ολοκληρωθούν επιτυχώς επιστρέφει 0

if (devStatus == 0) {
  // ξεκινάει η λειτουργία του DMP
  Serial.println(F("Enabling DMP..."));
  mpu.setDMPEnabled(true);
  // ενεργοποίηση ανίχνευσης διακοπών

  Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));

  attachInterrupt(0, dmpDataReady, RISING);

  mpuIntStatus = mpu.getIntStatus();

  // δίνει σήμα ότι το DMP είναι έτοιμο για χρήση από το main loop()
  Serial.println(F("DMP ready! Waiting for first interrupt..."));

  dmpReady = true;

  // δέχεται το μέγεθος του DMP για μελλοντική χρήση
  Packet Size = mpu.dmpGetFIFOPacketSize();

  } else {
    // μηνύματα λάθους

    // 1 = initial memory load failed

    // 2 = DMP configuration updates failed

    Serial.print(F("DMP Initialization failed (code ");
    Serial.print(devStatus);
    Serial.println(F(")"));
  }
}

Float Get Angle DMP(){
  // αν το πρόγραμμα δε λειτουργήσει, δε γίνεται τίποτα
  if (!dmpReady) return;

  // αλλιώς περιμένει για interrupt από το MPU και το μέγεθος του καταχωρητή

```

```

while (!mpuInterrupt && fifoCount < packetSize) {
    }

    // κάνει reset το interrupt flag και λαμβάνουμε INT_STATUS byte
mpuInterrupt = false;

mpuIntStatus = mpu.getIntStatus();
fifoCount = mpu.getFIFOCount();

// έλεγχος για overflow
if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    // ανυπάρχει overflow, κάνει reset
mpu.resetFIFO();
Serial.println(F("FIFO overflow!"));
}
else if (mpuIntStatus & 0x02) {
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount()

    // λαμβάνει το μέγεθος του πακέτου από τον FIFO
mpu.getFIFOBytes(fifoBuffer, packetSize);

    // αν είναι διαθέσιμο παραπάνω από ένα πακέτο
fifoCount -= packetSize;

    // γωνίες Euler σε μοίρες
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
}
return (ypr[1] * -180/M_PI);
}

float PID() {

```

```
float error = targetAngle - currentAngle; //υπολογισμόςσφάλματος
```

```
floatproportional = error*Kp;
```

```
static float integral = 0;
```

```
integral += error * Ki;
```

```
if(integral >iLimit) integral = iLimit;
```

```
if(integral < -iLimit) integral = -iLimit;
```

```
float derivative = (error - previousError) * Kd;
```

Πράξεις ανάλογα με το στοιχείο του ελεγκτή

```
float result = proportional + integral + derivative; //άθροιση σφάλματος
```

```
previousError = error;
```

```
return result;}
```

```
void loop() {
```

```
currentAngle = GetAngleDMP(); //γωνία οχήματος
```

```
deltaTime = millis() - loopStartTime; //υπολογισμόςdelta time
```

```
pidValue = PID()* -1; //τιμέςτουPID
```

```
//επικοινωνία με bluetooth
```

```
if(Serial.available() > 0)
```

```
{
```

```
btData = Serial.read();
```

```
Serial.print(btData);
```

```
Serial.print("\n");
```

```
if(btData == '1') //αν σταλεί «1»
```

```
btOn = true; //κάνετο btOn True
```

```
else if(btData == '0') //ανσταλεί «0»
```

```
btOn = false; //κάνετοbtOn False
```

```
}
```

```

if ((state == rsInit) && (millis() - startTime > 7000))
{
state = rsReady;
}
elseif (state == rsReady)
{ //αν έχει πατηθεί το «1»
if (btOn == true)
{
//υπολογισμός γωνίας και ενέργειες για επίτευξη ισορροπίας
if ((currentAngle > targetAngle - angleOffsetToStart) && (currentAngle < targetAngle +
angleOffsetToStart))
{
wheelRight.attach(9); //σύνδεση του δεξιού κινητήρα
wheelLeft.attach(10); //σύνδεση του αριστερού κινητήρα
state = rsBalancing; //κατάσταση ισορροπίας
}}
else if (state == rsBalancing)
{
//αν έχει πατηθεί το «0»
MoveMotors();
if (btOn == false)
{state = rsReady;
wheelRight.detach();
wheelLeft.detach();} //απενεργοποίηση κινητήρων
loopStartTime = millis();
Print();}

```



```

voidMoveMotors()
//εφόσονέχειπατηθείτο «1»
{
floatmotorSpeed = map(pidValue, MAXPID * -1, MAXPID, 0, 180);
wheelRight.write(motorSpeed);
wheelLeft.write(map(motorSpeed, 0, 180, 180, 0));
}
}

```

Ορίζουμε τιμές στους κινητήρες μετά από τους υπολογισμούς του PID και των σφαλμάτων

11.2 Σχόλια Κώδικα

Όπως προκύπτει και από τον παραπάνω κώδικα, οι τιμές που εντοπίζαμε για τους όρους PID είναι οι εξής:

$K_p = 385$

$K_i = 43$

$K_d = 670$

Θέσαμε ως target angle ή αλλιώς ως επιθυμητή γωνία την 1.7 διότι πειραματικά παρατηρήσαμε πως στη συγκεκριμένη γωνία το ρομπότ ισορροπεί. Οι κινητήρες ενεργοποιούνται αφού το ρομπότ σηκωθεί και περάσει τη γωνία των 3°.

Στο παρακάτω κομμάτι γίνεται ο υπολογισμός του PID ελεγκτή, που όπως παρατηρούμε για να μπορέσουμε να βρούμε το σφάλμα αφαιρούμε από την πραγματική γωνία την επιθυμητή. Στη συνέχεια κάνουμε τους υπολογισμούς για τους τρεις όρους P, I και D με βάση τη θεωρία.

```
float error = targetAngle - currentAngle;
```

```
//if(Kp < 0) Kp = 0;
```

```
float proportional = error * Kp;
```

```
//if(Ki < 0) Ki = 0;
static float integral = 0;
integral += error * Ki;
if(integral > iLimit) integral = iLimit; // limit wind-up
if(integral < -iLimit) integral = -iLimit;

//if(Kd < 0) Kd = 0;
float derivative = (error - previousError) * Kd;

float result = proportional + integral + derivative;
previousError = error;
return result;
```

Κεφάλαιο 12

Συμπεράσματα

Για την κατασκευή του self-balancing robot στηριχθήκαμε στη φιλοσοφία της λειτουργίας του ανάστροφου εκκρεμούς. Για την υλοποίησή της ήταν απαραίτητη η προσθήκη αισθητηρίου που μας έδινε τα στοιχεία που χρειαζόνταν για να διατηρεί σταθερό τον προσανατολισμό της η συσκευή. Για το λόγο αυτό επιλέχθηκε το αισθητήριο MPU και συγκεκριμένα το MPU 6050 το οποίο μας έδινε τιμές για τρεις άξονες, το x, y και το z, δηλαδή το γυροσκόπιο, το επιταχυνσιόμετρο και το αξελερόμετρο. Αφού αντλήσαμε τα στοιχεία που χρειαζόμαστε από το MPU 6050, χρησιμοποιήσαμε έναν ελεγκτή στον οποίο στέλναμε τα στοιχεία αυτά και εκείνος αναλάμβανε την ισορροπία του ρομπότ. Έτσι επιλέξαμε να χρησιμοποιήσουμε τον PID ελεγκτή.

Το Arduino που χρησιμοποιήθηκε, επιλέχθηκε αρχικά λόγω του μεγέθους του και κατά δεύτερον λόγω της πλακέτας που συγκεντρώνει όλα τα I/O που θέλαμε να χρησιμοποιήσουμε. Επίσης ο επεξεργαστής του Arduino UNO μπορεί να ανταπεξέλθει στις απαιτήσεις της συγκεκριμένης εργασίας. Πιθανότατα και οποιοδήποτε άλλο Arduino αν είχε χρησιμοποιηθεί να οδηγούσε στο ίδιο αποτέλεσμα με κάποιες τροποποιήσεις όμως στην κατασκευή.

Επιλέχθηκαν servo motors και συγκρίνοντας την κατασκευή του self-balancing robot με άλλες παρόμοιες κατασκευές, καταλήγουμε στο συμπέρασμα ότι θα μπορούσαν να είχαν χρησιμοποιηθεί DC motors με τις αντίστοιχες ρόδες που υπάρχουν για αυτού του είδους τα motors και μια γέφυρα L293D.

Συγκρίνοντας επίσης ένα μεγάλο αριθμό από self-balancing robot παρατηρήσαμε πως για το 90% των υλοποιημένων κατασκευών επιλέχθηκαν μοντέλα από 3D printer. Καταλήγουμε έτσι στο συμπέρασμα ότι τα συγκεκριμένα υλικά που χρησιμοποιούνται για την κατασκευή τέτοιων μοντέλων είναι τα πλέον κατάλληλα διότι είναι αρκετά ελαφριά συγκριτικά με οποιοδήποτε άλλο υλικό.

Στην αρχή της κατασκευής, χρησιμοποιήσαμε δύο κομμάτια plexiglas, για να μπορούσαμε να δημιουργήσουμε μια κατασκευή με δύο ορόφους στους οποίους θα μοιράζαμε τα εξαρτήματά της, το Arduino, τις μπαταρίες, την πλακέτα για τις συνδεσμολογίες κ.ά. Παρατηρήθηκε όμως ότι το ρομπότ δεν μπορούσε να ισορροπήσει για τους παρακάτω τρεις λόγους, όπως διαπιστώσαμε στην πράξη.

- Πρώτον το βάρος των Plexiglas ήταν μεγαλύτερο από το βάρος του 3DPLA που χρησιμοποιήθηκε.
- Δεύτερον το βάρος όλων των εξαρτημάτων βρισκόταν μοιρασμένο και όχι στο υψηλότερο σημείο της κατασκευής, όπως δηλαδή λειτουργεί το ανάστροφο εκκρεμές.
- και τρίτον το αισθητήριο που μας έδινε τη θέση που βρισκόταν το ρομπότ και ήταν υπεύθυνο για τη σωστή λειτουργία του, έπρεπε να βρίσκεται στην υψηλότερη θέση του ρομπότ πάνω από όλα τα υπόλοιπα εξαρτήματα που χρησιμοποιήθηκαν.

Υλοποιήσαμε λοιπόν τη συγκεκριμένη κατασκευή γνωρίζοντας ότι θα μπορούσαν να υπάρξουν επιπλέον λειτουργίες, όπως ο έλεγχος μέσω εφαρμογής, ώστε να μπορεί ο χρήστης να κινεί το ρομπότ, μετακινώντας το προς όλες τις κατευθύνσεις (δεξιά, αριστερά, εμπρός και πίσω). Θέσαμε όμως κάποιους περιορισμούς στις απαιτήσεις μας, διότι η κατασκευή μας θα έπρεπε να τυπωθεί πιο ελαφριά (τα μέρη της να μην έχουν γέμισμα), φοβούμενοι ότι θα κινδυνεύαμε να χαθεί η ισορροπία που έχει τώρα. Επιπλέον για τα motors που χρησιμοποιήσαμε, είχαμε αμφιβολίες αν θα μπορούσαν να αντέξουν τις συχνές μεταβολές, κυρίως όταν το ρομπότ θα έπρεπε να στρίψει δεξιά και αριστερά, διότι το συγκεκριμένο μοντέλο δεν είναι αρκετά σταθερό και αξιόπιστο κατασκευαστικά. Οι συχνές μεταβολές λοιπόν θα είχαν ως αποτέλεσμα από ένα σημείο και μετά να προκαλέσουν πρόβλημα σωστής λειτουργίας στα motors, τα οποία θα έπρεπε να βιδωθούν ξανά για να γίνει κάποιο καλιμπράρισμα.

Θα μπορούσε επίσης να υπάρξει προσθήκη αισθητηρίων. Ένα αισθητήριο για αποφυγή εμποδίων ή ένα αισθητήριο το οποίο θα διάβαζε και θα εμφάνιζε τη θερμοκρασία στην κατασκευή μας. Οι λόγοι που δεν προστέθηκε ένα αισθητήριο αποφυγής εμποδίων συμπίπτουν με τους λόγους που αφορούν στη μη χρήση ελέγχου κίνησης μέσω εφαρμογής, διότι όταν το ρομπότ έβρισκε κάποιο εμπόδιο θα αναγκαζόταν να στρίψει προκαλώντας τα προβλήματα λειτουργίας που προαναφέρθηκαν.

Μελετώντας τέλος τον ελεγκτή PID αντιλαμβανόμαστε τη χρησιμότητα που θα μπορούσε να έχει μελλοντικά σε κατασκευές που επιδιώκουν την εξισορρόπηση και τον μηδενισμό σφάλματος. Ένα τέτοιο παράδειγμα είναι οι συσκευές που υπάρχουν για να βοηθούν τα άτομα που πάσχουν από Parkinson. Συσκευές δηλαδή οι οποίες με τα κατάλληλα αισθητήρια μπορούν να αντιληφθούν μεταβολή κίνησης και να μηδενίσουν αυτό το σφάλμα παρέχοντας σωστή ισορροπία.

Βιβλιογραφία

1. Ο μικροεπεξεργαστής Arduino, διαθέσιμο στον δικτυακό τόπο
<https://goo.gl/tMZ1kA>
2. Processing, γλώσσα προγραμματισμού, διαθέσιμο στον δικτυακό τόπο:
<https://goo.gl/TMA93L>
3. Κινητήρες Servo, διαθέσιμο στον δικτυακό τόπο:
<http://www.feetechrc.com/product/analog-servo/micro-1-3kg-cm-360-degree-continuous-rotation-servo-fs90r/>
4. Λάστιχα, διαθέσιμο στον δικτυακό τόπο:
<https://www.pololu.com/product/2819>
5. Εισαγωγή στους ελεγκτές PID, βρίσκονται στον δικτυακό τόπο:
<http://gun.teipir.gr/DSAELAB/Ergastiriakes/pidtutorial.pdf>
6. PID Controller, βρίσκεται στον δικτυακό τόπο:
https://en.wikipedia.org/wiki/PID_controller
7. Νικόλαος Καραμπετάκης. «Μοντέρνα Θεωρία Ελέγχου. Ενότητα 16. Ανάστροφο εκκρεμές (ανάδραση κατάστασης)». Έκδοση: 1.0. Θεσσαλονίκη 2014.
8. Χαρακτηριστικά PID ελεγκτών, βρίσκεται στον δικτυακό τόπο:
<http://gun.teipir.gr/DSAELAB/Ergastiriakes/pidtutorial.pdf>
9. Ανάστροφο εκκρεμές, βρίσκεται στον δικτυακό τόπο:
<https://opencourses.auth.gr/courses/OCRS431/>
10. 3D Εκτυπωτής, βρίσκεται στον δικτυακό τόπο:
<https://www.3dexpert.gr/main/3dprinters-intro/>
11. Υλικό 3D εκτυπωτή, βρίσκεται στον δικτυακό τόπο:
<https://3dedge.eu/el/filaments/185-631-3dedge-pla-filament-175mm-1kg.html#/5->

